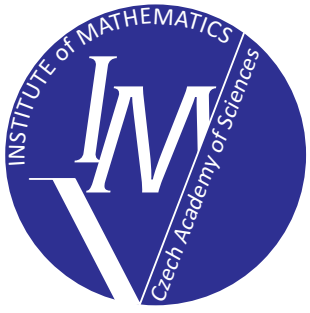
Programs and Algorithms of Numerical Mathematics 21

Jablonec nad Nisou, June 19–24, 2022

Proceedings of Seminar

Edited by
J. Chleboun, P. Kůs, J. Papež
M. Rozložník, K. Segeth, J. Šístek



Institute of Mathematics
Czech Academy of Sciences
Prague 2023

ISBN 978-80-85823-73-8
Matematický ústav AV ČR, v. v. i.
Praha 2023

Contents

Preface

These proceedings contain peer-reviewed papers that are based on the invited lectures, short communications, and posters presented at the 21st seminar Programs and Algorithms of Numerical Mathematics (PANM) held in Merkur Hotel, Jablonec nad Nisou, Czech Republic, June 19–24, 2022.

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, and with the financial support provided by the RSJ Foundation. 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, Prague, and Hejnice in the period 1983–2020. The objective of this series of seminars is to provide a forum for presenting and discussing advanced topics in numerical analysis, computer implementation of numerical algorithms, new approaches to mathematical modeling, and single- or multi-processor applications of computational methods.

The attendance, 70 participants, was the highest in the history of the seminar. Most of the participants came from Czech universities and from institutes of the Czech Academy of Sciences, several also from abroad. We appreciate the traditional participation of a number of young scientists, PhD students, and also some undergraduate students. We wish to believe that also those, who took part in the PANM seminar for the first time, have found the atmosphere of the seminar friendly and working, and will join the PANM community.

The conference photo is taken in front of the Merkur Hotel that hosted the seminar. Unexpected situation with the war and Ukrainian refugees forced the organizers to search for a new location quite shortly before the event. We are grateful that premises of the hotel allowed PANM21 to happen. We enjoyed Mšeno Reservoir, one of the attractions in Jablonec nad Nisou, and held on its bank a welcome drink (in Základna stall) and the traditional evening with a barbecue at Volt Restaurant and Brewery.

The organizing committee consisted of Jan Chleboun, Pavel Kůs, Jan Papež, Petr Přikryl, Miro Rozložník, Karel Segeth, Jakub Šístek, and Tomáš Vejchodský. Ms Hana Bílková kindly prepared manuscripts for the electronic version of the book and for print.

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

Editors

Programs and Algorithms of Numerical Mathematics 21
J. Chleboun, P. Kůs, J. Papež, M. Rozložník, K. Segeth, J. Šístek (Eds.)
Institute of Mathematics CAS, Prague 2023

VISUALISATION OF THE ELECTROMAGNETIC VECTOR FIELDS

Stanislav Bartoň

Opole University of Technology
Faculty of Electrical Engineering, Automatic Control and Informatics
Prószkowska 76 Street, 45-758 Opole, Poland
s.barton@po.edu.pl

Abstract: Modern computer algebra software can be used to visualize vector fields. One of the most used is the Maple program. This program is used to visualize two and three-dimensional vector fields. The possibilities of plotting direction vectors, lines of force, equipotential curves and the method of colouring the surface area for two-dimensional cases are shown step by step. For three-dimensional arrays, these methods are applied to various slices of three-dimensional space, such as a plane or a cylindrical surface. Finally, the temporal evolution of the vector fields is illustrated by animations based on the above methods. In contrast to the publication [2], which deals only with the problem of colouring vector fields, the present paper makes a completely comprehensive study of the problem, including the representation of vectors in a predefined network, the computation of the shape of power lines, and the animation of time changes, including the animation of the coloured vector fields.

Keywords: vector field, visualisation, coloring, line of force, animation, Maple

MSC: 65D18, 65L06, 26B15, 34K28, 78A30

1. Introduction

Visualising vector fields is a handy tool that allows an easy and obvious presentation of their basic properties. All the visualisations presented in this article are made using Maple. Because it is a very complex problem it is solved step by step. In some parts, the algorithms presented could be shortened, but at the cost of lower clarity.

The calculation of the magnetic induction vector is quite complex, and the analytical relationship can only be derived for the elementary shapes of the conductor - a line segment or a circular arc. For more complex conductor shapes, a numerical solution is required because a complicated integral must be calculated along the curve that corresponds to the conductor shape.

DOI: 10.21136/panm.2022.01

2. Magnetic field inside the stator of a three-phase electric motor

The magnetic induction vector inside the motor is given by the vector sum of the magnetic inductions that form the individual phase coils of the stator. Suppose we observe the magnetic induction in the longitudinal plane of symmetry of the stator. In that case, it is possible to neglect the influence of the parts of the coils that are wound parallel to the stator bases. The resulting magnetic field is then given by the sum of the magnetic fields formed by the six straight conductors evenly spaced around the stator circumference.

To simplify the calculation, assume that the length of the stator is much larger than its radius, which we choose as the unit of length. We choose a coordinate system with the origin at the centre of the stator axis, the x axis oriented horizontally, the y axis oriented vertically and the z axis identical to the stator axis.

The magnetic induction vector $\vec{B}$ of a linear conductor is calculated according to the Biot-Savart law

$$\vec{B} = \frac{\mu\, J}{4\pi} \int_C d\vec{w} \times \frac{\vec{p} - \vec{r}_w}{|\vec{p} - \vec{r}_w|^3}$$

where μ =permeability, J =current flowing through the conductor, $d\vec{w}$ =current element of the conductor, $\vec{p}$ =position vector of the observation point and $\vec{r}_w$ =position vector of the current element of the conductor $d\vec{w}$.

For a very long wires, $L \to \infty$, in X_i, Y_i coordinates, parallel to the z axis, the observation point $\vec{p} = [x, y, 0]$ and choosing the currents $\frac{2\,\mu\,|J_i|}{4\pi} = 1$, $\vec{B}_i$ can be expressed as:

$$\vec{B}_i(p) = \left[\frac{J_i\,(Y_i - y)}{(X_i - x)^2 + (Y_i - y)^2}, \frac{J_i\,(x - X_i)}{(X_i - x)^2 + (Y_i - y)^2}, 0 \right].$$

The simplifying condition for the absolute magnitude of the current combined with the choice of the length unit does not affect the shape of the resulting plots because they are only multiplicative constants.

The currents flowing through the single-phase coils of a three-phase electric motor are offset from each other by one-third of a phase. Each coil has two parts parallel to the stator axis. The first part sends a current in the positive direction of the z-axis and the second part flows back from the negative direction of the z-axis. Thus, six alternating currents with a gradual phase shift of one-sixth of the period flow through the stator,

$$J_i = \sin\left(\omega\, t + \frac{i\, 2\pi}{3}\right), i = 1..6.$$

If we choose one period of the alternating current as the time unit, i.e. $\omega = 2\pi$, this multiplicative constant will not appear in the derived expressions. This option does not affect the visualization or animation of the magnetic field.

The conductors forming the coils are located on the coordinates:

$$X = \left[r, \frac{r}{2}, -\frac{r}{2}, -r, -\frac{r}{2}, \frac{r}{2} \right], \quad Y = \left[0, \frac{r\sqrt{3}}{2}, \frac{r\sqrt{3}}{2}, 0, -\frac{r\sqrt{3}}{2}, -\frac{r\sqrt{3}}{2} \right].$$

If we restrict the study of the magnetic induction vector to the plane perpendicular to the stator symmetry axis, $z \equiv 0$, the resulting magnetic induction vector is

$$\vec{B}_f(x, y, t) = \sum_{i=1}^{6} \vec{B}_i.$$

The function that describes the magnetic induction vector depending on the coordinates x, y and time t we derive in Maple as follows:

```
> restart; with(plots): with(LinearAlgebra):
> b:=[J[i]*(Y[i]-y)/((X[i]-x)^2+(Y[i]-y)^2),
      J[i]*(x-X[i])/((X[i]-x)^2+(Y[i]-y)^2)]:
> X:=[seq(r2*cos(i*Pi/3),i=0..5)]: Y:=[seq(r2*sin(i*Pi/3),i=0..5)]:
> J:=[-seq(sin(2*Pi*t+i*Pi/3),i=0..5)]: Bf:=unapply(sum(b,i=1..6),x,y,t):
```

3. Mathematical foundations of vector field visualization

An ordinary differential equation `deq` gives the magnetic induction field line. This equation has no analytical solution, so it must be solved numerically. For accurate drawing, it is necessary to control `d`, the mutual distances of the points on the power line. We use Newton's iterative procedure to calculate the parameter step `p`, which ensures the desired distance between the last `x1, y1` and new end point `x, y` of the power line. The function `DP` gives the correction to the existing parameter value. `xp, yp` are the derivatives of x and y according to the parameter p, returned by the `NS` numerical procedure solving the `deq` differential equation. The intermediate numeric values of the parameter p are contained in the variable `q`.

```
> deq:={diff(x(p),p)=Bf(x(p),y(p),t)[1],diff(y(p),p)=Bf(x(p),y(p),t)[2],
        x(0)=x0,y(0)=y0}:
> DP:=(x1,y1,x,y,xp,yp,d)->-1/2*(x1^2-2*x1*x+x^2+y1^2-2*y1*y+y^2-d^2)
         /(-xp*x1+xp*x-yp*y1+yp*y):
```

3.1. Initial values of parameters for drawing power lines

The arrows representing the magnetic induction vectors are drawn for the points stored in the `AP` list. As an initial point for drawing the power lines, we choose $2\,$`n` $+ 1$ point `IP0` $\rightarrow$ `IP` lying on a line passing through the centre of the stator and perpendicular to the magnetic induction vector in the stator axis. We stop drawing the power line when its endpoint reaches the stator circumference, `R1` ≥ 1. The distances of the points on the power line are chosen as `d` $= 0.05$. We plot the power lines for 90-time instants of one revolution of the motor, `T`. The stator coils are wound around the stator axis on the radius `r2` $= 1.25$. In reality, the coils are wound on a smaller radius, but this choice will reduce the differences in the length of the arrows plotting the magnetic induction vectors near the inner circumference and the centre of the stator. This will make the magnetic field more homogeneous in the stator.

```
> AP:=[seq(seq([i/12,j/12],i=-12..12),j=-12..12)]:
> AP:=map(u->`if`(add(w^2,w=u)>1,NULL,u),AP): r2:=1.25: d:=0.05: n:=16:
> IP0:=[seq(i/n,i=-n..n)]: T:=[seq(i/90.,i=0..89)]:
```

3.2. Auxiliary variables used in the calculation

The meaning of the other variables used is as follows: `nt` = current value of the time loop step, `Bf00` = vector of magnetic induction in the stator axis, `alpha` perpendicular direction to `Bf00`, `t` = current value of time, `TXY` = list containing the completed power lines for the given time step, `ni` = current value of the initial value loop step, `x0, y0` = initial values for the numerical solution `NS` of the differential equation `deq`, `XY` = list of calculated points of the current force line, `Q` = list of numerical values of the formal parameter p for the points stored in `XY`, `dq` = correction of `q` for the relative distance of the power line points or for the location of the power line end points on the stator circumference, `X, Y` = current position of the power line point, `W` = magnetic induction vectors at `AP` grid points, `GLoF[nt], GLoA[nt]` = graphs for the time `t` showing the internal stator circuit and the power lines, and a graph showing the magnetic induction vectors.

3.3. Variables needed for colouring of vector fields

The colouring of the vector field is done in the grid that is stored in the `TP` variable as a list of rectangles. First, the magnetic induction vectors `BF` are calculated for the rectangle centre points `CP`. The absolute values `AB` and the arguments `Phi` are calculated for these vectors. For each time step, the maximum and minimum value of `AB` are stored in the list `MAB` and `mab`. The following procedure does the actual colouring of the vector field:

```
> read "TP.sav": CP:=map(u->add(w,w=u)/nops(u),TP):  MAB:=[]: mab:=[]:
```

3.4. Main computational loop

```
> for nt from 1 to nops(T) do;                                    # THE TIME LOOP
    t:=T[nt]: TXY:=[]: Bf00:=evalf(Bf(0.,0.,t)):
    alpha:=evalf(argument(Bf00[2]-Bf00[1]*I)):
    IP:=map(u->evalf([u*cos(alpha),u*sin(alpha)]),IP0):
    for ni from 1 to nops(IP) do:              # THE FORWARD, q=-0,02, NODE LOOP
      XY:=[IP[ni]]: x0,y0:=XY[-1][]: NS:=dsolve(deq,{x(p),y(p)},numeric);
      Q:=[0.]; q:=-0.02: R1:=0.:
      while R1<1.0 do:                          # THE NEW POWER LINE POINT LOOP
        dq:=1:
        while abs(dq)>1e-6 do;                             # THE d DISTANCE LOOP
          Xp,Yp:=evalf(Bf(XY[-1][],t))[]: X,Y:=map(u->rhs(u),NS(q)[2..3])[]:
          dq:=DP(XY[-1][],X,Y,Xp,Yp,d): dq:=dq/sqrt(4+dq^2): q:=q+dq:
        end do:
        XY:=[XY[],[X,Y]]; Q:=[Q[],q]: q:=2*Q[-1]-Q[-2]: R1:=sqrt(X^2+Y^2):
      end do:
      q:=Q[-1]: dq:=1:
```

```
      while abs(dq)>1e-6 do;                    # THE END POINT - PERIMETER LOOP
        X,Y:=map(u->rhs(u),NS(q))[2..3][]: Xp,Yp:=evalf(Bf(X,Y,t))[]:
        dq:=DP(0,0,X,Y,Xp,Yp,1): dq:=dq/sqrt(4+dq^2): q:=q+dq:
      end do:
      XY:=[XY[1..-2][],[X,Y]]: TXY:=[TXY[],XY]:
    end do:
    # for ni from 1 to nops(IP) do: ... end do: # THE BACKWARD, q=0.02, NODE LOOP
    # This loop repeats the calculation of the forward loop.
    # The only difference is the choice of the initial value of the parameter q.
    # Therefore, its listing is omitted
    GLoF[nt]:=display({plot([sin(f),cos(f),f=0..2*Pi],color=black,thickness=2),
      plot(TXY,color=red)}): W:=evalf(map(u->Bf(u[],t),AP)):
    GLoA[nt]:=arrow(zip((u,v)->[u,v/25],AP,W),shape=harpoon):
    GLIP[nt]:=plot(IP,style=point,symbol=circle,symbolsize=15,color=blue):
    BF:=map(u->evalf(Bf(u[],t)),CP): AB[nt]:=map(u->abs(u[1]+I*u[2]),BF):
    MAB:=[MAB[],max(AB[nt])]; mab:=[mab[],min(AB[nt])]:
    Phi[nt]:=map(u->argument(u[1]+I*u[2]),BF):
end do:
```

The resulting magnetic induction vector field is shown in Figure 1.

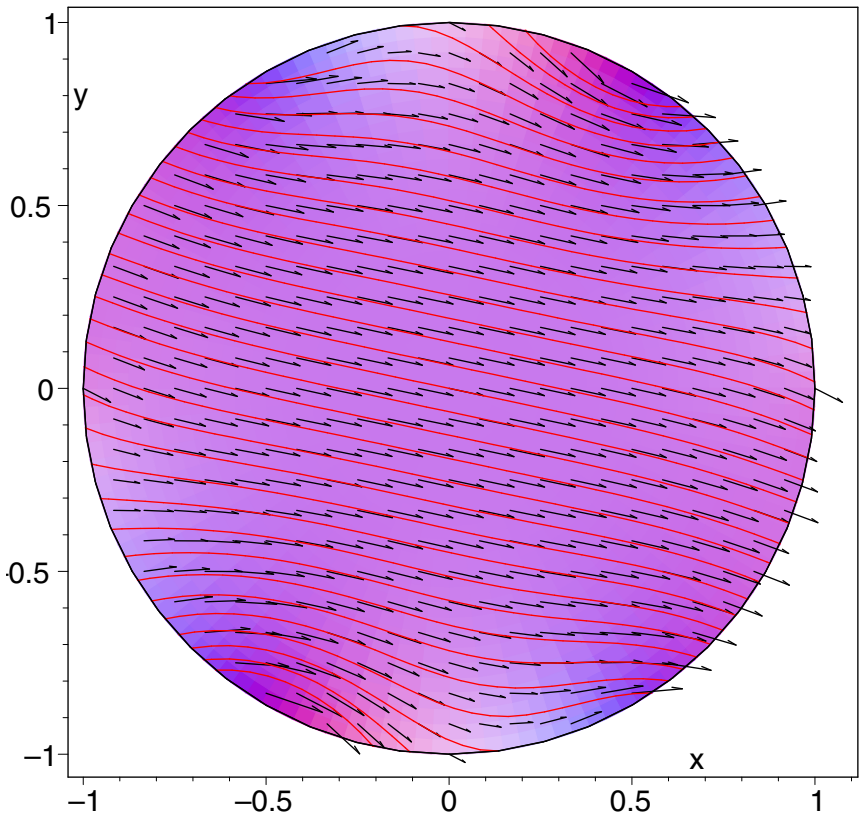


Figure 1: Electromagnetic induction inside stator, `nt` = 49

3.5. Colouring of the vector fields

The colouring of the vector field is done in the grid that is stored in the TP variable as a list of rectangles. First, the magnetic induction vectors BF are calculated for the rectangle centre points. The absolute values AB and the arguments Phi are calculated for these vectors.

To colour the vector field, three functions are needed to mix the fill color of the individual quadrilaterals from TP by selecting the shade of the red, green and blue components. The intensity of each colour component is controlled by the functions, R1, R2 and R3 which depend on the absolute value of the vector AB. The colour ratio depends on the direction of the vector, Phi.

```
> R1:=(av,phi)->evalf(1-(sin(phi)+1)*av/2):
> R2:=(av,phi)->evalf(1-(sin(phi+2*Pi/3)+1)*av/2):
> R3:=(av,phi)->evalf(1-(sin(phi+4*Pi/3)+1)*av/2):
```

Because Min is not equal to zero, it is convenient to choose 0.25 as the lowest intensity value, when the difference in hue is already noticeable. The upper limit of the intensity is 1.

To achieve the identical colouring of the vector fields for all time instants it is necessary to perform a linear transformation LT of the absolute values of the magnetic induction vector, AB. Therefore, the maximum and minimum values of the list MAB are saved in the variables Max and Min. This leads to a higher colour contrast in the resulting image.

```
> Max:=max(MAB): Min:=min(mab): Yh:=1: Yd:=0.25: k:=(Yh-Yd)/(Max-Min):
> q:=Yh-a*Max: LT:=unapply(k*u+q,u):
```

Colouring the vector field in Figure 1 for nt = 49 was done by the following procedure

```
> PAB:=zip((u,v)->[LT(u),v],AB[49],Phi[49]):
> display(zip((u,v)->polygonplot(u,color=COLOR(RGB,R1(v[]),R2(v[]),R3(v[]))),
    TP,PAB),style=patchnogrid,axes=boxed,scaling=constrained);
```

A reference vector field, see Figure 2, is required to properly evaluate the coloured vector field. This figure shows how the colour depends on the absolute magnitude of the vector and its direction. This means that from the system of polygons TP, it is necessary to remove those whose central point CP has a distance from the origin of the coordinate system less than 0.25, TP → TPr, CP → CPr. The remaining polygons are then coloured depending on the absolute size and the position vector argument of the centre point CPr.

```
> TPr:=zip((u,v)->`if`(u[1]^2+u[2]^2>Yd^2,v,NULL),CP,TP):
> CPr:=map(u->`if`(u[1]^2+u[2]^2>0.0625,u,NULL),CP):
> pab:=subs(Float(undefined)=0.,map(u->[abs(u[1]+I*u[2]),
      argument(u[1]+I*u[2])],CPr)):
> CW:=display(zip((u,v)>polygonplot(u,color=COLOR(RGB,R1(v[]),R2(v[]),R3(v[]))),
      TPr,pab),style=patchnogrid,axes=boxed,scaling=constrained):
```

```
> CWL:=plot([[Yd*cos(f),Yd*sin(f),f=0..2*Pi],[Yh*cos(f),Yh*sin(f),f=0..2*Pi]],
      color=[blue,red],thickness=2):
> CWTmin:=textplot([Yd*cos(Pi/4),Yd*sin(Pi/4),convert(round(Min*1000)*.001,
      string)],color=blue,align={LEFT,BELOW},font=[COURIER,BOLD,16]):
> CWTmax:=textplot([Yh*cos(Pi/4),Yh*sin(Pi/4),convert(round(Max*1000)*.001,
      string)],color=red,align={RIGHT,ABOVE},font=[COURIER,BOLD,16]):
> display({CW,CWL,CWTmin,CWTmax}):
```

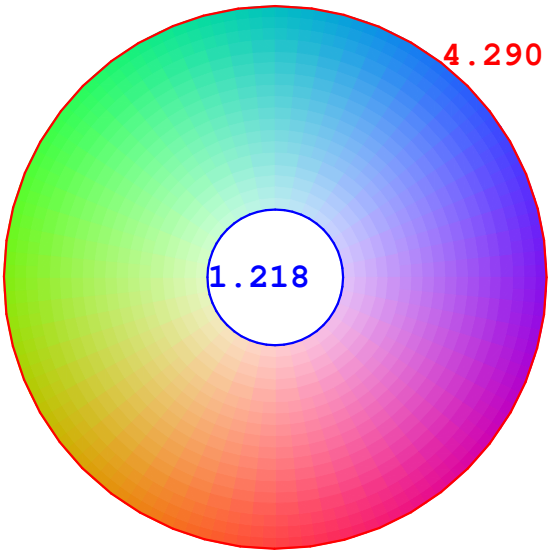


Figure 2: Reference vector field

3.6. Animation

The animation is created by displaying the images sequentially. It is possible to display several images simultaneously at each step of the animation. In the case of the magnetic induction vector visualisation, this will be three sub-images. The arrows indicating the magnitude and direction of the magnetic induction vector AA, the magnetic field lines AL, and the coloured vector field ACF. However, to fit the vector field, it is necessary to perform a linear transformation of the absolute values of the magnetic induction vector AB from the interval $\langle \mathtt{Min}, \mathtt{Max} \rangle$ to $\langle 0.25, 1 \rangle$.

```
> for nt from 1 to nops(T) do:
    PAB:=zip((u,v)->[LT(u),v],AB[nt],Phi[nt]):
    CF[nt]:=display(zip((u,v)->polygonplot(u,color=COLOR(RGB,R1(v[]),
           R2(v[]),R3(v[]))),TP,PAB),style=patchnogrid);
  end do:
> ACF:=display([seq(CF[i],i=1..nops(T))],insequence=true):
> AL:=display([seq(GLoF[i],i=1..90)],insequence=true,axes=boxed,
      scaling=constrained):
```

```
> AA:=display([seq(GLoA[i],i=1..90)],insequence=true,axes=boxed,
      scaling=constrained):
> TACF:=display({AA,AL,ACF}): TACF;
```

Creating the final animation and preparing its display is extremely computationally intensive and can take imately twenty minutes. It takes the same amount of time to free the operating memory after the animation is finished directly in the Maple environment. Therefore, it is preferable to save the final animation as an animated file in `gif` format. It takes less time to create than a Maple animation and can then be displayed at any time independently of Maple. However, it cannot be controlled as Maple allows.

```
> interface(plotdevice=gif,plotoutput="TACF.gif",plotoptions=
    `width=1000,height=1000`):
> TACF;
```

4. Conclusion

The procedures presented here for visualising vector fields offer superior and easy to understand results. However, they must be used judiciously. Firstly, the goal of the visualisation must be defined, and the whole procedure must be subordinated to this. This means deciding whether the visualisation is 2D or 3D, whether to use only arrows to indicate the size and direction of the vectors or whether to use force line drawing, as well as the decision to use area colouring or animation.

In the case of rendering 3D graphs, it must be remembered that graphs containing more complex vector fields are difficult to see. Therefore, it is preferable to display these fields as two-dimensional arrays using several planar slices through the viewed space.

A fundamental issue for drawing power lines is the appropriate choice of their starting points. Force lines may be concentrated in one part of the graph if the initial points are chosen inappropriately, and the resulting display may give a misleading impression. The second issue is the choice of the correct step length – the distance between the points forming the power line. Too small a step length increases the demands on computational capacity. Conversely, too large a step can lead to general detail distortion and loss of scientific accuracy. Therefore, it is always up to the user to tune the procedures published here for individual purposes.

References

[1] Gander, W., Gander, M. J., Kwok, F.: *Scientific Computing. An Introduction using Maple and Matlab.* Springer, 2014.

[2] Bartoň, S.: Color visualisation of the Vector Field. In: M. Ross (Ed.) *5th International Congress on Industrial Applied Mathematics.* UoT Sydney, 2003.

[3] Walker, J.: *Fundamentals of Physics.* John Willey, 2014.

[4] Urone, P., Hinrichs, R.: *College Physics.* OpenStax, 2017.

Programs and Algorithms of Numerical Mathematics 21
J. Chleboun, P. Kůs, J. Papež, M. Rozložník, K. Segeth, J. Šístek (Eds.)
Institute of Mathematics CAS, Prague 2023

REDUCED BASIS SOLVER FOR STOCHASTIC GALERKIN FORMULATION OF DARCY FLOW WITH UNCERTAIN MATERIAL PARAMETERS

Michal Béreš[1,2]

[1] Institute of Geonics, Czech Academy of Sciences
Studentská 1768, Ostrava, Czech Republic
michal.beres@ugn.cas.cz
[2] Department of Applied Mathematics, FEECS, VŠB-TUO
17. listopadu 2172, Ostrava, Czech Republic

Abstract: In this contribution, we present a solution to the stochastic Galerkin (SG) matrix equations coming from the Darcy flow problem with uncertain material coefficients in the separable form. The SG system of equations is kept in the compressed tensor form and its solution is a very challenging task. Here, we present the reduced basis (RB) method as a solver which looks for a low-rank representation of the solution. The construction of the RB consists of iterative expanding of the basis using Monte Carlo sampling. We discuss the setting of the sampling procedure and an efficient solution of multiple similar systems emerging during the sampling procedure using deflation. We conclude with a demonstration of the use of SG solution for forward uncertainty quantification.

Keywords: stochastic Galerkin method, reduced basis method, Monte Carlo method, deflated conjugate gradient method

MSC: 65C05, 65M60, 65M70

1. Introduction

This contribution briefly outlines the solution of stationary Darcy flow problem with uncertain hydraulic conductivity. The solution is obtained using the stochastic Galerkin (GM) method. A significant part of the contribution is the demonstration of the usage of SG solution for forward uncertainty quantification.

The work presented here is a continuation of author's results presented in [1].

2. Stochastic Galerkin method

We start with the problem setting. Let us assume a physical domain $\mathcal{D}$ and random vector $\boldsymbol{Z}$ (on sample space Ω) consisting of M independent standard normal

DOI: 10.21136/panm.2022.02

random variables. We assume the hydraulic conductivity field as a function of both points in domain $\mathcal{D}$ and random vector $\boldsymbol{Z}$, more specifically in the form

$$k(x,\boldsymbol{Z}) = \sum_{m=1}^{M} \underbrace{\chi_{\mathcal{D}_m}(x)}_{k_m^D(x)} \underbrace{\exp(\sigma_m Z_m + \mu_m)}_{k_m^S(\boldsymbol{Z})} = \sum_{m=1}^{M} k_m^D(x)\, k_m^S(\boldsymbol{Z}).$$

I.e. piecewise constant function with the value of constant on each of M subdomains $\mathcal{D}_m$ governed by m-th element of random vector $\boldsymbol{Z}$. The model problem (steady Darcy flow) than takes the form

$$\begin{cases} -\mathrm{div}_x\left(k(x,\boldsymbol{Z})\,\nabla_x u(x,\boldsymbol{Z})\right) = f(x) & \forall\, x \in \mathcal{D}, \boldsymbol{Z} \in \mathbb{R}^M, \\ u(x,\boldsymbol{Z}) = u_0(x) & \forall\, x \in \Gamma_D, \boldsymbol{Z} \in \mathbb{R}^M, \\ -k(x,\boldsymbol{Z})\,\frac{\partial u(x,\boldsymbol{Z})}{\partial n(x)} = g(x) & \forall\, x \in \Gamma_N, \boldsymbol{Z} \in \mathbb{R}^M. \end{cases}$$

For testing purposes, we choose the decomposition into subdomains via thresholding of the Gaussian random field realisation, see Figure 1.

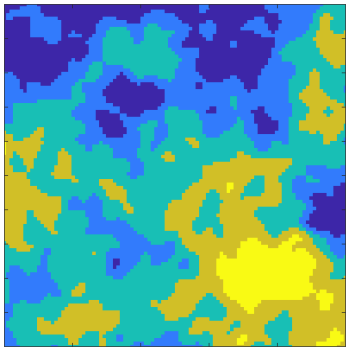


Figure 1: Illustration of decomposition into subdomains

2.1. Stochastic Galerkin matrix equations

The weak form of the problem takes the form

$$\begin{aligned} a(u_H, v) =& b(v), \ \forall v \in L^2\left(\Omega, H^1_{0,\Gamma_D}(\mathcal{D})\right), \\ a(u_H, v) =& \int_{\mathbb{R}^M}\int_{\mathcal{D}} k(x,\boldsymbol{Z})\,\nabla_x u_H(x,\boldsymbol{Z}) \cdot \nabla_x v(x,\boldsymbol{Z})\,\mathrm{d}x\,\mathrm{d}F\boldsymbol{Z}, \\ b(v) =& \int_{\mathbb{R}^M}\int_{\mathcal{D}} f(x)\,v(x,\boldsymbol{Z})\,\mathrm{d}x\,\mathrm{d}F\boldsymbol{Z} - \int_{\mathbb{R}^M}\int_{\Gamma_N} g(x)\,v(x,\boldsymbol{Z})\,\mathrm{d}x\,\mathrm{d}F\boldsymbol{Z} \\ & - \int_{\mathbb{R}^M}\int_{\mathcal{D}} k(x,\boldsymbol{Z})\,\nabla_x u_0(x) \cdot \nabla_x v(x,\boldsymbol{Z})\,\mathrm{d}x\,\mathrm{d}F\boldsymbol{Z}. \end{aligned}$$

The homogeneous part of the solution u_H lies in $L^2\left(\Omega, H^1_{0,\Gamma_D}(\mathcal{D})\right)$ which is isometrically isomorphic with $H^1_{0,\Gamma_D}(\mathcal{D}) \otimes L^2(\Omega)$. We choose the test space with the same tensor structure, i.e. $V_{h,K} := V_h \otimes V_K$, where the discretization of $H^1_{0,\Gamma_D}(\mathcal{D})$ are finite elements and the discretization of $L^2(\Omega)$ are polynomials

$$V_h = \{\varphi_1(x), \ldots, \varphi_{N_D}(x)\} \subset H^1_{0,\Gamma_D}(\mathcal{D}), \quad V_K = \{\psi_1(\omega), \ldots, \psi_{N_S}(\omega)\} \subset L^2(\Omega).$$

The dimension of $V_{h,K}$ is $N_D N_S$ with the basis

$$\xi_{i,j}(x,\omega) = \varphi_i(x)\,\psi_j(\omega) \quad \forall\, i = 1, \ldots, N_D,\ j = 1, \ldots, N_S.$$

Separable form of input data together with the tensor form of $V_{h,K}$ allow us to assemble the matrix in a compressed form. The resulting system of equations takes the form

$$\begin{aligned} \mathbb{A}\overline{u} = \overline{b}, \quad \mathbb{A} &= \sum_{m=1}^{M} G_m \otimes K_m, \overline{b} = \sum_{m=1}^{M_b} \overline{g}_m \otimes \overline{k}_m, \\ (K_m)_{il} &= \int_{\mathcal{D}} k_m^D(x)\,\nabla\varphi_i(x) \cdot \nabla\varphi_l(x)\,\mathrm{d}x, \\ (G_m)_{jn} &= \int_{\mathbb{R}^M} k_m^S(\boldsymbol{Z})\,\psi_j(\boldsymbol{Z})\,\psi_n(\boldsymbol{Z})\,\mathrm{d}F\boldsymbol{Z}. \end{aligned}$$

We simplify the right hand side as a sum over M_b ($M_b = M + 2$, M terms for Dirichlet boundary and one for forcing term and Neumann boundary) terms with vectors $\overline{g}_m$, $\overline{k}_m$, whose can be assembled in a similar way as G_m, K_m.

The system can be viewed as matrix equations, assuming reshaping $\overline{u}$ into $N_D \times N_S$ matrix $\boldsymbol{u}$

$$\sum_{m=1}^{M} K_m \boldsymbol{u} G_m^T = \sum_{m=1}^{M_b} \overline{k}_m \overline{g}_m^T. \tag{1}$$

3. Solving the stochastic Galerkin matrix equations

The solution of SG matrix equations (1) is quite a difficult task. We will solve it using conjugate gradients with Kronecker preconditioner (see [5]). With the full system, this could be prohibitively expensive ($N_D N_S$ dofs). Therefore, we reduce the test space via the reduced basis method.

3.1. Reduced basis method

The reduced basis (RB) method aims at reducing the number of basis functions while keeping the same approximating properties. In the SG method, it makes sense to create the reduced basis W of V_h as it is the larger part of the basis and we have the tools to create a meaningful subspace of it. The resulting SG test space will take the form of $V_{h,K} \approx W \otimes V_K$, where W is the reduced basis of V_h.

The reduced basis should fulfill all the conditions needed for the discretized system to be well-posed (e.g. discrete inf-sup condition). In the case of our elliptic problem, we can pick any linearly independent reduced basis W and we obtain a valid system

$$\sum_{m=1}^{M} W^T K_m W \boldsymbol{y} G_m^T = \sum_{m=1}^{M_b} W^T \overline{k}_m \overline{g}_m^T, \quad \boldsymbol{u} \approx \tilde{\boldsymbol{u}} = W\boldsymbol{y}.$$

Approximation error of reduced basis W in the context of SG system can be expressed via residual with respect to the original system

$$R = \sum_{m=1}^{M} K_m W \boldsymbol{y} G_m^T - \sum_{m=1}^{M_b} \overline{k}_m \overline{g}_m^T. \tag{2}$$

The most difficult task is to build the reduced basis itself. We do this via Monte Carlo method.

3.2. Construction of the reduced basis via Monte Carlo sampling

The Monte Carlo (MC) approach to the reduced basis construction is based on iterative refinement of the reduced basis. We denote by W_l a reduced basis at iteration l with $W_0 = \emptyset$. The iterative construction can be summarized in the following steps:

1. draw N_{MC} samples $Z_1, \ldots, Z_{N_{MC}}$ of random vector $\boldsymbol{Z}$
2. for every sample Z_j assemble and solve the reduced system of deterministic counterpart
$$W_l^T A_j W_l \widetilde{u}_j = W_l^T b_j$$
3. compute indicators for a sample selection based on the probability density function (pdf) of $\boldsymbol{Z}$ and the residual of reduced solutions $\widetilde{u}_j$
$$f_{\boldsymbol{Z}}(\boldsymbol{Z}_j) \left\| A_j W_l \widetilde{u}_j - b_j \right\|^2$$
4. select P (for simplicity, we use $P = 1$) highest values of identificators and compute solutions at corresponding samples Z_j
$$A_j u_j = b_j$$
5. use the collected solutions to expand the reduced basis W_l and check if the expanded reduced basis is good enough (e.g. with residual (2))

Computation of the reduced solutions and their residuals at samples $\boldsymbol{Z}_j$ is quite costly. We would like to avoid samples around those already contributing to the reduced basis, as they will not bring enough of "new information". We propose

avoiding already generated samples using sampling (changing Step 1) from a changed pdf (using Metropolis-Hastings algorithm)

$$\tilde{f}_l(\boldsymbol{Z}) \propto f(\boldsymbol{Z}) \min_{i=1,\dots,l} w_i(\boldsymbol{Z}), \; w_i(\boldsymbol{Z}) = 1 - \exp\left(-\|\boldsymbol{Z} - X_i\|^2_{\Sigma^{-1}}/2\right).$$

We choose the parameter Σ same as the covariance matrix of $\boldsymbol{Z}$. Illustration of altered pdf and comparison of generated samples can be seen in Figure 2. The benefits of this alternative sampling have diminishing returns when M increases, this can be seen in Figure 3.

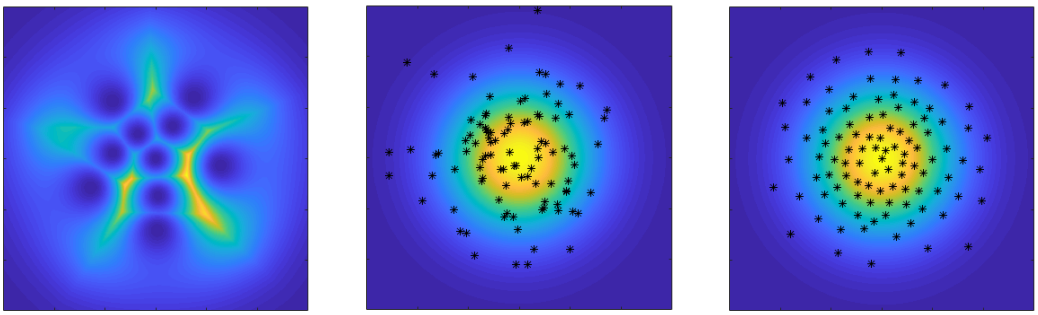


Figure 2: Illustration of altered pdf (left), crude MC samples (middle), samples using altered pdf (right)

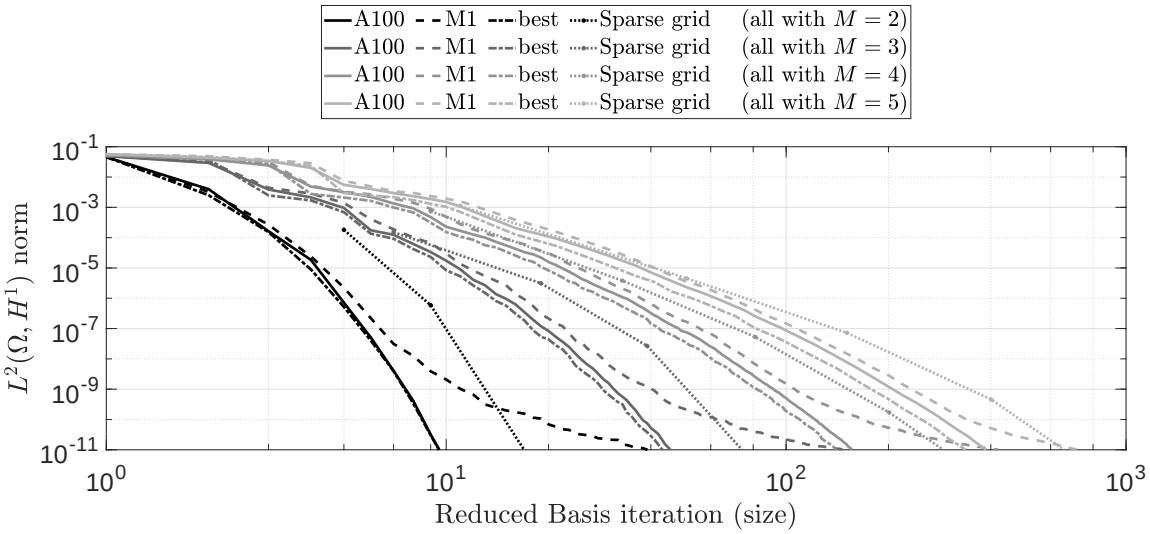


Figure 3: Efficiency of reduced basis construction using different N_{MC}, crude sampling and sampling using altered pdf, and comparison with optimal RB and sparse grid

In Figure 3, we demonstrate the efficiency of the MC approach to the construction of RB on a series of problems with an increasing number of subdomains/number of

random variables and $\mu_m = 0, \sigma_m = 0.3$. We compare two variants: $M1$ - crude MC sampling with $N_{MC} = 1$ and $A100$ - sampling using altered pdf and $N_{MC} = 100$. We add a comparison with the optimal ("best") case of RB constructed from the singular value decomposition of the computed full solution and point selection using Smolyak nested sparse grids (see [3]). We measure the quality of RB in the terms of "true" $L^2(\Omega, H^1(\mathcal{D}))$ error of the resulting SG solution compared to pathwise deterministic solution on the same finite element grid. The "true" error is approximated using 1000 MC samples.

3.3. Deflated conjugate gradients

During the construction of RB, we encounter a solution of many similar systems. We propose the use of deflated conjugate gradients (DCG) [4] with the current iteration of reduced basis W_l as a deflation space to speed up the solution. The main part of the deflation is to project preconditioned residual using the projector $P = I - W_l \left(W_l^T A_j W_l\right)^{-1} W_l^T A_j$. This is fairly cheap as the reduced basis W_l has only a small number of columns.

We show the reduction of the number of iterations when using deflation on a problem with 5 subdomains and $\mu_m = 0, \sigma_m = 0.3$ using target precision of the reduced basis 10^{-6} and precision for the solution of deterministic problems 10^{-9}. We test three very different preconditioners (additive Schwarz, incomplete Cholesky factorization with no filling allowed, and diagonal) to demonstrate that the benefit of the use of DCG is independent of used preconditioner. The comparison of the number of iterations with and without the use of deflation can be seen in Figure 4. The total number of saved iterations is over 80% for all tested preconditioners, i.e. the solution of the series of problems is approximately 5x cheaper.

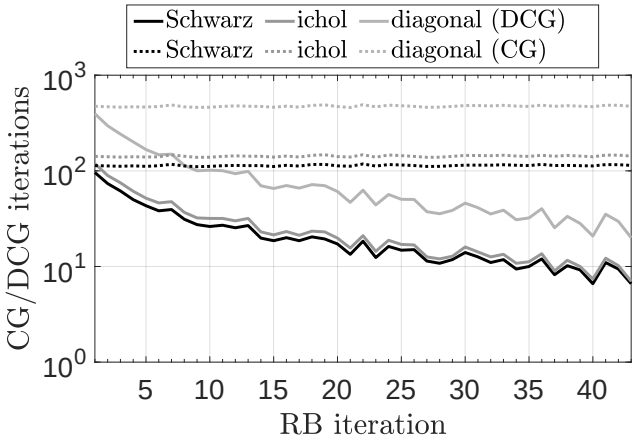


Figure 4: Comparison of number of iterations needed to solve the deterministic problems

4. Use of SG solution - TSX experiment

The main benefit of the SG solution is the result in form of a polynomial surrogate, i.e. an easy and cheap to evaluate approximation of the original problem. We can use this to perform extensive forward uncertainty quantification. We demonstrate this on a simplified tunnel sealing experiment (TSX) [2] modelled as stationary Darcy flow. We will be interested in the stochastic behaviour of pressure in different parts of the domain.

The problem domain is $\mathcal{D} = (0, 100) \times (0, 100) \setminus E$ (E is the ellipse with center $[50, 50]$ and height 2×1.75 and width 2×2.1875). The behaviour of pressure in the tunnel follows

$$\begin{cases} -\mathrm{div}_x \left(\left(\sum_{i=1}^{3} 1_{\mathcal{D}_i}(x)\, 10^{Z_i} \right) \nabla_x u(x, \boldsymbol{Z}) \right) = 0 & \forall\, x \in \mathcal{D}, \boldsymbol{Z} \in \mathbb{R}^3, \\ u(x, \boldsymbol{Z}) = 3 \cdot 10^6 & \forall\, x \in \Gamma_1, \boldsymbol{Z} \in \mathbb{R}^3, \\ u(x, \boldsymbol{Z}) = 0 & \forall\, x \in \Gamma_2, \boldsymbol{Z} \in \mathbb{R}^3, \end{cases}$$

where $Z_1 \sim \mathcal{N}\left(-16, \frac{1}{3}\right)$, $Z_2 \sim \mathcal{N}\left(-18, \frac{1}{3}\right)$, $Z_3 \sim \mathcal{N}\left(-21, \frac{1}{3}\right)$, Γ_1 is the outer boundary of the rectangle, Γ_2 is boundary of cut-out ellipse, and $\mathcal{D}_i$ (1-yellow, 2−teal, 3−blue) are marked in Figure 5.

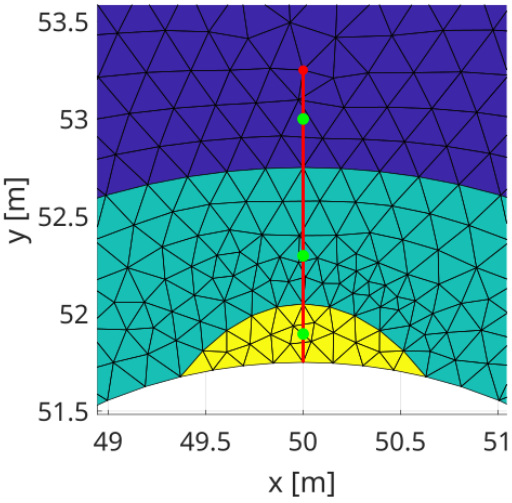


Figure 5: Problem geometry

4.1. Results of forward uncertainty quantification

In Figure 5, we can see a marked red line. We are mainly interested in the behaviour of pressure on this line. Figure 6 shows the comparison of the solution at mean values with the mean value of the stochastic results supplemented by 25%, 50% and 75% quantiles. Note the great difference between the solution at mean values and the mean value of the stochastic solution. The distribution of the pressure at each point on the selected line can be found in Figure 7. Finally, we include

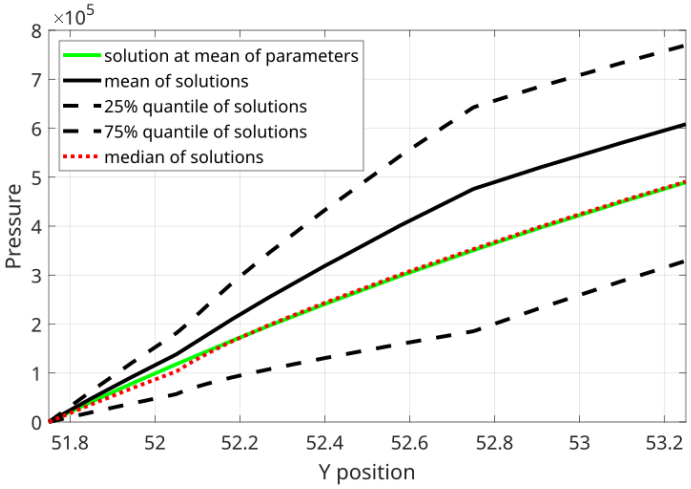


Figure 6: Behaviour on vertical line

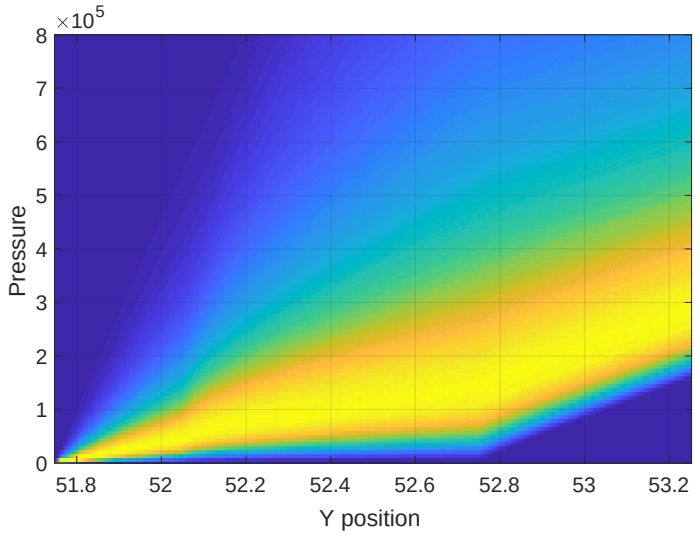


Figure 7: Behaviour on vertical line - distribution at each point

2-dimensional distributions of $\log_{10}$ of pressure for pairs of three selected points (black dots in Figure 8/green dots in Figure 5), see Figure 9. We choose to present $\log_{10}$ of pressures as the two dimensional distributions of pressures were very hard to read.

4.2. Overview of results

The presented results are mainly academic as we used a fairly simplified model. But we can draw some general conclusions. First, the behaviour of the mean value of the stochastic result can be wildly different from the result at the mean values of parameters. The medians are also different, but only slightly in our model. Second, it is very important to choose positions of "real-life" measurements carefully as we can easily pick measurements with overlapping information (as is clearly visible in Figure 8).

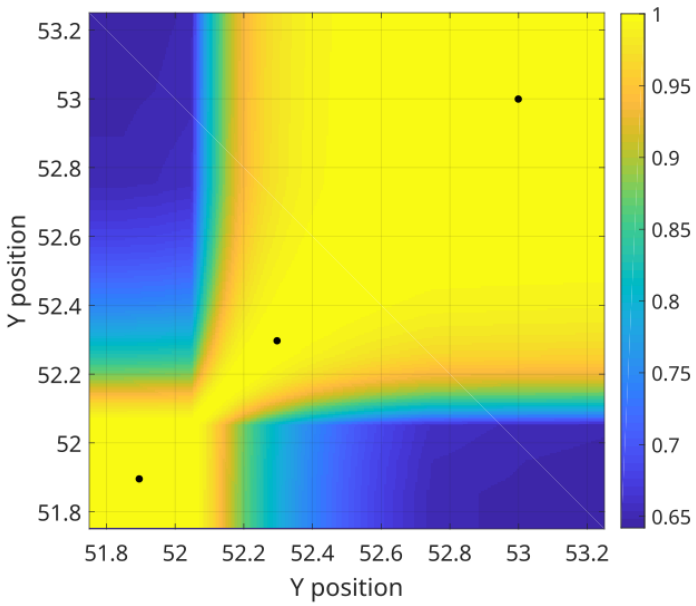


Figure 8: Behaviour on vertical line - correlation between points

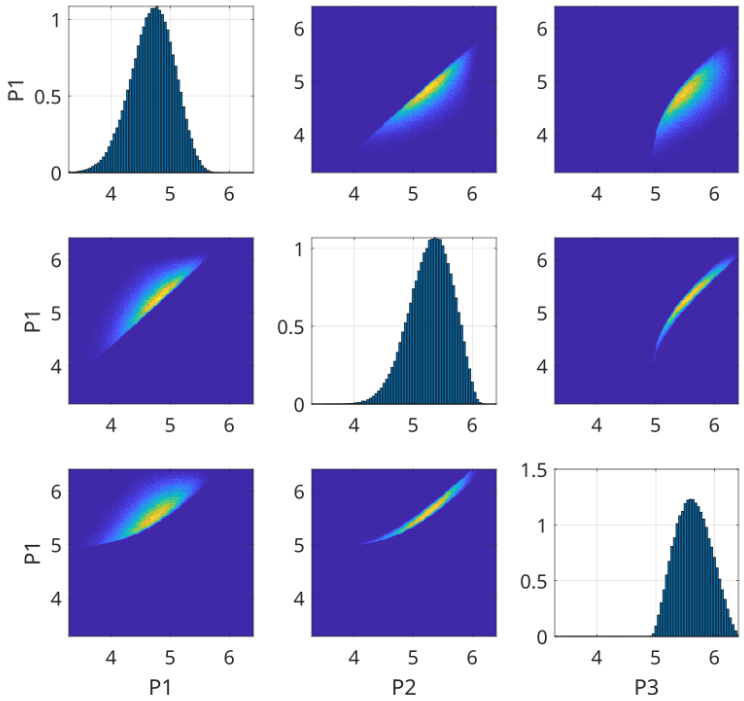


Figure 9: 2-dimensional distributions of $\log_{10}$ of pressure for pairs of selected points

5. Conclusions

The stochastic Galerkin method can be used to create a very precise polynomial surrogate model. Its main drawback is the need for the solution of a very large system of linear equations. In this contribution, we focus on reducing the SG system of equations using the reduced basis method. We present a sampling approach to the construction of the reduced basis, which is demonstrated to be very efficient. Moreover, we demonstrate that the series of similar deterministic systems, we need to solve during the reduced basis construction, can be solved almost five times cheaper using the deflated conjugate gradients. In Section 4, we showed a sample of the SG solution usage for forward uncertainty quantification. This type of analysis can be helpful in e.g. design of experiments.

Acknowledgements

This work was supported by grant No. TK02010118 of the Technology Agency of the Czech Republic.

References

[1] Béreš, M.: A comparison of approaches for the construction of reduced basis for stochastic Galerkin matrix equations. Appl. Math. **65** (2020), 191–225.

[2] Chandler, N., A., Cournut, A., and Dixon, D.: The five year report of the Tunnel Sealing Experiment: an international project of AECL, JNC, ANDRA and WIPP. Tech. rep., 2002.

[3] Petras, K.: Smolyak cubature of given polynomial degree with few nodes for increasing dimension. Numer. Math. **93** (2003), 729–753.

[4] Saad, Y., Yeung, M., Erhel, J., and Guyomarc'h, F.: A deflated version of the conjugate gradient algorithm. SIAM J. Sci. Comput. **21** (2000), 1909–1926.

[5] Ullmann, E.: A Kronecker product preconditioner for stochastic Galerkin finite element discretizations. SIAM J. Sci. Comput. **32** (2010), 923–946.

Programs and Algorithms of Numerical Mathematics 21
J. Chleboun, P. Kůs, J. Papež, M. Rozložník, K. Segeth, J. Šístek (Eds.)
Institute of Mathematics CAS, Prague 2023

NUMERICAL REALIZATION OF THE BAYESIAN INVERSION ACCELERATED USING SURROGATE MODELS

Simona Bérešová[1,2]

[1] Institute of Geonics, Czech Academy of Sciences
Studentská 1768, Ostrava, Czech republic
simona.beresova@ugn.cas.cz
[2] Department of Applied Mathematics, FEECS, VŠB-TUO
17. listopadu 2172, Ostrava, Czech republic

Abstract: The Bayesian inversion is a natural approach to the solution of inverse problems based on uncertain observed data. The result of such an inverse problem is the posterior distribution of unknown parameters. This paper deals with the numerical realization of the Bayesian inversion focusing on problems governed by computationally expensive forward models such as numerical solutions of partial differential equations. Samples from the posterior distribution are generated using the Markov chain Monte Carlo (MCMC) methods accelerated with surrogate models. A surrogate model is understood as an approximation of the forward model which should be computationally much cheaper. The target distribution is not fully replaced by its approximation; therefore, samples from the exact posterior distribution are provided. In addition, non-intrusive surrogate models can be updated during the sampling process resulting in an adaptive MCMC method. The use of the surrogate models significantly reduces the number of evaluations of the forward model needed for a reliable description of the posterior distribution. Described sampling procedures are implemented in the form of a Python package.

Keywords: Bayesian inversion, delayed-acceptance Metropolis-Hastings, Markov chain Monte Carlo, surrogate model

MSC: 65C40, 62F15, 35R30

1. Introduction

This contribution focuses on the acceleration of sampling methods in the Bayesian inversion using surrogate models and describes the resulting Python package created within author's PhD studies. The motivation for the development of the package was the solution of inverse problems from the field of geosciences. The underlying mathematical models are usually based on computationally expensive numerical solutions of boundary value problems and the observed data are corrupted with noise.

DOI: 10.21136/panm.2022.03

This contribution provides an analysis of implemented sampling methods. In order to carry out thorough numerical experiments, a computationally cheap model problem is considered. Applications to geoengineering problems can be found in previous publications [2, 4].

Section 2 outlines the principle of the Bayesian inversion that provides the probability distribution of the unknown parameters (called posterior distribution). Section 3 describes methods used to provide samples from the posterior distribution, focusing on the acceleration using surrogate models. Section 4 describes the Python package and its usage. Section 5 discusses the efficiency of the sampling process, the discussion is supported by numerical experiments.

2. Problem setting

Consider a mathematical model $G\colon \mathbb{R}^n \to \mathbb{R}^m$. The aim is to find a probabilistic description of input parameters to the model corresponding to a given vector of noisy outputs $y \in \mathbb{R}^m$.

Further consider a probability space $(\Omega, \mathcal{F}_\Omega, \mathbb{P})$ and measurable spaces $(\mathbb{R}^n, \mathcal{B}(\mathbb{R}^n))$, $(\mathbb{R}^m, \mathcal{B}(\mathbb{R}^m))$, $n, m \in \mathbb{N}$. We work with three (multivariate) random variables: unknown parameters $U\colon \Omega \to \mathbb{R}^n$, observed data $Y\colon \Omega \to \mathbb{R}^m$, observational noise $Z\colon \Omega \to \mathbb{R}^m$. Their relationship is described by the noise model $Y = G(U) + Z$. Therefore, y is a realization of Y.

The probability distribution of U is called the prior distribution, f_U denotes its probability density function (pdf). It expresses the information about the unknown input parameters known from experience (i.e. without the knowledge of y). Similarly, f_Z denotes the pdf of the noise distribution (i.e. probability distribution of Z). Now, the aim can be retold in the Bayesian way: We would like to obtain the conditional distribution of U given $Y = y$, called posterior distribution. We also say that we refine the prior distribution using the observed data y.

According to the Bayes' theorem, the pdf of the posterior distribution is given by the formula

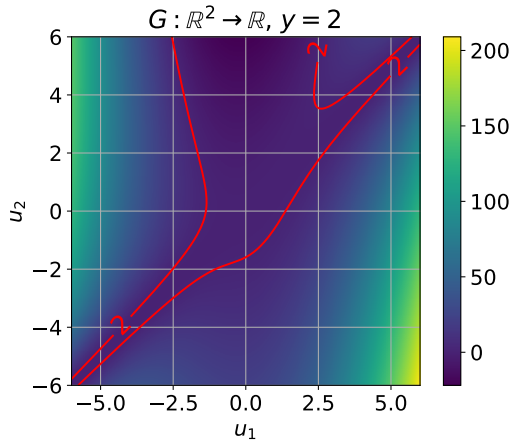
$$f_{U|Y}(u|y) = \frac{f_Z(y - G(u))\, f_U(u)}{\int_{\mathbb{R}^n} f_Z(y - G(v))\, f_U(v)\, \mathrm{d}v} \propto \underbrace{f_Z(y - G(u))}_{\text{likelihood}} \underbrace{f_U(u)}_{\text{prior}},$$

where $\propto$ denotes proportionality, $f_Z(y - G(u))$ (as a function of u) is called the data likelihood.

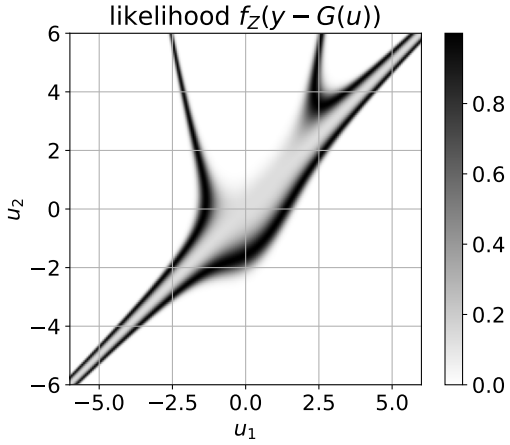
The objective is to generate samples from the posterior distribution, see Section 3. Figure 1 illustrates the principle of the Bayesian inversion on a sample observational operator $G\colon \mathbb{R}^2 \to \mathbb{R}$. This model problem will be also considered in numerical experiments in Section 5.1.

3. Markov chain Monte Carlo (MCMC) methods for posterior sampling

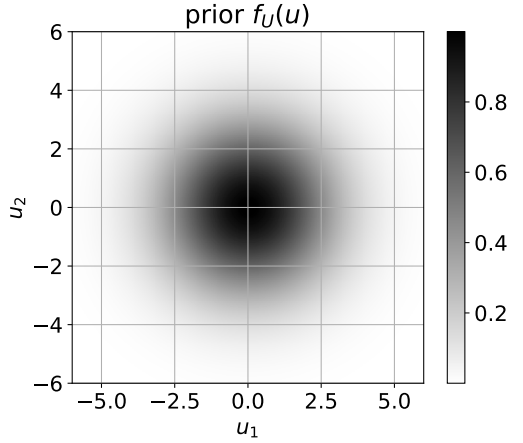
MCMC methods serve to generate samples from a target probability distribution. Here, the target distribution is given by the posterior pdf which is understood as



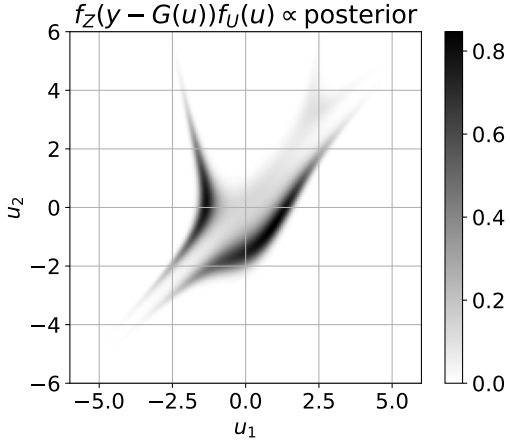
(a) $G(u_1, u_2) = \left(u_1^2 - u_2\right) \ln\left(\left(u_1 - u_2\right)^2 + 1\right)$



(b) Data likelihood



(c) Prior distribution



(d) Posterior distribution

Figure 1: Illustration of Bayesian inversion

a function of u (y is fixed). The target pdf f can be written as

$$f(u) \propto f_Z(y - G(u)) f_U(u); \tag{1}$$

$\propto$ denotes equality up to a multiplicative constant. We consider the Metropolis-Hastings (MH) algorithm and its modification called the delayed-acceptance MH (DAMH) algorithm, see [6, 7]. In recent years, the DAMH algorithm has been widely used and developed, see e.g. [8, 10].

The MH algorithm (Alg. 1) assumes a symmetric proposal distribution such as the Gaussian random walk. In that case,

$$q(\cdot|u) \text{ is the pdf of } \mathcal{N}_n(u; C), \tag{2}$$

$u \in \mathbb{R}^n$ denotes the mean vector and $C \in \mathbb{R}^{n \times n}$ the covariance matrix.

Algorithm 1 Metropolis-Hastings (MH) algorithm

Choose an initial sample $u^{(0)} \in \mathbb{R}^n$ such that $f_Z\left(y - G\left(u^{(0)}\right)\right) f_U\left(u^{(0)}\right) > 0$.
For $k = 0, 1, ...$

1. Propose a sample v from the proposal distribution with pdf $q\left(\cdot|u^{(k)}\right)$.
2. Accept v with probability $a\left(u^{(k)}, v\right) = \min\left\{1, \frac{f_Z(y-G(v))f_U(v)}{f_Z\left(y-G\left(u^{(k)}\right)\right)f_U\left(u^{(k)}\right)}\right\}$ (i.e., set $u^{(k+1)} = v$). Otherwise reject v (i.e., set $u^{(k+1)} = u^{(k)}$).

It can be noticed that the observational operator G is evaluated for each proposed sample v. Therefore, if G is computationally expensive, the MH algorithm is not suitable. Higher sampling efficiency can be achieved by the DAMH algorithm in combination with the use of a surrogate model $\widetilde{G}$ that approximates G. Evaluations of a suitable surrogate model should be much cheaper compared to the evaluations of G. The DAMH algorithm, as introduced in [6], works with the true posterior and also with its approximation. The key property of this algorithm is that it provides samples from the true posterior, the approximation serves only for the acceleration.

Using a surrogate model, the approximation of the posterior pdf can be obtained simply as

$$\widetilde{f}\left(u\right) \propto f_Z\left(y - \widetilde{G}\left(u\right)\right) f_U\left(u\right) \tag{3}$$

The application of the DAMH algorithm to the target distribution in the form of (1) and its approximation (3) leads to Alg. 2.

Algorithm 2 DAMH algorithm using a surrogate model; symmetric proposal pdf

Choose an initial sample $u^{(0)} \in \mathbb{R}^n$ such that $f_Z\left(y - G\left(u^{(0)}\right)\right) f_U\left(u^{(0)}\right) > 0$.
For $k = 0, 1, ...$

1. Propose a sample v from the proposal distribution with pdf $q\left(\cdot|u^{(k)}\right)$.
2. Pre-accept v with probability $\widetilde{a}\left(u^{(k)}, v\right) = \min\left\{1, \frac{f_Z\left(y-\widetilde{G}(v)\right)f_U(v)}{f_Z\left(y-\widetilde{G}\left(u^{(k)}\right)\right)f_U\left(u^{(k)}\right)}\right\}$. Otherwise reject v.
3. If v is pre-accepted, accept it with probability $a\left(u^{(k)}, v\right) = \min\left\{1, \frac{f_Z\left(y-\widetilde{G}\left(u^{(k)}\right)\right)f_Z(y-G(v))}{f_Z\left(y-\widetilde{G}(v)\right)f_Z\left(y-G\left(u^{(k)}\right)\right)}\right\}$. Otherwise reject v.

Further improvement can be achieved by increasing the quality of the surrogate model. During the DAMH algorithm, new snapshots $\left(u^{(k)}, G\left(u^{(k)}\right)\right)$ are obtained and it is beneficial to use them for the update of the surrogate model $\widetilde{G}$. The resulting

DAMH algorithm with surrogate model updates (DAMH-SMU) can be written as Alg. 2 with an added step:

4. Optionally, use $\left(u^{(k)}, G\left(u^{(k)}\right)\right)$ to update the surrogate model $\widetilde{G}$.

After $\widetilde{G}$ is modified, it is necessary to recalculate $\widetilde{G}\left(u^{(k)}\right)$; however, the computation cost is assumed to be negligible.

This natural modification of Alg. 2 utilizes all evaluations of G known so far. To implement the surrogate model updates, non-intrusive surrogate models should be used. As shown in numerical experiments in Section 5.1, suitable surrogate models can be constructed for example using a projection to a polynomial basis or using radial basis functions.

MCMC methods are sequential in principle. Therefore, typical utilization of computational resources consists in running several independent sampling processes in parallel. The DAMH-SMU algorithm allows additional acceleration - the parallel processes can share one surrogate model constructed using snapshots obtained by all of the sampling processes. Benefits of this approach are supported by numerical experiments, see Section 5.1.

4. Implementation

The sampling methods described in Section 3 are included in the author's Python library for the numerical realization of the Bayesian inversion (available at [5]). The implementation utilizes MPI processes via the mpi4py library, [9]. For the scheme of the parallel processes see Fig. 2. Several sampling processes (SAMPLER 1 to N) are running in parallel. These processes share one surrogate model which is refined using data from all of these processes. The COLLECTOR process serves for the construction of the surrogate model and for its distribution to the SAMPLERs. The SAMPLERs also share a pool of SOLVERs, i.e., processes that evaluate the forward model G. SOLVERs are typically implemented via a linked numerical library. The SOLVERS POOL assigns the computations required by SAMPLERs to individual SOLVERs. The evaluations of G typically form the majority of the computational time; therefore, for a good utilization of computational resources, the number of SOLVERs is typically lower than the number of SAMPLERs.

The typical sampling process can be divided into several phases:

1. The first phase is the basic MH algorithm, obtained snapshots are used for the construction of a shared surrogate model.
2. The main phase is the DAMH-SMU algorithm during which the surrogate model is updated and used for the acceleration of the sampling process.
3. When the surrogate model is accurate enough, its updates can stop, i.e., the DAMH algorithm is used.
4. Post-processing of obtained samples. This includes the computation of moments, visualization, autocorrelation analysis, etc.

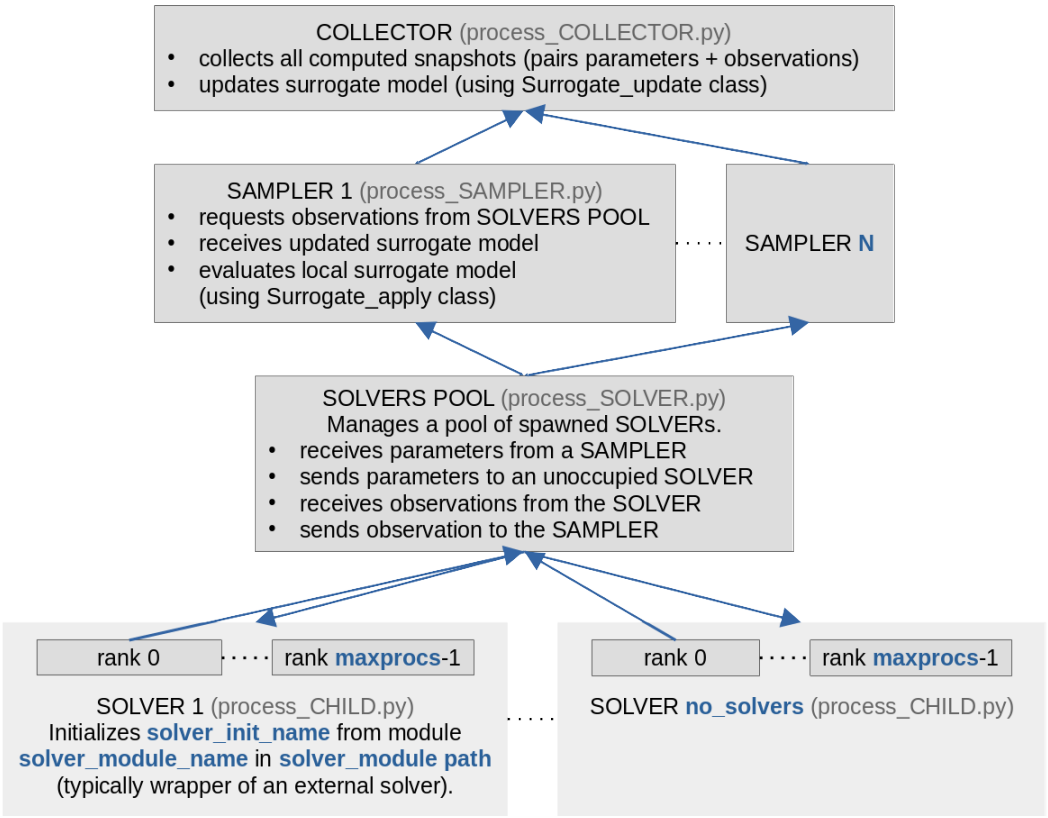


Figure 2: MPI processes

The sampling processes use the Gaussian random walk proposal distribution $\mathcal{N}_n(u, C)$. For each sampling phase, it is possible to specify the covariance matrix C, surrogate model type, and a stopping criterion. The stopping criterion can be a pre-specified length of the produced chain, number of G evaluations, or reaching the maximum sampling time. For the surrogate model construction, two methods are implemented: projection to a basis of Hermite polynomials and interpolation using radial basis functions (RBF); for details see [4]. In the case of the polynomial surrogate model, the polynomial degree is chosen based on the number of currently available snapshots up to a pre-specified maximal degree. In the case of the RBF model, it is possible to specify the RBF type (e.g. polyharmonic, Gaussian, etc.) and to set a limit on the number of snapshots used for the construction of the surrogate model.

5. Sampling efficiency

MCMC methods are usually understood as methods for the construction of an ergodic Markov chain invariant with respect to (wrt) a target probability distribution. MH and DAMH algorithms are valid (invariant and ergodic) under mild assumptions on supports of q, f, $\widetilde{f}$.

In the case of the basic MH algorithm, it can be shown that obtained samples form a realization of a Markov chain invariant wrt the posterior distribution. The use of the Gaussian random walk proposal (2) also implies the ergodicity of the Markov chain. For details see e.g. [11].

The DAMH algorithm can be understood as a specific version of the MH algorithm with a modified proposal distribution that includes the pre-acceptance step based on the surrogate model. For the explanation see [6]. Therefore, the resulting Markov chain is also invariant wrt the posterior distribution. Under the additional condition

$$\operatorname{supp} \widetilde{f} \supset \operatorname{supp} f,$$

the Markov chain ergodic, for a detailed explanation see [4].

The DAMH-SMU algorithm was obtained by a small modification of the DAMH algorithm; however, there is a significant difference in ensuring ergodicity. Since $\widetilde{G}$ changes, the proposal distribution also changes and the algorithm becomes an adaptive MCMC method. Several possibilities of ensuring the ergodicity of this type of adaptive MCMC methods are offered by [1, 12]. In situations when it is not possible to prove the ergodicity of the DAMH-SMU algorithm, the validity of the sampling process can be ensured simply by stopping the adaptations at some point. The whole DAMH-SMU algorithm is then understood as a means to find suitable parameters of proposal distribution for the next phase (DAMH algorithm).

Besides the theoretical properties, we should also deal with practical aspects that affect the sampling process efficiency. Specifically, the choice of q has a major impact on sampling efficiency. Too low variance of $q(\cdot|u)$ causes high autocorrelation of the resulting Markov chain; therefore, too many samples are required to explore the parameter space. Conversely, if $q(\cdot|u)$ has a high variance, large amount of proposed samples are likely to be rejected, which also results in a high autocorrelation.

There are various approaches attempting to find an "optimal" proposal distribution. Theoretical and experimental results are based on studying the impact of the average acceptance rate

$$\alpha = \mathbb{E}\left[a(u,v)\right] = \int_{u\in\mathcal{U}} \int_{v\in\mathcal{U}} a(u,v)\ \mathrm{d}Q(u,v)\ \mathrm{d}\mu(u)$$

on the integrated autocorrelation time (IAT) $\tau = 1 + 2\sum_{i=1}^{\infty} \rho_k$ (ρ_k is the autocorrelation at lag k). In practice, these values are estimated as a part the the post-processing phase. The average acceptance rate requires monitoring the ratio of accepted/rejected samples. A reliable empirical estimation of IAT is difficult in practice since it requires Markov chains many times longer than the value of IAT.

When an estimation $\widehat{\tau}$ of IAT is available, the sampling efficiency can be assessed using the efficiency criterion proposed in [3] – cost per almost uncorrelated sample (CpUS). In the case of the DAMH algorithm,

$$\mathrm{CpUS} = \left(\frac{N_{\mathrm{acc}} + N_{\mathrm{rej}}}{N} + \mathrm{cost}_{\widetilde{G}}\right)\tau, \tag{4}$$

N is the length of the generated Markov chain, N_{acc} (N_{rej}) is the number of accepted (rejected) samples, and $\text{cost}_{\widetilde{G}}$ is the ratio of the average cost of $\widetilde{G}$ evaluations to the average cost of G evaluations. In the case of the basic MH algorithm, CpUS is given by IAT (by definition), i.e. CpUS $= \tau$. This is in accordance with formula (4), since $N = N_{\text{acc}} + N_{\text{rej}}$ and $\text{cost}_{\widetilde{G}} = 0$ ($\widetilde{G}$ is not used).

5.1. Numerical experiments

The first set of numerical experiments examines the efficiency of MH and DAMH algorithms depending on the variance of the proposal distribution. These experiments can also be used to compare these algorithms in terms of the optimal CpUS.

The proposal pdf $q(\cdot|u)$ is the pdf of

$$\mathcal{N}_n(u; C) = \mathcal{N}_2\left(u; \sigma^2 I\right)$$

where $\sigma \in \{0.4, 0.6, \ldots, 5.0\}$. For each σ, 20 Markov chains of sufficient length (in order to obtain a good estimation of IAT) were generated in parallel using MH and DAMH algorithms. The following methods were considered:

"MH" Basic MH algorithm, CpUS $\approx \widehat{\tau}$.

"exact" DAMH algorithm with a hypothetical ideal surrogate model. This does not require any simulation, data obtained using the MH algorithm are recycled. The surrogate model is assumed to be exact; therefore, there are no rejections and CpUS estimation is obtained as

$$\text{CpUS} \approx \left(\frac{N_{\text{acc}}}{N} + \text{cost}_{\widetilde{G}}\right)\widehat{\tau}.$$

"constant" DAMH algorithm with a non-informative surrogate model, $\widetilde{G}(u) = 0$ for each u. CpUS is calculated using (4), τ is replaced by its estimation $\widehat{\tau}$.

Table 1 shows obtained data required for the calculation of CpUS for chosen values of σ. In this model problem, the evaluations of G are very cheap; therefore, for illustrative purposes, the value of $\text{cost}_{\widetilde{G}}$ is chosen artificially. Figure 3a shows the dependence of CpUS on σ for $\text{cost}_{\widetilde{G}} = 0.3$. For each algorithm, the optimal CpUS is marked. However, in typical applications, the value of $\text{cost}_{\widetilde{G}}$ is usually much lower. Figure 3b shows the corresponding results recalculated for $\text{cost}_{\widetilde{G}} = 0.001$. As indicated by these results, the DAMH algorithm requires higher values of σ than the MH algorithm for higher efficiency. Furthermore, the optimal value of σ increases with decreasing value of $\text{cost}_{\widetilde{G}}$.

In addition, Figure 3b contains results obtained for more realistic surrogate models:

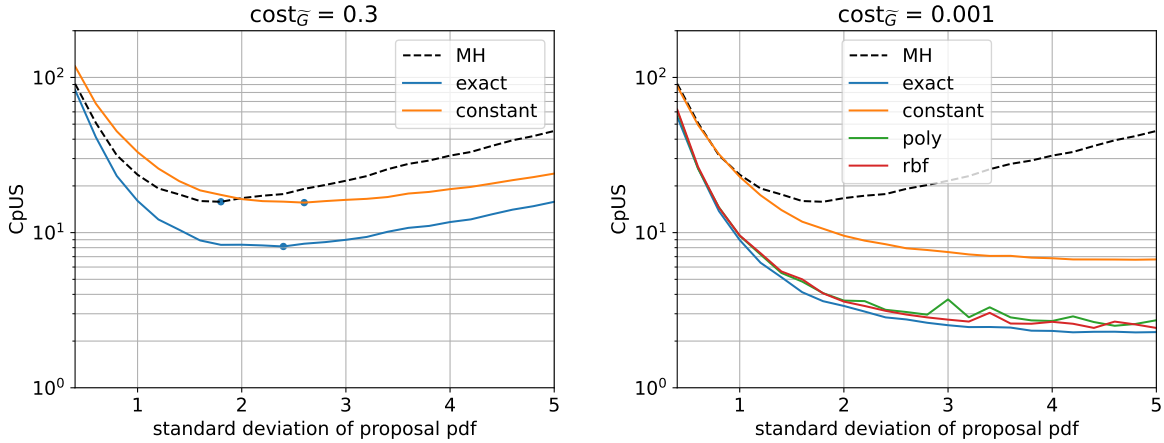
"poly" DAMH algorithm with a polynomial surrogate model.

"rbf" DAMH algorithm with a RBF surrogate model.

	MH, DAMH "exact"					
σ	0.4	1.0	2.0	3.0	4.0	5.0
$\frac{N_{\text{acc}}}{N}$	0.61	0.38	0.20	0.12	0.07	0.05
$\frac{N_{\text{rej}}}{N}$	$1 - \frac{N_{\text{acc}}}{N}$					
$\widehat{\tau}$	91.0	23.6	16.7	21.6	31.3	45.2

DAMH "constant"					
0.4	1.0	2.0	3.0	4.0	5.0
0.56	0.32	0.16	0.09	0.06	0.04
0.31	0.36	0.25	0.16	0.11	0.08
100.8	33.9	23.1	29.3	40.9	57.8

Table 1: Data required for the calculation of CpUS



(a) Non-negligible surrogate model cost

(b) Computationally cheap surrogate model

Figure 3: Comparison of MH and DAMH algorithms in terms of CpUS

Naturally, the values of CpUS cannot be lower than with the "exact" surrogate model. The lowest achieved CpUS is approximately 16 for MH, 2.5 for "poly" and "rbf", and 2.3 for "exact".

In the previous experiment, standard DAMH algorithm without surrogate model updates was considered. The second numerical experiment is designed to show the benefits of surrogate model updates during the sampling process. This will be shown through monitoring the amount of rejected samples in the DAMH algorithm. These samples require the evaluation of the (usually computationally expensive) observational operator G but they are rejected afterwards; therefore, the number of rejected samples should be as low as possible. In the hypothetical ideal case of the exact surrogate model, there would be no rejected samples.

For this purpose, the RFB surrogate model is used and the sampling process is divided into several phases:

$$\text{MH (100)} \rightarrow \text{DAMH (long)} \overbrace{\rightarrow \text{DAMH-SMU (100)}}^{\text{5 times}} \rightarrow \text{DAMH (long)}$$

The initial phase is the basic MH algorithm with stopping criterion set to 100 evaluations of G. Then, there are five DAMH-SMU phases, all of them terminated after

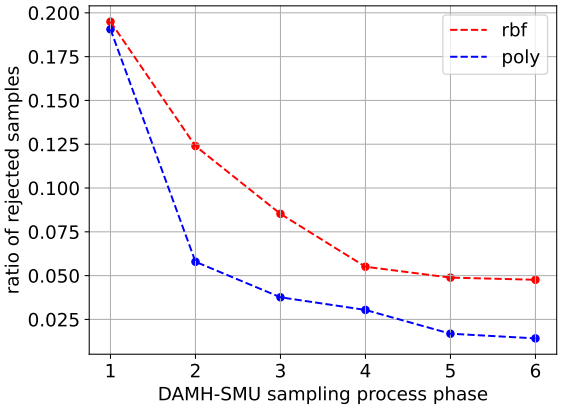


Figure 4: Benefits of surrogate model updates

100 evaluations of G. After each of the phases, there is a long DAMH phase. These additional DAMH phases serve only for monitoring the surrogate model quality, they do not affect the DAMH-SMU phases in any way. Figure 4 shows the ratio of rejected samples (to the number of all samples) for each DAMH phase corresponding to surrogate models constructed from increasing numbers of snapshots (from 100 to 600). The figure shows the intended behavior – with increasing quality of the surrogate model, the ratio of rejected samples decreases.

6. Conclusions

The contribution focused on MCMC methods providing samples from the exact posterior distribution. Such methods require many evaluations of the observational operator. It was shown that the use of surrogate models and the DAMH algorithm can spare a significant number of G evaluations compared to the basic MH algorithm. Also, it was shown that the surrogate model can be updated during the sampling process, leading to a further increase in the efficiency of the sampling process.

An advantage of the presented Python framework is that the implemented methods have general use, the forward model G can be a "black box". The only requirements for the use of this Python package are the specification of the prior distribution and the observational noise, and the availability of a solver that evaluates the observational operator G.

Acknowledgement

The work has been supported by the Programme for funding of applied research, experimental development, and innovation THETA of Technology Agency of the Czech Republic under contract ID TK02010118.

References

[1] Bai, Y., Roberts, G.O., and Rosenthal, J.S.: On the containment condition for adaptive Markov chain Monte Carlo algorithms. Advances and Applications in Statistics **21** (2011), 1–54.

[2] Blaheta, R., Béreš, M., Domesová, S., and Horák, D.: Bayesian inversion for steady flow in fractured porous media with contact on fractures and hydro-mechanical coupling. Computational Geosciences (2020).

[3] Blaheta, R., Béreš, M., Domesová, S., and Pan, P.: A comparison of deterministic and Bayesian inverse with application in micromechanics. Applications of Mathematics **63** (2018), 665–686.

[4] Bérešová, S.: *Bayesian approach to the identification of parameters of differential equations*. PhD thesis, VSB - Technical University of Ostrava, Ostrava, 2022. URL http://hdl.handle.net/10084/148521.

[5] Bérešová, S.: surrDAMH, 2022. URL https://github.com/dom0015/surrDAMH/tree/Version1.

[6] Christen, J.A. and Fox, C.: Markov chain Monte Carlo Using an Approximation. Journal of Computational and Graphical Statistics **14** (2005), 795–810.

[7] Cui, T., Fox, C., and O'Sullivan, M.J.: Bayesian calibration of a large-scale geothermal reservoir model by a new adaptive delayed acceptance Metropolis Hastings algorithm. Water Resources Research **47** (2011).

[8] Cui, T., Fox, C., and O'Sullivan, M.J.: A posteriori stochastic correction of reduced models in delayed-acceptance MCMC, with application to multiphase subsurface inverse problems. International Journal for Numerical Methods in Engineering **118** (2019), 578–605.

[9] Dalcin, L.: mpi4py, 2022. URL https://mpi4py.readthedocs.io.

[10] Lykkegaard, M.B., Mingas, G., Scheichl, R., Fox, C., and Dodwell, T.J.: Multilevel Delayed Acceptance MCMC with an Adaptive Error Model in PyMC3. Tech. Rep. arXiv:2012.05668, 2020.

[11] Robert, C.P. and Casella, G.: *Monte Carlo statistical methods*. Springer texts in statistics, Springer, New York, NY, 2004. OCLC: 837651914.

[12] Roberts, G.O. and Rosenthal, J.S.: Coupling and Ergodicity of Adaptive Markov Chain Monte Carlo Algorithms. Journal of Applied Probability **44** (2007), 458–475.

Programs and Algorithms of Numerical Mathematics 21
J. Chleboun, P. Kůs, J. Papež, M. Rozložník, K. Segeth, J. Šístek (Eds.)
Institute of Mathematics CAS, Prague 2023

THE IMPACT OF UNCERTAIN PARAMETERS ON RATCHETTING TRENDS IN HYPOPLASTICITY

Jan Chleboun, Judita Runcziková, Pavel Krejčí

Faculty of Civil Engineering, Czech Technical University in Prague
Thákurova 7, 166 29 Prague 6, Czech Republic
jan.chleboun@cvut.cz, judita.runczikova@fsv.cvut.cz, pavel.krejci@cvut.cz

Abstract: Perturbed parameters are considered in a hypoplastic model of granular materials. For fixed parameters, the model response to a periodic stress loading and unloading converges to a limit state of strain. The focus of this contribution is the assessment of the change in the limit strain caused by varying model parameters.

Keywords: granular material, hypoplasticity, ratchetting, uncertain parameters

MSC: 65Z05, 74C15, 74H40, 90C70

1. Introduction

In this contribution, the hypoplastic model [6] of granular materials is considered together with uncertain input parameters. The focus is concentrated on the influence the uncertainty in inputs has on the limit state of ratchetting. The limit state is, of course, determined only approximately through a numerical modeling of a finite number of cyclic loading and unloading steps.

Let us imagine a cohesionless granular material such as soil, sand, or gravel. Its behavior is different from common solid materials and cannot be modeled by common models widely used in elasticity and plasticity. Various models of hypoplastic granular materials have been proposed, see, for instance, a micromechanical approach [1, 3], a macromechanical approach [4], or a survey in [2] or [6]. As already indicated, we will use the model presented in the paper [6] that is a continuation of [2], and both follow the hypoplastic concept proposed in [5].

Unlike in common elasticity models where a body is loaded by forces that, through a strain, produce a stress, loading by stress is considered and strain inferred in hypoplastic models. An effect called ratchetting is then observed. It can be briefly characterized as a behavior in which deformation accumulates due to cyclic mechanical stress. As a consequence, the hypoplastic material is made denser and more compacted than the material before cycles of loading and unloading. The material

DOI: 10.21136/panm.2022.04

in an initial state is looser and has a greater void ratio (imagine a pile of fluffy soil under a cyclic loading). The material response to the cyclic loading is more and more stable. Finally, a limit state both in the strain and the void ratio is reached. In the limit, the void ratio takes its minimum that is given by the parameter e_d in our model, see (9) in Section 2. Our contribution is concerned with an arising question: How strongly is the limit strain influenced by the uncertainty in parameters that enter the model?

2. The hypoplastic model

The model whose response will be investigated is introduced in this section. We follow [6] but make the presentation significantly condensed. The reader can also get an idea of the model in [2] where, however, the exposition is not as straightforward as in [6].

The model [6] in a general form is given by

$$\dot{\boldsymbol{\sigma}}(t) = c_1(t)\big(L(\boldsymbol{\sigma}(t)) : \dot{\boldsymbol{\varepsilon}} + f(t)N(\boldsymbol{\sigma}(t))\|\dot{\boldsymbol{\varepsilon}}(t)\|\big), \tag{1}$$

where $\boldsymbol{\sigma}$ is the stress tensor, $\boldsymbol{\varepsilon}$ is the strain tensor, $c_1(t)$ and $f(t)$ are "time" dependent quantities, see (6) and (9), and $L(\boldsymbol{\sigma}(t))$ and $N(\boldsymbol{\sigma}(t))$ are tensors, see the next paragraph. The canonical scalar product is denoted by : and the dot stands for the derivative with respect to t, a time-like parameter on which the evolution of the loading process depends. Since a stress-controlled loading is considered, the stress $\boldsymbol{\sigma}(t)$ is given and the strain $\boldsymbol{\varepsilon}(t)$ is to be determined.

Let us particularize (1) in terms of matrices and scalar functions. The loading is specified as proportional to a given 3×3 symmetric matrix $\mathbf{S} = (s_{ij})_{i,j=1,2,3}$, that is, $\boldsymbol{\sigma}(t) = \sigma(t)\mathbf{S}$, where $\sigma : [b, b+T] \to (0, \infty)$ is a given monotone scalar function defined on an interval $[b, b+T]$ of the length T. The proportional loading or unloading is determined by the increasing or decreasing function σ, respectively.

Let us consider matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{3\times3}$, and introduce

$$\langle \mathbf{A}, \mathbf{B} \rangle = \mathbf{A} : \mathbf{B} = \mathrm{tr}(\mathbf{B}^{\mathrm{T}}\mathbf{A}) = \sum_{1\le j,k\le 3} a_{jk}b_{jk},$$

the Frobenius norm

$$\|\mathbf{A}\| = \langle \mathbf{A}, \mathbf{A} \rangle^{1/2}$$

as well as the identity matrix $\mathbf{I} \in \mathbb{R}^{3\times3}$.

Then the detailed form of the model (1) is as follows:

$$\dot{\sigma}(t)\mathbf{S} = c_1(t)\sigma(t)\left(a^2\langle \mathbf{S}, \mathbf{I} \rangle \dot{\boldsymbol{\varepsilon}}(t) + \frac{1}{\langle \mathbf{S}, \mathbf{I} \rangle}\langle \mathbf{S}, \dot{\boldsymbol{\varepsilon}}(t) \rangle \mathbf{S} + af(t)\|\dot{\boldsymbol{\varepsilon}}(t)\|\Big(2\mathbf{S} - \frac{1}{3}\langle \mathbf{S}, \mathbf{I} \rangle \mathbf{I}\Big)\right), \tag{2}$$

where $a > 0$ is a real parameter and f is a positive t-dependent function. Both quantities will be considered uncertain.

Let us assume $\dot{\sigma}(t) \neq 0$ and define a matrix-valued function $\mathbf{X}$ as well as a constant matrix $\mathbf{A}$

$$\mathbf{X}(t) = c_1(t)\frac{\sigma(t)}{\dot{\sigma}(t)}\dot{\boldsymbol{\varepsilon}}(t), \quad \mathbf{A} = \frac{\mathbf{Q}}{a^2 + \|\mathbf{Q}\|^2}, \tag{3}$$

where

$$\mathbf{Q} = \frac{\mathbf{S}}{\langle \mathbf{S}, \mathbf{I} \rangle}.$$

After some manipulation, see [6], the model (2) takes the following form

$$\mathbf{X}(t) = \mathbf{A} + af(t)\|\mathbf{X}\|\mathbf{B} \quad \text{for } \dot{\sigma}(t) > 0, \tag{4}$$
$$\mathbf{X}(t) = \mathbf{A} - af(t)\|\mathbf{X}\|\mathbf{B} \quad \text{for } \dot{\sigma}(t) < 0, \tag{5}$$

where

$$\mathbf{B} = \left(2 + \frac{1}{3a^2}\right)\mathbf{A} - \frac{1}{3a^2}\mathbf{I}, \quad f(t) = F(e(t)) = f_0\left(\frac{e(t) - e_d}{e_c - e_d}\right)^{\alpha}, \tag{6}$$

$e(t)$ is the void ratio and α, $e_c > e_d$ are positive constants that will be considered uncertain. Instead of f_0, a fixed value of 1 is used in [6]. Since we wish to perturb this value too, we introduce a parameter f_0 and consider it uncertain.

The mass balance equation implies a differential equation for the void ratio $e(t)$, namely,

$$\dot{e}(t) = (1 + e(t))\langle \dot{\boldsymbol{\varepsilon}}(t), \mathbf{I} \rangle. \tag{7}$$

Recall $\mathbf{X}(t) = c_1(t)\frac{\sigma(t)}{\dot{\sigma}(t)}\dot{\boldsymbol{\varepsilon}}(t)$, then, with the help of (7)

$$\langle \mathbf{X}(t), \mathbf{I} \rangle = c_1(t)\frac{\sigma(t)}{\dot{\sigma}(t)}\langle \dot{\boldsymbol{\varepsilon}}(t), \mathbf{I} \rangle = c_1(t)\frac{\sigma(t)}{\dot{\sigma}(t)}\frac{\dot{e}(t)}{1 + e(t)}. \tag{8}$$

Now, it is the right time to reveal that the function c_1 is also e dependent. Indeed,

$$c_1(t) = -\bar{c}(e(t) - e_d)^{-\beta}, \tag{9}$$

where the constants $\bar{c}$, β, and e_d will be considered uncertain but constrained by $0 < \bar{c}$, $1 \leq \beta$, and $e_d \in (0, 1)$. The meaning of (9) is that the material becomes rigid when $e(t)$ asymptotically converges (decreases) to e_d, see [6]. The parameter e_d represents the minimum void ratio of the granular material. It is one of the model input parameters and its physically realistic value can be obtained from measurements.

Remark 1: Unlike [6], where $1 < \beta$ is considered, we allow for $\beta = 1$ to represent uncertainty in β by a closed interval, see Section 3. The value $\beta = 1$ does not cause any singularity or discontinuity in the model behavior.

Let us choose a special sort of loading and unloading. In particular, let σ be continuous, piecewise exponential, and periodic with the period equal to $2T$. Moreover, let $\dfrac{\sigma(t)}{\dot\sigma(t)}$ be alternately equal to 1 if $\dot\sigma(t) > 0$, and -1 if $\dot\sigma(t) < 0$. This choice simplifies (8) and (3) to

$$\langle \mathbf{X}(t), \mathbf{I}\rangle = \eta c_1(t) \frac{\dot e(t)}{1+e(t)} \quad \text{and} \quad \mathbf{X}(t) = \eta c_1(t) \dot{\boldsymbol\varepsilon}(t), \tag{10}$$

where $\eta = 1$ if $\dot\sigma(t) > 0$, and $\eta = -1$ if $\dot\sigma(t) < 0$.

The left equality in (10) and the equalities (4)–(5) lead, after a clever manipulation, see [6, Section 4], to

$$\dot e(t) \frac{\widehat{c}_1(e(t))}{1+e(t)} \big(h(F(e(t))) - g(F(e(t)))\big) = 1 \text{ for } \dot\sigma > 0, \tag{11}$$

$$\dot e(t) \frac{\widehat{c}_1(e(t))}{1+e(t)} \big(h(F(e(t))) - g(F(e(t)))\big) = -1 \text{ for } \dot\sigma < 0, \tag{12}$$

where $\widehat{c}_1(e(t)) = c_1(t)$ and

$$h(f) = \frac{\langle \mathbf{A}, \mathbf{I}\rangle + \left(\langle \mathbf{B}, \mathbf{I}\rangle \langle \mathbf{A}, \mathbf{B}\rangle - \langle \mathbf{A}, \mathbf{I}\rangle \|\mathbf{B}\|^2\right) a^2 f^2}{\langle \mathbf{A}, \mathbf{I}\rangle^2 - \|\langle \mathbf{A}, \mathbf{I}\rangle \mathbf{B} - \langle \mathbf{B}, \mathbf{I}\rangle \mathbf{A}\|^2 a^2 f^2},$$

$$g(f) = \frac{\langle \mathbf{B}, \mathbf{I}\rangle \sqrt{\|\mathbf{A}\|^2 - a^2 f^2 \left(\|\mathbf{A}\|^2 \|\mathbf{B}\|^2 - \langle \mathbf{A}, \mathbf{B}\rangle^2\right)}\, af}{\langle \mathbf{A}, \mathbf{I}\rangle^2 - \|\langle \mathbf{A}, \mathbf{I}\rangle \mathbf{B} - \langle \mathbf{B}, \mathbf{I}\rangle \mathbf{A}\|^2 a^2 f^2}.$$

In (11)–(12), the functions $\widehat{c}_1$ and F are used instead of c_1 and f to emphasize the dependence on the function e, which is the unknown function in the ordinary differential equation (11)–(12).

The matrix function $\mathbf{X}(t)$ can be expressed through $\mathbf{A}$, $\mathbf{B}$, a, and $F(e(t))$, see [6, Section 4]. As a consequence, if $e(t)$ is known by solving (11)–(12), then $\mathbf{X}(t)$ is known too, and $\boldsymbol\varepsilon(t)$ can be determined from the right equality in (10).

Remark 2: If $\mathbf{S} = -\mathbf{I}$, then (4)–(5) represent the case of isotropic loading. A detailed analysis of this special case together with a convergence analysis is presented in [6, Sections 2 and 3].

Remark 3: Let the function σ that reduces (8) to (10) be defined as in [6, Section 2]. In detail, let σ_1 and σ_2 be two positive real numbers such that $\sigma_1 < \sigma_2$, let $T = \ln\sigma_2 - \ln\sigma_1$, and let $t_k = kT$ for $k = 0, 1, 2, \dots$. We define

$$\sigma(t) = \sigma_1 \mathrm{e}^{t - t_{2j}} \text{ for } t \in (t_{2j}, t_{2j+1}), \quad \sigma(t) = \sigma_2 \mathrm{e}^{t_{2j+1} - t} \text{ for } t \in (t_{2j+1}, t_{2j+2}). \tag{13}$$

3. Uncertain inputs

Before we elaborate on uncertain parameters, let us show the model response to crisp inputs. As in [6, Section 5], we fix $a = 0.4$, $e_c = 0.8$, $e_d = 0.4$, $f_0 = 1$, $\alpha = 0.1$,

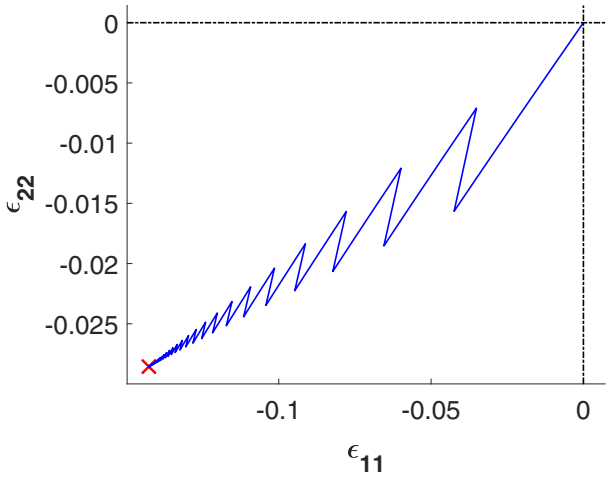


Figure 1: Ratchetting. The cross marks the numerical limit after 50 loading-unloading cycles. The strain evolution is fully characterized in the $\varepsilon_{11}\varepsilon_{22}$-plane because $\varepsilon_{22} = \varepsilon_{33}$.

$\bar{c} = 2$, $\beta = 1.03$, $\sigma_1 = 10$, and $\sigma_2 = 12$. The initial conditions are $e_0 = e(0) = 0.7$ and $\boldsymbol{\varepsilon}(0) = 0 \cdot \boldsymbol{I}$. Let the matrix $\mathbf{S}$ be diagonal with $s_{11} = -1.5$ and $s_{22} = s_{33} = -1$.

Let us focus on strain as the model monitored response. We have $\varepsilon_{22} = \varepsilon_{33}$ by virtue of the chosen values of $\mathbf{S}$. This feature allows for graphing the full strain evolution in the $\varepsilon_{11}\varepsilon_{22}$-plane in the course of 50 loading-unloading cycles, see Fig. 1. Although the loading and unloading are periodic, the strain cycles are not. As they converge to an equilibrium, we observe a shift in their cycles accompanied by a decreasing amplitude. This phenomenon is called ratchet(t)ing, see, for example, [1, 3].

Let us define the quantity of interest (QoI) as the numerical limit (after 100 cycles) of the strain in the $\varepsilon_{11}\varepsilon_{22}$-plane, see Fig. 1. We will investigate the sensitivity of the QoI with respect to eight parameters, namely, a, f_0, e_c, e_d, α, $\bar{c}$, β, and the initial condition e_0. The initial condition $\boldsymbol{\varepsilon}(0) = 0 \cdot \mathbf{I}$ and the loading parameters σ_1 and σ_2, see (13), remain fixed.

Let us make the listed parameters uncertain. To this end, we define the vector $u = (0.75, 0.8, 0.85, 0.9, 0.95, 1,\ 1.05, 1.1, 1.15, 1.20, 1.25)$ and vectors $a^{\text{unc}} = au$, $f_0^{\text{unc}} = f_0 u$, $e_c^{\text{unc}} = e_c u$, etc., where the nominal values from the first paragraph of this section are used in a, f_0, e_c, etc. In other words, the nominal values will be decreased or increased in 5% steps up to 25%.

The only exception is β, for which we define $\beta^{\text{unc}} = 0.55 + 0.6u$ to allow for a significant amount of uncertainty covering also the nominal value of 1.03.

Our goal is to record the QoI (the limit of the strain) if uncertainty is considered in one parameter and the other parameters remain fixed at their nominal values.

For $s_{11} = -1.5$ and $s_{22} = s_{33} = -1$, the computation shows that the influence of the uncertainty in the parameters can be divided into two groups. One group is

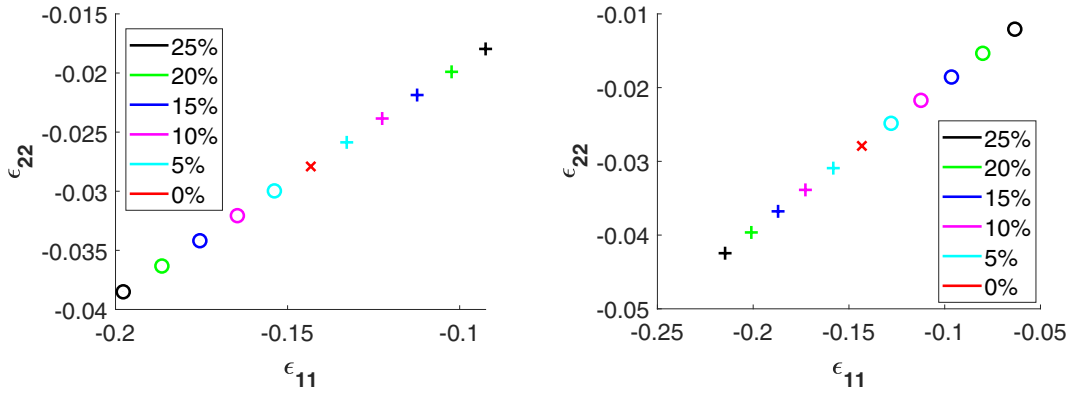


Figure 2: Response to the uncertainty in e_d (left) and e_0 (right). The symbol $\circ$ at the extreme position corresponds to the -25% change of the parameter nominal value, the symbol $+$ at the extreme position corresponds to the $+25\%$ change of the parameter nominal value, the symbol $\times$ marks the QoI if the parameter has its nominal value.

formed by f_0^{unc}, e_c^{unc}, α^{unc}, $\bar{c}^{\text{unc}}$, β^{unc} and the other by a^{unc}, e_d^{unc}, e_0^{unc}. In the former group, the uncertain input causes a response (in ε_{11} as well as ε_{22}) within 0.2%–5% of the response to the nominal values. The response is significantly stronger in the latter group and reaches 30%–60% as we can see in Fig. 2. Regarding a, the responses are located on a segment-like curve that starts at $(-0.18, -0.1)$ for -25% and ends at $(-0.125, -0.036)$ for $+25\%$.

Let us make the anisotropy stronger by setting $s_{11} = -2.0$ and $s_{22} = s_{33} = -1$. The basic division into two groups of parameters has been preserved though the response to $\bar{c}^{\text{unc}}$ has increased to $\pm 10\%$ of the reference value in ε_{22}, for instance. The range of the response to a^{unc} is larger and comprises both positive and negative values, that is, even the sign of the ε_{22} strain component is uncertain if sufficient amount of uncertainty is present in a, see Fig. 3. The responses to e_d^{unc} and e_0^{unc} are quite similar, only the latter is depicted in Fig. 3.

Since, as we observe, the anisotropy of $\boldsymbol{\sigma}$ has a strong influence on the range spread of QoI, we also investigate the model response to uncertainty in s_{11} with $s_{22} = s_{33}$ fixed to -1. We choose $s_{11} = -2$ as the nominal value that will be perturbed up to $\pm 25\%$. Fig. 4 shows the model responses.

4. Observations and comments

In Fig. 4 (right), we observe that the response of $\varepsilon_{22} = \varepsilon_{33}$ is close to zero if $s_{11} = -1.8$ or $s_{11} = -1.9$. As a consequence, a relatively small perturbation can cause a sign change. Negative strain means compression (in the corresponding direction), whereas positive strain means expansion. The latter phenomenon appears if the **S**-anisotropy is strong enough to make the material compressed in the 1-direction and (with a smaller magnitude) expanded in the 2- and 3-direction. This is also il-

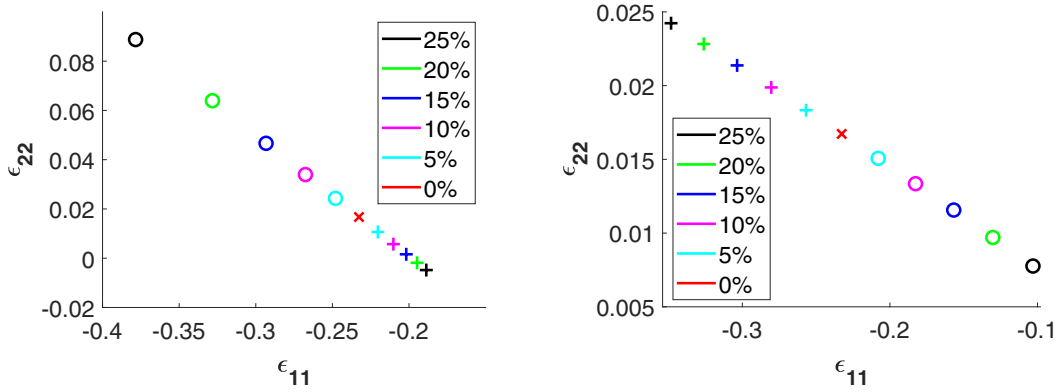


Figure 3: Response to the uncertainty in a (left) and e_0 (right). The symbol $\circ$ at the extreme position corresponds to the -25% change of the parameter nominal value, the symbol $+$ at the extreme position corresponds to the $+25\%$ change of the parameter nominal value, the symbol $\times$ marks the QoI if the parameter has its nominal value.

lustrated by Fig. 2 (right) and Fig. 3 (right). In the former case ($s_{11} = -1.5$), compression is observed in the 2-direction, whereas expansion comes in the latter case where $s_{11} = -2$.

Nevertheless, the computation shows that the void ratio $e(t)$ tends to its limit e_d (not depicted), that is, that the material is compacted for $s_{11} \in [-2.5, -1.5]$.

We also observe that changes of parameter values can result in an amplified or attenuated total effect. The latter is illustrated by Fig. 3, where an increase in a decreases the magnitude of both ε_{11} and ε_{22}, but an increase in e_0 has an opposite effect.

In some cases, a parameter-to-response mapping is nonlinear, see, for instance, Fig. 3 (left), where the negative perturbations have significantly stronger effect than the positive perturbations.

Although most of the graphs show points $[\varepsilon_{11}, \varepsilon_{22}]$ distributed along a line, a closer inspection would reveal a slight nonlinearity. A stronger nonlinear behaviour is depicted in Fig. 4 (left). The question arises whether the observed tendency to form an almost linear pattern has a deep reason rooted in the setting of the model, or whether it is simply the consequence of a limited amount of uncertainty that prevents nonlinearities to be fully developed; see Fig. 4 (left) where the circles form a linear pattern for $\beta \in [1, 1.12]$ that becomes curved if β belongs to $[1.15, 1.3]$, that is, to the interval relatively distant from the nominal value $\beta = 1.03$.

Readers familiar with fuzzy sets certainly noticed that relevant membership functions can be constructed on the basis of Fig. 2–4. Indeed, the ε_{11}-distance between two marks determined by the same (except for the sign) perturbation percent is the length of an α-cut of a membership function representing the fuzziness of ε_{11} induced by the fuzziness of one input parameter. Similarly for ε_{22}. However, a more elaborate approach is necessary if a fuzzy set based on Fig. 4 (left) is to be inferred.

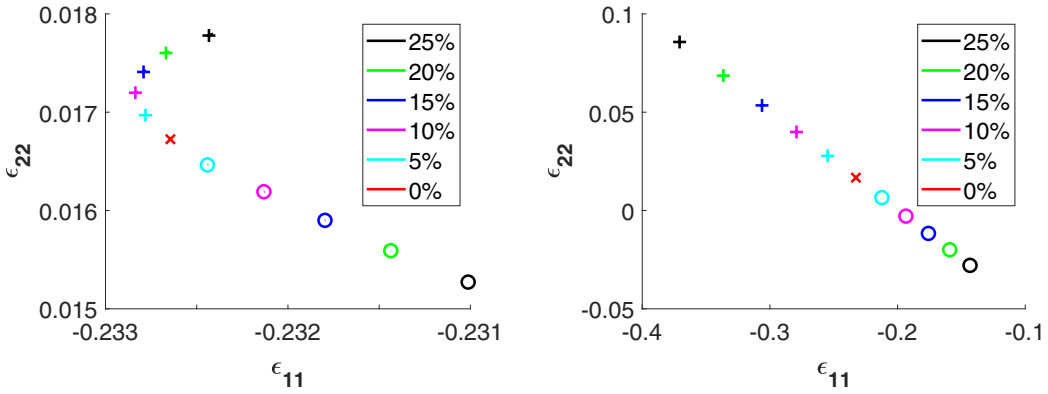


Figure 4: Left: Response to the uncertainty in β. The symbol $\circ$ at the extreme position corresponds to $\beta = 1$, whereas the symbol $+$ at the extreme position marks the response to $\beta = 1.3$; the β-step is equal to 0.03. Right: Response to the uncertainty in s_{11}. The symbol $\circ$ at the extreme position corresponds to the response to $s_{11} = -1.5$, whereas the symbol $+$ at the extreme position marks the response to $s_{11} = -2.5$; the s_{11}-step is equal to 0.1.

All the calculations were performed in the MATLAB® environment.

Acknowledgments

The research of the first author was supported by the project Centre of Advanced Applied Sciences (CAAS) with the number: CZ.02.1.01/0.0/0.0/16_019/0000778. CAAS is co-financed by the European Union. The first author is also grateful to Dr. Richard (Dick) Haas for fruitful discussions. The work of the second author was supported by the Grant Agency of the Czech Technical University in Prague, grant No. SGS22/005/OHK1/1T/11. The work of the third author was supported by the Czech Science Foundation, project No. 20-14736S. The research was also supported by the OeAD Scientific & Technological Cooperation (WTZ CZ 18/2020: "Hysteresis in Hypo-Plastic Models") financed by the Austrian Federal Ministry of Science, Research and Economy (BMWFW) and by the Czech Ministry of Education, Youth and Sports (MŠMT).

References

[1] Alonso-Marroquín, F., Mühlhaus, H.B., and Herrmann, H.J.: Micromechanical investigation of granular ratcheting using a discrete model of polygonal particles. Particuology **6** (2008), 390–403. URL https://www.sciencedirect.com/science/article/pii/S1674200108001430.

[2] Bauer, E. et al.: On proportional deformation paths in hypoplasticity. Acta Mechanica **231** (2020), 1603–1619. URL https://link.springer.com/article/10.1007/s00707-019-02597-3.

[3] Calvetti, F. and di Prisco, C.: Discrete numerical investigation of the ratcheting phenomenon in granular materials. Comptes Rendus Mécanique **338** (2010), 604–614. Micromechanics of granular materials, URL https://www.sciencedirect.com/science/article/pii/S1631072110001567.

[4] Karrech, A., Seibi, A., and Duhamel, D.: Finite element modelling of rate-dependent ratcheting in granular materials. Computers and Geotechnics **38** (2011), 105–112. URL https://www.sciencedirect.com/science/article/pii/S0266352X10001187.

[5] Kolymbas, D.: An outline of hypoplasticity. Arch. Appl. Mech. **61** (1991), 143–151. URL https://link.springer.com/article/10.1007/BF00788048.

[6] Krejčí, P., Monteiro, G.A., Runcziková, J., Bauer, E., and Kovtunenko, V.A.: Stress-controlled ratchetting in hypoplasticity (submitted to Acta Mechanica in October 2022).

Programs and Algorithms of Numerical Mathematics 21
J. Chleboun, P. Kůs, J. Papež, M. Rozložník, K. Segeth, J. Šístek (Eds.)
Institute of Mathematics CAS, Prague 2023

VALUATION OF TWO-FACTOR OPTIONS UNDER THE MERTON JUMP-DIFFUSION MODEL USING ORTHOGONAL SPLINE WAVELETS

Dana Černá

Department of Mathematics and Didactics of Mathematics
Technical University of Liberec
Studentská 2, 461 17 Liberec, Czech Republic
dana.cerna@tul.cz

Abstract: This paper addresses the two-asset Merton model for option pricing represented by non-stationary integro-differential equations with two state variables. The drawback of most classical methods for solving these types of equations is that the matrices arising from discretization are full and ill-conditioned. In this paper, we first transform the equation using logarithmic prices, drift removal, and localization. Then, we apply the Galerkin method with a recently proposed orthogonal cubic spline-wavelet basis combined with the Crank–Nicolson scheme. We show that the proposed method has many benefits. First, as is well-known, the wavelet-Galerkin method leads to sparse matrices, which can be solved efficiently using iterative methods. Furthermore, since the basis functions are cubic splines, the method is higher-order convergent. Due to the orthogonality of the basis functions, the matrices are well-conditioned even without preconditioning, computation is simplified, and the required number of iterations is less than for non-orthogonal cubic spline-wavelet bases. Numerical experiments are presented for European-style options on the maximum of two assets.

Keywords: wavelet-Galerkin method, Crank–Nicolson scheme, orthogonal spline wavelets

MSC: 65T60, 65M60, 47G20, 60G51

1. Introduction

Numerous models have been developed for the fair pricing of options. These models include the famous Black–Scholes and stochastic volatility models, which assume that the underlying asset price is a continuous function of time. This assumption, however, is not always consistent with the behavior of real market prices. Therefore, several models have been developed which allow for jumps in the price of the underlying. This paper focuses on one of these models, the Merton jump-diffusion model

DOI: 10.21136/panm.2022.05

with two assets, represented by a nonstationary partial integro-differential equation (PIDE) with two state variables. From the mathematical point of view, it is not straightforward to solve this model numerically due to several difficulties. First, the integral term results in linear systems with full matrices for many standard methods, such as the finite difference and finite element methods. Moreover, the integral term requires the computation of four-dimensional integrals. Furthermore, the differential operator is degenerate, and functions representing the initial conditions are typically not smooth.

Option pricing is a central topic in financial mathematics, and there is a vast amount of literature concerning the numerical valuation of options. However, when it comes to multi-dimensional jump-diffusion models, due to the difficulties mentioned above, there are only a few studies on numerical methods for their solution. Thus, this remains an important and active field of research. An implicit finite difference scheme combined with fixed-point iterations was proposed for two-asset jump-diffusion models in [9]. Operator-splitting methods and various-time stepping schemes were studied in [4]. Wavelet-based methods have also been employed for multi-dimensional models, for example, in [7, 11, 13]. In [7], the wavelet-Galerkin method was used for the two-asset Merton model. Compared with [7], the method proposed in this article uses orthogonal wavelet bases and includes transformation into logarithmic prices and drift removal, resulting in a different variational problem.

As already mentioned, the standard methods used for PIDEs typically lead to full matrices. In contrast, the Galerkin method with a wavelet basis leads to matrices that can be closely approximated by sparse matrices, as discussed in [2, 6, 11]. This paper uses the Galerkin method with orthogonal cubic spline wavelets combined with the Crank–Nicolson scheme. The aim is to show that this method is suitable and efficient for the two-asset Merton model. Its advantages are that the resulting system's matrices are sparse and uniformly conditioned, higher-order convergence is achieved, and a small number of iterations is needed to solve the system to the required accuracy.

2. The two-asset Merton model

The two-asset Merton model is a generalization of the original Merton jump-diffusion model developed in [12]. The model assumes that the price S_τ^i of the asset i at time τ follows the jump-diffusion process

$$\ln\left(\frac{S_\tau^i}{S_0^i}\right) = \left(r - \frac{\sigma_i^2}{2} - \lambda\kappa_i\right)\tau + \sigma_i W_\tau^i + \sum_{k=1}^{N_\tau} Y_k^i, \quad i = 1, 2, \tag{1}$$

see [3, 4]. The parameters in the model have the following interpretation. The parameter r represents the risk-free interest rate, and σ_i is the volatility of asset i corresponding to the diffusion part of the process. The processes W_τ^1 and W_τ^2 are Wiener processes with correlation coefficient ρ. The number of price jumps is represented by the Poisson process N_τ with intensity λ. The random variables Y_k^i are

independent and identically distributed for a given i. The parameter κ_i is the expected relative jump size, $\kappa_i = E(\mathrm{e}^{Y_k^i} - 1)$.

The process represented by (1) is a general jump-diffusion process. The Merton model further assumes that $\mathrm{e}^{Y_k^1}$ and $\mathrm{e}^{Y_k^2}$ have the bivariate log-normal distribution with density

$$f(y_1, y_2) = \frac{K}{y_1 y_2} \exp\left(- \frac{\left(\frac{\ln y_1 - \gamma_1}{\delta_1}\right)^2 + \left(\frac{\ln y_2 - \gamma_2}{\delta_2}\right)^2 - 2\hat{\rho}\left(\frac{\ln y_1 - \gamma_1}{\delta_1}\right)\left(\frac{\ln y_2 - \gamma_2}{\delta_2}\right)}{2(1-\hat{\rho}^2)} \right), \quad (2)$$

where $K = 1/2\pi\delta_1\delta_2\sqrt{1-\hat{\rho}^2}$. Let T be the maturity date, and let S_i be the price of asset i. Then, $t = T - \tau$ is the time to maturity and the option value $V(S_1, S_2, t)$ satisfies [4, 9, 12]

$$\frac{\partial V}{\partial t} - \mathcal{L}_D(V) - \mathcal{L}_I(V) = 0, \quad S_1, S_2 \in (0, \infty), \, t \in (0, T), \quad (3)$$

where $\mathcal{L}_D$ is a degenerate differential operator defined as

$$\begin{aligned} \mathcal{L}_D(V) = & \frac{\sigma_1^2 S_1^2}{2}\frac{\partial^2 V}{\partial S_1^2} + \rho\sigma_1\sigma_2 S_1 S_2 \frac{\partial^2 V}{\partial S_1 \partial S_2} + \frac{\sigma_2^2 S_2^2}{2}\frac{\partial^2 V}{\partial S_2^2} \\ & + (r - \lambda\kappa_1) S_1 \frac{\partial V}{\partial S_1} + (r - \lambda\kappa_2) S_2 \frac{\partial V}{\partial S_2} - (r + \lambda) V \end{aligned} \quad (4)$$

and $\mathcal{L}_I$ is an integral operator given by

$$\mathcal{L}_I(U) = \lambda \int_0^\infty \int_0^\infty V(S_1 y_1, S_2 y_2, t) \, f(y_1, y_2) \, \mathrm{d}y_1 \, \mathrm{d}y_2. \quad (5)$$

The degeneracy means that for $S_1 = 0$ and $S_2 = 0$, some second-order terms of the differential operator $\mathcal{L}_D$ vanish. The first-order terms of $\mathcal{L}_D$ represent drift.

The initial and boundary conditions depend on the type of option. We consider a European option on the maximum of two assets as an example. This option gives its holder the right, but not the obligation, to sell (for a put option) or buy (for a call option) the most expensive of two underlying assets at the strike price K at maturity T. In this case, the initial condition representing the value of an option at maturity is

$$V(S_1, S_2, 0) = \begin{cases} \max(K - \max(S_1, S_2), 0) & \text{for a put option,} \\ \max(\max(S_1, S_2) - K, 0) & \text{for a call option.} \end{cases} \quad (6)$$

3. Transformation and variational formulation

Two main approaches are typically employed for the numerical solution of PDEs and PIDEs representing option-pricing problems. The first approach moves directly

to a variational formulation of the equation with the degenerate differential operator. In this case, the analysis has to be carried out using weighted Sobolev spaces, and error estimates are available only in the norms of these spaces. This approach has been studied for PDE models, for example, in [1], and for the Merton model in [7].

This paper focuses on the second approach, which is based on transformation into logarithmic prices. This has the advantage that it removes the degeneracy of the differential operator. Therefore, standard Sobolev spaces are used for the analysis and error estimates. Various papers have studied this approach, but mainly for PDE models. For PIDEs, we refer to [11, 13].

Hence, we first adjust (3). The degeneracy and drifts are removed using the substitution $U(x_1, x_2, t) = V(S_1, S_2, t)$, where $x_i = \log S_i - (\sigma_i^2/2 + \lambda\kappa_i - r)\, t$ for $i = 1, 2$. Then, the unbounded domain $\mathbb{R}^2$ for (x_1, x_2) is approximated by a bounded domain $\Omega = I \times I$, where I is a chosen finite interval. Finally, as in [11], we set U to zero outside Ω, that is,

$$U(x_1, x_2, t) = 0, \quad (x_1, x_2) \in \mathbb{R}^2 \backslash \Omega, \quad t \in (0, T). \tag{7}$$

Note that this homogeneous Dirichlet boundary condition is artificial and does not describe the actual situation. However, setting this condition simplifies the method and does not affect the solution significantly in the parts of Ω which are not close to the boundary, when Ω is large enough, see [11].

After these adjustments, we obtain an elliptic differential operator

$$\mathcal{D}(U) = \frac{\sigma_1^2}{2}\frac{\partial^2 U}{\partial x_1^2} + \rho\sigma_1\sigma_2 \frac{\partial^2 U}{\partial x_1 \partial x_2} + \frac{\sigma_2^2}{2}\frac{\partial^2 U}{\partial x_2^2} - (r + \lambda)\, U \tag{8}$$

and an integral operator

$$\mathcal{I}(U) = \lambda \iint_{\Omega} U(t_1, t_2, t)\, g(t_1 - x_1, t_2 - x_2)\, \mathrm{d}t_1\, \mathrm{d}t_2, \tag{9}$$

where $g(x_1, x_2) = f(\mathrm{e}^{x_1}, \mathrm{e}^{x_2})\, \mathrm{e}^{x_1} \mathrm{e}^{x_2}$. The transformed equation has the form

$$\frac{\partial U}{\partial t} = \mathcal{D}(U) + \mathcal{I}(U). \tag{10}$$

Let $\langle \cdot, \cdot \rangle$ denote the L^2 inner product. To derive a variational formulation, we define a bilinear form $a = a_D - a_I$, where

$$\begin{aligned} a_D(u, v) = &\frac{\sigma_1^2}{2}\left\langle \frac{\partial u}{\partial x_1}, \frac{\partial v}{\partial x_1} \right\rangle + \rho\sigma_1\sigma_2 \left\langle \frac{\partial u}{\partial x_1}, \frac{\partial v}{\partial x_2} \right\rangle \\ &+ \frac{\sigma_2^2}{2}\left\langle \frac{\partial u}{\partial x_2}, \frac{\partial v}{\partial x_2} \right\rangle + (r + \lambda)\, \langle u, v \rangle \end{aligned} \tag{11}$$

and $a_I(u, v) = \langle \mathcal{I}(u), v \rangle$. Let $U_0 \in L^2(\Omega)$ be the transformed payoff function. The variational formulation consists in determining the function $U \in L^2(0, T; H_0^1(\Omega))$ such that $\frac{\partial U}{\partial t} \in L^2(0, T; H^{-1}(\Omega))$ and

$$\left\langle \frac{\partial U}{\partial t}, v \right\rangle + a(U, v) = 0 \quad \forall v \in V, \text{ a.e. in } (0, T), \quad U(x_1, x_2, 0) = U_0(x_1, x_2). \tag{12}$$

4. The orthogonal cubic spline-wavelet basis

There is not a universally accepted definition of a wavelet basis in the mathematical literature. Here, we consider a wavelet basis of the space $L^2(I)$, where I is a bounded interval, in the following sense. Let $\mathcal{J}$ be an index set such that $\lambda \in \mathcal{J}$ takes the form $\lambda = (j,k)$ and $|\lambda| = j$ denotes the level. Then, $\Psi = \{\psi_\lambda, \lambda \in \mathcal{J}\}$ is a wavelet basis of $L^2(I)$ if it satisfies the following four conditions:

(i) The set Ψ is an orthogonal basis for $L^2(I)$.

(ii) The basis functions are local, i.e., diam supp $\psi_\lambda \leq C2^{-|\lambda|}$ for all $\lambda \in \mathcal{J}$.

(iii) The set Ψ has a hierarchical structure,

$$\Psi = \Phi_{j_0} \cup \bigcup_{j=j_0}^{\infty} \Psi_j, \quad \Phi_{j_0} = \{\phi_{j_0,k}, k \in \mathcal{I}_{j_0}\}, \quad \Psi_j = \{\psi_{j,k}, k \in \mathcal{J}_j\}. \tag{13}$$

The functions $\phi_{j_0,k}$ are called scaling functions and the functions $\psi_{j,k}$ are called wavelets.

(iv) The wavelets have vanishing moments, that is, $\langle p, \psi_{j,k}\rangle = 0$, $k \in \mathcal{J}_j$, $j \geq j_0$, for any polynomial p of degree less than $L \geq 1$, where L depends on the wavelet type.

The method in this paper uses orthogonal cubic spline wavelets on the interval with four vanishing moments, recently constructed in [8] using general principles from [10]. The scaling functions in the inner part of the interval are defined as translations and dilations of six generators, which are illustrated in Fig. 1. In addition, boundary functions are constructed near the endpoints of the interval.

Similarly, wavelets in the inner part of the interval are constructed as translations and dilations of six generators, and several boundary functions need to be added. Plots of the wavelet generators are shown in Fig. 2. Since all the basis functions are cubic splines, they are given in closed form and can be handled easily. The resulting basis satisfies the conditions $i) - iv)$ above.

The two-dimensional wavelet basis is constructed using so-called anisotropic tensor products of these one-dimensional bases see [6, 8], that is, it contains the functions $\psi_\lambda = \psi_{\lambda_1} \otimes \psi_{\lambda_2}$, where ψ_{λ_1} and ψ_{λ_2} are univariate basis functions. Then $|\lambda| = \max(\lambda_1, \lambda_2)$ is a level of ψ_λ. Furthermore, we denote $[\lambda] = \min(\lambda_1, \lambda_2)$.

5. The orthogonal wavelet method

Let Ψ^k contain all basis functions up to level k and let $X_k = \operatorname{span} \Psi^k$. Let $U_{k,0}$ be an approximation of $U_0 \in L^2(\Omega)$ in X_k. The wavelet-Galerkin method consists in finding a solution $U_k \in L^2(0,T;X_k)$ such that $\frac{\partial U_k}{\partial t} \in L^2(0,T;X_k')$ and the equation

$$\left\langle \frac{\partial U_k}{\partial t}, v_k \right\rangle + a(U_k, v_k) = 0, \quad U_k(x_1, x_2, 0) = U_{k,0}(x_1, x_2) \tag{14}$$

is satisfied for all $v_k \in X_k$ and almost everywhere in $(0,T)$.

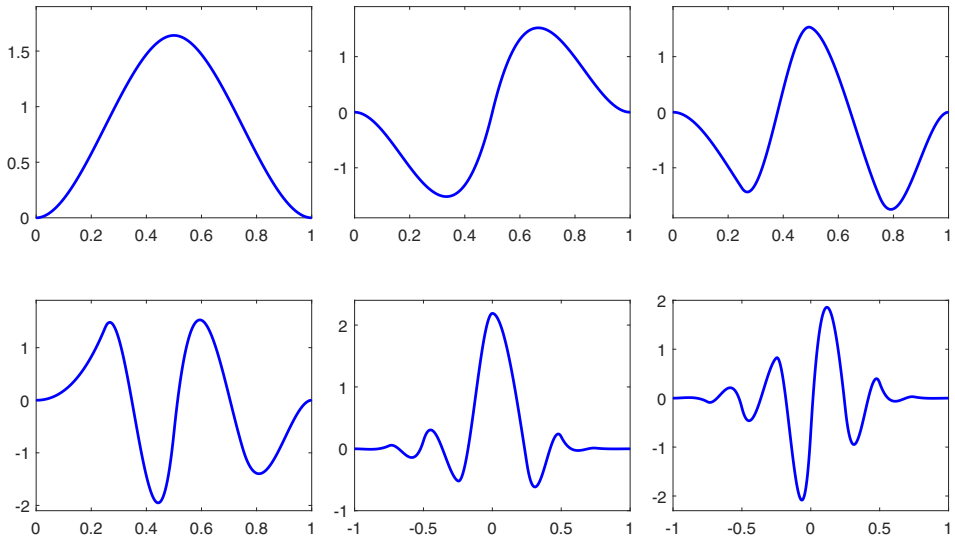


Figure 1: Generators of inner orthogonal cubic spline scaling functions.

The Crank–Nicolson scheme is used for temporal discretization to obtain a fully discrete scheme. Let $M \in \mathbb{N}$, $\tau = T/M$, $t_l = l\tau$ for $l = 0, \ldots, M$, and let $U_k^l(x_1, x_2) = U_k(x_1, x_2, t_l)$. The aim is to find a solution U_k^l of the equation

$$\frac{\langle U_k^{l+1}, v_k \rangle}{\tau} - \frac{\langle U_k^l, v_k \rangle}{\tau} + \frac{a\left(U_k^{l+1}, v_k\right)}{2} + \frac{a\left(U_k^l, v_k\right)}{2} = 0, \quad U_k^0 = U_{k,0} \tag{15}$$

for all $v_k \in X_k$.

Now, we expand the solution U_k^l in the basis Ψ^k,

$$U_k^l = \sum_{\psi_\lambda \in \Psi^k} \left(c_k^l\right)_\lambda \psi_\lambda, \tag{16}$$

set $v_k = \psi_\mu$, and substitute it into (15).

Let $\mathbf{G}^k$ and $\mathbf{K}^k$ be matrices corresponding to the differential and integral terms, respectively, defined as

$$\mathbf{G}_{\mu,\lambda}^k = \frac{\langle \psi_\lambda, \psi_\mu \rangle}{\tau} + a_D(\psi_\lambda, \psi_\mu), \quad \mathbf{K}_{\mu,\lambda}^k = a_I(\psi_\lambda, \psi_\mu), \quad \psi_\lambda, \psi_\mu \in \Psi^k. \tag{17}$$

Furthermore, let the vector $\mathbf{f}_k^l$ be defined as

$$\left(\mathbf{f}_k^l\right)_\mu = \frac{\left(U_k^l, \psi_\mu\right)}{\tau} - \frac{a\left(U_k^l, \psi_\mu\right)}{2}, \quad \psi_\mu \in \Psi^k. \tag{18}$$

Then, for $l = 1, \ldots, M$, the column vector $\mathbf{c}_k^{l+1}$ of coefficients $\left(c_k^{l+1}\right)_\lambda$ is a solution of the linear system $\mathbf{A}^k \mathbf{c}_k^{l+1} = \mathbf{f}_k^l$, where $\mathbf{A}^k = \mathbf{G}^k - \mathbf{K}^k$.

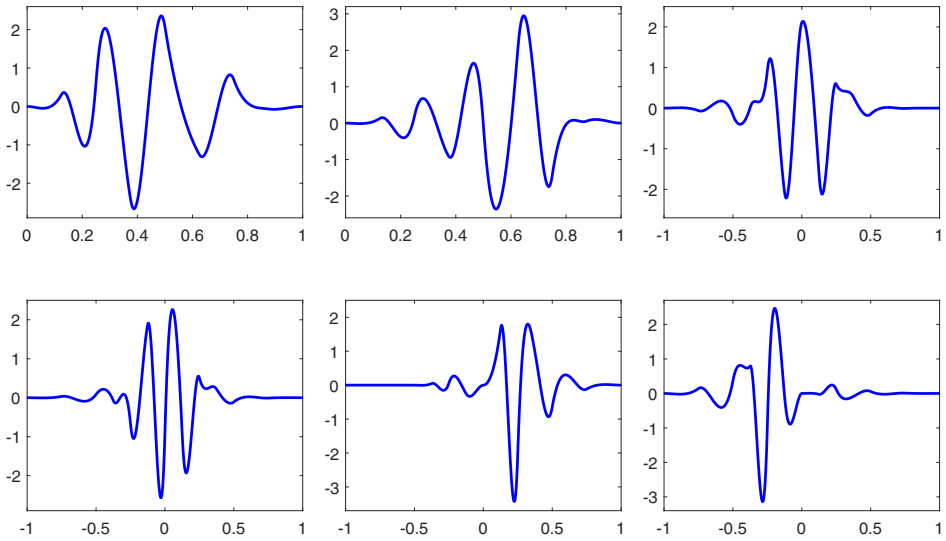


Figure 2: Generators of inner orthogonal cubic spline wavelets.

For the numerical solution of this system, the GMRES method is used. No preconditioning of the system is necessary because the use of orthogonal wavelets ensures that the system is well-conditioned similarly as in [8].

Since the matrix $\mathbf{G}^k$ corresponds to the differential operator, it is sparse. The GMRES method requires multiplying the matrix $\mathbf{G}^k$ with a vector. This can be realized using the Kronecker product of matrices corresponding to one-dimensional differential operators, as detailed in [8]. Due to the L^2 orthogonality of the basis, some of these matrices are identity matrices, which positively affects the resulting condition number of the matrix $\mathbf{G}^k$ and greatly simplifies the computation.

The next theorem, for which a proof can be found in [6], yields a decay estimate for the entries of the matrix $\mathbf{K}^k$.

Theorem 1. *Let ψ_λ and ψ_μ be wavelets with $L = 4$ vanishing moments, as defined in Section 4. Then there exists a real constant C such that*

$$|a_I(\psi_\lambda, \psi_\mu)| \leq C 2^{-(L+1)([\lambda]+[\mu])}. \tag{19}$$

By Theorem 1, the entries of the matrix $\mathbf{K}^k$ decrease exponentially. Thus, many of them are very small and can be set to zero. This process is called compression of the matrix $\mathbf{K}^k$. For the compression strategy and the effect of compression, we refer to [6].

6. Numerical example

Numerical results are presented for a benchmark example from [4, 7]. The market values of European put and call options on the maximum of two assets are evaluated

using the proposed method. The advantage of considering options on the maximum of two assets is that, in this special case, the analytic solution is known [3], which enables us to compute the errors of the numerical solution.

The parameters for the options are set as in [4, 7]. The strike price is $K = 100$, the risk-free interest rate is $r = 0.05$, the volatilities of the asset prices are $\sigma_1 = 0.12$ and $\sigma_2 = 0.15$, and the correlation coefficient for the asset prices is $\rho = 0.3$. The parameters for the jump part of the process are $\lambda = 0.6$, $\gamma_1 = -0.1$, $\gamma_2 = 0.1$, $\hat{\rho} = -0.20$, $\delta_1 = 0.17$, and $\delta_2 = 0.13$. The time to maturity is $T = 1$ year. A sufficiently large domain for (S_1, S_2) has to be chosen, therefore we set it to be $(0.1, 5K)^2$. Plots of the resulting functions representing prices of put and call options are shown in Fig. 3. Since artificial boundary conditions are used, the plot of the put option price is shown only in the region $(1, 200)^2$ and the plot of the call option price is shown only in the region $(1, 150)^2$, to avoid the area near $S_1 = 0$ and $S_2 = 0$.

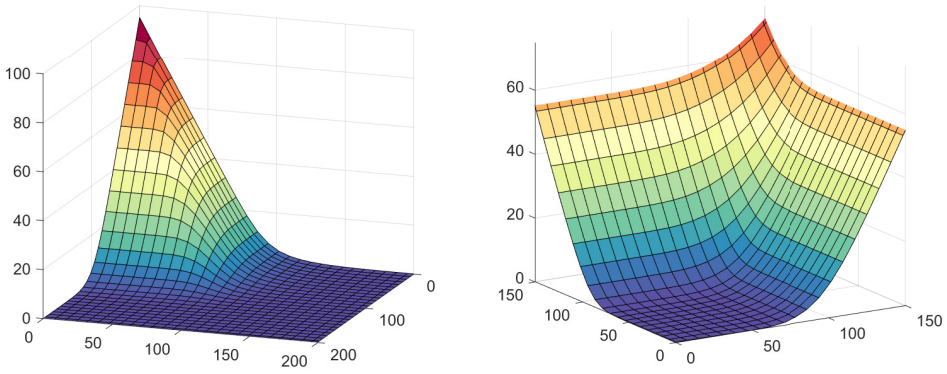


Figure 3: The functions representing prices of European put (left) and call (right) options at one year to maturity.

Table 1 lists the resulting values of options, errors, and numbers of iterations. In this table, N denotes the number of basis functions and M denotes the number of time steps. The values V_P represent the computed prices of options for asset prices (S_1, S_2) equal to $P = (100, 100)$. The corresponding pointwise error is denoted by ρ_P. We set a region of interest as $\mathrm{ROI} = (K/2, 3K/2)^2$ and compute the L^∞ (ROI) error ρ_∞ and the L^2 (ROI) error ρ_2. For the numerical solution, the GMRES method without restart is used. The GMRES iterations are set to stop when the relative residual is less than 10^{-9}. The number of iterations is denoted by it.

Conclusions

A wavelet-based method is proposed for pricing European-style two-factor options under the Merton jump-diffusion model. The first important step of the method is adjusting the original integro-differential equation, including transformation into logarithmic prices, drift removal, and localization. After these adjustments, it is possible

type	N	M	V_P	ρ_P	ρ_∞	ρ_2	it
put	144	2	2.98889	-1.46e0	6.55e0	2.30e-1	7
	576	8	1.30073	-1.19e-1	1.29e0	4.14e-2	7
	2304	32	1.19320	-1.11e-2	2.90e-1	4.61e-3	6
	9216	128	1.18752	-5.46e-3	2.67e-2	6.30e-4	6
call	144	2	20.6315	-4.23e0	2.63e1	3.37e-1	8
	576	8	15.7047	6.93e-1	2.73e0	3.29e-2	7
	2304	32	16.3922	5.51e-3	1.50e-1	1.30e-3	6
	9216	128	16.3923	5.46e-3	6.30e-2	4.99e-4	6

Table 1: Option values V_P for $P = (100, 100)$, pointwise errors ρ_P, L^∞ errors ρ_∞, L^2 errors ρ_2, and numbers of GMRES iterations it.

to remove the degeneracy of the differential operator and derive a variational formulation using standard Sobolev spaces. The variational problem is solved by the Galerkin method with an orthogonal cubic spline-wavelet basis combined with the Crank-Nicolson scheme. It is shown that the method is suitable for the given equation and has many advantages. Due to the vanishing moments of the wavelets, the matrix corresponding to the integral term can be efficiently represented by a sparse matrix, which is not the case in many standard methods. Furthermore, the L^2 orthogonality of the basis results in matrices with uniformly bounded condition numbers even without any preconditioning. Therefore, a sufficiently accurate solution can be obtained using a small number of iterations. Since the basis functions are cubic splines, the method is higher-order convergent. The proposed method could be used to price various options and could be generalized to other jump-diffusion models and options with more assets.

Acknowledgements

This work was supported by grant No. GA22-17028S funded by the Czech Science Foundation.

References

[1] Achdou, Y. and Pironneau, O.: *Computational methods for option pricing.* SIAM, Philadelphia, 2005.

[2] Beylkin, G., Coifman, R., and Rokhlin, V.: Fast wavelet transforms and numerical algorithms I. Communications on Pure and Applied Mathematics **44** (1991), 141–183.

[3] Boen, L.: European rainbow option values under the two-asset Merton jump-diffusion model. J. Comput. Appl. Math. **364** (2021), article no. 112344.

[4] Boen, L. and Hout, K.J.: Operator splitting scheme for the two-asset Merton jump-diffusion model. J. Comput. Appl. Math. **387** (2021), article no. 112309.

[5] Černá, D. and Finěk, V.: Galerkin method with new quadratic spline wavelets for integral and integro-differential equations. J. Comput. Appl. Math. **363** (2020), 426–443.

[6] Černá, D. and Finěk, V.: Wavelet-Galerkin method for second-order integro-differential equations on product domains. In: Singh, H., Dutta, H., and Cavalcanti, M.M. (Eds.), *Topics in Integral and Integro-Differential Equations*, pp. 1–40. Springer, Switzerland, 2021.

[7] Černá, D.: Wavelet method for option pricing under the two-asset Merton jump-diffusion model. In: Chleboun, J., Kůs, P., Přikryl, P., Rozložník, M., Segeth, K., and Šístek, J. (Eds.), *Programs and Algorithms of Numerical Mathematics, Proceedings of Seminar*, pp. 30–39. Institute of Mathematics CAS, Prague, 2021.

[8] Černá, D. and Fiňková, K.: Option pricing under multifactor Black-Scholes model using orthogonal spline wavelets. Preprint, arXiv:2211.13890 [math.NA], 2022.

[9] Clift, S.S. and Forsyth, P.A.: Numerical solution of two asset jump diffusion models for option valuation. Appl. Numer. Math. **58** (2008), 743–782.

[10] Donovan, G.C., Geronimo, J.S., and Hardin, D.P.: Intertwining multiresolution analyses and the construction of piecewise-polynomial wavelets. SIAM J. Math. Anal. **27** (1996), 1791–1815.

[11] Hilber, N., Reichmann, O., Schwab, C., and Winter, C.: *Computational methods for quantitative finance*. Springer, Berlin, 2013.

[12] Merton, R.C.: Option pricing when underlying stock returns are discontinuous. J. Financ. Econ. **3** (1976), 125–144.

[13] Winter, C.: *Wavelet Galerkin schemes for option pricing in multidimensional Lévy models.* Ph.D. thesis, ETH Zurich, 2009.

Programs and Algorithms of Numerical Mathematics 21
J. Chleboun, P. Kůs, J. Papež, M. Rozložník, K. Segeth, J. Šístek (Eds.)
Institute of Mathematics CAS, Prague 2023

IDENTIFICATION OF QUASIPERIODIC PROCESSES IN THE VICINITY OF THE RESONANCE

Cyril Fischer, Jiří Náprstek

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

Abstract: In nonlinear dynamical systems, strong quasiperiodic beating effects appear due to combination of self-excited and forced vibration. The presence of symmetric or asymmetric beatings indicates an exchange of energy between individual degrees of freedom of the model or by multiple close dominant frequencies. This effect is illustrated by the case of the van der Pol equation in the vicinity of resonance. The approximate analysis of these nonlinear effects uses the harmonic balance method and the multiple scale method.

Keywords: dynamical systems, quasiperiodic response, van der Pol equation
MSC: 37C60, 37N05, 70G60, 34A12

1. Introduction

The frequency lock-in effect occurs, e.g., when an elastic profile vibrates in buffeting flow. It is characterised by the fact that the vibration frequency does not follow the vortex shedding frequency but locks onto the natural frequency of the profile. Such behaviour is illustrated in Fig. 1a, see [6], where the linear dependence of the frequency ratio on the stream velocity (which directly relates to the vortex shedding frequency) is interrupted at the ratio 1 for a non-negligible wind velocity interval.

This lock-in effect is usually modelled using the van der Pol equation, which corresponds to a single-degree-of-freedom (SDOF) physical system representing a circular bar in an air flow, see Fig. 1b. The spring in the SDOF model is considered linear, the damping term has the (quadratic) van der Pol character. The flow around the body induces a regular vortex shedding that is, in general, perturbed by a random pressure fluctuations.

The measured response in the lock-in region and its neighbourhood consists of the following cases, cf. also Fig. 1a: *(a)* small stationary amplitudes; the velocity is lower than the critical velocity and the vortex-shedding frequency is lower that the natural frequency; *(b)* the lock-in regime, a stationary vortex-induced resonance, maximal

DOI: 10.21136/panm.2022.06

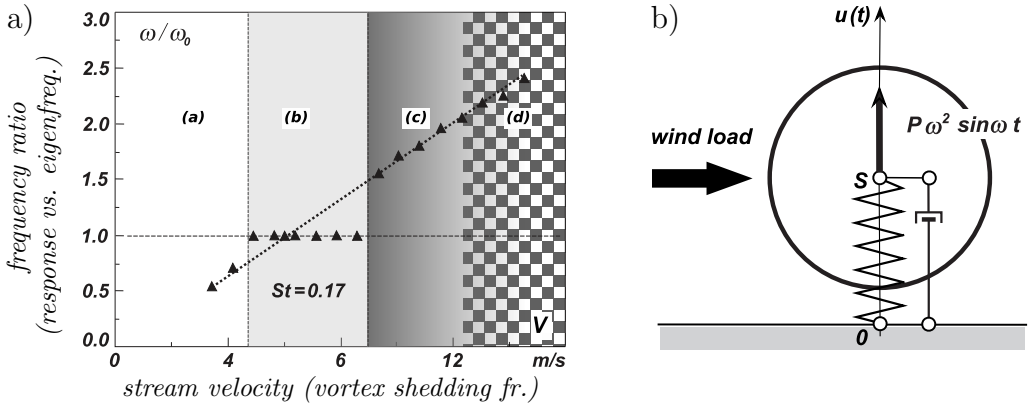


Figure 1: a) frequency ratio vs. the flow velocity and the lock-in domain; b) scheme of the SDOF model, [6]

amplitudes; *(c)* large beating amplitudes, caused by a small detuning of the forcing and natural frequencies outside the lock-in region; *(d)* small non-stationary post-critical vibrations of forced (non-resonant) vibrations caused by vortex shedding with a frequency larger than the natural frequency. The transition between regions *(c)* and *(d)* is not sharp; the influence of the natural frequency decays exponentially with increasing distance from the boundary of the lock-in region *(b)*.

The case *(c)*, i.e., the regime in the neighbourhood of the stationary lock-in, is studied in this contribution. The beating effect is caused by multiple close dominant frequencies which are present in the response. When a small random component is also present, the response has a character of a cyclostationary process [2].

This paper, as part of a larger project, restricts itself to the behaviour of the van der Pol equation solution under deterministic harmonic excitation in a region that is closely adjacent to the lock-in region. The general mathematical model presented in Section 2 is further studied numerically in Sect. 2.1, using the "harmonic balance" method (Sect. 2.2) and the "multiple scales" method (2.3).

2. Mathematical model

Vibration of a slender structure in an airflow is usually modelled using the van der Pol equation with a harmonic right hand side:

$$\ddot{u} - (\eta - \nu u^2)\dot{u} + \omega_0^2 u = P\omega^2 \cos\omega t + h \cdot \xi(t)\,. \tag{1}$$

In Eq. (1), $u(t)$ – displacement $[m]$, $\dot{u}(t)$ – velocity $[ms^{-1}]$, η, ν – parameters of the damping $[s^{-1}, s^{-1}m^{-2}]$, ω_0– eigen-frequency of the adjoint linear SDOF system, ω – excitation frequency of the vortex shedding $[s^{-1}]$, $P\omega^2$ – amplitude of the harmonic excitation (acceleration) $[ms^{-2}]$, h – multiplicative constant $[ms^{-2}]$, $\xi(t)$ – stationary Gaussian random process [1]. In the rest of the paper, $h = 0$ is assumed.

When regarded as a dynamical system, the solution exhibits one stable limit cycle, the rest position $u(t) = 0$ is unstable.

2.1. Numerical analysis

For the first view, the stationary/non-stationary character of the solution to (1) can be analysed numerically in the frequency domain. Figure 2 shows the Fourier analysis of the set of responses obtained for the frequency range near the resonance frequency $\omega_0 = 1$. The dominant peaks of the periodogram for each value of ω are plotted vertically. This way, the ordinate represents the Fourier frequency coefficients present in the response. The color intensity shows the absolute values of the dominant Fourier coefficients on a logarithmic scale. The detuning on the abscissa is defined as $\Delta = \frac{\omega_0^2-\omega^2}{2\omega}$. The stationary lock-in interval of the harmonic response appears for $-0.1 \lesssim \Delta \lesssim 0.12$, although there are clearly two superharmonic components ($\omega = 3\omega_0, 5\omega_0$). The complex behaviour of the nonlinear response is evident from the existence of a subharmonic resonance interval for negative Δ (i.e., for $\omega \approx 1/3$).

The most important aspect for this work is the behaviour at the boundaries of the lock-in intervals. There the dominant frequencies divide into a series of close but distinct frequencies that cause the beating character of the response. Their mutual distances increase rapidly with increasing distance from the lock-in region and cause shortening of the beating periods in the response. It is clear from this that the monochromatic representation of the solution used in the remaining text is only approximate and more accurate estimates will need to be used in the future.

2.2. Analysis based on the harmonic balance

Following the more general approach by the authors in [5], for a weak excitation force and a small detunig, the response can be expected to have an approximately harmonic form with slowly varying amplitude $U(\tau)$ and phase $\varphi(\tau)$, $\tau = \varepsilon t, \varepsilon \ll 1$:

$$u \to U(\tau)\cos(\omega t + \varphi(\tau))\,. \tag{2}$$

The *harmonic balance* procedure consists in multiplying Eq. (1) by $\sin\omega t$ or $\cos\omega t$ and subsequent integration over one period $t \in (0, 2\pi/\omega)$. Since (τ) is "slow time", the variability of U and φ within one period can be neglected and both functions can be treated as constants. Then:

$$\dot{U} = \frac{1}{2}U\left(\eta - \frac{1}{4}\nu U^2\right) - \frac{1}{2}P\omega\sin\varphi\,, \tag{3a}$$

$$\dot{\varphi} = \Delta - \frac{1}{2U}P\omega\cos\varphi\,, \tag{3b}$$

where $\Delta = \frac{\omega_0^2-\omega^2}{2\omega} \approx \omega_0 - \omega$ and the derivative $\dot{U}$, $\dot{\varphi}$ is taken with respect to τ. The stationary amplitude for $\dot{U} = 0$, $\dot{\varphi} = 0$ is given by

$$U^2\left(4\Delta^2 + \left(\eta - \frac{\nu}{4}U^2\right)^2\right) = \omega^2 P^2\,. \tag{4}$$

Stability of admissible solutions can be assessed using two Routh-Hurwitz conditions

$$64\Delta^2 + \left(4\eta - 3\nu U^2\right)\left(4\eta - \nu U^2\right) \geq 0\,, \tag{5a}$$

$$\nu U^2 - 2\eta \geq 0\,. \tag{5b}$$

Amplitudes of possible stationary solutions following Eq. (4), depending on the value of detuning Δ, are shown in Fig. 3. The stable solutions according to the Routh-Hurwitz conditions are shown in solid lines, the unstable parts are dashed. The greyed areas denote negative values of conditions (5a,5b), respectively, i.e., the unstable regions. To complete the picture, the results from numerical simulations of the original Eq. (1) are shown in Fig. 5b. The stationary amplitudes are numerically

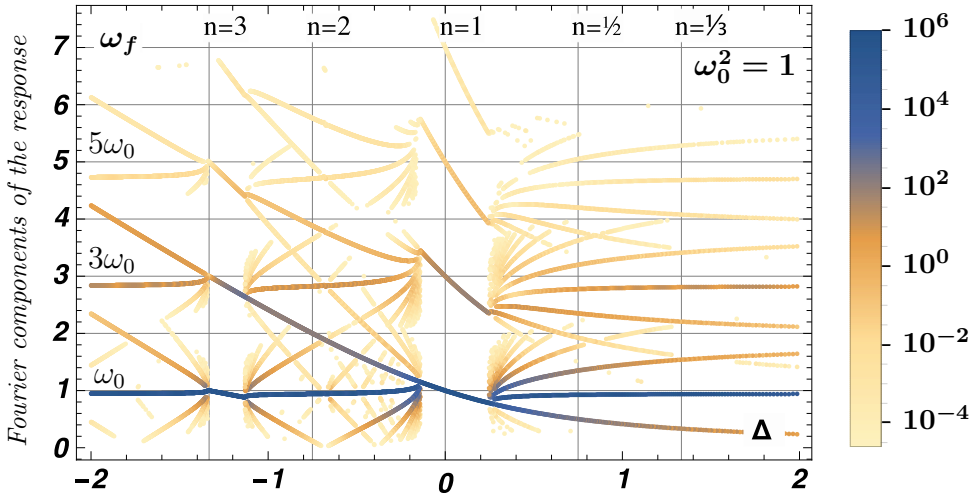


Figure 2: Frequency response characteristics of Eq. (1). The nonzero coefficients of the angular frequency are plotted versus detuning Δ, the colour scale corresponds to the absolute values of the respective Fourier coefficients. Values used: $\eta = 1$, $\nu = 0.5$, $\omega_0 = 1$, $P = 1$.

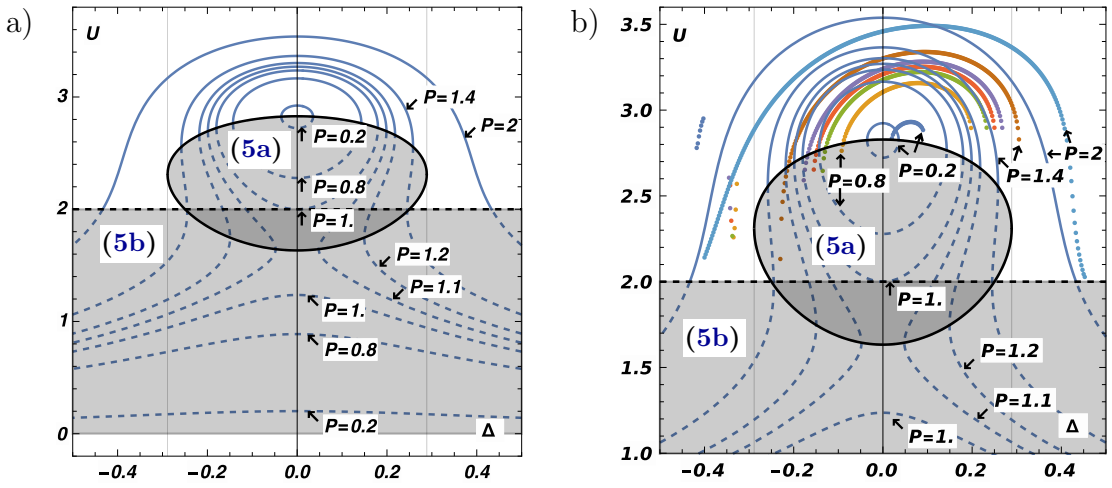


Figure 3: Theoretical amplitudes U of the harmonic solution given by Eq. (4) depending on detuning Δ. Left: stationary amplitudes for different values of the excitation parameter P; right: detailed view together with results from numerical simulations, indicated as colour dots. Values used: $\eta = 1$, $\nu = 0.5$, $\omega = 1$.

identified as those, where the variance of local maxima of the response for fixed values of Δ, P is lower than certain threshold. The numerical results appear to incline to the positive values of detuning Δ, however, the global tendency respects the theoretical mono-harmonic results.

The stationarity condition for the phase shift, $\dot{\varphi} = 0$ in Eq. (3b), introduces a limit value of detuning

$$\Delta_0 = \frac{\omega P}{2U_0}\,, \quad \text{such that} \quad \cos\varphi_0 = \frac{\Delta}{\Delta_0}\,, \tag{6}$$

which indicates the state when the phase shift in Eq. (3b) vanishes for $U_0^2 = 4\,\frac{\nu}{\eta}$. This amplitude corresponds to the horizontal tangent at the top of the region defined by condition (5a). When Δ value varies, the sign of the phase shift changes when crossing $\Delta_0 = \pm\,\frac{1}{4}\,P\omega\sqrt{\nu/\eta}$.

The stationary solution exists for the detunig up to value Δ_s, which is given by the condition of existence of a real solution of Eq. (4). The discriminant of Eq. (4) represents a cubic polynomial equation in Δ^2:

$$\left(64\left(12\Delta^2 - \eta^2\right)^3 + \left(288\Delta^2\eta + 8\eta^3 - 27\nu P^2\omega^2\right)^2\right) = 0\,, \tag{7}$$

which can have up to three real roots. The largest of these, if it exists, defines the boundary detuning of Δ_s, depending on the system and excitation parameters. Unfortunately, there is no simple expression for Δ_s. From the root of the discriminant with respect to $P^2\omega^2$ of Eq. (7), it is possible to find the limiting excitation value for which Eq. (7) is applicable, i.e.,

$$P\omega \geq \frac{4\eta\sqrt{2}}{3\sqrt{3}}\sqrt{\frac{\eta}{\nu}}\,. \tag{8}$$

The limiting amplitude for the values used in Fig. 3 would be $P = 1.53$. For larger values of excitation $P\omega$, the existence of a stable stationary solution is governed only by the RH condition (5b). In such a case, the "ultimate" limit Δ_{s2} follows from Eqs. (4,5b):

$$\Delta_{s2} = \frac{1}{2\sqrt{2}}\sqrt{\frac{\nu}{\eta}}\sqrt{P^2\omega^2 - \frac{\eta^3}{2\nu}}\,. \tag{9}$$

The role of the detuning limit value Δ_s becomes apparent also when a general non-stationary solution to Eq. (3) is assumed. In such a case, after integration

$$\Delta^2 > \Delta_s^2\,, \qquad \varphi = 2\arctan\left(\frac{\Delta - \Delta_s}{D}\,\tan\frac{1}{2}Dt\right)\,, \tag{10a}$$

$$\Delta^2 < \Delta_s^2\,, \qquad \varphi = 2\arctan\left(\frac{D(1 - \mathrm{e}^{Dt})}{2\Delta}\right)\,, \tag{10b}$$

where $D = \sqrt{|\Delta_s^2 - \Delta^2|}$ and (without loss of generality) $t_0 = 0$ has been assumed. The phases in Eqs. (10) represent the asymptotically constant (10b) and periodic

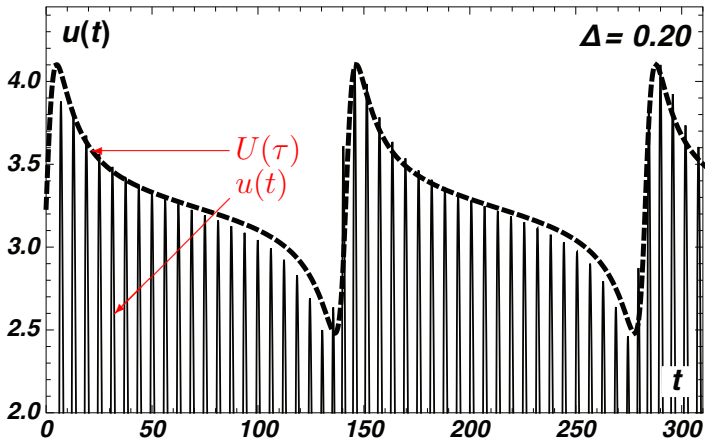


Figure 4: Time plot of numerical solution $u(t)$ and analytical amplitude $U(\tau)$ calculated from Eqs. (10b).

solutions (10a), which represent the stationary and nonstationary amplitudes, respectively. The amplitude can be finally enumerated from Eq. (3).

Figure 4 shows the agreement between numerical and analytical solutions that can be achieved when the initial conditions are carefully matched. In general, however, the agreement is not so good, as it was shown in Fig. 3b. This implies that, if necessary, a multi-harmonic Ansatz for the harmonic balance method or different levels of the perturbation method must be used to obtain more accurate results.

2.3. Analysis based on the multiple scales method

An alternative analytic approach is based on the multiple scales method, [1, 4, 3]. For this purpose, Eq. (1) will be rewritten so that its nonlinear term can be treated as a small quantity

$$\ddot{u} - \epsilon\,(\eta - \nu u^2)\dot{u} + \omega_0^2 u = P\omega^2 \cos\omega t\,, \tag{11}$$

where ϵ is assumed to be a small parameter, $\epsilon \ll 1$. The solution will then be sought in the form of an expansion combining the slow and fast time scale:

$$u(t) \to u_0\,(T_0, T_1) + \epsilon\, u_1\,(T_0, T_1)\,, \quad T_k \to \epsilon^k t. \tag{12}$$

Substituting Eq. (12) into (11) and comparing coefficients of similar powers of ϵ:

$$\epsilon^0 : \frac{\mathrm{d}^2 u_0}{\mathrm{d}T_0^2} + \omega_0^2 u_0 = P\omega^2 \cos(t\omega) \tag{13a}$$

$$\epsilon^1 : \frac{\mathrm{d}^2 u_1}{\mathrm{d}T_0^2} + \omega_0^2 u_1 = \frac{\mathrm{d}u_0}{\mathrm{d}T_0}\left(\nu u_0^2 - \eta\right) - 2\frac{\mathrm{d}^2 u_0}{\mathrm{d}T_0 \mathrm{d}T_1}. \tag{13b}$$

For the homogeneous case ($P = 0$), u_0 satisfying Eq. (13a) can be written as

$$u_0 = A(T_1)\mathrm{e}^{\mathrm{i}\omega_0 T_0} + \overline{A(T_1)\mathrm{e}^{\mathrm{i}\omega_0 T_0}}\,, \tag{14}$$

where A is the function to be determined. The condition of avoiding secular terms in u_1 yields

$$A\left(T_1\right)\left(\nu A\left(T_1\right)\bar{A}\left(T_1\right)-\eta\right)+2A'\left(T_1\right)=0\,. \tag{15}$$

Writing $A\left(T_1\right)=\alpha(T_1)\mathsf{e}^{\mathsf{i}\beta(T_1)}$ for real functions α, β, the stationary ($A'\left(T_1\right)=0$) response amplitude agrees for $\eta>0$ with the parallel solution to Eq. (4):

$$u_0(t)=2\sqrt{\frac{\eta}{\nu}}\cos\left(\omega_0 t\right)\quad\text{and}\quad u_1(t)=-\sqrt{\frac{\eta}{\nu}}\frac{\eta}{4\omega_0}\sin\left(3\omega_0 t\right). \tag{16}$$

For $P>0$, the analogy of Eq. (14) reads

$$u_0=\mathsf{i}\,\Omega_p\mathsf{e}^{\mathsf{i}\omega T_0}-\mathsf{i}\Omega_p\mathsf{e}^{-\mathsf{i}\omega T_0}+A(T_1)\mathsf{e}^{\mathsf{i}\omega_0 T_0}+\overline{A(T_1)}\mathsf{e}^{-\mathsf{i}\omega_0 T_0};\quad \Omega_p=\frac{P}{2\left(\omega^2-\omega_0^2\right)} \tag{17}$$

Then, the RHS in Eq. (13b) for u_1 comprise the following components

$$\begin{aligned}\kappa_1\mathsf{e}^{\mathsf{i}T_0\omega}+\kappa_2\mathsf{e}^{\mathsf{i}T_0\omega_0}+\kappa_3\mathsf{e}^{\mathsf{i}T_0(2\omega+\omega_0)}+\kappa_4\mathsf{e}^{\mathsf{i}T_0(\omega+2\omega_0)}+\\ +\kappa_5\mathsf{e}^{\mathsf{i}T_0(2\omega-\omega_0)}+\kappa_6\mathsf{e}^{\mathsf{i}T_0(\omega-2\omega_0)}+\kappa_7\mathsf{e}^{3\mathsf{i}T_0\omega_0}+\kappa_8\mathsf{e}^{3\mathsf{i}T_0\omega}+\\ +\text{complex conj. terms}\end{aligned} \tag{18}$$

where it has been denoted

$$\begin{aligned}\kappa_1&=\omega\Omega_p\left(\nu\left(2\left|A\left(T_1\right)\right|^2+\Omega_p^2\right)-\eta\right),\\ \kappa_2&=-\mathsf{i}\omega_0\left(2A'\left(T_1\right)+A\left(T_1\right)\left(\nu\left|A\left(T_1\right)\right|^2-\eta+2\nu\Omega_p^2\right)\right),\\ \kappa_3&=\mathsf{i}\nu\Omega_p^2\left(2\omega+\omega_0\right)A\left(T_1\right),\qquad \kappa_4=\nu\Omega_p\left(\omega+2\omega_0\right)A\left(T_1\right)^2,\\ \kappa_5&=\mathsf{i}\nu\Omega_p^2\left(2\omega-\omega_0\right)\bar{A}\left(T_1\right),\qquad \kappa_6=\nu\Omega_p\left(\omega-2\omega_0\right)\overline{A\left(T_1\right)}^2,\\ \kappa_7&=-\mathsf{i}\nu\omega_0 A\left(T_1\right)^3,\qquad \kappa_8=-\nu\omega\Omega_p^3.\end{aligned}$$

For the non-resonant solution in the first approximation, the elimination of secular terms reduces to the condition

$$\kappa_2=0\,. \tag{19}$$

Assuming again $A=\alpha\mathsf{e}^{\mathsf{i}\beta}$ and expanding real and imaginary parts of Eq. (19) one obtains

$$\alpha=0,\,\pm\sqrt{\frac{\eta}{\nu}-\Omega_p^2},\qquad \beta=k\pi\,,\ k\in\mathbb{Z}\,. \tag{20}$$

The solution $\alpha=0$ is stable in the vicinity of the resonance, but is not applicable there due to the unmet assumptions. The non-zero value of α applies only at some distance from the eigenfrequency, where the expression below the square root becomes positive. This later case, when substituted into Eq. (17), gives

$$u_0=-\frac{P\omega^2}{\omega^2-\omega_0^2}\sin(t\omega)-\frac{2}{\omega^2-\omega_0^2}\sqrt{\frac{\eta}{\nu}\left(\omega^2-\omega_0^2\right)^2-\frac{1}{2}P^2\omega^4}\,\cos\left(t\omega_0\right). \tag{21}$$

The other option, $\alpha = 0$, would nullify the coefficient $\sin(\omega_0 t)$ in Eq. (21). This way the quasiperiodic character of u_0 will appear only when the non-zero α attains the real value, i.e., for $\eta/\nu > \Omega_p^2$. Due to different assumptions used in the multiple scales method, this condition does not correspond exactly to Δ_s defined above, however, except for a factor of $2^{-1/2}$, it captures Δ_0 defined in Eq. (6).

The correction term u_1 would involve elimination of more secular terms originating from sub-/super-harmonic cases when $\omega \approx {}^1\!/_3\,\omega_0, {}^1\!/_2\,\omega_0, 2\omega_0, 3\omega_0$, etc., and their combinations; this is, however, out of scope of the current work.

3. Conclusions

A simple van der Pol deterministic system with a harmonic right-hand side was studied in the vicinity of the resonance. In addition to the previously reported results, the boundaries of the lock-in region due to the primary resonance were derived using the harmonic balance method. A limited complementary analysis based on the multiple scales method was also presented. It turns out that the weakness of the harmonic balance method is its link to the specific frequency that is assumed in the solution. In this respect is the multiple scales method more flexible because it allows more resonant frequencies to be identified in the solution. On the other hand, the use of the multiple scales method is limited to the assumption of small nonlinearity, which can be limiting in some cases. In both approaches, a more detailed analysis will have to be adopted in order to qualitatively assess the fine details of the quasiperiodic processes surrounding the resonance region.

Acknowledgement

The support of CSF project 21-32122J and RVO 68378297 support are acknowledged.

References

[1] Afzali, F., Kharazmi, E., and Feeny, B.F.: Resonances of a forced van der Pol equation with parametric damping. In: *NODYCON Conference Proceedings Series*, pp. 477–487. Springer International Publishing, 2021.

[2] Gardner, W. and Franks, L.: Characterization of cyclostationary random signal processes. IEEE Transactions on Information Theory **21** (1975), 4–14.

[3] Nayfeh, A.H.: *Perturbation Methods*. Wiley, 2000.

[4] Nayfeh, A.H. and Mook, D.T.: *Nonlinear Oscillations*. Wiley, 1995.

[5] Náprstek, J. and Fischer, C.: Analysis of the quasiperiodic response of a generalized van der Pol nonlinear system in the resonance zone. Computers & Structures **207** (2018), 59–74.

[6] Náprstek, J. and Fischer, C.: Super and sub-harmonic synchronization in generalized van der Pol oscillator. Computers & Structures **224** (2019), 106 103.

Programs and Algorithms of Numerical Mathematics 21
J. Chleboun, P. Kůs, J. Papež, M. Rozložník, K. Segeth, J. Šístek (Eds.)
Institute of Mathematics CAS, Prague 2023

RESIDUAL NORM BEHAVIOR FOR HYBRID LSQR REGULARIZATION

Eva Havelková, Iveta Hnětynková

Faculty of Mathematics and Physics, Charles University
Department of Numerical Mathematics, Sokolovská 366/84, Prague, Czech Republic
eva.havelkova@karlin.mff.cuni.cz, iveta.hnetynkova@karlin.mff.cuni.cz

Abstract: Hybrid LSQR represents a powerful method for regularization of large-scale discrete inverse problems, where ill-conditioning of the model matrix and ill-posedness of the problem make the solutions seriously sensitive to the unknown noise in the data. Hybrid LSQR combines the iterative Golub-Kahan bidiagonalization with the Tikhonov regularization of the projected problem. While the behavior of the residual norm for the pure LSQR is well understood and can be used to construct a stopping criterion, this is not the case for the hybrid method. Here we analyze the behavior of norms of approximate solutions and the corresponding residuals in Hybrid LSQR with respect to the Tikhonov regularization parameter. This helps to understand convergence properties of the hybrid approach. Numerical experiments demonstrate the results in finite precision arithmetic.

Keywords: inverse problem, noise, Hybrid LSQR, Tikhonov regularization

MSC: 15A29, 65F22, 65F50

1. Introduction

We are concerned with an ill-posed inverse linear approximation problem

$$Ax \approx b, \quad A \in \mathbb{R}^{m\times n},\ b \in \mathbb{R}^m, \tag{1}$$

where $m \geq n$, $m, n \in \mathbb{N}$. The matrix A represents a (possibly large-scale) discretized smoothing operator, b stands for the data typically polluted by unknown additive noise e. Formally,

$$b = b^{\text{exact}} + e,$$

where b^{exact} denotes noise-free data. Further, denote $\eta = \|e\|/\|b\|$ the noise level, with $\|.\|$ being the standard Euclidean norm. Problems of the form (1) arise in many applications such as medical imaging or gravity surveying, see for example [4, 6]. Since the approximate solution is here seriously sensitive to the noise in b, regularization needs to be applied in order to obtain a meaningful solution. A wide variety

DOI: 10.21136/panm.2022.07

of regularization techniques have been developed, where for large-scale problems, iterative schemas are often the methods of choice. Here regularization is achieved by early termination of the process. Determining a reliable stopping criteria is crucial, because iterative methods applied on (1) typically exhibit semiconvergence. Alternatively, iterations can be further combined with direct regularization yielding the so-called hybrid methods such as Hybrid LSQR or Hybrid GMRES, see [1] for an overview. Hybrid methods are known for their ability to stabilize the computation and making it less sensitive to stopping criteria. Analysis of the properties of hybrid methods is, however, significantly more complicated.

In this paper we focus on Hybrid LSQR combining iterative projection on a Krylov subspace with the Tikhonov regularization of the projected small problem. We analyze residual norm behavior, since its stagnation indicates stabilization of the method and is thus used in stopping criteria when solving ill-posed problems. While properties of the residuals for standard LSQR regularization have already been analyzed (see, e.g, [2, 8]), this is not the case for Hybrid LSQR, where the behavior is highly dependent on the inner Tikhonov regularization parameter λ_k that changes in each outer iterative step k. Note that some analysis of LSQR combined with Tikhonov regularization for constant λ_k was provided already in [10]. A variety of parameter-choice methods have been introduced for selecting λ_k, e.g., the Discrepancy principle, L-curve or Generalized Cross Validation, see [4, 9, 12]. Their suitability for hybrid framework was studied in [3]. Here we, however, do not restrict ourselves to a particular parameter-choice strategy. We provide conditions on parameters λ_k to guarantee a decrease of the residual norm in hybrid LSQR and discuss its meaning in regularization process. Throughout the paper we assume exact arithmetic. Numerical experiments then demonstrate the presented properties in finite precision arithmetic.

2. Krylov projection and Tikhonov regularization

Hybrid LSQR represents a combination of the well known Golub-Kahan iterative bidiagonalization [10, 11] with the Tikhonov regularization. The Golub-Kahan bidiagonalization starting with $s_1 = b/\|b\|$ produces after k iterations the matrices W_k and S_{k+1}, having orthogonal basis of $\mathcal{K}_k(A^TA, A^Tb)$ and $\mathcal{K}_k(AA^T, b)$ in their columns, respectively. Assuming that the algorithm does not stop early, bidiagonalization coefficients $\alpha_i > 0,\ \beta_i >$ are stored in a lower bidiagonal matrix L_k,

$$L_k = \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_k & \alpha_k \end{bmatrix} \in \mathbb{R}^{k\times k}, \quad \text{and we denote} \quad L_{k+} = \begin{bmatrix} L_k \\ e_k^T\beta_{k+1} \end{bmatrix} \in \mathbb{R}^{(k+1)\times k},$$

where e_k is the k-th Euclidean vector of an appropriate size. Then it holds that

$$AW_k = S_{k+1}L_{k+}, \qquad A^TS_k = W_kL_k^T. \tag{2}$$

In the standard LSQR, the original problem (1) is replaced by the problem

$$\min_{y\in\mathbb{R}_k}\{\|AW_k y - b\|\}. \tag{3}$$

Using relations (2) and the orthogonality of S_k, we have

$$\|AW_k y - b\| = \|S_{k+1}L_{k+}y - b\| = \left\|S_{k+1}^T S_{k+1}L_{k+}y - S_{k+1}^T b\right\| = \|L_{k+}y - \beta_1 e_1\| \tag{4}$$

for any $y \in \mathbb{R}^k$. The projected problem (3) thus translates to

$$\min_{y\in\mathbb{R}_k}\{\|L_{k+}y - \beta_1 e_1\|\}, \quad \text{where} \quad \beta_1 = \|b\|,$$

having a unique solution y_k.

For inverse problems, however, the projected problem subsequently inherits their ill-posedness and noise gradually propagates to the projections, see [7]. Thus, Hybrid LSQR further applies Tikhonov regularization on the projected problem and solves

$$\min_{y\in\mathbb{R}^k}\{\|L_{k+}y - \beta_1 e_1\|^2 + \lambda_k^2 \|y\|^2\}, \tag{5}$$

for some regularization parameter $\lambda_k > 0$, $\lambda_k \in \mathbb{R}$. The obtained minimization problem has also a unique solution, further denoted $\overline{y}_k$. Putting the initial approximation $x_0 = 0$, the approximate solution to the original problem (1) is then obtained by

$$x_k = W_k y_k \quad \text{and} \quad \overline{x}_k = W_k \overline{y}_k \tag{6}$$

for LSQR and Hybrid LSQR, respectively.

Let us further clarify some notation. Denote the residuals corresponding to LSQR and Hybrid LSQR in the iteration k as follows

$$\begin{aligned} r_k(x) &= b - Ax, & p_k(y) &= \beta_1 e_1 - L_{k+}y, \\ \bar{r}_k(x) &= \begin{pmatrix} b \\ 0 \end{pmatrix} - \begin{pmatrix} A \\ \lambda_k I \end{pmatrix} x, & \bar{p}_k(y) &= \begin{pmatrix} \beta_1 e_1 \\ 0 \end{pmatrix} - \begin{pmatrix} L_{k+} \\ \lambda_k I \end{pmatrix} y, \end{aligned}$$

where for each k we have $x = W_k y,\ y \in \mathbb{R}^k$. We deliberately include the index k in the notation of the residuals for clarity when discussing their properties throughout iterations. Using (4), we get

$$\|r_k(x)\| = \|p_k(y)\|, \quad \|\overline{r}_k(x)\| = \|\overline{p}_k(y)\| \tag{7}$$

for any $x = W_k y,\ y \in \mathbb{R}^k$. Moreover, clearly it holds

$$\begin{aligned} \|\overline{r}_k(x)\|^2 &= \|r_k(x)\|^2 + \lambda_k^2 \|x\|^2, \\ \|\overline{p}_k(y)\|^2 &= \|p_k(y)\|^2 + \lambda_k^2 \|y\|^2. \end{aligned} \tag{8}$$

2.1. Interchangeability of projection and regularization

The above presented Hybrid LSQR applies the so called first project then regularize approach. It is well known that for selected hybrid methods this is equivalent to the first regularize then project approach, see [4, Chap. 6], even though the meaning of the equivalency is for various methods slightly different. Here, we briefly explain why for Hybrid LSQR the two approaches are fully interchangeable. The important consequence of this relationship is that many properties of LSQR hold also for Hybrid LSQR with a constant λ_k.

The first regularize then project approach starts with an application of the Tikhonov regularization to the original problem, schematically

$$\min_x\{\|Ax-b\|\} \to \min_x \left\{ \left\| \begin{pmatrix} b \\ 0 \end{pmatrix} - \begin{pmatrix} A \\ \lambda I \end{pmatrix} x \right\| \right\}.$$

Subsequently, k iterations of the Golub-Kahan bidiagonalization are computed for the extented problem above yielding the projected problem

$$\min_{y\in\mathbb{R}^k} \left\{ \left\| \begin{pmatrix} b \\ 0 \end{pmatrix} - \begin{pmatrix} A \\ \lambda I \end{pmatrix} \overline{W}_k y \right\| \right\}, \tag{9}$$

where $\overline{W}_k$ is an orthogonal basis of $\mathcal{K}_k\left(\begin{pmatrix} A \\ \lambda I \end{pmatrix}^T \begin{pmatrix} A \\ \lambda I \end{pmatrix}, \begin{pmatrix} A \\ \lambda I \end{pmatrix}^T \begin{pmatrix} b \\ 0 \end{pmatrix}\right)$. This shows the main disadvantage of the first regularize strategy - the parameter λ must be selected apriori based on the large problem (1). However, the obtained minimization (9) is clearly equivalent to

$$\min_{y\in\mathbb{R}_k}\{\left\|A\overline{W}_k y - b\right\|^2 + \lambda^2 \|y\|^2\}, \tag{10}$$

thanks to the orthogonality of $\overline{W}_k$. It remains to show that $\overline{W}_k = W_k$. From a simple multiplication and application of shift invariance of Krylov subspaces it follows that

$$\mathcal{K}_k\left(\begin{pmatrix} A \\ \lambda I \end{pmatrix}^T \begin{pmatrix} A \\ \lambda I \end{pmatrix}, \begin{pmatrix} A \\ \lambda I \end{pmatrix}^T \begin{pmatrix} b \\ 0 \end{pmatrix}\right) = \mathcal{K}_k(A^TA, A^Tb).$$

Thus, the first column of orthogonal matrices W_k and $\overline{W}_k$ is the same and their first i columns span the same subspace for any admissible i. It follows from the sequential form of the bidiagonalization process that in such a case $W_k = \overline{W}_k$. Using (4), the minimization problem (10) (first regularize then project approach) is equivalent to

$$\min_{y\in\mathbb{R}^k}\{\|L_{k+}y - \beta_1 e_1\|^2 + \lambda^2 \|y\|^2\}.$$

Consequently, provided λ is the same, the minimization problem is identical to the one in Hybrid LSQR (5) (first project then regularize approach).

Some further relations can be derived. Similarly to (2) we have

$$\begin{pmatrix} A \\ \lambda I \end{pmatrix} W_k = \overline{S}_{k+1}\overline{L}_{k+}, \quad \begin{pmatrix} A \\ \lambda I \end{pmatrix}^T \overline{S}_k = W_k \overline{L}_k^T,$$

and therefore similarly to (4) we obtain

$$\left\| \begin{pmatrix} A \\ \lambda I \end{pmatrix} - \begin{pmatrix} b \\ 0 \end{pmatrix} W_k y \right\| = \left\| \overline{S}_{k+1} \overline{L}_{k+} y - b \right\| = \left\| \overline{L}_{k+} y - \beta_1 e_1 \right\|.$$

The Hybrid LSQR minimization (5) can be thus equivalently written as

$$\min_{y \in \mathbb{R}_k} \{ \left\| \overline{L}_{k+} y - \beta_1 e_1 \right\| \}, \tag{11}$$

where $\overline{L}_{k+}$ has the same properties as L_{k+}, but its entries depend on λ (unlike for L_{k+}). Clearly, for $\lambda = 0$ it holds that $L_{k+} = \overline{L}_{k+}$.

3. Behavior of residual and solution norms

Recall that we assume $x_0 = 0$. It is well known [11] that then for LSQR the norm of the solution is strictly increasing,

$$\|x_{k+1}\| > \|x_k\|,$$

and the corresponding residual norm is strictly decreasing,

$$\|r_{k+1}(x_{k+1})\| < \|r_k(x_k)\|.$$

In combination with (6), (7) and the orthogonality of W_k the same holds for the projected problem, i.e.,

$$\|y_{k+1}\| > \|y_k\|,$$
$$\|p_{k+1}(y_{k+1})\| < \|p_k(y_k)\|.$$

Assume for a moment a constant regularization parameter, i.e., $\lambda_k = \lambda$ for all iterations k. It follows from the equivalency between project then regularize and regularize then project strategy (see Section 2), that the above described properties of LSQR hold also for Hybrid LSQR. Specifically,

$$\|\overline{x}_{k+1}\| > \|\overline{x}_k\|, \tag{12}$$
$$\|\overline{r}_{k+1}(\overline{x}_{k+1})\| < \|\overline{r}_k(\overline{x}_k)\|, \tag{13}$$

and similarly for the residual and solution of the projected problem

$$\left\|\overline{y}_{k+1}\right\| > \|\overline{y}_k\|, \tag{14}$$
$$\left\|\overline{p}_{k+1}(\overline{y}_{k+1})\right\| < \|\overline{p}_k(\overline{y}_k)\|. \tag{15}$$

It is useful to recall some properties of the Tikhonov regularization. Consider the minimization problem (5) for some fixed k. The corresponding solution can be expressed as a function of the regularization parameter λ_k as $\overline{y}_k(\lambda_k)$. Then

$$\|\overline{y}_k(\lambda_k)\| \text{ is decreasing with increasing } \lambda_k, \tag{16}$$
$$\|\overline{p}_k(\overline{y}_k(\lambda_k))\| \text{ is increasing with increasing } \lambda_k. \tag{17}$$

For the proof using the SVD decomposition see, e.g., [4].

3.1. Hybrid LSQR residual monotonicity

Hybrid LSQR minimizes the residual norm (8) which consists of two terms, the solution norm and the data fidelity term $p_k(\overline{y}_k)$. Thus (unlike LSQR) Hybrid LSQR does not minimize the residual corresponding to the original problem (1). Furthermore, the residual norm can generally oscillate and then it is hard to design a reliable stopping criterion for the iterations. For large-scale problems direct computation of $\|r_k(\overline{x}_k)\|$ may be infeasible. Thus we study the projected residual norm $\|p_k(\overline{y}_k)\|$ and then take advantage of (7). Stabilization of the inner residual norm can be used as a marker of stabilization of the method and for setting appropriate stopping criteria. The behavior of $\|p_k(\overline{y}_k)\|$ for Hybrid LSQR is highly dependent on the choice of the regularization parameter λ_k which is often chosen heuristically. We now investigate the behavior of $\|p_k(\overline{y}_k)\|$ with respect to the choice of λ_k.

Let us start with the case of constant regularization parameter, i.e., $\lambda_k = \lambda$.

Lemma 1. *Let $\overline{y}_k$ be the solution of (5) with $\lambda_k = \lambda,\ k = 1, 2, \ldots$. Then it holds*

$$\left\|p_{k+1}(\overline{y}_{k+1})\right\| < \|p_k(\overline{y}_k)\|\,.$$

Proof. Combining together (15) and (8) yields

$$\left\|p_{k+1}(\overline{y}_{k+1})\right\|^2 + \lambda^2 \left\|\overline{y}_{k+1}\right\|^2 < \|p_k(\overline{y}_k)\|^2 + \lambda^2 \|\overline{y}_k\|^2\,,$$

Using (14) then gives the result. □

A straightforward corollary of Lemma 1 and the property of Tikhonov regularization (17) is that

$$\lambda_{k+1} \le \lambda_k \quad \Rightarrow \quad \left\|p_{k+1}(\overline{y}_{k+1})\right\| < \|p_k(\overline{y}_k)\|\,.$$

In other words, if the value of λ_k is non-increasing, for Hybrid LSQR both $\|\overline{p}_k(\overline{y}_k)\|$ and $\|p_k(\overline{y}_k)\|$ are strictly decreasing (and thus also $\|\overline{r}_k(\overline{x}_k)\|$ and $\|r_k(\overline{x}_k)\|$). Moreover, it follows from (14) and (16) that

$$\lambda_{k+1} \le \lambda_k \quad \Rightarrow \quad \left\|\overline{y}_{k+1}\right\| > \|\overline{y}_k\|\,. \tag{18}$$

In practice, however, the regularization parameter λ_k is typically increasing rather then decreasing, because stronger regularization is needed with increasing k as noise subsequently propagates to the projected problem. In the paper [6], we have shown that

$$\left\|\overline{y}_{k+1}\right\| = \|\overline{y}_k\| \quad \Rightarrow \quad \left\|p_{k+1}(\overline{y}_{k+1})\right\| \le \|p_k(\overline{y}_k)\|\,.$$

Thus, also $\|r_{k+1}(\overline{x}_{k+1})\| \le \|r_k(\overline{x}_k)\|$. In words, stabilization of the inner solution norm is a sufficient condition for the residual norm to be nonincreasing. Theorem 2 generalizes this and states our main result.

Theorem 2. *Let λ_k, λ_{k+1} be such that the solutions $\overline{y}_k$, $\overline{y}_{k+1}$ of (5) satisfy $\|\overline{y}_k\| \leq \left\|\overline{y}_{k+1}\right\|$. Then*

$$\left\|p_{k+1}(\overline{y}_{k+1})\right\| \leq \|p_k(\overline{y}_k)\|\,, \quad k = 1,\, 2,\, 3,\, \ldots\,.$$

Given $\overline{x}_k = W_k\overline{y}_k$, it also holds that

$$\|r_{k+1}(\overline{x}_{k+1})\| \leq \|r_k(\overline{x}_k)\| \quad k = 1,\, 2,\, 3,\, \ldots\,.$$

Proof. Denote $y^*_{k+1} = [\overline{y}_k^T, 0]^T$. Then directly $\left\|y^*_{k+1}\right\| = \|\overline{y}_k\|$ and

$$\left\|p_{k+1}(y^*_{k+1})\right\| = \left\|L_{(k+1)+}y^*_{k+1} - \beta_1 e_1\right\| = \left\|L_{(k)+}\overline{y}_k - \beta_1 e_1\right\| = \|p_k(\overline{y}_k)\|\,.$$

Since $\overline{y}_{k+1}$ is a minimizer of (5), we obtain

$$\left\|p_{k+1}(\overline{y}_{k+1})\right\|^2 + \lambda_{k+1}^2\left\|\overline{y}_{k+1}\right\|^2 \leq \left\|p_{k+1}(y^*_{k+1})\right\|^2 + \lambda_{k+1}^2\left\|y^*_{k+1}\right\|^2 = \|p_k(\overline{y}_k)\| + \lambda_{k+1}^2\left\|\overline{y}_k\right\|^2.$$

Because $\left\|\overline{y}_{k+1}\right\|^2 \geq \|\overline{y}_k\|^2$, we get

$$\|p_{k+1}(y_{k+1})\|^2 \leq \|p_k(y_k)\|^2$$

and thus also $\|r_{k+1}(x_{k+1})\|^2 \leq \|r_k(x_k)\|^2$, see (7). □

Let us discuss how to satisfy the condition in Theorem 2. Clearly, by setting $\lambda_{k+1} = \lambda_k = 0$ we obtain standard LSQR for which the solution norm is increasing. Generally, it is possible to select the regularization parameter λ_{k+1} such that $\|\overline{y}_k\| = \left\|\overline{y}_{k+1}\right\|$, which also satisfies the condition. In such a case, the value of λ_k must be increasing. This holds because if λ_k was non-increasing, $\|\overline{y}_k\|$ would be increasing, see (18). In summary, Theorem 2 states that in order to maintain the residual norm $\|r_k(\overline{x}_k)\|$ decreasing, λ_k can be increasing but not too much. Provided monotonicity of the residual norm can then simplify the detection of stabilization of the regularization process. It is also important to note that the assumption in Theorem 2 is sufficient but not necessary.

4. Numerical experiments

Now we illustrate the above presented behavior in finite precision arithmetic. We consider two standard benchmark discrete ill-posed problems from the Regularization toolbox in MATLAB. For simplicity, a fixed number of iterations k is computed. The 1D problem *gravity* with $A \in \mathbb{R}^{50\times 50}$ and the noise level $\eta = 10^{-3}$ is solved in 30 iterations. For the 2D problem *blur* with $A \in \mathbb{R}^{2500\times 2500}$, $\eta = 10^{-1}$ and the Gaussian blur parameter $\sigma = 1$, we compute 50 iterations. The parameter λ_k for the Tikhonov regularization is chosen from 1000 samples logarithmically distributed in the interval $(0.0001, 10)$. We use the L-curve criterion [4, Chap. 5] for the *gravity* problem and the prescribed norm criterion [6] for the *blur* problem.

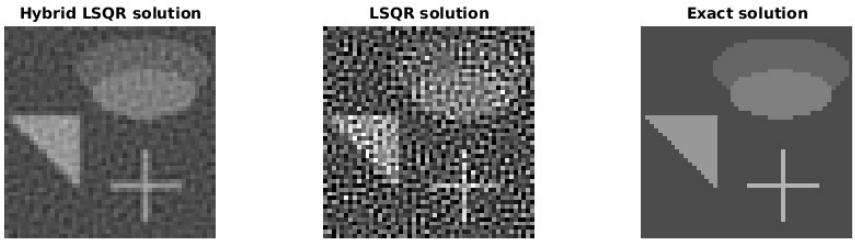


Figure 1: Comparison of approximate solutions of *blur* computed by Hybrid (left) and pure (middle) LSQR in 50 iterations. Hybrid method clearly provides a better reconstruction of the exact solution (right).

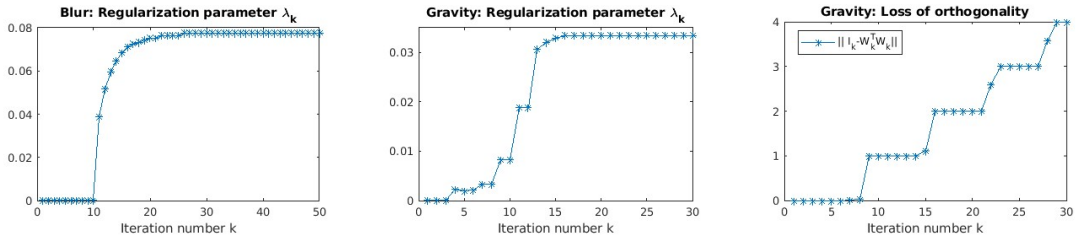


Figure 2: Regularization parameters λ_k computed for the two studied problems (left and middle). As expected, λ_k is mostly increasing. The right image illustrates significant loss of orthogonality among the columns of W_k for the *gravity* problem.

The effect of the inner regularization on improvement of the approximation is illustrated in Figure 1 comparing Hybrid LSQR and LSQR approximations for the problem *blur*. Figure 2 (left) then shows the corresponding regularization parameters λ_k determined during Hybrid LSQR. As expected, λ_k is non-decreasing and in latter iterations it stabilizes. The middle figure shows analogous behavior for the *gravity* problem. The effect is present despite the serious loos of orthogonality between the constructed bidiagonalization vectors, see the figure on the right. Figure 3 provides norms of the computed approximate solutions and the corresponding residuals for both testing problems. Their behavior corresponds nicely to the presented theory. If the solution norm increases, the residual norm is decreasing. A detailed view given in figures on the right for *gravity* shows, that from iterations 12 to 13 and 14 to 15 the solution norm decreases. Even though the assumption of Theorem 2 does not hold here, the corresponding residual norm still decreases from iteration 12 to 13 (but increases from iteration 14 to 15). This illustrates that the assumption is sufficient but not necessary. Note also how small the discrepancy is between the inner and outer residual and solution norms in finite precision, despite the severe loss of orthogonality. This property of Hybrid methods is explained in details in [5, Chap. 5] and [6]. If necessary, several re-orthogonalization strategies can be applied to improve orthogonality of the computed bidiagonalization vectors. For comparison, we illustrate the behavior when full re-orthogonalization (against all previous vectors)

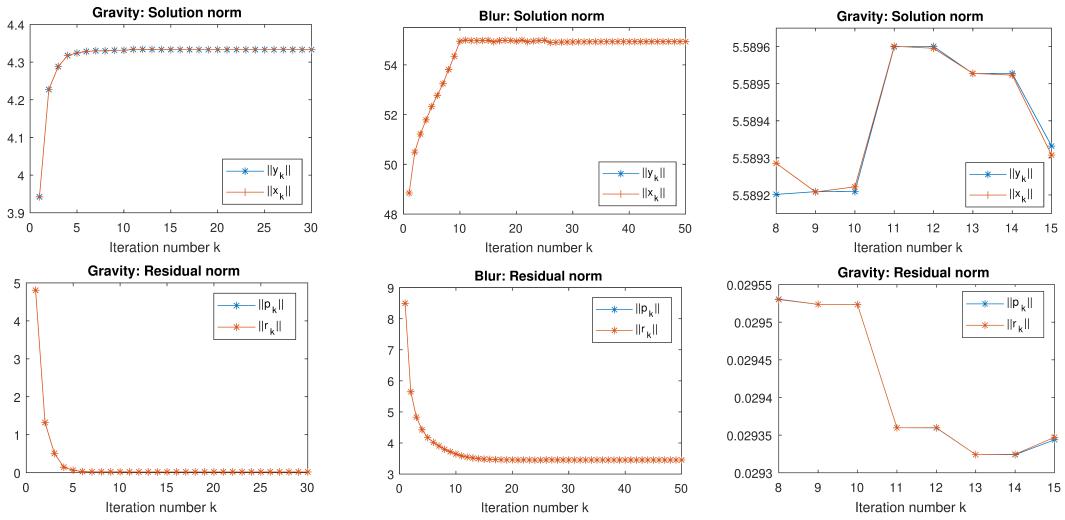


Figure 3: Behavior of the norm of the computed solutions and the corresponding residuals for the two studied problems. The right images show in detail several iterations for the *gravity* problem.

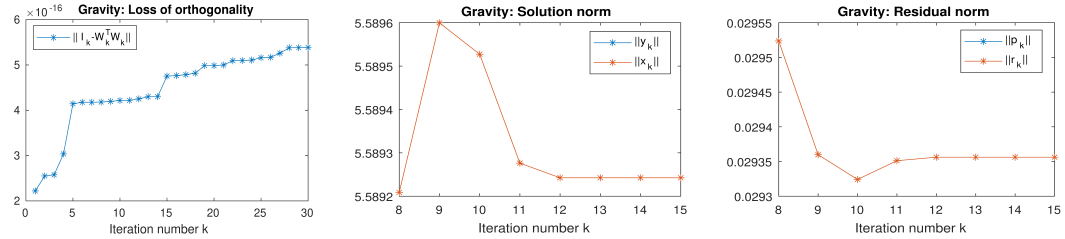


Figure 4: Illustration of the effect of re-orthogonalization. Orthogonality among the columns of W_k is at the machine precision (left). The discrepancy between the inner and outer norms is negligible (middle and right). Compare to Figure 3.

is applied on both sets W_k and S_k, see Figure 4. The orthogonality between the columns of W_k is at the machine precision (left figure). Consequently, the inner and outer solution norms match and the residual norms behave similarly as we observed in computations without re-orthogonalization.

Acknowledgements

This work was supported by the Charles University, project GAUK 376121 and by the grant SVV-2020-260583.

References

[1] Chung, J. and Gazzola, S.: Computational methods for large-scale inverse problems: a survey on hybrid projection methods, 2021.

[2] Gazzola, S. and Novati, P.: Inheritance of the discrete Picard condition in Krylov subspace methods. BIT Numerical Mathematics **56** (2016), 893–918.

[3] Gazzola, S., Novati, P., and Russo, M.R.: On Krylov projection methods and Tikhonov regularization. Electron. Trans. Numer. Anal. **44** (2015), 83–123.

[4] Hansen, P.C.: *Discrete Inverse Problems: Insight and Algorithms.* Fundamentals of Algorithms, Society for Industrial and Applied Mathematics, 2010.

[5] Havelková, E.: *Regularization methods for discrete inverse problems in Single Particle Analysis.* Master's thesis, Charles University, Prague, 2019.

[6] Havelková, E. and Hnětynková, I.: Iterative hybrid regularization for extremely noisy full models in Single Particle Analysis. Linear Algebra and its Applications **656** (2023), 131–157.

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

[8] Jia, Z.: Some properties of LSQR for large sparse linear least squares problems. Journal of Systems Science and Complexity **23** (2010), 815–821.

[9] Kilmer, M. and O'Leary., D.: Choosing Regularization Parameters in Iterative Methods for Ill-Posed Problems. SIAM Journal on Matrix Analysis and Applications **22** (2001), 1204–1221.

[10] Paige, C.C. and Saunders, M.A.: Algorithm 583: LSQR: Sparse linear equations and least squares problems. ACM Trans. Math. Softw. **8** (1982), 195–209.

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

[12] Renaut, R., Hnětynková, I., and Mead, J.: Regularization parameter estimation for large-scale Tikhonov regularization using a priori information. Computational Statistics and Data Analysis **54** (2010), 3430–3445.

Programs and Algorithms of Numerical Mathematics 21
J. Chleboun, P. Kůs, J. Papež, M. Rozložník, K. Segeth, J. Šístek (Eds.)
Institute of Mathematics CAS, Prague 2023

DGM FOR REAL OPTIONS VALUATION: OPTIONS TO CHANGE OPERATING SCALE

Jiří Hozman[1], Tomáš Tichý[2]

[1] Technical University of Liberec, Faculty of Science, Humanities and Education
Studentská 2, 461 17, Liberec, Czech Republic
jiri.hozman@tul.cz
[2] VSB — Technical University of Ostrava, Faculty of Economics
Sokolská třída 33, 702 00, Ostrava, Czech Republic
tomas.tichy@vsb.cz

Abstract: The real options approach interprets a flexibility value, embedded in a project, as an option premium. The object of interest is to valuate real options to change operating scale, typical for natural resources industry. The evolution of the project as well as option prices is decribed by partial differential equations of the Black-Scholes type, linked through a payoff function given by a type of the flexibility provided. The governing equations are discretized by the discontinuous Galerkin method over a finite element mesh and they are integrated in temporal variable by an implicit Euler scheme. The special attention is paid to the treatment of early exercise feature that is handled by additional penalty term. The capabilities of the approach presented are documented on the selected individual real options from the reference experiments using real market data.

Keywords: real option, option pricing, American option, partial differential equation, discontinuous Galerkin method, penalty method

MSC: 65M60, 35Q91, 91G60

1. Introduction

The real options approach plays an important role in the decision making process, because it provides a solution to the optimal investment decision that captures the flexibility value embedded in a project. As a result, this methodology enables to recognize the important qualitative and quantitative characteristic of some of the intrinsic attributes of the investment opportunities, namely, irreversibility of investments, choice of timing and last but not least uncertainty of the future rewards from investments, see [3]. The foundations of this modern investment theory were laid more than four decades ago by linking valuation of investment opportunities as pricing of financial options on real assets, see the pioneering paper by Myers [12]. Due to

DOI: 10.21136/panm.2022.08

the analogy with an option on financial asset, the methodology has become known as real options approach that interprets the flexibility value as the option premium. Since then, a large number of various solution techniques have been developed, from a simulation approach, over dynamic programming to contingent claims analysis, see [11] for a brief overview.

In this contribution we deal with real options valuation arising in natural resources industry, especially options to change operating scale. Following a contingent claim analysis [3] the values of both the project and the embedded flexibility, expressed as functions of time and underlying output price (following a stochastic process), can be identified as solutions of relevant partial differential equations (PDEs) of the Black-Scholes type. More precisely, the link between project and flexibility values is realized through a payoff function, which is enforced with respect to the flexibility type at any time prior to or at expiration date. Taking into account our recent results on pricing of conventional financial options, see, e.g., [5] and [6], a discontinuous Galerkin method (DGM) with an implicit time stepping scheme is applied to solve the relevant governing equations and to improve the numerical pricing valuation as a whole.

The concept of the paper is based on the contributions in proceedings [7] and [8], where options to expand and options to contract were studied in a separate way. The aim is to provide readers the methodological insight to real options pricing issues, documented on simplified case studies. First, the relevant PDE models are formulated, describing a value of the project as well as the option as the solution of the terminal-boundary value problem. Next, a numerical valuation scheme is presented. Finally, a simple numerical experiment, arising from an iron ore mining industry and related to reference data [9] and [10], is provided.

2. PDE models

Consider a one-stage investment project to change (i.e., expand or contract) the production of some output commodity. More precisely, such an investment project has an embedded option to expand the production rate or an embedded option to contract the production rate, exercisable any time prior to or at prespecified time $T > 0$ and requiring the additional implementation cost $\mathcal{K} > 0$. In terms of conventional financial options, the situation is described by a call option (on expansion) or a put option (on contraction) under American exercise right with strike $\mathcal{K}$ and maturity date T.

Next, we recall valuation models from [9] and [10] to price the embedded option as well as the project itself. We assume that project/option values can be expressed as functions of the actual time t and the output commodity price P following a geometric Brownian motion (proposed in [2]):

$$\mathrm{d}P(t) = (r - \delta)P(t)\mathrm{d}t + \sigma P(t)\mathrm{d}W(t), \quad P(0) > 0, \tag{1}$$

where $r > 0$ is the risk-free interest rate, $\delta > 0$ is the mean convenience yield on

holding one unit of the commodity, $W(t)$ is a standard Brownian motion and $\sigma > 0$ is the volatility of the commodity price.

Further, we denote by $V_0(P,t)$ the value of the project, which does not have any options to change operating scale. In contrast, the function $V_1(P,t)$ stands for the value of an investment project with the embedded option to expand (or contract) the production rate. Let $T^* > T$ be the maximum lifetime of both projects and $\varphi_0(P,t)$ and $\varphi_1(P,t)$ represent (after-tax) cash flow rates associated with the given project. Intuitively, from the definitions above we expect that

$$V_1(P,T^*) = V_0(P,T^*) = 0, \quad P \geq 0, \tag{2}$$

$$\varphi_1(P,t) = \varphi_0(P,t), \quad P \geq 0,\ t \in [0,T). \tag{3}$$

Following [1] one can characterize value functions V_0 and V_1 between expiry date T and project lifetime T^* as solutions of a couple of deterministic backward PDEs:

$$\frac{\partial V_i}{\partial t} + \underbrace{\frac{1}{2}\sigma^2 P^2 \frac{\partial^2 V_i}{\partial P^2} + (r-\delta)P\frac{\partial V_i}{\partial P} - rV_i}_{\mathcal{L}_{\mathrm{BS}}(V_i)} = -\varphi_i, \tag{4}$$

for $P \in (0,\infty)$, $t \in [T,T^*)$ with the terminal conditions (2).

In what follows we present the governing equation for the embedded flexibility representing the value added to the project function, i.e., $V_1(P,t) \geq V_0(P,t)$ for all $P \geq 0$ and $t \in [0,T)$. More precisely, we set $F(P,t) = V_1(P,t) - V_0(P,t)$ as the option value at the current price P and actual time $t \in [0,T)$. In view of the notation above, it is possible to track values of both projects and the embedded option value simultaneously within one timeline on $[0,T)$, that are linked at the expiry date T through the function

$$\Pi(P) \equiv \max(V_1(P,T) - V_0(P,T) - \mathcal{K}, 0) = \Pi(V_1(P,T), V_0(P,T)), \qquad P \geq 0, \tag{5}$$

which plays the role equivalent to a payoff function with strike $\mathcal{K}$, well-known from financial options pricing.

Further, taking into account an equivalence of cash flow rates (3) and encompassing the early exercise constraint of American options, i.e.,

$$F(P,t) \geq \Pi(V_1(P,T), V_0(P,T)), \quad P \geq 0,\ t \in [0,T), \tag{6}$$

the value function F satisfies the so-called moving-boundary problem, where it is also necessary to determine exercise and continuation regions separated by a free boundary driven by the optimal exercise price $P^*(t)$, see [13].

There are several approaches how to handle the early exercise feature, among the widely used ones, just penalty techniques [14] allow us to reformulate moving-boundary problem as follows

$$\frac{\partial F}{\partial t} + \mathcal{L}_{\mathrm{BS}}(F) + q_F = 0, \quad P \in (0,\infty),\ t \in [0,T), \tag{7}$$

where an additional nonlinear source term q_F is defined to ensure American constraint (6) and satisfy the conditions:

$$q_F(P,t) = 0, \text{ if } F(P,t) > \Pi(P), \quad q_F(P,t) > 0, \text{ if } F(P,t) = \Pi(P). \tag{8}$$

Note that the penalty approach can be unified for both European and American exercise features, if we put $q_F(P,t) = 0$ in (7) for all $P > 0$ and $t \in [0,T)$ in the case of a European exercise right.

3. DG approach

In order to determine the present value of flexibility to expand/contract the production rate, it is necessary to proceed in backward induction, from a pair of project value functions V_0 and V_1, over a construction of the payoff function Π, to the real option value function F. Since there are no analytical formulae for finite maturity American options in general, the valuation should rely on numerical approaches. The proposed valuation methodology is based on DGM, successfully used in the field of financial option pricing, see, e.g., [5] and [6].

At first, we localize the governing equations to a bounded interval $\Omega = (0, P_{\max})$, where maximal commodity price satisfies $P_{\max} > P^*(t)$ for all $t \in [0,T)$. Then, we have to impose project as well as option values at both endpoints $P = 0$ and $P = P_{\max}$. The project values are estimated by the net present value approach for the given cash flow rates as follows

$$V_i(z,t) = \int_t^{T^*} \varphi_i(z,\xi) e^{-r(\xi - t)} \mathrm{d}\xi, \quad z \in \{0, P_{\max}\}, t \in [T, T^*), \ i = 0, 1. \tag{9}$$

The real option value has to reflect the type of flexibility that this option provides. In accordance with the European exercise right, we prescribe a couple of Dirichlet boundary conditions in the form

$$\begin{aligned} &F(0,t) = 0, \quad F(P_{\max},t) = e^{-r(T-t)}\Pi(P_{\max}), && \text{(expansion)} \\ &F(0,t) = e^{-r(T-t)}\Pi(0), \quad F(P_{\max},t) = 0, \ \ t \in [0,T). && \text{(contraction)} \end{aligned} \tag{10}$$

Moreover, in the case of American options, boundary conditions (10) have to be set in the accordance with the early exercise feature which leads to the elimination of the discounted factor $e^{-r(T-t)}$ in (10).

Secondly, to handle the American early exercise feature and force the solution of (7) not to fall below its payoff function at any time $t \in [0,T)$, we introduce (as in [6]), for a sufficiently regular function v, the variational form of penalty term q_F as

$$(q_F(t), v) = c_{\mathrm{p}} \int_\Omega \chi_{\mathrm{exe}}(t) \Big(\Pi(P) - F(P,t) \Big) v \, \mathrm{d}P, \tag{11}$$

where $(\cdot,\cdot)$ denotes in fact the inner product in $L^2(\Omega)$. The function $\chi_{\mathrm{exe}}(t)$ in (11) is defined as an indicator function of the exercise region at time instant t and $c_{\mathrm{p}} > 0$ represents a weight to enforce the early exercise.

The cornerstone of the method applied is to construct a numerical solution as a composition of piecewise polynomial, generally discontinuous, functions on a spatial mesh without any requirements on the continuity of the solution across the partition nodes. We introduce the finite dimensional space

$$S_h^p = \{v_h \in L^2(\Omega)\colon v_h|_{(P_l,P_{l+1})} \in P^p((P_l, P_{l+1})),\ 0 \le l < N\}, \tag{12}$$

defined over the partition $0 = P_0 < P_1 < \ldots < P_N = P_{\max}$ of the domain Ω with the assigned mesh size h. Similarly as in [6], we carried out the DG spatial semi-discretization and temporal time discretization using an implicit Euler scheme. As a result, we obtain a sequence of linear algebraic problems related to a time partition $T^* = t_0 > t_1 > \cdots > t_R = T > t_{R+1} > \cdots > t_M = 0$ with fixed time step $\tau = T^*/M$. Further, denote $u_{h,m}^{(i)} \in S_h^p$, $i = 0, 1$, the approximation of the corresponding project value functions V_i from (4) at time level $t_m \in [T, T^*]$, $m = 0, \ldots, R$. Similarly, we define the DG approximate solution of problem (7) as functions $w_h^m \approx F(\cdot, t_m)$, $t_m \in [0, T]$, $m = R, \ldots, M$. Starting from zero initial project values $u_{h,0}^{(0)}$ and $u_{h,0}^{(1)}$, the desired value of flexibility $w_h^M \approx F(\cdot, 0)$ is computed in the following three steps

$$\begin{aligned}\left(u_{h,m+1}^{(i)}, v_h\right) - \tau\mathcal{A}_h\left(u_{h,m+1}^{(i)}, v_h\right) = \left(u_{h,m}^{(i)}, v_h\right) - \tau\ell_h^{(i)}(v_h)(t_{m+1}) \\ +\tau\left(\varphi_i(t_{m+1}), v_h\right) \quad \forall\, v_h \in S_h^p,\ m = 0, 1, \ldots, R-1,\ i = 0, 1,\end{aligned} \tag{13}$$

$$\left(w_h^R, v_h\right) = \left(\Pi\left(u_{h,R}^{(1)}, u_{h,R}^{(0)}\right), v_h\right) \quad \forall\, v_h \in S_h^p, \tag{14}$$

$$\begin{aligned}\left(w_h^{m+1}, v_h\right) - \tau\mathcal{A}_h\left(w_h^{m+1}, v_h\right) + \tau\mathcal{Q}_h\left(w_h^{m+1}, v_h\right) = \left(w_h^m, v_h\right) \\ -\tau\ell_h(v_h)(t_{m+1}) + \tau q_h(v_h)(t_{m+1}) \quad \forall\, v_h \in S_h^p,\ m = R, \ldots, M-1,\end{aligned} \tag{15}$$

where the bilinear form $\mathcal{A}_h(\cdot,\cdot)$ stands for the discrete variant of the operator $\mathcal{L}_{\mathrm{BS}}$ from (4). The linear forms $\ell_h^{(i)}(\cdot)(t)$ and $\ell_h(\cdot)(t)$ are associated with boundary conditions (9) and (10), related to the particular project value V_i and the option value F, respectively. Further, the treatment of the American constraint leads to new forms $\mathcal{Q}_h(\cdot,\cdot)$ and $q_h(\cdot)(t)$ in scheme (15), defined as discrete variants of the bilinear and linear part of (11), respectively. For the detailed derivation of the above-mentioned forms we refer the interested reader to [5].

Moreover, for practical purpose, to evaluate forms $\mathcal{Q}_h$ and q_h we use

$$\chi_{\mathrm{exe}}(t_m)\big|_{[P_l,P_{l+1}]} \approx \widetilde{\chi_{\mathrm{exe}}}(t_m)\big|_{[P_l,P_{l+1}]} := \begin{cases} 1, & \text{if } w_h^{m-1}\left(P_{\mathrm{c}}^{l+1}\right) < w_h^R\left(P_{\mathrm{c}}^{l+1}\right) \\ 0, & \text{if } w_h^{m-1}\left(P_{\mathrm{c}}^{l+1}\right) \ge w_h^R\left(P_{\mathrm{c}}^{l+1}\right) \end{cases} \tag{16}$$

for $t_m \in [0, T)$, $0 \le l \le N-1$, where P_{c}^{l+1} is the midpoint of the interval $[P_l, P_{l+1}]$ and w_h^R is given as S_h^p-approximation of the payoff function Π depending on states $u_{h,R}^{(i)}$, $i = 0, 1$, see (14).

4. Numerical experiments

In this section, we briefly illustrate the usage of the DG approach on idealized case studies from the iron ore mining industry. The three-step valuation scheme (13)–(15) is implemented in the solver Freefem++, incorporating GMRES as a solver for non-symmetric sparse systems, for more details, see [4].

As in [9] and [10] we consider iron ore mine, having the value given by the function $V_0(P,t)$, that depends on commodity price P, expressed in USD per dry metric tonne (dmt) of iron ore. Further, we have a mining project of value $V_1(P,t)$, adopting the embedded option $F(P,t)$ to scale up (or down) the production rate any time $t \in [0,T]$. Let Q denote the total reserve of the iron ore mine (in thousands of million dmt) and $q_i(t) \geq 0$, $i = 0,1$, be the iron ore production rates (in thousands of million dmt per year) associated with projects V_i, $i = 0,1$. Depending on how the mine is operated, project lifetimes are defined as minimum admissible values T_0 and T_1 (in years) that satisfy the relationship

$$Q = \int_0^{T_0^*} q_0(\xi)\mathrm{d}\xi = \int_0^{T_1^*} q_1(\xi)\mathrm{d}\xi, \tag{17}$$

where

$$q_0(t) = \begin{cases} s(t), & \text{if } t \in [0, T_0^*), \\ 0, & \text{if } t \in [T_0^*, T^*], \end{cases} \quad q_1(t) = \begin{cases} s(t), & \text{if } t \in [0, T), \\ \kappa \cdot s(t), & \text{if } t \in [T, T_1^*), \\ 0, & \text{if } t \in [T_1^*, T^*], \end{cases} \tag{18}$$

for $s(t)$ corresponding to the production rate related to the project having no embedded options and factor $\kappa > 0$ representing the extracted ($\kappa > 1$) or contracted ($\kappa < 1$) mining production rate. Further, we define the after-tax cash flow rates of relevant projects as follows

$$\varphi_i(P,t) = q_i(t)\Big((1-D)P - c(t)\Big)(1-B), \qquad i = 0,1, \tag{19}$$

for $P \in [0, P_{\max}]$ and $t \in [0, T^*]$, where $c(t)$ is the average cash cost rate of iron ore production per dmt, D is the rate of state royalties and B is the income tax rate. The numerical experiments are performed on the following (reference) project and market data:

$$\begin{aligned} &Q = 10, \qquad s(t) = 0.1\, e^{0.007t}, \qquad D = 0.05, \qquad B = 0.3, \\ &c(t) = C_0 e^{0.005t}, C_0 > 0, \qquad r = 0.06, \qquad \delta = 0.02, \end{aligned} \tag{20}$$

which are the representatives of parameter values of practical significance.

4.1. European expansion option

Referring to [9] we price an expansion option exercisable only at maturity date $T = 2$ under discretization parameters $p = 2$, $P_{\max} = 100$, $h = 1$ and $\tau = 0.02$.

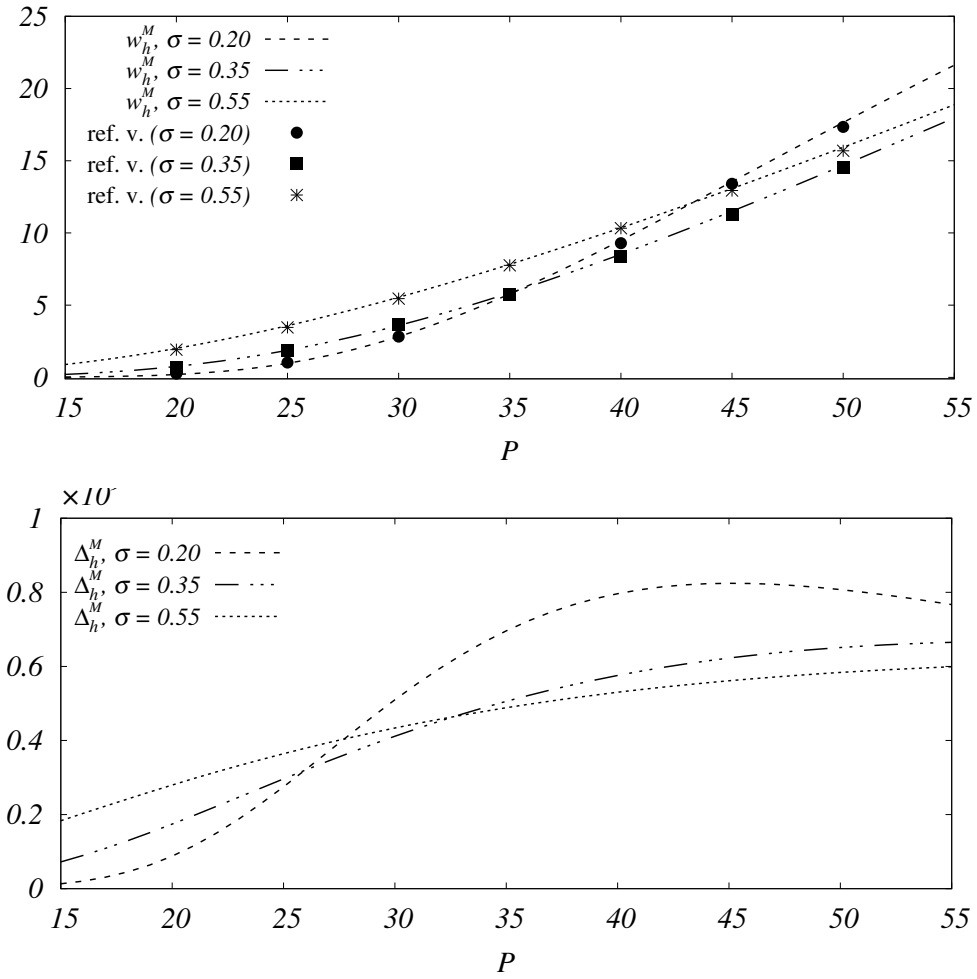


Figure 1: The approximate option values (in 10^9 USD) for different scenarios (top) and the corresponding Delta values (bottom).

Further, we take $C_0 = 35$ USD (based on prices from 2007) and the implementation cost to double production ($\kappa = 2$) is set as $\mathcal{K} = 10$, given in 10^9 USD. Using (17), (18) and (20), easy calculation leads to $T^* \doteq 75.8$ and $T_1^* \doteq 43.6$.

Consistent with the referenced experiment we investigate the behaviour of the option values for various values of volatility. Figure 1 (top) records flexibility values at present time ($t = 0$) for all scenarios considered. One can easily observe that plots are similar to the conventional financial European call options with the relevant Black-Scholes model parameters. Moreover, piecewise quadratic DG approximations match well the reference values (evaluated at underlying reference prices) and give fairly the same results as the upwind finite difference methods from [9]. More precisely, we can deduce that option values seem to be an increasing function

of volatility σ in the region of low commodity prices (i.e., for less than some critical value). On the other hand, in the case of high commodity prices, the situation is quite opposite and the most valuable option is the one with the smallest volatility ($\sigma = 0.2$). This intuitive expectation is well illustrated in Figure 1 (bottom), where the corresponding Delta sensitivity measures, $\Delta_h^M \approx \frac{\partial F}{\partial P}(\cdot, 0)$, are depicted. At first glance, the most sensitive flexibility value with respect to the commodity price is related to the low volatility scenario, because in this case the commodity price has little chance to fluctuate. From this point of view, we come to the same conclusions as in the paper [9].

4.2. American contraction option

Secondly, we price a contraction option exercisable any time prior to or at $T = 1$ under discretization parameters $p = 2$, $P_{\max} = 60$, $h = 0.6$, $\tau = 0.01$ with early exercise weight $c_{\mathrm{p}} = 10/\tau$. As in [10] we set $C_0 = 25$ USD (prices from 1988) and the implementation cost $\mathcal{K} = 1 - \kappa$ (given in 10^9 USD) and investigate the behaviour of the option values with the fixed volatility $\sigma = 0.3$ for various contraction factors under American as well as European exercise rights. The lifetime T_1^* is determined in a similar way as in preceding experiment for various κ. The approximate option values at present time for selected contraction factors are depicted in Figure 2 (top). Analogously to the previous experiment, plots are similar to the conventional financial put options and illustrate an intuitive expectation that the value of flexibility to contract F is a decreasing function of the factor κ in the region of low commodity prices. Moreover, it is apparent for all cases that American options cost more than their European counterparts, i.e., early exercise feature increases value of the project flexibility. This distinctive feature of American options is also well resolved by Delta measures in Figure 2 (bottom), i.e., $|\Delta_h^M(\mathrm{Am})| \geq |\Delta_h^M(\mathrm{Eu})|$ for a particular κ. Thus, these observations are in good agreement with the expectations of practitioners.

5. Conclusion

The real options approach and especially related valuation techniques pose a very challenging part of corporate finance. In this paper we have recalled PDE models to valuation of investment projects together with the embedded flexibility of a one-stage expansion or contraction of the production rate. The particular governing equations were solved by a numerical scheme based on DGM. The presented numerical experiments, arising from the iron ore mining industry, provides financially meaningful results and thus illustrates a suitability of DGM for real options pricing issues that take into account fluctuations in commodity prices as well as different expansion/contraction factors. One possible future research objective should be addressed to extend the DG approach to advanced combinations of options to change operating scale incorporated into a compound option that enables to properly capture changing investment strategies in a long time horizon.

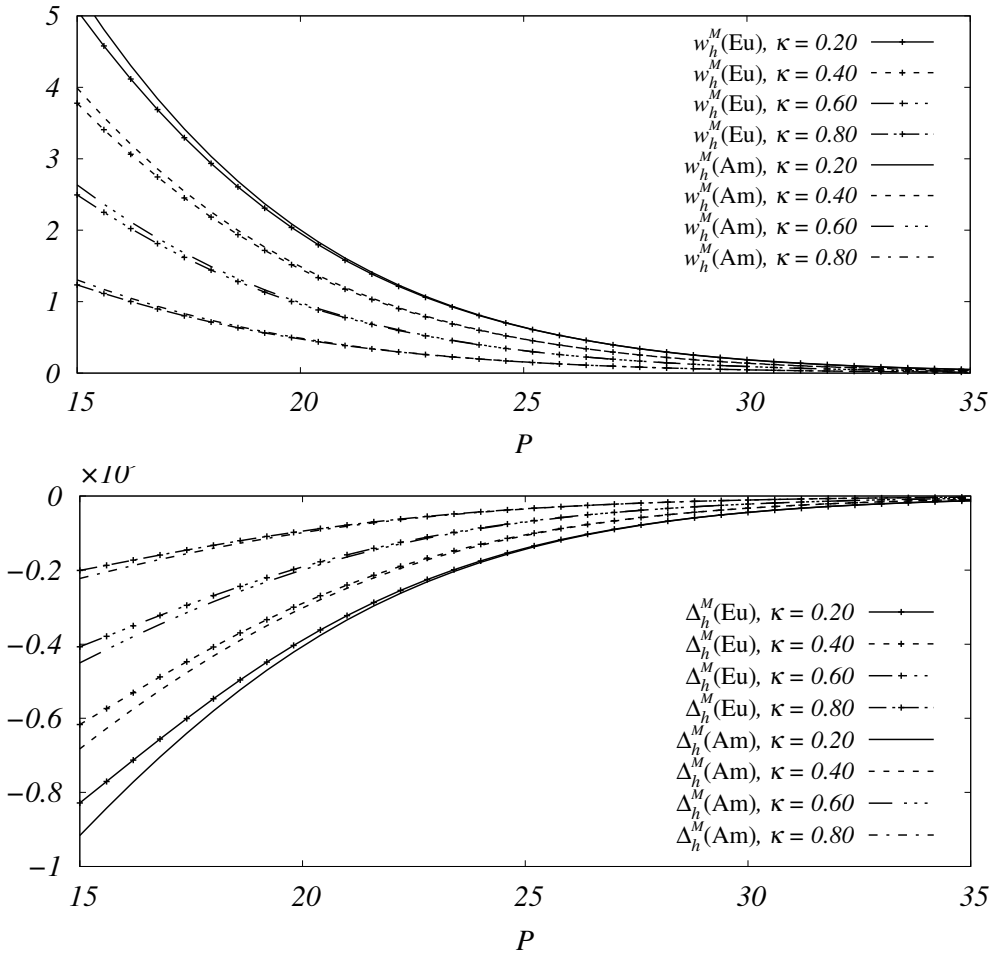


Figure 2: The approximate option values (in 10^9 USD) for different scenarios (top) and the corresponding Delta values (bottom) under European and American constraints.

Acknowledgements

Both authors were supported through the Czech Science Foundation (GAČR) under project 22-17028S. The support is greatly acknowledged. Furthermore, the second author also acknowledges the support provided within SP2022/4, an SGS research project of VSB-TU Ostrava.

References

[1] Black, F. and Scholes, M.: The pricing of options and corporate liabilities. J. Polit. Econ. **81** (1973), 637–659.

[2] Cortazar, G., Schwartz, E., and Casassus, J.: Optimal exploration investments under price and geological-technical uncertainty: a real options model. R&D Manage. **31** (2001), 181–189.

[3] Dixit, A. and Pindyck, R.: *Investment Under Uncertainty*. Princeton University Press, Princeton, 1994.

[4] Hecht, F.: New development in freefem++. J. Numer. Math. **20** (2012), 251–265.

[5] Hozman, J. and Tichý, T.: DG framework for pricing European options under one-factor stochastic volatility models. J. Comput. Appl. Math. **344** (2018), 585–600.

[6] Hozman, J. and Tichý, T.: The discontinuous Galerkin method for discretely observed Asian options. Math. Method. Appl. Sci. **43** (2020), 7726–7746.

[7] Hozman, J. and Tichý, T.: Numerical valuation of the investment project with expansion options based on the PDE approach. In: R. Hlavatý (Ed.), *Proceedings of the 39th International Conference Mathematical Methods in Economics*, pp. 185–190. Czech University of Life Sciences Prague, Prague, 2021.

[8] Hozman, J. and Tichý, T.: Numerical valuation of the investment project flexibility based on the PDE approach: An option to contract. In: H. Vojáčková (Ed.), *Proceedings of the 40th International Conference Mathematical Methods in Economics*, pp. 122–128. College of Polytechnics Jihlava, Jihlava, 2022.

[9] Li, N. and Wang, S.: Pricing options on investment project expansions under commodity price uncertainty. J. Ind. Manag. Optim. **15** (2019), 261–273.

[10] Li, N., Wang, S., and Zhang, S.: Pricing options on investment project contraction and ownership transfer using a finite volume scheme and an interior penalty method. J. Ind. Manag. Optim. **16** (2020), 1349–1368.

[11] Mun, J.: *Real Options Analysis: Tools and Techniques for Valuing Strategic Investments and Decisions*. Wiley, Hoboken, 2002.

[12] Myers, S.: Determinants of corporate borrowing. J. Financ. Econ. **5** (1977), 147–175.

[13] Wong, H. and Zhao, J.: An artificial boundary method for American option pricing under the CEV model. J. Comput. Appl. Math. **46** (2008), 2183–2209.

[14] Zvan, R., Forsyth, P., and Vetzal, K.: Penalty methods for american options with stochastic volatility. J. Comput. Appl. Math. **91** (1998), 199–218.

Programs and Algorithms of Numerical Mathematics 21
J. Chleboun, P. Kůs, J. Papež, M. Rozložník, K. Segeth, J. Šístek (Eds.)
Institute of Mathematics CAS, Prague 2023

VALIDATION OF NUMERICAL SIMULATIONS OF A SIMPLE IMMERSED BOUNDARY SOLVER FOR FLUID FLOW IN BRANCHING CHANNELS

Radka Keslerová[1], Anna Lancmanová[1,2], Tomáš Bodnár[1,2]

[1]Czech Technical University in Prague, Faculty of Mechanical Engineering,
Department of Technical Mathematics,
Karlovo nám. 13, Prague, 121 35, Czech Republic
[2]Institute of Mathematics, Czech Academy of Sciences,
Žitná 25, 115 67 Prague 1, Czech Republic.
Radka.Keslerova@fs.cvut.cz, Anna.Lancmanova@fs.cvut.cz,
Tomas.Bodnar@fs.cvut.cz

Abstract: This work deals with the flow of incompressible viscous fluids in a two-dimensional branching channel. Using the immersed boundary method, a new finite difference solver was developed to interpret the channel geometry. The numerical results obtained by this new solver are compared with the numerical simulations of the older finite volume method code and with the results obtained with OpenFOAM. The aim of this work is to verify whether the immersed boundary method is suitable for fluid flow in channels with more complex geometries with difficult grid generation.

Keywords: immersed boundary method, finite volume method, OpenFOAM

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

1. Introduction

Fluid flow in the system of branching channels is a part of many technical or biological applications, for example blood flow in the fine and complex branching of the cardiovascular system. This work is focused on the flow of blood in the venous system, for simplification it is possible to consider blood flow as flow of incompressible viscous fluid in branching channels.

The network of such a branching system can be imagined as a main channel followed by multilevel branching, and each of these new branches can have a different diameter and can be connected to the main channel at a different angle.

Complex formation of the channel system causes problems related to the description of the geometry, its mesh generation, and the mathematical formulation of the related problem, including appropriate boundary conditions. The description of the

DOI: 10.21136/panm.2022.09

channel geometry can be done using the standard grid generation inside the channel. A grid can be either structured or unstructured. This approach is quite common, but it is associated with certain disadvantages. This includes the difficulty of mesh generation and the need to re-generate the mesh in case of even small geometric modifications. Also, CFD solvers for general unstructured grids are more complex, making it difficult to implement any non-standard mathematical models or boundary conditions.

Some problems that arise when using classical methods on grids inside the area (limited by the channel boundary) can be avoided by adopting the immersed boundary method. In this case a larger area of space is discretized, e.g. the rectangle enclosing the tested branch channel. A grid (Cartesian grid) is constructed throughout such a domain, where model equations are also solved. The specific geometry of the channel is represented only at the level of the mathematical model used, one model in the region occupied by the fluid and another elsewhere. Switching between models is simply implemented using a characteristic function specifying the inner and outer parts of the considered channel. In this case, due to the very simple grid structure and domain shape, the CFD solver can be very simple. Any changes in the geometry of the channel are easily solved, it is only necessary to redefine the characteristic function describing the fluid region.

The aim of this work is to compare the results of a standard method based on finite volumes, which uses the grid built inside the channel, i.e. the grid bounded by the channel edges, with a much simpler finite difference code working on the regular Cartesian grid using a general implementation of the immersed boundary method. A simple straight channel with one branch inclined at different angles was chosen as the test case.

2. Mathematical model

The fundamental system of equations is the system of Navier–Stokes equations for incompressible Newtonian fluids. This system is based on the balance laws of mass and momentum for incompressible fluids

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

$$\rho \left(\frac{\partial \boldsymbol{u}}{\partial t} + \operatorname{div}(\boldsymbol{u} \otimes \boldsymbol{u}) \right) = -\nabla P + \mu \Delta \boldsymbol{u}, \tag{2}$$

where P is the pressure, ρ is the constant density, $\boldsymbol{u}$ is the velocity vector and μ represents the constant dynamic viscosity.

3. Numerical methods

The numerical methods which solve the system of incompressible Navier-Stokes equations can be divided according to velocity-pressure coupling stategy into two main groups, coupled methods (e.g. artificial compressibility and dual time-stepping

methods) and pressure correction methods (including e.g. SIMPLE or PISO algorithms). The SIMPLE algorithm [11, 12] is the main method used for the numerical solution of incompressible fluid flow problems (also due to its ability to treat unsteady flows). This algorithm is included in Open source Field Operation And Manipulation (OpenFOAM) and is described in detail in [7, 14].

The artificial compressibility method (designed to treat steady flows) was used in our in-house built FDM and FVM codes. This method [4, 6] is used to obtain 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. The parameter β for the steady case is chosen approximately equal to the maximum velocity in the domain.

3.1. Finite difference method

The finite difference approximation of the governing system of equations (1) and (2) is a natural choice because of the use of immersed boundary method on Cartesian grids. In such case the discretization is simple, allowing for easy implementation and modification of various numerical methods and algorithms.

The system including the modified (for artificial compressibility) continuity equation (1) and the momentum equation (2) can be written in vector form as [1]:

$$\boldsymbol{D}_\beta \mathbf{W}_t + \mathbf{F}_x + \mathbf{G}_y = \mathbf{R}_x + \mathbf{S}_y, \tag{3}$$

where $\boldsymbol{D}_\beta = \operatorname{diag}\left(\frac{1}{\rho\beta^2}, 1, 1\right)$, $\mathbf{W} = \operatorname{col}(p, u, v)$ is the vector of unknowns,

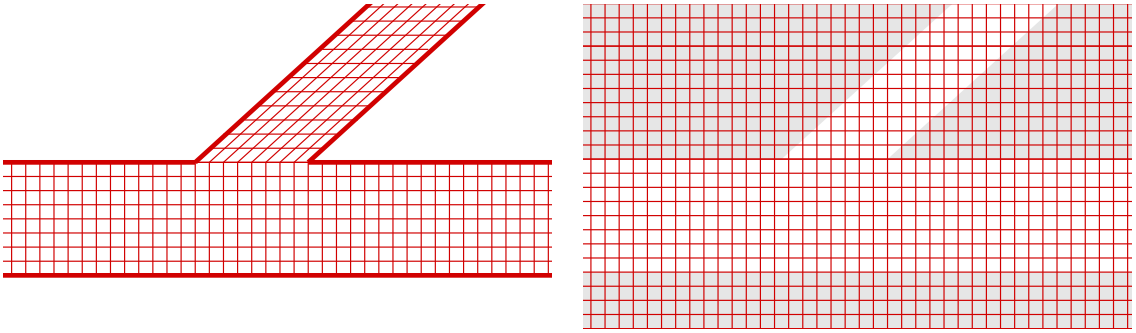
$$\mathbf{F} = \begin{pmatrix} u \\ u^2 + p \\ vu \end{pmatrix}, \quad \mathbf{G} = \begin{pmatrix} v \\ uv \\ v^2 + p \end{pmatrix}, \quad \mathbf{R} = \begin{pmatrix} 0 \\ \nu u_x \\ \nu v_x \end{pmatrix}, \quad \mathbf{S} = \begin{pmatrix} 0 \\ \nu u_y \\ \nu v_y \end{pmatrix} \tag{4}$$

where p is the kinematic pressure ($p = P/\rho$), u, v are velocity components and ν is the kinematic viscosity.

3.1.1. Immersed boundary method

In computational fluid dynamics, the immersed boundary method was first used in reference to the method developed by Charles Peskin in 1972 (see [13]) to simulate fluid-structure interactions.

A characteristic feature of this method is that the numerical simulation of fluid flow is performed on Cartesian grid that does not have to directly copy the geometry of the computational (fluid) domain, see e.g. [3, 10]. The situation can be described using the schematic sketch (shown in Fig. 1) of the grids used for Finite Volume Method (FVM) and Finite Difference Method (FDM) in this work. The structured grid used in FVM simulations has a simple structure with the grid lines fitted to boundaries of the computational domain. This results in grids that are aligned to boundaries. This situation is shown in Fig. 1 (a).



(a) Grid for FVM (b) Grid for FDM with the immersed geometry

Figure 1: Detail of the grid for finite-volume and finite-difference simulations.

In the immersed boundary FDM method, the governing system of equations is discretized in the whole rectangular domain and used boundary conditions are only imposed on its boundary. The unknown values of velocity and pressure are sought in all internal points of the domain, distinguishing the points inside of the fluid domain (marked by white color in Fig. 1 (b)) and inside of the solid domain (marked by gray color in Fig. 1 (b)). The velocity fields in the solid domain is set to zero, so that the governing equations are only solved in the points in the fluid region, see e.g. [3, 8].

3.1.2. MacCormack scheme

In computational fluid dynamics, the MacCormack method is a widely used discretization method for the numerical solution of hyperbolic partial differential equations. This second-order finite difference method was introduced by Robert W. MacCormack in 1969 [9]. It is the method written in the predictor-corrector form using asymmetric forward/backward discretization stencil to approximate spatial derivatives to provide finally a central (second order) approximation.

To describe the MacCormack scheme, rearrange equation (3) to the form where all terms except the time derivative are placed on the right hand side

$$\mathbf{W}_t = \boldsymbol{D}_\beta^{-1}\left[-\left(\mathbf{F}_x + \mathbf{G}_y\right) + \nu \boldsymbol{D}\Delta\mathbf{W}\right], \quad \nu\boldsymbol{D}\Delta\mathbf{W} = \mathbf{R}_x + \mathbf{S}_y. \tag{5}$$

To update in time the values of the vector $\mathbf{W}^n$ to $\mathbf{W}^{n+1}$ an approximation of $\mathbf{W}_t$ is constructed from (5). This approximation is built differently, asymmetrically, in predictor (e.g. with backward differences) and in corrector (by forward differences). The final update is performed using linear combination of the two values obtained. The expressions for predicted and corrected values are shown in (6) and (7).

$$\begin{aligned}\widetilde{\mathbf{W}}_{i,j} =& \mathbf{W}_{i,j}^n + \Delta t\, \boldsymbol{D}_\beta^{-1}\Bigg[- \frac{\mathbf{F}_{i,j}^n - \mathbf{F}_{i-1,j}^n}{\Delta x} - \frac{\mathbf{G}_{i,j}^n - \mathbf{G}_{i,j-1}^n}{\Delta y} \\ &+ \nu\, \boldsymbol{D}\left(\frac{\mathbf{W}_{i+1,j}^n - 2\mathbf{W}_{i,j}^n + \mathbf{W}_{i-1,j}^n}{\Delta x^2} + \frac{\mathbf{W}_{i,j+1}^n - 2\mathbf{W}_{i,j}^n + \mathbf{W}_{i,j-1}^n}{\Delta y^2}\right)\Bigg]\end{aligned} \tag{6}$$

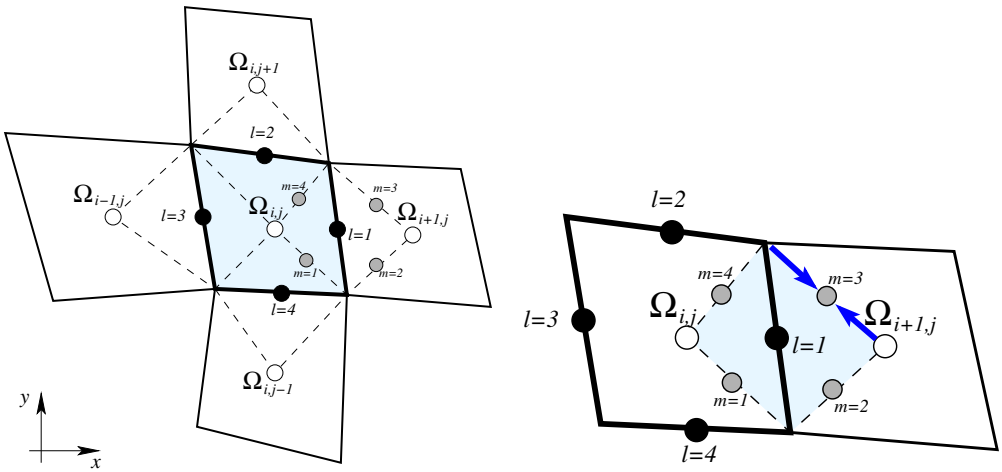
$$\mathbf{W}_{i,j}^{n+1} = \frac{1}{2}\left(\mathbf{W}_{i,j}^{n} + \widetilde{\mathbf{W}}_{i,j}\right) + \frac{\Delta t}{2}\,\boldsymbol{D}_{\beta}^{-1}\Bigg[- \frac{\widetilde{\mathbf{F}}_{i+1,j}^{n} - \widetilde{\mathbf{F}}_{i,j}^{n}}{\Delta x} - \frac{\widetilde{\mathbf{G}}_{i,j+1}^{n} - \widetilde{\mathbf{G}}_{i,j}^{n}}{\Delta y} \tag{7}$$

$$+ \nu\,\boldsymbol{D}\left(\frac{\widetilde{\mathbf{W}}_{i+1,j}^{n} - 2\widetilde{\mathbf{W}}_{i,j}^{n} + \widetilde{\mathbf{W}}_{i-1,j}^{n}}{\Delta x^2} + \frac{\widetilde{\mathbf{W}}_{i,j+1}^{n} - 2\widetilde{\mathbf{W}}_{i,j}^{n} + \widetilde{\mathbf{W}}_{i,j-1}^{n}}{\Delta y^2}\right)\Bigg]. \tag{8}$$

3.2. Finite volume method

In this work the finite volume discretization is used as a reference for comparison and validation of the newly developed finite-difference solver. Within the presented study, the finite volume method was used in two codes. First, in an in-house developed simple 2D code, and second, in OpenFOAM.

The spatial discretization is based on the cell-centered finite volume approximation on a multi-block structured grid. While the mesh is handled as block structured by the in-house solver, the same grid is treated as unstructured by OpenFOAM. The finite volumes are quadrilaterals in 2D. For the in-house code the central scheme is used for convective terms, including the pressure gradient calculated from the approximation. The viscous terms are also discretized in the central way on dual quadrilateral mesh (diamond type scheme), see Fig. 2.



(a) Primary & secondary cells (b) Approximation of viscous fluxes

Figure 2: Grid configuration for approximation of inviscid and viscous fluxes.

The resulting semi-discrete system of ODEs (based on (5)) is integrated in time by the explicit multistage Runge–Kutta scheme:

$$\begin{aligned} \mathbf{W}_{i,j}^{(0)} &= \mathbf{W}_{i,j}^{n} \\ \mathbf{W}_{i,j}^{(r+1)} &= \mathbf{W}_{i,j}^{(0)} - \alpha_{(r)}\Delta t\mathcal{L}\mathbf{W}_{i,j}^{(r)} \qquad r = 1,\ldots,s \\ \mathbf{W}_{i,j}^{n+1} &= \mathbf{W}_{i,j}^{(s)} \end{aligned} \tag{9}$$

The three-stage explicit Runge-Kutta scheme used to obtain results presented here had coefficients: $\alpha_{(1)} = 1/2$, $\alpha_{(2)} = 1/2$, $\alpha_{(3)} = 1$. More details on this type of finite volume discretization and associated Runge-Kutta methods can be found, e.g., in [1, 2, 5].

OpenFOAM uses a co-located grid, i.e., the fluid dynamic quantities are all stored at the control volumes centroids. The convective terms are discretized using the central difference scheme and also for the viscous fluxes the central differences are used. In this case, however, due to grid curvature an extra correction term (for non-orthogonality) is added to the discretization, subject to certain limiter, for more details see [11].

4. Numerical tests

The numerical results shown in this section are used to compare different numerical methods to verify that the newly developed immersed boundary method is sufficiently accurate. At the same time, the flow at the branching point of the channel was also tested depending on the connection angle of the secondary branch.

4.1. Domain geometry

For the immersed boundary implementation of finite-difference method, the $2D$ computational domain was chosen as a rectangle in $x - y$ plane with dimensions $30D \times 10D$. The numerical simulations were performed on the structured (Cartesian) grid with different number of equidistant nodes.

The used domain is shown in Fig. 3. The diameter of the main channel is denoted by symbol D and $D = 0.006\ m$ and the width of the branch inclined at the angle α was chosen to be $D/2$. The same configuration was kept for all simulations, just changing the angle α by setting it to values 30°, 60°, 90°, 120° and 150°. For finite volume simulations, just the interior of the channel (marked by white color in Fig. 3) was used to construct the grid.

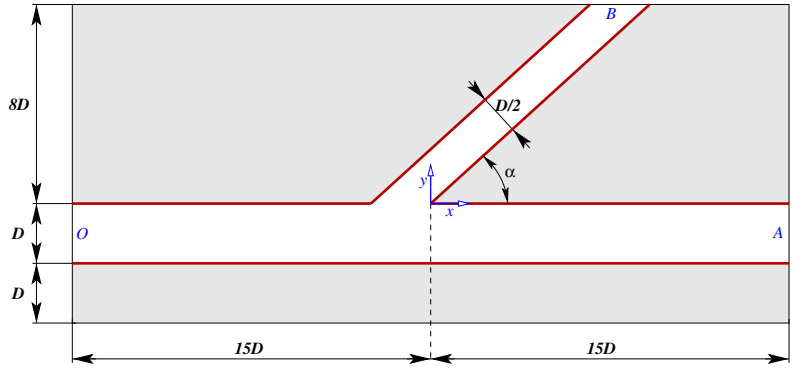


Figure 3: Computational domain of a planar branching channel.

4.2. Boundary conditions

Boundary conditions were chosen in such a way, that the flow is driven by the prescribed pressure drop between the inlet and outlet parts of the boundary. So only different values of pressure were prescribed at inlet ($p_{in} = 60\ Pa$) and in outlet parts ($p_A = p_B = 0\ Pa$) of the boundary. Otherwise the homogeneous Neumann condition was prescribed for velocity components on those parts of boundary to mimic a fully developed flow. On the channel wall of course the no-slip, i.e., homogeneous Dirichlet condition $\boldsymbol{u} = (0, 0)$ was prescribed for velocity.

4.3. Numerical results

The aim of presented numerical results is to demonstrate the applicability of the chosen methods and their settings for the considered class of problems. The newly developed FD method based immersed boundary code is compared with an in-house finite volume code (both using artificial compressibility approach) and the open-source OpenFOAM finite volume code (using a variant of SIMPLE algorithm). Both FVM codes share the same computational grid. For the FDM method with immersed boundary channel representation two different grids were used. The coarser grid had resolution 1200×200 cells, while the finer grid doubled the number of cells in the vertical y direction, i.e., having 1200×400 cells.

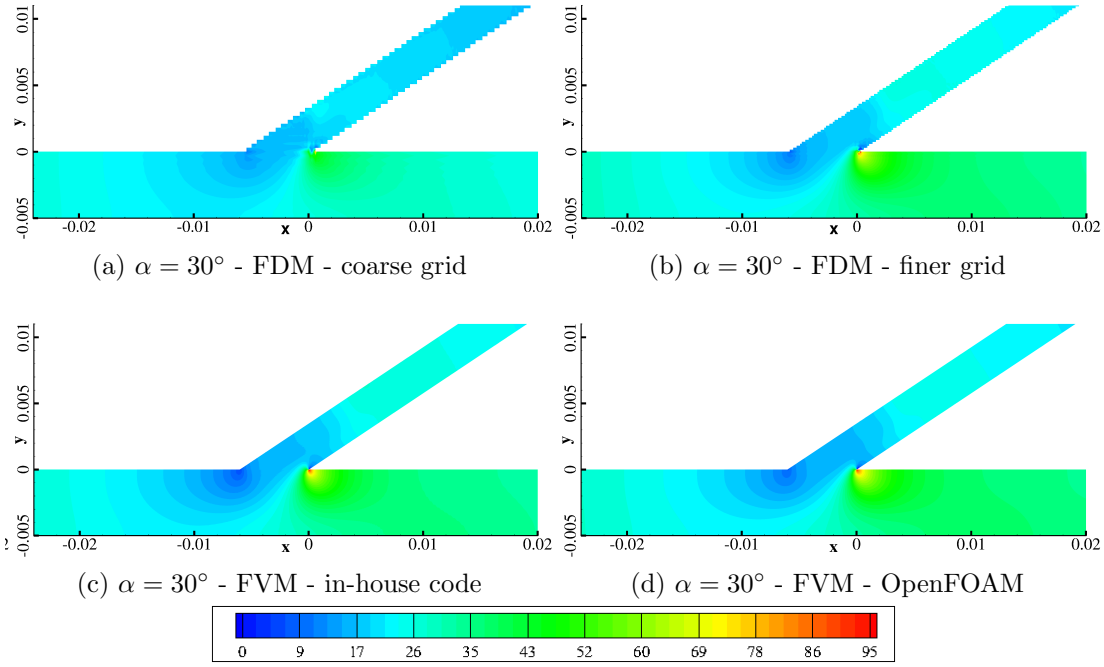
Figs. 4 and 5 show the comparison of pressure and velocity fields obtained using all the considered codes for the case of oblique branching at angle $\alpha = 30°$. The pressure fields in Fig. 4 have very similar character and except the FDM results on coarse grid all results are almost identical. The comparison of velocity magnitude fields in Fig. 5 reveals that the in-house FVM code and FDM code on the finer grid provide almost identical results. The OpenFOAM results predict a bit higher velocity in the main channel, while the FDM code on coarse grid predicts lower velocity.

It is interesting to see that the level of agreement between the results changes for different angles α of the secondary branch. The comparison of pressure and velocity fields in the case of $\alpha = 60°$ is shown in Figs. 6 and 7. Here it seems that the OpenFOAM results are closest to the FDM on the finer grid.

The comparison of results in the case of $\alpha = 90°$ (shown in Figs. 8 and 9) shows that even the results obtained by FDM on the coarse grid are almost identical to the other methods. The orthogonality of the grid allows for optimal use of all computational points and leads to highest accuracy of numerical approximation.

Similar results were obtained for the remaining two tested angles, $\alpha = 120°$ and $\alpha = 150°$ (not shown here). Also here the mutual agreement between the finer grid of immersed boundary method and the in-house code can be seen.

Similar comparison for cases with different branching angle α is shown in presented Figs. 4-9 for finite volume method (in-house code and OpenFOAM with SIMPLE algorithm) and finite difference method (coarse and finer grid). The mutual agreement between the finer grid for FDM and in-house FVM code depends on the angle α, with best results (smallest solution differences) achieved for angles close to



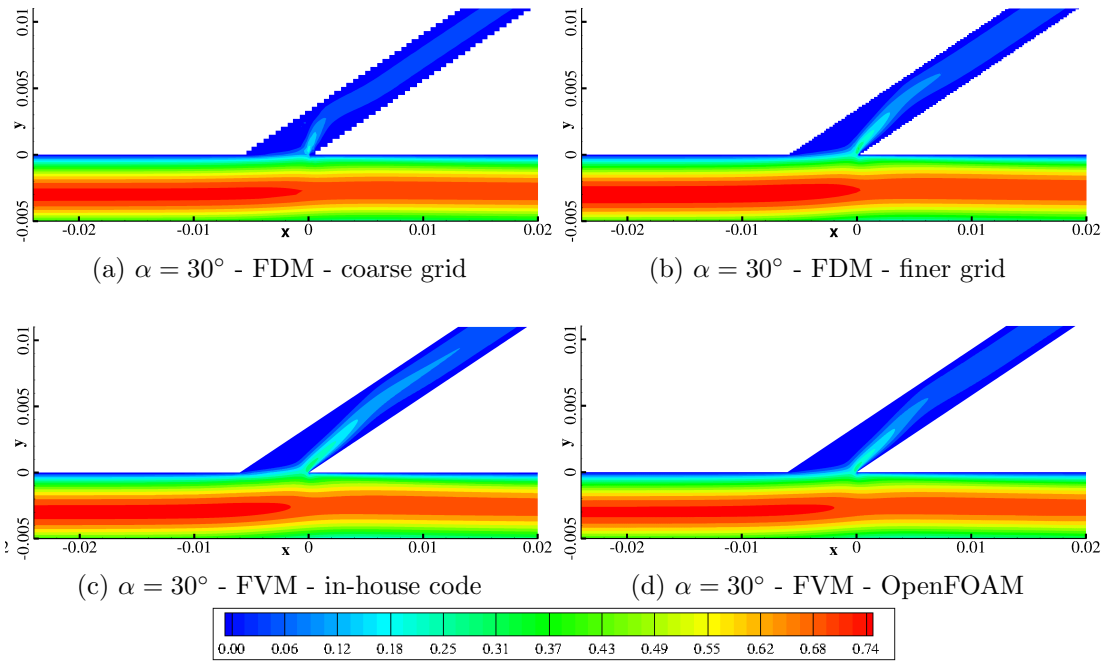
(a) $\alpha = 30°$ - FDM - coarse grid

(b) $\alpha = 30°$ - FDM - finer grid

(c) $\alpha = 30°$ - FVM - in-house code

(d) $\alpha = 30°$ - FVM - OpenFOAM

Figure 4: Pressure field in detail for the case $\alpha = 30°$, different solvers and grids.



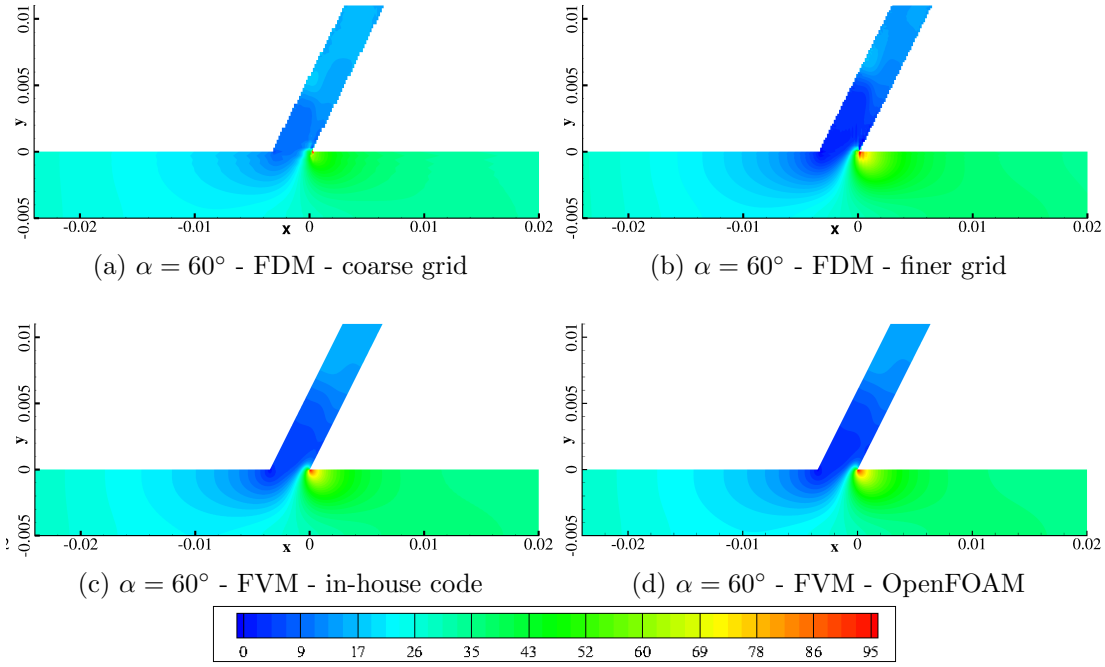
(a) $\alpha = 30°$ - FDM - coarse grid

(b) $\alpha = 30°$ - FDM - finer grid

(c) $\alpha = 30°$ - FVM - in-house code

(d) $\alpha = 30°$ - FVM - OpenFOAM

Figure 5: Velocity magnitude in detail for the case $\alpha = 30°$, different solvers and grids.



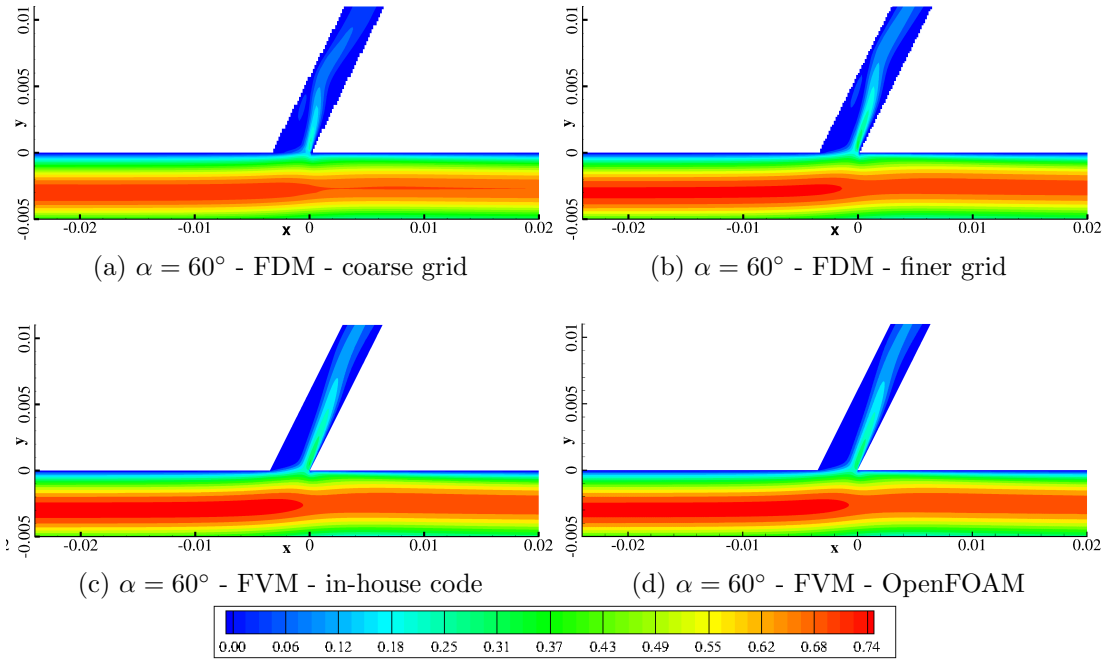
(a) $\alpha = 60°$ - FDM - coarse grid

(b) $\alpha = 60°$ - FDM - finer grid

(c) $\alpha = 60°$ - FVM - in-house code

(d) $\alpha = 60°$ - FVM - OpenFOAM

Figure 6: Pressure field in detail for the case $\alpha = 60°$, different solvers and grids.



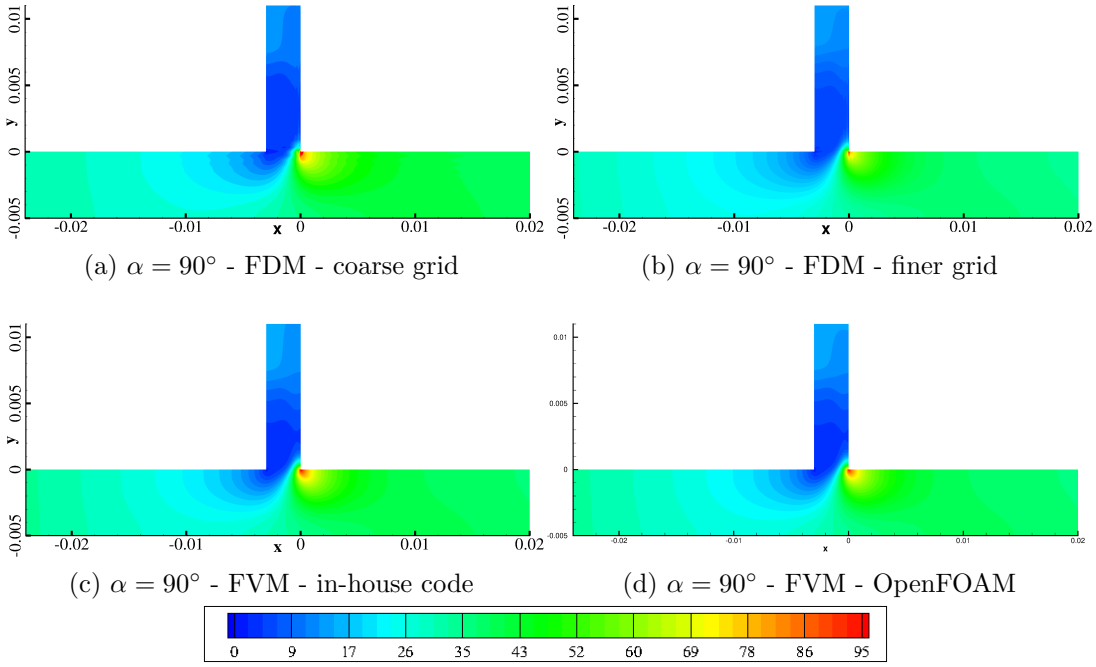
(a) $\alpha = 60°$ - FDM - coarse grid

(b) $\alpha = 60°$ - FDM - finer grid

(c) $\alpha = 60°$ - FVM - in-house code

(d) $\alpha = 60°$ - FVM - OpenFOAM

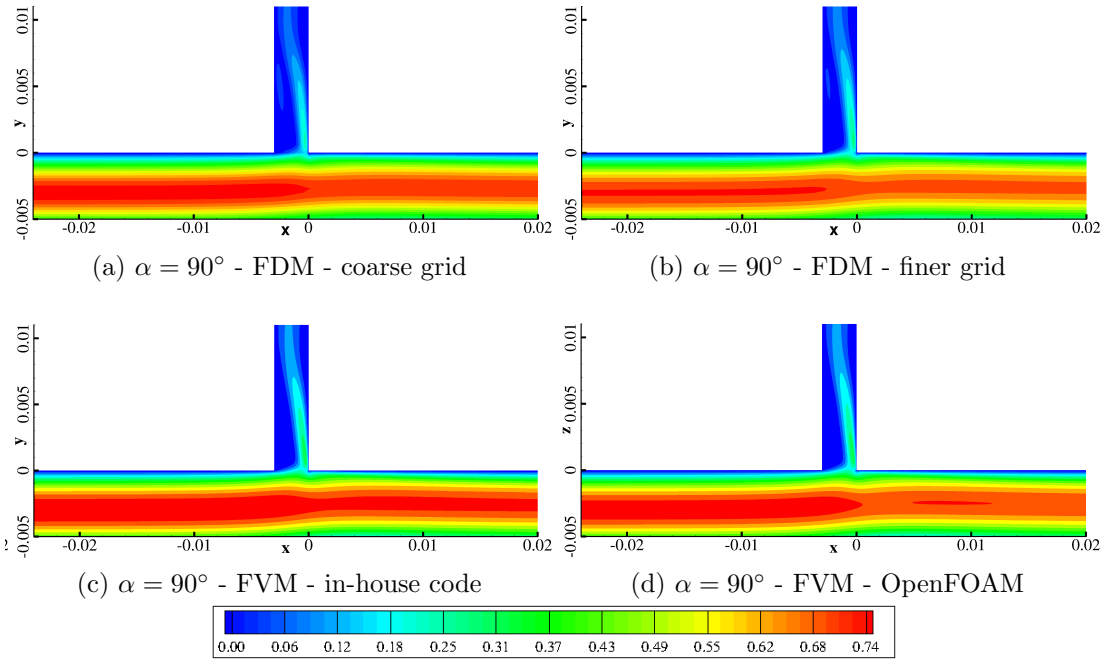
Figure 7: Velocity magnitude in detail for the case $\alpha = 60°$, different solvers and grids.



(a) $\alpha = 90°$ - FDM - coarse grid (b) $\alpha = 90°$ - FDM - finer grid

(c) $\alpha = 90°$ - FVM - in-house code (d) $\alpha = 90°$ - FVM - OpenFOAM

Figure 8: Pressure field in detail for the case $\alpha = 90°$, different solvers and grids.



(a) $\alpha = 90°$ - FDM - coarse grid (b) $\alpha = 90°$ - FDM - finer grid

(c) $\alpha = 90°$ - FVM - in-house code (d) $\alpha = 90°$ - FVM - OpenFOAM

Figure 9: Velocity magnitude in detail for the case $\alpha = 90°$, different solvers and grids.

$\alpha = 90°$, while in the case $\alpha = 30°$ (and $\alpha = 150°$, not shown here) the differences are more pronounced.

5. Conclusions

A new numerical code was developed based on a finite difference method using the immersed boundary approach, which was applied to the numerical simulation of the flow of viscous incompressible fluid in planar branching channels.

The numerical results were presented in this work showed that the results obtained by the newly developed code are comparable to the results provided by the previously used code based on the finite volume method and also to the results from the open-source package OpenFOAM.

The dependence of the immersed boundary method on the grid resolution was found, especially during numerical simulations in channels with oblique branching. In the case of a perpendicular connection, the differences between coarser and finer grids were not so large. Although the results obtained on the coarse and finer grids are qualitatively very similar (showing the same flow structure), some quantitative parameters (such as the maximum velocity or discharge) may differ.

In the presented comparison, a simple pressure-based setup was chosen, where the flow is controlled only by the prescribed pressure differences between the inlet/outlet boundaries of the channel branches. Such setup is very sensitive to the numerical method, the grid structure, and the way the boundary conditions are imposed. This sensitivity is due to the fact that the flow in the channel branches is unknown in advance, and the flow field develops only due to the pressure difference. In this context, the agreement between the numerical predictions of the three considered methods and codes can be evaluated as satisfactory.

Our future work will focus on the extension of the presented comparison for unsteady flows and non-Newtonian fluids, which is crucial for the intended investigation of various biomedical applications.

Acknowledgements

This work was supported by the grant SGS22/148/OHK2/3T/12 and partly by the *Praemium Academiae* of Š. Nečasová.

References

[1] Beneš L., Louda P., Keslerová R., Kozel K. Štigler J.: Numerical simulations of flow through channels with T-junction, Journal of Applied Mathematics and Computation **219** (2013) 7225–7235.

[2] Bodnár T., Sequeira A.: Numerical simulation of the coagulation dynamics of blood, Computational and Mathematical Methods in Medicine **9** (2008) 83–104.

[3] Bodnár, T., Keslerová, R., Lancmanová, A.: Numerical Investigation of Incompressible Fluid Flow in Planar Branching Channels *Recent Advances in Mechanics and Fluid-Structure Interaction with Applications*. Basel: Birkhäuser (2022) 95–126.

[4] Chorin, A. J.: A numerical method for solving incompressible viscous flow problem. Journal of Computational Physics **135** (1967) 118–125.

[5] Keslerová, R., Trdlička, D.: Numerical solution of viscous and viscoelastic fluids flow through the branching channel by finite volume scheme, Journal of Physics: Conference Series **633** (2015).

[6] Keslerová R., Trdlička D., Řezníček H.: Numerical simulation of steady and unsteady flow for generalized Newtonian fluids, Journal of Physics, Conference Series **738** (2016).

[7] Keslerová R., Řezníček H., Padělek T.: Numerical modelling of generalized Newtonian fluids in bypass tube, Advances in Computational Mathematics **45** (2019) 2047–2063.

[8] Lancmanová A., Bodnár T., Keslerová R.: Numerical Validation of a Simple Immersed Boundary Solver for Branching Channels Simulations, In: D. Šimurda and T. Bodnár (Eds.) *Proceedings Topical Problems of Fluid Mechanics 2022*, IT CAS, Prague 2022, 127–134.

[9] MacCormack R. W.: The effect of viscosity in hypervelocity impact cratering, AIAA (1969) 69–354.

[10] Mittal R., Iaccarino G.: Immersed boundary method, Annual Review of Fluid Mechanics **37** (2005) 239–261.

[11] Moukalled F., Mangani L., Darwish M.: *The Finite Volume Method in Computational Fluid Dynamics*, Springer, Heidelberg, 2016.

[12] Patankar S. V., Spalding D. B.: A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows, International Journal of Heat and Mass Transfer **15** (1972) 1787–1806.

[13] Peskin C. S.: The immersed boundary method, Acta Numerica (2002) 479–517.

[14] Winter O., Bodnár T.: Simulations of viscoelastic fluids flows using a modified log-conformation reformulation, In: D. Šimurda and T. Bodnár (Eds.) *Proceedings Topical Problems of Fluid Mechanics 2017*, IT CAS, Prague 2017, 321–328.

Programs and Algorithms of Numerical Mathematics 21
J. Chleboun, P. Kůs, J. Papež, M. Rozložník, K. Segeth, J. Šístek (Eds.)
Institute of Mathematics CAS, Prague 2023

FINDING VERTEX-DISJOINT CYCLE COVER OF UNDIRECTED GRAPH USING THE LEAST-SQUARES METHOD

Jan Lamač, Miloslav Vlasák

Faculty of Civil Engineering, Czech Technical University in Prague
Thákurova 7, 166 29 Prague 6, Czech Republic
jan.lamac@cvut.cz, miloslav.vlasak@cvut.cz

Abstract: We investigate the properties of the least-squares solution of the system of equations with a matrix being the incidence matrix of a given undirected connected graph G and we propose an algorithm that uses this solution for finding a vertex-disjoint cycle cover (2-factor) of the graph G.

Keywords: cycle cover, 2-factor, Hamiltonian cycle, incidence matrix, least-square method

MSC: 05C50, 05C38, 93E24

1. Introduction

Finding a vertex-disjoint cycle cover (called a 2-factor) of a given undirected graph G consists in finding a set of disjoint cycles which are subgraphs of G and contain all vertices of G (see Figure 1). It is well known that a 2-factor of an undirected 2-factorable graph can be found in polynomial time by finding a perfect matching in some larger graph (cf. [10]). When we prescribe further conditions (e.g. number of components, minimal cycle length) the problem of finding a 2-factor becomes NP-hard (cf. [4]). This includes a 2-factor formed by one component only, i.e. the Hamiltonian cycle of the graph G.

In this paper we investigate the properties of the least-squares solution of the system of equations with a matrix being the incidence matrix of the given undirected connected[1] graph G and propose an algorithm that uses this solution for finding a 2-factor of the graph G. In this algorithm we successively erase the edges from the graph until we obtain the desired 2-factor of the graph. For determination of which edge will be erased we employ and test three strategies: S1, S2 and S3.

DOI: 10.21136/panm.2022.10

[1] All the strategies considered can be easily extended to disconnected graphs.

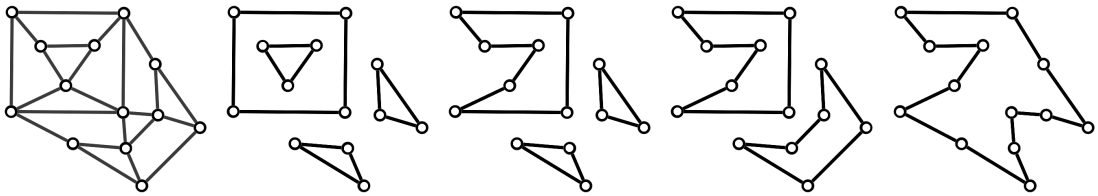


Figure 1: Graph G (left) and its vertex-disjoint cycle covers (2-factors). The last one is the Hamiltonian cycle of the graph G.

2. Graph, its representation and notation

By graph G we consider an ordered pair $G = (V, E)$, where

$$\begin{aligned} V &= V(G) = \{v_1, v_2, \ldots, v_n\} \\ &\text{is a set of vertices of graph } G \text{ and} \\ E &= E(G) = \{e_1, e_2, \ldots, e_m\} \subseteq \binom{V}{2}, \quad e_j = \{v_k, v_l\},\ k \neq l, \\ &\text{is a set of edges of the graph } G. \end{aligned}$$

We denote by $B \in \{0,1\}^{n\times m}$ the incidence matrix of G satisfying $B_{ij} = 1$ if $v_i \in e_j$ and $B_{ij} = 0$ if $v_i \notin e_j$. Arbitrary set of edges can be represented by the vector $x \in \{0,1\}^{m\times 1}$, which is a characteristic vector of the set $X \subseteq E$ satisfying $x_i = 1$ if $e_i \in X$ and $x_i = 0$ otherwise. If we want to refer to a particular edge $e \in X$ we also use a notation $[x]_e$ (instead of x_i). Further, we denote by $B_e \in \{0,1\}^{n\times(m-1)}$ the matrix obtained from B by deleting the column corresponding to the edge e. Similarly, we denote by x_e the vector that we obtain from x by deleting $[x]_e$. Finally, 1_k stands for a column vector formed by k ones.

Using this notation we may define the vertex-disjoint cycle cover x of the graph G being any set of edges satisfying

$$x \in \{0,1\}^{m\times 1} \quad \& \quad 1_m^T x = n \quad \& \quad Bx = 2 \cdot 1_n. \tag{1}$$

While the second condition ensures the cycle cover contains n edges, the third one guarantees that each vertex coincides with exactly 2 edges.

3. Basic properties of the vector x_{LS}

The least-square solution of the system $Bx = 2 \cdot 1_n$ is defined using the Moore–Penrose pseudo-inverse of the matrix B (see e.g. [8]) as follows

$$x_{LS} = B^\dagger(2 \cdot 1_n) = 2 \cdot B^\dagger 1_n. \tag{2}$$

In this section we investigate the properties of the vector x_{LS}.

Lemma 1. *Let the graph G be non-bipartite, then the least-square solution x_{LS} of the system of equations $Bx = 2 \cdot 1_n$ satisfies*

$$1_m^T x_{LS} \;=\; n. \tag{3}$$

For bipartite graph $G = (V_1 \cup V_2, E)$ with $|V_1| = n_1$ and $|V_2| = n_2$ there holds

$$1_m^T x_{LS} \;=\; \frac{4n_1 n_2}{n}. \tag{4}$$

Proof. Since the rows of the incidence matrix B are linearly independent for non-bipartite connected graphs (cf. [11]), the pseudo-inverse of the matrix B satisfies $B^\dagger = B^T(BB^T)^{-1}$ and, thus, the least-square solution x_{LS} satisfies $Bx_{LS} = BB^T(BB^T)^{-1}(2 \cdot 1_n) = 2 \cdot 1_n$. Consequently, there holds

$$2 \cdot n \;=\; 2 \cdot 1_n^T 1_n \;=\; 1_n^T(2 \cdot 1_n) \;=\; 1_n^T(Bx_{LS}) \;=\; (B^T 1_n)^T x_{LS} \;=\; 2 \cdot 1_m^T x_{LS}. \tag{5}$$

If G is bipartite (and connected), then the rank of B is $n-1$ (cf. [11]) and its rows are linearly dependent. Hence, one can order columns of B^T (i.e. vertices of G) so that

$$B^T w = 0 \quad \text{for} \quad w = (\underbrace{1, 1, \ldots, 1}_{n_1-\text{times}}, \underbrace{-1, -1, \ldots, -1}_{n_2-\text{times}})^T. \tag{6}$$

Considering the singular value decomposition of B in the form $B = U\Sigma V^T$, the Moore-Penrose inverse of B has a form $B^\dagger = V\Sigma^\dagger U^T$ with $\Sigma_{nn} = \Sigma^\dagger_{nn} = 0$ being the singular value corresponding to the left singular vector[2] $u = \frac{1}{\|w\|}\, w = \frac{1}{\sqrt{n}}\, w$, i.e. to the last column of the matrix U. Consequently, for bipartite graphs there holds

$$\begin{aligned}
1_m^T x_{LS} \;&=\; \frac{1}{2}(B^T 1_n)^T x_{LS} \;=\; \frac{1}{2}(B^T 1_n)^T (2B^\dagger 1_n) \;=\; 1_n^T B B^\dagger 1_n \\
&=\; 1_n^T U\Sigma V^T V \Sigma^\dagger U^T 1_n \;=\; 1_n^T U \Sigma\Sigma^\dagger U^T 1_n \;=\; 1_n^T U(I_n - e_n e_n^T) U^T 1_n \\
&=\; 1_n^T 1_n - (1_n^T U e_n)^2 \;=\; n - (1_n^T u)^2 \;=\; n - \frac{1}{n}(1_n^T w)^2 \;=\; n - \frac{(n_1 - n_2)^2}{n} \\
&=\; n - \frac{(n_1 + n_2)^2 - 4n_1 n_2}{n} \;=\; \frac{4n_1 n_2}{n},
\end{aligned} \tag{7}$$

where we applied the equality $B^T 1_n = 2 \cdot 1_m$ resulting from the fact that each row of B^T contains exactly two ones (i.e. each edge connects two vertices). □

Lemma 2. *Let $x \in \{0,1\}^m$ be a vertex-disjoint cycle cover, then*

$$\|x - x_{LS}\|^2 \;=\; n - \|x_{LS}\|^2. \tag{8}$$

[2] In the whole paper by the expression $\|x\| = \sqrt{x^T x}$ we denote the standard Euclidean norm of the vector x.

Proof. Since $Bx = 2 \cdot 1_n$ and $x_{LS} = 2B^\dagger 1_n$, there holds

$$\begin{aligned}
\|x - x_{LS}\|^2 &= (x - 2B^\dagger 1_n)^T (x - 2B^\dagger 1_n) = \\
&= \|x\|^2 - 4x^T B^\dagger 1_n + 4 \cdot 1_n^T (B^\dagger)^T B^\dagger 1_n \\
&= n - 4x^T B^\dagger B B^\dagger 1_n + 4 \cdot 1_n^T (B^\dagger)^T B^\dagger 1_n \\
&= n - 4x^T (B^\dagger B)^T B^\dagger 1_n + 4 \cdot 1_n^T (B^\dagger)^T B^\dagger 1_n \\
&= n - 4(B^\dagger B x)^T B^\dagger 1_n + 4 \cdot 1_n^T (B^\dagger)^T B^\dagger 1_n \\
&= n - 8(B^\dagger 1_n)^T B^\dagger 1_n + 4 \cdot 1_n^T (B^\dagger)^T B^\dagger 1_n \;=\; n - \|x_{LS}\|^2, \qquad (9)
\end{aligned}$$

where we applied the equalities $B^\dagger = B^\dagger B B^\dagger$ (see e.g. [8]) and $Bx = 2 \cdot 1_n$. □

Corollary 3. *Let the graph G with the incidence matrix B contain a 2-factor. Then the least-square solution to the system $Bx = 2 \cdot 1_n$ satisfies*

$$\|x_{LS}\|^2 \;\leq\; n. \qquad (10)$$

When $\|x_{LS}\|^2 = n$, then $x = x_{LS}$ is the only 2-factor of the graph G.

Proof. The inequality (10) follows from the inequality $n - \|x_{LS}\|^2 = \|x - x_{LS}\|^2 \geq 0$. When $\|x_{LS}\|^2 = n$, we obtain $\|x - x_{LS}\|^2 = n - \|x_{LS}\|^2 = 0$ for any 2-factor x. This is possible for $x = x_{LS}$ only. □

Corollary 4. *All 2-factors x satisfy*

$$x^T x_{LS} \;=\; \|x_{LS}\|^2. \qquad (11)$$

Proof. The equality (11) results from the fact that $\|x\|^2 = n$ and from the relation

$$2 \cdot x^T x_{LS} \;=\; \|x\|^2 + \|x_{LS}\|^2 - \|x - x_{LS}\|^2 \;=\; \|x\|^2 + \|x_{LS}\|^2 - n + \|x_{LS}\|^2. \qquad (12)$$

□

Remark 5. *From the equality (9) it follows that each 2-factor x lies on the m-dimensional sphere centered in x_{LS} with the radius $\sqrt{n - \|x_{LS}\|^2}$. Thus, assuming the graph G contains k different 2-factors x_i, $i = 1, 2, \ldots, k$, with the mean value $\overline{x}_k = \frac{1}{k} \sum_{i=1}^{k} x_i$, the multi-dimensional version of the Berry–Esseen theorem gives*

$$\|x_{LS} - \overline{x}_k\| \;\leq\; C \cdot \frac{\sqrt{n - \|x_{LS}\|^2}}{\sqrt{k}} \;\overset{k \to \infty}{\longrightarrow}\; 0, \qquad (13)$$

providing x_i are independent and identically distributed on the sphere (see e.g. [2] and [3]). However, we expect that this assumption is not fulfilled in this case and the proper formulation for 2-factors needs more investigation. Nevertheless, from the experiments it follows that x_{LS} is, indeed, a good approximation of $\overline{x}$ for large k and that a result similar to (13) really holds (see Figure 2).

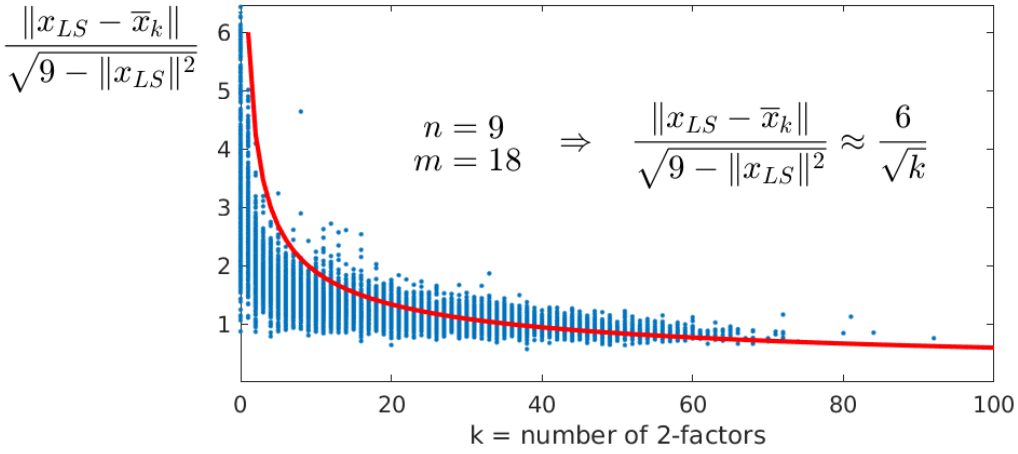


Figure 2: For graphs with a large number of 2-factors the least-square solution x_{LS} is a good approximation of $\overline{x}_k$. Here we considered all non-isomorphic graphs (see [7]) on $n = 9$ vertices and $m = 18$ edges. Each point corresponds to a single graph. The curve is a graph of the function $6/\sqrt{k}$.

Remark 6. *From the equality* (11) *it also follows that finding a* 2*-factor can be interpreted as finding* n *entries of the vector* x_{LS} *that sum up to* $\|x_{LS}\|^2$*. Hence, we obtain the so-called* 0-1 *knapsack problem with the prescribed number of items to include in a collection (for more details about knapsack problems, see e.g.* [5]*).*

Example 7. *Let us consider a graph formed by* 7 *vertices and* 9 *edges depicted on the Figure* 3 *(top left). It contains two* 2*-factors. If we compute the respective vector* x_{LS} *we realize that the values of* x_{LS} *entries are significantly higher for edges belonging to both* 2*-factors. This observation leads us to the strategy (S1) consisting in removing edges with the smallest* x_{LS}*-value.*

4. Sufficient condition

The following theorem provides a useful tool for determining which edge can be removed from the graph. Unfortunately, in most cases, none of the edges satisfy the condition (14) (see Figure 5*a*). In that situation we remove the edge with the highest value of the left-hand side of (14) (strategy S2).

Theorem 8. *Let* G *be a graph with the incidence matrix* B*, let* $e \in E(G)$ *be any edge such that* $G \setminus e$ *is a connected non-bipartite graph and let* x_{LS} *be the least-square solution to the system* $Bx = 2 \cdot 1_n$*. If there is*

$$\frac{(1 - [x_{LS}]_e)^2}{1 - e^T(BB^T)^{-1}e} \quad > \quad n - \|x_{LS}\|^2, \tag{14}$$

then the following implication holds

$$G \text{ has a 2-factor} \quad \Rightarrow \quad G \setminus e \text{ has a 2-factor.} \tag{15}$$

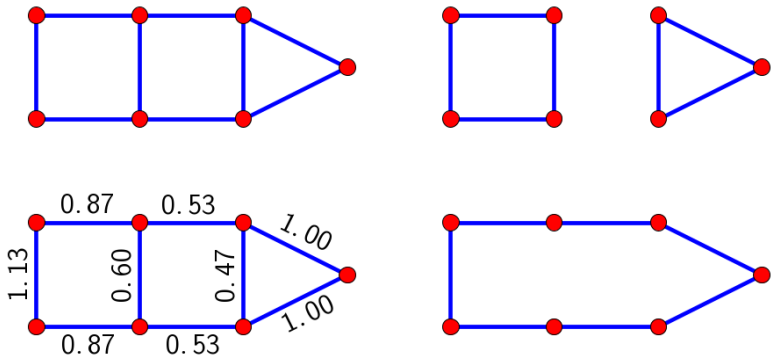


Figure 3: An example of a graph (top left) with two 2-factors (right). The values of x_{LS} entries (bottom left) are significantly higher for edges belonging to both 2-factors.

Proof. For a contradiction, let us suppose that the inequality (14) holds and all 2-factors $x \in \{0,1\}^{m\times 1}$ of the graph G satisfy $[x]_e = 1$. Let us denote by $B_e \in \{0,1\}^{n\times(m-1)}$ the matrix obtained from B by deleting the column corresponding to the edge e. Similarly, let us denote by $x_{LS,e}$ the vector that we obtain from x_{LS} by deleting the entry corresponding to the edge e. If we choose any 2-factor x of the graph G then the following equality holds

$$Bx \;=\; B_e x_e + e \;=\; 2\cdot 1_n \;=\; Bx_{LS} \;=\; B_e x_{LS,e} + [x_{LS}]_e \cdot e, \tag{16}$$

where x_e is obtained from x by deleting the entry corresponding to the edge e.

Hence, $B_e(x_{LS,e} - x_e) = (1 - [x_{LS}]_e)\cdot e$ and for the least-square solution z_{LS} of the system $B_e z = (1-[x_{LS}]_e)\cdot e$ there holds

$$\|z_{LS}\|^2 \;=\; (1-[x_{LS}]_e)^2\|B_e^\dagger e\|^2 \;\le\; \|x_{LS,e} - x_e\|^2 \;=\; \|x_{LS}-x\|^2 - ([x_{LS}]_e - 1)^2. \tag{17}$$

Thus, using the equality (9) we obtain an estimate

$$\|B_e^\dagger e\|^2 \;\le\; \frac{n - \|x_{LS}\|^2}{([x_{LS}]_e - 1)^2} - 1. \tag{18}$$

It remains to simplify the expression $\|B_e^\dagger e\|^2 = e^T(B_eB_e^T)^{-1}e$. For this purpose we apply the Sherman-Morrison formula (see [9])

$$(B_eB_e^T)^{-1} \;=\; (BB^T - e^Te)^{-1} \;=\; (BB^T)^{-1} + \frac{(BB^T)^{-1}ee^T(BB^T)^{-1}}{1 - e^T(BB^T)^{-1}e} \tag{19}$$

and obtain

$$\begin{aligned} e^T(B_eB_e^T)^{-1}e \;&=\; e^T(BB^T)^{-1}e + \frac{e^T(BB^T)^{-1}ee^T(BB^T)^{-1}e}{1 - e^T(BB^T)^{-1}e} \\ &=\; \frac{e^T(BB^T)^{-1}e}{1 - e^T(BB^T)^{-1}e} \;=\; \frac{1}{1 - e^T(BB^T)^{-1}e} - 1. \end{aligned} \tag{20}$$

Hence

$$\frac{1}{1-e^T(BB^T)^{-1}e} \leq \frac{n-\|x_{LS}\|^2}{([x_{LS}]_e-1)^2}, \tag{21}$$

which is in contradiction with the assumption (14). □

Remark 9. *A similar inequality to* (14) *can be derived in the case when* $G \setminus e$ *is a bipartite graph using formulas for the Moore–Penrose inverse of modified matrices, for more details see e.g.* [1] *or* [6].

5. Minimizing the length of x_{LS}

With the aid of [7] we have computed the values of $\|x_{LS}\| = \left\|x_{LS}^G\right\|$ for all connected non-isomorphic graphs G on 9 vertices with a minimal vertex degree 2 (see Figure 4) and found out that the value of $\left\|x_{LS}^G\right\|$ is significantly smaller for graphs G containing a large number of 2-factors. This has lead us to the strategy (strategy S3) consisting in removing the edge $e \in E(G)$ with a property

$$e = \arg\min_{\widehat{e}\in E(G)} \left\|x_{LS}^{G\setminus\widehat{e}}\right\| \tag{22}$$

(see Figure 5*b* for an example of an application of the strategy S3).

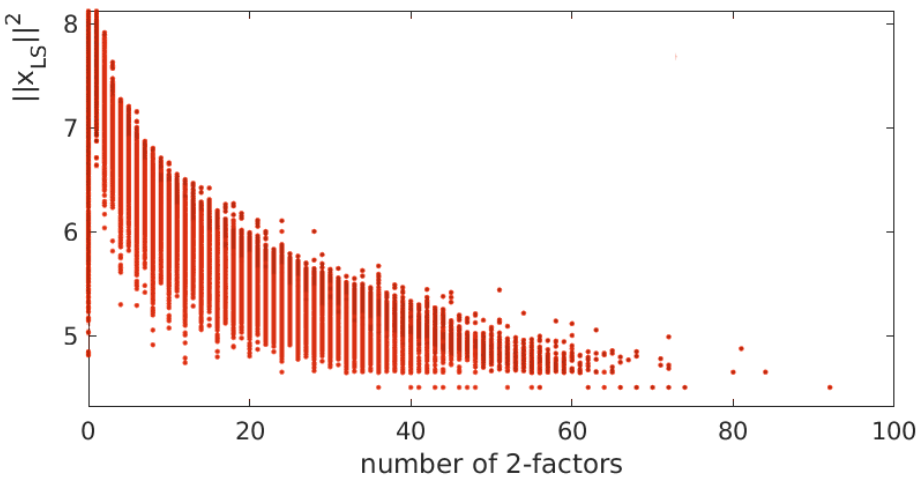


Figure 4: For graphs with a large number of 2-factors the norm of the least-square solution x_{LS} is significantly smaller. Thus, in order to preserve a maximum number of 2-factors in the graph we always try to remove the edge that minimizes $\|x_{LS}^{G\setminus e}\|$ (strategy S3). Here the results for connected non-isomorphic graphs with 9 vertices and minimal vertex degree 2 are shown (each point represents one graph).

For computing $\left\|x_{LS}^{G\setminus e}\right\|$ we use the following lemma.

Lemma 10. *Let G be a graph with the incidence matrix B and let $e \in E(G)$ be any edge such that $G \setminus e$ is a connected non-bipartite graph. Further, let x_{LS}^{G} be the least-square solution to the system $Bx = 2 \cdot 1_n$ and let $x_{LS}^{G\setminus e}$ be the least-square solution to the system $B_e x = 2 \cdot 1_n$. Then there holds*

$$\left\|x_{LS}^{G\setminus e}\right\|^2 - \left\|x_{LS}^{G}\right\|^2 \;=\; \frac{[x_{LS}^{G}]_e^2}{1 - e^T(BB^T)^{-1}e}. \tag{23}$$

Proof. We employ the relations from the equalities (9) and (19) and obtain

$$\begin{aligned}
\left\|x_{LS}^{G\setminus e}\right\|^2 &= \left\|2B_e^{\dagger}1_n\right\|^2 = 4 \cdot 1_n^T (B_e B_e^T)^{-1} 1_n \\
&= 4 \cdot 1_n^T (BB^T)^{-1} 1_n + \frac{4 \cdot 1_n^T (BB^T)^{-1} e e^T (BB^T)^{-1} 1_n}{1 - e^T (BB^T)^{-1} e} \\
&= \left\|x_{LS}^{G}\right\|^2 + \frac{(2 \cdot e^T (BB^T)^{-1} 1_n)^2}{1 - e^T (BB^T)^{-1} e} = \left\|x_{LS}^{G}\right\|^2 + \frac{[x_{LS}^{G}]_e^2}{1 - e^T (BB^T)^{-1} e},
\end{aligned} \tag{24}$$

where we used the fact that

$$2 \cdot e^T (BB^T)^{-1} 1_n \;=\; [2 \cdot B^T (BB^T)^{-1} 1_n]_e \;=\; [2 \cdot B^{\dagger} 1_n]_e \;=\; [x_{LS}^{G}]_e \tag{25}$$

is the entry of the vector x_{LS}^{G} corresponding to the edge e. □

Remark 11. *As in the case of the inequality (14) similar equality to (23) can be derived when $G \setminus e$ is a bipartite graph using formulas for the Moore–Penrose inverse of modified matrices, for more details see again e.g. [1] or [6].*

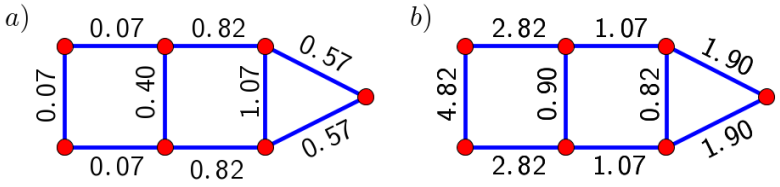


Figure 5: For the edges of the graph from the Example 7 we compute the values of the left-hand side of the inequality (14) (Figure 5*a*). The edge with the highest value (1.07) will be removed (strategy S2). Unfortunately, $n - \|x_{LS}\|^2 = 1.07$ in this case, hence, the condition (14) is not fulfilled. Analogously, we compute the values of the right-hand side of the equality (23) (Figure 5*b*). Then the edge with the smallest value (0.82) will be removed (strategy S3) in order to minimize the norm of $x_{LS}^{G\setminus e}$. Thus, for this graph, all three strategies lead to the deletion of the same edge.

6. Numerical experiments

We consider 10 000 randomly generated graphs with 32 vertices and 64 edges containing a Hamiltonian cycle. For each graph we apply all three strategies and successively remove edges. In each row of the Table 1 one can find results for each strategy employed. The numbers of graphs for which the algorithm failed are stored in the second column of the table. In the third to seventh column one can find the numbers of graphs for which the algorithm succeeded and the resulting 2-factor is formed by 1 to 5 components. In the last column an average number of components for each successfully ended strategy is shown.

strategy	failed	1 cmp	2 cmp	3 cmp	4 cmp	5 cmp	avg cmp
S1	372	4498	4297	774	59	0	1.6255
S2	59	5487	3337	930	161	26	1.5818
S3	3582	2096	2816	1232	247	27	1.9550

Table 1: Numerical results for all considered strategies.

7. Conclusion

Numerical experiments show that all three strategies considered were successful in more than 50 percent of all cases and from this point od view we shall say that the considerations from which they were derived were justified. The best result has been achieved by the strategy S2, which succeeded 99.41 percent of the time. The combination of all three strategies, as well as the involvement of some properties of the graph in the edge deletion decision, will be the subject of the future research.

Acknowledgements

The research was partially supported by the Czech Science Foundation grant 20-14736S and by project No. CZ.02.1.01/0.0/0.0/16_019/0000778 financially supported by the European Regional Development Fund (Center of Advanced Applied Sciences – CAAS).

References

[1] Baksalary, J. K , Baksalary, O. M. and Trenkler, G.: A revisitation of formulae for the Moore–Penrose inverse of modified matrices, Linear Algebra and its Applications, 372 (2003), pp. 207-224.

[2] Bentkus, V.: A Lyapunov type bound in R^d, Teor. Veroyatnost. i Primenen., **49** (2), 400–410, 2004.

[3] Esseen, C.-G.: On the Liapounoff limit of error in the theory of probability. Ark. Mat. Astr. Fys. 28A, (1942). no. 9, 19 pp.

[4] Garey, M. and Johnson, D. S.: Computers and Intractability: A Guide to the Theory of NP-Completeness, A Series of Books in the Mathematical Sciences. San Francisco, Calif.: W. H. Freeman and Co., 1979.

[5] Kellerer, H., Pferschy, U. and Pisinger, D.: Knapsack problems, Springer, 2004.

[6] Meyer, Jr. and Carl D.: Generalized Inversion of Modified Matrices, SIAM Journal on Applied Mathematics, **24** (3), 315–323, 1973.

[7] McKay, B.: Collection of combinatorial data - list of simple connected graphs: http://users.cecs.anu.edu.au/~bdm/data/graphs.html

[8] Penrose, R.: A generalized inverse for matrices. Mathematical Proceedings of the Cambridge Philosophical Society, **51**(3), 406-413, 1955.

[9] Sherman, J. and Morrison, W. J.: Adjustment of an Inverse Matrix Corresponding to a Change in One Element of a Given Matrix, Annals of Mathematical Statistics, **21** (1), 124–127, 1950.

[10] Tutte, W.: A Short Proof of the Factor Theorem for Finite Graphs, Canadian Journal of Mathematics, 6 (1954), 347–352.

[11] Van Nuffelen, C.: On the incidence matrix of a graph, IEEE Transactions on Circuits and Systems, **23** (9), pp. 572-572, 1976.

Programs and Algorithms of Numerical Mathematics 21
J. Chleboun, P. Kůs, J. Papež, M. Rozložník, K. Segeth, J. Šístek (Eds.)
Institute of Mathematics CAS, Prague 2023

DETERMINATION OF THE INITIAL STRESS TENSOR FROM DEFORMATION OF UNDERGROUND OPENING — THEORETICAL BACKGROUND AND APPLICATIONS

Josef Malík, Alexej Kolcun

Institute of Geonics, The Czech Academy of Sciences
Studetská 1768, 708 00 Ostrava-Poruba, Czech Republic
josef.malik@ugn.cas.cz, alexej.kolcun@ugn.cas.cz

Abstract: In this paper a method for the detection of initial stress tensor is proposed. The method is based on measuring distances between some pairs of points located on the wall of underground opening in the excavation process. This methods is based on the solution of eighteen auxiliary problems in the theory of elasticity with force boundary conditions. The optimal location of the pairs of points on the wall of underground work is studied. The pairs must be located so that the condition number of a certain matrix has the minimal value, which guarantees a reliable estimation of initial stress tensor.

Keywords: initial stress tensor, first boundary value problem of the theory of elasticity, condition number of matrices

MSC: 65N30, 74-XX, 93E24

1. Introduction

The knowledge of initial stress tensor is very important when one evaluates the stability of underground openings like tunnels, compressed gas tanks or radioactive waste deposits. The knowledge of initial stress tensor enables to optimize the reinforcement of tunnels, choose the suitable shape of underground works and their orientation in the rock environment. The mathematical modeling of stress fields in the vicinity of underground openings requires precise boundary conditions, which can be derived from initial stress tensor. Extensive literature is devoted to the determination of initial stress tensor. An overview of these methods can be found in the papers [1]–[3] that describe the development of these methods to the present. Theoretical and practical aspects of these methods are studied in [4] and [5]. These methods are based on the installation of probes equipped with sensors that measure deformations occurring after removal rock, overcoring, in their vicinity. Due to the stress in the rock, the removal of a part of the rock causes deformation of the remaining rock, which is transfered to the sensors. The probes are relatively small, a few

DOI: 10.21136/panm.2022.11

centimeters, and the accuracy of such measurements is not high. A very interesting method that allows to determine the whole initial stress tensor is presented in [6].

In this paper we present a new method, which is based on measuring the distances between pairs of selected points on the walls of the underground work. When a part of the rock is excavated, the distance between these points changes and the magnitude of these changes depends on the initial stress tensor. A procedure which allows to determine the initial stress tensor from the measured distances is developed. A criterion showing how to select measuring points so that errors of measurement do not affect the results very much is presented. This method guarantees a reliable estimate of the initial stress tensor. The procedure of optimal choice of measuring points is based on the conditional number of the matrix corresponding with choice of measuring points.

2. Auxiliary results

The method described in this section is based on the solution of the first boundary problem of the theory of elasticity, e.i. only the force conditions are prescribed on the boundary of the domain, where the problem is solved. A typical problem solving domain is shown in Figure 1.

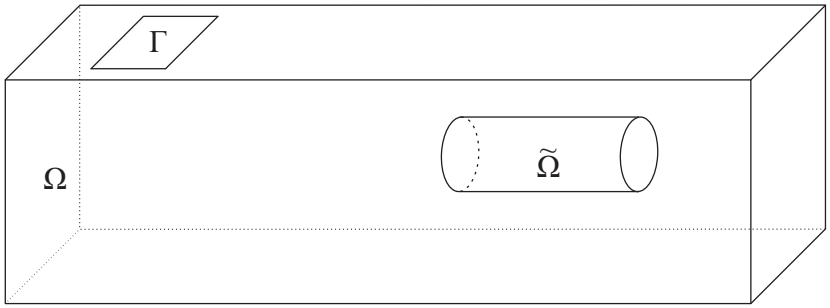


Figure 1: Typical problem solving area.

The symbol Ω in Figure 1 is the domain that corresponds to the prism and the symbol $\widetilde{\Omega}$ is the subdomain that represents the excavated space in the domain Ω. The symbol Ω_1 corresponds to domain $\Omega - \widetilde{\Omega}$ and $\Gamma \subset \partial\Omega$ has a nonzero measure. Let us have a space $V = [H^1(\Omega_1)]^3$, where $H^1(\Omega_1)$ is a Sobolev space of functions having first-oder derivatives that are integrable with the second power. We will continue to apply the Einstein summation convention.

Let us formulate the first variational problem $\mathcal{D}_1$ whose solution is a minimum of the following functional on V

$$\frac{1}{2}\int_{\Omega_1} c_{ijkl}e_{ij}(u)e_{kl}(u)\,\mathrm{d}x - \int_{\partial\Omega} P_i u_i\,\mathrm{d}S, \tag{1}$$

where $u = (u_1, u_2, u_3)$ belongs to V and

$$e_{ij}(u) = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$$

is the tensor of small deformations. The symbol $P = (P_1, P_2, P_3)$ represents the forces on $\partial\Omega$ and $P_i \in L^2(\partial\Omega)$. The coefficient $c_{ijkl} \in L^\infty(\Omega_1)$ meet the following conditions

$$c_{ijkl} = c_{jikl} = c_{ijlk} = c_{klij}. \tag{2}$$

There is a constant $C > 0$ such that the inequality

$$c_{ijkl} e_{ij} e_{kl} \geq C e_{ij} e_{ij} \tag{3}$$

holds for all symmetric tensors e_{ij}. The problem $\mathcal{D}_1$ is solvable when the conditions

$$\int\limits_{\partial\Omega} P_i \,\mathrm{d}S = 0, \quad \int\limits_{\partial\Omega} (x \times P)_i \,\mathrm{d}S = 0 \tag{4}$$

are met. This problem is not uniquely solvable and it has infinite number of solutions. If $u_1(x)$ and $u_2(x)$ are two solutions then

$$u_2(x) - u_1(x) = Ax + b, \tag{5}$$

where A is an antisymmetric matrix 3×3 and b is a vector from R^3. This problem can be modified so that it will be uniquely solvable and this solution will be the minimum of the functional (1), i.e., the solution of the problem $\mathcal{D}_1$, provided the conditions (4) are met.

Let us define functionals on V

$$\begin{aligned} g_\alpha(u) &= \int\limits_\Gamma u_\alpha \,\mathrm{d}S, \quad \alpha = 1, 2, 3, \\ g_\alpha(u) &= \int\limits_\Gamma (x \times u)_{\alpha-3} \,\mathrm{d}S, \quad \alpha = 4, 5, 6. \end{aligned} \tag{6}$$

Then there is a constant $C > 0$ such that the following inequality

$$C \parallel u \parallel_V \leq \int\limits_{\Omega_1} c_{ijkl} e_{ij} e_{kl} \,\mathrm{d}x + g_\alpha(u) g_\alpha(u) \tag{7}$$

holds for all $u \in V$.

Let us formulate the second variational problem $\mathcal{D}_2$ whose solution is a minimum of the functional

$$\frac{1}{2} \int\limits_{\Omega_1} c_{ijkl} e_{ij}(u) e_{kl}(u) \,\mathrm{d}x + \frac{1}{2} g_\alpha(u) g_\alpha(u) - \int\limits_{\partial\Omega} P_i u_i \,\mathrm{d}S, \tag{8}$$

on V. The minimum of functional (8) is unique. Moreover the following inequality

$$\| u \|_V \leq C \| P \|_{[L^2(\partial\Omega)]^3} \tag{9}$$

holds, where C is a positive constant independent of u and P. The last inequality expresses the continuous dependence of the solution of the problem $\mathcal{D}_2$ on the force boundary conditions. Note that solving the problem $\mathcal{D}_2$ does not require the equilibrium conditions (4) to be met. But if these conditions are satisfied, the solution of $\mathcal{D}_2$ is a solution of $\mathcal{D}_1$. All these results can be found in the book [7].

Let τ_{ij} be a symmetric tensor. We say that the force boundary conditions P_i are generated by the tensor τ_{ij} when at every $x \in \partial\Omega$ the equation

$$P_i(x) = \tau_{ij} n_j(x) \tag{10}$$

holds, where $n_j(x)$ is a normal to the boundary $\partial\Omega$ at the point x.

Lemma 1. *Let τ_{ij} be a symmetric tensor and let P_i be defined by the formula (10) on the boundary $\partial\Omega$, then P_i satisfy the equilibrium conditions (4).*

Proof. If we use the Gaussian theorem on the surface integral, then

$$\int_{\partial\Omega} P_i(x)\,\mathrm{d}S = \int_{\partial\Omega} \tau_{ij} n_j(x)\,\mathrm{d}S = \int_{\Omega} \frac{\partial \tau_{ij}}{\partial x_j}\,\mathrm{d}x.$$

Since τ_{ij} is constant, then the last integral is zero. We express the formula $x \times P$ in the individual components, then

$$\begin{aligned}(x \times P)_1 &= x_2\tau_{3j}n_j - x_3\tau_{2j}n_j,\\ (x \times P)_2 &= x_3\tau_{1j}n_j - x_1\tau_{3j}n_j,\\ (x \times P)_3 &= x_1\tau_{2j}n_j - x_2\tau_{1j}n_j,\end{aligned}$$

where $n = (n_1, n_2, n_3)$ is the normal to the boundary $\partial\Omega$ at x. If we use the Gaussian theorem on the surface integral, then

$$\int_{\partial\Omega} (x \times P)_1\,\mathrm{d}S = \int_{\partial\Omega} x_2\tau_{3j}n_j - x_3\tau_{2j}n_j\,\mathrm{d}S = \int_{\Omega} \frac{\partial(x_2\tau_{3j} - x_3\tau_{2j})}{\partial x_j}\,\mathrm{d}x.$$

Since τ_{ij} is symmetric and constant, then the last integral is zero. The same equations can be proved for the components $(x \times P)_2$ and $(x \times P)_3$. □

The inequality (9) implies the existence of a continuous mapping

$$K\colon S^{\mathrm{sym}} \longrightarrow V, \tag{11}$$

where S^{sym} is the set of all second oder symmetric tensors. This mapping assigns a solution to the problem $\mathcal{D}_2$ to each second order symmetric tensor. Lemma 1 indicates that the value of this mapping is also a solution to the problem $\mathcal{D}_1$.

The following lemma will be useful in formulating the problem for determining the initial stress tensor.

Lemma 2. *Let* $u_1, u_2, x_1, x_2 \in R^3$ *and the value*

$$a = \frac{\|u_1 - u_2\|}{\|x_1 - x_2\|} < 1$$

be such small that a^2 *can be neglected, then the following equality*

$$\|u_1 + x_1 - u_2 - x_2\| - \|x_1 - x_2\| = \frac{\langle u_1 - u_2, x_1 - x_2\rangle}{\|x_1 - x_2\|}$$

holds approximately, where $\langle \cdot, \cdot \rangle$ *is the scalar product in* R^3*. Moreover, if*

$$v_1 = u_1 + Ax_1 + b, \qquad v_2 = u_2 + Ax_2 + b,$$

where A *is an antisymmetric matrix* 3×3 *and* b *is a vector from* R^3*, then*

$$\frac{\langle v_1 - v_2, x_1 - x_2\rangle}{\|x_1 - x_2\|} = \frac{\langle u_1 - u_2, x_1 - x_2\rangle}{\|x_1 - x_2\|}.$$

The proof of this lemma can be found in [8].

3. Description of the method

The method will be described on the geometry of two intersecting tunnels, which correspond to the real situation when the original stress tensor was determined. The situation is shown in Figures 2–4.

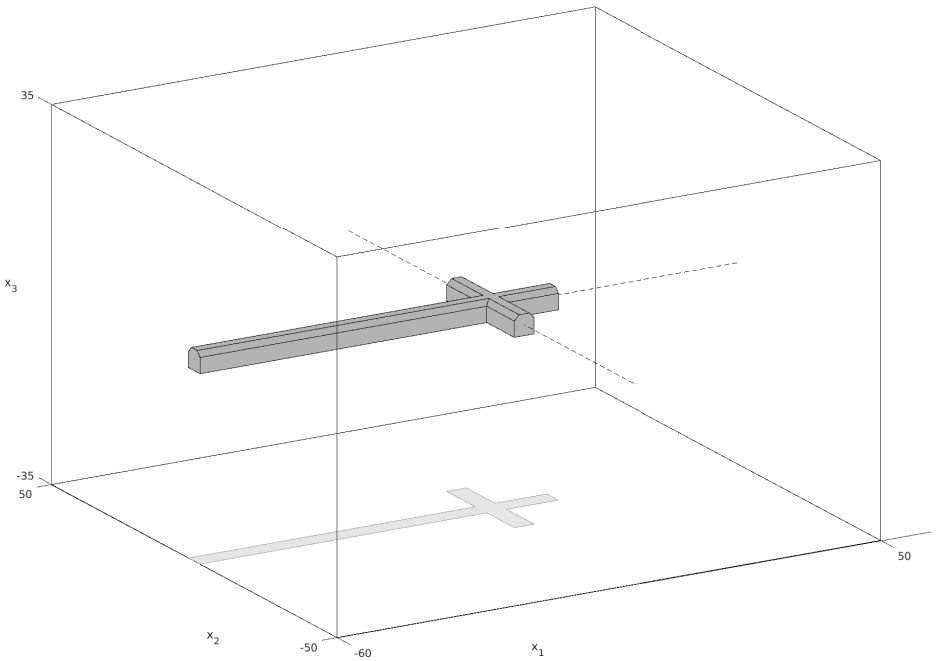


Figure 2: Underground opening in homogeneous domain Ω.

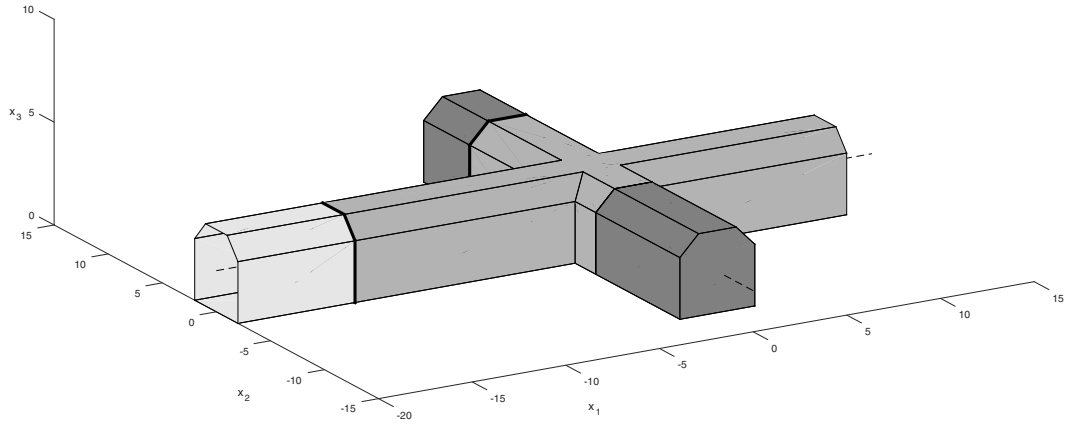


Figure 3: Detail of underground opening with highlighted three steps of the excavation process.

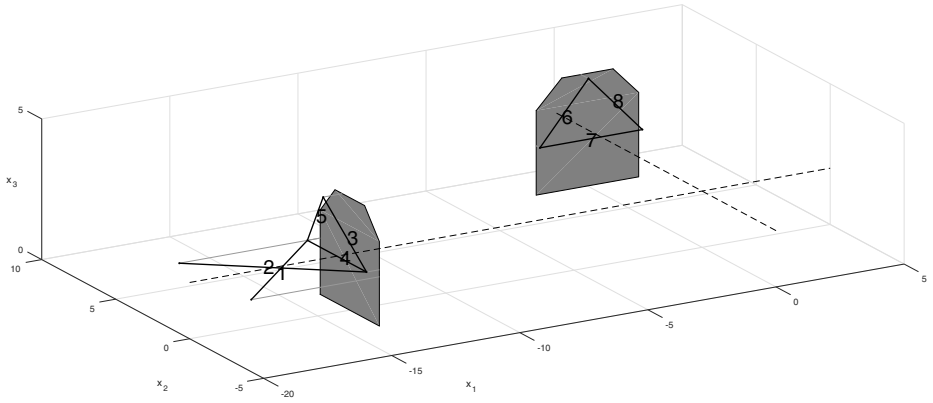


Figure 4: The pairs of measuring points corresponding to the highlighted faces of the tunnels from Figure 2.

Figure 2 shows the domain, where the underground opening is located, Figure 3 shows three steps of the excavation process. Highlighted faces of the tunnels indicate the location of the measuring points. Figure 4 shows the position of the pairs of measuring points to the highlighted faces from Figure 3. The set of the pairs of measuring points is divided into the two subsets $I_1 = \{1, 2, 3, 4, 5\}$ and $I_2 = \{6, 7, 8\}$.

Let Ω be the domain that corresponds to the prism shown in Figure 2. Let Ω_1, Ω_2, Ω_3 be the subdomains derived from the domain Ω by extraction of the parts corresponding to three steps of the excavation process shown in Figure 3.

In *Step* 1 (light gray color in Figure 3) the measuring points are installed on the walls of the tunnel and the distances between the selected pairs I_1 of measuring points are measured, see pairs $1, 2, 3, 4, 5$ in Figure 4. These measuring points are located at the ends of short steel bars, which are glued to the rock on the walls of the tunnel.

In *Step* 2 (medium gray color in Figure 3) another part of the main tunnel and a part of the tunnel oriented perpendicular to the main tunnel are excavated. The distances between the pairs I_1 are re-measured. The values obtained in *Step* 1 are subtracted from the values measured in *Step* 2 and the resulting values are marked d_i, $i \in I_1$. Moreover, another group of measuring points is installed on the walls of the tunnel perpendicular to the main tunnel – pairs 6, 7, 8 in Figure 4. The distances between the selected pairs I_2 of points are measured.

In *Step* 3 (dark gray color in Figure 3) remaining part of the perpendicular tunnel is extracted and then the distances in the second set I_2 of pairs of measuring points are re-measured. The values obtained in *Step* 2 are subtracted from the values measured in *Step* 3 and the resulting values are marked d_j, $j \in I_2$. The procedure for selecting the pairs of measuring points will be described below. We now explain how to obtain the initial stress tensor from these measurements.

Let's approach the formulation of our task. Let τ_{ij} be a symmetric second-order tensor that corresponds to the original stress tensor. We say that the force boundary conditions $\mathbf{P} = (P_1, P_2, P_3)$ are generated at the boundary of the domain by this tensor if

$$P_i(\mathbf{x}) = \tau_{ij} n_j(\mathbf{x}) \tag{12}$$

holds, where $\mathbf{n}(\mathbf{x})$ is the normal to the boundary of the domain at the point $\mathbf{x}$.

Assume that we know the solutions $\mathbf{u}_1(\mathbf{x}), \mathbf{u}_2(\mathbf{x}), \mathbf{u}_3(\mathbf{x})$ of the problem (1) on Ω_1, Ω_2, Ω_3 with the force boundary conditions given by the expression (12) on $\partial\Omega$ and be equal to zero of the remainders of the boundaries $\partial\Omega_1$, $\partial\Omega_2$, $\partial\Omega_3$. Due to Lemma 1 the solutions exist. Let the pairs of the points $\mathbf{x}_i, \mathbf{y}_i$, $i \in I_1$ in the first set and the pairs $\mathbf{x}_j, \mathbf{y}_j$, $j \in I_2$ in the second set be selected and the calculated differences in distances between these points are compared with numbers d_i, $i \in I_1$ and d_j, $j \in I_2$. These numbers represents the differences in the distances measured in the excavation process. The differences are equal to the following expressions

$$\begin{aligned} \|\mathbf{u}_2(\mathbf{x}_i) + \mathbf{x}_i - \mathbf{u}_2(\mathbf{y}_i) - \mathbf{y}_i\| - \|\mathbf{u}_1(\mathbf{x}_i) + \mathbf{x}_i - \mathbf{u}_1(\mathbf{y}_i) - \mathbf{y}_i\| = d_i, \quad i \in I_1, \\ \|\mathbf{u}_3(\mathbf{x}_j) + \mathbf{x}_j - \mathbf{u}_3(\mathbf{y}_j) - \mathbf{y}_j\| - \|\mathbf{u}_2(\mathbf{x}_j) + \mathbf{x}_j - \mathbf{u}_2(\mathbf{y}_j) - \mathbf{y}_j\| = d_j, \quad j \in I_2. \end{aligned} \tag{13}$$

In geo-mechanical practice, the displacements $\|\mathbf{u}(\mathbf{x}) - \mathbf{u}(\mathbf{y})\|$ is much smaller than $\|\mathbf{x} - \mathbf{y}\|$. Under these assumptions and Lemma 2, the relations (13) can be rewritten into the following form

$$\begin{aligned} \frac{\langle \mathbf{u}_2(\mathbf{x}_i) - \mathbf{u}_2(\mathbf{y}_i), \mathbf{x}_i - \mathbf{y}_i \rangle}{\|\mathbf{x}_i - \mathbf{y}_i\|} - \frac{\langle \mathbf{u}_1(\mathbf{x}_i) - \mathbf{u}_1(\mathbf{y}_i), \mathbf{x}_i - \mathbf{y}_i \rangle}{\|\mathbf{x}_i - \mathbf{y}_i\|} = d_i, \quad i \in I_1, \\ \frac{\langle \mathbf{u}_3(\mathbf{x}_j) - \mathbf{u}_3(\mathbf{y}_j), \mathbf{x}_j - \mathbf{y}_j \rangle}{\|\mathbf{x}_j - \mathbf{y}_j\|} - \frac{\langle \mathbf{u}_2(\mathbf{x}_j) - \mathbf{u}_2(\mathbf{y}_j), \mathbf{x}_j - \mathbf{y}_j \rangle}{\|\mathbf{x}_j - \mathbf{y}_j\|} = d_j, \quad j \in I_2, \end{aligned} \tag{14}$$

what is proved in [8]. Moreover the equations (14) are more suitable for further analysis and all mathematical aspects are also explained in [8]. The symbol $\langle \mathbf{x}, \mathbf{y} \rangle = x_i y_i$ in the expression (14) is the scalar product in R^3.

Now let's focus on finding the original stress tensor and consider the following six stress tensors.

$$\tau_{ij}^1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \tau_{ij}^2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \tau_{ij}^3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$\tau_{ij}^4 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \tau_{ij}^5 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad \tau_{ij}^6 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$

Next, let us denote $\mathbf{u}_1^k(\mathbf{x})$, $\mathbf{u}_2^k(\mathbf{x})$, $\mathbf{u}_3^k(\mathbf{x})$, $k = 1, \ldots, 6$ the solutions of the boundary value problems of the theory of elasticity on the domains Ω_1, Ω_2, Ω_3 with the force boundary conditions given by the tensors τ_{ij}^k using the relation (3) on $\partial\Omega$. The forces prescribed on $\partial\Omega_1 - \partial\Omega$, $\partial\Omega_2 - \partial\Omega$, $\partial\Omega_3 - \partial\Omega$, which corresponds to the walls of the gradually excavated tunnels, are zero.

As far as we use linear elastic model, we assume, that resulting displacements are linear combination of the displacements induced by auxiliary problems. So, the solution to our problem is the vector $\mathbf{z} = (z_1, z_2, z_3, z_4, z_5, z_6)$ such that the functions

$$\mathbf{u}_1(\mathbf{x}) = z_k \mathbf{u}_1^k(\mathbf{x}), \quad \mathbf{u}_2(\mathbf{x}) = z_k \mathbf{u}_2^k(\mathbf{x}), \quad \mathbf{u}_3(\mathbf{x}) = z_k \mathbf{u}_3^k(\mathbf{x})$$

satisfy the relations (5) and the tensor

$$\tau_{ij} = z_k \tau_{ij}^k \tag{15}$$

is the original stress tensor. To simplify further analysis, we will use the following designations

$$\begin{aligned} h_i^k &= \frac{\langle \mathbf{u}_2^k(\mathbf{x}_i) - \mathbf{u}_2^k(\mathbf{y}_i), \mathbf{x}_i - \mathbf{y}_i \rangle}{\|\mathbf{x}_i - \mathbf{y}_i\|} - \frac{\langle \mathbf{u}_1^k(\mathbf{x}_i) - \mathbf{u}_1^k(\mathbf{y}_i), \mathbf{x}_i - \mathbf{y}_i \rangle}{\|\mathbf{x}_i - \mathbf{y}_i\|}, \quad i \in I_1, \\ h_j^k &= \frac{\langle \mathbf{u}_3^k(\mathbf{x}_j) - \mathbf{u}_3^k(\mathbf{y}_j), \mathbf{x}_j - \mathbf{y}_j \rangle}{\|\mathbf{x}_j - \mathbf{y}_j\|} - \frac{\langle \mathbf{u}_2(\mathbf{x}_j) - \mathbf{u}_2(\mathbf{y}_j), \mathbf{x}_j - \mathbf{y}_j \rangle}{\|\mathbf{x}_j - \mathbf{y}_j\|}, \quad j \in I_2, \end{aligned} \tag{16}$$

which are connected with the relations (14). It is possible to use the least squares method that was proposed in [8], but now we will use a simpler method. Let us choose the set $J = J_1 \cup J_2$ such that $J_1 \subset I_1$, $J_2 \subset I_2$ and the number of elements of the set J is six. The vector $\mathbf{z}$ is a solution of the system of linear equations

$$\mathbf{H}\mathbf{z} = \mathbf{d}, \tag{17}$$

where the elements of 6×6 matrix $\mathbf{H} = (h_j^k)$, $1 \le k \le 6$, $j \in J$, we can find according to (16) and the components of the vector $\mathbf{d}$ are results of the measuring process described in *Step* 1–*Step* 3 for $j \in J$.

After solving the system (17), the initial stress tensor can be expressed in the form (15). It is very important to install pairs of measuring points so that the matrix $\mathbf{H}$ has favorable properties for solving the problem. Note that in the case of an insulated tunnel that does not intersect another tunnel, we could not find the location of the measuring points so that the matrix $\mathbf{H}$ has favorable properties. We will deal with this issue in the following section.

4. Optimal choice of measuring points

In this section, we will deal with the question of how to select pairs of points so that the system (17) has good properties from the point of view of the solvability of the problem. We select a pair of measuring points so that the matrix $\mathbf{H}$, which is constructed using formulas (16), has the property that a small change of the vector $\mathbf{d}$, the right side of the system (17), does not have much effect on the solution. This property is connected to the condition number $\kappa(\mathbf{H})$ of the matrix $\mathbf{H}$, which is expressed by the following formula

$$\kappa(\mathbf{H}) = \|\mathbf{H}\|\|\mathbf{H^{-1}}\|, \tag{18}$$

where $\|\mathbf{H}\|$ is the matrix norm of the matrix $\mathbf{H}$ and $\mathbf{H^{-1}}$ is the inverse matrix to $\mathbf{H}$. Then the relationship between the change of the solution $\delta\mathbf{z}$ of the system (16) and the change of the right side $\delta\mathbf{d}$ can be expressed by the following formula

$$\frac{\|\delta\mathbf{z}\|}{\|\mathbf{z}\|} \leq \kappa(\mathbf{H})\frac{\|\delta\mathbf{d}\|}{\|\mathbf{d}\|}.$$

The last inequality implies that for the reliability of the solution of the system (17) it is necessary that the condition number (18) is as small as possible. These results can be found in textbooks of linear algebra, for example in [9]. When we look at the way the matrices $\mathbf{H}$ are constructed using the expressions (16), we find that by a suitable choice of pairs of measuring points we are able to influence the condition number. The optimal distribution of measuring points is achieved by solving several auxiliary problems of the theory of elasticity and choosing a finite set of measuring points at the boundary of the underground opening. Then we select different sets of pairs of measuring points and check the value of the condition number of the matrix $\mathbf{H}$ composed of these pairs. As an optimal selection, we choose the set of pairs of measuring points for which the conditional number of the matrix $\mathbf{H}$ is the smallest. The selection of this set is based on mathematical modeling and specific cases will be analyzed in the next section. Note that if we can find the pairs of measuring points so that the conditional number of the corresponding matrix is less than 10 and we are able to guarantee a measurement accuracy of 1%, then the original stress tensor obtained by the methods described above will differ from the actual original stress tensor by 10 %.

We can say that the optimal selection of pairs of measuring points eliminates the effect of measurement errors. When applying the above method, we proceed as follows. We approximate the domains, shown in Figures 2–3 by finite element mesh and solve 18 auxiliary problems as described in Section 2. Then we select six pairs of points – nodes of the finite element mesh, compile the matrix and calculate the conditional number of this matrix. It is necessary to choose six pairs of points as far as the stress tensor has six independent components. We are looking for pairs of points on the walls of the underground opening so that it is possible to measure the distance between these points. We pass all suitable pairs of points and select the six pairs of points for which the conditional number of the matrix $\mathbf{H}$ is the smallest.

We will test this procedure on a pair of perpendicular vertical tunnels in the following section. We selected eight pairs of points as shown in Figure 4 and from this set we selected six pairs of points to show that the conditional number of the matrix depends on this selection. This procedure can be applied to various shapes of underground openings and mining operations. The shape of the underground opening and the excavation process were chosen in accordance with the research plan at the Bukov locality in the Czech Republic. This site serves as an underground laboratory and model repository for radioactive waste and is described in the report [11]. If we know the future shape of the underground opening and the progress of the mining work, we can use mathematical modeling, as described above, to select six pairs of measuring points in a suitable way and determine the tensor of the original stress.

We tried to apply this procedure to a direct tunnel and two steps of mining operations, but we were unable to find six pairs of measuring points so that the conditional number of the matrix $\mathbf{H}$ would be less than 60.

5. In situ experiment

The above-mentioned underground laboratory is located in a metamorphic rock, which is considered isotropic and homogeneous in the vicinity of the underground opening. The elastic properties of this rock are known from laboratory measurements, namely Yong's modulus $E = 60\,\mathrm{GPa}$ and Poisson's ratio $\mu = 0.25$. The expected shape of the underground opening and the progress of the mining works were known, which corresponds to Figure 2. Block Ω, into which two intersecting tunnels are nested, has dimensions $110\,\mathrm{m} \times 100\,\mathrm{m} \times 70\,\mathrm{m}$. The diameter of the tunnels is 4 m. The tunnel in the x_1 direction is long 70 m and the tunnel in the x_2 direction is long 30 m.

The GEM program, see [10], was used for analysis of the planned in situ experiment. The program has been developed at Institute of Geonics for solving geomechanical problems and allows solving elastic problems with force boundary conditions. The program has been used to solve 18 auxiliary problems as described in Section 3. However, it is possible to use any commercial program that allows you to solve such tasks.

A MATLAB program was written that tested all six possible pairs of points that were on the walls of the underground opening and the points were nodes of the finite element mesh used to solve the auxiliary problems. To demonstrate that the conditional number strongly depends on the selection of six pairs of measuring points, we selected six pairs of points from a set of eight elements, as shown in Figure 4. Six pairs from an eight-element set can be selected in twenty-eight ways. The results of this analysis are shown in Table 1. The selection is realized by two subsets $J_1 \subset I_1$ and $J_2 \subset I_2$, as described in Section 3. The configurations of the pairs of measuring points **c1** and **c2** correspond to the results with the two smallest conditional numbers of the matrix $\mathbf{H}$ and the configurations **c3** and **c4** correspond to the results with the two largest conditional numbers. The values of the conditional numbers are on the last line of Table 1.

c1		**c2**		**c3**		**c4**	
J_1	J_2	J_1	J_2	J_1	J_2	J_1	J_2
1,2,3	6,7,8	1,2,5	6,7,8	1,2,3,4	6,7	3,4,5	6,7,8
4.43		4.95		393.16		839.21	

Table 1: Selection of subsets J_1, J_2 and corresponding conditional numbers $\kappa(\mathbf{H})$.

d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8
-2.88	-0.60	-2.46	-5.32	-1.04	-1.75	-3.90	-2.79

Table 2: Measured differences in deformations in excavation process in [mm].

After this analysis, measurements were performed on selected pairs of measuring points in the corresponding steps of the mining process, as shown in Figures 2–3. The results of these measurements are shown in Table 2. The indexes at the numbers d_i correspond to the indexes of the pairs of measuring points as shown in Figure 4. The measuring points were placed at the ends of 60 cm long steel anchors, which were fixed in the rock with LOKSET ampules with polyester resin. The distances between the points were measured with a tape extensometer. This extensometer makes possible to measure distances between points with an accuracy of 0.01 mm. Considering the measured values in Table 2, the resolution of the extensometer guarantees a measurement accuracy of 1%.

The calculation was then performed, and the components of the original stress tensors τ_{ij} with the main stresses λ_i are shown in Table 3. The tensors thus obtained are shown in the principal stresses in Figure 5.

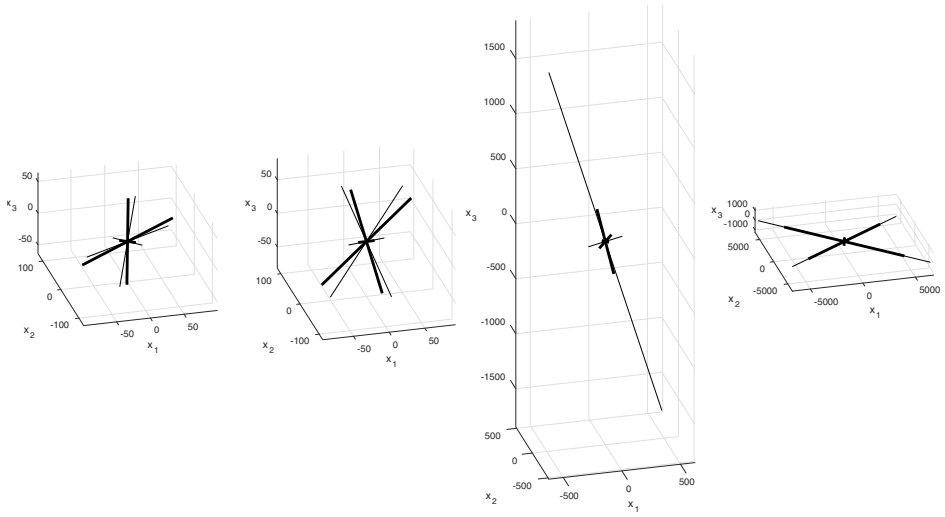
	τ_{11}	τ_{22}	τ_{33}	τ_{12}	τ_{23}	τ_{13}	λ_1	λ_2	λ_3
c1	-69.1	-95.6	-64.5	-70.0	10.7	13.3	-156.6	-61.7	-10.9
c2	-70.2	-94.8	-69.5	-63.1	-12.2	12.8	-146.8	-74.9	-12.8
c3	-12.5	-8.5	306.0	-47.7	-68.6	-100.8	-94.6	36.3	343.4
c4	-182.5	-12.6	143.0	-6362.7	588.6	656.9	-6576.3	258.3	6266.0

Table 3: Original stress tensor τ_{ij} and principal stresses λ_i in [MPa].

The directions of the principal stresses in the form of the unit eigenvectors $\mathbf{n}_1$, $\mathbf{n}_2$, $\mathbf{n}_3$ of the matrix $\mathbf{H}$ are shown in Table 4.

From Figure 5 and Tables 3–4, we can see that the tensors corresponding to the configurations **c1** and **c2** differ by about 10%, which is consistent with the hypothesis formulated in Section 4.

The configurations of pairs of measuring points **c3** and **c4** leads to very unlikely results, which corresponds very well to the fact that the conditional number of the matrix is large, as can be seen from Table 1.



c1 **c2** **c3** **c4**

Figure 5: The original stress tensors in principal stresses and directions obtained by analysis of the configurations **c1**–**c4** of measurements. Bold lines – full precision of measured differences from Table 2 is used. Thin lines – measured values rounded to milimeters are used.

c1			**c2**		
$\mathbf{n}_1$	$\mathbf{n}_2$	$\mathbf{n}_3$	$\mathbf{n}_1$	$\mathbf{n}_2$	$\mathbf{n}_3$
-0.630	0.063	0.773	-0.630	-0.240	-0.730
-0.750	0.179	-0.630	-0.770	0.176	0.609
0.178	0.982	0.066	-0.010	0.955	-0.290
c3			**c4**		
$\mathbf{n}_1$	$\mathbf{n}_2$	$\mathbf{n}_3$	$\mathbf{n}_1$	$\mathbf{n}_2$	$\mathbf{n}_3$
0.721	0.646	-0.250	-0.706	0.088	-0.703
0.630	-0.760	-0.152	-0.696	0.096	0.712
0.289	0.048	0.956	0.130	0.992	-0.007

Table 4: Principal stress vectors.

The dependence of the stability of numerical solutin on the condition number we can see from the Figure 5 and from the Table 5. When we round the measured values in Table 2 to whole milimeters, tensors for configurations **c1** and **c2** changed slightly and tensors for configurations **c3** and **c4** changed significantly, which is also in line with the hypothesis formulated in Section 4.

Table 5 shows the differences

$$\Delta d = \frac{|d - \hat{d}|}{|d|}, \quad \Delta\tau = \frac{|\tau - \hat{\tau}|}{|\tau|},$$

	c1	**c2**	**c3**	**c4**
Δd	0.11	0.09	0.10	0.08
$\Delta\tau$	0.13	0.14	6.78	0.45

Table 5: Sensitivity of the value of resulting stress tensor to input data.

where $\hat{d}$ are the approximated values form Table 2 rounded to the milimeters and $\hat{\tau}$ is the stress tensor for these approximated values for the configurations **c1**–**c4**.

At the Bukov site, it was impossible to use the over-coring method or the cone probe method, which are described in [1], [6]. The rock in the Bukov locality is granular with a grain size of several millimeters, which is comparable to the size of the sensors on the probes in the above-mentioned. The individual measurements performed by these methods showed completely different results. The method proposed in this article shows stability of tensors for configurations **c1** and **c2**, while the resulting tensors for configurations **c3** and **c4** are not applicable. At the same time, the reasons for this behavior were explained.

In addition, the principal stress directions for tensors in configirations **c1** and **c2** coincide with the principal stress directions obtained by the hydraulic fracturing method, see [11]. The tunnel is located 550 m below the surface and at the specific rock mass 2700 $\mathrm{kg\,m^{-3}}$ so the component of the initial stress tensor τ_{33} should be equal to 14.8 MPa. The Young's modulus E must be changed so that the component τ_{33} of the original stress tensor is equal to this value. In Table 6, the original stress tensors for configurations **c1** and **c2** are recalculated, and the corresponding Young's modulus values are in the last column.

	τ_{11}	τ_{22}	τ_{33}	τ_{12}	τ_{23}	τ_{13}	λ_1	λ_2	λ_3	E
c1	-15.8	-22.0	-14.8	-16.1	2.5	3.1	-36.0	-14.2	-2.5	13.8
c2	-15.4	-20.9	-14.8	-13.9	-2.7	2.8	-32.3	-16.5	-2.8	12.8

Table 6: Modified original stress tensor and principal stresses in [MPa] and reduced Young's modulus E in [GPa].

The Young's modulus values were measured in the laboratory on a completely homogeneous sample, so such a reduction in value is acceptable. Note that in our case, the x_1 axis is oriented west-east and the original stress tensors are in natural coordinates and need not be transformed. Given the previous discussion, we can accept tensors for configurations **c1** and **c2** in Table 6 as the original stress tensor. These tensors differ by 10%.

6. Conclusion

In this paper, the authors presented a new method for determining the original stress tensor. This method is based on measuring the distance between pairs of selected points located on the walls of the underground work in the process of

the excavation. Part of the method is a procedure for selecting pairs of measuring points so that the estimate of the original stress tensor is reliable. This method is applicable to coarse-grained and anisotropic rocks, where other methods are not so successful. The measurements themselves are easy to carry out and the main work is connected with the selection of pairs of measuring points and the evaluation of measurements, which is based on mathematical modeling.

The authors believe that this method will find application in the construction of new underground openings.

Acknowledgment: This research has been supported by Czech Science Foundation (GAČR) through project No. 19-11441S, by European Union's Horizon 2020 research and innovation programme under grant agreement number 847593, and by The Czech Radioactive Waste Repository Authority (SÚRAO) under grant agreement number SO2020-017.

References

[1] Hudson, J. A., Cornet, F. H., Christiansson, R.: ISRM suggested methods for rock stress estimation - Part 1: strategy for rock stress estimation. International Journal of Rock Mechanics and Mining Sciences **40** (2003), 991–998.

[2] Sjőberg, J., Christiansen, R., Hudson, J. A.: ISRM suggested methods for rock stress estimation - Part 2: overcoring methods. International Journal of Rock Mechanics and Mining Sciences **40** (2003), 999–1010.

[3] Haimson, B. C., Cornet, F. H.: ISRM suggested methods for rock stress estimation - Part 3: hydraulic fracturing (HF) and/or hydraulic testing of pre-existing fractures (HTPF). International Journal of Rock Mechanics and Mining Sciences **40** (2003), 1011–1020.

[4] Wiles, T. D., Kaiser, P. K.: In situ stress determination using the under-excavation technique-I. Theory. International Journal of Rock Mechanics and Mining Sciences **31** (5) (1994), 439–446.

[5] Wiles, T. D., Kaiser, P. K.: In situ stress determination using the under-excavation technique-II. Applications. International Journal of Rock Mechanics and Mining Sciences **31** (5) (1994), 447–456.

[6] Sugawara, K., Obara, Y.: Draft ISRM suggested method for in situ stress measurement using the compact conical-ended borehole overcoring (CCBO)technique. International Journal of Rock Mechanics and Mining Sciences **36** (1999), 307–322.

[7] Nečas, J., Hlaváček, I.: *Mathematical Theory of Elastic and Elasto-plastic Bodies: An Introduction*. Elsevier, 1981.

[8] Malík, J., Kolcun, A.: Determination of initial stress tensor from deformation of underground opening in excavation process. Appl. Math., accepted, available on line.

[9] Lax, P. D.: *Linear Algebra and Its Applications*. Wiley, 2007.

[10] Blaheta, R., Jakl, O., Kohut, R., Starý, J.: GEM – a platform for advanced mathematical geosimulations. In: R. Wyrzykowski (Ed.), *Proceedings of Parallel Processing and Applied Mathematics, Lecture Notes in Computer Science*, pp. 66–275. Springer-Verlag, 2010.

[11] Souček, K. et al.: *Comprehensive geological characterization of URF Bukov-part II Geotechnical characterization*. Final report 221/2018. SÚRAO, IG CAS, 2017, 109–121.

Programs and Algorithms of Numerical Mathematics 21
J. Chleboun, P. Kůs, J. Papež, M. Rozložník, K. Segeth, J. Šístek (Eds.)
Institute of Mathematics CAS, Prague 2023

NUMERICAL OPTIMIZATION OF PARAMETERS IN SYSTEMS OF DIFFERENTIAL EQUATIONS

Josef Martínek, Václav Kučera

Faculty of Mathematics and Physics, Charles University
Sokolovská 83, Praha 8, 186 75, Czech Republic
martinek@karlin.mff.cuni.cz, kucera@karlin.mff.cuni.cz

Abstract: We present results on the estimation of unknown parameters in systems of ordinary differential equations in order to fit the output of models to real data. The numerical method is based on the nonlinear least squares problem along with the solution of sensitivity equations corresponding to the differential equations. We will present the performance of the method on the problem of fitting the output of basic compartmental epidemic models to data from the Covid-19 epidemic. This allows us to draw several conclusions on the natural limitations of these models and their validity.

Keywords: Ordinary differential equations, parameter estimation, nonlinear least squares, mathematical epidemiology

MSC: 65L05, 92D30, 65K10

1. Introduction

Ordinary differential equations (ODEs) are one of the most common mathematical tools to describe natural phenomena. Extensive literature exists on how to build more or less sophisticated mathematical models leading to ODEs. Typically the resulting equations contain unknown parameters (constants) which must be tailored to the specific application. These can be obtained by measurement, theoretical considerations, etc., but in certain situations it is difficult to come up even with a rough estimate of the real-life parameters of the model. One possibility then is to tune the parameters of the model so that its output agrees best with measured data. There are many approaches to solve such a data-fitting problem, cf. [6]. Here we build on the approach of [2] which uses so-called sensitivity equations to obtain the dependence of the solution of the ODEs on the considered parameters (Section 2). This is then used in a gradient-based Levenberg-Marquardt optimization algorithm which solves a nonlinear least-squares problem of fitting the output to the data (Section 3). We test the data-fitting algorithm on compartmental models from epidemiology (Section 4), specifically we take data from the COVID-19 epidemic in the Czech Republic (Section 5) and discuss the validity of such simple models.

DOI: 10.21136/panm.2022.12

2. Ordinary differential equations and sensitivity equations

We use the notion of a system of ODEs in the following way:

Definition 1. *Let* $n \in \mathbb{N}$, *and* $f_i \colon \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}$ *for* $i \in \{1, \dots, n\}$. *By a* system of differential equations *we mean a system of the form*

$$\begin{aligned} y_1' &= f_1(y_1, \dots, y_n, t), \\ y_2' &= f_2(y_1, \dots, y_n, t), \\ &\vdots \\ y_n' &= f_n(y_1, \dots, y_n, t). \end{aligned} \tag{1}$$

We use vector notation $y' = f(t, y(t))$ *for brevity. By an* initial value problem *we mean the system* (1) *along with a point* $(t_0, y^0) \in \mathbb{R} \times \mathbb{R}^n$ *called the* initial condition. *We seek a solution of the system of differential equations such that* $y(t_0) = y^0$.

Throughout this contribution, we consider the case when the system of ODEs (1) contains some known or unknown parameters, in which case the resulting solution also depends on the choice of the parameter. Specifically, instead of y being only a function of t, i.e. $y(t)$, we will have also the dependence on some parameter(s) c: hence we write $y(t, c)$. To simplify the notation, for some fixed value of the parameter c we will sometimes omit the second argument and write $y(t, c) = y(t)$. Similarly, we write $y'(t, c) = y'(t) = \frac{\partial y}{\partial t}(t, c)$ if the right-hand side is defined. This will simplify the notation for ODEs, where t is the relevant variable and c is only a parameter.

Now we follow the paper of Dickinson and Gelinas [2] and the monograph [6] by Schittkowski. Let us consider an initial value problem

$$y'(t, c) = f\big(y(t, c), t, c\big), \quad y(0, c) = y^0, \tag{2}$$

which depends on a real parameter c. For now we assume that the initial condition y^0 does not depend on c. In order to optimize the parameters in our models we need to determine the so called *sensitivity* of the system with respect to c.

Definition 2. *Let* $i \in \{1, \dots, n\}$. *We define the* sensitivity of the i-th variable with respect to the parameter c *by*

$$z_i(t, c) = \frac{\partial y_i}{\partial c}(t, c).$$

The sensitivities defined above can be obtained as a solution of a system of ODEs called the sensitivity equations which we derive now. Let $i \in \{1, \dots, n\}$. We assume that all functions involved are sufficiently smooth. Then we obtain by Definition 2 and the rule for interchanging the order of differentiation

$$\frac{\partial z_i}{\partial t}(t, c) = \frac{\partial}{\partial t}\left(\frac{\partial y_i}{\partial c}(t, c)\right) = \frac{\partial}{\partial c}\left(\frac{\partial y_i}{\partial t}(t, c)\right). \tag{3}$$

By using (2), the chain rule for differentiation and Definition 2, we have

$$\begin{aligned}\frac{\partial z_i}{\partial t}(t,c) &= \frac{\partial}{\partial c}\big[f_i\big(y_1(t,c),\dots,y_n(t,c),t,c\big)\big] \\ &= \frac{\partial f_i}{\partial c}\big(y_1,\dots,y_n,t,c\big) + \sum_{j=1}^{n}\frac{\partial f_i}{\partial y_j}\big(y_1,\dots,y_n,t,c\big)\frac{\partial y_j}{\partial c}(t,c) \\ &= \frac{\partial f_i}{\partial c}\big(y_1,\dots,y_n,t,c\big) + \sum_{j=1}^{n}\frac{\partial f_i}{\partial y_j}\big(y_1,\dots,y_n,t,c\big)\, z_j(t,c).\end{aligned} \tag{4}$$

We have obtained the so-called sensitivity equations. These are a system of n ODEs which can be solved simultaneously with the original system (2). We now determine the initial condition of the sensitivity equations. Since the initial condition of the original system (2) does not depend on the parameter c, we have by Definition 2:

$$z_i(0,c) = \frac{\partial y_i}{\partial c}(0,c) = \frac{\partial y_i^0}{\partial c} = 0. \tag{5}$$

Definition 3. *Let $y'(t,c) = f(y(t,c),t,c)$, $y(0,c) = y^0$ be an initial value problem of the form (2) and suppose that the initial condition does not depend on the parameter $c \in \mathbb{R}$. We define the corresponding* sensitivity equations *by*

$$z_i'(t,c) = \frac{\partial f_i}{\partial c}\big(y_1,\dots,y_n,t,c\big) + \sum_{j=1}^{n}\frac{\partial f_i}{\partial y_j}\big(y_1,\dots,y_n,t,c\big)\, z_j(t,c), \quad z_i(0,c) = 0,$$

for $i \in \{1,\dots,n\}$.

2.1. Multiple parameters and parameter in initial condition

The previous derivation generalizes straightforwardly to the case of multiple parameters (in the equation only), where we use the vector form $c = (c_1,\dots,c_m)^T \in \mathbb{R}^m$. We define the *sensitivity of the i-th variable with respect to the parameter c_j* by

$$z_i^j(t,c) = \frac{\partial y_i}{\partial c_j}(t,c).$$

Proceeding similarly as in the derivation in the previous case (4) we obtain the sensitivity equation for the sensitivity z_i^j in the form

$$(z_i^j)' = \frac{\partial f_i}{\partial c_j} + \sum_{k=1}^{n}\frac{\partial f_i}{\partial y_k} z_k^j,$$

along with the initial conditions $z_i^j(0,c) = 0$.

Until now we have discussed the case when the initial condition does not depend on c. However, a parameter may appear both in the equation and in the initial condition. Consider for example the following initial value problem:

$$y'(t,c) = f\big(y(t,c),t,c\big), \quad y(0,c) = (c, y_2^0,\dots,y_n^0)^T, \tag{6}$$

The sensitivity equations themselves are identical to those in Definition 3. As for the initial condition, for the first variable z_1 we have by Definition 2

$$z_1(0,c) = \frac{\partial y_1}{\partial c}(0,c) = \frac{\partial}{\partial c}c = 1$$

and for $i \in \{2,\dots,n\}$ we get $z_i(0,c) = 0$ as in the previous section.

3. Algorithms for parameter optimization

We now address the problem of optimizing the parameters in ODEs, i.e. finding the set of parameters for which the solution of the ODE has the best agreement with given data obtained e.g. from measurement or observation. There are many possibilities how to approach this problem, see [6]. Our approach is the following: The resulting function obtained as a solution to the considered model fits the measured data in the least squares sense. More precisely, consider the initial value problem (2) which depends on m parameters $c = (c_1,\dots,c_m)^T \in \mathbb{R}^m$. Suppose we have a set of data points $\{(t_j, Y^j) \in \mathbb{R}^{n+1},\ j = 0,\dots,M\}$. We want to find a vector of parameters $c_{\min} \in \mathbb{R}^m$ such that it satisfies the condition

$$c_{min} = \underset{c\in\mathbb{R}^m}{\operatorname{argmin}} \sum_{j=0}^{M} \|y(t_j,c) - Y^j\|^2 = \underset{c\in\mathbb{R}^m}{\operatorname{argmin}} \sum_{j=0}^{M}\sum_{i=1}^{n} \big(r_{ij}(c)\big)^2, \tag{7}$$

where $\|\cdot\|$ is the Euclidean norm in $\mathbb{R}^n$ and the *residuals* are defined by

$$r_{ij}(c) = y_i(t_j,c) - Y_i^j. \tag{8}$$

A minimization problem of the form (7) is called a *nonlinear least squares* problem. In the case when $r_{ij}(c)$ depend linearly on c, the problem reduces to (linear) least squares. Since we are typically unable to find analytic solutions to our ODEs, we cannot write the explicit formulae for $r_{ij}(c)$. However, one can solve the equations numerically, in our case by a fourth order Runge-Kutta method. Moreover, we can also compute the partial derivatives of the residuals w.r.t. the parameters:

$$\frac{\partial r_{ij}}{\partial c_k}(c) = \frac{\partial y_i}{\partial c_k}(t_j,c) = z_i^k(t_j,c), \quad k \in \{1,\dots,m\}.$$

We can therefore evaluate the partial derivatives of r_{ij} by solving the sensitivity equations (also using Runge-Kutta) in parallel with the original ODEs. This allows us to apply a gradient-based optimization algorithm for the numerical solution of problem (7). Specifically, we use the Levenberg-Marquardt method, which produces the a sequence of approximations to c_{min} using the iterative process

$$c^{(l+1)} = c^{(l)} - (J_l^T J_l + \lambda_l I)^{-1} J_l^T r(c^{(l)}), \quad l = 0,\dots,$$

where J_l is the Jacobi matrix of the residuals r_{ij} w.r.t. the parameter vector c at the l-th iteration. The constant λ_l is a 'damping' parameter which interpolates

between Gauss-Newton method ($\lambda_l = 0$) and steepest descent ($\lambda_l \to \infty$). There are various choices of λ_l, we adopted the simple strategy from the original paper of Marquardt [4], which proved sufficient in our case. The more basic method, Gauss-Newton's method, did not converge in several of our test cases or converged very locally probably due to the near-singularity of the Jacobi matrices. The Levenberg-Marquardt method can be viewed as Gauss-Newton using a trust region approach.

4. Compartmental epidemiological models

We will test the performance of the parameter optimization algorithm on systems of ODEs coming from mathematical biology, namely models for the spreading of infections diseases in a population. Mathematical models in epidemiology may be sorted into various categories according to different criteria – discretization of time (models with discrete intervals and continuous time models), allowing for randomness (stochastic and deterministic models), structure of the population etc. Here we take into account exclusively *deterministic*, *continuous time* models where the population is assumed to be a *homogeneous continuum*. Presumably the most widely known representatives of this kind of models are the standard compartmental models. These models are based on the principle of dividing the population into several labeled *compartments* (eg. Infectious, Recovered etc.) under certain simplifying assumptions. The development of the epidemic is then determined by relations describing the flow between compartments, namely the rate of flow between a pair of compartments. The model is formulated mathematically as a system of ODEs.

4.1. SIR model

The *SIR model* is the most basic compartmental model, cf. [5]. The population is divided into three groups, each group a function of time:

- *Susceptible* (S) — those who have not come across the disease and can fall ill if they come into contact with an infectious person, thus becoming infectious.
- *Infectious* (I) — those who spread the disease among the susceptible population. After recovery they move to the compartment R:
- *Recovered* (R) — those who are removed from the compartment I either due to recovery or due to death.

The relations between the compartments are based on four fundamental assumptions:

1. The vital dynamics is neglected and the size of the population is supposed to be constant, we denote it by $N > 0$.
2. The population is assumed to be a homogeneous continuum, i.e. all people have an equal number of contacts, the probability of the transmission of the disease between a susceptible and an infectious person during their contact remains constant and the infectious are equally distributed among the population.

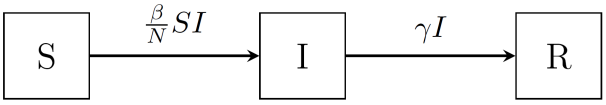


Figure 1: SIR model.

3. The rate of flow between the compartments I and R is directly proportional to the size of the compartment I.

4. The recovered acquire immunity and cannot spread the infection. Those who fall victims to the disease are treated as recovered.

Let r be the number of contacts of a person per unit time and let $p \in (0, 1)$ be the probability of the transmission between an infectious and a susceptible person when they meet. It is desired to find the number of people an infectious person infects per unit time. The fraction of susceptible population within the total population is $\frac{S}{N}$. Therefore, the infectious person meets a total of $r\frac{S}{N}$ susceptible people per unit time. It follows that the number of infected susceptible people per infectious person per unit time is $pr\frac{S}{N}$. It proves convenient to define a new constant $\beta = pr$. Because the total number of infectious people is equal to I, it can be concluded that the total number of people an infectious person infects per unit time is $\beta I\frac{S}{N}$.

We now determine the relation between compartments I and R. As stated in the assumption 2, the rate of flow between the compartments I and R is directly proportional to the size of the compartment I. Denote by γ the coefficient of proportionality. The rate of flow is then equal to γI. The value $\frac{1}{\gamma}$ can be interpreted as the expected time spent in the compartment I, cf. [5].

The resulting model is described mathematically by the system of ODEs

$$S' = -\frac{\beta}{N}SI, \quad I' = \frac{\beta}{N}SI - \gamma I, \quad R' = \gamma I. \tag{9}$$

The model is shown schematically in Figure 1. The system (9) is equipped with the following initial conditions. Let $I_0 > 0$ and $R_0 \geq 0$. We set

$$S(0) = N - R_0 - I_0, \quad I(0) = I_0, \quad R(0) = R_0. \tag{10}$$

There are many generalizations of the SIR model, usually based on the introduction of various other compartments. For example, the *SIQR* model is based on the additional assumption that every infectious subject is quarantined after the infection is detected. In addition to S, I, and R, we define a new compartment called *Quarantined* denoted Q. The infectious move from the compartment I to the compartment Q with a rate of flow directly proportional to the size of I. Analogously, the quarantined leave the compartment Q and move on to the compartment R with

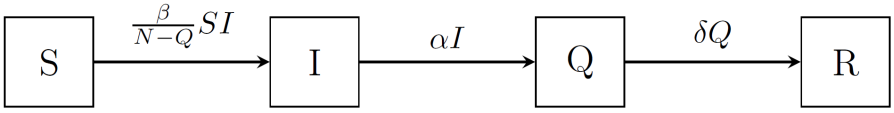


Figure 2: SIQR model.

a rate of flow directly proportional to the size of Q. The coefficients of proportionality are denoted by α and δ, respectively. Finally, we assume that the quarantined are unable to interact with the rest of the population. Thus the flow rate between S and I in the SIR model has to be modified appropriately. The model is shown schematically in Figure 2. The resulting system of ODEs reads

$$S' = -\frac{\beta}{N-Q}SI, \quad I' = \frac{\beta}{N-Q}SI - \alpha I, \quad Q' = \alpha I - \delta Q, \quad R' = \delta Q. \tag{11}$$

Apart from SIR and SIQR, we considered several other variants, such as the SEIR and SEIQR models and a different version of the SIQR model. Here E stands for *Exposed*, this compartment contains infected people who are not infectious yet, effectively adding a latency period to the standard model. We only mention these models in passing, since they gave us results almost identical with the basic SIR model on the considered data and thus present no added value in our case.

5. Numerical results

The approach to parameter optimization described in Sections 2 and 3 was implemented in MATLAB and tested on COVID-19 epidemiological data from the Czech Republic using the models from Section 4. However first we have tested the algorithms on artificially generated data and, more interestingly, on a standard test-case of data from a well studied and documented local influenza epidemic.

5.1. Influenza epidemic in a boarding school

The SIR model is derived under certain assumptions on the population and the disease. This may significantly affect the accuracy of the model in practice. We present here one case, which is as close as possible to satisfying the assumptions, the case of an influenza outbreak in an English boarding school from 1978, cf. [5].

In total, 763 boys were present, one boy had an influenza-like illness from the A/USSR/90/77(H1N1) virus. Over the next two weeks, a total of 512 boys developed similar symptoms spending between three and seven days in the college infirmary. We want to estimate the values of parameters β and γ from the SIR model (9) corresponding to this epidemic. The population remains constant over the whole period, i.e. $N = 763$. Contacts of the pupils were limited to the people in school, thus forming a closed community – it seems that the population is as homogeneous as possible. The presymptomatic period is short, no deaths occurred and the recovered

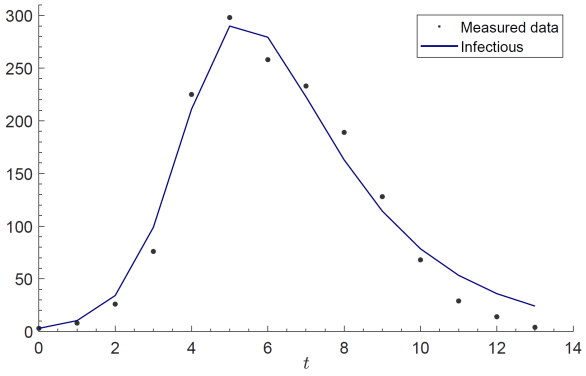


Figure 3: Measured data of the flu epidemic and the estimate of compartment I.

acquired sufficient immunity. One problem concerning the available data may occur, since in practical cases we do not possess the data which fit into the structure of the SIR model precisely. The data consists of the number of students confined to bed each day. Following [5], we assume the data to be from the compartment I.

Since we have data only from compartment I, we define the optimization problem of the form: Find β_m, γ_m satisfying

$$(\beta_m, \gamma_m)^T = \underset{(\tilde{\beta},\tilde{\gamma})^T \in \mathbb{R}^2}{\operatorname{argmin}} \sum_{j=0}^{D} |\tilde{I}(j\tau, \tilde{\beta}, \tilde{\gamma}) - I^j|^2, \tag{12}$$

where $\tau = 1$ corresponds to one day, which is the period with which we know the number of infected, I^j on the j-th day, $j = 0, \ldots, D$ with $D = 13$. The initial estimate is given by $(\beta^{(0)}, \gamma^{(0)})^T = (1, \frac{1}{7})^T$ and the stopping criterion

$$\|(\beta_m, \gamma_m)^T - (\beta_{m+1}, \gamma_{m+1})^T\|_\infty < 10^{-5}, \tag{13}$$

was satisfied after six iterations of the Levenberg-Marquardt algorithm. The resulting estimate of the parameters is $(\beta_m, \gamma_m)^T \approx (1.6998, 0.4469)^T$. Figure 3 shows that the estimated values of the compartment I are in good agreement with the data. However, after closer examination we find that the results do not quite correspond to the available data. Namely, the SIR model with the optimized parameters shows that the total number of people who suffered from the illness is 744, whereas the true number was 512. In addition, the value $\frac{1}{\gamma} \approx 2.24$ represents the expected time (in days) one spends in the Infectious compartment. This value is less then the observed value, which was three to seven days. This suggests that even in this simple case some unexpected issues limiting the accuracy of the model occur. This is a consequence of several facts. As stated above, the available data do not fit the model precisely – a person diagnosed with the illness has limited possibilities of spreading the disease because their contacts with the susceptible population are restricted. In addition, the pattern of the SIR model may not be entirely convenient for this particular disease. In order to adjust the model in accordance with the disease we need additional medical information which is not available.

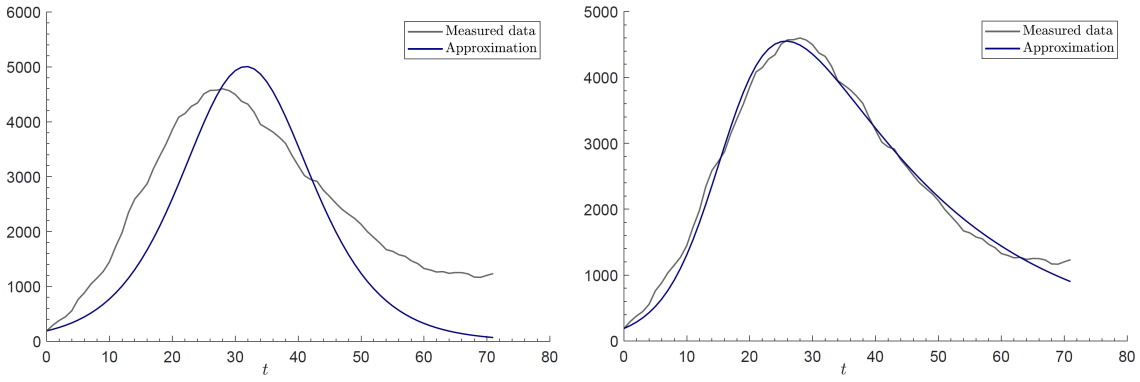


Figure 4: COVID-19, Infectious: Full population (left), effective population (right)

5.2. COVID-19 epidemic in the Czech Republic

Finally, we apply the presented numerical methods to the COVID-19 epidemiological data from the Czech Republic provided by the Ministry of Health of the Czech republic [3]. We chose the period from March 13, 2020, to May 24, 2020. The reasons are the following: On 13 March, the key measure forbidding retail sales and the sales of services in business premises came into effect and on 25 May the crucial part of the restrictive measures ended. It is therefore reasonable to assume that β and γ remain constant within this period, since adopting some restrictive measures against the spread of the disease decreases the value of parameter β, because the number of contacts of a person is reduced. The chosen period was the longest during the epidemic, where external conditions remained the same.

We optimized the parameters β and γ using the data from the compartment I only, i.e. the function to minimize is of the form (12) with $D = 71$ and $N = 1.065{\cdot}10^7$. The initial guess of the parameters is again given by $(\beta^{(0)}, \gamma^{(0)})^T = (1, 1)^T$. The stopping criterion (13) was achieved after 10 iterations. The computed estimate is $(\beta_m, \gamma_m)^T \approx (4.6687, 4.5244)^T$. The results from compartment I can be seen in Figure 4 (left). We note that the computed estimate gives the expected time a person remains infectious $\frac{1}{\gamma_m} \approx 0.22$ days, which is clearly unrealistic. Moreover, the model shows that the total number of recovered people at the and of the considered time interval is $6.15 \cdot 10^5$, while the actual value was 7750.

The reason why the SIR model gives such unrealistic results for the presented data is that the number of infected was very small in proportion to the total population of the Czech Republic and the population was not homogeneous, since the epidemic consisted of small local outbreaks, thus violating one of the basic assumptions of the SIR model. This consideration leads us to the introduction of an *effective population size*. The idea is to use a reduced population size which reflects the assumption of homogeneity within that smaller sub-population. The question is how to determine the size of the effective population. Our approach is to consider N not as a fixed constant (as it has been until now), but to treat it as an unknown parameter. For-

mally, the change is that instead of the parameter vector $(\beta, \gamma)^T$ for the SIR model, we now have the extended parameter vector $(\beta, \gamma, N)^T$. We note that N is present not only in the equations (9), but also in the initial condition (10), thus we use the approach from Section 2.1.

The initial guess was $(\beta^{(0)}, \gamma^{(0)}, N^{(0)})^T = (1, 1, 10^6)^T$. The computed results are $(\beta_m, \gamma_m, N_m)^T \approx (0.2587, 0.0444, 8593)$ and were reached after 50 iterations. Agreement with measured data has improved, cf. Figure 4 (right). The estimated total number of recovered is 7636, which is a good approximation of the true value 7750. The expected length of the infectious period is approximately 22 days. This is close to the length of the potential maximal infectious period of 15 to 21 days estimated in meta-analysis [1]. The estimate of the basic reproduction number $R_0 = \frac{\beta_m}{\gamma_m} \approx 5.8$ exceeds the values in the interval 2.4 to 3.4 estimated by meta-analysis.

To conclude, the presented method of the effective population considerably increased the accuracy of the basic SIR model in the situation when the SIR model itself failed due to high inconsistency of the measured data with the assumptions of the model. We have also tried other compartmental models such as the SIQR model, however not much improvement was observed over the basic SIR model with optimization of the effective population size.

Acknowledgements

This work was supported by grant No. 20-01074S of the Czech Science Foundation.

References

[1] Byrne, A.W. et al.: Inferred duration of infectious period of SARS-CoV-2: rapid scoping review and analysis of available evidence for asymptomatic and symptomatic COVID-19 cases. BMJ Open, **10**(8) (2020).

[2] Dickinson, R.P. and Gelinas, R.J.: Sensitivity analysis of ordinary differential equation systems - a direct method. J. Comput. Phys., **21** (1976), 123–143.

[3] Komenda, M. et al.: COVID-19: Přehled aktuální situace v ČR, onemocnění aktuálně, https://onemocneni-aktualne.mzcr.cz/covid-19, accessed 30-11-2022.

[4] Marquardt, D.W.: An algorithm for least-squares estimation of nonlinear parameters. J. Soc. Ind. Appl. Math., **11**(2) (1963): 431–441.

[5] Murray, J.D.: *Mathematical Biology I. An Introduction*, Springer, New York, 2002.

[6] Schittkowski, K.: *Numerical Data Fitting in Dynamical Systems: A Practical Introduction with Applications and Software*. Kluwer Academic Publishers, 2002.

Programs and Algorithms of Numerical Mathematics 21
J. Chleboun, P. Kůs, J. Papež, M. Rozložník, K. Segeth, J. Šístek (Eds.)
Institute of Mathematics CAS, Prague 2023

NEW METHODS IN COLLISION OF BODIES ANALYSIS

Ivan Němec[1,3], Jiří Vala[2], Hynek Štekbauer[1,3], Michal Jedlička[1,3], Daniel Burkart[3]

[1] Brno University in Technology, Faculty of Civil Engineering, Institute of Structural Mechanics, 602 00 Brno, Veveří 95, Czech Republic
nemec.i@fce.vutbr.cz, stekbauer.h@fce.vutbr.cz, jedlicka.m@fce.vutbr.cz

[2] Brno University in Technology, Faculty of Civil Engineering, Institute of Mathematics and Descriptive Geometry, 602 00 Brno, Veveří 95, Czech Republic
vala.j@fce.vutbr.cz

[3] FEM Consulting Ltd., 602 00 Brno, Veveří 95, Czech Republic
nemec@fem.cz, stekbauer@fem.cz, jedlicka@fem.cz, Daniel.Burkart@vut.cz

Abstract: The widely used method for solution of impacts of bodies, called the penalty method, is based on the contact force proportional to the length of the interpenetration of bodies. This method is regarded as unsatisfactory by the authors of this contribution, because of an inaccurate fulfillment of the energy conservation law and violation of the natural demand of impenetrability of bodies. Two non-traditional methods for the solution of impacts of bodies satisfy these demands exactly, or approximately, but much better than the penalty method. Namely the energy method exactly satisfies the conservation of energy law, whereas the kinematic method exactly satisfies the condition of impenetrability of bodies. Both these methods are superior in comparison with the penalty method, which is demonstrated by the results of several numerical examples.

Keywords: contact / impact of elastic bodies, finite element method, method of dicretization in time, energy and kinematic approaches.

MSC: 74M15, 74S05, 74S20

1. Introduction

Robust, reliable and effective computational analysis of collision of deformable bodies belongs to the important tasks of engineering mechanics, conditioned by the successful cooperation in formulation of physical models with reasonable parameters, evaluable from rather simple experiments, in mathematical and numerical analysis and in software development. Models based on the conservation principles of classical thermomechanics by [4], supplied by appropriate constitutive relations, lead

DOI: 10.21136/panm.2022.13

to partial differential equations or even to their systems of hyperbolic type, supplied with the pair of Cauchy initial conditions for displacements and their rates, with the Dirichlet boundary conditions for prescribed supports and with the Neumann boundary conditions for exterior loads, together with the contacts / impacts of colliding deformable bodies. This brings substantial nonlinearities to the system, even under the hypothetical, not very realistic assumptions on both the geometrical linearity (small strains) and the physical one (linear reversible strain - stress relations). Theoretical formulations containing variational inequalities, after the discretisation both in the time and in the Euclidean space, 3-dimensional in general, replace their exact fulfilment by the introduction of some additional penalty terms, as introduced by [23]. Other serious problems are the incorporation of contact friction, non-expensive search for potential contacts – cf. the distributed and parallel computations required by [6] and [17], as well as the description of contact geometry, characterized as node-to-node, node-to-segment or segment-to-segment approaches.

The progress in this research area in more than last 3 decades can be traced from the review articles [2], [7], [10] and [18]. When treating contact problems within the finite element method, 7 steps of analysis should be followed by [22]: i) continuum based contact kinematics, ii) constitutive equations for contact interfaces, iii) weak form of contact contributions and overall solution strategies for contact problems, iv) discretisation of contact surfaces, v) algorithms for the integration of constitutive equations in the contact area, vi) contact search algorithms, vii) adaptive methods for contact problems. The detailed primal and dual variational formulations of contact problems are demonstrated by [15]. The comparison of classical Lagrange multiplier and penalty computational approaches is presented by [20]. The classical recommendations for the choice of penalty stiffness, needed for the evaluation of the contact force proportional to the length of the interpenetration of bodies, are presented in [3]; the so-called exact penalty improvement, working with the updated penalty stiffness, was suggested by [13].

Other alternatives can be found in literature, too, as i) energy conserving algorithms, introduced by [9], revisited by [24], applying certain penalty-based regularization, or ii) perturbed Lagrangian formulations, stemming from [14], working with certain kinematic conditions, developed by [12]. The implementation of b) by [1] utilizes augmented Lagrange multipliers to force all prescribed kinematic conditions, which requires an additional iterative solutions of systems of algebraic equations, unwelcome for explicit time discretisation. Another implementation of b) by [21] avoids such iterative process, but leads to a non-physical increase of energy at contact / impact interfaces typically, which must be suppressed by some artificial computational reduction of contact forces.

Two promising computational methods, presented in this paper, can be seen as certain variants of i) and ii). We shall refer to them as to i) the energy method and to ii) the kinematic method, although such nomenclature is not quite unified in the literature, to highlight i) the exact energy conservation, or ii) the exact fulfillment of kinematic conditions, involved in any space- and time-discretised computational

scheme. Assuming the space discretisation using the finite element technique, we shall work with the explicit time integration, due to the need of short time steps, forced by collision phenomena. The approach i) generalizes the 2-dimensional formulation of [16] naturally. The approach ii) here does not evaluate any contact forces as separate variables, but its specific use in the explicit time stepping forces the correction of nodal displacement at all potential interfaces under the assumptions of a) impenetrability of colliding bodies, b) evaluation of exact collision time t_c, c) decomposition of any time step of length $\mathcal{D}t$, considered as $[0, \mathcal{D}t]$ for simplicity, to $[0, t_c]$ and $[t_c, \mathcal{D}t]$, d) conservation of momentum of contact entities and e) perfectly inelastic collision.

After this introductory remarks (*Section* 1) we shall come to the general discussion of collision of bodies (*Section* 2), to the energy method (*Section* 3) and to the kinematic method (*Section* 4), supported by some illustrative examples (*Section* 5). The brief concluding remarks (*Section* 6) will be oriented to the need and priorities of further research.

2. Collisions of bodies

To demonstrate the advantages of 2 announced methods, we shall consider a finite number of deformable bodies discretised into finite elements, with the surfaces consisting of flat triangles, whereas mass is assigned to nodes. These bodies can arbitrarily collide.

2.1. Finding the time and space coordinates of the collision

For simplicity, we shall suppose that each line consists only of straight elements and each surface, or a boundary of a solid, is decomposed to triangles. That being the case, following the node-to-segment approach, just two kinds of collision can occur: collision of two line segments (element edges), or collision of a node and a triangle surface segment, as shown by Fig. 1 schematically. All parameters of collision will be evaluated using the explicit method, with sufficiently small time steps. In each such time step, constant velocities and geometric linearity are assumed for finding the time and position of the contact of discretised bodies.

Let us have a line segment with its end points A, B and another line segment with

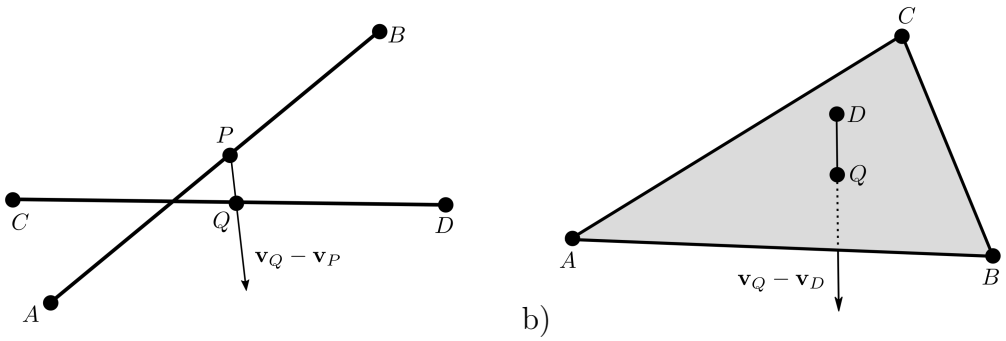


Figure 1: Two possibilities of collision of discretised bodies: a) edge to edge and b) node to surface.

its end points C, D, or, alternatively, a triangle surface element with its nodes A, B, C and another node D. At the beginning of present time step, $t = 0$ for simplicity, all these 4 points have their initial positions given by vectors $\mathbf{x}_i(0)$, and in such time step they move by velocities $\mathbf{v}_i$, assumed as constant during the whole time step. Thus, in any positive time t we come to positions

$$\mathbf{x}_i(t) = \mathbf{x}_i(0) + t\mathbf{v}_i \text{ for all } i \in \{A, B, C, D\}\,. \tag{1}$$

We need to find out, whether a) the line segments AB and CD, or b) the node D and the surface triangle ABC, will collide in the considered time step. Let P be the point of collision in the case a) and Q such point in the case b). For b) Q will be the point of the triangle ABC hit by the point D. At the collision time t_c all nodes A, B, C, D must lie in the same plane, thus for a known collision time their position can be determined, It can be also detected whether the points of the collision lie inside the pertinent segments, i. e. a) if the points P, Q lie inside the line segments AB and CD, or b) the point Q lies inside the triangle ABC. For the sake of brevity of the following formulae (2), (3) and (4), we shall write x_i instead of $x_i(t_c)$ now.

At first let us investigate whether the point D lies in the plane given by the points A, B, C. The symbols $\times$ and $\cdot$ will be reserved for the vector and scalar products in the 3-dimensional real Euclidean space. The normal vector to this plane can be then defined as $(\mathbf{x}_B(t_c) - \mathbf{x}_A(t_c)) \times (\mathbf{x}_C(t_c) - \mathbf{x}_A(t_c))$. If the point D lies in the plane ABC, then the vector connecting him with an arbitrary point of this plane, as with A in particular, must be perpendicular to the above introduced normal one; this can be written as

$$(\mathbf{x}_D - \mathbf{x}_A) \cdot ((\mathbf{x}_B - \mathbf{x}_A) \times (\mathbf{x}_C - \mathbf{x}_A)) = 0\,; \tag{2}$$

this cannot hold for any point D not belonging to the plane ABC for any non-degenerated triangle ABC, i. e. a triangle with non-zero area. Rearranging (2) formally, we obtain

$$\mathbf{x}_D \cdot (\mathbf{x}_A \times \mathbf{x}_B + \mathbf{x}_B \times \mathbf{x}_C + \mathbf{x}_C \times \mathbf{x}_A) = \mathbf{x}_A \cdot (\mathbf{x}_B \times \mathbf{x}_C)\,. \tag{3}$$

Substituting (1) with $t = t_c$ into (3), we come to the cubic equation

$$\begin{aligned}
&\mathcal{C}_3 t_c^3 + \mathcal{C}_2 t_c^2 + \mathcal{C}_1 t_c + \mathcal{C}_0 = 0\,, \qquad (4)\\
\mathcal{C}_0 \;&=\; \mathbf{x}_D \cdot (\mathbf{x}_A \times \mathbf{x}_B) + \mathbf{x}_D \cdot (\mathbf{x}_B \times \mathbf{x}_C) + \mathbf{x}_D \cdot (\mathbf{x}_C \times \mathbf{x}_A) - \mathbf{x}_A \cdot (\mathbf{x}_B \times \mathbf{x}_C)\,,\\
\mathcal{C}_1 \;&=\; \mathbf{x}_D \cdot (\mathbf{x}_A \times \mathbf{v}_B) + \mathbf{x}_D \cdot (\mathbf{v}_A \times \mathbf{x}_B) + \mathbf{v}_D \cdot (\mathbf{x}_A \times \mathbf{x}_B) + \mathbf{x}_D \cdot (\mathbf{x}_B \times \mathbf{v}_C)\\
&\quad+\; \mathbf{x}_D \cdot (\mathbf{v}_A \times \mathbf{x}_C) + \mathbf{v}_D \cdot (\mathbf{x}_B \times \mathbf{x}_C) + \mathbf{x}_D \cdot (\mathbf{x}_C \times \mathbf{v}_A) + \mathbf{x}_D \cdot (\mathbf{v}_C \times \mathbf{x}_A)\\
&\quad+\; \mathbf{v}_D \cdot (\mathbf{x}_C \times \mathbf{x}_A) - \mathbf{x}_A \cdot (\mathbf{x}_B \times \mathbf{v}_C) - \mathbf{x}_A \cdot (\mathbf{v}_B \times \mathbf{x}_C) - \mathbf{v}_A \cdot (\mathbf{x}_B \times \mathbf{x}_C)\,,\\
\mathcal{C}_2 \;&=\; \mathbf{v}_D \cdot (\mathbf{v}_A \times \mathbf{x}_B) + \mathbf{v}_D \cdot (\mathbf{x}_A \times \mathbf{v}_B) + \mathbf{x}_D \cdot (\mathbf{v}_A \times \mathbf{v}_B) + \mathbf{v}_D \cdot (\mathbf{v}_B \times \mathbf{x}_C)\\
&\quad+\; \mathbf{v}_D \cdot (\mathbf{x}_A \times \mathbf{v}_C) + \mathbf{x}_D \cdot (\mathbf{v}_B \times \mathbf{v}_C) + \mathbf{v}_D \cdot (\mathbf{v}_C \times \mathbf{x}_A) + \mathbf{v}_D \cdot (\mathbf{x}_C \times \mathbf{v}_A)\\
&\quad+\; \mathbf{x}_D \cdot (\mathbf{v}_C \times \mathbf{v}_A) - \mathbf{v}_A \cdot (\mathbf{v}_B \times \mathbf{x}_C) - \mathbf{v}_A \cdot (\mathbf{x}_B \times \mathbf{v}_C) - \mathbf{x}_A \cdot (\mathbf{v}_B \times \mathbf{v}_C)\,,\\
\mathcal{C}_3 \;&=\; \mathbf{v}_D \cdot (\mathbf{v}_A \times \mathbf{v}_B) + \mathbf{v}_D \cdot (\mathbf{v}_B \times \mathbf{v}_C) + \mathbf{v}_D \cdot (\mathbf{v}_C \times \mathbf{v}_A) - \mathbf{v}_A \cdot (\mathbf{v}_B \times \mathbf{v}_C)\,.
\end{aligned}$$

Clearly (4) can be solved analytically by the Cardano formulae, or iteratively, using e. g. the Newton method. If its positive solution t_c exists, not exceeding the time step $\mathcal{D}t$, it refers to the collision time; in the case of multiple solutions the smallest one corresponds to the needed first collision time.

Its smallest positive solution t_c (if exists, not exceeding the time step $\mathcal{D}t$) refers to the first collision time. Consequently, all position vectors $\mathbf{x}_i(t_c)$ for the points $i \in \{A, B, C, D\}$ can be evaluated by (1).

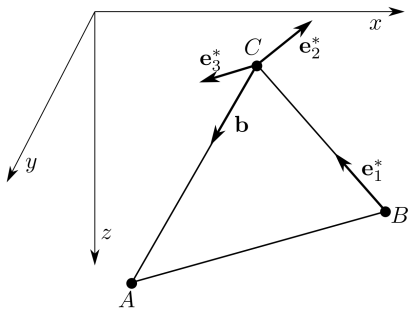


Figure 2: Definition of the contact plane by the nodes A, B, C.

2.2. Determination of the contact plane and its properties

Let us notice that the determination of contact time and location is based on the existence of a plane containing all points A, B, C, D. Such plane, as sketched by Fig. 2, is determined by an arbitrary triple selected from these 3 points, e. g. A, B, C for simplicity. Let us introduce a local coordinate system, whose 2 basis vectors $\mathbf{e}_1, \mathbf{e}_2$ can be chosen as arbitrary orthogonal vectors in this plane, whereas the remaining basis vector $\mathbf{e}_3$ is normal to this plane, with an appropriate orientation to satisfy $\mathbf{e}_1 \cdot (\mathbf{e}_2 \times \mathbf{e}_3) > 0$; thus we have a new coordinate system $\mathbf{x}^* = x_1^*\mathbf{e}_1 + x_2^*\mathbf{e}_2 + x_3^*\mathbf{e}_3$. In particular, we can introduce the unit vectors $\mathbf{e}_1^* = (\mathbf{x}_C - \mathbf{x}_B)/|\mathbf{x}_C - \mathbf{x}_B|$, $\mathbf{b} = (\mathbf{x}_A - \mathbf{x}_C)/|\mathbf{x}_A - \mathbf{x}_C|$, $\mathbf{e}_3 = \mathbf{b} \times \mathbf{e}_1^*$ and $\mathbf{e}_2^* = \mathbf{e}_3^* \times \mathbf{e}_1^*$. Thus we can work the local transform of coordinates $\mathbf{x}^* = \mathbf{R}\mathbf{x}$, containing certain rotation matrix $\mathbf{R} = (\mathbf{e}_1^{*\mathrm{T}}, \mathbf{e}_2^{*\mathrm{T}}, \mathbf{e}_3^{*\mathrm{T}})^{\mathrm{T}}$. Clearly $x_3^* = 0$ only for all points lying in the contact plane, whereas 2 remaining axes create a contact plane coordinate system, needed in our following considerations.

The contact plane can be used for definition of arbitrary friction models. Here, due to the limited extent of this paper, let us introduce a very simple property of such contact plane, which can be characterized as "the elastic friction", based on the introduction of 2 limit cases. The 1st one can be called "the zero friction", which means that both surfaces of elastic bodies are perfectly slippery, so no in-plane contact force and no friction dissipation can occur. The 2nd one can be called "the absolute friction", which means that no mutual sliding can occur during the collision, thus, no dissipation occurs in this case as well. More general cases of the elastic friction can be received as linear combinations of these 2 limit cases. Potential dissipation could be easily put into the energy balance in the energy method.

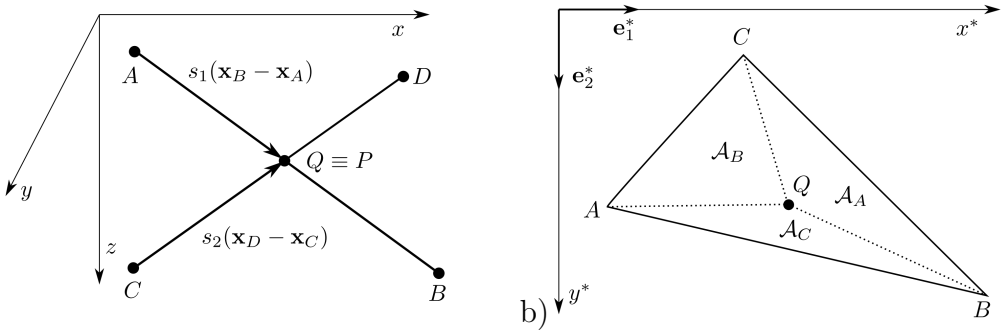


Figure 3: Collision of: a) two line segments or edges, or b) a node and a triangle surface.

2.3. Determination of the positions and velocities of the colliding points

To be able to determine the positions and velocities of all colliding points, we shall discuss only collisions of a) two segments or edges and b) a node and a triangle surface, as sketched by Fig. 3, in details. Collisions of the types node-to-node, or segment-to-segment (rarely exact) can be derived from a), b) using the limit passage. For the evaluation of the contact force direction, it is necessary to accept a suitable hypothesis. Let P, Q be the colliding points; thus, for the case of a collision of a node and a general point of a surface segment let us assume $Q \equiv D$.

2.4. Collision of two segments or edges

In the case a) positions of the colliding points P, Q can be determined as the intersection points of 2 lines, whose equations for real parameters s_1, s_2 are $\mathbf{x}_P(s_1) = \mathbf{x}_A + s_1(\mathbf{x}_B - \mathbf{x}_A)$, $\mathbf{x}_Q(s_2) = \mathbf{x}_C + s_2(\mathbf{x}_D - \mathbf{x}_C)$, and, moreover, $\mathbf{x}_P(s_1) = \mathbf{x}_Q(s_2)$ is required, thus

$$\mathbf{x}_A + s_1(\mathbf{x}_B - \mathbf{x}_A) = \mathbf{x}_C + s_2(\mathbf{x}_D - \mathbf{x}_C)\,. \tag{5}$$

For the evaluation of 2 parameters s_1, s_2 we have 3 equations (5) now; arbitrary 2 of them are sufficient for practical computations. Taking only the line segments AB, CD into account, unlike the whole lines, we have $s_1 \in [0,1] \Rightarrow P \in AB$, $s_2 \in [0,1] \Rightarrow Q \in CD$ evidently. Consequently, we can write $\mathbf{x}_P(t_c) = \mathbf{x}_Q(t_c)$ and

$$\begin{aligned} \mathbf{x}_P &= \mathbf{x}_A N_A + \mathbf{x}_B N_B\,, & \mathbf{x}_Q &= \mathbf{x}_C N_C + \mathbf{x}_D N_D\,, \\ \mathbf{v}_P &= \mathbf{v}_A N_A + \mathbf{v}_B N_B\,, & \mathbf{v}_Q &= \mathbf{v}_C N_C + \mathbf{v}_D N_D\,, \end{aligned} \tag{6}$$

taking $N_A = s_1$, $N_B = 1 - s_1$, $N_C = s_2$, $N_D = 1 - s_2$.

2.5. Collision of a node and a triangle surface element

Coming back to the case b), we have $Q = D$, thus $\mathbf{x}_Q(t_c) = \mathbf{x}_D(t_c)$. Here we need the usual area coordinates $N_j = \mathcal{A}_j/\mathcal{A}$ for $j \in \{A, B, C\}$ where the areas $\mathcal{A}_j$ are evident from Fig. 3 and $\mathcal{A}$ is their sum. If any of $N_j < 0$, the pertinent point does not lie in the triangle ABC. The vectors $\mathbf{x}_Q$ and $\mathbf{v}_Q$ in the plane ABC can be obtained in the terms of coordinates N_j as

$$\mathbf{x}_Q = \mathbf{x}_j N_j\,, \qquad \mathbf{v}_Q = \mathbf{v}_j N_j\,. \tag{7}$$

3. Energy method

Let us suppose that P, Q are the colliding points, as in the case a). The similar details of the case b) are left to the curious reader. For the simple implementation of friction, let us consider two limit cases, announced by *Section* 2. We shall work with the mutual velocity of the points P, Q, denoted as $\mathbf{v}_{PQ}(t) = v_Q(t) - v_P(t)$, and with the contact force $\mathbf{f}_{PQ}$ in this case, which can be interpreted as an internal force, acting by its components $\mathbf{f}_P$ and $\mathbf{f}_Q$ in sense of the 3rd Newton law. Their upper indices ^{a}, ^{z} will refer to the absolute friction, or to the zero friction, respectively.

3.1. Absolute friction

At first let us assume that the friction is absolute, without any slippage between the collision points P, Q during the contact at $t = t_c$. Thus, these points will bounce in the same relative direction $\mathbf{v}_{PQ}$ as before the collision; $\mathbf{f}_P^a = -\mathbf{f}_Q^a$ evidently. The unit vector in the direction of $\mathbf{f}_Q$, needed in the following considerations, can be introduced as $\mathbf{e}_Q^a = \mathbf{f}_Q^a/|\mathbf{f}_Q^a| = \mathbf{v}_{PQ}(t_c)/|\mathbf{v}_{PQ}(t_c)|$.

3.2. Zero friction

In this case, the direction of the contact force is in the direction $\mathbf{n}$ perpendicular to the plane given by the triangle ABC; this can be extended from b) to a) naturally, without all details here. Thus we have $\mathbf{n} = (\mathbf{x}_B - \mathbf{x}_A) \times (\mathbf{x}_D - \mathbf{x}_C)$. We are allowed to introduce $\mathbf{n}_Q$ using the relations $\mathbf{n} \cdot \mathbf{v}_{PQ} \geq 0 \Rightarrow \mathbf{n}_Q = \mathbf{n}$, $\mathbf{n} \cdot \mathbf{v}_{PQ} \leq 0 \Rightarrow \mathbf{n}_Q = -\mathbf{n}$. Finally, we can evaluate, similarly to $\mathbf{e}_Q^a$, $\mathbf{e}_Q^a = \mathbf{f}_Q^z/|\mathbf{f}_Q^z| = \mathbf{n}_Q/|\mathbf{n}_Q|$.

3.3. General friction

The simplest way for the interpolation between 2 preceding cases is to consider $\mathbf{e}_Q = \beta\mathbf{e}_Q^a + (1-\beta)\mathbf{e}_Q^z$, working with certain friction coefficient $\beta \in [0, 1]$. Consequently, we can write $\mathbf{e}_Q = \mathbf{f}_Q/|\mathbf{f}_Q| = \mathbf{e}_Q/|\mathbf{e}_Q|$, $|\mathbf{f}_Q| = |\mathbf{f}_P|$, $\mathbf{f}_Q = -\mathbf{f}_P$.

3.4. Determination of the magnitude of the contact force

The direction of the contact force is already known; its magnitude remains to be determined. We shall assume that the above introduced forces cause such accelerations of the nodes A, B, C, D that the velocities and positions of these nodes at the end of certain fictious time step $\mathcal{D}t$ conserve the total potential energy Π, decreased by dissipation caused by plasticizing or damage due to the collision, i. e. its new value can be expressed as $\Pi^{\times}(\mathbf{f}_{PQ}, \mathcal{D}t) = \Pi - \mathfrak{E}$ where $\mathfrak{E}$ denotes the dissipated energy, coming from additional considerations about irreversible plastic strains, fracture, etc. The acceleration of the point P, caused by the force $\mathbf{f}_P$, is $\mathbf{a}_P = \mathbf{f}_P/m_P$ where m_P is the mass assigned to the point P, which causes the velocity increment $\Delta\mathbf{v}_P = \mathbf{a}_P\mathcal{D}t$. For the point Q we can write $\mathbf{a}_Q = \mathbf{f}_Q/m_Q$, $\Delta\mathbf{v}_Q = \mathbf{a}_Q\mathcal{D}t$ similarly. Our distinguishing between the cases a) and b) will be useful in the following considerations. The obvious motivation is that the points P, Q are not the nodes in the original discretised system, therefore no discretised mass is assigned to them.

3.5. Collision of two line segments or edges

The substitution of $\mathbf{f}_P, \mathbf{f}_Q$ with $\mathbf{f}_A, \mathbf{f}_B, \mathbf{f}_C, \mathbf{f}_D$ can be done by the static equivalence of forces $\mathbf{f}_P = \mathbf{f}_A + \mathbf{f}_B$, $\mathbf{f}_Q = \mathbf{f}_C + \mathbf{f}_D$, together with $\mathbf{f}_A N_A + \mathbf{f}_B N_B = \mathbf{o}$, $\mathbf{f}_C N_C + \mathbf{f}_D N_D = \mathbf{o}$, coming from the equivalence of moments; $\mathbf{o}$ denotes the 3-dimensional zero vector. Solving this system of 4 linear algebraic equations, we obtain the formally simple relation

$$\mathbf{f}_i = N_i \mathbf{f}_P \text{ for any } i \in \{A, B, C, D\} \tag{8}$$

applying the coefficients N_i stemming from (6).

3.6. Collision of a node and a triangle surface segment

Since $Q = D$ in this case, the calculated values for the node D can be applied to the collision point Q, too, whereas for the collision point P the needed values of force, acceleration, velocity and position need to be expresses by the nodal values of the triangle ABC. Let us assume that all forces $\mathbf{f}_i$ for $i \in \{A, B, C\}$ are parallel, in the direction of the unit vector $\mathbf{e}_Q$. The static equivalence conditions then are $\mathbf{f}_P = \mathbf{f}_A + \mathbf{f}_B + \mathbf{f}_C$, $\mathbf{x}_P \times \mathbf{f}_P = \mathbf{x}_A \times \mathbf{f}_A + \mathbf{x}_B \times \mathbf{f}_B + \mathbf{x}_C \times \mathbf{f}_C$; moreover the identity condition $\mathbf{f}_Q = \mathbf{f}_D$ is valid. Consequently, expressing $\mathbf{f}_Q$ as $\mathbf{e}_Q|\mathbf{f}_Q|$, we come to $|\mathbf{f}_P| = |\mathbf{f}_A| + |\mathbf{f}_B| + |\mathbf{f}_C|$, $\mathbf{x}_P \times \mathbf{e}_Q|\mathbf{f}_P| = \mathbf{x}_A \times \mathbf{e}_Q|\mathbf{f}_A| + \mathbf{x}_B \times \mathbf{e}_Q|\mathbf{f}_B| + \mathbf{x}_C \times \mathbf{e}_Q|\mathbf{f}_C|$, $|\mathbf{f}_Q| = |\mathbf{f}_D|$. This implies (8) again, using the coefficients N_i from the text preceding (7).

3.7. Calculation of the change of the position

For any $i \in \{A, B, C, D\}$ the increments of velocities in the considered time step can be evaluated as $\Delta\mathbf{v}_i(\mathbf{f}_P, \mathcal{D}t) = (\mathbf{f}_i/m_i)\mathcal{D}t$, thus, with respect to (8), we receive $\mathbf{v}_i^{\times}(\mathbf{f}_P, \mathcal{D}t) = \mathbf{v}_i + \Delta\mathbf{v}_i$ in the form

$$\mathbf{v}_i^{\times}(\mathbf{f}_P, \mathcal{D}t) = \mathbf{v}_i + (N_i/m_i)\,\mathbf{f}_P \mathcal{D}t\,. \tag{9}$$

Consequently, since the increments of displacements can be expressed as $\Delta\mathbf{u}_i(\mathbf{f}_P, \mathcal{D}t) = \mathbf{v}_i \mathcal{D}t$ then $\mathbf{u}_i^{\times}(\mathbf{f}_P, \mathcal{D}t) = \mathbf{u}_i + \Delta\mathbf{u}_i$ gets the form

$$\mathbf{u}_i^{\times}(\mathbf{f}_P, \mathcal{D}t) = \mathbf{u}_i + \mathbf{v}_i \mathcal{D}t + (N_i/m_i)\,\mathbf{f}_P \mathcal{D}t^2\,. \tag{10}$$

Both (9) and (10) will be needed in the calculation of the change of energy for the contact forces $\mathbf{f}_P, \mathbf{f}_Q$ during the fictious time step $\mathcal{D}t$, which consists of 3 parts: i) the change of the kinetic energy $\Delta\Pi_k$, ii) the change of the elastic potential energy Π_σ and iii) the change of the potential energy of the position $\Delta\Pi_p$.

3.8. Calculation of the change of the kinetic energy

During the fictitious time step, only the velocities of the nodes A, B, C, D are influenced by the contact force. The kinetic energy of these mass points before and after the collision, i. e. at the beginning and at the end of the fictitious time step, using $i \in \{A, B, C, D\}$ as the Einstein summation index here, is $\Pi_k = (m_i/2)\,\mathbf{v}_i \cdot \mathbf{v}_i$, at its end $\Pi_k^{\times}(\mathbf{f}_P, \mathcal{D}t) = (m_i/2)\,\mathbf{v}_i^{\times} \cdot \mathbf{v}_i^{\times}$, thus, applying (9), for $\Delta\Pi_k(\mathbf{f}_P, \mathcal{D}t) = \Pi_k^{\times}(\mathbf{f}_P, \mathcal{D}t) - \Pi_k$ we have $\Delta\Pi_k^{\times}(\mathbf{f}_P, \mathcal{D}t) = (m_i/2)(2\mathbf{v}_i \cdot \Delta\mathbf{v}_i + \Delta\mathbf{v}_i \cdot \Delta\mathbf{v}_i)$, which yields

$$\Delta\Pi_k(\mathbf{f}_P, \mathcal{D}t) = N_i \mathbf{v}_i \mathbf{f}_P N_i \mathcal{D}t + (N_i^2/(2m_i))\,\mathbf{f}_P \cdot \mathbf{f}_P \mathcal{D}t^2\,. \tag{11}$$

3.9. Calculation of the change of the elastic potential energy

Since only the positions of nodes A, B, C, D will be influenced by the contact force, only the elastic energy of such j-th elements is relevant here.

$$\Delta\Pi_\sigma(\mathbf{f}_P, \mathcal{D}t) = \mathbf{f}_j^e \Delta\mathbf{d}_k^e \,; \tag{12}$$

here $\mathbf{f}_j^e$ and $\Delta\mathbf{d}_j^e$ form the vectors of the element nodal forces and of the element deformation parameters, derived using (10), respectively; e must be understood as an element index and j as an index referring to the above introduced list, both taken as the Einstein summation indices. Let us remind that $\Delta\mathbf{u}_i = \mathbf{v}_i\delta$ is satisfied for other nodes than $i \in \{A, B, C, D\}$, too.

3.10. Calculation of the change of the elastic potential energy

Since only the positions of the nodes will change in the (very short) fictitious time step, the change of the elastic potential energy $\Delta\Pi_\sigma(\mathbf{f}_P, \mathcal{D}t) = -\Delta u_i \mathbf{f}_i^{\text{ext}}$, $\mathbf{f}_i^{\text{ext}}$ being the components of external forces, can be formulated as

$$\Delta\Pi_p(\mathbf{f}_P, \mathcal{D}t) = (N_i/m_i)\, \mathbf{f}_P \cdot \mathbf{f}_i^{\text{ext}} \mathcal{D}t^2 \,. \tag{13}$$

3.11. Final evaluation of the magnitude of the contact force

The aim of this method is to satisfy the energy conservation law in collisions of bodies exactly. For all elastic bodies this means that the total energy after collision must remain the same as before the collision. To achieve this goal, it is necessary to adopt the equation of the energy conservation into the solution. The change of total energy during the collision must be zero. Since we have $\mathbf{f}_P = \mathbf{e}_p|\mathbf{f}_P|$ in all cases – cf. the discussion on friction, in the fictious time step we are allowed to write

$$\Delta\Pi_k(|\mathbf{f}_P|, \mathcal{D}t) + \Delta\Pi_\sigma(|\mathbf{f}_P|, \mathcal{D}t) + \Delta\Pi_p(|\mathbf{f}_P|, \mathcal{D}t) + \mathfrak{E} = 0 \,, \tag{14}$$

replacing $\mathbf{f}_P$ in all additive terms by (11), (12) and (13) by $|\mathbf{f}_P|$ only; $\mathfrak{E}$ here refers to the eventual energy dissipation by plasticizing or damage. It is clear that we are looking for a nontrivial solution $|\mathbf{f}_P|$ of (14), i. e. for its non-zero root, which can be performed e. g. using some inexact version of Newton iterations, avoiding the evaluation of the derivatives of particular additive terms of the left-hand side of (14). The 1st estimate for $|\mathbf{f}_P|$ can exploit the fact that the contribution of the elastic potential energy by (13) can be neglected for this purpose, as well as the contribution of $\mathfrak{E}$, not analyzed in more details here; therefore (14) degenerates to a quadratic equation, which can be solved analytically. Thus we have all data for the evaluation of (9) and (10), thus all positions of the nodes A, B, C, D at the end of the time step can be adjusted as

$$\mathbf{x}_i^\times = \mathbf{x}_i + w t_c \mathbf{v}_i + (1-w) t_c \mathbf{v}_i^\times \,, \tag{15}$$

using some appropriate weight $w \in [0, 1]$, e. g. $w = 1/2$ (if no better arguments for this choice are available).

4. Kinematic method

The fundamental assumption for the collision of bodies Ω_1 and Ω_2 in this method is the condition of their impenetrability, i. e. $\Omega_1 \cup \Omega_2$ must be empty. We have the discretised masses m_i related to the points $i \in \{A, B, C, D\}$, as in *Section* 3. The velocity vectors before the impact (a priori known) are $\mathbf{v}_i$, the velocity vectors after the impact (undetermined yet) are $\mathbf{v}_i^{\times}$. Altogether, four velocity vectors have to be determined, i. e. 12 scalar unknowns. The equations for determining the components of $\mathbf{v}_i^{\times *}$ can be obtained from the law of conservation of linear and angular momentum, then from kinematic condition, expressing the impossibility of change of shape of any colliding line or triangle in time of the impact, and lastly from the properties of the contact plane and the influence of friction. This approach is applicable to both cases a) and b) from *Section* 2. The obvious transformation to the local coordinate system $\mathbf{v}_i^* = \mathbf{R}\mathbf{v}_i$ is available again, as well as its inverse $\mathbf{v}_i = \mathbf{R}^{\mathrm{T}}\mathbf{v}_i^*$.

4.1. Absolute friction

We shall start with the choice $\beta = 1$, as introduced in *Section* 3.

4.2. Conservation of momentum

Generally, due to the conservation of linear momentum we can write $m_i\mathbf{v}_i^{\times *} = m_i\mathbf{v}_i^*$, $i \in \{A, B, C, D\}$ being considered as the Einstein summation index again. The conservation of angular momentum can be related to an arbitrary point, e. g. to the origin of coordinates; then it reads

$$\mathbf{x}_i^* \times m_i\mathbf{v}_i^{\times *} = \mathbf{x}_i^* \times m_i\mathbf{v}_i^* \,. \tag{16}$$

For the case a) the conservation of angular momentum can be related to the point $P \equiv Q$ and consequently, we can write two vector equations

$$\begin{aligned} m_A N_B \mathbf{v}_A^{\times} - m_B N_A \mathbf{v}_B^{\times} &= m_A N_B \mathbf{v}_A - m_B N_A \mathbf{v}_B \,, \\ m_C N_D \mathbf{v}_C^{\times} - m_D N_C \mathbf{v}_D^{\times} &= m_C N_D \mathbf{v}_C - m_D N_C \mathbf{v}_D \,. \end{aligned} \tag{17}$$

4.3. Kinematic conditions

For the case b) all in-planar velocity components must satisfy the condition of rigid body motion in the element plane. It will be useful to omit all directions $\mathbf{x}_{3i}$ for both position and velocity vectors, since they have no influence on deformation of the involved elements All upper indices * will be omitted for brevity, considering the local coordinate system in the compatible way with *Section* 3. Let $\mathfrak{I}_m$ be the mass moment of inertia, related to the axis x_3, in the center T of gravity of the total mass $m_T = m_A + m_B + m_C + m_D$, due to the absolute friction and the condition $\mathbf{v}_D = \mathbf{v}_Q$. Then the angular momentum to such axis is $\mathfrak{I}_m\omega = (x_{i1} - x_{T1})m_i v_{i2} - (x_{i2} - x_{T2})m_i v_{i1}$, where ω denotes the angular velocity to the axis x_3, introduced as $\omega = \mathfrak{I}_m/\mathfrak{I}$, $\mathfrak{I}$ being the moment of inertia to such axis. Then we have $v_{T1}^{\times} = m_i v_{i1}^{\times}/m_T$, $v_{T2}^{\times} = m_i v_{i2}^{\times}/m_T$. Taking also the impact rigidity into account, for $j \in \{A, B, C\}$ we can write finally

$$v_{j1}^{\times} = v_{T1}^{\times} - \omega(x_{j2} - x_{T2}) \,, \qquad v_{j1}^{\times} = v_{T2}^{\times} + \omega(x_{j1} - x_{T1}) \,. \tag{18}$$

4.4. Contact conditions, influence of friction

In *Section* 4 we have still considered, up to now, $\beta = 1$, thus $\mathbf{v}_P^\times = \mathbf{v}_Q^\times$, as explained in *Section* 3. Thus for the case a) we can write $N_A\mathbf{v}_A^\times + N_B\mathbf{v}_B^\times = N_C\mathbf{v}_C^\times + N_D\mathbf{v}_D^\times$, whereas for the case b) with $\mathbf{v}_D^\times = \mathbf{v}_Q^\times$ we have $N_j\mathbf{v}_j^\times = \mathbf{v}_D^\times$, using the notation compatible with (18) and (17).

4.5. Zero friction

For $\beta = 0$ all velocity vector components parallel to the sliding plane remain the same after impact as before, i. e. $v_1^\times = v_{i1}$, $v_2^\times = v_{i2}$, which decreases the number of unknowns from 12 to 4. Due to the conservation of linear momentum we have $m_i v_{i3}^\times = m_i v_{i3}$. Using the same notation as in the considerations related to $\beta = 1$, the conservation of angular momentum gives two scalar equations $x_{i1} m_i v_{i3}^\times = x_{i1} m_i v_{i3}$, $x_{i2} m_i v_{i3}^\times = x_{i2} m_i v_{i3}$. The velocity vector components perpendicular to the sliding plane at the colliding points P, Q after the impact are the same, i. e. $v_{Q3}^\times = v_{P3}^\times$, respecting the impenetrability assumption, which provides the last needed equation. Then for the case a) we have $N_A v_{A3}^\times + N_B v_{B3}^\times = N_C v_{C3}^\times + N_D v_{D3}^\times$ and for the case b) $N_j v_{j3}^\times = v_{D3}^\times$ analogously.

4.6. General friction

For $\beta \in [0, 1]$ the interpolation $\mathbf{v}^\times = \beta\mathbf{v}_i^{a\times *} + (1-\beta)\mathbf{v}_i^{z\times *}$ for all $i \in \{A, B, C, D\}$ can be recommended again, as in *Section* 3.

4.7. Adjustment of the coordinates of nodes

Let us remind the transformations of the type $\mathbf{v}_i^\times = \mathbf{R}^{\mathrm{T}}\mathbf{v}_i^{\times *}$, needed for the final update. Thus we come back to (15). Let us also remark that it is useful to keep the sign of the difference of velocities $\mathbf{v}_i^\times$ of the colliding points P, Q in memory until the next time step: if the sign does not change then the contact must be still handled, unlike the opposite case.

5. Illustrative examples

The first example is the problem of an elastic rod impacting a rigid barrier. The input values were taken from [8]. Fig. 4 shows a model of a rod of total length $L = 1\,\mathrm{m}$, cross section area $A = 1\,\mathrm{m}^2$, Young's modulus $E = 1\,\mathrm{MPa}$ and mass density $\rho = 1\,\mathrm{kg/m}^3$, divided into 100 elements with its mass discretised to the nodes, which is situated at distance $g_0 = 0\,\mathrm{m}$ towards the rigid barrier; its initial velocity is $v_0 = 1\,\mathrm{m/s}$. The time step for calculation $\mathcal{D}t = 10^{-8}\,\mathrm{s}$ is applied. Fig. 5 shows the comparison of results obtained by the energy, kinematic and penalty method, particularly displacement of the impacting node and the change of the components of energy.

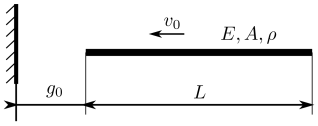


Figure 4: An elastic rod impacting a rigid barrier.

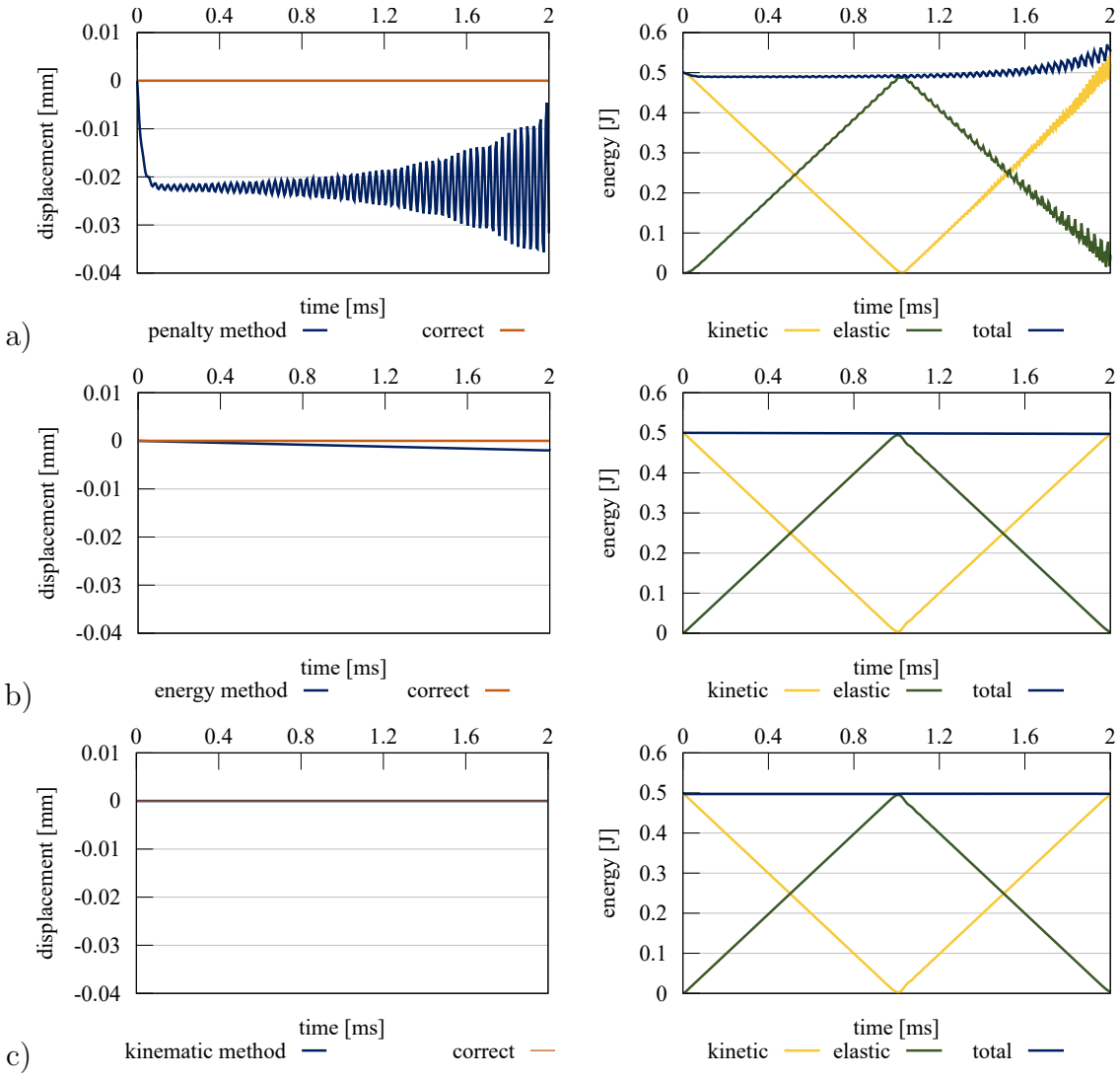


Figure 5: Time distribution of displacement and energy balance for a) the penalty method, using the penalty stiffness $P = 10^{11}\,\mathrm{Nm}^{-1}$, as introduced by [3], b) the energy method and c) the kinematic method.

The second example presents collision of two symmetrical cylinders, described in detail in [11], where the input data were also taken from. Fig. 6 shows two identical cylinders with radius $R = 4\,\mathrm{m}$, Young's modulus $E = 1000\,\mathrm{MPa}$, Poisson's ratio $\nu = 0.2$ and mass density $\rho = 1000\,\mathrm{kg/m}^3$ moving with the initial velocity $v_0 = 2\,\mathrm{m/s}$ against each other. The time step for calculation $\mathcal{D}t = 5 \cdot 10^{-6}\,\mathrm{s}$ is applied. Symmetry boundary conditions are applied. Fig. 7 demonstrates the time propagation and stress in time for the energy and kinematic methods separately, whereas Fig. 8 shows the time distribution of energy balance.

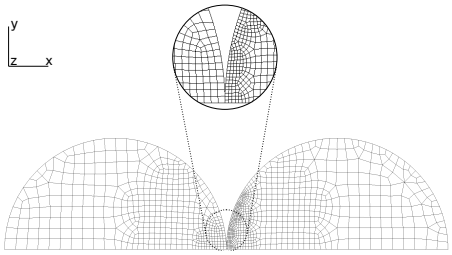
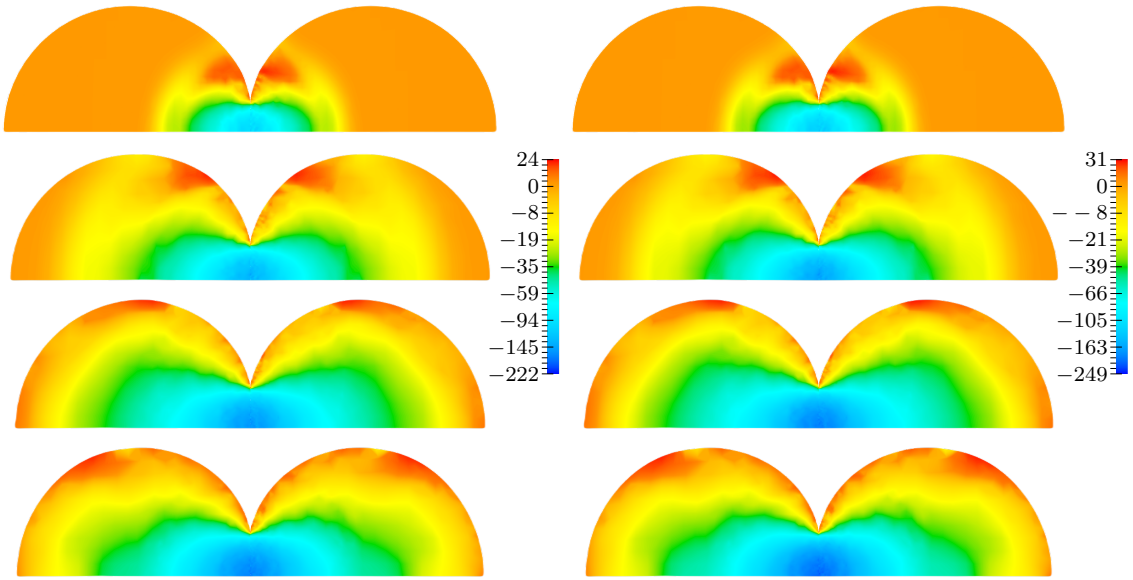
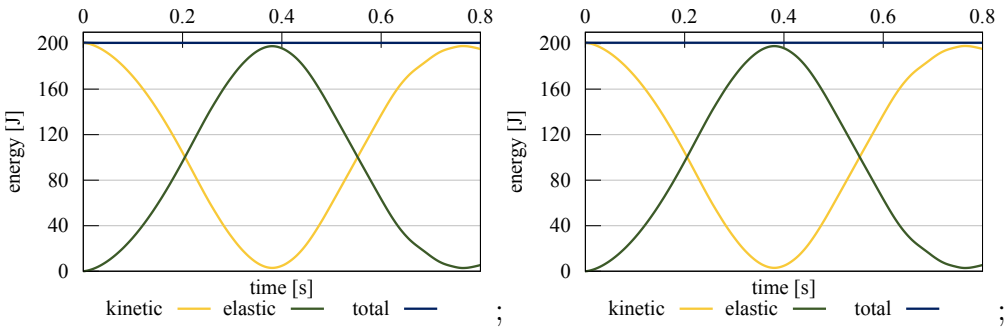


Figure 6: FE mesh, impact of two cylinders.



energy method kinematic method

Figure 7: Stress in the horizontal direction $\sigma_x[\mathrm{N/m^2}]$ during the wave propagation in times $t = 0.1, 0.2, 0.3, 0.4\,\mathrm{s}$.



Energy method Kinematic method

Figure 8: Time distribution of energy balance.

6. Conclusions

The most commonly used method for impact of bodies, called the penalty method, showed itself as unsatisfactory. This method is based on the idea that the contact force is proportional to the penetration of the colliding bodies. Therefore a violation of the principle of impenetrability of bodies is assumed and even necessary for it to work. This method also does not satisfy the conservation of energy law with sufficient precision and provides rather random results. Both methods introduced in this paper validated their superiority over the penalty method. The energy method satisfies the conservation of energy exactly, whereas the kinematic method preserves the principle of impenetrability. In the last decades several improvement of the penalty method and new approaches to the impact of bodies have been published, some of them being mentioned in References.

The authors of this paper have introduced two methods for transient analysis of impacts of bodies suitable for the explicit method. Both methods proved their good accuracy, efficiency and robustness. The energy conservation law is fulfilled very well without necessity of substantial shortage of the global time step of numerical integration and without necessity of introducing additional computational parameters, understanding them and determining their values. Both methods take the exact time of the impact for each contact into consideration. In the case of the kinematic method, all deformations, velocities and accelerations are determined with help of division of the time step into its substeps before and after the impact. The energy method introduces the equation of conservation of energy in each time step when a contact occurs, so all unwanted energy changes are eliminated.

The suggested approaches enable contacts of one surface with more nodes, as well as of one line with more lines, in one time step, as presented by the second numerical example. The methods do not demand any use of neither penalty method nor Lagrangian multipliers.

As for the problem of friction, all methods of static friction, based on the idea of pulling a burden on a surface, are problematic for the impact analysis. Unlike them, a very general model of the impact friction, assuming the theoretically clear limits of the friction, namely the zero friction and the absolute friction, is introduced in this paper. Then the real friction can be seen as a linear combination of those two limit cases. The friction coefficient, which is the relative weight factor of the absolute friction, can be determined by simple experiments.

Acknowledgements

This work was supported by the project of specific university research No. FAST-S-22-7867 at Brno University of Technology.

References

[1] Armero, F., and Petöcz, E. Formulation and analysis of conserving algorithms for frictionless dynamic contact / impact problems. *Comput. Methods Appl. Mech. Eng.* **158** (1998), 269–300.

[2] Banerjee, A., Chanda, A., and Das, R.: Historical origin and recent development on normal directional impact models for rigid body contact simulation: a critical review. *Arch. Comput. Methods Eng.* **24** (2017), 397–422.

[3] Benson, D. J., and Hallquist, J.: Computation for transient and impact dynamics. In: *Encyclopedia of Vibration*, Elsevier, Amsterdam, 2001.

[4] Bermúdez de Castro, A.: *Continuum Thermomechanics*. Birkhäuser, Basel, 2005.

[5] Betsch, P., and Hesch, Ch.: Energy-momentum conserving schemes for frictionless dynamic contact problems. In: *Proc. IUTAM Symposium on Computational Methods in Contact Mechanics* in Hannover (2006), Springer, Berlin, 2017, pp. 77–96.

[6] Dostál, Z., Gomes Neto, F. A. M., and Santos, S. A.: Solution of contact problems by FETI domain decomposition with natural coarse space projections. *Comp. Methods Appl. Mech. Eng.* **190** (2000), 1611–1627.

[7] Gilardi, G., and Sharf, I.: Literature survey of contact dynamics modelling. *Mech. Mach. Theory* **37** (2002), 1213–1239.

[8] Kopačka, J., Gabriel, D., Kolman, R., Plešek, J., and Ulbin, M.: Studies in numerical stability of explicit contact-impact algorithm to the finite element solution of wave propagation problems. In: *Proc. 4th COMPDYN (Computational Methods in Structural Dynamics and Earthquake Engineering)* in Kos (2013), ECCOMAS, Athens, 2013, pp. 787–800.

[9] Laursen, T. A., and Chavla, V.: Desing of energy conserving algorithms for frictionless dynamic contact problems. *Comp. Methods Appl. Mech. Eng.* **40** (1997), 863–886.

[10] Mijar, A. R., and Arora, J. S.: Review of formulations for elastostatic frictional contact problems. *Struct. Multidisc. Optim.* **20** (2000), 167–189.

[11] Otto, P., De Lorenzis, L., and Unger, J. F.: Explicit dynamics in impact simulation using a NURBS contact interface. *Int. J. Numer. Methods Eng.* **121** (2020), 1248–1267.

[12] Papadopulos, P., and Taylor, R. L.: A mixed formulation for the finite element solution of contact problems. *Comp. Methods Appl. Mech. Eng.* **94** (1992), 373–389.

[13] Sewerin, F., and Papadopoulos, P.: On the finite element solution of frictionless contact problems using an exact penalty approach. *Comput. Methods Appl. Mech. Eng.* **368** (2020), 113108 / 1–24.

[14] Simo, J. C., Wriggers, P., and Taylor, R. L.: A perturbed Lagrangian formulation for the finite element solution of contact problems. *Comput. Methods Appl. Mech. Eng.* **50** (1985), 163–180.

[15] Sofonea, M., Danan, D., and Zheng, C.: Primal and dual variational formulation of frictional contact problem. *Mediterr. J. Math.* **13** (2016), 857–872.

[16] Štekbauer, H., Němec, I., Lang, R., Burkart, D., and Vala, J.: On a new computational algorithm for impacts of elastic bodies. *Appl. Math.* **67** (2022), 775–804.

[17] Vala, J., and Rek, V.: On a computational approach to multiple contacts / impacts of elastic bodies. In: *Proc. 21st PANM (Programs and Algorithms of Numerical Mathematics)* in Jablonec n. N. (2022), Institute of Mathematics CAS, Prague, 2023, in print, 12 pp.

[18] Wang, D., de Boer, G., Neville, A., and Ghanbarzadeh, A.: A review on modelling of viscoelastic contact problems. *MDPI Lubricants* **10** (2022), 358 / 1–28.

[19] Wang, G., Liu, C., and Liu, Y.: Energy dissipation analysis for elastoplastic contact and dynamic dashpot models. *Int. J. Mech. Sci.* **221** (2022), 107214 / 1–14.

[20] Weyler, R., Oliver, J., Sain, T., and Cante, J. C.: On the contact domain method: a comparison of penalty and Lagrange multiplier implementations. *Comput. Methods Appl. Mech. Eng.* **205-208** (2008), 68–82.

[21] Wong, S.V., Hamouda, A.M.S., and Hashmi, M.S.J. Kinematic contact-impact algorithm with friction. *Int. J. Crashworthiness* **6** (2001), 65–82.

[22] Wriggers, P.: Finite element algorithms for contact problems. *Arch. Comput. Methods Eng.* **2** (1995), 1–49.

[23] Wu, S. R.: A variational principle for dynamic contact with large deformation. *Comput. Methods Appl. Mech. Eng.* **198** (2009), 2009–2015, and **199** (2009), p. 220.

[24] Zolghadr Jahromi, H., and Izzuddin, B. A.: Energy conserving algorithms for dynamic contact analysis using Newmark methods. *Comput. Struct.* **118** (2013), 74–89.

Programs and Algorithms of Numerical Mathematics 21
J. Chleboun, P. Kůs, J. Papež, M. Rozložník, K. Segeth, J. Šístek (Eds.)
Institute of Mathematics CAS, Prague 2023

MIXED PRECISION GMRES-BASED ITERATIVE REFINEMENT WITH RECYCLING

Eda Oktay, Erin Carson

Faculty of Mathematics and Physics, Charles University
Prague, Czech Republic
oktay@karlin.mff.cuni.cz, carson@karlin.mff.cuni.cz

Abstract: With the emergence of mixed precision hardware, mixed precision GMRES-based iterative refinement schemes for solving linear systems $Ax = b$ have recently been developed. However, in certain settings, GMRES may require too many iterations per refinement step, making it potentially more expensive than the alternative of recomputing the LU factors in a higher precision. In this work, we incorporate the idea of Krylov subspace recycling, a well-known technique for reusing information across sequential invocations, of a Krylov subspace method into a mixed precision GMRES-based iterative refinement solver. The insight is that in each refinement step, we call preconditioned GMRES on a linear system with the same coefficient matrix A. In this way, the GMRES solves in subsequent refinement steps can be accelerated by recycling information obtained from previous steps. We perform numerical experiments on various random dense problems, Toeplitz problems, and problems from real applications, which confirm the benefits of the recycling approach.

Keywords: GMRES, iterative refinement, mixed precision, recycling

MSC: 65F08, 65F10, 65G50, 65Y10

1. Introduction and background

There are various algorithms for solving linear systems of equations $Ax = b$, where $A \in \mathbb{R}^{n\times n}$ and $x, b \in \mathbb{R}^n$. One approach is iterative refinement (IR) which is based on improving the approximate solution in each refinement step [23]. Iterative refinement typically starts with using Gaussian elimination with partial pivoting (GEPP) to compute an initial approximate solution. Then using the L and U factors of A and the residual r_i, the system $Ad_{i+1} = r_i$ is solved for the correction term d_i to improve the approximate solution via $x_{i+1} = x_i + d_{i+1}$. A general IR scheme is shown in Algorithm 1.

Recently, mixed-precision capabilities have become available in hardware, which can have significant performance benefits [4, 1]. In Algorithm 1, the authors in [6]

DOI: 10.21136/panm.2022.14

used three hardware precisions: u_r for computing the residual, u_f for the LU factorization, and working precision u for the remaining calculations. There is also a fourth "effective solve precision" u_s, which depends on the particular solver and precisions used in line 5. The idea is that low precision can be used for the LU factorization, which is the most expensive part of the computation, and accuracy can be recovered through use of higher precisions in other parts of the computation. Indeed, it has been shown that on NVIDIA V100 GPUs, using half precision instead of double precision for the LU factorization can give over 4× speedups; see [12, Figure 3(b)]. Throughout this work we assume that $u_f \geq u \geq u_r$ e.g., (u_f, u, u_r)= (half, single, double).

Algorithm 1 General Iterative Refinement Scheme

Input: $n \times n$ matrix A; right-hand side b; max. number of refinement steps $i_{\max}$.
Output: Approximate solution x_{i+1} to $Ax = b$.

1: Compute LU factorization $A \approx LU$ in precision u_f.
2: Solve $Ax_0 = b$ by substitution in precision u_f; store x_0 in precision u.
3: **for** $i = 0 : i_{\max} - 1$ **do**
4: Compute $r_i = b - Ax_i$ in precision u_r; store in precision u.
5: Solve $Ad_{i+1} = r_i$ in precision u_s; store d_{i+1} in precision u.
6: Compute $x_{i+1} = x_i + d_{i+1}$ in precision u.
7: **if** converged **then** return x_{i+1} in precision u. **end if**

For a given combination of precisions and choice of solver, it is well-understood under which conditions Algorithm 1 will converge and what the limiting accuracy will be. The constraint for convergence in line 7 is usually stated via a constraint on the infinity-norm condition number of the matrix A. Table 1 shows the constraints on $\kappa_\infty(A)$ required for convergence of the normwise relative backward and forward errors to the level of the working precision for various precision combinations and solvers used in this study. For further information, see, e.g., [6, 3]. For a description of stopping criteria used for detecting convergence within iterative refinement in practice, see, e.g., [9], [19].

From Table 1, we see that if the computed LU factors are used to solve for the correction in line 5, often referred to as "standard IR" (SIR), then $\kappa_\infty(A) = \|A\|_\infty \|A^{-1}\|_\infty$ must be less than u_f^{-1} in order for convergence to be guaranteed. To relax this constraint on condition number, the authors of [5] and [6] devised a mixed precision GMRES-based iterative refinement scheme (GMRES-IR). In GMRES-IR, the correction equation in line 5 is solved via left-preconditioned GMRES, where the computed LU factors of A are used as preconditioners. This results in a looser constraint on condition number in order to guarantee the convergence of forward and backward errors; in the case that the preconditioned matrix is applied to a vector in each iteration of GMRES in double the working precision, we require $\kappa_\infty(A) \leq u^{-1/2} u_f^{-1}$, and in the case that a uniform precision is used within GMRES (a variant

u_f	u	u_r	SIR	GMRES-IR	SGMRES-IR
half	single	double	$2 \cdot 10^3$	$8 \cdot 10^6$	$4 \cdot 10^4$
half	double	quad	$2 \cdot 10^3$	$2 \cdot 10^{11}$	$3 \cdot 10^7$
single	double	quad	$2 \cdot 10^7$	$2 \cdot 10^{15}$	$1 \cdot 10^{10}$

Table 1: Constraints on $\kappa_\infty(A)$ for which the relative forward and normwise backward errors are guarantee to converge to the level u for a given combination of precisions for SIR, GMRES-IR (which uses double the working precision in applying the preconditioned matrix to a vector) and SGMRES-IR (which uses the working precision throughout).

which we call SGMRES-IR), we require $\kappa_\infty(A) \leq u^{-1/3}u_f^{-2/3}$; see [3]. If these constraints are met, preconditioned GMRES is guaranteed to converge to a backward stable solution after n iterations and the iterative refinement scheme will converge to its limiting accuracy. We note that to guarantee backward stability, all existing analyses (e.g., [5, 6, 3]) assume that unrestarted GMRES is used within GMRES-IR.

We also note that existing analyses do not guarantee how fast GMRES will converge in each refinement step, only that it will do so within n iterations. However, if indeed n iterations are required to converge in each GMRES solve, this can make GMRES-IR more expensive than simply computing the LU factorization in higher precision and using SIR. Indeed, high-performance experiments show that slow GMRES convergence can negatively impact the achievable performance; see [12, Figure 7 (b)]. To make more precise the relative costs of each step of SIR and GMRES-IR, we list their costs in terms of asymptotic computational complexity in Table 2.

Unfortunately, GMRES convergence speed is difficult to predict. In fact, for any set of prescribed eigenvalues, one can construct a linear system for which GMRES will stagnate entirely until the nth iteration [11]. The situation is better understood at least in the case of normal matrices; see, e.g., [15]. The worst-case scenario in the case of normal A is when eigenvalues are clustered near the origin, which can cause complete stagnation of GMRES [15]. After the preconditioning step in GMRES-IR, all eigenvalues of the preconditioned matrix $(U^{-1}L^{-1}A)$ ideally become 1 in the absence of finite precision error in computing LU and within GMRES. However, in practice, since we have inexact LU factors, if A has a cluster of eigenvalues near the origin, this imperfect preconditioner may fail to shift some of them away from the origin, which can cause GMRES to stagnate. For instance, when random dense matrices having geometrically distributed singular values are used in the multistage iterative refinement algorithm devised in [19], the authors showed that for relatively large condition numbers relative to precision u_f, GMRES tends to perform n iterations in each refinement step.

Figure 1 shows the eigenvalue distribution of a double-precision 100×100 random dense matrix having geometrically distributed singular values with condition number $\kappa_2(A) = 10^{12}$, generated via the command `gallery('randsvd',100,1e12,3)` in

Once per IR solve (both variants)	$O(n^3)$	in precision u_f	(LU fact.)
SIR step	$O(n^2)$	in precision u_f	(tri. solves)
GMRES-IR step (k GMRES iterations)	$O(nk^2)$	in precision u	(orthog.)
	$O(nnz \cdot k)$	in precision u or u^2	(SpMV)
	$O(n^2k)$	in precision u or u^2	(precond.)
Once per refinement step (both variants)	$O(nnz)$	in precision u_r	(residual comp.)
	$O(n)$	in precision u	(sol. update)

Table 2: Asymptotic computational complexity of operations in each refinement step for SIR and GMRES-IR.

MATLAB. In the unpreconditioned case (left), the eigenvalues are clustered around the origin, a known difficult case for GMRES. When double-precision LU factors are used for preconditioning (middle), the eigenvalues of the preconditioned system are now clustered around 1. On the other hand, using half-precision LU factors as preconditioners (right) causes a cluster of eigenvalues to remain near the origin, indicating that GMRES convergence will likely be slow (we note that these are nonnormal matrices and so the theory of [15] does not apply, but our experimental evidence indicates that this is the case).

Thus, even when low-precision LU factors can theoretically be used in GMRES-IR, they may not be the best choice from a performance perspective. In this scenario, we are left with two options: either increase the precision in which the LU factors are computed, or seek to improve the convergence behavior of GMRES through other means. It is the latter approach that we take in this work. In particular, we investigate the use of Krylov subspace recycling.

In Section 2, we give a background on the use of recycling in Krylov subspace methods and describe our approach. Extensive numerical experiments that demonstrate the potential benefit of recycling within GMRES-based iterative refinement are presented in Section 3. We outline open problems in Section 4.

2. Iterative refinement with Krylov subspace recycling

One way to speed up the convergence of GMRES is using recycling [20, 21]. The idea of recycling is that if we have a sequence of linear systems ($Ax_1 = b_1, Ax_2 = b_2, \ldots$) to solve involving the same (or a similar) coefficient matrix A, then we can reuse the Krylov subspace information generated in solving $Ax_1 = b_1$ to speed up converge of the method in solving $Ax_2 = b_2$, etc. This is exactly the situation we have in GMRES-IR: within the iterative refinement loop, we call GMRES on the matrix A many times, and only the right-hand side changes between refinement steps. Thus we can use Krylov subspace recycling within GMRES across iterative refinement steps, and theoretically the convergence of GMRES should improve as the refinement proceeds.

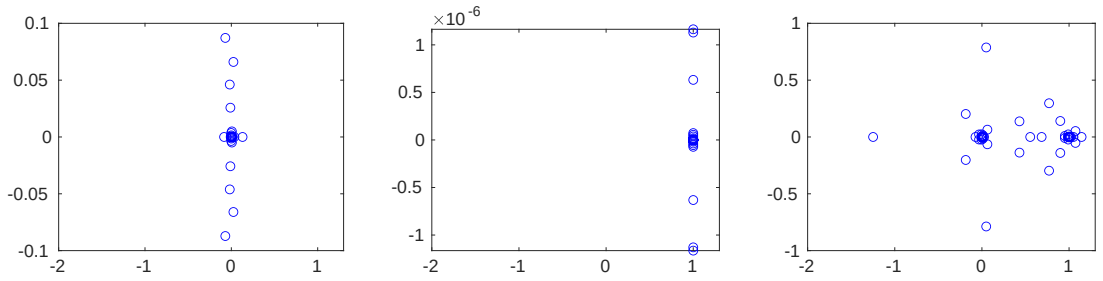


Figure 1: Eigenvalue distribution of a double-precision random dense matrix with $\kappa_2(A) = 10^{12}$ without preconditioner (left), with a double-precision LU preconditioner (middle), and with a half-precision LU preconditioner (right).

GMRES-DR [17] is a truncated and restarted solver developed for solving single nonsymmetric linear systems. The method deflates small eigenvalues for the new subspace to improve the convergence of restarted GMRES. Another truncated solver used for recycling is called GCROT [8]. The method recycles a subspace that minimizes the loss of orthogonality with Krylov subspace from the previous system.

By far the most popular Krylov subspace method implementing recycling is GCRO-DR [20]. GCRO-DR can be seen as a combination of GMRES-DR and a modified GCROT. In GCRO-DR, the residual minimization and orthogonalization are performed over the recycled subspace, leading to an adaptive truncated recycling method. GCRO-DR uses the deflated restarting idea in GMRES-DR in the same manner as GCROT. Let m denote the maximum size of the Krylov subspace and let k denote the number of vectors to recycle. In one cycle of GCRO-DR(m, k), first, using the k harmonic Ritz vectors corresponding to the k smallest harmonic Ritz values, the solution space is constructed. After finding the optimal solution over the solution space and computing the residual, GCRO-DR constructs the Arnoldi relation by generating a Krylov subspace of dimension $m - k + 1$. After completing the Arnoldi process, the algorithm solves a minimization problem at the end of each cycle, which reduces to an $(m + 1) \times m$ least-squares problem. After solving the least-squares problem and computing the residual, a generalized eigenvalue problem is solved, and harmonic Ritz vectors are recovered. Since harmonic Ritz vectors are constructed differently than in GMRES-DR, GCRO-DR is suitable for solving individual linear systems and sequences of them. See [20] for further details.

The use of recycling may also be favorable from a performance perspective. GCRO-DR performs only $m - k$ Arnoldi steps implying that it performs $m - k$ matrix-vector multiplications per cycle, whereas GMRES(m) performs m matrix-vector multiplications. It is also mentioned in [20] that since GCRO-DR stores U_k and C_k, it performs $2kn(1 + k)$ fewer operations during the Arnoldi process. On the other hand, since we are using k eigenvectors, GCRO-DR(m, k) requires storing k more vectors than GMRES(m).

In an effort to reduce the overall computational cost of the GMRES solves within GMRES-IR, we develop a recycled GMRES-based iterative refinement algorithm, called RGMRES-IR. In line 5 of Algorithm 1, RGMRES-IR uses preconditioned GCRO-DR(m, k) instead of preconditioned GMRES to solve the correction equation. As in GMRES-IR, RGMRES-IR will use three precisions: u_f, u, and u_r. Again we note that we assume $u_r \leq u \leq u_f$. Using different precision settings results in different constraints on the condition number to guarantee convergence of the forward and backward errors. Although our experiments here will use three different precisions, two precisions (only computing residuals in higher precision) or fixed (uniform) precision can also be used in the RGMRES-IR algorithm.

3. Numerical experiments

In this section, we compare GMRES-IR and RGMRES-IR for solving $Ax = b$. We adapted MATLAB implementations of the GMRES-IR method from [6], and the GCRO-DR method from [20]. To simulate half-precision, we use the `chop` library and associated functions from [13], available at `https://github.com/higham/chop` and `https://github.com/SrikaraPranesh/LowPrecision_Simulation`. For single and double precision, we use the MATLAB built-in data types and to simulate quadruple precision we use the Advanpix multiprecision computing toolbox, see [2]. We restrict ourselves to IEEE precisions, although we note that one could also use formats like bfloat16 [14]. The experiments are performed on a computer with Intel Core i7-9750H having 12 CPUs and 16 GB RAM with OS system Ubuntu 20.04.1. Our RGMRES-IR algorithm and associated functions are available at `https://github.com/edoktay/rgmresir`, which includes scripts for generating the data and plots in this work.

The GMRES convergence tolerance τ dictates the stopping criterion for the inner GMRES iterations. GMRES is considered converged if the relative (preconditioned) residual norm drops below τ. In tests here with single working precision, we use $\tau = 10^{-4}$. For double working precision, we use $\tau = 10^{-8}$. Note that for the outer iterative refinement scheme, we explicitly compute the true solution x and stop the iterations if the forward and backward errors are less than u. For practical stopping criteria relevant to GMRES-IR schemes, see [19]. To ensure that we fully exhibit the behavior of the methods, we set the maximum number of refinement steps to $i_{\max} = 10000$, which is large enough to allow all approaches that eventually converge sufficient time to do so.

The results are compared in two different metrics: the number of GMRES iterations per refinement step and the total number of GMRES iterations. For simplicity, b is chosen to be the vector of ones for all matrices, and the precisions are chosen such that $u \leq u_f^2$, and $u_r \leq u^2$. We compare GMRES-IR and RGMRES-IR in the setting where precision u^2 is used for preconditioning, except for the experiments in Section 3.3.1 where we investigate the use of uniform precision within the solver.

For a fair comparison between GMRES-IR and RGMRES-IR, GMRES-IR is used with restart value m, which is the maximum size of the subspace used in RGMRES-IR. Since the first refinement step of RGMRES-IR does not have a recycled subspace, it is the same as the first step of GMRES-IR. We thus expect a decrease in the number of GMRES iterations per refinement step starting from the second refinement step. For RGMRES-IR, the optimal number $k < m$ of harmonic Ritz vectors is chosen for each group of matrices with the desired precision settings after several experiments on various (m, k) scenarios. The optimum k differs for each matrix. The least total number of GMRES iterations is obtained for $k =$ (# of GMRES iterations in the first refinement step) -1 since, in this case, we are recycling the whole generated subspace, which is expensive. Thus one should choose a k value as small as possible to reduce computational cost while benefiting from recycling. Figure 2 shows the change in the total GMRES iterations according to the given k values for two matrices. From the plots, one can easily *find the knee*, i.e., find the smallest k value that gives a reasonably small number of GMRES iterations.

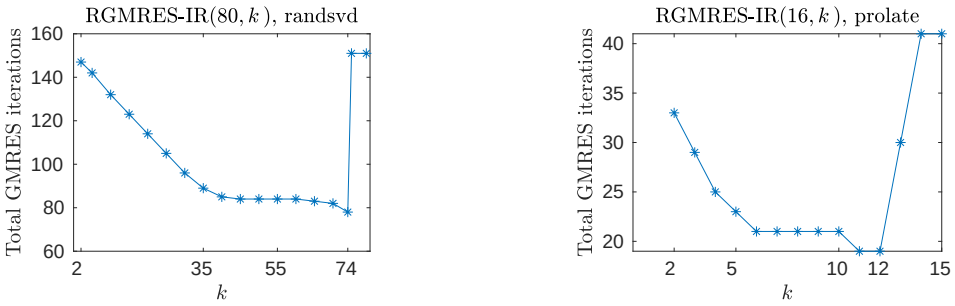


Figure 2: Total GMRES iterations for various k for a randsvd matrix with $\kappa_2(A) = 10^{13}$ (left) and a prolate matrix with $\alpha = 0.434$ (right) for $(u_f, u, u_r) =$ (single, double, quad).

In the tables we will present, the first number shows the total number of GMRES iterations. The numbers in the parentheses indicate the number of GMRES iterations performed in each refinement step. For instance, 5(2,3) implies that there are 2 refinement steps, the first of which performs 2 GMRES iterations and the second of which performs 3, giving a total of 5 GMRES iterations. We note that we use the GMRES iteration count as a proxy for performance, although this is not a perfect measure; for recent results on mixed precision GMRES-IR with restarting see [16]. We also note that additional numerical experiments for RGMRES-IR can be found in the associated technical report [18].

3.1. Prolate matrices

We first test our algorithm on prolate (symmetric, ill-conditioned Toeplitz matrices whose eigenvalues are distinct, lie in the interval $(0, 1)$, and tend to cluster around 0 and 1) matrices [22] of dimension $n = 100$, generated using the MATLAB

command `gallery('prolate',n,alpha)`, where `alpha` is the array of the desired parameters $\alpha = \{0.475, 0.47, 0.467, 0.455, 0.45, 0.4468, 0.44, 0.434\}$. When $\alpha < 0.5$, it becomes difficult for GMRES-IR to solve the system since the eigenvalues skew more towards zero. Table 3 shows the number of GMRES iterations performed by GMRES-IR and RGMRES-IR for the setting (u_f, u, u_r) = (half, single, double).

From Table 3, we can see that GMRES-IR diverges for $\alpha < 0.45$ with and without recycling. However, when $\alpha = 0.45$, we see that RGMRES-IR diverges although GMRES-IR converges. This is because of the multiple periods of stagnation in the second refinement step due to recycling. GCRO-DR cannot converge in the first $16 - 5 = 11$ iterations in the second step, causing an infinite restart which results in divergence. However, for cases where both GMRES-IR and RGMRES-IR converge, RGMRES-IR always requires fewer GMRES iterations. For $\alpha = 0.455$, GMRES restarts in the second refinement step of GMRES-IR, while recycling allows RGMRES-IR to converges without restarting, decreasing the cost.

α	$\kappa_\infty(A)$	$\kappa_2(A)$	GMRES-IR (16)	RGMRES-IR (16,5)
0.475	$1 \cdot 10^6$	$4 \cdot 10^5$	12 (6,6)	8 (6,2)
0.47	$3 \cdot 10^7$	$8 \cdot 10^6$	16 (8,8)	10 (8,2)
0.467	$2 \cdot 10^8$	$5 \cdot 10^7$	19 (9,10)	11 (9,2)
0.455	$3 \cdot 10^{11}$	$8 \cdot 10^{10}$	50 (15,25,10)	19 (15,4)
0.45	$7 \cdot 10^{12}$	$2 \cdot 10^{12}$	89 (14,43,32)	-
0.4468	$5 \cdot 10^{13}$	$1 \cdot 10^{13}$	-	-
0.44	$3 \cdot 10^{15}$	$9 \cdot 10^{14}$	-	-
0.434	$3 \cdot 10^{16}$	$9 \cdot 10^{15}$	-	-

Table 3: Number of GMRES-IR and RGMRES-IR refinement steps/GMRES iterations for prolate matrices with various α values, using precisions (u_f, u, u_r) = (half, single, double) and (m, k) = (16,5).

3.2. SuiteSparse matrices

We now test our algorithm on three real matrices taken from the SuiteSparse Collection [7]. Table 4 compares the performance of GMRES-IR and RGMRES-IR for precisions (u_f, u, u_r) = (half, double, quad). It is seen that RGMRES-IR successfully reduces the total number of GMRES iterations in all cases.

3.3. Random dense matrices

Finally, we test our algorithm on random dense matrices of dimension $n = 100$ having geometrically distributed singular values. We generated the matrices using the MATLAB command `gallery('randsvd',n,kappa(i),3)`, where `kappa` is the array of the desired 2-norm condition numbers $\kappa_2(A) = \{10^4, 10^5, 10^6, 10^7, 10^8, 10^9, 10^{10}, 10^{11}, 10^{12}, 10^{13}\}$, and mode `3` corresponds to the matrix having geometrically

Matrix	n	nnz	$\kappa_\infty(A)$	GMRES-IR (40)	RGMRES-IR(40,10)
orsirr_1	1030	6858	$1 \cdot 10^5$	22 (11,11)	20 (11,9)
comsol	1500	97645	$3 \cdot 10^6$	52 (25,27)	34 (25,9)
circuit204	1020	5883	$9 \cdot 10^9$	59 (18,20,21)	47 (18,14,15)

Table 4: Number of GMRES-IR and RGMRES-IR refinement steps/GMRES iterations for real matrices, using precisions (u_f, u, u_r) = (half, double, quad) and (m, k) = (40,10).

distributed singular values. For reproducibility, we use the MATLAB command `rng(1)` each time we run the algorithm. We compare methods using precisions (u_f, u, u_r) = (single, double, quad) and (u_f, u, u_r) = (half, double, quad). As shown in Figure 1, these matrices have eigenvalues clustered around the origin, which can be a difficult case for GMRES convergence. This class of problems thus represents a good use case for the RGMRES-IR algorithm.

3.3.1. SGMRES-IR versus RSGMRES-IR

In practice, implementations often use a uniform precision within GMRES (i.e., applying the preconditioned matrix to a vector in precision u rather than u^2). This is beneficial from a performance perspective (in particular if precision u^2 must be implemented in software). The cost is that the constraint on condition numbers for which the refinement scheme is guaranteed to converge becomes tighter. To illustrate the benefit of recycling in this scenario, we first compare what we call SGMRES-IR (GMRES-IR but with a uniform precision within GMRES) to the recycled version, RSGMRES-IR. For a fair comparison, restarted SGMRES-IR (SGMRES-IR(m)) is compared with recycled SGMRES-IR (RSGMRES-IR(m, k)).

Table 5 shows the number of GMRES iterations performed by SGMRES-IR and RSGMRES-IR in the (u_f, u, u_r) = (single, double, quad) setting. We observe that recycling reduces the number of GMRES iterations in this case as well. The reason why SGMRES-IR does not converge for $\kappa_2(A) \geq 10^{14}$ is that in the first refinement step, restarted SGMRES does not converge (restarting an infinite number of times). For RSGMRES-IR, in the first GCRO-DR call, recycling after the first restart cycle helps, allowing GCRO-DR to converge. We note that this is another benefit of the recycling approach, as it can improve the reliability of restarted GMRES, which is almost always used in practice.

3.3.2. GMRES-IR versus RGMRES-IR

We now return to our usual setting and compare GMRES-IR and RGMRES-IR for random dense matrices with condition numbers $\kappa_2(A) = \{10^4, 10^5, 10^6, 10^7, 10^8, 10^9, 10^{10}, 10^{11}, 10^{12}, 10^{13}, 10^{14}, 10^{14}\}$. Results using precisions (u_f, u, u_r) = (half, double, quad) and two different choices of (m, k) are displayed in Table 6.

$\kappa_\infty(A)$	$\kappa_2(A)$	SGMRES-IR (80)	RSGMRES-IR (80,18)
$6 \cdot 10^9$	10^9	64 (19,23,22)	34 (19,8,7)
$6 \cdot 10^{10}$	10^{10}	120 (39,40,41)	65 (39,13,13)
$6 \cdot 10^{11}$	10^{11}	160 (52,54,54)	94 (52,21,21)
$6 \cdot 10^{12}$	10^{12}	196 (65,65,66)	163 (65,32,32,34)
$5 \cdot 10^{13}$	10^{13}	301 (75,75,75,76)	199 (75,41,41,42)
$5 \cdot 10^{14}$	10^{14}	-	493 (131,51,51,52,52,52,52,52)
$5 \cdot 10^{15}$	10^{15}	-	2093*

Table 5: Number of SGMRES-IR and RSGMRES-IR refinement steps/GMRES iterations for precisions (u_f, u, u_r) = (single, double, quad) and (m, k) = (80,18). For $\kappa_2(A) = 10^{15}$, RSGMRES-IR required 2093 total GMRES iterations over 37 refinement steps.

$\kappa_\infty(A)$	$\kappa_2(A)$	GMRES-IR (100)	RGMRES-IR (100,30)	GMRES-IR (90)	RGMRES-IR (90,40)
$9 \cdot 10^4$	10^4	33 (16,17)	33 (16,17)	33 (16,17)	33 (16,17)
$8 \cdot 10^5$	10^5	85 (41,44)	71 (41,15,15)	85 (41,44)	50 (41,9)
$7 \cdot 10^6$	10^6	134 (66,68)	85 (66,19)	134 (66,68)	81 (66,15)
$7 \cdot 10^7$	10^7	167 (83,84)	113 (83,30)	167 (83,84)	100 (83,17)
$7 \cdot 10^8$	10^8	193 (96,97)	138 (96,42)	-	149 (119,30)
$6 \cdot 10^9$	10^9	200 (100,100)	151 (100,51)	-	179 (134,45)
$6 \cdot 10^{10}$	10^{10}	200 (100,100)	158 (100,58)	-	470 (388,41,41)
$6 \cdot 10^{11}$	10^{11}	200 (100,100)	165 (100,65)	-	-
$6 \cdot 10^{12}$	10^{12}	200 (100,100)	170 (100,70)	-	-
$5 \cdot 10^{13}$	10^{13}	3954*	241 (171,70)	-	-

Table 6: Number of GMRES-IR and RGMRES-IR refinement steps/GMRES iterations for random dense matrices having geometrically distributed singular values (mode 3) with various condition numbers, using precisions (u_f, u, u_r) = (half, double, quad) and settings (m, k) = (100,30) and (m, k)=(90,40). For $m = 100$ and 10^{13}, GMRES-IR required 3954 total GMRES iterations over 45 refinement steps.

For both choices of (m, k), when $\kappa_\infty(A) > 10^5$, recycling reduces the total number of GMRES iterations. This class of matrices with a low-precision LU preconditioner is a known difficult case for GMRES, and thus we can clearly see the benefits of recycling. We see the most significant improvement for the matrix with $\kappa_2(A) = 10^{13}$, in which RGMRES-IR requires over 16× fewer GMRES iterations than GMRES-IR when $m = 100$. We note that GMRES-IR is only guaranteed to converge up to $\kappa_2(A) < 10^{12}$ for this combination of precisions; for detailed information, see [6].

The reason that RGMRES-IR outperforms GMRES-IR for $m = 100$ and $\kappa_2(A) = 10^{13}$ is different than in the previous cases (caused by stagnation due to restarting), and is almost accidental in this case. We investigate this more closely

in Figure 3. In the left plot, we see the convergence trajectory of GMRES(100). In the first restart cycle, the residual decreases from 10^6 to 10^3 after 100 GMRES iterations. GMRES restarts and performs two more iterations, at which point it converges to a relative residual of 10^{-8} (absolute residual of around 10^{-2}). Hence, the first refinement step of GMRES-IR does $100+2=102$ iterations. The right plot shows the residual trajectory for GCRO-DR. The first restart cycle is the same as in GMRES; however, once the method restarts, the residual stagnates just above the level required to declare convergence. After $m-k=70$ more iterations, GCRO-DR restarts again, and this time, the residual drops significantly lower. So while GCRO-DR requires more iterations (171) to converge to the specified tolerance, when it does converge, it converges to a solution with a smaller residual. This phenomenon can in turn reduce the total number of refinement steps required. It is possible that we could reduce the overall number of GMRES iterations within GMRES-IR (and also RGMRES-IR) by making the GMRES convergence tolerance τ smaller. We did not experiment with changing the GMRES tolerance within GMRES-IR or RGMRES-IR, but this trade-off would be interesting to explore in the future.

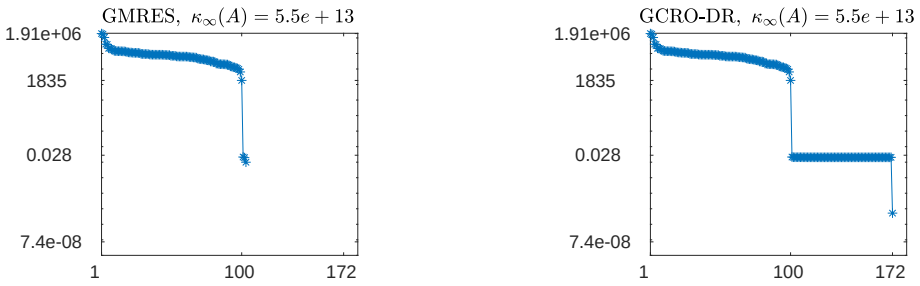


Figure 3: Residual trajectory of GMRES (left) and GCRO-DR (right), used within GMRES-IR and RGMRES-IR, respectively, for a randsvd matrix with $\kappa_2(A) = 10^{13}$ and precisions (u_f, u, u_r) = (half, double, quad).

We stress that the convergence guarantees for GMRES-IR for various precisions stated in [5, 6, 3] hold only for the case of unrestarted GMRES, i.e., $m = n$. When $m < n$, there is no guarantee that GMRES converges to a backward stable solution and thus no guarantee that GMRES-IR will converge. Choosing a restart parameter m that allows for convergence is a difficult problem, and a full theory regarding the behavior of restarted GMRES is lacking. The behavior of restarted GMRES is often unintuitive; whereas one would think that a larger restart parameter is likely to be better than a smaller one as it is closer to unrestarted GMRES, this is not always the case. In [10], the author gives examples where a larger restart parameter causes complete stagnation, whereas a smaller one results in fast convergence.

In Table 6 for the case $m = 90$, we can see that both methods converge for $\kappa_\infty(A) < 10^8$. After this point, GMRES-IR does not converge, whereas RGMRES-IR does. This serves as an example where the convergence guarantees given in [5, 6]

do not hold for GMRES-IR with restarted GMRES; for unrestarted GMRES, convergence is guaranteed up to $\kappa_\infty(A) \leq 10^{12}$ for this precision setting. Here, GMRES-IR does not converge because of the stagnation caused by restarting in the first refinement step. Aided by the recycling between restart cycles, RGMRES-IR does converge up to $\kappa_2(A) = 10^{10}$, although the large number of GMRES iterations required in the first refinement step makes this approach impractical.

4. Conclusion and future work

In this work, we have incorporated Krylov subspace recycling into mixed precision GMRES-based iterative refinement in order to reduce the total number of GMRES iterations required. We call our algorithm RGMRES-IR. Instead of preconditioned GMRES, RGMRES-IR uses a preconditioned GCRO-DR algorithm to solve for the approximate solution update in each refinement step. Our numerical experiments on random dense matrices, prolate matrices, and matrices from SuiteSparse [7] show the potential benefit of the recycling approach. Even in cases where the number of GMRES iterations does not preclude the use of GMRES-based iterative refinement, recycling can have a benefit. In particular, it can improve the reliability of restarted GMRES, which is used in most practical scenarios.

One major caveat for GMRES-based iterative refinement schemes is that the analysis and convergence criteria discussed in the literature all rely on the use of unrestarted GMRES. When restarted GMRES is used, we cannot give such concrete guarantees, as restarted GMRES may not converge even in infinite precision. A greater understanding of the theoretical behavior of restarted GMRES (and GCRO-DR) both in infinite and finite precision would be of great interest. Another potential future direction is the exploration of the potential for the use of mixed precision within GCRO-DR. We expect that it may be possible to use low precision within GCRO-DR, for example, in the computation of harmonic Ritz pairs.

Acknowledgements

We acknowledge funding from Charles University project PRIMUS/19/SCI/11, Charles University Research Program No. UNCE/SCI/023, and the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration.

References

[1] Abdelfattah, A. et al.: A survey of numerical linear algebra methods utilizing mixed-precision arithmetic. The International Journal of High Performance Computing Applications **35** (2021), 344–369.

[2] Advanpix LLC. Multiprecision Computing Toolbox for MATLAB. URL http://www.advanpix.com/.

[3] Amestoy, P. et al.: Five-precision GMRES-based iterative refinement. MIMS EPrint 2021.5, Manchester Institute for Mathematical Sciences, The University of Manchester, Manchester, UK, 2021. URL http://eprints.maths.manchester.ac.uk/2807/.

[4] Baboulin, M. et al.: Accelerating scientific computations with mixed precision algorithms. Computer Physics Communications **180** (2009), 2526–2533.

[5] Carson, E. and Higham, N.J.: A new analysis of iterative refinement and its application to accurate solution of ill-conditioned sparse linear systems. SIAM Journal on Scientific Computing **39** (2017), A2834–A2856.

[6] Carson, E. and Higham, N.J.: Accelerating the solution of linear systems by iterative refinement in three precisions. SIAM Journal on Scientific Computing **40** (2018), A817–A847.

[7] Davis, T.A. and Hu, Y.: The University of Florida Sparse Matrix Collection. ACM Transactions on Mathematical Software **38** (2011).

[8] De Sturler, E.: Truncation strategies for optimal Krylov subspace methods. SIAM Journal on Numerical Analysis **36** (1999), 864–889.

[9] Demmel, J. et al.: Error bounds from extra-precise iterative refinement. ACM Trans. Math. Softw. **32** (2006), 325–351.

[10] Embree, M.: The tortoise and the hare restart GMRES. SIAM Review **45** (2003), 259–266.

[11] Greenbaum, A., Pták, V., and Strakoš, Z.: Any nonincreasing convergence curve is possible for GMRES. SIAM Journal on Matrix Analysis and Applications **17** (1996), 465–469.

[12] Haidar, A., Tomov, S., Dongarra, J., and Higham, N.J.: Harnessing gpu tensor cores for fast fp16 arithmetic to speed up mixed-precision iterative refinement solvers. In: *SC18: International Conference for High Performance Computing, Networking, Storage and Analysis*. 2018 pp. 603–613.

[13] Higham, N.J. and Pranesh, S.: Simulating low precision floating-point arithmetic. SIAM Journal on Scientific Computing **41** (2019), C585–C602.

[14] Intel Corporation: Bfloat16 – hardware numerics definition. Tech. Rep. 338302-001US, Revision 1.0, Intel, 2018.

[15] Liesen, J. and Tichý, P.: The worst-case GMRES for normal matrices. BIT Numerical mathematics **44** (2004), 79–98.

[16] Lindquist, N., Luszczek, P., and Dongarra, J.: Accelerating restarted GMRES with mixed precision arithmetic. IEEE Transactions on Parallel and Distributed Systems **33** (2022), 1027–1037.

[17] Morgan, R.B.: GMRES with deflated restarting. SIAM Journal on Scientific Computing **24** (2002), 20–37.

[18] Oktay, E. and Carson, E.: Mixed precision GMRES-based iterative refinement with recycling. arXiv preprint arXiv:2201.09827 (2022).

[19] Oktay, E. and Carson, E.: Multistage mixed precision iterative refinement. Numerical Linear Algebra with Applications (2022), e2434.

[20] Parks, M.L., de Sturler, E., Mackey, G., Johnson, D.D., and Maiti, S.: Recycling Krylov subspaces for sequences of linear systems. SIAM Journal on Scientific Computing **28** (2006), 1651–1674.

[21] Soodhalter, K.M., de Sturler, E., and Kilmer, M.: A survey of subspace recycling iterative methods. arXiv preprint arXiv:2001.10347 (2020).

[22] Varah, J.: The prolate matrix. Linear Algebra and its Applications **187** (1993), 269–278.

[23] Wilkinson, J.H.: *Rounding errors in algebraic processes.* Prentice-Hall, 1963.

Programs and Algorithms of Numerical Mathematics 21
J. Chleboun, P. Kůs, J. Papež, M. Rozložník, K. Segeth, J. Šístek (Eds.)
Institute of Mathematics CAS, Prague 2023

TESTING THE METHOD OF MULTIPLE SCALES AND THE AVERAGING PRINCIPLE FOR MODEL PARAMETER ESTIMATION OF QUASIPERIODIC TWO TIME-SCALE MODELS

Štěpán Papáček[1], Ctirad Matonoha[2]

[1] Institute of Information Theory and Automation of the Czech Academy of Sciences
Pod Vodárenskou věží 4, 182 00 Prague 8, Czech Republic
papacek@utia.cas.cz
[2] Institute of Computer Science of the Czech Academy of Sciences
Pod Vodárenskou věží 2, 182 07 Prague 8, Czech Republic
matonoha@cs.cas.cz

Abstract: Some dynamical systems are characterized by more than one time-scale, e.g. two well separated time-scales are typical for quasiperiodic systems. The aim of this paper is to show how singular perturbation methods based on the slow-fast decomposition can serve for an enhanced parameter estimation when the slowly changing features are rigorously treated. Although the ultimate goal is to reduce the standard error for the estimated parameters, here we test two methods for numerical approximations of the solution of associated forward problem: (i) the multiple time-scales method, and (ii) the method of averaging. On a case study, being an under-damped harmonic oscillator containing two state variables and two parameters, the method of averaging gives well (theoretically predicted) results, while the use of multiple time-scales method is not suitable for our purposes.

Keywords: dynamical system, singular perturbation, averaging, parameter estimation, slow-fast decomposition, damped oscillations

MSC: 92C45, 34A34, 65F60, 65K10

1. Introduction

In this study, we present a development and testing of suitable methods for the numerical simulation of the forward problem associated with the inverse problem of model parameter estimation. The key feature of a process under study is that two well separated time-scales are present, which is typical for quasiperiodic systems, i.e. there is a periodic behavior in a fast time-scale and some other phenomenon, evolving in much slower time-scale, to be identified.

Although the ultimate goal is to quantify and reduce the standard error (confidence interval) for the model parameter estimates, an accurate and fast numerical

DOI: 10.21136/panm.2022.15

approximation of the forward problem associated with the inverse problem is wanted. Here, we test two methods: (i) the multiple time-scales method, and (ii) the method of averaging. In summary, we highlight how singular perturbation methods serve the corresponding problem of model parameter estimation. On a (linear) model of an under-damped harmonic oscillator containing two state variables and two parameters, we demonstrate both the known pitfalls of the multiple time-scales method and the feasibility of the averaging method (the first order averaging) employed for a numerical simulation of the associated forward problem.

2. Preliminaries

2.1. State variables, model parameters, and governing equations

As follows, we present an ODE system in general (linearized) form describing the first order dynamics of a process depending on model parameters $p_1, \ldots, p_m$ and evolving in continuous time. A general form of a linear first order ODE system describing the dynamics of state variable vector $x \in \mathbb{R}^n$ is

$$\frac{\mathrm{d}\, x(t;p)}{\mathrm{d}t} = A(p)\, x(t;p) \tag{1}$$

with the square matrix $A(p)$ of order n. Vector $p \in \mathbb{R}^m$ contains all model parameters defining the system under study. Finally, there are the initial conditions $x_0 = x(t_0;p)$ which can be taken as system inputs.

Although our motivation for studying first order dynamical systems (1) dwells on a prospect to study inverse problems of parameter estimation arising from pharmacokinetics models, here, as a case study, we shall consider the governing dynamic equations for a weakly damped linear (harmonic) oscillator. In branch of mechanics, a mechanical oscillator under the influence of a linear restoring force and friction is described (using the Newton second law) by the ODE

$$m\,\ddot{y} = -k\,y - b\,\dot{y} + m\,g, \tag{2}$$

where y is the vertical position of the center of mass (the positive direction is upside down), $m > 0$ is the mass, $k > 0$ is the spring force constant, and $b > 0$ measures the strength of the damping. Setting the origin of y-axis at the equilibrium (i.e. at the position $y = 0$ the force of gravity is acting on the mass equalized by an adequate spring force), the governing equation of the system becomes

$$\ddot{y} + 2\delta\,\dot{y} + \omega_0^2\,y = 0, \tag{3}$$

where $\delta \equiv \frac{b}{2m}$ and $\omega_0 \equiv \sqrt{k/m}$ are a usual damping constant and an undamped oscillation frequency, respectively [cf., Equation (8) for $\delta = 0$ below].

We shall refer to the preceding equation (3) as the damped harmonic oscillator

equation. Let the initial conditions be[1]

$$y(0) = 1, \quad \dot{y}(0) = 0. \tag{4}$$

Further, inspired by [6], let us introduce the following transformation of state variables

$$x_1 = y, \quad x_2 = \frac{\dot{y}}{\omega_0}, \tag{5}$$

then the single ODE of the second order (3) can be described in the form of (1), where the state variables vector is

$$x(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix},$$

the corresponding form of matrix A is

$$A(p) = \begin{pmatrix} 0 & \omega_0 \\ -\omega_0 & -2\delta \end{pmatrix} = \omega_0 \begin{pmatrix} 0 & 1 \\ -1 & -2\frac{\delta}{\omega_0} \end{pmatrix}, \tag{6}$$

and the initial conditions are

$$x(0) = \begin{pmatrix} x_1(0) \\ x_2(0) \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}. \tag{7}$$

2.2. Exact solution of the system (3)-(4)

Assuming that $\delta < \omega_0$ and setting $\omega := \sqrt{{\omega_0}^2 - \delta^2}$, the exact solution of system (3) with initial conditions (4) is given by

$$y_{\mathrm{ex}}(t) = e^{-\delta t} \left(\cos \omega t + \frac{\delta}{\omega} \sin \omega t \right). \tag{8}$$

Remark 2.1. *If we define a scalar dimensionless quantity* $\varepsilon := \frac{\delta}{\omega_0} \ll 1$, *then* $\omega = \omega_0 \sqrt{1 - \varepsilon^2}$. *Furthermore, employing a usual scaling of time* t, *i.e.* $t_{scaled} := \omega_0 \, t$, *then the exact solution* (8) *has the form*

$$y_{\mathrm{ex}}(t_{\mathrm{scaled}}) = \mathrm{e}^{-\varepsilon \, t_{\mathrm{scaled}}} \left(\cos \sqrt{1 - \varepsilon^2} \, t_{\mathrm{scaled}} + \frac{\varepsilon}{\sqrt{1 - \varepsilon^2}} \sin \sqrt{1 - \varepsilon^2} \, t_{\mathrm{scaled}} \right). \tag{9}$$

As follows, the above single parameter form (9) *is used and the scaled time is (in seek of simplicity) expressed as* t, *which in fact is fulfilled for the value* $\omega_0 = 1$. *Moreover, given the transformation of state variables* (5), *it holds* $x_2 = \frac{\mathrm{d}\, x_1}{\mathrm{d}t_{\mathrm{scaled}}}$.

Remark 2.2. *Let us underline that the expression* (9) *is employed in Section 4.3 for the quantification of errors corresponding to different numerical approximation methods.*

The numerical values of two (only) model parameters used in equations (3)–(9) within some other related quantities are summarized in Table 1.

[1] This is done without loss of generality because the y coordinate can be scaled or normalized by the maximum value of $y(t)$, i.e. $y(0)$ value. Another usual setting of initial conditions for Equation (3) is $y(0) = 0$, $\dot{y}(0) = 1$.

Parameter	Formula	Value	Meaning
ω_0	$\sqrt{k/m}$	$1.0\ [\mathrm{s}^{-1}]$	undamped oscillation frequency
δ	$\frac{b}{2m}$	$10^{-3}\ [\mathrm{s}^{-1}]$	damping constant
ε	$\frac{\delta}{\omega_0}$	$10^{-3}\ [-]$	dimensionless damping constant

Table 1: Description and values of model parameters used in (3)–(9).

3. Perturbation theory and the averaging principle

Some dynamical systems can be represented by differential equations that are a small perturbation of an integrable problem.[2] Therefore, methods that allow to approximate the solutions of a perturbed problem, like $\dot{x} = f(t, x; \varepsilon)$, where $0 < \varepsilon \ll 1$, starting from the solutions of the unperturbed one (for $\varepsilon = 0$), are forming the perturbation theory, see e.g. [1, 2]. Instead of providing a detailed theoretical description of the singular perturbation (SP) techniques and their variants, for a class of systems defined by (1) we mention only two of them: (i) the method of multiple scales (MMS),[3] and (ii) the first order averaging.

3.1. Failure of the naive implementation of the MMS technique

Consider a first order expansion of a solution vector x, i.e. $x(t, \varepsilon) = x^{(0)}(t) + \varepsilon x^{(1)}(t) + O(\varepsilon^2)$. Then (1) reads

$$\frac{\mathrm{d}\left(x^{(0)}(t) + \varepsilon x^{(1)}(t)\right)}{\mathrm{d}t} = A(\varepsilon)\ \left(x^{(0)}(t) + \varepsilon x^{(1)}(t)\right).$$

For our weakly damped oscillator (1), with matrix A as (6), and after applying the scaling for both state variables and time, we have

$$\frac{\mathrm{d}\,x(t)}{\mathrm{d}t} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} x(t) + \begin{pmatrix} 0 & 0 \\ 0 & -2\varepsilon \end{pmatrix} x(t) \tag{10}$$

with initial conditions (4), i.e.

$$x^{(0)}(0) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad x^{(1)}(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Then we find for the leading order problem that

$$\frac{\mathrm{d}\,x^{(0)}(t)}{\mathrm{d}t} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} x^{(0)}(t) \tag{11}$$

[2] We say that a system of ODEs is integrable if its solutions can be expressed by analytic formulas up to inversions (by the implicit function theorem) or quadratures; we call the system non-integrable if this is not possible.

[3] Because of the inconvenience of method of multiple scales for numerical solution of a class of pharmacokinetic models, the setting of *solvability conditions* in frame of MMS is omitted here, for more details see Remark 3.2.

with a solution

$$x^{(0)}(t) = \begin{pmatrix} \cos t \\ -\sin t \end{pmatrix}.$$

At next order (for ε^1), we find that

$$\frac{\mathrm{d}\, x^{(1)}(t)}{\mathrm{d}t} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} x^{(1)}(t) + \begin{pmatrix} 0 & 0 \\ 0 & -2 \end{pmatrix} x^{(0)}(t), \tag{12}$$

which is in fact the ODE for a resonantly forced oscillator, and the solution for the first component is

$$x_1^{(1)}(t) = \sin t - t \cos t.$$

Therefore, a two-term (first order) approximate solution for the component x_1 is

$$x_1(t) = x_1^{(0)}(t) + \varepsilon\, x_1^{(1)}(t) = \cos t + \varepsilon \sin t - \varepsilon\, t \cos t = (1 - \varepsilon\, t) \cos t + \varepsilon \sin t. \tag{13}$$

Remark 3.1. *The above example clearly shows the failure when the naive implementation of the regular expansion is employed. On the result* (13) *it can be observed that the weakly damping system undergoes small changes of the amplitude of the oscillation (as* $(1 - \varepsilon\, t)$*) which cannot be longer negligible on a time scale* $\varepsilon^{-1} \sim 1\,000$*, i.e. when the so-called secular terms invalidate the expansion, see Fig.* 1*.*

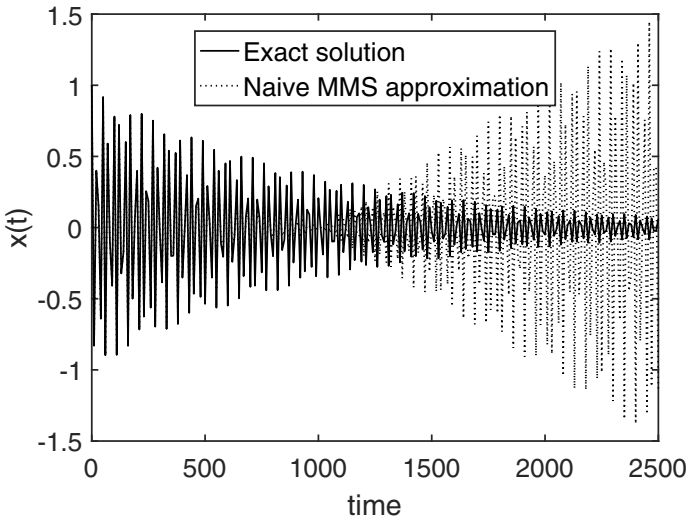


Figure 1: Comparison of the exact solution (9) (solid black curve) with the naive MMS approximation (13) (dotted curve).

Remark 3.2. *The correct employement of the MMS techniques resides in the use of what are known as solvability conditions in the formal derivation. It can be seen as a trick to avoid secular terms, and actually it is. Here, we reject this method because it poses big problem for the numerical implementation of the method.*

Remark 3.3. *There is some similarity of MMS to the Poincaré-Lindstedt method which provides a way to construct asymptotic approximations of periodic solutions. Nevertheless, the Poincaré-Lindstedt method cannot be used to obtain solutions that evolve aperiodically on a slow time-scale. Thus, the method of multiple scales, and mainly its WKB variant (WKB method requires the sate variable x to be 2π-periodic function of the "fast" time variable θ, see e.g. [2] and references within), is a more general approach.*

3.2. Averaging principle

The averaging principle consists in solving averaged equations, obtained by an integral average of the original equations (which can be put into a periodic standard form) over some angular variables; then we consider the solutions of the averaged equations as representative of the solutions of the original equations for a long time span (of the order $1/\varepsilon$). A review of the classical results on averaging methods in perturbation theory can be found in [2, 6]. Further, in sake of completeness, we announce (without proofs) two theorems dealing with approximation error estimation (published in [4]) and we present the approximate solution to (3)–(4) using the first order averaging (see Section 4.1).

Theorem 3.4. *Consider a system of ODEs for $x(t) \in \mathbb{R}^n$ which can be written in the standard form*

$$\dot{x} = \varepsilon f(x, t; \varepsilon). \tag{14}$$

Here, $f\colon \mathbb{R}^n \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}^n$ is a smooth function, 2π-periodic in "fast" variable t:

$$f(x, t + 2\pi, \varepsilon) = f(x, t, \varepsilon).$$

For $R > 0$ let $B_R(x_0) = \{x(t) \in \mathbb{R}^n;\ |x - x_0| < R\}$ and $M = \sup_{x \in B_R(x_0),\, t \in \mathbb{T}} |f(x, t)|$. Then there is a unique solution of the IVP,

$$x\colon\ (-T/\varepsilon, T/\varepsilon) \to B_R(x_0) \subset \mathbb{R}^n$$

that exists for $|t| < T/\varepsilon$, where $T = \frac{R}{M}$.

Theorem 3.5 (Krylov-Bogoliubov-Mitropolski)**.** *With the same notation as in the previous theorem: There exists a unique solution*

$$\bar{x}\colon\ (-T/\varepsilon, T/\varepsilon) \to B_R(x_0) \subset \mathbb{R}^n$$

of the averaged equation

$$\dot{\bar{x}} = \varepsilon \bar{f}(\bar{x}), \quad \bar{x}(0) = x_0, \tag{15}$$

where $\bar{f}(x) = \frac{1}{2\pi} \int_T f(x, t) \mathrm{d}t$. Moreover, there exist constants $\varepsilon_0 > 0$ and $C > 0$ such that for all $0 \le \varepsilon \le \varepsilon_0$

$$|x(t) - \bar{x}(t)| \le C\,\varepsilon \quad \text{for} \quad |t| \le T/\varepsilon. \tag{16}$$

4. Numerical example

On the ODE system (3)–(4) we now perform some numerical experiments. As mentioned, the MMS method produces naive numerical results, and thus we use the averaging principle and compare it with the backward Euler method. First, we introduce approximate solutions for both approaches.

4.1. Approximate solution to (3)–(4) using the first order averaging

Consider the ODE system (3)–(4) in the following form:

$$\ddot{y} + y = -2\varepsilon\,\dot{y}, \quad y(0) = 1, \quad \dot{y}(0) = 0.$$

Using the transformation

$$y = r\sin(t-\phi), \quad \dot{y} = r\cos(t-\phi), \tag{17}$$

the new variables r, ϕ satisfy the system

$$\dot{r} = \varepsilon\cos(t-\phi)(-2r\cos(t-\phi)) \equiv \varepsilon f_r(t),$$

$$\dot{\phi} = \varepsilon\tfrac{1}{r}\sin(t-\phi)(-2r\cos(t-\phi)) \equiv \varepsilon f_\phi(t).$$

Applying the averaging principle to the above equations leads to solving the system

$$\dot{\bar{r}} = \varepsilon\bar{f}_r, \quad \dot{\bar{\phi}} = \varepsilon\bar{f}_\phi, \tag{18}$$

where

$$\bar{f}_r = \frac{1}{2\pi}\int_0^{2\pi} f_r(t)\,\mathrm{d}t, \quad \bar{f}_\phi = \frac{1}{2\pi}\int_0^{2\pi} f_\phi(t)\,\mathrm{d}t. \tag{19}$$

Remark 4.1. *Clearly it holds $\bar{f}_r = -r$ and $\bar{f}_\phi = 0$ but unlike our simple case study, the integrals of functions f_r, f_ϕ in (19) cannot be usually computed easily. For this purpose we compute it numerically using the trapezoidal rule at points $0 = t_0, t_1, \ldots, t_n = 2\pi$.*

Let F_r, F_ϕ be the values of integrals of functions f_r, f_ϕ computed numerically, i.e.

$$F_r \approx \int_0^{2\pi}\cos^2(t)\,\mathrm{d}t, \quad F_\phi \approx \int_0^{2\pi}\sin(t)\cos(t)\,\mathrm{d}t.$$

Then

$$\bar{f}_r = -\frac{r}{\pi}F_r, \quad \bar{f}_\phi = -\frac{1}{\pi}F_\phi$$

and substituting into (18), the system of equations which approximates (3)–(4) is

$$\dot{\bar{r}} = -\varepsilon\frac{F_r}{\pi}\bar{r}, \quad \dot{\bar{\phi}} = -\varepsilon\frac{F_\phi}{\pi}.$$

The solution is

$$\bar{r} = C_r \exp\left(-\varepsilon \frac{F_r}{\pi}\, t\right), \quad \bar{\phi} = -\varepsilon \frac{F_\phi}{\pi}\, t + C_\phi,$$

where C_r and C_ϕ are some constants. Substituting into (17), the approximate averaging solution is then

$$y(t) = \bar{r}\sin(t - \bar{\phi}) = C_r \exp\left(-\varepsilon \frac{F_r}{\pi}\, t\right) \sin\left(t + \varepsilon \frac{F_\phi}{\pi}\, t - C_\phi\right),$$

$$\dot{y}(t) = \bar{r}\cos(t - \bar{\phi}) = C_r \exp\left(-\varepsilon \frac{F_r}{\pi}\, t\right) \cos\left(t + \varepsilon \frac{F_\phi}{\pi}\, t - C_\phi\right).$$

The constants C_r and C_ϕ will be obtained from initial conditions:

$$y(0) = C_r \sin(-C_\phi) = 1, \quad \dot{y}(0) = C_r \cos(-C_\phi) = 0,$$

which implies

$$\cos(-C_\phi) = 0 \quad \Rightarrow \quad C_\phi = \tfrac{3}{2}\pi \quad \text{and} \quad C_r = 1.$$

Finally, the approximate solution using the first order averaging has the form

$$\begin{aligned} y_{\text{av}}(t) &= \exp\left(-\varepsilon \frac{F_r}{\pi}\, t\right) \sin\left(t + \varepsilon \frac{F_\phi}{\pi}\, t - \frac{3}{2}\,\pi\right) \\ &= \exp\left(-\varepsilon \frac{F_r}{\pi}\, t\right) \cos\left((1 + \varepsilon \frac{F_\phi}{\pi}\right) t). \end{aligned} \tag{20}$$

4.2. Approximate solution to (3)–(4) using the backward Euler method

Transformation

$$x_1 = y, \quad x_2 = \dot{y}$$

leads to a system

$$\dot{x} = Ax, \quad x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad A = \begin{pmatrix} 0 & 1 \\ -1 & -2\varepsilon \end{pmatrix},$$

cf. (6), with initial conditions (7). The implicit backward Euler method leads to solving a linear system

$$(I - \Delta t A)x(t + \Delta t) = x(t).$$

The numerical solution to (3)-(4) is the first component, i.e.

$$y_{\text{be}}(t_j) = x_1(t_j), \quad j = 0, \ldots, m, \quad t_j = j\,\Delta t, \quad t_m = T. \tag{21}$$

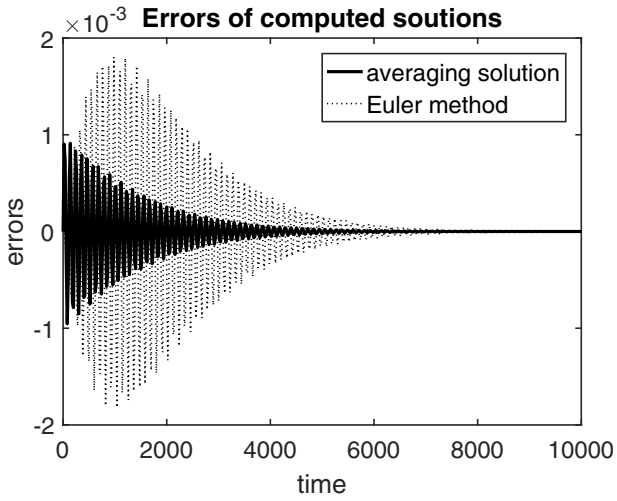


Figure 2: Errors of averaging solution and the backward Euler method from the exact solution, $t \in [0, 10\,000]$.

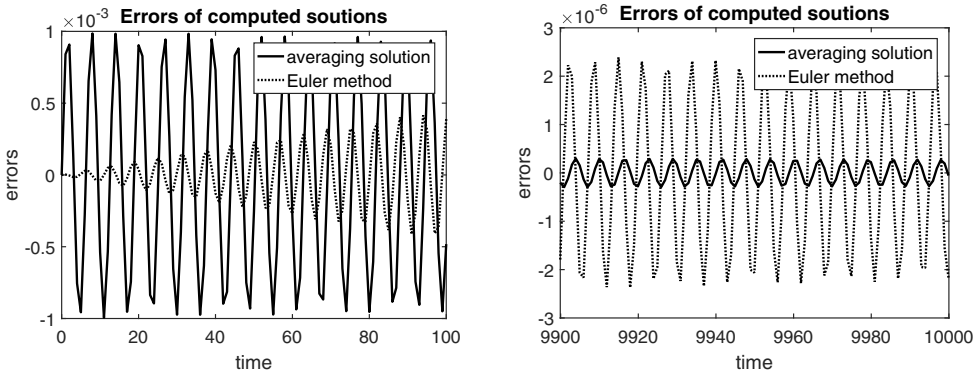


Figure 3: Zoom of errors. Left: $t \in [0, 100]$, Right: $t \in [9\,900, 10\,000]$.

4.3. Comparison of solution errors

Consider problem (3)–(4) and take the exact solution (9), the approximate averaging solution (20), and the solution obtained using the backward Euler method (21). Define the errors of computed solutions from the exact solution as follows:

$$\text{error}_{\text{av}}(t_j) = y_{\text{ex}}(t_j) - y_{\text{av}}(t_j), \quad \text{error}_{\text{be}}(t_j) = y_{\text{ex}}(t_j) - y_{\text{be}}(t_j), \tag{22}$$

where $t_j = j\Delta t,\ j = 0, \ldots, m,\ t_m = T$ (final time). For our numerical computations we consider the values

$$\varepsilon = 1.0\text{E-}3 \text{ (see Table 1)}, \quad \Delta t = 1.0\text{E-}5, \quad T = 10\,000.$$

Figure 2 shows the errors (22) of averaging solution and the backward Euler method from the exact solution for $t \in [0, 10\,000]$. Figures 3 show zooms. On the left there are errors for the initial time interval $t \in [0, 100]$, while on the right there are errors for the final time interval $t \in [9\,900, 10\,000]$.

The results show that the solution obtained using the averaging principle is really of order $C\,\varepsilon$ as stated in Theorem 3.5 (here for $\varepsilon = 10^{-3}$ we have $C \approx 1$) and the error envelope is decreasing from the beginning. On the other hand, the error envelope of the Euler method grows at the beginning until the time t reaches the value $1/\varepsilon$, i.e. $t \approx 1\,000$. This maximum value is even for a relatively small step $\Delta t = 10^{-5}$ twice the maximum value of averaging envelope ($C \approx 2$). For $t > 1/\varepsilon$ the error envelope of the Euler method finally tends to zero. Thus, the averaging principle outperforms the Euler method.

5. Conclusion

We showed the behavior of the **Method of Multiple (time)Scales** (MMS) and mainly the **Averaging method** to approximate the solutions of perturbation problems. The Naïve implementation of MMS generates wrong results due to presence of secular terms which cannot be avoided when using a numerical approach. On the other hand, the averaging method gives satisfactory results, the error is of order $C\,\varepsilon$ (as predicted by the KBM theorem), and the results are better than those using the Euler method.

Acknowledgement

The work of Štěpán Papáček was supported by the Czech Science Foundation through the research grant project No. 21-03689S. The work of Ctirad Matonoha was supported by the long-term strategic development financing of the Institute of Computer Science (RVO:67985807).

References

[1] Kokotovic, P., Khalil, H. K., O'Reilly, J.: Singular perturbation methods in control: Analysis and design. Classics in Applied Mathematics, xv+366 p., 1999.

[2] Murdock, J.: Perturbation Methods. In: digital Encyclopedia of Applied Physics, Wiley-VCH Verlag GmbH & Co. KGaA (Ed.), 2015. https://doi.org/10.1002/3527600434.eap315.pub3

[3] Liu, C.-S., Chen, Y.-W.: A simplified Lindstedt-Poincaré method for saving computational cost to determine higher order nonlinear free vibrations. Mathematics 9, no. 23, 3070, 2021. https://doi.org/10.3390/math9233070

[4] Sanders, J. A., Verhulst, F., Murdock, J.:, Averaging methods in nonlinear dynamical systems. Springer, XXIV+434 p., 2007. https://doi.org/10.1007/978-0-387-48918-6

[5] Duintjer Tebbens, J., Matonoha, C., Matthios, A., Papáček, Š.: On parameter estimation in an *in vitro* compartmental model for drug-induced enzyme production in pharmacotherapy. Applications of Mathematics, Vol. 64, p. 253–277, 2019.

[6] Khalil, H. K.: Nonlinear systems, 3rd Edition. Prentice Hall, 750 p., 2002.

Programs and Algorithms of Numerical Mathematics 21
J. Chleboun, P. Kůs, J. Papež, M. Rozložník, K. Segeth, J. Šístek (Eds.)
Institute of Mathematics CAS, Prague 2023

WILDFIRES IDENTIFICATION: SEMANTIC SEGMENTATION USING SUPPORT VECTOR MACHINE CLASSIFIER

Marek Pecha[1,2], Zachary Langford[3], David Horák[1,2], Richard Tran Mills[4]

[1] VŠB–Technical University of Ostrava
17. listopadu 2172/15, Ostrava–Poruba, Czech Republic
[2] Institute of Geonics, Czech Academy of Sciences
Studentská 1768/9, Ostrava–Poruba, Czech Republic
marek.pecha@vsb.cz, david.horak@vsb.cz
[3] Oak Ridge National Laboratory
1 Bethel Valley Rd, Oak Ridge, TN, United States
langfordzl@ornl.gov
[4] Argonne National Laboratory
9700 S Cass Ave, Lemont, IL, United States
rtmills@anl.gov

Abstract: This paper deals with wildfire identification in the Alaska regions as a semantic segmentation task using support vector machine classifiers. Instead of colour information represented by means of BGR channels, we proceed with a normalized reflectance over 152 days so that such time series is assigned to each pixel. We compare models associated with $\ell 1$-loss and $\ell 2$-loss functions and stopping criteria based on a projected gradient and duality gap in the presented benchmarks.

Keywords: wildfire identification, semantic segmentation, support vector machines, distributed training

MSC: 68T09, 68T45, 68W15

1. Introduction

Global climate change is increasing the frequency and intensity of ecological disturbance; this is particularly true in high latitudes, where projects such as the NASA ABoVE project (`https://above.nasa.gov`) are working to understand the effects of increased climate-driven disturbances. Wildfires are one important source of disturbance, and can significantly affect forest carbon balance. Despite their importance, however, it can be difficult to accurately quantify the effects of wildfire in places such as boreal forests that are far from human habitation and infrastructure. Data from remote sensing platforms and observatory networks can be of great use of this task,

DOI: 10.21136/panm.2022.16

but these data sets can be vast, and analyzing them can require powerful computing resources and tools that are designed to fully utilize them.

Popular methods are based on machine learning approaches including deep learning [8], where U-Net architecture or inception networks are typically used. In this paper, we discuss an alternative approach for wildfire identification in the Alaska regions using semantic segmentation that support vector machine classifiers are exploited. Instead of colour information, we assign changes of normalized reflectance over time to each pixel so that corresponding attributes are represented by time series with 8-day period. Pixels are then categorized using the Monitoring Trends in Burn Severity product. Additionally, we study the influence of stopping criteria on model performance and training time on benchmarks presented in Section 3.2.

2. Support Vector Machines

Support Vector Machines is a set of methods belonging to supervised learning algorithms used for classification, regression, or outliers detection. Since wildfire identification is essentially a binary classification task, i.e. we have to decide if an area is affected by fire or not, we will focus on formulations associated with classification approaches employing SVMs in this paper. Considering underlying structures related to SVMs, we can see them as a single perceptron that finds a learning function (called model in the machine learning community) maximizing a geometric margin between (training) samples and a discriminant hyperplane. This implicit ability guarantees a generalization performance of the model, which can be described by means of a particular case of the Tikhonov regularization in the following form:

$$\underset{f\in\mathscr{H}}{\arg\min}\ \ m^{-1}\sum_{i=1}^{m} V\left(y_i, f\left(\boldsymbol{x}_i\right)\right)^2 + \lambda\|f\|_{\mathscr{H}}^2, \tag{1}$$

where $\mathscr{H}$ is a hypothesis space of functions, $\|\cdot\|_{\mathscr{H}}$ is a norm on the hypothesis space, $f\colon \mathbb{R}^n \to Y$ denotes mapping data (m training samples) to a label space, $V\colon Y \to Y$ is a loss function, and $\lambda \in \mathbb{R}$ is a regularization parameter such that $\lambda = \frac{1}{2C}$. Moreover, this theoretical framework provides us with an explanation related to the regularization perspective of the SVM models so that a trade-off between bias and variance is driven by parameter C.

In the following part of this paper, we introduce certain C-SVM formulations that are associated with classification tasks concerning non-linearly separable (training) samples and their relaxed-bias versions, where a bias term is considered as a scaled parameter and included in an optimization problem by means of augmenting the normal vector of a hyperplane $\boldsymbol{w}$ and samples with an additional dimension.

2.1. Soft maximum-margin classifier

Let us start with a standard SVM formulation introduced by Vapnik et al. in [3]. It was initially developed as a supervised binary classifier, i.e. an algorithm that determines a function (model), which maps a training sample to a label (related

to 2 categories in this case) such that it adapts itself to unseen data drawn from the same distribution as the training ones. This essential model ability is called generalization.

To describe the training phase of the SVM classifier more in detail, let us firstly denote the training data set as follows:

$$T := \{(\boldsymbol{x}_1, y_1)\,,\ (\boldsymbol{x}_2, y_2)\,,\ \ldots,\ (\boldsymbol{x}_m, y_m)\}, \tag{2}$$

where $\boldsymbol{x}_i \in \mathbb{R}^n$ ($n \in \mathbb{N}$) is an i-th sample and $y_i \in \{-1, 1\}$ is its label, m is a number of training samples. Further, let us consider that the samples are linearly separable, i.e. it exists a separating hyperplane between the clusters of samples belonging to these two categories. A model of a linear SVM is then represented in the form of a maximum-margin hyperplane H so that:

$$H = \langle \boldsymbol{w}, \boldsymbol{x} \rangle - \hat{b}, \tag{3}$$

where $\boldsymbol{w}$ is a normal vector of the hyperplane H, and $\hat{b} = \frac{b}{\|\boldsymbol{w}\|}$ is a scalar called a bias term that determines an offset in a direction of $\boldsymbol{w}$, or $-\boldsymbol{w}$ in a case when $\hat{b}$ is negative. Let us denote a bias $\hat{b}$ as b for a more convenient notation in equations in the following text. Remark that the maximum margins are defined by means of locations associated with support vectors, and the width between these margins is equal to $\frac{2}{\|\boldsymbol{w}\|}$.

Maximizing the distance d corresponds to regularization of the weights $\boldsymbol{w}$, which is basically the prevention of overfitting a model to the training data set T. Regarding the constraints arising from geometric margins, we can write an optimization problem for finding a normal vector $\boldsymbol{w}$ and a bias b as follows:

$$\underset{\boldsymbol{w},\ b}{\arg\min}\ \frac{1}{2}\langle \boldsymbol{w}, \boldsymbol{w} \rangle\ \text{ s.t. }\ \begin{cases} y_i\left(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle - b\right) \geq 1, \\ i \in \{1, 2, \ldots, m\}, \end{cases} \tag{4}$$

the constraint $y_i\left(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle - b\right) \geq 1$ can be interpreted so that all training categorical samples must lie on or above corresponding margins equal to -1 and 1, respectively. Note, a solution of the optimization problem (4) exists only when the training samples T are linearly separable. To sort out the separability issue, we can exploit the soft-margin SVM [3]. An idea beyond the approach is based on adding an auxiliary (regularization) term to (4), particularly, $C\sum_{i=1}^{m}\xi_i$ [1], and, also, an additional relaxation of the constraints related to the margins such that:

$$\underset{\boldsymbol{w},\ b,\ \xi_i}{\arg\min}\ \frac{1}{2}\langle \boldsymbol{w}, \boldsymbol{w} \rangle + C\sum_{i=1}^{m}\xi_i\ \text{ s.t. }\ \begin{cases} y_i\left(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle - b\right) \geq 1 - \xi_i, \\ \xi_i \geq 0,\ i \in \{1, 2, \ldots, m\}, \end{cases} \tag{5}$$

[1] The term $C\sum_{i=1}^{m}\xi_i$ regularises misclassification errors and restricts the complexity of the classifier in sense of overfitting a classification model.

where $\xi_i := \max\{0, 1 - [\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle - b]\}$. Essentially, the function quantifies the error between the predicted and correct sample classification $\boldsymbol{x}_i$. If sample $\boldsymbol{x}_i$ is correctly classified, a value of the hinge loss function equals 0. In order of sample misclassification, a value of hinge loss function is proportional to the distance between the respective margin and a misclassified sample.

The parameter C is a user-defined penalty, which determines the influence associated with the misclassification of samples on the objective function. Generally, a higher value of C increases the importance of minimizing the hinge loss functions ξ_i and also maximizing $\|\boldsymbol{w}\|$. This leads to minimizing the width of the margin and may cause overfitting of a classifier to a training data set consequently. It means a model has a high variance. A smaller value of the penalty C results in a wider margin that may cause a large number of misclassifications, i.e. a high bias of a model[2]. The goal is to find a reasonable value of C such that a resulting model balances a bias-variance tradeoff. Typically, the value is determined using hyperparameter optimization techniques, e.g. grid-search combined with cross-validation.

To reduce the number of unknowns and employ our approach based on a deterministic approach that uses the MPRPG [4] as an underlying solver, it stands for Modified Proportioning with Reduced Gradient Projection, we can modify the primal formulation (5) so that it turns into an optimization problem with the following structure:

$$\boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha} \in \Omega} \frac{1}{2} \boldsymbol{\alpha}^T \boldsymbol{A} \boldsymbol{\alpha} - \boldsymbol{b}^T \boldsymbol{\alpha}, \tag{6}$$

where Ω is a convex closed set defined by means of box constraints $\Omega := \{\boldsymbol{\alpha} \in \mathbb{R}^m \mid \boldsymbol{u} \leq \boldsymbol{\alpha} \leq \boldsymbol{l}\}$. Practically, we can obtain a formulation analogous to the structure (6) by dualizing primal formulation (5) such that:

$$\arg\min_{\boldsymbol{\alpha}} \ \frac{1}{2} \boldsymbol{\alpha}^T \underbrace{\boldsymbol{Y}^T \boldsymbol{K} \boldsymbol{Y}}_{\boldsymbol{G}} \boldsymbol{\alpha} - \boldsymbol{\alpha}^T \mathbf{1} \ \text{ s.t. } \begin{cases} \boldsymbol{o} \leq \boldsymbol{\alpha} \leq C\mathbf{1}, \\ \boldsymbol{y}^T \boldsymbol{\alpha} = 0, \end{cases} \tag{7}$$

where $\boldsymbol{Y} = diag(\boldsymbol{y})$, $\boldsymbol{y} = [y_1,\ y_2,\ \ldots,\ y_m]^T$, and $\mathbf{1} = [1,\ 1,\ \ldots,\ 1] \in \mathbb{R}^m$. $\boldsymbol{K} \in \mathbb{R}^{m \times m}$ is the SPS (Symmetric Positive Semi-definite) matrix of inner products called the Gram matrix such that $\boldsymbol{K} = \boldsymbol{X}^T \boldsymbol{X}$, where $\boldsymbol{X} = [\boldsymbol{x}_1,\ \boldsymbol{x}_2,\ \ldots, \boldsymbol{x}_m]$. $\boldsymbol{G}$ denotes the Hessian matrix, which is SPS either. Exploiting a derivative of the Lagrangian with respect to $\boldsymbol{\xi}$, we can determine the vector of Lagrange multipliers $\boldsymbol{\beta}$ so that $\boldsymbol{\beta} = C\mathbf{1} - \boldsymbol{\alpha}$, thus $\boldsymbol{\beta}$ does not occur in (7). This formulation is called dual ℓ1-loss.

For obtaining a solution of the original (primal) problem, we introduce dual to primal reconstruction formulas as follows:

$$\boldsymbol{w} = \boldsymbol{X} \boldsymbol{Y} \boldsymbol{\alpha}, \tag{8}$$

[2]In this case, the term bias corresponds to a systematic error arising from wrong assumptions that may lead to missing relevant relations between features and labels caused by means of a low capability of a model.

associated with the normal vector of the separating hyperplane, and the bias is reconstructed by means of:

$$b = \frac{1}{\operatorname{card}(J)} \left(\boldsymbol{X}_{*J}^T \boldsymbol{w} - \boldsymbol{y}_J \right) \mathbf{1}_J^T, \tag{9}$$

where $J = \{i \mid 0 < \alpha_i < C,\ i = 1, 2, \ldots, k\}$ is the support vector index set, $\operatorname{card}(J)$ presents its cardinality, $\boldsymbol{X}_{*J}$ denotes the submatrix of the matrix $\boldsymbol{X}$ with the column indices belonging to J; $\boldsymbol{y}_J$ and $\mathbf{1}_J$ are subvectors of the vectors $\boldsymbol{y}$ and $\mathbf{1}$, respectively. Using the reconstructed normal vector $\boldsymbol{w}$ and bias b, we set the decision rule:

$$\operatorname{sgn}\left(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b\right) = \begin{cases} +1 \ldots\ \boldsymbol{x}_i \in \text{Class A}, \\ -1 \ldots\ \boldsymbol{x}_i \in \text{Class B}. \end{cases} \tag{10}$$

2.2. Hessian matrix regularization

The Hessian matrix $\boldsymbol{G}$ corresponding to the dual formulation (7) is SPS, which implies the underlying optimization problem has a non-unique solution. In this subsection, we modify the primal formulation (5) in such a way that the Hessian in dual formulation becomes SPD (Symmetric Positive Definite) [10, 9]. It implies that the resulting optimization problem is strictly convex, and its solution is unique. An idea beyond the adjustment is based on substitution $\ell 1$-norm of loss function by the $\ell 2$-norm, i.e. the squared loss function, in the objective function so that (7) results into the following form:

$$\underset{\boldsymbol{w},\ b,\ \xi_i}{\arg\min} \frac{1}{2}\langle \boldsymbol{w}, \boldsymbol{w} \rangle + \frac{C}{2} \sum_{i=1}^{m} \xi_i^2 \ \text{ s.t. } \begin{cases} y_i \left(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b \right) \geq 1 - \xi_i, \\ i \in \{1, 2, \ldots, m\}. \end{cases} \tag{11}$$

Analysing the formulation above, we can simply observe the term that quantifies misclassification error $\sum_{i=1}^{m} \xi_i^2 \geq 0$. Therefore, we do not consider $\xi_i \geq 0$ as a constraint. The formulation (11) is called the primal $\ell 2$-loss SVM. As in the case of the $\ell 1$-loss SVM, we derive a dual formulation. Using the Lagrange duality and evaluating the Karush–Kuhn–Tucker conditions, the primal formulation (11) transforms into the dual one so that for any $C > 0$:

$$\underset{\boldsymbol{\alpha}}{\arg\min}\ \frac{1}{2} \boldsymbol{\alpha}^T \left(\boldsymbol{G} + C^{-1} \boldsymbol{I} \right) \boldsymbol{\alpha} - \boldsymbol{\alpha}^T \mathbf{1} \ \text{ s.t. } \begin{cases} \mathbf{0} \leq \boldsymbol{\alpha}, \\ \boldsymbol{y}^T \boldsymbol{\alpha} = \mathbf{0}. \end{cases} \tag{12}$$

While the Hessian $\boldsymbol{G}$ is regularized by a matrix $C^{-1}\boldsymbol{I}$, it avoids linear dependency of columns also arising from possible multicollinearity of the training samples. Then, the matrix becomes full-rank SPD. The optimization problem and the quality of its solution are practically data-driven, i.e. highly dependent on the data nature. Therefore, we can say precisely that the associated optimization problem could be more computationally stable, and a convergence rate of an underlying solver could

be faster than in the case of the $\ell 1$-loss SVM. On the other hand, $\ell 1$-loss SVM could produce a sparse and more robust model in the sense of performance score. Then, we adapt the support vector index set J such that:

$$J = \{i \mid 0 < \alpha_i,\ i = 1, 2,\ \ldots, k\} \tag{13}$$

for the reconstruction formulas (8), (9) related to normal vector $\boldsymbol{w}$ of hyperplane H and bias b, respectively.

2.3. Relaxed-bias approaches

The standard soft-margin SVM solves the problem of finding a classification model in the form of the maximal-margin hyperplane (3). In the case of the relaxed-bias classification [7], we do not consider the bias b in a classification model. However, we include it into the problem by means of augmenting the vector $\boldsymbol{w}$ and each sample $\boldsymbol{x}_i$ with an additional dimension so that $\widehat{\boldsymbol{w}} \leftarrow \begin{bmatrix} \boldsymbol{w} \\ B \end{bmatrix}$, $\widehat{\boldsymbol{x}}_i \leftarrow \begin{bmatrix} \boldsymbol{x}_i \\ \gamma \end{bmatrix}$, where $\gamma \in \mathbb{R}^+$ is a user-defined variable, which is typically set to 1. Let $p \in \{1, 2\}$ then the problem of finding a hyperplane $\widehat{H} = \langle \widehat{\boldsymbol{w}}, \widehat{\boldsymbol{w}} \rangle$ can be formulated as a constrained optimization problem in the following primal formulation:

$$\underset{\widehat{\boldsymbol{w}},\ \xi_i}{\arg\min}\ \frac{1}{2}\langle \widehat{\boldsymbol{w}}, \widehat{\boldsymbol{w}} \rangle\ +\ \frac{C}{p}\sum_{i=1}^{n} \widehat{\xi}_i^{\,p}\ \text{ s.t. }\ \begin{cases} y_i \langle \widehat{\boldsymbol{w}}, \widehat{\boldsymbol{x}}_i \rangle \geq 1 - \widehat{\xi}_i, \\ \widehat{\xi}_i \geq 0 \ \text{ if } p = 1,\ i \in \{1, 2, \ldots, n\}, \end{cases} \tag{14}$$

where $\widehat{\xi}_i = \max\{0,\ 1 - y_i \langle \widehat{\boldsymbol{w}}, \widehat{\boldsymbol{x}}_i \rangle\}$ is the hinge loss function related to augmented samples $\widehat{\boldsymbol{x}}_i$. Generally, we can say the minimizer associated with formulation (14) corresponding to a rotation of the separating hyperplane $\widehat{H} \in \mathbb{R}^n$ in a one-dimension higher feature-space $\mathbb{R}^{n+1}$ such that the maximizing of geometric margins are satisfied.

3. Wildfire identification as semantic segmentation task

Semantic segmentation is a computer vision task for which most recent methods are based on deep learning approaches, where neural networks of U-Net type architectures are typically used. Actually, semantic segmentation is associated with image classification at a pixel level. It means that every pixel is assigned to a category such that an image segmentation mask is created. In common semantic segmentation, labelled colour images with the BGR (blue-green-red) channel order are used as inputs. The pre-trained encoder of the U-Net extracts features and patterns from spatial images, and the decoder projects these lower resolution feature onto the pixel space in higher resolution to get a dense classification.

We show up an alternative approach that exploits a spectral reflectance corrected for the atmospheric condition instead of colour information. An essential idea of these corrections follows up simulating the propagation of electromagnetic

waves in a geogas system to obtain surface reflectance without emission, e.g. remove a contribution of atmospheric aerosol scattering.

To estimate a spectral surface reflectance corresponding to 500 m spatial resolution at a pixel, we use the MODIS (Moderate Resolution Imaging Spectroradiometer) instrument that data was extracted by the Google Earth Engine running at cloud https://earthengine.google.com. Essentially, the reflectance is a ratio of reflected energy to incident radiation $\frac{\phi_r}{\phi}$ as a function of the wavelength. The MODIS product called MOD09A1 (https://modis.gsfc.nasa.gov/data/dataprod/mod09.php) provides 7 bands associated with this electromagnetic spectrum ranging from 459 nm to 2155 nm as an 8-day composite. To describe a region affected by fire, we study changes in normalized reflectance over time periods so that features corresponding to each pixel are represented by time series related to the 7 bands mentioned above with an 8-day period. The pixels are then categorized using boundaries collected from Monitoring Trends in Burn Severity (https://www.mtbs.gov/). Such samples are being classified using SVM implemented in our in-house software PermonSVM.

3.1. PermonSVM: Classification tool based on PETSc framework

The PermonSVM package [11] is a part of the PERMON toolbox . This toolbox is designed for usage in a distributed environment containing hundreds or thousands of computational cores. Technically, it is an extension of the core package called PermonQP [6], from which it inherits environment basic structures, initialization routines, a build system, and utilizes computational routines implemented in the core package PermonQP. Programmatically, a core functionality associated with the PERMON toolbox is written on top of the PETSc framework [1]. It follows the same design and coding style that makes it easy to use for anyone familiar with PETSc.

PermonSVM currently supports parallel reading of the SVMLight, HDF5, and PETSc binary file formats, solutions of more than 4 problem formulations of the related classification problems, k-fold and stratified k-fold cross-validation. The underlying QP problem related to SVM with implicitly represented the Hessian matrix, in which the Gram matrix $\boldsymbol{X}^T\boldsymbol{X}$ is not assembled, and is computed by means of solvers provided by the PermonQP package or PETSc framework. All PERMON modules are developed as open-source software under the BSD–2–Clause license.

3.2. Benchmarks

We present results associated with state-of-the-art investigating wildfire detection so that data was collected and processed in a way already mentioned above. In our experiments, we study wildfires in the Alaska regions in 2004. The wildfires across Alaska are the dominant disturbance, and creating frameworks for quantification is important to long-term scientific projects such as the U.S. Department of Energy project Next Generation of Arctic Ecosystem Experiments. The 2004 Alaska wildfire season was the worst on record in the U.S. state of Alaska in terms of area burned (27,000 km^2). Looking at Table 1, our toy data set used to present

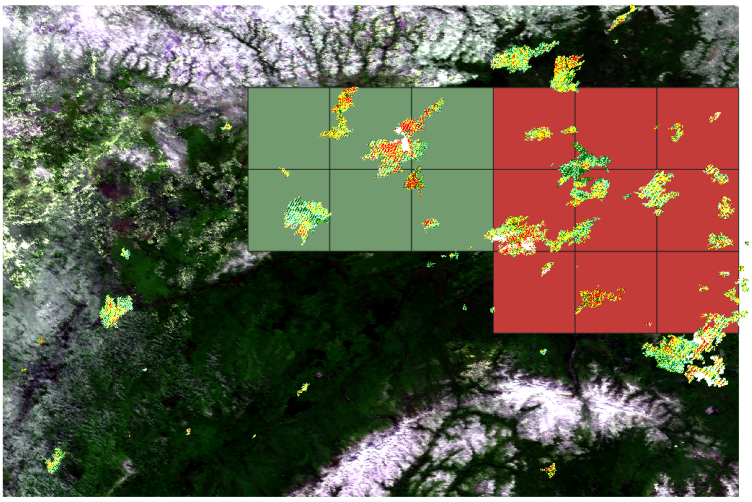


Figure 1: Areas in Alaska affected by wildfires that we model in our experiments. Red squares represent the training data set and green ones are related to test data set. Data are accumulated over 152 days from May to Semptember

state-of-the-art results contains $500,000$ samples split into training and test data sets consisting of $360,000$ and $240,000$ samples, respectively. These samples are associated with changing reflectance over 152 days from May to September.

We computed the following semantic segmentation results on KAROLINA, which is a combination of HPE Apollo 2000 and Apollo 6500 systems used for HPC workloads such as AI and other data-intensive applications, for example.

mod09ak_2004	#Wildfires	#Background	#Attributes
Training	$46,851$ (13.01%)	$313,149$ (86.99%)	133
Test	$28,351$ (11.81%)	$211,649$ (88.19%)	133

Table 1: The mod09ak_2004 data set description related to training and test ones. Proportions of classes in the data sets are pointed out as percents.

A critical part of any data-related pipeline is associated with stopping criteria. Choosing the right strategy to terminate an underlying optimization solver influences the quality of a resulting model. In our experiments, we explored an optimization solver called MPRGP employed in our classification problems that models were computed employing relaxed-bias formulations for $\ell 1$ and $\ell 2$ loss types presented in Section 2.3. An expansion of an active set was performed using a projected conjugate (CG) gradient, and $\Gamma = 100$ was set to determine proportionality. Misclassification errors were penalized with $C = 1$, i.e. a default value. A standard stopping criterion used in MPRGP involves a norm of projected gradient $\|g^p\|$ compared with a relative norm of a dual right-hand side $\boldsymbol{b}_{\text{dual}}$ as follows:

$$\|\boldsymbol{g}^p\| \leq \epsilon \|\boldsymbol{b}_{\text{dual}}\|. \tag{15}$$

However, this terminating condition does not take into account model quality. A reasonable approach could be based on monitoring a loss function and including it in a stopping criterion. In the case of SVM, we consider a specific type of a loss function called a hinge loss function $\xi := \max\{0, 1 - [\langle \boldsymbol{w}, \boldsymbol{x} \rangle - b]\}$ defined in Section 2.1 and this term is incorporated in a primal functional. Moreover, we can prove there is no gap between primal and dual functional at its optimal solution for the case of the $\ell 2$-loss SVM formulation. It holds a strong duality. Regarding these properties, we can use stopping criteria based on a duality gap for the $\ell 2$-loss SVM as follows:

$$|p\left(\boldsymbol{w}, b, \boldsymbol{\xi}\right) - d(\boldsymbol{\alpha})\,| \leq \epsilon |p\left(\boldsymbol{w}, b, \boldsymbol{\xi}\right)|, \tag{16}$$

where

$$p\left(\boldsymbol{w}, b, \boldsymbol{\xi}\right) = \frac{1}{2}\left\langle \widehat{\boldsymbol{w}}, \widehat{\boldsymbol{w}} \right\rangle \; + \; \frac{C}{2}\sum_{i=1}^{n} \widehat{\xi}_i^2, \tag{17}$$

and

$$d\left(\boldsymbol{\alpha}\right) = \frac{1}{2}\boldsymbol{\alpha}^T\left(\boldsymbol{G} + C^{-1}\boldsymbol{I}\right)\boldsymbol{\alpha} - \boldsymbol{\alpha}^T\mathbf{1} \tag{18}$$

are a primal and a dual functional related to relaxed-bias $\ell 2$-loss SVM formulations, respectively; ϵ represents a relative tolerance. The attained results are summarized in Table 2 and Table 3.

Dataset	Loss	Stop. criteria	Hessian mult.	Loss val.	Train. time [s]
mod09ak_2004	$\ell 1$	(15)	2962	2.28e4	22.67
	$\ell 2$	(15)	1025	3.03e4	6.96
		(16)	1029	3.00e4	15.60

Table 2: Attained results using 64 MPI processes (KAROLINA). Solver: MPRGP so that an expansion step is performed using the projected CG step, $\Gamma = 100$ in proportion criterion, a relative tolerance ϵ was set to 0.1; penalty $C = 1$.

The overall performance of attained models does not significantly differ as measured by the $F1$ score, which is a harmonic mean of precision and sensitivity, as presented in Table 3. However, we can see that some models perform slightly better than others when we compare them using other metrics. Analyzing the influence of the proposed stopping criteria based on the duality gap on model scores, we can see that the $\ell 2$-loss model, when the training process was terminated using the condition (16), behaves slightly better both precision and sensitivity scores on a test data set than the $\ell 2$-loss model trained employing the MPRGP solver stopped by means of the terminate condition (15).

Dataset	Loss	Stopping criteria	precision [%]	sensitivity [%]	F1
mod09ak_2004	$\ell 1$	(15)	84.12	94.58	0.89
	$\ell 2$	(15)	83.58	92.81	0.89
		(16)	85.33	93.13	0.89

Table 3: Influence of stopping criteria on the model performance scores on the test data set.

As we mentioned in Section 2.2, the $\ell 1$-loss model could be a more robust in the sense of its performance than the one based on the $\ell 2$-loss function. It could be ambitious to make such a conclusion merely by looking at the performance scores since we can see that a precision score is higher for $\ell 2$-loss and, on the other side, sensitivity is higher for $\ell 1$ loss. Nevertheless, it differs in the value of loss functions, which represent overall misclassification errors, pointed out in Table 2. From this table, we can easily see that the $\ell 1$-loss-based model generalizes a training data set better than the $\ell 2$-loss-based one. Assuming the training times of each model, we can see that evaluating the time of stopping criteria (16) is time-consuming and almost 2 times slower than for (15), and training the $\ell 1$-loss model is nearly 3.3 times slower than for $\ell 2$-loss trained to employ the MPRGP solver terminated using the condition (15). From the observations above, it seems the $\ell 2$-loss model that its training was stopped exploiting the stopping criteria (16) could be a good trade-off among $\ell 1$-loss and $\ell 2$-loss models, when the stopping condition (15) was used in during the models training.

Dataset	Loss	$\sum$ Hes. mult.	Loss val.	Train. time [s]
mod09ak_2004	$\ell 1$	34622	1.80e5	73.82
	$\ell 2$	51967	2.24e5	112.28

Table 4: Solutions related to the complete SVM formulations using SMALXE + MPRGP. A default stopping condition is used. Results are attained using 64 MPI processes on KAROLINA. Setting of an inner solver: $\Gamma = 100$, a relative tolerance $\epsilon_{\text{inner}} = 0.1$; $\epsilon_{\text{outer}} = 1\text{e}-2$ and divtol $=$ 1e10 for an outer loop (SMALXE). Misclassification penalty $C = 1$.

The attained models presented above can be viewed as solutions related to a special case of the Tikhonov regularization (1) such that a bias term b is relaxed. This approach simplifies the SVM formulations (7), (12), i.e. the complete SVM formulations with bounds and equality constraints. It leads to problems that are numerically cheaply to solve than the original ones. We demonstrate computational demands on training models employing the complete SVM formulations in the following numerical experiments. We employed the Semimonotonic augmented Lagrangian (SMALXE) algorithm [5] that is "pass-through" solver taking care of equality constraints (a default stopping condition for SMALXE is used in the following numerical experiments). By this approach, we splitted (7), or (12) for $\ell 2$-loss case, into two

sub-problems such that an equality constraint and bounds are handled separately, one after another. An outer loop is performed using the augmented Lagrangians and bound constrained optimization problem is computed by means of an inner solver – MPRGP in our case. The results are summarized in Table 4 and Table 5.

Looking at elapsed times presented in Table 4, we can see that training a model is 3.26 times slower for the complete $\ell 1$-loss formulation (7) than in case of a relaxed formulation of this problem (14) (for $p = 1$). Moreover, we can observe that a value of a loss function (a quantification of misclassification error) is 7.89 times higher than its relaxed version. It is similar to training a model employing the complete $\ell 2$-loss formulation when a default stopping condition is exploited. A training time is 16.1 times slower, and a value associated with a loss function is 7.39 times higher.

Dataset	Loss	precision [%]	sensitivity [%]	F1
mod09ak_2004	$\ell 1$	82.80	96.18	0.89
	$\ell 2$	82.98	95.63	0.89

Table 5: The best perfomance scores of models trained employing the complete SVM formulations (on the test data set).

The performance scores of models on the test data set are summarized in Table 5. They do not significantly differ from the scores attained employing the relaxed versions of the SVM formulation in Table 3; however, a true positive rate (sensitivity) is slightly higher. It means that models identify fire occurrences (true positives) better than the ones with relaxed bias at the cost of decreasing precision, i.e. a false positive rate.

Dataset	Solver	precision	sensitivity	F1
mod09ak_2004	XGBoost	97.05	89.00	0.93

Table 6: Results attained using the XGBoost solver.

We compared the performance scores of attained classification models employing PermonSVM with a model trained by means of the XGBoost (eXtreme Gradient Boosting) solver [2]. It is based on a boosted tree method. The results are presented in Table 6. The overall scores measured by means of F1 score are higher for a model trained by XGBoost. However, PermonSVM produces models with higher sensitivity over precision, while the XGBoost model has a higher precision over sensitivity. This means PermonSVM models perform better at predicting positive events (wildfires) over determining pixel areas that are non-affected by fire. Predicting more false negatives (FN) over false positives (FP) would be more acceptable for natural hazard applications than the other way around.

4. Conclusions

We studied state-of-the-art semantic segmentation for wildfire identification in the Alaska regions so that a classification part was based on the SVM methods implemented in the toolbox PERMON for distributed computing, specifically in an extension called PermonSVM. Instead of BGR channels associated with pixel colour information, we assigned time series monitoring changes in reflectance over 152 days. In the presented numerical experiments, we focused on the influence of two stopping criteria based on a norm of projected gradient and a duality gap on model performance for the relaxed $\ell 2$-loss SVM. Attained results were compared and discussed with the $\ell 1$-loss SVM (relaxed). As an underlying solver for model training, we employed the MPRGP solver – a deterministic active-set method.

From the qualities of models in the sense of performance scores and training times, it seems than a terminating training process using stopping criteria based on duality gap for $\ell 2$-loss is a good trade-off between the $\ell 1$-loss and the $\ell 2$-loss models that a training process stopped exploiting a terminating condition incorporating a projected gradient. Such attained model performs better than $\ell 2$-loss case. However, the training process was almost 2 times slower. Compared to the $\ell 1$-loss model, it performs worse in the sense of a hinge-loss function value even so the training process is 1.45 times faster.

We compared the attained models employing relaxed formulations related to the SVM problems with models trained using the complete SVM formulations for both $\ell 1$-loss and $\ell 2$-loss functions as well. From the numerical experiments, we concluded that it is suitable to use relaxed versions of the SVM formulation for training models related to our classification problem because it takes a longer time to train models using complete SVM formulations than in the cases of their relaxed versions and attained models are slightly worse.

We studied qualities related to SVM models trained by means of PermonSVM with boosted tree methods implemented in XGBoost software. We observed that PermonSVM produces models with a higher sensitivity over precision (better at predicting positive events (wildfires) over determining pixel areas that are non-affected by fire). In contrast, the XGBoost model has a higher precision over sensitivity. We think predicting more false negatives (FN) over false positives (FP) would be more acceptable for natural hazard applications than the other way around.

5. Acknowledgements

This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic through the e-INFRA CZ (ID:90140) and by Grant of SGS No. SP2022/42, VŠB-Technical University of Ostrava, Czech Republic. The research has been also supported by European Union's Horizon 2020 research and innovation programme under grant agreement number 847593 and by The Czech Radioactive Waste Repository Authority (SÚRAO) under grant agreement number SO2020-017. Further, we acknowledge that the results of this research have been achieved using

the DECI resource Archer2 based in Great Britain at EPCC with support from the PRACE aisbl.

R. T. Mills was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration, and by Laboratory Directed Research and Development (LDRD) funding from Argonne National Laboratory, provided by the Director, Office of Science, of the U.S. Department of Energy under Contract No. DE-AC02-06CH11357.

References

[1] Balay, S. et al.: PETSc/TAO users manual. Tech. Rep. ANL-21/39 - Revision 3.18, Argonne National Laboratory, 2022.

[2] Chen, T. and Guestrin, C.: In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016.

[3] Cortes, C. and Vapnik, V.: Support-vector networks. Machine Learning (1995).

[4] Dostal, Z. and Schoberl, J.: Minimizing quadratic functions subject to bound constraints with the rate of convergence and finite termination. Computational Optimization and Applications **30** (2005).

[5] Dostál, Z.: *Optimal Quadratic Programming Algorithms, with Applications to Variational Inequalities*, vol. 23. SOIA, Springer, New York, US, 2009.

[6] Hapla, V. et al.: PermonQP, 2022. URL http://permon.vsb.cz/qp/.

[7] Hsieh, C. J., Chang, K. W., Lin, C. J., Keerthi, S. S., and Sundararajan, S.: A dual coordinate descent method for large-scale linear SVM. In: *Proceedings of the 25th international conference on Machine learning - ICML'08*. ACM Press, 2008.

[8] LeCun, Y., Bengio, Y., and Hinton, G.: Deep learning. Nature **521** (2015), 436.

[9] Lee, C. P. and Lin, C. J.: A study on l2-loss (squared hinge-loss) multiclass SVM. Neural Computation **25** (2013), 1302–1323.

[10] Pecha, M. and Horák, D.: Analyzing l1-loss and l2-loss support vector machines implemented in PERMON toolbox. In: *Lecture Notes in Electrical Engineering*, pp. 13–23. Springer International Publishing, 2020.

[11] PERMON: PermonSVM, 2022. URL http://github.com/permon/permonsvm.

Programs and Algorithms of Numerical Mathematics 21
J. Chleboun, P. Kůs, J. Papež, M. Rozložník, K. Segeth, J. Šístek (Eds.)
Institute of Mathematics CAS, Prague 2023

THE $\star$-PRODUCT APPROACH FOR LINEAR ODES: A NUMERICAL STUDY OF THE SCALAR CASE

Stefano Pozza, Niel Van Buggenhout

Charles University
Sokolovská 83 186, 75 Praha 8, Czech Republic
pozza@karlin.mff.cuni.cz, buggenhout@karlin.mff.cuni.cz

Abstract: Solving systems of non-autonomous ordinary differential equations (ODE) is a crucial and often challenging problem. Recently a new approach was introduced based on a generalization of the Volterra composition. In this work, we explain the main ideas at the core of this approach in the simpler setting of a scalar ODE. Understanding the scalar case is fundamental since the method can be straightforwardly extended to the more challenging problem of systems of ODEs. Numerical examples illustrate the method's efficacy and properties in the scalar case.

Keywords: ordinary differential equations, Volterra composition, Legendre polynomials

MSC: 34A25, 65L05, 94A11

1. Introduction

Systems of non-autonomous linear ordinary differential equations arise in a variety of contexts [1–3,10,11,13]. Yet, their solution remains surprisingly difficult to obtain, both formally and numerically, especially when dealing with systems of large-to-huge size. Consider an $N \times N$ matrix $\tilde{A}(t)$ depending on the variable $t \in I \subseteq \mathbb{R}$. The unique solution $U_s(t)$ of the system

$$\tilde{A}(t)U_s(t) = \frac{\mathrm{d}}{\mathrm{d}t}U_s(t), \quad U_s(s) = I_N, \qquad \text{for } t \geq s,\ t, s \in I, \tag{1}$$

with I_N the $N \times N$ identity matrix, is an $N \times N$ matrix-valued function known as the *time-ordered exponential* of $\tilde{A}(t)$. If $\tilde{A}(\tau_1)\tilde{A}(\tau_2) = \tilde{A}(\tau_2)\tilde{A}(\tau_1)$ for all $\tau_1, \tau_2 \in I$, then the time-ordered exponential can be expressed as

$$U_s(t) = \exp\left(\int_s^t \tilde{A}(\tau)\,\mathrm{d}\tau\right).$$

In general, however, $U_s(t)$ has no known simple expression in terms of $\tilde{A}(t)$.

DOI: 10.21136/panm.2022.17

In [5, 6], a new expression for the solution is given using the *path-sum approach*, a method able to express each element of $U_s(t)$ as a finite sequence of integral equations. However, this requires solving an NP-hard problem. In [7–9], the NP-hard problem is overcome by introducing the $\star$-*Lanczos method*, a constructive method able to tridiagonalize $\tilde{A}(t)$. At the heart of both the path-sum and $\star$-Lanczos method is a non-commutative convolution-like product, denoted by $\star$, defined between certain distributions [12]. Thanks to this product, the solution of (1) can be expressed through the $\star$-product inverse [7].

In this work, we aim to illustrate to the numerical mathematics community the $\star$-product and how it can be used to solve an ODE numerically. For this reason, we restrict the presentation to the simpler case in which the ODE (1) is a scalar equation. While this framework may look too simple to show the potential of the newly introduced technique, the reader should keep in mind that the results and construction we illustrate for the scalar case can be straightforwardly extended to the matrix case in full generality.

In Section 2, we give an introduction to the $\star$-product and the related expression for the solution of a scalar ODE. Section 3 discretizes the $\star$-product. As a consequence, the ODE solution can be obtained by solving a linear system. Several properties of the linear system are numerically investigated in Section 4. The numerical experiments in Section 5 show that the presented strategy can compute the solution up to machine precision. Section 6 concludes the presentation.

2. ODE solution by the $\star$-product approach

Given two appropriate bivariate functions $\tilde{f}_1(t,s)$, $\tilde{f}_2(t,s)$, the *Volterra composition*, introduced by Vito Volterra (e.g., [16]), is defined as

$$\big(\tilde{f}_2 \star_v \tilde{f}_1\big)(t,s) := \int_s^t \tilde{f}_2(t,\tau)\tilde{f}_1(\tau,s)\,\mathrm{d}\tau.$$

For our purposes, it suffices to assume $\tilde{f}_1$ and $\tilde{f}_2$ to be smooth (i.e., infinitely differentiable) on both variables over a bounded interval $I = [0,T]$ to have a well-defined operation for every $t,s \in I$. Therefore, from now on, a function marked with a tilde will stand for a smooth function in both t and s over I. Since the Volterra composition is closed for such functions, we are allowed to define the kth $\star_v$-power of a function $\tilde{f}$, that is, $\tilde{f}^{\star_v 1} = \tilde{f}$, and

$$\begin{aligned}\tilde{f}^{\star_v k} &:= \tilde{f} \star_v \tilde{f} \cdots \star_v \tilde{f} = \\ &= \int_s^t \tilde{f}(t,\tau_1) \int_s^{\tau_1} \tilde{f}(\tau_1,\tau_2) \cdots \int_s^{\tau_{k-2}} \tilde{f}(\tau_{k-2},\tau_{k-1})\tilde{f}(\tau_{k-1},s)\,\mathrm{d}\tau_{k-1}\cdots\,\mathrm{d}\tau_2\,\mathrm{d}\tau_1\end{aligned}$$

for $k > 1$ with the convention $\tau_0 = t$. Moreover, the operation is also defined for univariate functions $\tilde{f}_2(t)$:

$$\big(\tilde{f}_2(t) \star_v \tilde{f}_1(t,s)\big)(t,s) := \int_s^t \tilde{f}_2(t)\tilde{f}_1(\tau,s)\,\mathrm{d}\tau = \tilde{f}_2(t)\int_s^t \tilde{f}_1(\tau,s)\,\mathrm{d}\tau.$$

It is possible to use the Volterra composition to express the solution of the following differential equation for every initial time $s \in I$.

$$\frac{\mathrm{d}}{\mathrm{d}t} y_s(t) = \tilde{f}(t) y_s(t), \quad y_s(s) = 1, \ t \in [s, T] \subseteq \mathbb{R}; \tag{2}$$

see, e.g., [5]. In fact, using Picard iterations, we get

$$\begin{aligned}
\frac{d}{dt} y_s(t) &= \tilde{f}(t) y_s(t), \quad y_s(s) = 1 \\
&\downarrow \ \text{integration} \\
y_s(t) &= 1 + \int_s^t \tilde{f}(\tau) y_s(\tau) \mathrm{d}\tau \\
&\downarrow \ \text{integration} \\
y_s(t) &= 1 + \int_s^t \tilde{f}(\tau) \left(1 + \int_s^\tau \tilde{f}(\rho) y_s(\rho) \mathrm{d}\rho \right) \mathrm{d}\tau \\
&= 1 + \int_s^t \tilde{f}(\tau) + \int_s^\tau \tilde{f}(\tau) \tilde{f}(\rho) y_s(\rho) \, \mathrm{d}\rho \, \mathrm{d}\tau \\
&\downarrow \ \dots \\
y_s(t) &= 1 + \int_s^t \tilde{f}(\tau) \mathrm{d}\tau + \int_s^t \tilde{f}^{\star_v 2}(\tau) \mathrm{d}\tau + \dots,
\end{aligned}$$

from which we obtain the expression

$$y_s(t) = 1 + \int_s^t \sum_{k=1}^{\infty} \tilde{f}^{\star_v k}(\tau) \, \mathrm{d}\tau. \tag{3}$$

The Volterra composition is not a product and lacks essential features, for instance, the identity. For this reason, the Volterra composition has been extended, obtaining the so-called $\star$-product [7] that we briefly introduce in the following. Consider the class $\mathcal{D}(I)$ of all the distributions d that can be written as

$$d(t, s) = \widetilde{d}(t, s) \Theta(t - s) + \sum_{i=0}^{N} \widetilde{d}_i(t, s) \delta^{(i)}(t - s),$$

where N is a finite integer, $\tilde{d}, \tilde{d}_i$ are smooth bivariate functions over $I \times I$, $\Theta(\cdot)$ stands for the Heaviside theta function

$$\Theta(t - s) = \begin{cases} 1, & t \geq s, \\ 0, & t < s \end{cases},$$

and $\delta^{(i)}(\cdot)$ is the ith derivative of the Dirac delta distribution $\delta(\cdot) = \delta^{(0)}(\cdot)$. We can endow the class $\mathcal{D}(I)$ with a non-commutative algebraic structure by defining the $\star$-product as

$$\big(f_2 \star f_1\big)(t, s) := \int_I f_2(t, \tau) f_1(\tau, s) \, \mathrm{d}\tau, \quad f_1, f_2 \in \mathcal{D}(I). \tag{4}$$

The $\star$-product is associative over $\mathcal{D}(I)$, $\mathcal{D}(I)$ is closed under $\star$-multiplication, and the identity element with respect to the $\star$-product is the Dirac delta distribution, $1_\star := \delta(t-s)$, see, e.g., [7].

Consider the subclass $\mathcal{C}^\infty_\Theta(I) \subset \mathcal{D}(I)$ comprising those distributions of the form

$$f(t,s) = \widetilde{f}(t,s)\Theta(t-s).$$

Then, the $\star$-product between $f_1, f_2 \in \mathcal{C}^\infty_\Theta(I)$ reduces to the Volterra composition

$$\begin{aligned}\big(f_2 \star f_1\big)(t,s) &= \int_I \widetilde{f}_2(t,\tau)\widetilde{f}_1(\tau,s)\Theta(t-\tau)\Theta(\tau-s)\,\mathrm{d}\tau, \\ &= \Theta(t-s)\int_s^t \widetilde{f}_2(t,\tau)\widetilde{f}_1(\tau,s)\,\mathrm{d}\tau = \Theta(t-s)(\tilde{f}_2 \star_v \tilde{f}_1)(t,s).\end{aligned}$$

As a consequence, using (3), we can express the solution of (2) for every $s \in I$ as

$$y_s(t) = u(t,s) = \Theta(t-s) \star R_\star(f), \tag{5}$$

where $f(t,s) = \tilde{f}(t)\Theta(t-s)$ and $R_\star(f)$ is the $\star$-resolvent of f, i.e.,

$$R_\star(f) = \delta(t-s) + \sum_{k=1}^{\infty} f(t,s)^{\star k},$$

with $f(t,s)^{\star k} = \Theta(t-s)\tilde{f}(t)^{\star_v k}$. Note that the series $\sum_{k=1}^\infty f(t,s)^{\star k}$ converges for every $f \in \mathcal{C}^\infty_\Theta(I)$. The $\star$-product easily extends to matrices composed of elements from $\mathcal{D}(I)$ by extending the scalar multiplication appearing in the integrand in (4) to the usual matrix-matrix multiplication; see [9] for more details.

While expression (5) is compact, the $\star$-resolvent definition hides an infinite series of nested integrals. Therefore, at first sight, it does not seem like a convenient expression. In the next section, we effectively solve this problem by showing that it is possible to approximate the $\star$-product by the usual matrix-matrix product between (time-independent) matrices. Consequently, for a fixed s, expression (5) can be approximated relatively cheaply by solving a linear system.

3. Discretization of the $\star$-product

In this section, we describe an effective strategy for approximating the $\star$-product. Consider a sequence of orthonormal functions $\{p_k\}_k$ over the bounded interval $I = [0,T]$, i.e.,

$$\int_I p_k(\tau)p_\ell(\tau)d\tau = \begin{cases} 0, & \text{if } k \neq \ell, \\ 1, & \text{if } k = \ell, \end{cases}$$

so that $\{p_k\}_k$ is a basis for the space of smooth functions over I. Note that the functions p_k are not in $\mathcal{D}(I)$; hence we cannot (formally) $\star$-multiply them. Consider

a distribution $f \in \mathcal{C}^{\infty}_{\Theta}(I)$. The function $f(t,s) = \tilde{f}(t,s)\Theta(t-s)$ is piecewise smooth, therefore, we can choose the basis $\{p_k\}_k$ so that

$$f(t,s) = \sum_{k=0}^{\infty}\sum_{\ell=0}^{\infty} f_{k,\ell}\, p_k(t)p_\ell(s), \quad t \neq s,\ t,s \in I, \tag{6}$$

with coefficients

$$f_{k,\ell} = \int_I \int_I f(\tau,\rho)p_k(\tau)p_\ell(\rho)\, \mathrm{d}\rho\, \mathrm{d}\tau.$$

For instance, the basis $\{p_k\}_k$ can be set as the sequence of shifted Legendre polynomials (e.g., [14, p. 55]). Defining the *coefficient matrix* F_M and the vector $\phi_M(t)$ as

$$F_M := \begin{bmatrix} f_{0,0} & f_{0,1} & \cdots & f_{0,M-1} \\ f_{1,0} & f_{1,1} & \cdots & f_{1,M-1} \\ \vdots & \vdots & & \vdots \\ f_{M-1,0} & f_{M-1,1} & \cdots & f_{M-1,M-1} \end{bmatrix}, \quad \phi_M(t) := \begin{bmatrix} p_0(s) \\ p_1(s) \\ \vdots \\ p_{M-1}(s) \end{bmatrix}, \tag{7}$$

the truncated expansion series can be written in the matrix form:

$$f_M(t,s) := \sum_{k=0}^{M-1}\sum_{\ell=0}^{M-1} f_{k,\ell}\, p_k(t)p_\ell(s) = \phi_M(t)^T F_M\, \phi_M(s).$$

Consider $f, g, h \in \mathcal{C}^{\infty}_{\Theta}(I)$ so that $h = f \star g$, and the related coefficient matrices (7), respectively, F_M, G_M, H_M. By replacing f and g with their expansion (6), it is not difficult to show that the expansion coefficients for h are given by

$$h_{k,\ell} = \sum_{j=0}^{\infty} f_{k,j}\, g_{j,\ell}, \tag{8}$$

assuming the latter series converges (such an assumption is grounded on the numerical experiments of the next section). As a consequence, we can approximate H_M by the expression

$$H_M \approx \hat{H}_M := F_M G_M, \tag{9}$$

i.e., the $\star$-product can be approximated by the usual matrix-matrix multiplication of the related coefficient matrices.

The approximation (9) is affected by a truncation error. Therefore, fixing k and ℓ, if the magnitude of the product $f_{k,j} \cdot g_{j,\ell}$ in (8) does not decay quickly enough for $j \to \infty$, then the truncation error $(H_M)_{k,\ell} - (\hat{H}_M)_{k,\ell}$ can be too large for practical purposes. Luckily, since $f \in \mathcal{C}^{\infty}_{\Theta}$, numerical considerations illustrate that F_M and G_M are numerically banded for a certain choice of $\{p_k\}_k$; for instance, see Section 4 where we choose the shifted Legendre polynomials. Therefore, M does not need to be too large to reach a small truncation error in the approximation (9), excluding the last rows of the matrix $\hat{H}_M$ where the truncation error can still be significant. Further

details and explanations on this matter are being developed and will be presented in future work. For the moment, in Section 4, we provide numerical evidence of these claims.

To conclude the presentation, we must discuss the convergence behavior of expansion (6). Indeed, since f is discontinuous for $t = s$, the expansion may not converge quickly (or may not converge) to $f(t,s)$ for every $t, s \in I$; see, e.g., [14,15] for the polynomial case. Nevertheless, fixing $s = 0$, the univariate function $f(t,0) = \tilde{f}(t,0)\Theta(t-0) = \tilde{f}(t,0)$ is smooth over $I = [0,T]$. Therefore

$$f(t,0) = \sum_{k=0}^{\infty} a_k p_k(t) = \sum_{k=0}^{\infty} p_k(t) \sum_{\ell=0}^{\infty} f_{k,\ell}\, p_\ell(0),$$

with $a_k = \sum_{\ell=0}^{\infty}(f_{k,\ell}\, p_\ell(0))$. As a consequence, we can approximate the function $f(t,0)$ by the expression

$$f(t,0) \approx \phi_M(t)^T F_M\, \phi_M(0),$$

and expect to reach a small enough accuracy for a (relatively) small M. Section 5 illustrates with several numerical examples that it is possible to achieve machine precision accuracy for a small value of M.

Consider the function $u(t,s)$ in (5). Using the previous construction, the related coefficient matrix U_M, i.e., such that $u(t,s) \approx \phi(t)_M^T\, U_M\, \phi_M(s)$, can be approximated by

$$U_M \approx T_M(I_M - F_M)^{-1},$$

where T_M is the coefficient matrix of $\Theta(t-s)$, and F_M is the coefficient matrix of $\tilde{f}(t)\Theta(t-s)$, with $\tilde{f}(t)$ from (2). Since $u(t,s) \in \mathcal{C}^{\infty}_{\Theta}$, for $s = 0$ we can approximate the solution of (2) by the formula:

$$y_0(t) \approx \phi_M(t)^T U_M\, \phi_M(0) \approx \phi_M(t)^T T_M(I_M - F_M)^{-1}\phi_M(0).$$

Then, the vector $u_M = T_M x$ contains the approximated expansion coefficients of $y_0(t)$, i.e.,

$$y_0(t) \approx \phi_M(t)^T u_M, \quad \text{for every } t \in I,$$

where x is the solution of the linear system

$$(I_M - F_M)x = \phi_M(0). \tag{10}$$

4. Properties of the coefficient matrix

In this section, we illustrate several properties of the coefficient matrices (7) through numerical examples. We set $I = [0,1]$, and, as the sequence of orthonormal functions, we choose the sequence of orthonormal shifted Legendre polynomials, i.e., the sequence of polynomials $\{p_k\}_k$ such that

$$\int_0^1 p_k(\tau)p_\ell(\tau)\,\mathrm{d}\tau = \begin{cases} 1, & \text{if } k = \ell, \\ 0, & \text{if } k \neq \ell, \end{cases}$$

Functions	$\tilde{f}_1 = 1$	$\tilde{f}_2 = t$	$\tilde{f}_3 = t^3$	$\tilde{f}_4 = \cos(t)$	$\tilde{f}_5 = \log(t+1)$
			$M = 25$		
Num. band.	1	2	4	13	20
Spectral radius	0.0592	0.0357	0.0238	0.0480	0.0271
$\sigma_{\min}$	$2.42e-3$	$3.50e-5$	$9.68e-9$	$1.74e-3$	$3.42e-5$
$\sigma_{\max}$	1.2732	0.9447	0.6864	0.9694	0.6938
			$M = 100$		
Num. band.	1	2	4	13	20
Spectral radius	0.0556	0.0296	0.0155	0.0444	0.0223
$\sigma_{\min}$	$1.56e-4$	$1.57e-7$	$2.33e-13$	$1.10e-4$	$1.56e-7$
$\sigma_{\max}$	1.2732	0.9447	0.6864	0.9694	0.6938
			$M = 500$		
Num. band.	1	2	4	13	20
Spectral radius	0.0554	0.0297	0.0146	0.0458	0.0215
$\sigma_{\min}$	$6.27e-6$	$2.59e-10$	$6.58e-19$	$2.59e-10$	$3.42e-5$
$\sigma_{\max}$	1.2732	0.9447	0.6864	0.9694	0.6938

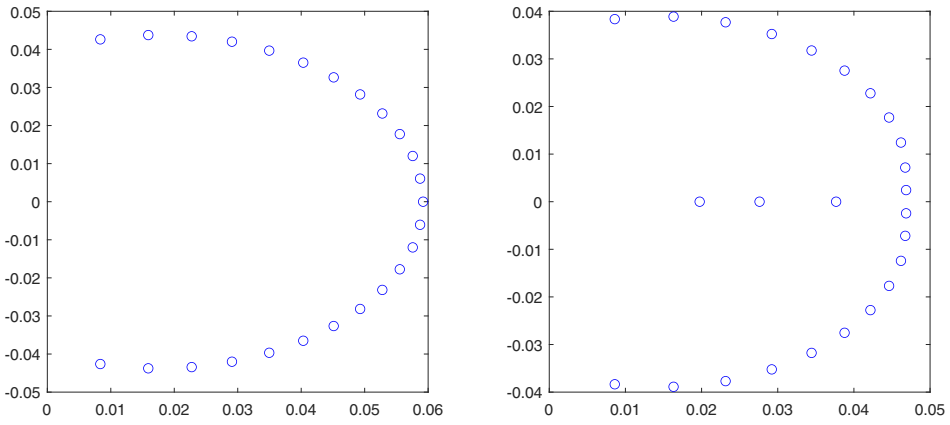
Table 1: Properties of coefficient matrices $F_M^{(k)}$.

with k, ℓ the degree of the polynomial. In the following, we consider the functions $f_k(t,s) = \tilde{f}_k(t)\Theta(t-s)$ from Table 1 and the related $M \times M$ coefficient matrix $F_M^{(k)}$ defined in (7). The numerical experiments were performed using MatLab R2022a.

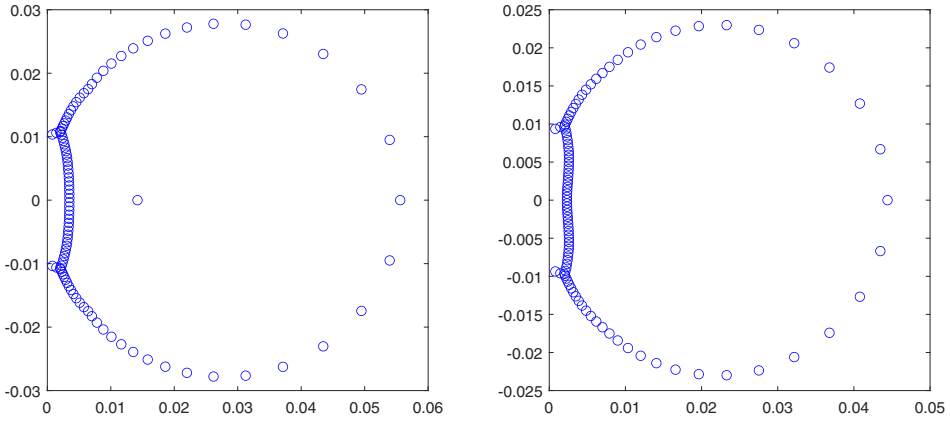
Table 1 reports the numerical bandwidth of each coefficient matrix $F_M^{(k)}$ for $M = 25, 100, 500$. With numerical bandwidth, we mean the bandwidth of the matrix once all its elements with a magnitude smaller than the machine precision have been rounded to zero. First, we observe that the numerical bandwidth is the same for every value of M. Moreover, we note that for the polynomials $\tilde{f}_1, \tilde{f}_2, \tilde{f}_3$, the corresponding bandwidth is equal to the degree of the polynomial plus one. Finally, the functions $f_4(t,s) = \cos(t)\Theta(t-s)$, $f_5(t,s) = \log(t+1)\Theta(t-s)$ are also numerically banded.

Table 1 also reports the spectral radius and the minimal and maximal singular values (respectively $\sigma_{\min}$, $\sigma_{\max}$) of each $F_M^{(k)}$. While both the spectral radius and $\sigma_{\max}$ do not vary significantly for $M = 25, 100, 500$, $\sigma_{\min}$ becomes smaller as M increases. As the linear system (10) involves the shifted matrix $I_M - F_M^{(k)}$, it is important to note that all the computed spectral radii are smaller than 1.

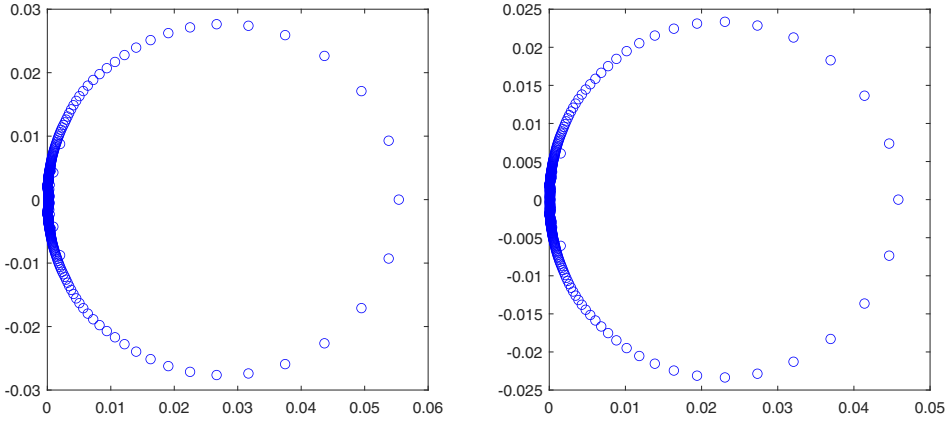
Finally, Figure 1 presents the spectra of the matrices $F_M^{(1)}$ and $F_M^{(4)}$ for $M = 25, 100, 500$. For both the functions, as M increases, the spectrum tends to distribute in a circle on the right-half of the complex plane, closer and closer to the origin. We do not report the spectrum plots of the other matrices considered above since they display analogous behavior.



$M = 25$



$M = 100$



$M = 500$

Figure 1: Spectrum of the coefficient matrices $F_M^{(1)}$ (left), $F_M^{(4)}$ (right) defined in (7), for $M = 25, 100, 500$.

5. Numerical experiments

In this section, we test the numerical method explained in Section 3 on the ODE

$$\frac{\mathrm{d}}{\mathrm{d}t}y(t) = \tilde{f}_k(t)y(t), \quad y(0) = 1,\ t \in I = [0,1], \tag{11}$$

for each function $\tilde{f}_k$ from Table 1. More precisely, the method works as follows:

1. We discretize $f_k(t,s) = \tilde{f}_k(t)\Theta(t-s)$ as described in Section 3, obtaining the matrix $F_M^{(k)}$. We use as an orthonormal basis the shifted orthonormal Legendre polynomials from Section 4.

2. Let b be the numerical bandwidth of $F_M^{(k)}$; we define the matrix $\hat{F}_M^{(k)}$ by setting the last b rows of the matrix $F_M^{(k)}$ to zero. This has proven helpful in reducing the accumulation of truncation errors in the last rows of the solution of (10).

3. We solve the (banded) linear system

$$\left(I_M - \hat{F}_M^{(k)}\right) x = \phi_M(0),$$

   using the Matlab backslash \ operation.

4. The solution of (11) is approximated by

$$y(t) \approx \hat{y}_M(t) := \phi_M(t)^T u_M \quad \text{with } u_M = T_m\, x. \tag{12}$$

In Table 2, we report the maximal relative error of approximation (12) over I for $M = 25, 100$. The relative errors were computed on an equispaced mesh of 100 points over $[0,1]$. As a reference value for the solution, we considered the function $\exp(\int_0^t \tilde{f}_k(\tau)\,\mathrm{d}\tau)$. We compare our results with the maximal relative errors obtained using the Matlab methods `ode45` and `ode89` with relative and absolute tolerances set equal to `eps` $= 2.2204e-16$. For $M = 100$, Table 2 shows that approximation (12) is always better than the others. On the other hand, for $M = 25$, we obtain worse results for $k = 3, 4, 5$, showing that it is possible to calibrate the accuracy of the solution by the matrix size.

With these experiments, we do not want to claim anything about the performance of our method compared to well-established explicit methods such as `ode45` and `ode89`. The examples considered here are certainly not enough for drawing any conclusion. The table aims to show that approximation (12) can compete in accuracy with well-established approaches, a promising result for our future work.

6. Conclusion and future work

In this work, we have explained how to express the solution of a scalar linear ODE using the so-called $\star$-product. Moreover, we have shown how to derive a numerical method from this expression and successfully tested it on several examples.

Functions	$\tilde{f}_1 = 1$	$\tilde{f}_2 = t$	$\tilde{f}_3 = t^3$	$\tilde{f}_4 = \cos(t)$	$\tilde{f}_5 = \log(t+1)$
$\hat{y}_{25}(t)$	$1.20e-15$	$1.11e-15$	$3.36e-14$	$1.37e-09$	$4.04e-04$
$\hat{y}_{100}(t)$	$1.20e-15$	$1.11e-15$	$8.88e-16$	$1.22e-15$	$9.77e-16$
`ode45`	$9.76e-15$	$4.61e-13$	$1.53e-12$	$1.13e-13$	$9.72e-13$
`ode89`	$1.17e-13$	$6.69e-14$	$3.63e-14$	$1.13e-13$	$9.92e-14$

Table 2: Maximal relative error over $I = [0, 1]$ of `ode45`, `ode89` methods and of the approximation $\hat{y}_M(t)$ in (12) for the solution of the ODE (11).

The numerical method requires solving a linear system whose properties have also been numerically investigated. Concerning the numerical efficiency of the introduced method, other possible approaches in the solution of the linear system may be used – for instance, Krylov subspace methods. Furthermore, since the solution depends continuously on the initial time parameter s, we are also investigating the use of acceleration methods such as the one in [4]. In addition, we are currently developing an efficient method for computing the coefficient matrix F_M.

Given a smooth matrix-valued function $\tilde{A}(t) \in \mathbb{C}^{N\times N}$, the solution of the system

$$\frac{\mathrm{d}}{\mathrm{d}t}Y_s(t) = \tilde{A}(t)Y_s(t), \quad Y(s) = I_N,\ t \geq s,\ t, s \in I,$$

can also be expressed as

$$Y_s(t) = U(t, s) = \Theta(t - s) \star R_\star(\tilde{A}(t)\Theta(t - s)), \quad t, s \in I,$$

following the results in [5]. Therefore, the scalar method we have described can be generalized to the more challenging problem offered by systems of non-autonomous linear ODEs. The results discussed in this work are thus promising for developing new efficient methods for computing $Y_s(t)$.

Acknowledgements

This work was supported by Charles University Research programs UNCE/SCI/023 and PRIMUS/21/SCI/009 and by the Magica project ANR-20-CE29-0007 funded by the French National Research Agency.

References

[1] Autler, S.H. and Townes, C.H.: Stark effect in rapidly varying fields. Phys. Rev. **100** (1955), 703–722. URL https://doi.org/10.1103/PhysRev.100.703.

[2] Benner, P., Cohen, A., Ohlberger, M., and Willcox, K.: *Model reduction and approximation: Theory and algorithms*. Computational Science and Engineering, SIAM, Philadelphia, 2017.

[3] Blanes, S.: High order structure preserving explicit methods for solving linear-quadratic optimal control problems. Numer. Algorithms **69** (2015), 271–290. URL https://doi.org/10.1007/s11075-014-9894-0.

[4] Buoso, D., Karapiperi, A., and Pozza, S.: Generalizations of Aitken's process for a certain class of sequences. Appl. Numer. Math. **90** (2015), 38–54. URL https://doi.org/10.1016/j.apnum.2014.12.002.

[5] Giscard, P.L., Lui, K., Thwaite, S.J., and Jaksch, D.: An exact formulation of the time-ordered exponential using path-sums. J. Math. Phys. **56** (2015), 053 503. URL https://doi.org/10.1063/1.4920925.

[6] Giscard, P.L. and Bonhomme, C.: Dynamics of quantum systems driven by time-varying Hamiltonians: Solution for the Bloch-Siegert Hamiltonian and applications to NMR. Phys. Rev. Research **2** (2020), 023 081. URL https://link.aps.org/doi/10.1103/PhysRevResearch.2.023081.

[7] Giscard, P.L. and Pozza, S.: Lanczos-like algorithm for the time-ordered exponential: the $*$-inverse problem. Appl. Math. **65** (2020), 807–827. URL https://doi.org/10.21136/AM.2020.0342-19.

[8] Giscard, P.L. and Pozza, S.: Tridiagonalization of systems of coupled linear differential equations with variable coefficients by a Lanczos-like method. Linear Algebra Appl. **624** (2021), 153–173. URL https://doi.org/10.1016/j.laa.2021.04.011.

[9] Giscard, P.L. and Pozza, S.: A lanczos-like method for non-autonomous linear ordinary differential equations. Bol. Unione Mat. Ital. (2022). URL https://doi.org/10.1007/s40574-022-00328-6.

[10] Kwakernaak, H. and Sivan, R.: *Linear optimal control systems*, vol. 1. Wiley-interscience, New York, 1972.

[11] Lauder, M., Knight, P., and Greenland, P.: Pulse-shape effects in intense-field laser excitation of atoms. Opt. Acta **33** (1986), 1231–1252. URL https://doi.org/10.1080/713821874.

[12] Schwartz, L.: *Théorie des distributions*. Hermann, Paris, 1978.

[13] Shirley, J.H.: Solution of the Schrödinger equation with a Hamiltonian periodic in time. Phys. Rev. **138** (1965), B979–B987. URL https://doi.org/10.1103/PhysRev.138.B979.

[14] Silverman, R.A. et al.: *Special functions and their applications*. Courier Corporation, 1972.

[15] Trefethen, L.N.: *Approximation theory and approximation practice*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2013.

[16] Volterra, V. and Pérès, J.: *Leçons sur la composition et les fonctions permutables*. Éditions Jacques Gabay, Paris, 1928.

Programs and Algorithms of Numerical Mathematics 21
J. Chleboun, P. Kůs, J. Papež, M. Rozložník, K. Segeth, J. Šístek (Eds.)
Institute of Mathematics CAS, Prague 2023

IDENTIFICATION PROBLEM FOR NONLINEAR BEAM — EXTENSION FOR DIFFERENT TYPES OF BOUNDARY CONDITIONS

Jana Radová, Jitka Machalová

Department of Mathematical Analysis and Applications of Mathematics
Faculty of Science, Palacký University Olomouc
17. listopadu 12, 779 00, Olomouc, Czech Republic
jana.radova@upol.cz, jitka.machalova@upol.cz

Abstract: Identification problem is a framework of mathematical problems dealing with the search for optimal values of the unknown coefficients of the considered model. Using experimentally measured data, the aim of this work is to determine the coefficients of the given differential equation. This paper deals with the extension of the continuous dependence results for the Gao beam identification problem with different types of boundary conditions by using appropriate analytical inequalities with a special attention given to the Wirtinger's inequality and its modification. On the basis of these results for the different types of the boundary conditions the existence theorem for the identification problem can be proven.

Keywords: identification problem, nonlinear Gao beam, Wirtinger's inequality, Wirtinger-Poincaré-Almansi inequality

MSC: 26D20, 49J15, 65L09, 74K10

1. Introduction

Beams are commonly used in engineering constructions and there are many practical applications for parameter identification problem. This paper deals with a nonlinear Gao beam model. A problem of identifying coefficients in the Gao beam is presented in a recent paper [11], where the aim is to find unknown material parameters for this beam by using an optimal control approach. The existence of at least one solution of the optimal control problem is proven by using continuous dependence of the solution on the material parameters. But the results are proven only for one type of physically relevant boundary conditions. In this paper, in Section 3, we prove the continuous dependence for other types of boundary conditions. The proof is based on analytical inequalities presented in Section 2.

DOI: 10.21136/panm.2022.18

First, let us start with the nonlinear Gao beam model, which was firstly introduced in [5]. With respect to a small correction of this model which was proposed in [9], the nonlinear Gao beam is given by the fourth order equation:

$$E\,I\,w^{IV} - E\,\alpha\,(w')^2\,w'' + P\,(1-\nu^2)\,w'' = f \qquad \text{in } (0,L), \tag{1}$$

where

$$I = \frac{2}{3}t^3 b, \qquad \alpha = 3\,t\,b\,(1-\nu^2), \qquad f = (1-\nu^2)\,q.$$

Here, E denotes Young's elastic modulus of the material, I is the constant area moment of inertia, w is the deflection of the beam, $2t$ and b represents the thickness and width of the beam, respectively. The Poisson's ratio is represented by the symbol ν, q is the applied transverse load and P stands for the constant axial force acting at the end point of the beam $x = L$. We distinguish two types of acting axial force: $P > 0$ and $P < 0$ causing a compression and a tension, respectively. The beam model needs to be completed by one of the following boundary conditions:

(B1) simply supported beam: $w(0) = w(L) = w''(0) = w''(L) = 0$;

(B2) clamped beam: $w(0) = w'(0) = w(L) = w'(L) = 0$;

(B3) propped cantilever beam: $w(0) = w'(0) = w(L) = w''(L) = 0$;

(B4) cantilever beam: $w(0) = w'(0) = 0$,

$$w''(L) = E\,I\,w'''(L) - \tfrac{1}{3}E\,\alpha\,(w'(L))^3 + P\,(1-\nu^2)\,w'(L) = 0.$$

The spaces of admissible displacements are denoted as V_i, $i = 1,\dots,4$, and defined by the corresponding stable boundary conditions contained in (B1),...,(B4):

$$\begin{aligned}
V_1 &= \{v \in H^2((0,L))\colon v(0) = v(L) = 0\},\\
V_2 &= \{v \in H^2((0,L))\colon v(0) = v'(0) = v(L) = v'(L) = 0\},\\
V_3 &= \{v \in H^2((0,L))\colon v(0) = v'(0) = v(L) = 0\},\\
V_4 &= \{v \in H^2((0,L))\colon v(0) = v'(0) = 0\},
\end{aligned}$$

where $H^2((0,L))$ is the Sobolev space which consists of those square integrable functions for which all generalized partial derivatives up to the order two are also square integrable on the interval $(0,L)$. In the following, V will be one of above $V_1,\dots,V_4$.

The variational formulation of the problem (1) reads as follows:

$$\begin{cases} \text{Find } w \in V \text{ such that} \\ a(w,v) + \pi(w,v) = \mathcal{L}(v), \qquad\qquad \forall v \in V, \end{cases} \tag{2}$$

where

$$\begin{aligned}
a(w,v) &= \int_0^L EIw''v''\mathrm{d}x - \int_0^L P(1-\nu^2)w'v'\mathrm{d}x,\\
\pi(w,v) &= \int_0^L E\,t\,b\,(1-\nu^2)(w')^3v'\mathrm{d}x, \qquad \mathcal{L}(v) = \int_0^L (1-\nu^2)\,q\,v\,\mathrm{d}x.
\end{aligned}$$

The following theorem provides the essential assumptions to the existence of a unique solution to problem (2), see [8].

Theorem 1. *Let E, t, b be positive constants, $\nu \in (0, 0.5\rangle$, $q \in L^2(0, L)$ and $P < \overline{P}$, where $\overline{P} = \dfrac{1}{1-\nu^2} P_{cr}^E$. Then the problem (2) has a unique solution.*

Remark 1. *Since it is not possible to find analytical expression for the critical force for the Gao beam, we use a lower bound $\overline{P}$ which can be expressed by Euler's critical load P_{cr}^E as*

$$\overline{P} = \frac{1}{1-\nu^2} P_{cr}^E = \frac{1}{1-\nu^2} \frac{\pi^2 EI}{(\mathcal{K} \cdot L)^2}, \tag{3}$$

see [4], [11]. *The constant $\mathcal{K}$ depends on the boundary conditions as follows:*

	(B1)	(B2)	(B3)	(B4)
$\mathcal{K}$	1	0.5	0.7	2

The existence and uniqueness of a solution to (2) can be established under stronger assumptions on physical data. In section 3 we will consider the piecewise constant material parameters E and ν. In this case, the assumptions of Theorem 1 have to be modified as follows: let E, ν are positive, piecewise constant functions over a finite, fixed partition of $\langle 0, L\rangle$, b, t are positive constants in $\langle 0, L\rangle$ and

$$P < \overline{P}_{\min}, \quad \overline{P}_{\min} = \frac{\pi^2 \, \overline{E} \, I}{(1-\overline{\nu}^2)(\mathcal{K} \cdot L)^2},$$

where $\overline{E}$, $\overline{\nu}$ are the minimal values of E, and ν, respectively. The proof of Theorem 1 generalized for piecewise constant material parameters can be done in a similar way as in [8].

2. Analytical inequalities

In this section we introduce several analytical inequalities that will be used in the next section for extension of results for the identification problem. Let us start with *Wirtinger's inequality* in its original version, see [10].

Theorem 2. *Let $y(x) \in L^2(\mathbb{R})$ be a periodic function with period 2π and let $y'(x) \in L^2(\mathbb{R})$. If $\int_0^{2\pi} y(x)\,\mathrm{d}x = 0$, then the following inequality holds:*

$$\int_0^{2\pi} (y(x))^2 \,\mathrm{d}x \le \int_0^{2\pi} (y'(x))^2 \,\mathrm{d}x. \tag{4}$$

The proof of the inequality is based on the Fourier expansions of f and f', see [2]. To better suit our needs, let the Theorem 2 be interpreted for $f \in H^1((0, 2\pi))$: if $\int_0^{2\pi} f(x)\,\mathrm{d}x = 0$, then (4) holds.

The inequality (4) from Theorem 2 can be generalized for a function defined on the interval $\langle 0, L\rangle$. Let us assume that $y(x)$ is a periodic function with period L and let $y'(x) \in L^2(0, L)$. Substituting $t = \frac{L}{2\pi}x$ we obtain a modification of (4):

$$\int_0^L (\widehat{y}(t))^2 \,\mathrm{d}t \;\leq\; \left(\frac{L}{2\pi}\right)^2 \int_0^L (\widehat{y}'(t))^2 \,\mathrm{d}t, \tag{5}$$

where $\widehat{y}(t) = y(x(t)) = y(\frac{2\pi t}{L})$. If we consider the nonlinear Gao beam with boundary conditions (B2) and set $y(x) = w'(x)$, the assumptions of (5) are satisfied and we get:

$$\int_0^L (w'(x))^2 \,\mathrm{d}x \;\leq\; \left(\frac{L}{2\pi}\right)^2 \int_0^L (w''(x))^2 \,\mathrm{d}x. \tag{6}$$

The following inequality is known as the *Wirtinger-Poincaré-Almansi inequality*, see [7].

Theorem 3. *Let $y(x)$ be a function defined on the interval $\langle 0, \pi\rangle$ such that $y(0) = y(\pi) = 0$ and $y'(x) \in L^2(0, \pi)$. Then*

$$\int_0^\pi (y(x))^2 \,\mathrm{d}x \;\leq\; \int_0^\pi (y'(x))^2 \,\mathrm{d}x. \tag{7}$$

The inequality (7) can be generalized for a function y on $\langle 0, L\rangle$. If we have $y(0) = y(L) = 0$ and $y'(x) \in L^2(0, L)$ than

$$\int_0^L (y(x))^2 \,\mathrm{d}x \;\leq\; \left(\frac{L}{\pi}\right)^2 \int_0^L (y'(x))^2 \,\mathrm{d}x. \tag{8}$$

Similar inequality can be defined on $\langle 0, L\rangle$ for functions satisfying only a single condition $y(0) = 0$, for details see [6].

The key idea is to symmetrize the problem by defining the function $y(x)$ on interval $\langle 0, 2L\rangle$, i.e. for any $x \in \langle L, 2L\rangle$ we define $y(x) = y(L + \xi) = y(L - \xi) = y(2L - x)$, where $\xi = x - L$. Thus, from $y(0) = y(2L)$, $y \in L^2(0, 2L)$ and (8) we have:

$$\int_0^{2L} (y(x))^2 \,\mathrm{d}x \;\leq\; \left(\frac{2L}{\pi}\right)^2 \int_0^{2L} (y'(x))^2 \,\mathrm{d}x\,.$$

Due to the symmetry on $\langle 0, 2L\rangle$ we get:

$$\int_0^L (y(x))^2 \,\mathrm{d}x \;\leq\; \left(\frac{2L}{\pi}\right)^2 \int_0^L (y'(x))^2 \,\mathrm{d}x. \tag{9}$$

This idea could be used for the cantilever beam, i.e. the nonlinear beam with the boundary conditions (B4). We can symmetrize the deflection w on the interval $\langle 0, 2L\rangle$, by setting $y(x) = w'(x)$ and using (9) we get:

$$\int_0^L (w'(x))^2 \,\mathrm{d}x \;\leq\; \left(\frac{2L}{\pi}\right)^2 \int_0^L (w''(x))^2 \,\mathrm{d}x. \tag{10}$$

If we consider the nonlinear Gao beam with the boundary conditions (B1) we can use the same idea as for boundary conditions (B2). The function w' satisfies the assumptions of Theorem 2 modified on interval $\langle 0, L\rangle$, so with respect to (5) we get

$$\int_0^L (w'(x))^2 \,\mathrm{d}x \;\le\; \left(\frac{L}{2\pi}\right)^2 \int_0^L (w''(x))^2 \,\mathrm{d}x. \tag{11}$$

It is obvious that

$$\int_0^L (w'(x))^2 \,\mathrm{d}x \;\le\; \left(\frac{L}{2\pi}\right)^2 \int_0^L (w''(x))^2 \,\mathrm{d}x \;\le\; \left(\frac{L}{\pi}\right)^2 \int_0^L (w''(x))^2 \,\mathrm{d}x. \tag{12}$$

Using Theorem 3, its generalization (8) and (6) we get

$$\int_0^L (w(x))^2 \,\mathrm{d}x \;\le\; \left(\frac{L}{\pi}\right)^2 \int_0^L (w'(x))^2 \,\mathrm{d}x \;\le\; \frac{1}{4}\left(\frac{L}{\pi}\right)^4 \int_0^L (w''(x))^2 \,\mathrm{d}x. \tag{13}$$

Finally, for the propped cantilever beam, i.e. for the nonlinear beam with the boundary conditions (B3), we can use the same idea as for cantilever beam which leads to the inequality (10).

3. Identification problem - extension for other types of boundary conditions

In this section we extend the results presented in [11], where the identification of the material parameters given by the Young modulus E and Poisson ratio ν in the Gao beam equation (1) is studied by using an optimal control approach. We suppose that the beam is piecewise homogeneous, i.e. the parameters E, ν are piecewise constant. For this reason let the interval $(0, L)$ be decomposed into mutually disjoint open intervals K_i, called material elements, $i = 1, \cdots, r$, i.e. $K_i \cap K_j = \emptyset$, $\forall i \neq j$ and $\langle 0, L\rangle = \bigcup_{i=1}^{r} \bar{K}_i$. The material parameters are chosen from an admissible set U_{ad} :

$$\begin{aligned} U_{ad} \;=\; \{(E,\nu) \in (L^\infty(0,L))^2 : 0 < E_{\min} \le E \le E_{\max} < \infty \text{ in } (0,L),\\ 0 < \nu \le 0.5 \text{ in } (0,L), (E,\nu)|_{K_i} \in (P_0(K_i))^2, i = 1,\dots,r\}, \end{aligned} \tag{14}$$

where $E_{\min}$, $E_{\max}$ are given constants and $P_0(K_i)$ is the set of constant functions on K_i. Therefore, the admissible set U_{ad} is the closed, convex subset of couples of piecewise constant functions on the partition of $(0, L)$.

The variational formulation of the state problem with respect to the corresponding boundary conditions (B1)–(B4), see [11], reads as follows:

$$\begin{cases} \text{For given } \ (E,\nu) \in U_{ad} & \\ \text{find } w := w(E,\nu) \in V \text{ such that} & (\mathcal{P}(E,\nu)) \\ a(w,v) + \pi(w,v) = \mathcal{L}(v), \qquad\quad \forall v \in V, & \end{cases}$$

where the forms a, π and $\mathcal{L}$ have the same meaning as above. According to Remark 1, to have a unique solution to the problem $(\mathcal{P}(E,\nu))$ for any $(E,\nu) \in U_{ad}$, let t and b be positive constants, $q \in L^2(0,L)$ and

$$P < \widehat{P}_{\min}, \text{ where } \widehat{P}_{\min} = \frac{\pi^2 E_{\min} I}{(\mathcal{K}\cdot L)^2} \leq \frac{\pi^2 E I}{(1-\nu^2)(\mathcal{K}\cdot L)^2}, \tag{15}$$

$E_{\min}$ is the lower bound of E in (14) and constant $\mathcal{K}$ is given in Remark 1. The inequality (15) is obvious with respect to Theorem 1 and Remark 1.

The parameter identification problem reads as follows:

$$\begin{cases} \text{Find } (E^*,\nu^*) \in U_{ad}, \text{ such that} \\ J(w(E^*,\nu^*)) = \min\limits_{(E,\nu)\in U_{ad}} J(w(E,\nu)), \\ \text{where } w(E,\nu) \text{ solves } (\mathcal{P}(E,\nu)) \\ \text{and } J : V \longrightarrow \mathbb{R} \text{ is a cost functional.} \end{cases} \tag{$\mathbb{P}$}$$

Continuous dependence of the solution $w(E,\nu)$ on the material parameters (E,ν) is stated in the following theorem which was published in [11] but only for the boundary conditions (B1) which correspond to the space V_1. Here, we will present the extension of the previous results for the remaining boundary conditions (B2), (B3) and (B4) and the spaces V_2, V_3 and V_4. Unless distinguished the space V will be one of above $V_1,\ldots,V_4$. In the previous section we introduced the inequalities which will be used in a proof of the following Theorem. In case of the boundary conditions (B3) we have to consider a stronger assumption for axial force with respect to (15) and (10). Thus let

$$P < \widehat{P}^i_{\min}, \text{ where } \widehat{P}^i_{\min} = \frac{\pi^2 E_{\min} I}{(\mathcal{K}_i\cdot L)^2} \leq \frac{\pi^2 E I}{(1-\nu^2)(\mathcal{K}_i\cdot L)^2}, \tag{16}$$

where $\mathcal{K}_i, i = 1,2,3$, is given with respect to the inequalities from Section 2 and the boundary conditions. It means that in the following we will be working under the assumptions: let $\mathcal{K}_1 = 1$ for the boundary conditions (B1), $\mathcal{K}_2 = 0.5$ for (B2) and $\mathcal{K}_3 = 2$ for the boundary conditions (B3) and (B4).

Theorem 4. *Let $(E_n,\nu_n) \in U_{ad}$, $n = 1,2,\ldots$ and $(E,\nu) \in U_{ad}$, such that*

$$E_n \underset{n\to\infty}{\longrightarrow} E \text{ in } L^\infty(0,L) \quad \text{and} \quad \nu_n \underset{n\to\infty}{\longrightarrow} \nu \text{ in } L^\infty(0,L)$$

and $w_n := w(E_n,\nu_n) \in V$ be the solution to $(\mathcal{P}(E_n,\nu_n))$. Then

$$w_n \underset{n\to\infty}{\longrightarrow} w(E,\nu) \in V,$$

and $w(E,\nu)$ solves $(\mathcal{P}(E,\nu))$.

Proof. The proof consists of three steps. First, we show that the sequence $\{w_n\}$ is bounded in V.
Step 1. Let $w_n \in V$ solve $(\mathcal{P}(E_n, \nu_n))$:

$$\int_0^L E_n I w_n'' \, v'' \, \mathrm{d}x + tb \int_0^L E_n (1-\nu_n^2)(w_n')^3 \, v' \, \mathrm{d}x \\ - \int_0^L P(1-\nu_n^2) \, w_n' \, v' \, \mathrm{d}x = \int_0^L (1-\nu_n^2) \, q \, v \, \mathrm{d}x, \quad \forall v \in V.$$

We set $v := w_n$ and get

$$\int_0^L E_n I (w_n'')^2 \, \mathrm{d}x + tb \int_0^L E_n (1-\nu_n^2)(w_n')^4 \, \mathrm{d}x - \int_0^L P(1-\nu_n^2) \, (w_n')^2 \, \mathrm{d}x = \int_0^L (1-\nu_n^2) \, q \, w_n \, \mathrm{d}x. \tag{17}$$

From (14) it is clear that $1-\nu_n^2 > 0$, since $0 < \nu_n \leq 0.5$. Therefore,

$$tb \int_0^L E_n (1-\nu_n^2) \, (w_n')^4 \, \mathrm{d}x \ \geq \ 0, \quad \forall (E_n, \nu_n) \in U_{ad}. \tag{18}$$

To estimate the term with the axial force P, we will apply the inequalities and their modifications presented in Section 2 according to the boundary conditions (B1)–(B4). It is clear that for the space V of admissible displacements $H_0^2((0,L)) \subset V \subset H^2((0,L))$ holds. In the following, we will use the fact that the space $H^k((0,L))$, $k = 1, 2, \ldots$, can be continuously embedded into $C^{k-1}(\langle 0, L\rangle)$, see [1]. Especially, we have

$$\exists \, c > 0 \colon \ \max_{x \in \langle 0, L\rangle} |v'(x)| \ \leq \ c \|v\|_2 \quad \forall v \in H^2((0,L)),$$

where $\|\cdot\|_k$, $k = 0, 1, \ldots$, denotes the norm in $H^k((0,L))$. We will also use the following inequality

$$\exists \, \bar{c} > 0 \colon \|v''(x)\|_0^2 \ \geq \ \bar{c} \|v\|_2^2 \quad \forall v \in V, \tag{19}$$

which holds for any V defined by the boundary conditions (B1), (B2), (B3), or (B4). For functions v from V_2, V_3 and V_4 we have $v(0) = v'(0) = 0$, thus we can use twice the generalization (9) of Theorem 3, first for $y = v$ and then for $y = v'$. So we can write

$$\int_0^L (v(x))^2 \, \mathrm{d}x \ \leq \ \left(\frac{2L}{\pi}\right)^2 \int_0^L (v'(x))^2 \, \mathrm{d}x \ \leq \ \left(\frac{2L}{\pi}\right)^4 \int_0^L (v''(x))^2 \, \mathrm{d}x,$$

which gives us the inequality (19) with $\bar{c} = \left(\frac{\pi}{2L}\right)^4$. For functions from V_1 the inequality (19) holds with constant $\bar{c} = 4\left(\frac{\pi}{L}\right)^4$, which follows from (13).

First, we consider the simply supported beam with boundary conditions (B1). Since $0 < \nu_n \leq 0.5$, we have $1 - \nu_n^2 < 1$ and using the inequality (12), we get that

$$\int_0^L P(1-\nu_n^2)(w_n'(x))^2 \, \mathrm{d}x \ \leq \ \left(\frac{L}{\pi}\right)^2 \int_0^L P \, (w_n''(x))^2 \, \mathrm{d}x \tag{20}$$

holds for any $P \geq 0$. Then we can write

$$\begin{aligned}\int_0^L E_n\, I\, (w_n'')^2\,\mathrm{d}x + tb\int_0^L E_n(1-\nu_n^2)(w_n')^4\,\mathrm{d}x \;-\; \int_0^L P(1-\nu_n^2)(w_n')^2\,\mathrm{d}x \\ \geq \int_0^L E_n\, I(w_n'')^2\,\mathrm{d}x \;-\; P\left(\frac{L}{\pi}\right)^2\int_0^L (w_n'')^2\,\mathrm{d}x \\ \geq \int_0^L E_{min}\, I(w_n'')^2\,\mathrm{d}x \;-\; P\left(\frac{L}{\pi}\right)^2\int_0^L (w_n'')^2\,\mathrm{d}x \\ =\; c_1\,\|w_n''\|_0^2 \;\geq\; \bar{c}\,c_1\,\|w_n\|_2^2, \end{aligned} \tag{21}$$

where we used (18), (20), (19) and the notation $c_1 := E_{\min}I - P\left(\frac{L}{\pi}\right)^2$. The constant c_1 is positive due to assumption (16). If $P < 0$ then (21) trivially holds with $c_1 = E_{\min}I$.

For the clamped beam with the boundary conditions (B2), i.e. $w(0) = w'(0) = w(L) = w'(L) = 0$, we estimate the term with the axial force by the inequality (6) and by using $1 - \nu_n^2 < 1$. Therefore, we have

$$\int_0^L P(1-\nu_n^2)(w_n'(x))^2\,\mathrm{d}x \;\leq\; \left(\frac{L}{2\pi}\right)^2\int_0^L P\,(w_n''(x))^2\,\mathrm{d}x. \tag{22}$$

For the propped cantilever and cantilever beam with the boundary conditions (B3) and (B4), respectively, the inequality (10) together with $1 - \nu_n^2 < 1$ can be used, i.e.

$$\int_0^L P(1-\nu_n^2)(w_n'(x))^2\,\mathrm{d}x \;\leq\; \left(\frac{2L}{\pi}\right)^2\int_0^L P\,(w_n''(x))^2\,\mathrm{d}x. \tag{23}$$

Similarly as for (B1) now we can get by using the inequalities (22), (23) that

$$\begin{aligned}\int_0^L E_n\, I\, (w_n'')^2\,\mathrm{d}x + tb\int_0^L E_n(1-\nu_n^2)(w_n')^4\,\mathrm{d}x \;-\; \int_0^L P(1-\nu_n^2)(w_n')^2\,\mathrm{d}x \\ \geq\; c_i\,\|w_n''\|_0^2 \;\geq\; \bar{c}\,c_i\,\|w_n\|_2^2, \end{aligned} \tag{24}$$

where $i = 2, 3$, $c_2 := E_{\min}I - P\left(\frac{L}{2\pi}\right)^2 > 0$ and $c_3 := E_{\min}I - P\left(\frac{2L}{\pi}\right)^2 > 0$. If $P < 0$ then (24) is trivially valid with $c_2 = c_3 = E_{\min}I$.

For the right hand side in (17) we get

$$\int_0^L (1-\nu_n^2) q\, w_n\,\mathrm{d}x \;\leq\; \|q\|_{L^2((0,L))}\|w_n\|_2, \tag{25}$$

where Hölder's inequality and (14) were used. Finally, from (17), (21), (24) and (25) we see that $\{w_n\}$ is bounded in V. Therefore, there exists its subsequence, for simplicity we denote it as $\{w_n\}$ again, such that

$$w_n \underset{n\to\infty}{\rightharpoonup} w \quad \text{(weakly) in } V.$$

Step 2. Now we show that w solves (2). Similarly as in [11], it can be proven for each $v \in V$ that

$$\int_0^L E_n\, I\, w_n'' \, v''\, \mathrm{d}x \,+\, tb \int_0^L E_n(1-\nu_n^2)(w_n')^3\, v'\, \mathrm{d}x \,-\, \int_0^L P(1-\nu_n^2) w_n'\, v'\, \mathrm{d}x$$
$$= \int_0^L (1-\nu_n^2) q\, v\, \mathrm{d}x \underset{n\to\infty}{\longrightarrow} \int_0^L EI w''\, v''\, \mathrm{d}x \,+\, tb \int_0^L E(1-\nu^2)(w')^3\, v'\, \mathrm{d}x$$
$$- \int_0^L P(1-\nu^2) w' v' \mathrm{d}x \,=\, \int_0^L (1-\nu^2) q v\, \mathrm{d}x.$$

Step 3. To prove the strong convergence, it is sufficient to show that $[[w_n]] \to [[w]]$ for $n \to \infty$ in V, where

$$[[w]]^2 \,:=\, \int_0^L EI(w''(x))^2\, \mathrm{d}x.$$

For more details, see [11], [3]. □

To prove the existence of at least one solution of the identification problem ($\mathbb{P}$), see [11], we suppose that the cost functional J is continuous in V, i.e.

$$v_n \underset{n\to\infty}{\longrightarrow} v \quad \Longrightarrow \quad J(v_n) \underset{n\to\infty}{\longrightarrow} J(v). \tag{26}$$

Theorem 5. *Let U_{ad} be given by (14) and let J satisfy (26). Then the identification problem ($\mathbb{P}$) has a solution.*

4. Conclusion

In this paper, we discuss the extension of the results presented in [11]. Several analytical inequalities and their modifications were used to prove the continuous dependence of the solution to the state problem on the material parameters for different types of boundary conditions for the nonlinear Gao beam.

Acknowledgements

The authors acknowledge both the support by the grant IGA PrF IGA_PRF_2021_008 Mathematical Models of the Internal Grant Agency of Palacký University in Olomouc, Czech Republic, and by the Ministry of Education, Youth and Sports of the Czech Republic under the project CZ.02.1.01/0.0/0.0/17_049/0008408 Hydrodynamic design of pumps.

References

[1] Adams, R. A.: *Sobolev Spaces.* Academic Press: London, UK, 2003.

[2] Blaschke, W.: *Kreis und Kugel.* Verlag von Veit & Comp, Leipzig, 1916.

[3] Burkotová, J., Machalová, J., Radová, J.: Optimal thickness distribution of stepped nonlinear Gao beam. Mathematics and Computers in Simulation **189** (2021), 21–35.

[4] Eisley, J. G., Waas, A. M.: *Analysis of Structures. An Introduction Including Numerical Methods*. John Wiley and Sons, 2011.

[5] Gao, D. Y.: Nonlinear elastic beam theory with application in contact problems and variational approaches. Mechanics Research Communications **23** (1) (1996), 11–17.

[6] Komkov, V.: Euler's buckling formula and Wirtinger's inequality. International Journal of Mathematical Education in Science and Technology **14** (6) (1983), 661–668.

[7] Komkov, V.: *Variational Principles of Continuum Mechanics with Engineering Applications. Volume 1: Critical Points Theory.* Vol. 24. Springer Science & Business Media, 1986.

[8] Machalová, J., Netuka, H.: Control variational method approach to bending and contact problems for Gao beam. Applications of Mathematics **62** (6) (2017), 661–677.

[9] Machalová, J., Netuka, H.: Comments on the large deformation elastic beam model developed by D. Y. Gao. Mechanics Research Communications **110** (2020), 103607.

[10] Mitrinović, D. S., Vasić, P. M.: *Analytic Inequalities*. Vol. 1, Springer, 1970.

[11] Radová, J., Machalová, J., Burkotová, J.: Identification problem for nonlinear Gao Beam. Mathematics **8** (11) (2020), 1916.

Programs and Algorithms of Numerical Mathematics 21
J. Chleboun, P. Kůs, J. Papež, M. Rozložník, K. Segeth, J. Šístek (Eds.)
Institute of Mathematics CAS, Prague 2023

DIFFERENT BOUNDARY CONDITIONS FOR LES SOLVER PALM 6.0 USED FOR ABL IN TUNNEL EXPERIMENT

Hynek Řezníček[1,3], Jan Geletič[1], Martin Bureš[1], Pavel Krč[1], Jaroslav Resler[1], Kateřina Vrbová[2], Arsenii Trush[2], Petr Michálek[2], Luděk Beneš[3], Matthias Sühring[4]

[1] Institute of Computer Science, Czech Academy of Sciences, Prague (cs.cas.cz - reznicek@cs.cas.cz)
[2] Institute of Theoretical and Applied Mechanics, Czech Academy of Sciences, Prague (itam.cas.cz)
[3] Faculty of Mechanical Engineering, Czech Technical University in Prague (fs.cvut.cz)
[4] Institute of Meteorology and Climatology, Leibniz University in Hanover (meteo.uni-hannover.de)

Abstract: We tried to reproduce results measured in the wind tunnel experiment with a CFD simulation provided by numerical model PALM. A realistic buildings layout from the Prague-Dejvice quarter has been chosen as a testing domain because solid validation campaign for PALM simulation of Atmospheric Boundary Layer (ABL) over this quarter was documented in the past. The question of input data needed for such simulation and capability of the model to capture correctly the inlet profile and its turbulence structure provided by the wind-tunnel is discussed in the study.
The PALM dynamical core contains a solver for the Navier-Stokes equations. By default, the model uses the Large Eddy Simulation (LES) approach in which the bulk of the turbulent motions is explicitly resolved. It is well validated tool for simulations of the complex air-flow within the real urban canopy and also within its reduced scale provided by wind tunnel experiments. However the computed flow field between the testing buildings did not correspond well to the measured wind velocity in some points. Different setting of the inlet boundary condition was tested but none of them gave completely developed turbulent flow generated by vortex generators and castellated barrier wall place at the entrance of the aerodynamic section of the wind tunnel.

Keywords: large eddy simulation, wind tunnel, atmospheric boundary layer, PALM model, turbulence

MSC: 65Z05, 86A10, 76F65

DOI: 10.21136/panm.2022.19

1. Introduction

PALM model is capable to simulate turbulent air-flow within the lowest part of the ABL. By default, it uses the LES approach in which the bulk of the turbulent motions is explicitly resolved [4]. The core was already validated according to tunnel measurements in [2], therefore our expectations were high.

The realistic buildings layout from Prague-Dejvice quarter is chosen as the testing domain. The choice of this particular domain is motivated by existing validation for the PALM model in Dejvice quarter [5]. The same domain was 3D-printed and placed to the test section of the wind tunnel in Telč (Vincenc Strouhal) owned by ITAM which calibration is documented in [3]. To achieve the flow similar to real ABL in reduced scale the wind tunnel used three elements of the turbulence generation - vortex spikes and castellated barrier wall before the atmospheric section and roughness elements inside the atmospheric section (before the model test section).

Originally we were interested to model the influence of passageways inside the buildings on the flow field in courtyards and we wanted to compare our numbers to ones measured in the tunnel by 5-holes probe. The inconsistency in the results for the base-case forced us to study the problem how to correctly reproduce the well defined but still vaguely described (in a certain sense) flow field in the wind tunnel's atmospheric test section.

The question should be which data and in which form are needed from the measurement (or calibration) for the CFD models to set the inflow properly.

2. Mathematical model

The simulated air is considered as incompressible (due to much lower velocities in comparison to the speed of sound), viscid (the molecular viscosity is neglected everywhere except for the turbulent dissipation) and neutrally stratified (for testing the dynamical core only without unfavourable stratification effects) gas.

The dynamical core of PALM model is based on Navier-Stokes equations in Boussinesq approximation for filtered quantities (filtering usually denoted with overbar is omitted here due to readability)

$$\begin{aligned} \nabla \cdot \mathbf{u} &= 0 \\ \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} &= -\frac{1}{\rho_0}\nabla\pi + \mathbf{g} - \nabla \cdot \underline{\underline{\boldsymbol{\tau}}} \end{aligned} \tag{1}$$

The velocity vector $\mathbf{u} = u_i = (u, v, w)$ describes the movement of air which is assumed to be dry with constant density $\rho_0 = 1\ \mathrm{kg/m^3}$. The gravitational acceleration denoted as $\mathbf{g} = -g\delta_{i3}$ is acting only in vertical direction (here written using Kronecker's delta δ_{ij} in third component), its value is set to $g = 9.81\ \mathrm{m/s^2}$. The modified pressure fluctuation can be expressed as $\pi = p + \frac{2}{3}\rho_0 e$ using the pressure fluctuation p and sub-grid-scale (sgs = unresolved) turbulent kinetic energy e. The residual stress tensor $\underline{\underline{\boldsymbol{\tau}}} = \tau_{ij}$ symbolises the turbulent part of the flow.

The modified Deardorff's model is employed for turbulent closure (written in Einstein summation convention follows)

$$\begin{aligned} \tau_{ij} = \overline{u_i'' u_j''} - \frac{2}{3} e \delta_{ij} &= -K_m \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \\ \frac{\partial e}{\partial t} + u_j \frac{\partial e}{\partial x_j} &= 2K_m \nabla^2 e + \overline{u_i'' u_j''} \frac{\partial u_i}{\partial x_j} - \epsilon \end{aligned} \tag{2}$$

A double prime indicates sgs velocities, the overbar indicating filtering is added for the sgs flux terms. The local (sgs) eddy diffusivity coefficient of momentum K_m is approximated as $K_m \approx 0.1\Delta\sqrt{e}$, where distance $\Delta = \min\{\Delta_{x_i}\}$ is minimal grid spacing. This distance serves also as implicit filter for large eddies. The dissipation rate is approximated as $\epsilon \approx 0.93\frac{\sqrt{e^3}}{\Delta}$. For more details please see the documentation in [4]. Further the dimensions are referenced as $x_i = (x, y, z)$.

2.1. Numerical solver

The equations are spatially discretized by using finite differences at equidistant horizontal grid spacing while the vertical grid is stretched above the surface layer to save CPU-time. The stretching factor applied above 100 m height set to 1.01 is limited by maximal vertical step ($\max \Delta_z = 2\Delta_x$). Arakawa staggered C-grid is used for velocity **u** defined at edges of the grid cell while the scalars are defined in the grid cell center (see Fig. 1). The Upwind-biased 5^{th} order advection scheme based on flux formulation according to [6] is used.

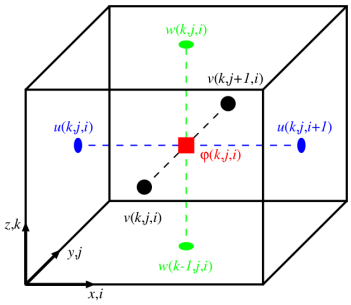


Fig. 1: Arakawa staggered C-grid [4]

The time integration is done by 3^{rd} order low-storage (3 stages) Runge-Kutta method according to [1]. It is proved that the CFL condition in such case can be $C_{\mathrm{CFL}} = \frac{\max_i\{u_i\}\Delta_t}{\Delta} < 1.4$ which limits the maximal time step Δ_t.

To enforce incompressibility (divergence free velocity field needed by Bouissinesq approx.) a predictor-corrector method is used where Poisson equation is solved for the modified perturbation pressure (π) after every time step. The resulting system of linear equations is solved with Gauss-Seidel method and multi-grid scheme is employed if number of cells per core is even . The detailed description can be found in [4].

3. Set-up and boundary conditions

As mentioned above, the measurement was done in wind tunnel at the ITAM in Telč, detailed description can be found in [3] and mentioned references. What is important to notice here is the arrangement of elements generating the ABL flow before the aerodynamic test section as can be seen in the Fig. 2, because the castellated barrier wall and vortex generators are not a part of the numerically modeled

domain. Their placement to the numerical domain close to the inlet would caused a significant extension of the domain and therefore slowing down the computations.

The real world building configuration in Prague-Dejvice between streets Jugoslavskych partyzanu and Terronska was chosen as testing area (serves as inner domain in the model). The Fig. 3 shows the map of the area with blue circle indicating the passage-way in the building in Rooseveltova street. The area is rotated clockwise to adjust the air flow with x-direction. The situation in the test section (inner domain in the model) is shown in the Fig. 4 with marked measuring point locations.

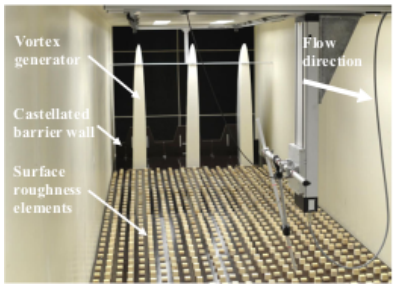


Fig. 2: View from the aerodynamic test section backwards (against the flow).

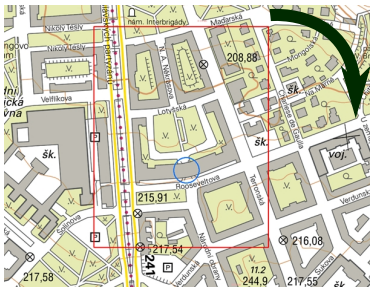


Fig. 3: Map of chosen area. The domain is rotated clockwise in the model.

Fig. 4: The same area as inner domain with measuring points.

The scale of the model is 1:300 which holds for time and space meaning that the 1 min. average in the tunnel is 5 hours average in real. A characteristic length is chosen as the height of the vortex generator which is $H = 1.5$ m in the tunnel which corresponds to 450 m in reality. The advantage of the scale setting is that the velocities can be compared 1:1. If the reference velocity $U_{\text{ref}} = 6.6$ m/s is considered, the Reynolds number in the tunnel is $Re \approx 10^6$ which is large enough. If Townsend's hypothesis applies, the flow in the wind tunnel should be dynamically comparable to the real one. The computational domain contains all the roughness elements (simulated directly) as can be seen in the Fig. 5. The whole domain 3000×600 m large includes the test section with dimensions 600×500 m (all listed as real here). The resolution of the grid is set to $\Delta = 1$ m.

The wind tunnel measurements were performed using five-hole fast response pressure probe "Cobra" with integrated pressure-to-voltage transducers. The probe was mounted on a traversing device that could move it in all three directions x, y, z, see Fig. 2 right. The used sampling frequency was 1000 Hz and sampling time was 60 s for each measuring record. The probe records evaluations were made in MATLAB with the use of routines and calibrations from the probe manufacturer, Aeroprobe Corporation.

The wind velocity and Turbulent Intensity (TI) profiles measured by a hot-wire above the roughness elements (before the inner domain) were available from the validation article [3]. For this study the profiles measured before the inner domain

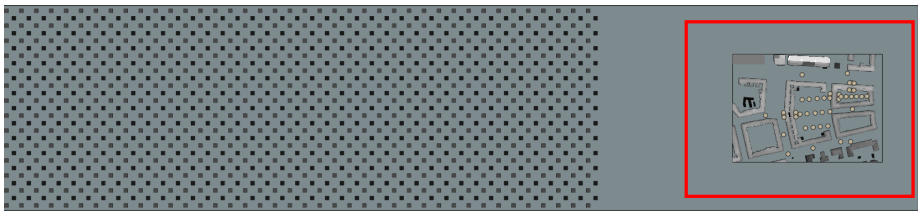


Fig. 5: Computational domain with buildings in the test section (marked in red).

are important, specially profile measured at the beginning of inner domain and profile measured approximately one meter before inner domain. The maximal height of the profiles is 0.47H and the height of our simulation domain was chosen accordingly (the height of the wind tunnel channel is 1.3H). The values for the profiles were taken from wind tunnel validation measurement provided in [3] since the measurement of these profiles during our experiment wasn't accomplished.

3.1. Boundary conditions

The boundary conditions (b.c.) are set as follows:

On **inlet** the vertical velocity profile driven by $U_{\text{ref}} = 6.6$ m/s is prescribed for the first velocity component $u = u(z)$ profile (different profiles were tested - uniform, logarithmic and power law) with statistically created disturbances every 60 s with amplitude ± 0.25 m/s from the mean velocity. The other components are set to $v = w = 0$. Homogeneous (homog.) Neumann b.c. is prescribed for the other quantities (e, p).

On **outlet** a radiation b.c. [4] is used for all velocity components where a constant phase velocity is considered as maximum value allowed by CFL condition. Homog. Neumann condition is assumed for scalar quantities (e, p).

At the **bottom** homog. Dirichlet b.c. for velocity vector $\mathbf{u}(0) = \mathbf{0}$ (no slip) is used. Homog. Neumann b.c. is prescribed for the other quantities (e, p).

At the **top** boundary Dirichlet b.c. for the first velocity component is given by the inlet profile as $u(z_{\max}) = \max u(z)$. For the other components and the pressure perturbation the homog. Dirichlet b.c. is utilized. Homog. Neumann condition is assumed for sgs-tke (e).

On **sides** the cyclic b.c. is prescribed for all quantities.

4. Results

The whole aerodynamic section (including the roughness elements in front of the test section) of the wind tunnel was simulated and the results were compared to the measurements. The main comparison was done for velocity components in the given points (see Fig. 4) obtained by five-hole probe for three different heights: 3, 10

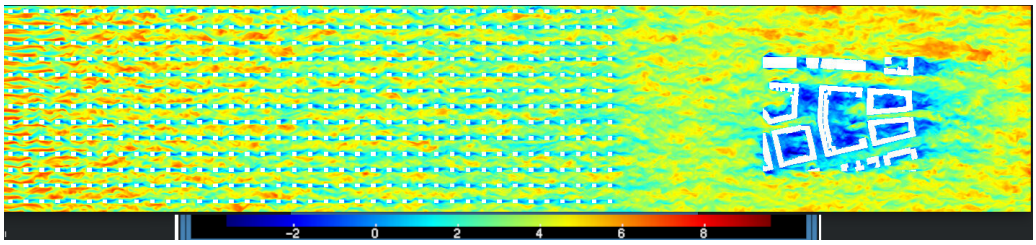


Fig. 6: Horizontal velocity field in $z = 9$ m captured after 1 hour simulation (u - instantaneous values).

and 30 m (listed as real dimensions). Nevertheless, the agreement of the velocity and turbulent intensity profiles measured by hot-wire probe in the tunnel axes in front of the test section was also important.

The first numerical experiment was performed with uniform inlet velocity profile $u(z) = U_{\text{ref}}$ and was considered as naive attitude serving as technical preview. The flow was decelerated and its turbulent intensity was increased within the roughness elements section. The example of such horizontal velocity field (u) in 9 m height captured in a moment when the simulation time hits 1 hour is shown in the Fig. 6.

The model outputs were mainly saved every 30 minutes as time averages and then their mean over 5 hours simulation were computed. Example of such output for velocity component u in the test section is rendered in the Fig. 7. There, one can easily identify the influences of the buildings and their recirculation zones. Also, the influence of the passageway is identified in the middle of the U-shaped building which allows some air to go through. Therefore the flow behind the passageway is faster than the flow in the surrounding area.

The hit ratio for velocity magnitudes displayed in the Fig. 8 shows the discrepancy between model and experimental results. The values given by the model are seriously under-predicted in most points (somewhere more than by 25% - as indicate the lower green line). The reason probably lies in wrong velocity profile at the entrance of the

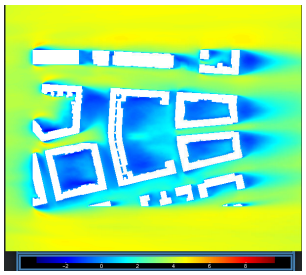


Fig. 7: 30 minutes average for u, in $z = 9$ m, $t = 1$ h (In the test section).

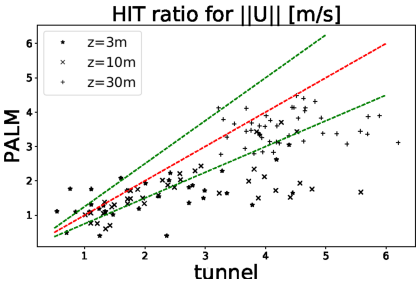


Fig. 8: Mean velocity magnitudes hit.

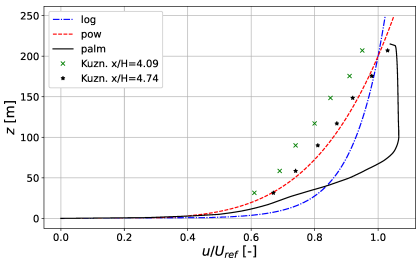


Fig. 9: Velocity profiles at the beginning of the test section.

inner domain. In the Fig. 9 the vertical profiles of u velocity in the tunnel axes in front the test section are plotted (at the distance $x/H = 4.74$). For illustration, two possible theoretical profiles are plotted in the graph as well as experimental profile at the different distance ($x/H = 4.09$). The profile from PALM (black line) behaves differently than the experimental profile (Kuzn. [3]).

The Fig. 10 displays a situation inside the test section for each point in height 30 m. The points are coloured according to formula

$$\left(\frac{u_{\text{palm}}}{u_{\text{exp.}}} - 1\right) \cdot 100\%,$$

which means how accurately they hit the experimental value. Some patterns can be identified in the figure, such that the velocities inside the closed building block fit well and the values in front of the closed building block are very underpredicted, but the rest seems quite random. That leads us to consideration of wrong turbulent structure in the simulation probably caused by lacking of right tools and information how to prescribe it at the inlet (as the results from the first experiment suggested).

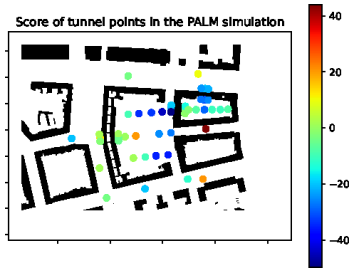


Fig. 10: Measuring points colored according $(u_{\text{palm}}/u_{\text{exp.}} - 1) \cdot 100\%$

Profile in the Fig. 9 reveals problems with different flow rate between simulated and measured profiles, but this issue can be related to the wrong top boundary condition. Or the upper boundary condition (the top of the computational domain) is simply placed too low to satisfy fully Dirichlet b.c. (without any inflow through upper boundary). Sadly the provided information from the validation data doesn't contained any velocities for higher heights (z-coordinate). Yet the data suggests the flow rate is changed between profiles - the first profile at $x/H = 4.09$ contains smaller velocities than the second one at $x/H = 4.74$ (at the beginning of inner domain). Therefore the flow rate to the numerical domain from above is unknown and it cannot be properly simulated.

The first numerical experiment at least confirmed that convergence of the model is achieved relatively quickly. As Fig. 11 shows, the steady state in terms of kinetic energy and resolved Turbulent Kinetic Energy (TKE) conservation is reached approximately after 15 min. of the simulation. The spectral density of TKE corresponds well to the Kolmogorov's cascade as plotted in the graph of Fig. 12.

The comparison of Turbulent Intensity (TI) profiles in the Fig. 13 for the PALM outputs and calibration measurement with hot-wire (in the tunnel axes ahead of the test section) indicates the good ability of the model to capture well the empty tunnel with roughness elements only (when the castellated barrier wall and the vortex generators weren't present). On the other hand the fully developed profiles of turbulence and velocity (will be described further) of the fully equipped wind tunnel (with all three elements generating similar flow to ABL) are difficult to get from the model. Probably they have to be prescribed as an inlet b.c. which includes the complete

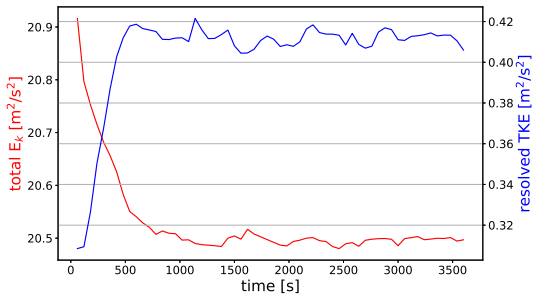


Fig. 11: Convergence for kinetic energy and Turbulent Kinetic Energy (TKE) conservation.

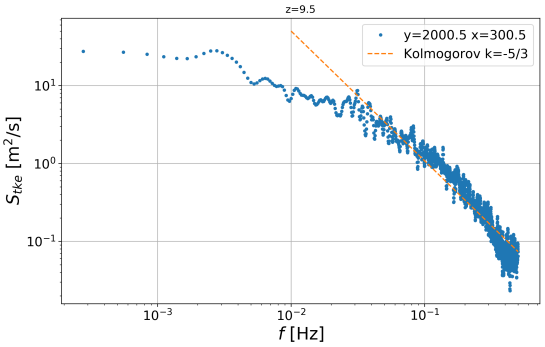


Fig. 12: Spectral density of resolved TKE.

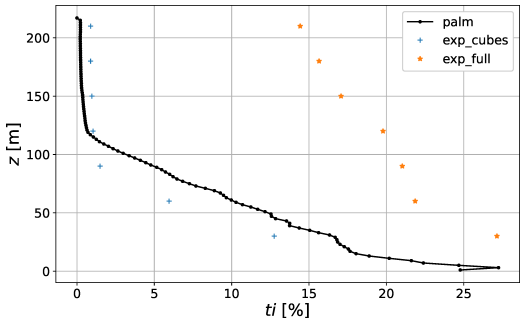


Fig. 13: Vertical profiles of Turbulent Intensity at the beginning of the test section.

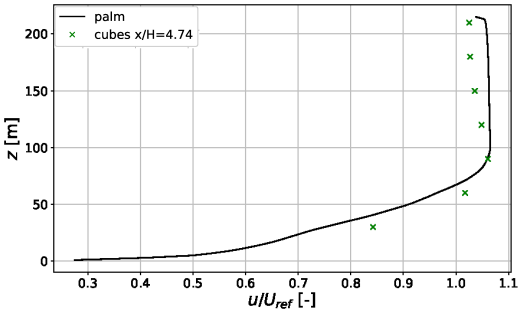


Fig. 14: Velocity profile compared to empty tunnel with roughness elements only (cubes).

information about turbulence structure. When the dimensionless velocity profiles are compared to the empty tunnel profiles with roughness elements (cubes) only in the Fig. 14, they fit quite well.

Other numerical experiments were conducted with the cyclic b.c. (inlet/outlet), different inputs (logarithmic law, power law) and with switching (ON/OFF) the turbulence disturbances at the inlet, but none of them led to systematical improvement. Even the simulation with artificial tunnel walls (by adding high buildings to sides) was tried but the changes were in terms of percents (and not systematically).

The last numerical experiment to be mentioned here used the known velocity profile to obtain results closer to the experiment. The power law velocity profile

$$u(z) = U_{\text{ref}} \left(\frac{z}{z_{\text{ref}}} \right)^{\alpha}, \tag{3}$$

with coefficient $\alpha = 0.22$ and $z_{\text{ref}} = 200$ m, was prescribed on the inlet. The turbulence on the inlet was generated by the disturbances (without synthetic turbulent generator). As shown in Fig. 15 the profile still doesn't match fully developed state. The flow is decelerated near the ground much more than expected.

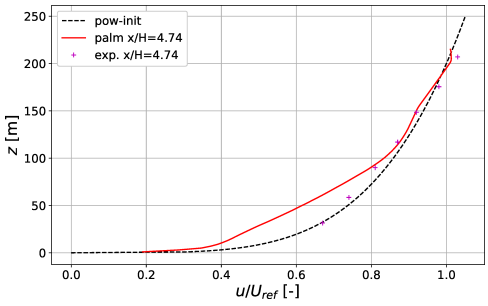


Fig. 15: Velocity profiles for the second numerical experiment.

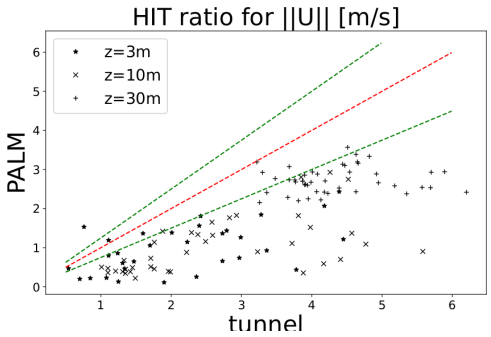


Fig. 16: Mean velocity magnitudes.

The hit ratio in that case is even worse as is indicated in the Fig. 16. However, it is not surprising because the simulated flow entering the testing domain in this case is much slower than the physical flow in the wind-tunnel.

5. Conclusions

A big simulation (containing circa 25×10^8 cells) of the whole wind-tunnel atmospheric section was performed by atmospheric LES model PALM and its results were compared to the measurements. It was shown that kinetic energy conservation is achieved relatively quickly and the calculated turbulence spectrum corresponds to the theory. The results obtained for the "naive" uniform initial velocity profile were promising but not satisfying. The model was able to develop the correct profile over the roughness elements without vortex spikes and castellated barrier wall quite well in the case of velocity and even of turbulent intensity.

The reproduction of ABL is a challenging question even for smaller scales and well defined condition of a wind tunnel. Based on the data provided by the validation paper the model PALM is unable to reproduce the fully developed wind profile with correctly generated turbulence structures. It leads to strong under-estimation of the velocities inside the street canyon. The measurement of the main flow should be provided much higher or the flow rate through top boundary should be known. As is shown PALM can reproduce the boundary layer created with the roughness cubes only. For recreation of the boundary layer produced by the other elements (vortex spikes and castellated barrier wall) the complete recirculation zone measurements is probably needed and the top boundary of the computing domain has to be probably placed much higher. To conclude that the results of the model are limited with prescription of correct turbulent structure and the known (well developed) velocity profile. Unfortunately, such profile wasn't provided by the experimenters and during our numerical experiments it wasn't found. The question, how to impose the correct profile (even if we know it), remains for the future testing.

The future endeavors are pointed to the simulation of cyclic domain (infinite) with smaller part serving as precursor where the correct profile could be developed.

Also we hope that we can adopt some knowledge obtained by testing original code provided by [2]. If it was possible we would ask for the new measurements with the empty tunnel with roughness elements only to see whether the well defined inlet improved our hit ratio.

Acknowledgements

This work was supported by the Strategy AV21 projects "Energy interactions of buildings and the outdoor urban environment" and "City as a Laboratory of Changes", financed by the Czech Academy of Sciences. The data post-processing was possible due to the student grant SGS22/148/OHK2/3T/12 of the Czech Technical University in Prague.
The measurements were possible due to the support of grant GACR 22-08786S financed by The Czech Science Foundation (GACR). The 3D-geometry for the model was kindly provided by Operátor ICT, a.s. (operatorict.cz).

References

[1] Baldauf, M.: Stability analysis for linear discretisations of the advection equation with Runge–Kutta time integration. Journal of Computational Physics **227** (2008).

[2] Gronemeier, T. et al.: Evaluation of the dynamic core of the PALM model system 6.0 in a neutrally stratified urban environment: comparison between LES and wind-tunnel experiments. Geoscientific Model Development **14** (2021).

[3] Kuznetsov, S. et al.: Flow and turbulence control in a boundary layer wind tunnel using passive hardware devices. Experimental Techniques **41** (2017).

[4] Maronga, B. et al.: The parallelized large-eddy simulation model (PALM) version 4.0 for atmospheric and oceanic flows: model formulation, recent developments, and future perspectives. Geoscientific Model Development **8** (2015).

[5] Resler, J. et al.: Validation of the PALM model system 6.0 in a real urban environment: A case study in Dejvice, Prague, the Czech Republic. Geoscientific Model Development **14** (2021).

[6] Wicker L. J., S. W.: Time-splitting methods for elastic models using forward time schemes. Mon. Weather Rev. **130** (2002).

Programs and Algorithms of Numerical Mathematics 21
J. Chleboun, P. Kůs, J. Papež, M. Rozložník, K. Segeth, J. Šístek (Eds.)
Institute of Mathematics CAS, Prague 2023

SPHERICAL BASIS FUNCTION APPROXIMATION WITH PARTICULAR TREND FUNCTIONS

Karel Segeth

Institute of Mathematics, Czech Academy of Sciences
Prague, Czech Republic
segeth@math.cas.cz

Abstract: The paper is concerned with the measurement of scalar physical quantities at nodes on the $(d-1)$-dimensional unit sphere surface in the d-dimensional Euclidean space and the spherical RBF interpolation of the data obtained. In particular, we consider $d = 3$. We employ an inverse multiquadric as the radial basis function and the corresponding trend is a polynomial of degree 2 defined in Cartesian coordinates. We prove the existence of the interpolation formula of the type considered. The formula can be useful in the interpretation of many physical measurements. We show an example concerned with the measurement of anisotropy of magnetic susceptibility having extensive applications in geosciences and present numerical difficulties connected with the high condition number of the matrix of the system defining the interpolation.

Keywords: spherical interpolation, spherical radial basis function, trend, inverse multiquadric, magnetic susceptibility

MSC: 65D12, 65D05, 65Z05

1. Introduction

The aim of this paper is to present some ways of approximating and mapping measured physical quantities exhibiting anisotropy that can be expressed by means of a second-order tensor. A typical example is the measurement of magnetic susceptibility of rock having extensive applications in geosciences [11].

In the paper, we develop the data interpolation and approximation with the help of spherical radial basis functions in such a case, cf. [6]. The functions appearing in the formula are the basis functions chosen as radial functions and the trends, cf. [1].

In the laboratory determination of raw data, cf. [5], [8], [11], the rock sample rotates in magnetic field in a set of selected directions and the data items s_i measured are of the form

$$s_i = z_i^{\mathrm{T}} K z_i + e_i, \tag{1}$$

DOI: 10.21136/panm.2022.20

where z_i are the unit vectors in the ith direction of measurement in Cartesian coordinates, K is a tensor and e_i are deviations from the theoretical tensor model.

An appropriate rotation of the coordinate system can make the tensor K diagonal,

$$K = \begin{bmatrix} K_1 & 0 & 0 \\ 0 & K_2 & 0 \\ 0 & 0 & K_3 \end{bmatrix},$$

where K_1, K_2, K_3 are *principal susceptibilities*. These Cartesian coordinates are basically used for the description of the problem in what follows. We call the graphical representation of the directional susceptibilities (1) the *lemniscate surface,* see Fig. 1. The function s corresponding to (1) is taken for the trend in our considerations that follow.

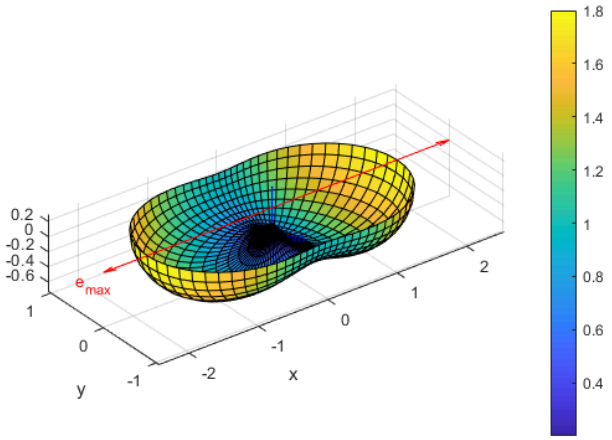


Figure 1: Lower half of the lemniscate surface with $K_1 = 1.8, K_2 = 1.0, K_3 = 0.2$. The magnitude of directional susceptibility in the ith direction z_i is given by the distance between the origin and the surface measured along the vector z_i. The red arrows indicate the direction of the first eigenvector of the susceptibility tensor.

2. Exact and smooth approximation of spherical data

Let d be the dimension of a real Euclidean space $\mathbb{R}^d$. Put

$$S^{d-1} = \{x = (x_1, \ldots, x_d) \in \mathbb{R}^d \mid \sum_{i=1}^{d} x_i^2 = 1\}.$$

Then S^{d-1} is the $(d-1)$-*dimensional surface of unit sphere* in the d-dimensional Euclidean space.

Choose a positive integer N and a nonnegative integer M, $N \geq M$. Given a set $X = \{X_j\}_{j=1}^N$ of mutually distinct nodes $X_j = (X_{j1}, X_{j2}, \ldots, X_{jd})$ on S^{d-1}, then a general formula for the *exact spherical approximant* (interpolant) v has for $x \in S^{d-1}$ the form

$$v(x) = \sum_{j=1}^{N} a_j \psi(g(x, X_j)) + \sum_{k=1}^{M} b_k p_k(x), \tag{2}$$

where a_j, $j = 1, \ldots, N$, and b_k, $k = 1, \ldots, M$, are real coefficients to be found. If $M = 0$, the second sum in (2) is empty.

Further, $\psi\colon [0, \pi] \to \mathbb{R}$ is a continuous real function called the *spherical basis function* (SBF) or *spherical radial basis function* (SRBF). A function $\sigma(x, y)$, $x, y \in \mathbb{R}^d$, is called *radial* if there exists a function $\tau(r)$, $r \geq 0$, such that $\sigma(x, y) = \tau(r)$, where $r = \|x - y\| \in \mathbb{R}$ is the Euclidean norm. The nonnegative function g is the *geodesic metric*, usually $g\colon S^{d-1} \times S^{d-1} \to [0, \pi]$, cf. [6], Section 2.3.

Finally, let $\Pi_t(\mathbb{R}^d)$ be the set of all polynomials $p\colon \mathbb{R}^d \to \mathbb{R}$ with real coefficients and of total degree less then or equal to some nonnegative integer t (called *trends*). Let us formulate the exact approximation (interpolation) problem to be solved, cf. [6], [7]. The smoothing problem will be mentioned in the end of this section.

Given a continuous real target function $f\colon S^{d-1} \to \mathbb{R}$, find the spherical interpolant (2), i.e., a continuous function $v\colon S^{d-1} \to \mathbb{R}$ that satisfies the *interpolation conditions*

$$v(X_i) = f(X_i), \quad i = 1, \ldots, N, \tag{3}$$

where $f(X_i)$ are the values measued at X_i. We use the SBF interpolation formula (2) with a proper geodesic metric g, spherical radial basis function ψ, and trends $p_k \in \Pi_t(\mathbb{R}^d)$, $k = 1, \ldots, M$. We confine ourselves only to real-valued functions and real data in this paper to make the exposition clearer.

Let us employ the matrix notation. We substitute X_i, $i = 1, \ldots, N$, for x in the formula (2) to get

$$v(X_i) = \sum_{j=1}^{N} a_j \psi(g(X_i, X_j)) + \sum_{k=1}^{M} b_k p_k(X_i), \quad i = 1, \ldots, N, \tag{4}$$

and replace the left hand parts $v(X_i)$ of the interpolation conditions (3) with the expressions (4).

Introduce an $N \times N$ symmetric matrix Ψ with the entries

$$\psi_{ij} = \psi(g(X_i, X_j)), \quad i, j = 1, \ldots, N, \tag{5}$$

and an $N \times M$ matrix P with the entries

$$p_{jk} = p_k(X_j), \quad j = 1, \ldots, N, \ k = 1, \ldots, M.$$

Moreover, we denote by $a \in \mathbb{R}^N$, $b \in \mathbb{R}^M$, and $f \in \mathbb{R}^N$ the vectors of the unknowns and the vector of the right hand parts $f(x_i)$ of the interpolation conditions (3).

Note that if $M > 0$ then we have only N interpolation conditions (3) for $N + M$ interpolation coefficients a_j and b_k in the formula (2). Thus, we can impose M additional linear constraints for the individual trends p_k,

$$\sum_{j=1}^{N} a_j p_k(X_j) = \sum_{j=1}^{N} a_j p_{jk} = 0, \quad k = 1, \ldots, M. \tag{6}$$

Now the system of linear algebraic equations to be solved for the unknown vectors a and b consists of (3) and (6), i.e.

$$\begin{aligned} \Psi a + Pb &= f, \\ P^{\mathrm{T}} a &= 0 \end{aligned}$$

or

$$\begin{bmatrix} \Psi & P \\ P^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix}. \tag{7}$$

We put

$$Q = \begin{bmatrix} \Psi & P \\ P^{\mathrm{T}} & 0 \end{bmatrix}, \tag{8}$$

which is a symmetric $(N + M) \times (N + M)$ matrix of the system (7).

We have formulated the general spherical interpolation problem. Apparently, the problem possesses the unique solution if and only if the matrix Q of the system (7) is nonsingular. We employ some conditions guaranteeing that Q is nonsingular. To prove them we need two statements.

Lemma 1. ([3], Theorem 1.23) *Let*

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

be a square matrix, A_{11} its nonsingular submatrix. Then

$$\det[A/A_{11}] = \det A / \det A_{11},$$

where

$$[A/A_{11}] = A_{22} - A_{21} A_{11}^{-1} A_{12}$$

is the Schur complement of the submatrix A_{11} in A.

Lemma 2. ([4], Theorem 4.2.1) *Let the $N \times N$ matrix A be symmetric positive definite and the $N \times M$ matrix Y have rank M, $N \geq M > 0$. Then the $M \times M$ matrix $Y^{\mathrm{T}} A Y$ is also symmetric positive definite.*

Theorem 1. *Let the $N \times N$ principal submatrix Ψ of the $(N+M) \times (N+M)$ matrix Q introduced in (8) be symmetric positive definite and let* rank $P = M$. *Then the matrix Q is nonsingular.*

Proof. (Cf. the proof of Theorem 1 in [9].) Let $[Q/\Psi] = -P^{\mathrm{T}}\Psi^{-1}P$ be the Schur complement of the submatrix Ψ in Q. Then

$$\det Q = \det[Q/\Psi] \det \Psi$$

according to Lemma 1. Further,

$$\det[Q/\Psi] = \det(-P^{\mathrm{T}}\Psi^{-1}P) \neq 0$$

follows for the positive definite matrix Ψ from Lemma 2 as we have assumed rank $P = M$. Finally, we get $\det Q \neq 0$, the matrix Q is nonsingular, and the system (7) has the unique solution. □

Theorem 1 holds only for Ψ positive definite. On the other hand, different assumptions can be imposed on the matrix Q of the system (7), e.g., positive definiteness or conditional positive definiteness of the function ψ, see [7].

Definition 1. ([6]) A continuous function $\psi\colon [0,\pi] \to \mathbb{R}$ is said to be *positive definite on S^{d-1}* (i.e., $\psi \in \mathrm{PD}(S^{d-1})$) if the quadratic form

$$c^{\mathrm{T}}\Psi c = \sum_{i=1}^{N}\sum_{j=1}^{N} c_i c_j \psi(g(Y_i, Y_j)) \tag{9}$$

is positive on $\mathbb{R}^N \setminus \{0\}$ for any finite set $Y = \{Y_k\}_{k=1}^N$ of distinct points on S^{d-1}.

Definition 2. ([7]) Let the span of the trends p_k, $k = 1, \ldots, M$, be the space $\pi_t(\mathbb{R}^d)$ of polynomials in d variables of total degree t, where t is a nonnegative integer. A continuous function $\psi\colon [0,\pi] \to \mathbb{R}$ is said to be *conditionally positive definite of order t on S^{d-1}* (i.e., $\psi \in \mathrm{CPD}_t(S^{d-1})$) if the quadratic form (9) is positive for any finite set $Y = \{Y_k\}_{k=1}^N$ of distinct points on S^{d-1} and scalars $c_1, c_2, \ldots, c_N$ such that

$$\sum_{j=1}^{N} c_j p(Y_j) = 0$$

for all $p \in \pi_t(\mathbb{R}^d)$.

Remark 1. In Theorem 1, we use the hypothesis that the matrix Ψ is positive definite and rank $P = M$. Moreover, any function $\psi \in \mathrm{PD}(S^{d-1})$ can be used to provide a unique interpolant of the form

$$v(x) = \sum_{j=1}^{N} a_j \psi(g(x, X_j)).$$

However, most books and papers (see, e.g., [1], [6], [7]) employ the condition that the spherical basis function ψ is positive definite or conditionally positive definite to prove that the matrix Q is nonsingular (cf., e.g., [6]).

Remark 2. The simplest spherical interpolant v can be considered if we omit the second sum in (2), i.e., we set $M = 0$ and employ no trends. If Ψ is symmetric positive definite, there is no matrix P in the formulation, $Q = \Psi$ and, instead of (7), we get the $N \times N$ symmetric positive definite system

$$\Psi a = f. \tag{10}$$

Apparently, the system (10) possesses the unique solution a.

Remark 3. Let us formulate the least squares smoothing problem. Keep the notation introduced. Further, let w_j, $j = 1, \ldots, N$, be positive weights chosen and put $W = \mathrm{diag}(w_1, w_2, \ldots, w_N)$. In solving the data smoothing problem we employ the least squares functional minimization. The approximant is assumed in the form

$$\widehat{v}(x) = \sum_{j=1}^{N} (\widehat{f}_j - \widehat{a}_j) w_j \psi(g(x, X_j)) + \sum_{k=1}^{M} \widehat{b}_k p_k(x), \tag{11}$$

where $\widehat{a}_j$, $j = 1, \ldots, N$, and $\widehat{b}_k$, $k = 1, \ldots, M$, are real coefficients to be found, and, moreover, we have

$$\widehat{v}(X_j) = \widehat{a}_j, \quad j = 1, \ldots, N.$$

If $M = 0$, the second sum in (11) is empty.

Now the system of linear algebraic equations to be solved for the unknown vectors $\widehat{a}$ and $\widehat{b}$ is

$$\begin{bmatrix} \Psi W + I & -P \\ P^{\mathrm{T}} W & 0 \end{bmatrix} \begin{bmatrix} \widehat{a} \\ \widehat{b} \end{bmatrix} = \begin{bmatrix} \Psi W f \\ P^{\mathrm{T}} W f \end{bmatrix}. \tag{12}$$

No interpolation conditions are imposed. An analog of Theorem 1 concerned with the system (12) is proved e.g. in [10], Theorem 2.

3. Magnetic susceptibility measurement

As we have mentioned in the introduction, the particular physical quantity whose measured values are approximated by the means presented in this paper is magnetic susceptibility. Put $d = 3$, then S^2 is the usual two-dimensional unit sphere in the three-dimensional space. Choose a fixed positive integer N and put $M = 1$. Consider the interpolation formula (2) in the form

$$v(x) = \sum_{j=1}^{N} a_j \psi(g(x, X_j)) + bs(x), \tag{13}$$

where $x, X_j \in S^2$, i.e., in (7), P is a single column N-vector and b and 0 are scalars.

The interpolation conditions (4) now read

$$v(X_i) = \sum_{j=1}^{N} a_j \psi(g(X_i, X_j)) + bs(X_i), \quad i = 1, \ldots, N. \tag{14}$$

Moreover, we add a single constraint

$$\sum_{j=1}^{N} a_j s(X_j) = 0 \tag{15}$$

corresponding to (6).

To define a SBF formula (13) uniquely, we have to choose a proper spherical basis function ψ, geodesic metric g, and trend s. For $x, y \in S^2$ (both x and y are unit vectors), one usually puts

$$g(x, y) = \sqrt{1 - (x^{\mathrm{T}} y)^2},$$

where the angle α, $0 \leq \alpha \leq \pi$, between the vectors x and y is given by

$$\cos \alpha = x^{\mathrm{T}} y. \tag{16}$$

For our purposes, we consider the angle between vectors of parallel directions to be zero regardless of their orientation. At the same time, we take into account always the acute angle α of the vectors x, y, i.e., the range of α is $[0, \frac{1}{2}\pi]$. We now change the formula (16) for

$$\cos \alpha = |x^{\mathrm{T}} y|, \quad \text{i.e. } \alpha = \cos^{-1}(|x^{\mathrm{T}} y|),$$

and use the geodesic metric

$$g(x, y) = \sqrt{1 - \cos^2 \alpha} = \sin \alpha = \sin(\cos^{-1}(|x^{\mathrm{T}} y|)) \tag{17}$$

with α acute. Thus, this geodesic metric is the function $g\colon S^2 \times S^2 \to [0, \frac{1}{2}\pi]$.

The metric (17) does not distinguish the vectors x and $-x$. Therefore, in what follows, we assume that the elements X_j of the set X are mutually distinct and, moreover, that it is $X_i \neq -X_j$ for every $i, j = 1, \ldots, N$.

We have chosen the *inverse multiquadric*

$$\psi(r) = \frac{1}{\sqrt{(r^2 + c^2)}} \tag{18}$$

for the spherical radial basis function, where $r \in [0, \frac{1}{2}\pi]$ (the range of the function g) and c is a positive shape parameter that controls tension of the interpolation surface.

Finally, we take the second degree polynomial (1),

$$s(z) = K_1 z_1^2 + K_2 z_2^2 + K_3 z_3^2, \quad z = (z_1, z_2, z_3) \in S^2, \tag{19}$$

where K_1, K_2, K_3 are proper positive constants for the trend.

Notice that the argument of the SBF function ψ is from the interval $[0, \frac{1}{2}\pi]$ since $g\colon S^2 \times S^2 \to [0, \frac{1}{2}\pi]$ is a geodesic metric, while the argument of the trend s is from S^2, similarly to [7].

The advantage of the formula proposed is apparent in cases when we know that the physical field measured does not principally differ from the ideal field whose values can be computed from some explicit formula. This description of an ideal field is then fitted by the trend part of the formula and the contributions obtained from the first, spherical part of the formula are only small.

Let us verify the existence of the formula (13). Since we use the inverse multiquadric (18) for the spherical radial basis function, we shall employ some results of [7] and [6]. Now we can prove that the matrix Ψ corresponding to (18) is symmetric positive definite.

Lemma 3. ([7], p. 19) *The $N \times N$ symmetric matrix Ψ with entries*

$$\psi_{ij} = (r_{ij}^2 + c^2)^{-\alpha},$$

where $r_{ij} = g(X_i, X_j)$, $c > 0$, and $\alpha > 0$, is symmetric positive definite.

Consider the interpolation formula (13) with the functions g, ψ, and s given by the formulae (17), (18), and (19), respectively. Choose positive constants c, K_1, K_2, K_3. For the interpolation formula (13), set up the system (14) corresponding to the interpolation conditions (3) and the equation (15) corresponding to the constraints (6).

Theorem 2. *Let the system (7) correspond to the formula (13). Let the block P in the block matrix Q given by (8) have rank 1. Then the interpolation problem (14), (15) has the unique solution, where the coefficients a_j, $j = 1, \ldots, N$, and b solve uniquely the linear algebraic system (7).*

Proof. According to Lemma 3, the principal submatrix Ψ of the block matrix Q of the system (7) is positive definite. On the assumption that rank $P = 1$, the matrix Q is nonsingular by Theorem 1 and the system (7) has the unique solution a_j, $j = 1, \ldots, N$, and b. □

Remark 4. P is a single column N-vector, $P^{\mathrm{T}} = (s(X_1), \ldots, s(X_N))$. The assumption of Theorem 2 that rank $P = 1$ is apparently fulfilled if at least one of the entries $s(X_k)$ is nonzero.

4. Numerical experiments. Conclusions

We present some numerical experience with the interpolation problem described in Section 3. According to Lemma 3, the matrix Ψ with entries (5) is symmetric positive definite and the matrix Q introduced in (8) is nonsingular when the matrix P has rank $P = 1$. But the use of the lemniscate s given by (19) does not prevent a very difficult solving the linear algebraic system (7).

We have chosen the interpolation nodes x_j on the south (lower) "hemisphere" roughly equally. The system (7) can be easily solved for $N = 15$ (i.e., 15 nodes, 16 equations), but for $N = 30$ and higher its solution computed in double precision is useless.

The condition number cond Q of the matrix Q of a linear algebraic system characterizes in some way the accuracy one can reach when solving the system: the higher the condition number, the more ill-conditioned system and the worse (less accurate) the solution. For a symmetric matrix Q, the condition number cond Q can be defined as the quotient of the largest and smallest singular value of Q, i.e. the quotient of the largest in magnitude and smallest in magnitude eigenvalue of the matrix Q, cf. [4].

In our computation with $c \in [0.125, 2.000]$, cond Q reaches about 10^3 in case of $N = 15$, but about 10^8 in case of $N = 60$, which thus provides no acceptable solution. Decreasing c, we can reach a lower condition number.

We have shown sufficient conditions for the existence of SRBF interpolant and approximant. We have considered a particular SRBF interpolation formula employing an inverse multiquadric and using a trend being a second degree polynomial (19) in Section 3.

We have carried out numerical tests with this interpolation formula. The formula performs efficiently only for a small number N of interpolation nodes X_j and the results exhibit week dependency on the parameter c. Further research shall provide a comparison of results obtained using various other SRBFs, e.g. direct multiquadrics [6], thin plate splines [2], etc.

Acknowledgments

The author expresses his sincere gratitude to Professor Josef Ježek from the Faculty of Science of Charles University in Prague for introducing him to approximation problems on sphere and ways of solving these problems.

This work has been supported by the Institute of Mathematics of the Czech Academy of Sciences project RVO 67985840.

References

[1] Buhmann, M. D.: *Radial basis functions.* Cambridge University Press, Cambridge, 2003.

[2] Duchon, J.: Splines minimizing rotation-invariant semi-norms in Sobolev spaces. In: *Constructive theory of functions of several variables, Lecture Notes in Mathematics*, vol. 571, pp. 85–100. Springer, Berlin–Heidelberg, 1977.

[3] Fiedler, M.: *Special matrices and their applications in numerical mathematics.* Martinus Nijhoff Publishers, Dordrecht, 1986.

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

[5] Hrouda, F., Ježek, J., and Chadima, M.: On the origin of apparently negative minimum susceptibility of hematite single crystals calculated from low-field anisotropy of magnetic susceptibility. Geophys. J. Int. **224** (2021), 1905–1917.

[6] Hubbert, S., Lê Gia, Q. T., and Morton, T. M.: *Spherical radial basis functions, theory and applications.* Springer, Cham, 2015.

[7] Micchelli, C. A.: Interpolation of scattered data: distance matrices and conditionally positive definite functions. Constr. Approx. **2** (1986), 11–22.

[8] Nagata, T.: *Rock magnetism.* Maruzen, 1961.

[9] Segeth, K.: Some computational aspects of smooth approximation. Computing **95** (Suppl. 1) (2013), 695–708.

[10] Segeth, K.: Multivariate data fitting using polyharmonic splines. J. Comput. Appl. Math. **397** (2021), 113651.

[11] Tarling, D. H. and Hrouda, F.: *The magnetic anisotropy of rocks.* Chapman and Hall, London, 1993.

Programs and Algorithms of Numerical Mathematics 21
J. Chleboun, P. Kůs, J. Papež, M. Rozložník, K. Segeth, J. Šístek (Eds.)
Institute of Mathematics CAS, Prague 2023

ESTIMATION OF EDZ ZONES IN GREAT DEPTHS BY ELASTIC-PLASTIC MODELS

Stanislav Sysala

Institute of Geonics of the Czech Academy of Sciences
Studentská 1768, 708 00 Ostrava, Czech Republic
stanislav.sysala@ugn.cas.cz

Abstract: This contribution is devoted to modeling damage zones caused by the excavation of tunnels and boreholes (EDZ zones) in connection with the issue of deep storage of spent nuclear fuel in crystalline rocks. In particular, elastic-plastic models with Mohr-Coulomb or Hoek-Brown yield criteria are considered. Selected details of the numerical solution to the corresponding problems are mentioned. Possibilities of elastic and elastic-plastic approaches are illustrated by a numerical example.

Keywords: tunnel stability, EDZ zones, elasto-plasticity, finite element method

MSC: 74C05, 65N30

1. Introduction

Zones with higher stress concentrations, formation and joining of cracks of various sizes or with V-shaped notches can be observed on tunnel walls and in their vicinity as a consequence of excavation and other effects. Such zones are usually called as excavation damage zones (EDZ). Prediction of EDZ is important for safety assessment in many applications. Our particular motivation is related to deep storage of spent nuclear fuel in crystalline rocks where EDZ can simplify transport of radionuclides. In order to predict EDZ and analyze coupled processes in these zones, various in-situ experiments have been carried out in underground research laboratories around the world. For example, we mention the Äspö pillar stability experiment carried out in Sweden [1] or the Tunnel Sealing Experiment (TSX) in Canada [8].

The most important factor that causes the formation of EDZ is the initial stress state in the rock mass. EDZ may depend on its magnitude, the ratios between the principal stresses and on the orientation of the principal stress directions with respect to the tunnel. EDZ also depends on the shape of the tunnel and its dimensions, the method of excavation, mechanical properties of the rock mass or its geological structure. EDZ can also expand after the excavation due to surrounding sites or

DOI: 10.21136/panm.2022.21

thermal heating [1]. On the contrary, bentonite barriers of a deep repository [1] can contribute to the stabilization of EDZ zones.

Mathematical modeling of EDZ can be based on continuum mechanics, fracture mechanics or on multiscale approaches. In this contribution, we focus on the continuum models, namely on elastic and elastic-plastic models. The elastic models are usually combined with a failure criterion to detect zones with high stress concentrations. Such a treatment is the simplest one and is convenient for large-scale 3D geometries. Next, one can consider elastic-plastic models where the failure criterion is directly a part of the model and admissible stress fields must satisfy the criterion. These models can be enriched with internal variables representing softening/hardening variable or damage variable. In the article [7] and related papers, different types of damage zones were classified based on elastic-plastic models, the so-called DISL approach.

This contribution consists of the following parts. Section 2 contains selected details to an abstract elastic-perfectly plastic problem and its solution scheme. Section 3 is devoted to the Mohr-Coulomb and Hoek-Brown constitutive models and their solution. Section 4 contains a numerical example illustrating possibilities of elastic and elastic-plastic approaches of modeling EDZ zones. Concluding remarks can be found in Section 5.

2. Numerical scheme of the elastic-perfectly plastic model

We consider a simplified 2D geometry of the rock mass around the tunnel depicted in Figure 1. The square domain and its subdomain without the tunnel will be denoted as $\widehat{\Omega}$ and Ω, respectively. We prescribe zero normal displacements on the outer boundary $\partial\widehat{\Omega}$ (far the from tunnel) and zero normal stress on the inner boundary Γ, that is, $\boldsymbol{u}\cdot\boldsymbol{n} = 0$ on $\partial\widehat{\Omega}$ and $\boldsymbol{\sigma}\boldsymbol{n} = \boldsymbol{0}$ on Γ, where $\boldsymbol{u}$, $\boldsymbol{\sigma}$, and $\boldsymbol{n}$ denote the displacement field, the Cauchy stress field, and the outward unit normal vector to Ω, respectively. We prescribe the initial stress field $\boldsymbol{\sigma}_0$ defined in Ω. For the sake of simplicity, we simulate the tunnel excavation by the load history $t\boldsymbol{\sigma}_0/t_{\max}$, where $t \in [0, t_{\max}]$. Next ingredients of the elastic and elastic-plastic models are the infinitesimally small

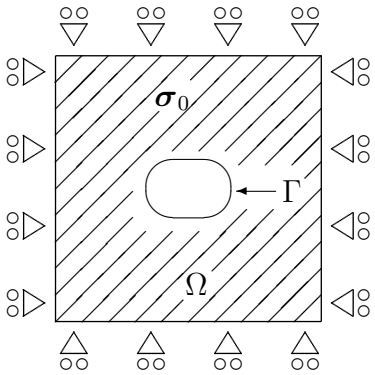


Figure 1: 2D geometry of the rock mass around the tunnel.

strain tensor

$$\boldsymbol{\varepsilon} := \boldsymbol{\varepsilon}(\boldsymbol{u}) = \frac{1}{2}(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T)$$

and the fourth-order elastic tensor $\mathbb{C}$,

$$\begin{aligned} \mathbb{C}\boldsymbol{\varepsilon} &= \frac{E}{1+\nu}\left\{\frac{\nu}{1-2\nu}(\text{tr } \boldsymbol{\varepsilon})\boldsymbol{I} + \boldsymbol{\varepsilon}\right\}, \\ \mathbb{C}^{-1}\boldsymbol{\sigma} &= -\frac{\nu}{E}(\text{tr } \boldsymbol{\sigma})\boldsymbol{I} + \frac{1+\nu}{E}\boldsymbol{\sigma}, \end{aligned}$$

where $E > 0$, $\nu \in (0, 1/2)$ denote Young's modulus and Poisson's ratio, respectively, $\boldsymbol{I}$ is the unit second-order tensor and $\text{tr } \boldsymbol{\varepsilon} = \boldsymbol{\varepsilon} : \boldsymbol{I} = \varepsilon_{11} + \varepsilon_{22} + \varepsilon_{33}$ is the trace of $\boldsymbol{\varepsilon}$.

In case of linear elasticity, we have the following constitutive (Hook's) law between the stress and strain tensors:

$$\boldsymbol{\sigma} = \mathbb{C}\boldsymbol{\varepsilon} + \boldsymbol{\sigma}_0 \quad \text{or} \quad \boldsymbol{\sigma} = \mathbb{C}[\boldsymbol{\varepsilon} + \boldsymbol{\varepsilon}_0], \;\; \boldsymbol{\varepsilon}_0 = \mathbb{C}^{-1}\boldsymbol{\sigma}_0.$$

Now, we introduce the elastic-perfectly plastic constitutive model, which is time-dependent. Let $\boldsymbol{\varepsilon}^e$ and $\boldsymbol{\varepsilon}^p$ denote the elastic and plastic parts of the strain tensor and λ is the plastic multiplier. We also define yield function $f := f(\boldsymbol{\sigma})$ and plastic potential $g := g(\boldsymbol{\sigma})$ and assume that these functions are convex. Then the corresponding evolution problem reads: for any $t \in (0, t_{\max})$, find $\boldsymbol{\sigma} := \boldsymbol{\sigma}(t)$, $\boldsymbol{\varepsilon} := \boldsymbol{\varepsilon}(t)$, $\boldsymbol{\varepsilon}^e := \boldsymbol{\varepsilon}^e(t)$, $\boldsymbol{\varepsilon}^p := \boldsymbol{\varepsilon}^p(t)$, $\lambda := \lambda(t)$ such that

- $\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^e + \boldsymbol{\varepsilon}^p$, $\boldsymbol{\sigma} = \mathbb{C}(\boldsymbol{\varepsilon}^e + t\boldsymbol{\varepsilon}_0/t_{\max})$,
- $\dot{\boldsymbol{\varepsilon}}^p \in \dot{\lambda}\partial g(\boldsymbol{\sigma})$, $\boldsymbol{\varepsilon}^p(0) = 0$,
- $\dot{\lambda} \geq 0$, $\dot{\lambda} f(\boldsymbol{\sigma}) = 0$, $f(\boldsymbol{\sigma}) \leq 0$.

Here, the dot symbol means the time derivative and $\partial g(\boldsymbol{\sigma})$ denotes the subdifferential of g at $\boldsymbol{\sigma}$. It is worth-noticing that subdifferentials are not so obvious in engineering practice and the plastic flow rule is usually written by the derivative of g:

$$\dot{\boldsymbol{\varepsilon}}^p = \dot{\lambda}\frac{\partial g(\boldsymbol{\sigma})}{\partial \boldsymbol{\sigma}},$$

despite the fact that g is often non-differentiable. One of the aim of our work is to show that the knowledge of an explicit form of the set $\partial g(\boldsymbol{\sigma})$ can simplify analysis and constitutive solution for various elastic-plastic models. This was shown in [9, 10].

The elastic-plastic constitutive problem is mostly discretized by the implicit Euler method. Consider the partition $0 = t_0 < t_1 < \ldots < t_{k-1} < t_k < \ldots < t_{\max}$ of the time interval. Then the discretized constitutive problem at the k-th step, $k = 1, 2, \ldots$, has the following scheme: given $\boldsymbol{\varepsilon}_0$, $\boldsymbol{\varepsilon}_k$, and $\boldsymbol{\varepsilon}^p_{k-1}$, find $\boldsymbol{\sigma}_k$ and $\boldsymbol{\varepsilon}^p_k$ such that

$$\boldsymbol{\sigma}_k = \boldsymbol{T}(\boldsymbol{\varepsilon}_k - \boldsymbol{\varepsilon}^p_{k-1} + t_k\boldsymbol{\varepsilon}_0/t_{\max}), \quad \boldsymbol{\varepsilon}^p_k = \boldsymbol{\varepsilon}_k + t_k\boldsymbol{\varepsilon}_0/t_{\max} - \mathbb{C}^{-1}\boldsymbol{\sigma}_k.$$

Here, the tensor-valued function $\boldsymbol{T}$ needs to be constructed. In general, one can find $\boldsymbol{T}$ only in an implicit form and construct it by an iterative procedure. This construction is based on the elastic prediction – plastic correction algorithm [4, 3, 9, 10]. Within the elastic prediction, we test whether the trial stress $\boldsymbol{\sigma}_k^{tr} = \mathbb{C}(\boldsymbol{\varepsilon}_k - \boldsymbol{\varepsilon}_{k-1}^p + t_k\boldsymbol{\varepsilon}_0/t_{\max})$ satisfies the failure criterion $f(\boldsymbol{\sigma}_k^{tr}) \leq 0$. If it is so then $\boldsymbol{\sigma}_k = \boldsymbol{\sigma}_k^{tr}$ and $\boldsymbol{\varepsilon}_k^p = \boldsymbol{\varepsilon}_{k-1}^p$. Otherwise, the plastic correction is applied to be the constraint $f(\boldsymbol{\sigma}_k) = 0$ satisfied. So we need to return the predicted stress to the failure surface and construct the so-called return mapping. Such a mapping can be interpreted as a generalized projection onto a convex set. It is also worth noticing that the function $\boldsymbol{T}$ is non-differentiable, but its semismoothness is expected and can be proven just by the subdifferential-based treatment [9, 10].

Using the function $\boldsymbol{T}$, the overall elastic-plastic problem in terms of displacements reads:

$$\text{find } \boldsymbol{u}_k \in V: \quad \int_\Omega \boldsymbol{T}(\varepsilon(\boldsymbol{u}_k) - \boldsymbol{\varepsilon}_{k-1}^p + t_k\boldsymbol{\varepsilon}_0/t_{\max}) : \boldsymbol{\varepsilon}(\boldsymbol{v})\,\mathrm{d}x = 0 \quad \forall \boldsymbol{v} \in V,$$

where

$$V = \{\boldsymbol{v} \in H^1(\Omega;\mathbb{R}^d) \mid \boldsymbol{v}\cdot\boldsymbol{n} = 0 \text{ on } \partial\widehat{\Omega}\}$$

is a space of admissible displacement fields. After a space discretization, we arrive at a system of non-linear equations. Such a system is usually solved by a non-smooth version of the Newton method. It requires to construct a generalized derivative of $\boldsymbol{T}$. Its construction for specific models will be briefly discussed in the next section.

3. The Mohr-Coulomb and Hoek-Brown constitutive models

The Mohr-Coulomb and Hoek-Brown constitutive models are usual in geotechnics. The functions f and g for these models are defined in terms of principal stresses. Therefore, we need to introduce the spectral decomposition of the Cauchy stress tensor:

$$\boldsymbol{\sigma} = \sum_{i=1}^{3} \sigma_i \mathbf{e}_i \otimes \mathbf{e}_i, \quad \sigma_1 \geq \sigma_2 \geq \sigma_3.$$

Here, $\sigma_i \in \mathbb{R}$, $\mathbf{e}_i \in \mathbb{R}^3$, $i = 1, 2, 3$, denote the eigenvalues (principle stresses), and the eigenvectors of $\boldsymbol{\sigma}$, respectively. We assume the ordering $\sigma_1 \geq \sigma_2 \geq \sigma_3$ of the eigenvalues of $\boldsymbol{\sigma}$. From now on, we shall work with a mechanical sign convention assuming positive values for a tension. (In geomechanics, opposite sign convention is usual.)

The Mohr-Coulomb model is defined by the functions

$$\begin{aligned} f(\boldsymbol{\sigma}) &= (1+\sin\phi)\sigma_1 - (1-\sin\phi)\sigma_3 - 2c\cos\phi, \\ g(\boldsymbol{\sigma}) &= (1+\sin\psi)\sigma_1 - (1-\sin\psi)\sigma_3, \end{aligned}$$

where $c > 0$, $\phi \in (0, \pi/2)$ and $\psi \in (0, \pi/2)$ are given material parameters denoting the cohesion, the friction angle and the dilatancy angle. It is expected that $\psi \leq \phi$.

The Hoek-Brown model is defined by the functions

$$\begin{aligned} f(\boldsymbol{\sigma}) &= \sigma_1 - \sigma_{ci}\Big(s - m_b\frac{\sigma_1}{\sigma_{ci}}\Big)^a - \sigma_3, \\ g(\boldsymbol{\sigma}) &= \sigma_1 - \sigma_{ci}\Big(s_g - m_g\frac{\sigma_1}{\sigma_{ci}}\Big)^{a_g} - \sigma_3, \end{aligned}$$

where $\sigma_{ci}, s, s_g, m_b, m_g > 0$ and $a, a_g \in (0,1)$ are given material parameters. More details to these parameters can be found in [3, 5, 6]. Briefly speaking, they are defined by empirical formulas containing usual material parameters for intact rock samples and two indices, the geological strength index GSI and the disturbance index D. GSI represents a structure of the surrounding rock mass and D characterizes a way of the excavation. To be the model well-defined, we assume that $s_g/m_g \geq s/m$, although we have not found such an assumption in literature. In the limit case $a = a_g = 1$, one can transform the Hoek-Brown model to the Mohr-Coulomb models one.

Admissible stress fields satisfy the condition $f(\boldsymbol{\sigma}) \leq 0$. For both the models, the corresponding set is convex and aligned with the hydrostatic axis (where $\sigma_1 = \sigma_2 = \sigma_3$). The Mohr-Coulomb set is a hexahedral pyramid in the space of the principle stresses with the apex at $\sigma_t = c/\tan\phi$. For the Hoek-Brown model, the pyramid is curved and has the apex at $\sigma_t = s\sigma_{ci}/m_b$, see [3]. Next, one can see that the function g has the following structure for both the models:

$$g(\boldsymbol{\sigma}) = \hat{g}_1(\sigma_1) - \hat{g}_3(\sigma_3),$$

where $\hat{g}_1$ and $\hat{g}_3$ are increasing, convex and twice differentiable functions. By extending the results from [10], it is possible to show that such functions g are convex and they subdifferentials satisfy

$$\begin{aligned} \partial g(\boldsymbol{\sigma}) = \Big\{ \boldsymbol{\nu} = \sum_{i=1}^{3} \nu_i \mathbf{e}_i \otimes \mathbf{e}_i \;\Big|\; & (\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3) \in V(\boldsymbol{\sigma}); \\ & \hat{g}_1'(\sigma_1) \geq \nu_1 \geq \nu_2 \geq \nu_3 \geq -\hat{g}_3'(\sigma_3);\ \sum_{i=1}^{3} \nu_i = \hat{g}_1'(\sigma_1) - \hat{g}_3'(\sigma_3); \\ & (\nu_1 - \hat{g}_1'(\sigma_1))(\sigma_1 - \sigma_2) = 0;\ (\nu_3 + \hat{g}_3'(\sigma_3))(\sigma_2 - \sigma_3) = 0 \Big\}, \end{aligned}$$

where $\hat{g}_1'$, $\hat{g}_3'$ denote the derivatives of $\hat{g}_1$, $\hat{g}_3$ and

$$V(\boldsymbol{\sigma}) = \{(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3) \in [\mathbb{R}^3]^3 \mid \mathbf{e}_i \cdot \mathbf{e}_j = \delta_{ij};\ \boldsymbol{\sigma}\mathbf{e}_i = \sigma_i \mathbf{e}_i,\ i,j = 1,2,3;\ \sigma_1 \geq \sigma_2 \geq \sigma_3\}.$$

If $\sigma_1 > \sigma_2 > \sigma_3$ then $\nu_1 = \hat{g}_1'(\sigma_1)$, $\nu_2 = 0$, and $\nu_3 = -\hat{g}_3'(\sigma_3)$, and thus g is differentiable at $\boldsymbol{\sigma}$. Otherwise, g is not differentiable at $\boldsymbol{\sigma}$ and ν_1, ν_2, ν_3 are not uniquely defined.

Let us recall that the unknown stresses tensor $\boldsymbol{\sigma} := \boldsymbol{\sigma}_k$ satisfies $f(\boldsymbol{\sigma}) = 0$ if the plastic correction (the return mapping) occurs. In such a case, $\boldsymbol{\sigma}$ lies on the

surface of the Mohr-Coulomb or Hoek-Brown pyramid. With respect to the ordering $\sigma_1 \geq \sigma_2 \geq \sigma_3$, we split the pyramidal surface into four parts: smooth portion ($\sigma_1 > \sigma_2 > \sigma_3$), the left (curved) edge ($\sigma_1 = \sigma_2 > \sigma_3$), the right (curved) edge ($\sigma_1 > \sigma_2 = \sigma_3$), and the apex ($\sigma_1 = \sigma_2 = \sigma_3 = \sigma_t$). This terminology was introduced in [4]. For each of these cases, one can specify the set $\partial g(\boldsymbol{\sigma})$ and consequently, the form of the return mapping. For example, if the return to the left edge occurs then $\hat{g}_1'(\sigma_1) \geq \nu_1 \geq \nu_2 \geq 0$, $\nu_1 + \nu_2 = \hat{g}_1'(\sigma_1)$, $\nu_3 = -\hat{g}_3'(\sigma_3)$ hold. These conditions are not usual in engineering practice but they can simplify the construction of the return mapping and help to find a correct return type.

In case of the elastic-perfectly plastic Mohr-Coulomb model, one can find decision criteria for each return type and even derive a close form of the constitutive operator $\boldsymbol{T}$, see e.g. [4, 10]. However, the function $\boldsymbol{T}$ is only in an implicit form for the Hoek-Brown model. In [3], the following return-mapping scheme was proposed. First, the return to the apex is tested. In this case, the solution must satisfy $\sigma_1 = \sigma_2 = \sigma_3 = \sigma_t$ and it is possible to derive necessary and sufficient conditions for this return type. If the return to the apex does not occur, the return to the smooth portion of the yield surface is tested and the corresponding problem has to be solved iteratively. After finding a solution candidate, we decide about its admissibility. If it is not admissible, one can decide using this candidate whether the return to the left or right curved edge occurs. We plan in our future work to complete this solution concept by rigorous analysis based on the subdifferential-based treatment and show that the operator $\boldsymbol{T}$ is well-defined.

In order to construct a generalized derivative of $\boldsymbol{T}$, we use the so-called eigenprojections and their derivatives, see [4]. For the sake of brevity, we introduce it only for a tensor $\boldsymbol{\varepsilon}^{tr}$ with three different eigenvalues $\varepsilon_1^{tr} > \varepsilon_2^{tr} > \varepsilon_3^{tr}$. Then, the spectral decomposition of $\boldsymbol{\varepsilon}^{tr}$ satisfies

$$\boldsymbol{\varepsilon}^{tr} = \sum_{i=1}^{3} \varepsilon_i^{tr} \mathbf{e}_i^{tr} \otimes \mathbf{e}_i^{tr}, \quad \mathbf{e}_i^{tr} \otimes \mathbf{e}_i^{tr} = \boldsymbol{E}_i^{tr} = \frac{(\boldsymbol{\varepsilon}^{tr} - \varepsilon_j^{tr}\boldsymbol{I})(\boldsymbol{\varepsilon}^{tr} - \varepsilon_k^{tr}\boldsymbol{I})}{(\varepsilon_i^{tr} - \varepsilon_j^{tr})(\varepsilon_i^{tr} - \varepsilon_k^{tr})}, \quad i = 1, 2, 3.$$

We say that the second-order tensors $\boldsymbol{E}_1^{tr}$, $\boldsymbol{E}_2^{tr}$, and $\boldsymbol{E}_3^{tr}$ are the *eigenprojections* of $\boldsymbol{\varepsilon}^{tr}$. If we consider the eigenvalues as functions depending on $\boldsymbol{\varepsilon}^{tr}$, then their derivatives satisfy $D\varepsilon_i^{tr}(\boldsymbol{\varepsilon}^{tr}) = \boldsymbol{E}_i^{tr}$, $i = 1, 2, 3$. Next, the derivative of $\boldsymbol{E}_i^{tr}$ is the fourth-order tensor and can be found in the following form:

$$\begin{aligned} D\boldsymbol{E}_i^{tr}(\boldsymbol{\varepsilon}^{tr}) := \mathbb{E}_i^{tr} \quad = \quad & \frac{\mathrm{D}((\boldsymbol{\varepsilon}^{tr})^2) - (\varepsilon_j^{tr} + \varepsilon_k^{tr})\mathbb{I} - (2\varepsilon_i^{tr} - \varepsilon_j^{tr} - \varepsilon_k^{tr})\boldsymbol{E}_i^{tr} \otimes \boldsymbol{E}_i^{tr}}{(\varepsilon_i^{tr} - \varepsilon_j^{tr})(\varepsilon_i^{tr} - \varepsilon_k^{tr})} \\ & - \frac{(\varepsilon_j^{tr} - \varepsilon_k^{tr})[\boldsymbol{E}_j^{tr} \otimes \boldsymbol{E}_j^{tr} - \boldsymbol{E}_k^{tr} \otimes \boldsymbol{E}_k^{tr}]}{(\varepsilon_i^{tr} - \varepsilon_j^{tr})(\varepsilon_i^{tr} - \varepsilon_k^{tr})}, \end{aligned}$$

where $i \neq j \neq k \neq i$.

Setting $\boldsymbol{\varepsilon}^{tr} := \boldsymbol{\varepsilon}_k - \boldsymbol{\varepsilon}^p_{k-1} + t_k\boldsymbol{\varepsilon}_0/t_{\max}$ and assuming that $\varepsilon^{tr}_1 > \varepsilon^{tr}_2 > \varepsilon^{tr}_3$, one can specify the form of the constitutive function $\boldsymbol{T}$ and its generalized derivative [10]:

$$\boldsymbol{\sigma}_k = \boldsymbol{T}\left(\boldsymbol{\varepsilon}^{tr}\right) = \sum_{i=1}^{3} \sigma_i(\boldsymbol{\varepsilon}^{tr})\boldsymbol{E}^{tr}_i, \quad \mathrm{D}\boldsymbol{T}\left(\boldsymbol{\varepsilon}^{tr}\right) = \sum_{i=1}^{3} \left[\sigma_i(\boldsymbol{\varepsilon}^{tr})\mathbb{E}^{tr}_i + \boldsymbol{E}^{tr}_i \otimes \mathrm{D}\sigma_i(\boldsymbol{\varepsilon}^{tr})\right].$$

Here, $\sigma_1, \sigma_2, \sigma_3$ are eigenvalues of the unknown stress tensor $\boldsymbol{\sigma}_k$. They depends on $\boldsymbol{\varepsilon}^{tr}$. $\mathrm{D}\sigma_i$ denotes a generalized derivative of σ_i, $i = 1, 2, 3$. It is necessary to use the implicit function theorem to find these derivatives.

4. Numerical example

In this section, we compare the elastic and elastic-plastic approaches to the prediction of EDZ. The comparison is illustrated on a plane strain problem inspired by a case study of the TSX experiment performed in the depth about 500 meters in Underground Research Laboratory in Canada, see [8].

The geometry and the finite element mesh are depicted in Figure 2. In particular, it is considered an elliptic tunnel profile with the diameters 4.375 and 3.5 meters. The initial stress tensor $\boldsymbol{\sigma}_0$ is assumed to be constant in the whole domain and its non-zero components have the following sizes: $\sigma_{0,1} = -45$ MPa, $\sigma_{0,2} = -11$ MPa, and $\sigma_{0,3} = -60$ MPa. The largest principle stress $\sigma_{0,3}$ is aligned with the tunnel axis and it is included to the model through the Mohr-Coulomb plastic criterion. The remaining principle stresses are depicted in Figure 2. The excavation process took time 17 days. So we choose $t_{\max} = 17$ days and consider 17 time steps. Next, we set $E = 60$ GPa, $\nu = 0.2$, $c = 17$ MPa, $\phi = \psi = 26°$. The strength parameters c and ϕ were chosen much lower than in [8] in order to highlight the difference between the elastic and elastic-plastic approaches. We use P2 finite elements and 7-point quadrature on each triangular element. The problems were implemented within in-house codes in Matlab. Some of them are available for download, see [2], and their Python's counterparts can be downloaded from [11].

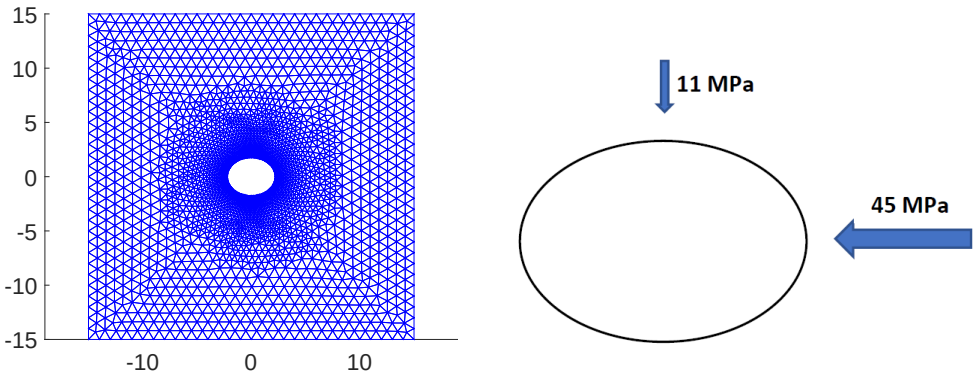


Figure 2: The geometry and the mesh for the plane strain problem. The sizes are in meters.

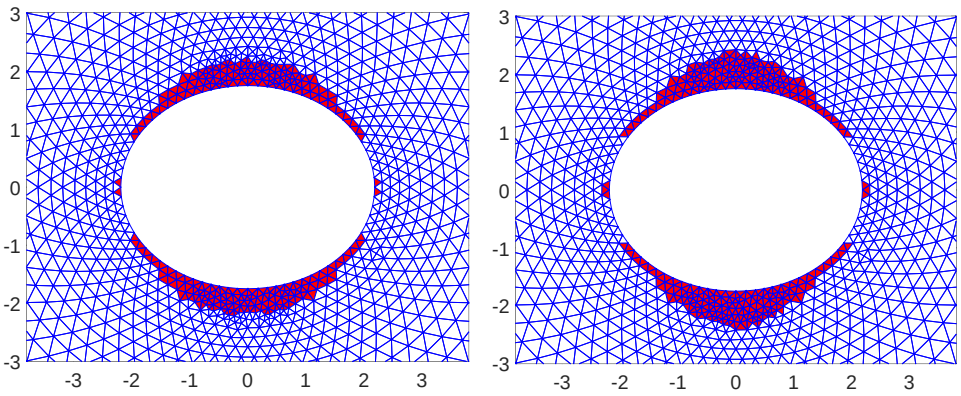


Figure 3: Comparison of failure zones for the elastic (left) and the elastic-plastic (right) models.

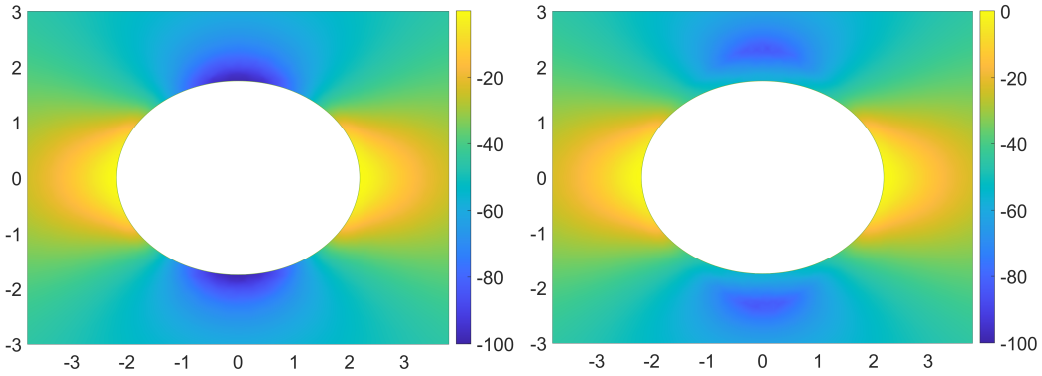


Figure 4: Comparison of horizontal stresses for the elastic (left) and the elastic-plastic (right) models. The scales are in MPa.

The comparison of failure zones computed for the elastic and elastic-plastic models are depicted in Figure 3. The zones for the elastic model are created by such elements where the Mohr-Coulomb criterion is not satisfied. They are rounded around the tunnel wall. In case of the elastic-plastic model, the zones represent elements with positive plastic multiplier. They have a typical V-notch shape that can be also observed within in-situ experiments.

In Figure 4, horizontal stresses are compared for the approaches. The elastic model admit higher stress concentrations (about 100 MPa) on the tunnel top and bottom unlike the elastic-plastic model where these concentrations are only about 50 MPa.

Figure 5 compares the total displacement and 300 times enlarged deformed shapes. For linear elasticity, we observe the contraction of the rock mass on the top and bottom of the tunnel. On the other hand, the dilatation is visible there in case of the elastic-plastic model. For better visualization of the contraction/dilatation, we compare vertical displacements on the tunnel top in Figure 6. We see that the plastic response is strongly nonlinear from the tenth time step leading to the dilatation.

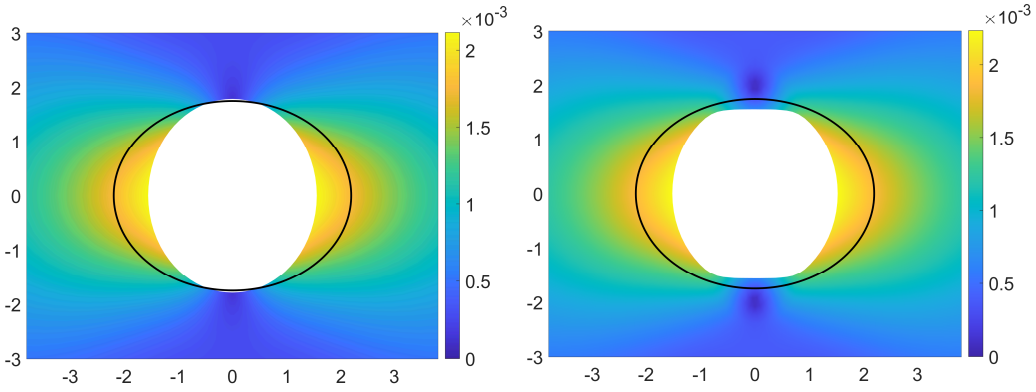


Figure 5: Comparison of total displacements and deformed shapes for the elastic (left) and the elastic-plastic (right) models. The scales are in meters.

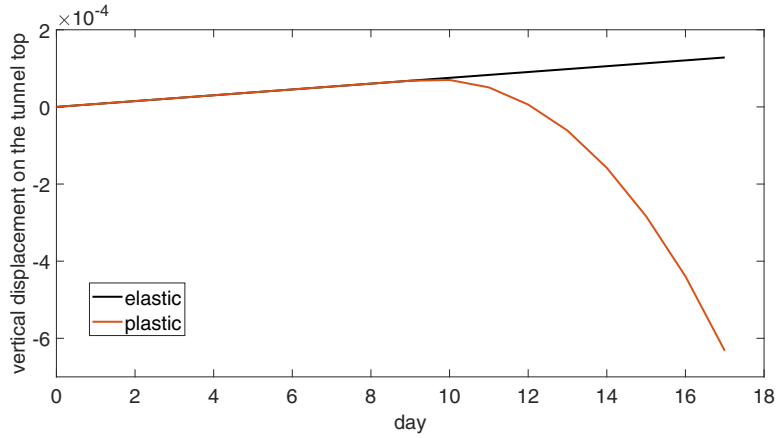


Figure 6: Evolution of the vertical displacements (in meters) on the tunnel top.

5. Conclusion

This contribution was a brief introduction to EDZ for deep tunnels in crystalline rocks. For prediction of EDZ, the elastic and elastic-plastic models were used. A scheme of numerical solution of the elastic-plastic problem was introduced. A particular interest was devoted to the Mohr-Coulomb and Hoek-Brown failure criteria. The subdifferential-based treatment to their constitutive solution was recommended. Finally, the elastic and elastic-plastic approaches to modeling of EDZ were compared on an illustrative numerical example.

Acknowledgements

This work was supported by grant No. TK02010118 of the Technology Agency of the Czech Republic.

References

[1] Andersson, J. C., Martin, C. D. and Stille, H.: The Äspö pillar stability experiment: part II—rock mass response to coupled excavation-induced and thermal-induced stresses. International Journal of Rock Mechanics and Mining Sciences **46**(5) (2009), 879–895.

[2] Čermák, M., Sysala, S. and Valdman, J.: Efficient and flexible MATLAB implementation of 2D and 3D elastoplastic problems. Applied Mathematics and Computation **355** (2019), 595–614.

[3] Clausen, J. and Damkilde, L.: An exact implementation of the Hoek–Brown criterion for elasto-plastic finite element calculations. International Journal of Rock Mechanics and Mining Sciences **45**(6) (2008), 831–847.

[4] de Souza Neto, E. A., Peric, D. and Owen, D. R.: *Computational methods for plasticity: theory and applications.* John Wiley & Sons, 2011.

[5] Hoek, E. and Brown, E. T.: The Hoek–Brown failure criterion and GSI–2018 edition. Journal of Rock Mechanics and Geotechnical Engineering **11**(3) (2019), 445–463.

[6] Hoek, E., Carranza-Torres, C. and Corkum, B.: Hoek-Brown failure criterion-2002 edition. In: *Proceedings of NARMS-Tac* **1**(1) (2002), 267–273.

[7] Perras, M. A. and Diederichs, M. S.: Predicting excavation damage zone depths in brittle rocks. Journal of Rock Mechanics and Geotechnical Engineering **8**(1) (2016), 60–74.

[8] Rutqvist, J., Börgesson, L., Chijimatsu, M., Hernelind, J., Jing, L., Kobayashi, A. and Nguyen, S.: Modeling of damage, permeability changes and pressure responses during excavation of the TSX tunnel in granitic rock at URL, Canada. Environmental Geology **57**(6) (2009), 1263–1274.

[9] Sysala, S., Cermak, M., Koudelka, T., Kruis, J., Zeman, J. and Blaheta, R.: Subdifferential-based implicit return-mapping operators in computational plasticity. ZAMM **96** (2016), 1318–1338.

[10] Sysala, S., Čermák, M. and Ligurský, T.: Subdifferential-based implicit return-mapping operators in Mohr-Coulomb plasticity. ZAMM **97** (2017), 1502–1523.

[11] https://github.com/MartinBeseda/FEM-ElastoPlasticity

Programs and Algorithms of Numerical Mathematics 21
J. Chleboun, P. Kůs, J. Papež, M. Rozložník, K. Segeth, J. Šístek (Eds.)
Institute of Mathematics CAS, Prague 2023

HOMOGENIZATION OF THE TRANSPORT EQUATION DESCRIBING CONVECTION-DIFFUSION PROCESSES IN A MATERIAL WITH FINE PERIODIC STRUCTURE

David Šilhánek, Michal Beneš

Department of Mathematics
Faculty of Civil Engineering, Czech Technical University in Prague
Thákurova 7, 166 29 Prague 6, Czech Republic
david.silhanek@fsv.cvut.cz, michal.benes@cvut.cz

Abstract: In the present contribution we discuss mathematical homogenization and numerical solution of the elliptic problem describing convection-diffusion processes in a material with fine periodic structure. Transport processes such as heat conduction or transport of contaminants through porous media are typically associated with convection-diffusion equations. It is well known that the application of the classical Galerkin finite element method is inappropriate in this case since the discrete solution is usually globally affected by spurious oscillations. Therefore, great care should be taken in developing stable numerical formulations. We describe a variational principle for the convection-diffusion problem with rapidly oscillating coefficients and formulate the corresponding homogenization results. Further, based on the variational principle, we derive a stable numerical scheme for the corresponding homogenized problem. A numerical example will be solved to illustrate the overall performance of the proposed method.

Keywords: variational principles, homogenization, Γ-convergence, convection-diffusion equation, optimal artificial diffusion

MSC: 35B27, 35B38, 70G75, 76R05

1. Introduction

Let Ω be a bounded domain in $\mathbb{R}^d$, $d = 1, 2, 3$. In particular, we assume that Ω is a domain with Lipschitz boundary $\partial\Omega$ (in case $d = 2, 3$). Further, Γ_D and Γ_N are open disjoint subsets of $\partial\Omega$ such that $\partial\Omega = \overline{\Gamma_D} \cup \overline{\Gamma_N}$, $\Gamma_D \neq \emptyset$. We use the standard function spaces $W^{1,2}(\Omega)$, $W^{1,\infty}(\Omega)$, $L^2(\Omega)$, $L^\infty(\Omega)$, $L^2(\Gamma_N)$. These function spaces we use are rather familiar and we omit the precise definitions, see e.g. [9] for details. Further, define the space V by $V := \{v \in W^{1,2}(\Omega);\ v = 0 \text{ on } \Gamma_D\}$ (more precisely,

DOI: 10.21136/panm.2022.22

$v = 0$ on Γ_D means that the trace of v is vanishing on Γ_D). We study the family of boundary-value problems

$$-\nabla \cdot \left(a\left(\frac{\boldsymbol{x}}{\varepsilon}\right) \nabla u^\varepsilon(\boldsymbol{x}) \right) + \boldsymbol{b}\left(\frac{\boldsymbol{x}}{\varepsilon}\right) \cdot \nabla u^\varepsilon(\boldsymbol{x}) = f(\boldsymbol{x}) \qquad \text{in } \Omega, \tag{1}$$

$$u^\varepsilon(\boldsymbol{x}) = 0 \qquad \text{on } \Gamma_D, \tag{2}$$

$$-\boldsymbol{n} \cdot a\left(\frac{\boldsymbol{x}}{\varepsilon}\right) \nabla u^\varepsilon(\boldsymbol{x}) = \alpha u^\varepsilon(\boldsymbol{x}) + \gamma_N(\boldsymbol{x}) \qquad \text{on } \Gamma_N. \tag{3}$$

Here, $\boldsymbol{n}$ denotes the unit exterior normal vector to the boundary $\partial\Omega$. We assume that the transport coefficients a and $\boldsymbol{b}$ periodically depend on a fine scale $\boldsymbol{x}/\varepsilon$ ($\varepsilon > 0$ being a small scalar parameter). We then let $\varepsilon \to 0_+$ and study the asymptotic behavior of the problem. In particular, our aim is to formulate a variational principle for (1)–(3). Note that, in general, (1) is not in a *divergence form*, however, under the assumptions below, there exists a functional $\mathcal{I}^\varepsilon$ on V whose *minimizers* are solutions of (1)–(3). Then the Γ-convergence of $\mathcal{I}^\varepsilon$ (as $\varepsilon \to 0_+$) is equivalent to the homogenization of (1)–(3).

The following assumptions will be needed throughout the paper.

- $\alpha > 0$ is a real positive parameter, fixed throughout the paper, $f \in L^2(\Omega)$ and $\gamma_N \in L^2(\Gamma_N)$.
- $a\colon \mathbb{R}^d \times \mathbb{R}$ is given by a strictly positive and bounded function, such that

$$0 < a_1 \le a(\boldsymbol{\xi}) \le a_2 < +\infty \quad \text{for all } \boldsymbol{\xi} \in \mathbb{R}^d \quad (a_1, a_2 = \text{const}).$$

- The coefficient functions are rapidly oscillating , i.e. of the form

$$a^\varepsilon(\boldsymbol{x}) := a\left(\frac{\boldsymbol{x}}{\varepsilon}\right),$$
$$\boldsymbol{b}^\varepsilon(\boldsymbol{x}) := \boldsymbol{b}\left(\frac{\boldsymbol{x}}{\varepsilon}\right)$$

  for all $\boldsymbol{x} \in \Omega$, where the functions a, b_1, $\dots b_d$ are Y-periodic in $\mathbb{R}^d$ with periodicity cell

$$Y = \{\boldsymbol{y} = (y_1, \dots, y_d : 0 < y_i < 1) \text{ for } i = 1, \dots, d\}$$

  and ε is a scale parameter.

- The coefficient functions are taken to be a gradient field in the sense that

$$-\nabla \varphi^\varepsilon(\boldsymbol{x}) = \frac{\boldsymbol{b}^\varepsilon(\boldsymbol{x})}{a^\varepsilon(\boldsymbol{x})}$$

  with potential φ^ε.

- The potential φ^ε is a Lipschitz function, $\varphi^\varepsilon \in W^{1,\infty}(\Omega)$ such that

$$\varphi^\varepsilon(\boldsymbol{x}) = \mathbf{R}_0 \cdot \boldsymbol{x} + \varepsilon\psi\left(\frac{\boldsymbol{x}}{\varepsilon}\right)$$

  with some $\mathbf{R}_0 \in \mathbb{R}^d$ and $\psi \in W^{1,\infty}(\Omega)$ Y-periodic in $\mathbb{R}^d$.

For smaller and smaller ε, the coefficients a^ε and $\boldsymbol{b}^\varepsilon$ oscillate more and more rapidly and it is natural to study the limit of u^ε in (1)–(3) as $\varepsilon \to 0$.

2. Standard weighted residual method and the Galerkin formulation

A weak formulation of the problem is to find $u \in V$ satisfying

$$\int_\Omega a^\varepsilon(\boldsymbol{x})\nabla u^\varepsilon \cdot \nabla v \,\mathrm{d}\Omega + \int_\Omega \boldsymbol{b}^\varepsilon(\boldsymbol{x}) \cdot \nabla u^\varepsilon v \,\mathrm{d}\Omega + \alpha \int_{\Gamma_N} u^\varepsilon v \,\mathrm{d}\sigma = \int_\Omega f v \,\mathrm{d}\Omega + \int_{\Gamma_N} \gamma_N v \,\mathrm{d}\sigma$$

for all $v \in V$. Here $\mathrm{d}\Omega$ denotes Lebesgue measure and $\mathrm{d}\sigma$ is the surface area measure on the boundary $\partial\Omega$. By $\mathcal{T}_h$ we denote an admissible partition of Ω with mesh size h with standard properties from the finite element theory (see e.g. [4]). Let $V_h \subset C(\overline{\Omega}) \cap V$ be the standard conforming linear finite element space over $\mathcal{T}_h$. A finite element formulation corresponding to the problem can be written as follows: find $u_h \in V_h$ satisfying

$$\int_\Omega a^\varepsilon(\boldsymbol{x})\nabla u_h^\varepsilon \cdot \nabla v_h \,\mathrm{d}\Omega + \int_\Omega \boldsymbol{b}^\varepsilon(\boldsymbol{x}) \cdot \nabla u_h^\varepsilon v_h \,\mathrm{d}\Omega + \alpha \int_{\Gamma_N} u_h^\varepsilon v_h \,\mathrm{d}\sigma = \int_\Omega f v_h \,\mathrm{d}\Omega + \int_{\Gamma_N} \gamma_N v_h \,\mathrm{d}\sigma \quad (4)$$

for all $v_h \in V_h$.

It is well-known that, as the convective term represents a nonsymmetric operator, the standard Galerkin finite element method loses the *best approximation property*. As a consequence, when the convective term is significant, the Galerkin formulation produces node-to-node spurious oscillations. One possible way is to choose a sufficiently fine grid such that the element Péclet number is less than one. However, this approach may not always be practical from the computational point of view. Therefore, several stabilized methods have been developed to avoid unphysical spurious oscillations on coarse grids, see e.g. [6] and the references given there. In particular, the authors in [11] have shown a variational basis for the *optimal artificial diffusion method*. Following this observation, we provide a variational principle for the problem (1)–(3) such that the solution u^ε minimizes a certain functional $\mathcal{I}^\varepsilon$ over the appropriate solution space V. Using the theory of Γ-convergence, we identify the limit $\mathcal{I}_{\mathrm{hom}}$ of $\mathcal{I}^\varepsilon$ as ε goes to 0, such that the minima of $\mathcal{I}^\varepsilon$ converge to the

minimum of the homogenized functional $\mathcal{I}_{\text{hom}}$. Based on the variational structure of this problem, i.e. from the fact that the *homogenized solution* minimizes a certain *homogenized functional*, the finite element solution possesses the *best approximation property*. Namely, the finite element approximation of the homogenized solution (the solution of the problem with the constant *homogenized* coefficients) gives nodally exact solutions for 1D problems with constant coefficients.

3. A variational principle for the advection-diffusion problems

Define $\chi^\varepsilon(\boldsymbol{x}) := \exp[\varphi^\varepsilon(\boldsymbol{x})]$. Then $\chi^\varepsilon \in W^{1,\infty}(\Omega)$ and $\chi^\varepsilon(\boldsymbol{x}) \geq c > 0$ on $\overline{\Omega}$. The function χ^ε will be called a *multiplier* for this variational problem. It is easily verified that sufficiently smooth function u^ε solves (1) provided

$$-\nabla \cdot (\chi^\varepsilon(\boldsymbol{x}) a^\varepsilon(\boldsymbol{x}) \nabla u^\varepsilon) = \chi^\varepsilon(\boldsymbol{x}) f(\boldsymbol{x}) \qquad \text{in } \Omega.$$

This equation is in divergence form so there is a variational principle for its solutions. Consider the problem of minimizing $\mathcal{I}^\varepsilon$ on V, where $\mathcal{I}^\varepsilon \colon V \to \mathbb{R}$ is defined by, $w \in V$,

$$\mathcal{I}^\varepsilon(w) := \int_\Omega \chi^\varepsilon(\boldsymbol{x}) \left(\frac{a^\varepsilon(\boldsymbol{x})}{2} |\nabla w|^2 - f(\boldsymbol{x}) w \right) \mathrm{d}\Omega + \int_{\Gamma_N} \chi^\varepsilon(\boldsymbol{x}) \left(\frac{\alpha}{2} w^2 - \gamma_N w \right) \mathrm{d}\sigma. \quad (5)$$

Note that this functional involves the advection field solely through the multiplier $\chi^\varepsilon(\boldsymbol{x})$. Using the theory in [13], there will be a minimizer of $\mathcal{I}^\varepsilon$. When $v \in V \cap C(\overline{\Omega})$, the first variation of $\mathcal{I}^\varepsilon$ at $u^\varepsilon \in V$,

$$\delta \mathcal{I}^\varepsilon(u^\varepsilon, v) = \lim_{t \to 0} \frac{1}{t} \left[\mathcal{I}^\varepsilon(u^\varepsilon + tv) - \mathcal{I}^\varepsilon(u^\varepsilon) \right],$$

exists and is given by

$$\delta \mathcal{I}^\varepsilon(u^\varepsilon, v) = \int_\Omega \chi^\varepsilon(\boldsymbol{x}) \left(a^\varepsilon(\boldsymbol{x}) \nabla u^\varepsilon \cdot \nabla v - f(\boldsymbol{x}) v \right) \mathrm{d}\Omega + \int_{\Gamma_N} \chi^\varepsilon(\boldsymbol{x}) \left(\alpha u^\varepsilon v - \gamma_N u^\varepsilon v \right) \mathrm{d}\sigma. \quad (6)$$

At the minimizer $u^\varepsilon \in V$, (6) will be zero. Hence, we have

$$\int_\Omega \chi^\varepsilon(\boldsymbol{x}) a^\varepsilon(\boldsymbol{x}) \nabla u^\varepsilon \cdot \nabla v \, \mathrm{d}\Omega + \int_{\Gamma_N} \chi^\varepsilon(\boldsymbol{x}) \alpha u^\varepsilon v \, \mathrm{d}\sigma = \int_\Omega \chi^\varepsilon(\boldsymbol{x}) f(\boldsymbol{x}) v \, \mathrm{d}\Omega + \int_{\Gamma_N} \chi^\varepsilon(\boldsymbol{x}) \gamma_N u^\varepsilon v \, \mathrm{d}\sigma \quad (7)$$

for all $v \in V$. It is easy to see that (7) is the weak formulation for the boundary value problem

$$-\nabla \cdot (\chi^\varepsilon(\boldsymbol{x}) a^\varepsilon(\boldsymbol{x}) \nabla u^\varepsilon) = \chi^\varepsilon(\boldsymbol{x}) f(\boldsymbol{x}) \qquad \text{in } \Omega, \quad (8)$$

$$u^\varepsilon(\boldsymbol{x}) = 0 \qquad \text{on } \Gamma_D, \quad (9)$$

$$-\boldsymbol{n} \cdot a^\varepsilon(\boldsymbol{x}) \nabla u^\varepsilon(\boldsymbol{x}) = \alpha u^\varepsilon(\boldsymbol{x}) + \gamma_N(\boldsymbol{x}) \qquad \text{on } \Gamma_N. \quad (10)$$

A corresponding finite element formulation for the problem (8)–(10) reads as follows: find $u_h \in V_h$ such that

$$\int_\Omega \chi^\varepsilon(\boldsymbol{x}) a^\varepsilon(\boldsymbol{x}) \nabla u_h^\varepsilon \cdot \nabla v_h \,\mathrm{d}\Omega + \int_{\Gamma_N} \chi^\varepsilon(\boldsymbol{x}) \alpha u_h^\varepsilon v_h \,\mathrm{d}\sigma = \int_\Omega \chi^\varepsilon(\boldsymbol{x}) f(\boldsymbol{x}) v_h \,\mathrm{d}\Omega + \int_{\Gamma_N} \chi^\varepsilon(\boldsymbol{x}) \gamma_N u_h^\varepsilon v_h \,\mathrm{d}\sigma$$

for all $v_h \in V_h$.

4. Γ-convergence

We now consider a family of functionals (5) depending on $w \in V$. Let $\mathcal{I}_{\mathrm{hom}}$ denote the homogenized functional defined by

$$\mathcal{I}_{\mathrm{hom}}(w) := \int_\Omega \exp(\mathbf{R}_0 \cdot \boldsymbol{x}) \left(\mathcal{W}_{\mathrm{hom}}(\nabla w(\boldsymbol{x})) - f(\boldsymbol{x}) w(\boldsymbol{x})\right) \,\mathrm{d}\Omega + \int_{\Gamma_N} \exp(\mathbf{R}_0 \cdot \boldsymbol{x}) \left(\frac{\alpha}{2} w(\boldsymbol{x})^2 - \gamma_N(\boldsymbol{x}) w(\boldsymbol{x})\right) \,\mathrm{d}\sigma, \quad (11)$$

where the homogenized energy $\mathcal{W}_{\mathrm{hom}}$ is given by

$$\mathcal{W}_{\mathrm{hom}}(\lambda) = \inf_{\xi \in W^{1,2}_{\mathrm{per}}(Y)} \int_Y \frac{a(\boldsymbol{y})}{2} |\lambda + \nabla_y \xi(\boldsymbol{y})|^2 \mathrm{d}Y, \quad (12)$$

where $W^{1,2}_{\mathrm{per}}(Y)$ is the space of elements of $W^{1,2}(Y)$ having the same trace on opposite face of Y.

Applying [10, Theorem 1.5] (see also [3, 7]), the sequence $\mathcal{I}^\varepsilon$ Γ-converges to $\mathcal{I}_{\mathrm{hom}}$. This implies the following fact on the minimizers: for each value $\varepsilon > 0$, let $u^\varepsilon \in V$ be an minimizer of the functional $\mathcal{I}^\varepsilon$. Then, up to a subsequence, u^ε converges weakly in V to a limit u which is precisely a minimizer of the homogenized functional $\mathcal{I}_{\mathrm{hom}}$, i.e.,

$$u^\varepsilon \rightharpoonup u \text{ weakly in } V$$

further

$$\mathcal{I}^\varepsilon(u^\varepsilon) \to \mathcal{I}(u), \quad \inf_{v \in V} \mathcal{I}^\varepsilon(v) \to \min_{v \in V} \mathcal{I}(v) \quad \text{and} \quad \mathcal{I}(u) = \min_{v \in V} \mathcal{I}(v).$$

Computing the infima in (12) and minimizers $u \in V$ of (11) yields the following homogenized problem,

$$\int_\Omega \exp(\mathbf{R}_0 \cdot \boldsymbol{x}) A^* \nabla u \cdot \nabla v \,\mathrm{d}\Omega + \int_{\Gamma_N} \exp(\mathbf{R}_0 \cdot \boldsymbol{x}) \alpha u v \,\mathrm{d}\sigma = \int_\Omega \exp(\mathbf{R}_0 \cdot \boldsymbol{x}) f(\boldsymbol{x}) v \,\mathrm{d}\Omega + \int_{\Gamma_N} \exp(\mathbf{R}_0 \cdot \boldsymbol{x}) \gamma_N u v \,\mathrm{d}\sigma \quad (13)$$

for all $v \in V$. The homogenized diffusion tensor is given by its entries

$$A_{ij} = \int_Y a(\boldsymbol{y}) \left(\mathbf{e}_i + \nabla_y w_i\right) \left(\mathbf{e}_j + \nabla_y w_j\right) \mathrm{d}Y, \tag{14}$$

where w_i are defined as the unique solutions in $W^{1,2}_{\mathrm{per}}(Y)$ of the cell problems:

$$-\nabla_y \cdot (a(\boldsymbol{y})(\mathbf{e}_i + \nabla_y w_i)) = 0 \ \text{ in } Y \quad \text{ and } \quad \int_Y w_i \,\mathrm{d}Y = 0, \quad i = 1, \ldots, d. \tag{15}$$

5. Application to the 1D problem

We now study a convection-diffusion process in layered medium which is described by the following one-dimensional problem. Let $\Omega = (0,1)$ be an interval in $\mathbb{R}$, $\varepsilon > 0$, and consider the problem

$$-\frac{du}{dx}\left(a^\varepsilon(x)\frac{du}{dx}\right) + b^\varepsilon(x)\frac{du}{dx} = 1 \qquad \text{in } (0,1) \tag{16}$$

$$u(x=0) = u(x=1) = 0. \tag{17}$$

Here, $a^\varepsilon(x) = a(x/\varepsilon)$ and $b^\varepsilon(x) = b(x/\varepsilon)$ and we assume that a and b are piecewise constant 1-periodic functions such that

$$a(y) = \begin{cases} a_1 & y \in (0,1/2) \\ a_2 & y \in (1/2,1) \end{cases} \qquad b(y) = \begin{cases} b_1 & y \in (0,1/2) \\ b_2 & y \in (1/2,1) \end{cases} \tag{18}$$

where $a_1, a_2 \in \mathbb{R}_+$ and $b_1, b_2 \in \mathbb{R}$. In the one-dimensional case, analytical solutions to (14)–(15) are well known, see e.g. [5]. In particular, for (18) we have

$$A^* = \frac{2a_1a_2}{a_1 + a_2} \quad \text{ and } \quad R_0 = -\frac{a_1b_2 + a_2b_1}{2a_1a_2}. \tag{19}$$

Given any positive integer N, let $\pi\colon 0 = x_0 < \cdots < x_{N+1} = 1$ denote a uniform partition of the unit interval with nodes $x_i = ih$, $h = 1/(N+1)$, $0 \le i \le N+1$. Then V_h denotes the set of all continuous functions defined on $[0,1]$ which are linear on each subinterval $[x_i, x_{i+1}]$, $0 \le i \le N$, and which vanish at the end points.

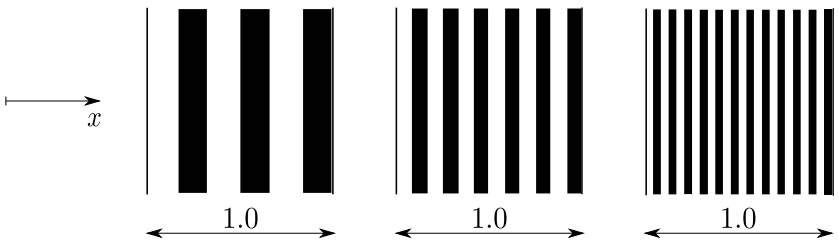


Figure 1: Layered medium.

A convenient basis on V_h can be constructed in a standard way as follows: let $v_i(x)$, $1 \leq iN$, be the element in V_h which satisfies $v_i(x_j) = \delta_{ij}$, $1 \leq j \geq N$. Then the collection $\{v_i(x),\ 1 \leq i \leq N\}$ constitutes a basis in V_h: any function $v_h(x) \in V_h$ can be written as

$$v_h(x) = \sum_{i=1}^{N} \zeta_i v_i(x).$$

A finite element formulation corresponding to the problem (16)–(17) can be written as follows: find $u_h^\varepsilon \in V_h$, $u_h^\varepsilon(x) = \sum_{i=1}^N \xi_i v_i(x)$, such that $(1 \leq i \leq N)$

$$\sum_{j=1}^{N} \left\{ \int_0^1 a^\varepsilon(x) \frac{dv_i}{dx} \frac{dv_j}{dx}\, \mathrm{d}x \right\} \xi_j + \sum_{j=1}^{N} \left\{ \int_0^1 b^\varepsilon(x) v_i \frac{dv_j}{dx}\, \mathrm{d}x \right\} \xi_j = \int_0^1 v_i\, \mathrm{d}x. \tag{20}$$

The results presented in this section are obtained using an in-house PYTHON code. Recall that the standard Galerkin formulation gives node-to-node spurious oscillations. In Figure 2, we compare the numerical solutions from the standard Galerkin formulation for various steps h. As one can see from the figure, the Galerkin formulation produces spurious node-to-node oscillations for high values of h (namely $h = 0.1$ and $h = 0.05$).

We now apply the new variational formulation to the 1D problem according to (13) (reformulated to the 1D case and Dirichlet boundary conditions). The corresponding finite element formulation reads as follows: find $u_h \in V_h$, $u_h(x) = \sum_{i=1}^N \eta_i v_i(x)$, such that

$$\sum_{j=1}^{N} \left\{ \int_0^1 \exp(R_0 x) A^* \frac{dv_i}{dx} \frac{dv_j}{dx}\, \mathrm{d}x \right\} \eta_j = \int_0^1 \exp(R_0 x) v_i\, \mathrm{d}x, \quad 1 \leq i \leq N. \tag{21}$$

According to a specific construction of the basis on V_h, it is easy to see that

$$u_h(x) = \eta_{j-1} g_{-1}(x) + \eta_j g_0(x) + \eta_{j+1} g_{+1}(x) \ \text{ on } \ \langle x_{j-1}, x_{j+1} \rangle,$$

where

$$g_{-1}(x) = \begin{cases} -\frac{x - x_j}{h} & -h \leq x - x_j \leq 0 \\ 0 & 0 < x - x_j \leq +h \end{cases}$$

$$g_0(x) = \begin{cases} \frac{x - x_j + h}{h} & -h \leq x - x_j \leq 0 \\ \frac{h - (x - x_j)}{h} & 0 < x - x_j \leq +h \end{cases}$$

$$g_{+1}(x) = \begin{cases} 0 & -h \leq x - x_j \leq 0 \\ \frac{x - x_j}{h} & 0 < x - x_j \leq +h \end{cases}$$

Hence, in view of (21), $(\eta_1, \eta_2, \ldots, \eta_N)$ is a solution of the following system of equations

$$\eta_{i-1}\omega_{-1} + \eta_i \omega_0 + \eta_{i+1}\omega_{+1} = 1, \quad 1 \leq i \leq N, \quad \eta_0 = \eta_{N+1} = 0,$$

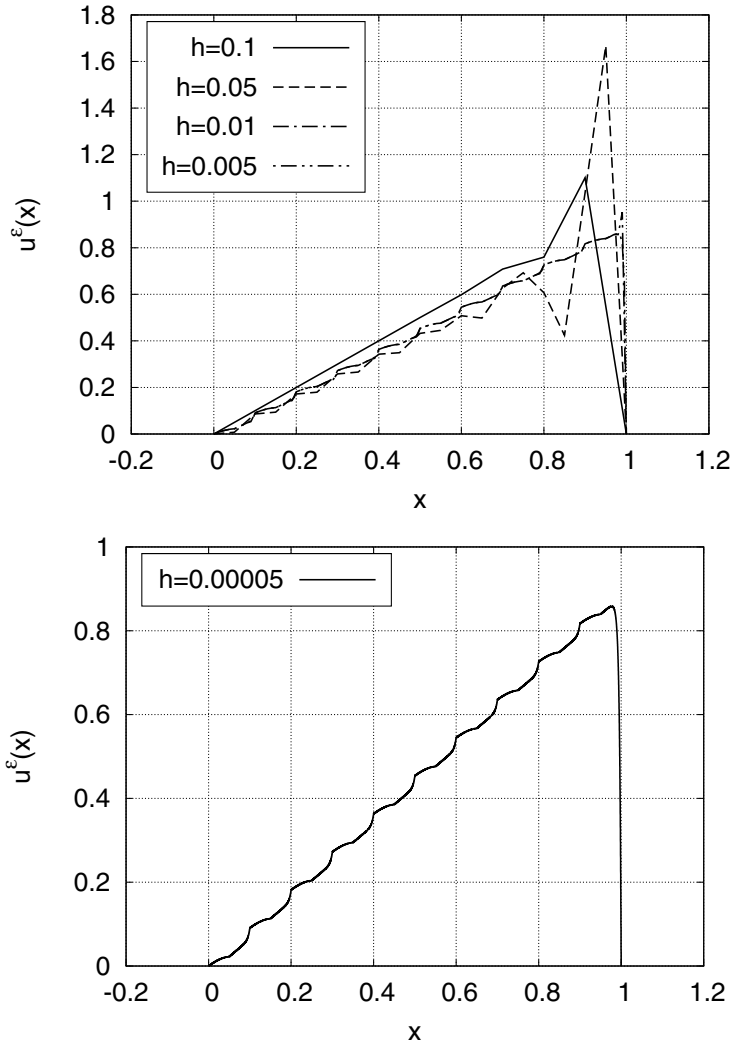


Figure 2: Comparison of Galerkin approximations for various steps h and fixed $\varepsilon = 0.1$. The following values have been chosen in this example: $a_1 = 0.05$, $a_2 = 0.005$, $b_1 = 0.8$ and $b_2 = 1.2$.

where

$$\omega_{-1} = \frac{\int_{-h}^{+h} \exp(R_0 x) A^* g_0'(x) g_{-1}'(x)\,\mathrm{d}x}{\int_{-h}^{+h} \exp(R_0 x) g_0(x)\,\mathrm{d}x} = \frac{R_0 A^*}{2h}(1 - \coth(R_0 h/2),$$

$$\omega_0 = \frac{\int_{-h}^{+h} \exp(R_0 x) A^* g_0'(x) g_0'(x)\,\mathrm{d}x}{\int_{-h}^{+h} \exp(R_0 x) g_0(x)\,\mathrm{d}x} = \frac{R_0 A^*}{h}\coth(R_0 h/2),$$

$$\omega_{+1} = \frac{\int_{-h}^{+h} \exp(R_0 x) A^* g_0'(x) g_{+1}'(x)\,\mathrm{d}x}{\int_{-h}^{+h} \exp(R_0 x) g_0(x)\,\mathrm{d}x} = \frac{-R_0 A^*}{2h}(1 + \coth(R_0 h/2)).$$

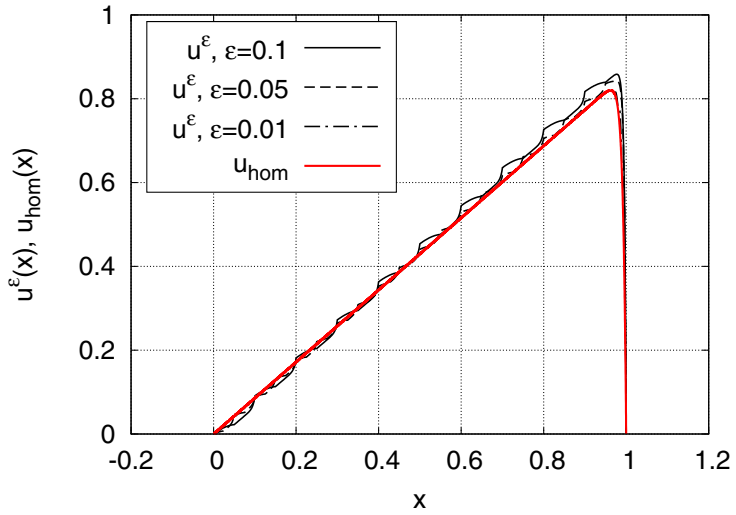


Figure 3: Comparison of the fine scale solutions and the homogenized results.

It is worth noting that the coefficients ω_{-1}, ω_0 and ω_{+1} are, respectively, the same as obtained using the optimal artificial diffusion method, c.f. [11]. In Figure 3, we compare the numerical solution u_{hom} of (21) obtained using the stable homogenized formulation with $h = 0.05$ (based on the variational principle, which gives nodally exact solutions) with the solutions of (20) using the standard Galerkin approximations with $h = 5.0 \times 10^{-5}$ and for various values of scale parameters of ε.

Acknowledgements

Michal Beneš has been financially supported by the European Regional Development Fund (project No. CZ.02.1.01/0.0/0.0/16-019/0000778) within activities of the Center of Advanced Applied Sciences (CAAS). The research of David Šilhánek was supported by SGS, project number SGS22/001/OHK1/1T/11.

References

[1] Auchmuty, G.: Variational principles for advection-diffusion problems. Computers and Mathematics with Applications **75** (2018) 1882–1886.

[2] Bensoussan, A., Lions, J.L. and Papanicolaou, G.: *Asymptotic analysis for periodic structures*, in Studies in Mathematics and its Applications, Vol. 5, North-Holland Publishing Co., Amsterdam, 1978.

[3] Braides A.: Homogenization of some almost periodic coercive functional. Rend. Acad. Naz. Sci. XL, 103 (1985) 313–322.

[4] Ciarlet, P.G.: *The finite element method for elliptic problems*. Elsevier, 1978.

[5] Cioranescu, D. and Donato, P.: *An Introduction to Homogenization*. Oxford University Press, 1999.

[6] Gresho, P.M. and Sani, R.L.: *Incompressible flow and the finite element method: Advection-Diffusion*, Volume 1, John-Wiley & Sons, Inc., Chichester, 2000.

[7] Hornung, U. (ed.): *Homogenization and porous media*, in Interdisciplinary Applied Mathematics, Vol. 6, Springer-Verlag, New York, 1997.

[8] Jikov, V.V., Kozlov, S.M. and Oleinik, O.A.: *Homogenization of differential operators and integral functionals*, Springer-Verlag, Berlin, 1994.

[9] Kufner, A., John, O., Fučík, S.: *Function spaces.* Academia, 1977.

[10] Müller, S.: Homogenization of nonconvex integral functionals and cellular elastic materials. Arch. Rat. Mech. Anal. 99 (1987) 189–212.

[11] Nakshatrala, K.B. and Valocchi, A.J.: Variational structure of the optimal artificial diffusion method for the advection-diffusion equation, Int. J. Comput. Methods, **7** (2010), 559–572.

[12] Ortiz, M.: A variational formulation for convection-diffusion problems, Internat. J. Engrg. Sci. **23** (1985), 717–731.

[13] Rektorys, K.: *Variational Methods in Mathematics, Science and Engineering*, Springer Dordrecht, 1977.

Programs and Algorithms of Numerical Mathematics 21
J. Chleboun, P. Kůs, J. Papež, M. Rozložník, K. Segeth, J. Šístek (Eds.)
Institute of Mathematics CAS, Prague 2023

GODUNOV-LIKE NUMERICAL FLUXES FOR CONSERVATION LAWS ON NETWORKS

Lukáš Vacek, Václav Kučera

Faculty of Mathematics and Physics, Charles University
Sokolovská 49/83, 186 00, Prague, Czech Republic
lvacek@karlin.mff.cuni.cz, kucera@karlin.mff.cuni.cz

Abstract: We describe a numerical technique for the solution of macroscopic traffic flow models on networks of roads. On individual roads, we consider the standard Lighthill-Whitham-Richards model which is discretized using the discontinuous Galerkin method along with suitable limiters. In order to solve traffic flows on networks, we construct suitable numerical fluxes at junctions based on preferences of the drivers. Numerical experiment comparing different approaches is presented.

Keywords: traffic flow, discontinuous Galerkin method, junctions, numerical flux

MSC: 65M60, 76A30, 90B20

1. Introduction

Let us have a road and an arbitrary number of cars. We would like to model the movement of cars on our road. We call this model a traffic flow model. We use *macroscopic models*, where we view our traffic situation as a continuum and study the density of cars in every point of the road. This model is described by partial differential equations.

Our aim is to numerically solve macroscopic models of traffic flow. Our unknown is density at point x and time t. As we shall see later, the solution can be discontinuous. Due to the need for discontinuous approximation of density, we use the *discontinuous Galerkin method*. The aim of modelling is understanding traffic dynamics and deriving possible control mechanisms for traffic.

1.1. Macroscopic traffic flow models

We begin with the mathematical description of macroscopic vehicular traffic, cf. [4] and [6] for details. First, we consider a single road described mathematically as a one-dimensional interval. In the basic macroscopic models, traffic flow is described by two basic fundamental quantities – *traffic flow* Q and *traffic density* ρ.

DOI: 10.21136/panm.2022.23

In article [3], Greenshields realized that traffic flow is a function which depends only on traffic density in homogeneous traffic. The relationship between the ρ and Q is described by the *fundamental diagram*. There are many different proposals for the traffic flow Q derived from real traffic data, cf. [4]. Here we present only *Greenshields model*, which define traffic flow as $Q(\rho) = v_{\max}\rho(1 - \frac{\rho}{\rho_{\max}})$, where $v_{\max}$ is the maximal velocity and $\rho_{\max}$ is the maximal density.

Since the number of cars is conserved, the basic governing equation is a *nonlinear first order hyperbolic partial differential equation*, cf.

$$\rho_t + \left(Q(\rho)\right)_x = 0, \qquad x \in \mathbb{R},\ t > 0. \tag{1}$$

Equation (1) must be supplemented by the initial condition $\rho(x, 0) = \rho_0(x)$, $x \in \mathbb{R}$ and an inflow boundary condition.

Following [2], we consider a complex *network* represented by a directed graph. Each vertex (junction) has a finite set of incoming and outgoing edges (roads). In our case it is sufficient to study our problem only on a *simple network* with one vertex J and its n incoming and m outgoing adjacent edges. On each road we consider equation (1), while at the vertex we consider a *Riemann solver*.

It is also necessary to take into account the preferences of drivers how the traffic from incoming roads is distributed to outgoing roads according to some predetermined coefficients. There is a *traffic–distribution matrix* A describing the distribution of traffic among outgoing roads, i.e.

$$A = \begin{bmatrix} \alpha_{n+1,1} & \cdots & \alpha_{n+1,n} \\ \vdots & \vdots & \vdots \\ \alpha_{n+m,1} & \cdots & \alpha_{n+m,n} \end{bmatrix}, \tag{2}$$

where for all $i \in \{1, \ldots, n\}$, $j \in \{n+1, \ldots, n+m\}$: $\alpha_{j,i} \in [0, 1]$ and for all $i \in \{1, \ldots, n\}$: $\sum_{j=n+1}^{n+m} \alpha_{j,i} = 1$. The i^{th} column of A describes how the traffic from the incoming road I_i distributes to the outgoing roads at J.

We denote the endpoints of road I_i as a_i, b_i. We introduce the notation of spatial limits $\rho_i^{(L)}(b_i, t) := \lim_{x \to b_{i-}} \rho_i(x, t)$ and $\rho_i^{(R)}(a_i, t) := \lim_{x \to a_{i+}} \rho_i(x, t)$.

Let $\rho = (\rho_1, \ldots, \rho_{n+m})^T$ be a *weak solution at* J, see [2, Definition 5.1.8], where ρ has bounded variation in space. Then ρ satisfies the *Rankine–Hugoniot condition*, which represents the conservation of cars at J:

$$\sum_{i=1}^{n} Q(\rho_i^{(L)}(b_i, t)) = \sum_{j=n+1}^{n+m} Q(\rho_j^{(R)}(a_j, t)) \tag{3}$$

for almost every $t > 0$, cf. [2, Lemma 5.1.9].

1.2. Discontinuous Galerkin method

As an appropriate method for the numerical solution of (1), we choose the *discontinuous Galerkin* (DG) method, which is essentially a combination of finite volume and finite element techniques, cf. [1].

Consider an interval $\Omega = (a, b)$. Let $\mathcal{T}_h$ be a partition of $\overline{\Omega}$ into a finite number of intervals (elements) $K = [a_K,\ b_K]$. We denote the set of all boundary points of all elements by $\mathcal{F}_h$. We seek the numerical solution in the space of discontinuous piecewise polynomial functions $S_h = \{v;\ v|_K \in P^p(K),\ \forall K \in \mathcal{T}_h\}$, where $p \in \mathbb{N}_0$ and $P^p(K)$ denotes the space of all polynomials on K of degree at most p. For a function $v \in S_h$ we denote the *jump* in the point $s \in \mathcal{F}_h$ as $[v]_s = v^{(L)}(s) - v^{(R)}(s)$.

We formulate the DG method for the general first order hyperbolic problem $u_t + f(u)_x = g$, $x \in \Omega$, $t \in (0, T)$, which is supplemented by the initial and boundary condition. The DG formulation then reads, cf. [1]: Find $u_h \colon [0, T] \to S_h$ such that

$$\int_\Omega (u_h)_t \varphi \,\mathrm{d}x - \sum_{K \in \mathcal{T}_h} \int_K f(u_h)\varphi_x \,\mathrm{d}x + \sum_{s \in \mathcal{F}_h} H(u_h^{(L)}, u_h^{(R)})\,[\varphi]_s = \int_\Omega g\varphi \,\mathrm{d}x,$$

for all $\varphi \in S_h$. On $\mathcal{F}_h$ we use the approximation $f(u_h) \approx H(u_h^{(L)}, u_h^{(R)})$, where H is a *numerical flux*. We use the *Godunov* numerical flux, which is defined as the flux at the exact solution of the Riemann problem with $u_i^{(L)}$ and $u_i^{(R)}$, cf. [5]. It can be expressed as

$$H_{\text{orig}}^{\text{God}}\left(u^{(L)}, u^{(R)}\right) = \begin{cases} \min_{u^{(L)} \le u \le u^{(R)}} f(u), & \text{if } u^{(L)} < u^{(R)}, \\ \max_{u^{(R)} \le u \le u^{(L)}} f(u), & \text{if } u^{(L)} \ge u^{(R)}. \end{cases} \tag{4}$$

For our purpose, we derive alternative form, which is inspired by maximum possible traffic flow (see Section 2) in case with one incoming and one outgoing road.

Definition 1 (Alternative form of Godunov numerical flux)**.** *Let the Greenshields traffic flow f have global maximum at u_*. Then the Godunov numerical flux is defined as*

$$H^{\text{God}}\left(u^{(L)}, u^{(R)}\right) = \min\left\{f_{\text{in}}(u^{(L)}), f_{\text{out}}(u^{(R)})\right\}, \tag{5}$$

where

$$f_{\text{in}}(u^{(L)}) = \begin{cases} f(u^{(L)}), & \text{if } u^{(L)} < u_*, \\ f(u_*), & \text{if } u^{(L)} \ge u_*, \end{cases} \quad f_{\text{out}}(u^{(R)}) = \begin{cases} f(u_*), & \text{if } u^{(R)} \le u_*, \\ f(u^{(R)}), & \text{if } u^{(R)} > u_*. \end{cases}$$

Definition 1 can be interpreted as the maximum possible flow through the boundary, where f_{in} is the maximum possible inflow from the left element and f_{out} is maximum possible outflow to the right element. The expressions (4) and (5) are equivalent in case of Greenshields traffic flow. For simplicity, by $H(\cdot\,, \cdot\,)$ we mean the Godunov numerical flux in the alternative form (5) in the rest of this paper.

For time discretization of the DG method we use the *explicit Euler method*. As a basis for S_h, we use *Legendre polynomials*. We use *Gauss–Legendre quadrature* to evaluate integrals over elements. Because we calculate physical quantity, the result must be in some interval, e.g. $\rho \in [0, \rho_{\max}]$. Thus, we use *limiters* in each time step to obtain the solution in the admissible interval. Following [5], we also apply limiting

to treat spurious oscillations near discontinuities. From the definition of limiters, the average value of the solution doesn't change, i.e. the number of vehicle is conserved. Limiters are necessary in the case of an oscillating solution in a sufficiently small neighborhood of one of the limit values.

2. Maximum possible traffic flow

Based on the traffic distribution matrix, the authors of [2] define an *admissible weak solution of (1)* at the junction J as $\rho = (\rho_1, \ldots, \rho_{n+m})^T$ satisfying

1) ρ is a weak solution at J such that ρ has bounded variation in space, i.e. the Rankine–Hugoniot condition holds.

2) $Q(\rho_j^{(R)}(a_j, \cdot)) = \sum_{i=1}^{n} \alpha_{j,i} Q(\rho_i^{(L)}(b_i, \cdot)),\ \forall j = n+1, \ldots, n+m.$

3) $\sum_{i=1}^{n} Q(\rho_i^{(L)}(b_i, \cdot))$ is a maximum subject to 1) and 2).

Assumption 1) is the conservation of cars at the junction. Assumption 2) takes into account the prescribed preferences of drivers. Assumption 3) postulates that drivers choose to maximize the total flux through the junction.

One problem with the approach of [2] is that explicitly constructing the fluxes requires the solution of a Linear Programming problem on the incoming fluxes. This is done in [2] for the purposes of constructing a Riemann solver at the junction and in [7] for the purposes of obtaining numerical fluxes at the junction in order to formulate the DG scheme. Closed-form solutions are provided in [7] in the special cases $n = 1$, $m = 2$ and $n = 2$, $m = 1$ and $n = 2$, $m = 2$.

Now, we will study the case with one incoming and two outgoing roads. This example is important for us, because it inspires us in the construction of α-inside Godunov flux (see Section 3.2). We use the method described in [7, Section 2.2] with our notation. In this case, we have distribution coefficient $\alpha_{2,1} = \alpha$ and $\alpha_{3,1} = 1 - \alpha$. Then we calculate maximum possible inflow to the junction from incoming road as

$$H_1(t) = \min \left\{ f_{\text{in}}(\rho_1^{(L)}(b_1, t)), \frac{f_{\text{out}}(\rho_2^{(R)}(a_2, t))}{\alpha}, \frac{f_{\text{out}}(\rho_3^{(R)}(a_3, t))}{1 - \alpha} \right\}. \tag{6}$$

The outflow from the junction to outgoing road is calculated as H_1 multiplied by the distribution coefficient, i.e. $H_2(t) = \alpha H_1(t)$ and $H_3(t) = (1 - \alpha) H_1(t)$.

Remark. We can notice, that traffic congestion on one of the outgoing road influences the traffic flow to the second outgoing road. For example, when $f_{\text{out}}(\rho_2^{(R)}) = 0$, then $H_1 = H_2 = H_3 = 0$.

3. Numerical fluxes at junctions

We take a different approach from that of [7] and [2]. Our approach has the advantage that it is simple and explicitly constructed for all junction types. We

shall prove the basic properties of this construction and discuss the differences with the approach of [7] and [2].

In our previous paper [6], we used Lax-Friedrichs numerical flux. When we calculate traffic distribution error, it was nearly impossible to obtain distribution error equal to zero. This phenomenon is hard to justify in cases with low traffic. That is the reason, why we choose Godunov numerical flux. As we show later in Section 3.3, distribution error makes much more sense and is more justified.

3.1. α-outside Godunov flux

At the junction, we consider an incoming road I_i and an outgoing road I_j. If these roads were the only roads at the junction, i.e. if they were directly connected to each other, the (numerical) flux of traffic from I_i to I_j would simply be $H\big(\rho_{hi}^{(L)}(b_i,t),\rho_{hj}^{(R)}(a_j,t)\big)$, where ρ_{hi} and ρ_{hj} are the DG solutions on I_i and I_j, respectively. From the traffic distribution matrix, we know the ratios of the traffic flow distribution to the outgoing roads. Thus, we take the numerical flux $H_j(t)$ at the left point of the outgoing road I_j, i.e. at the junction, at time t as

$$H_j(t) := \sum_{i=1}^{n} \alpha_{j,i} H\big(\rho_{hi}^{(L)}(b_i,t),\rho_{hj}^{(R)}(a_j,t)\big), \tag{7}$$

for $j = n+1,\dots,n+m$. The numerical flux $H_j(t)$ can be viewed as the DG analogue of taking the combined traffic outflow $\sum_{i=1}^{n}\alpha_{j,i}Q\big(\rho_i^{(L)}(b_i,t)\big)$ from all incoming roads and prescribing it as the inflow of traffic to the road I_j.

Similarly, we take the numerical flux $H_i(t)$ at the right point of the incoming road I_i, i.e. at the junction, at time t as

$$H_i(t) := \sum_{j=n+1}^{n+m} \alpha_{j,i} H\big(\rho_{hi}^{(L)}(b_i,t),\rho_{hj}^{(R)}(a_j,t)\big), \tag{8}$$

for $i = 1,\dots,n$. Again, this can be viewed as an approximation of the traffic flow $\sum_{j=n+1}^{n+m}\alpha_{j,i}Q\big(\rho_j^{(R)}(a_j,t)\big)$ being prescribed as the outflow of traffic from I_i.

3.2. α-inside Godunov flux

We find the main difference between maximum possible traffic flow and α-outside Godunov flux is in the position of the distribution coefficient, cf. (6) and (8). That is the reason, why we decide to insert distribution coefficient into Godunov numerical flux.

Definition 2 (Godunov numerical flux with parameter)**.** *Let Greenshields traffic flow f has global maximum at u_*. Then Godunov numerical flux with parameter is defined as*

$$H^{\mathrm{God}}\left(u^{(L)},u^{(R)},\alpha\right) = \min\left\{\alpha f_{\mathrm{in}}(u^{(L)}), f_{\mathrm{out}}(u^{(R)})\right\}, \tag{9}$$

where $f_{\mathrm{in}}(u^{(L)})$ and $f_{\mathrm{out}}(u^{(R)})$ are defined as in Definition 1.

The reason, why we put distribution coefficient in front of the f_{in} term, is the representation of the real supply from the incoming road. Only $\alpha_{j,i} f_{\text{in}}(\rho_i^{(L)}(b_i,t))$ cars per time want to go from incoming road i to outgoing road j. In case of $\alpha = 1$, the flux (9) is equivalent to the alternative form of Godunov numerical flux (5). For simplicity, by $H(\cdot\,,\cdot\,,\cdot\,)$ we mean the Godunov numerical flux with parameter in the rest of this paper.

Now we are able to take numerical flux with α-inside $H_j(t)$ at the left point of the outgoing road I_j at time t as

$$H_j(t) := \sum_{i=1}^{n} H\big(\rho_{hi}^{(L)}(b_i,t), \rho_{hj}^{(R)}(a_j,t), \alpha_{j,i}\big), \tag{10}$$

for $j = n+1, \ldots, n+m$. Similarly, we take the numerical flux with α-inside $H_i(t)$ at the right point of the incoming road I_i at time t as

$$H_i(t) := \sum_{j=n+1}^{n+m} H\big(\rho_{hi}^{(L)}(b_i,t), \rho_{hj}^{(R)}(a_j,t), \alpha_{j,i}\big), \tag{11}$$

for $i = 1, \ldots, n$.

3.3. Properties

It can be shown, that our choice of numerical fluxes conserves the number of cars at junctions, similarly as in (3), see Theorem 1. However, this choice does not distribute the traffic according to the traffic–distribution matrix (2) exactly, only approximately, see Theorem 2.

Firstly, we show the discrete version of Rankine–Hugoniot condition.

Theorem 1 (Discrete Rankine–Hugoniot condition)**.** *The numerical flux at junction J satisfies the discrete version of the Rankine–Hugoniot condition (3):*

$$\sum_{i=1}^{n} H_i(t) = \sum_{j=n+1}^{n+m} H_j(t) \tag{12}$$

whether

a) we use (7) and (8) with α-outside or

b) we use (10) and (11) with α-inside.

Proof. From the definition of H_i and H_j with α-outside, we immediately obtain:

$$\begin{aligned}\sum_{i=1}^{n} H_i(t) &= \sum_{i=1}^{n} \sum_{j=n+1}^{n+m} \alpha_{j,i} H\big(\rho_{hi}^{(L)}(b_i,t), \rho_{hj}^{(R)}(a_j,t)\big) \\ &= \sum_{j=n+1}^{n+m} \sum_{i=1}^{n} \alpha_{j,i} H\big(\rho_{hi}^{(L)}(b_i,t), \rho_{hj}^{(R)}(a_j,t)\big) = \sum_{j=n+1}^{n+m} H_j(t).\end{aligned}$$

Proof of the case b) is similar with the corresponding definition of H_i and H_j with α-inside. □

The second theorem is important for identifying the difference between maximum possible traffic flow described in Section 2 and our numerical fluxes at junction.

Theorem 2 (Traffic distribution error)**.** *The numerical flux at junction satisfies*

$$H_j(t) = \sum_{i=1}^{n} \alpha_{j,i} H_i(t) + E_j(t) \tag{13}$$

for all $j = n+1, \ldots, n+m$*, where*

a) in case of (7) *and* (8) *with* α*-outside, the error term is*

$$E_j(t) = \sum_{i=1}^{n} \sum_{\substack{l=n+1 \\ l \neq j}}^{n+m} \alpha_{j,i}\alpha_{l,i}\big(H_{i,j}(t) - H_{i,l}(t)\big), \tag{14}$$

where $H_{i,j}(t) := H\big(\rho_{hi}^{(L)}(b_i, t), \rho_{hj}^{(R)}(a_j, t)\big)$.

b) in case of (10) *and* (11) *with* α*-inside, the error term is*

$$E_j(t) = \sum_{i=1}^{n} \sum_{\substack{l=n+1 \\ l \neq j}}^{n+m} \left(\alpha_{l,i} H_{i,j}(t) - \alpha_{j,i} H_{i,l}(t)\right), \tag{15}$$

where $H_{i,j}(t) := H\big(\rho_{hi}^{(L)}(b_i, t), \rho_{hj}^{(R)}(a_j, t), \alpha_{j,i}\big)$.

Proof. We prove only the case a). By definition (7),

$$H_j(t) = \sum_{i=1}^{n} \alpha_{j,i} H_{i,j}(t) = \sum_{i=1}^{n} \alpha_{j,i} H_i(t) + \underbrace{\sum_{i=1}^{n} \alpha_{j,i}\big(H_{i,j}(t) - H_i(t)\big)}_{E_j(t)},$$

where $E_j(t)$ is the error term which we will show has the form (14): by definition (8), we have

$$\begin{aligned} E_j(t) &= \sum_{i=1}^{n} \alpha_{j,i}\Big(H_{i,j}(t) - \sum_{l=n+1}^{n+m} \alpha_{l,i} H_{i,l}(t)\Big) \\ &= \sum_{i=1}^{n} \alpha_{j,i} \sum_{l=n+1}^{n+m} \alpha_{l,i}\big(H_{i,j}(t) - H_{i,l}(t)\big) \\ &= \sum_{i=1}^{n} \sum_{\substack{l=n+1 \\ l \neq j}}^{n+m} \alpha_{j,i}\alpha_{l,i}\big(H_{i,j}(t) - H_{i,l}(t)\big), \end{aligned}$$

since $\sum_{l=n+1}^{n+m} \alpha_{l,i} = 1$. The proof of case b) is similar. □

An artifact of our model is that sometimes we do not satisfy the traffic–distribution coefficients exactly, cf. (13) and assumption 2) of maximum possible traffic flow (see Section 2). This corresponds to the real situation where some cars decide to use another road instead of staying in the traffic jam.

For the comparison of maximum possible traffic flow described by (6) and our approach, we take a junction with one incoming and two outgoing roads. As it was mentioned in Remark 2, if there is a traffic jam in one of the outgoing roads, the maximum possible flow through the junction is 0, thus the whole junction is blocked by a traffic jam in one of the outgoing roads. On the other hand, in our approach the junction is not blocked by a traffic jam on one of the outgoing roads and the cars can still go into the second outgoing road according to the traffic–distribution coefficients. So our choice of numerical fluxes corresponds to modelling turning lanes, which allow the cars to separate before the junction according to their preferred turning direction. Since macroscopic models are intended for long (multi–lane) roads with huge numbers of cars, our model makes sense in this situation. The original approach from [2, 7] works for one–lane roads, where splitting of the traffic according to preference is not possible.

Another difference is that we can use all varieties of traffic lights. The model of [2, 7] can use only the so-called full green lights. Our approach gives us an opportunity to change the lights for each direction separately.

4. Numerical results

We consider a simple network with one incoming road (Road 1) and two outgoing roads (Road 2 and Road 3). The network will be closed at their endpoints (a_1, b_2 and b_3). Thus, we can check the total number of cars, because we have neither inflow nor outflow. We choose $\alpha_{2,1} = 0.75$ and $\alpha_{3,1} = 0.25$. The length of all roads is 1. As we mention above, we use the combination of the explicit Euler method (step size $\tau = 10^{-4}$) and DG method (number of elements $N = 150$ on each road). We calculate the piecewise linear approximations of solutions and we use two Gaussian quadrature points in each element. We use Greenshields model with $v_{\max} = 1$ and $\rho_{\max} = 1$. We have initial conditions

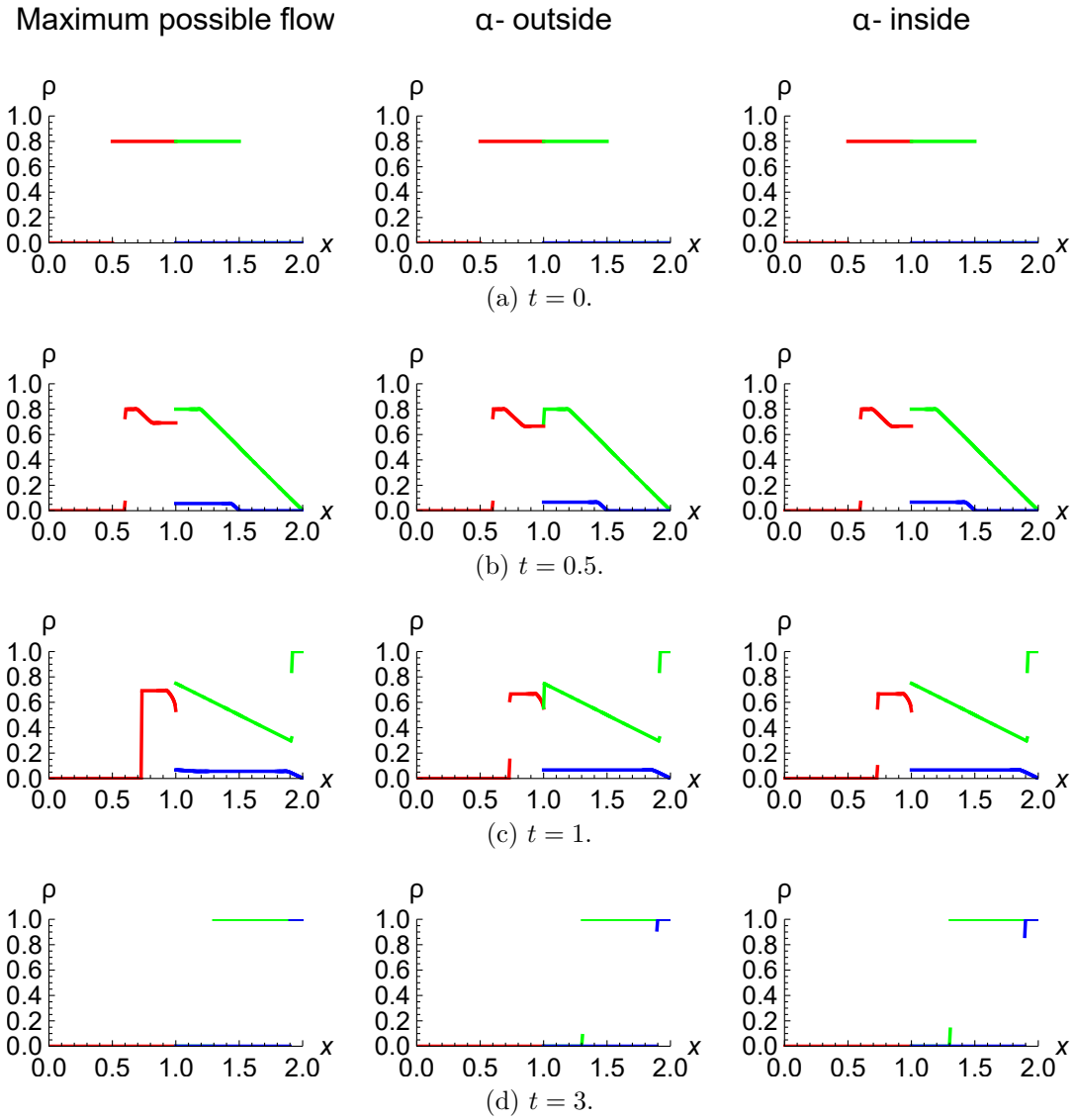
$$\rho_{0,1}(x) = \begin{cases} 0, \\ 0.8, \end{cases} \qquad \rho_{0,2}(x) = \begin{cases} 0.8, \\ 0, \end{cases} \qquad \rho_{0,3}(x) = \begin{cases} 0, & x \in [0, 0.5], \\ 0, & x \in (0.5, 1], \end{cases}$$

cf. Figure 1a. There is 0.4 cars on Road 1. These cars are distributed into Road 2 (it has 0.4 cars already) and Road 3 by distribution coefficients. At the end, we can expect 0.7 cars on Road 2 and 0.1 cars on Road 3.

We can see the results in Figure 1. Maximum possible flow is in the left column, the numerical flux with α-outside is in the middle column and with α-inside is in the right column. If we compare inflow to the Road 3 in Figure 1b between Maximum possible flow and our numerical flux (doesn't depend on the position of α), we can see that our numerical flux allows more inflow. If we look at inflow to the Road 2, see

Figures 1b and 1c, we observe similar inflow of maximum possible flow and numerical flux with α-inside. However, the inflow in case of numerical flux with α-outside is slightly smaller. In general, the numerical flux with α-inside is the combination of the two other approaches. It allows as much as possible cars go to the Road 2 like the maximum possible flow do. On the other hand, some drivers change their minds and choose Road 3 instead of Road 2 due to the congestion on Road 2, same as in the case of numerical flux with α-outside.

The final results are in Figure 1d. Maximum possible traffic flow has 0.7 cars on Road 2 and 0.1 on Road 3. Numerical flux with α-outside has 0.6936 cars on Road 2 and 0.1064 on Road 3. Numerical flux with α-inside has 0.6938 cars on Road 2 and 0.1062 on Road 3.



(a) $t = 0$.

(b) $t = 0.5$.

(c) $t = 1$.

(d) $t = 3$.

Figure 1: Comparison of network with Road 1, Road 2 and Road 3

We choose this congested example due to the demonstration of distribution error from Theorem 2. In the non–congested cases, the traffic distribution error is zero.

5. Conclusion

We have demonstrated the numerical solution of macroscopic traffic flow models using the discontinuous Galerkin method. For traffic networks, we construct special numerical fluxes at the junctions. The use of DG methods on networks is not standard. We have described the differences between our approach and the paper [7] by Čanić, Piccoli, Qiu and Ren, where the maximum possible flow at the junction is used.

6. Acknowledgement

The work of L. Vacek and V. Kučera is supported by the Czech Science Foundation, project No. 20–01074S.

References

[1] Dolejší, V. and Feistauer, M.: *Discontinuous Galerkin Method – Analysis and Applications to Compressible Flow.* Springer, Heidelberg, 2015.

[2] Garavello, M. and Piccoli, B.: *Traffic Flow on Networks*, vol. 1. American Institute of Mathematical Sciences (AIMS), Springfield, MO, 2006.

[3] Greenshields, B.D.: A Study of Traffic Capacity. Highway Research Board **14** (1935), 448–477.

[4] Kachroo, P. and Sastry, S.: *Traffic Flow Theory: Mathematical Framework.* University of California Berkeley, 2012.

[5] Shu, C.W.: Discontinuous Galerkin methods: general approach and stability. Numerical solutions of partial differential equations **201** (2009).

[6] Vacek, L. and Kučera, V.: Discontinuous Galerkin method for macroscopic traffic flow models on networks. Communications on Applied Mathematics and Computation **4** (2022), 986–1010.

[7] Čanić, S., Piccoli, B., Qiu, J., and Ren, T.: Runge-Kutta Discontinuous Galerkin Method for Traffic Flow Model on Networks. Journal of Scientific Computing **63** (2015), 233–255.

Programs and Algorithms of Numerical Mathematics 21
J. Chleboun, P. Kůs, J. Papež, M. Rozložník, K. Segeth, J. Šístek (Eds.)
Institute of Mathematics CAS, Prague 2023

ON FINITE ELEMENT APPROXIMATION OF FLUID STRUCTURE INTERACTION BY TAYLOR–HOOD AND SCOTT-VOGELIUS ELEMENTS

Karel Vacek[1], Petr Sváček[2]

[1] Institute of Mathematics of the Czech Academy of Sciences,
Žitná 25, Prague 1
[2] Czech Technical University, Department of Technical Mathematics,
Karlovo náměstí 13, 121 35 Praha 2
karel.vacek@fs.cvut.cz, petr.svacek@fs.cvut.cz

Abstract: This paper focuses on mathematical modeling and finite element simulation of fluid-structure interaction problems. A simplified problem of two-dimensional incompressible fluid flow interacting with a rigid structure, whose motion is described with one degree of freedom, is considered. The problem is mathematically described and numerically approximated using the finite element method. Two possibilities, namely Taylor-Hood and Scott-Vogelius elements are presented and implemented. Finally, numerical results of the flow around the cylinder are shown and compared with the reference data.

Keywords: finite element method, FSI problem, ALE method, Taylor-Hood element, Scott-Vogelius element

MSC: 65N15, 65M15, 65F08

1. Introduction

The numerical approximations of the fluid-structure interaction play an important role in many areas of science and engineering, such as the flutter of aircraft wings, flow around wind turbine blades and hydrodynamics compressors. Although in this contribution simpler case of incompressible fluid flow is considered, there are a lot of numerical difficulties to be addressed as treatment of the incompressibility constraint, treatment of the nonlinear convective term, dominating convective term, etc., see e.g. [13], [12], [2]. Moreover, the time change of the computational fluid domain needs to be included. Here we use the well-known arbitrary Lagrangian-Eulerian (ALE) method due to it straightforward manner.

This paper focuses on the finite element method approximation of the Navier-Stokes equations. There are many available strategies, see e.g. [7], [6], but we will further deal only with finite elements which satisfy the Babuška-Brezzi (BB) inf-sup condition. The fulfillment of BB condition guarantees stability of the numerical

DOI: 10.21136/panm.2022.24

scheme, for an overview of such elements, see [7]. Here, we compare two of them. The first one is the well-known Taylor-Hood (TH) finite element (continuous piecewise quadratic velocities and continuous piecewise linear pressures) which satisfies the inf-sup condition only discretely. The second element is the Scott-Vogelius (SV) finite element, i.e. continuous piecewise quadratic velocities and discontinuous piecewise linear pressures, see [3], [6]. In order to satisfy the BB condition, the finite element (FE) approximation space is constructed over a barycentric refinement of an admissible triangulation, see [4]. By choosing this element, the divergence constraint on each element of the mesh is strongly guaranteed, see [6]. This provides us better theoretical convergence of the method.

This paper presents the numerical realization and comparison of numerical results for both TH and SV finite elements by using an in-house solver written in C language. The benchmark problem of nonstationary flow around the vibrating cylinder is chosen and the numerical results are compared with the reference data [1].

2. Governing equation

The mathematical model that describes the fluid-structure interaction consists of movement of the rigid structure (i.e. described by ordinary differential equations) and incompressible Navier-Stokes equations in the Eulerian-Lagrangian (ALE) formulation.

2.1. Incompressible fluid flow

Let us assume a computation fluid domain $\Omega_t \subset \mathbb{R}^2$ to be bounded and polygonal at any time $t \in (0,T)$. Furthermore, its boundary $\partial\Omega$ is assumed to be continuous Lipschitz boundary formed of three disjoint parts Γ_D, Γ_O and Γ_{W_t} (i.e. $\partial\Omega = \Gamma_D \cup \Gamma_O \cup \Gamma_{W_t}$). Flow in the domain Ω_t is described by incompressible Navier-Stokes equations in the ALE formulation. The ALE method is based on ALE mapping A_t which maps the reference domain configuration Ω_0 into the actual domain Ω_t

$$A_t : \Omega_{\text{ref}} \to \Omega_t, \quad X \mapsto x(X,t) = A_t(X), \quad x \in \Omega_{\text{ref}},\ t \in (0,T).$$

The ALE mapping is chosen in order to map reference position of the interface Γ_{W_0} into Γ_{W_t} whose position is defined by the motion of the cylinder, and the positions of boundaries Γ_D and Γ_O are static and they are not dependent on time, for more information see [13].

The Navier-Stokes equations in the ALE formulation for unknown velocity $\mathbf{u}(x,t)\colon \Omega_t \to \mathbb{R}^2$ with components $\mathbf{u} = (u,v)^T$ and the kinematic pressure $p(x,t)\colon \Omega_t \to \mathbb{R}$ read

$$\begin{aligned} \frac{D^A}{Dt}\mathbf{u} + [(\mathbf{u}-\mathbf{w})\cdot\boldsymbol{\nabla}]\mathbf{u} - \nu\boldsymbol{\Delta}\mathbf{u} + \boldsymbol{\nabla}p &= 0 \quad \text{in } \Omega_t, t \in (0,T], \\ \boldsymbol{\nabla}\cdot\mathbf{u} &= 0 \quad \text{in } \Omega_t, t \in (0,T], \end{aligned} \tag{1}$$

where $\frac{D^A}{Dt}$ is the ALE derivative, $\mathbf{w} = \partial A^t/\partial t$ is the domain velocity, see [13] and ν means the kinematic viscosity. We consider the following boundary conditions

$$\mathbf{u}(x,t) = \mathbf{g}(x,t) \quad \text{on } \Gamma_D \times (0,T], \tag{2}$$

$$\mathbf{u}(x,t) = \mathbf{w}(x,t) \quad \text{on } \Gamma_{W_t},\ t \in (0,T], \tag{3}$$

$$-(p - p_{\text{ref}})\mathbf{n} + \nu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} = 0 \quad \text{on } \Gamma_O \times (0,T], \tag{4}$$

where $\mathbf{n}$ is the unit outward normal vector to $\partial\Omega$ and p_{ref} is a reference pressure value at the outlet. Condition (2) is used at the inlet. Furthermore, on the surface of the cylinder, the continuity of velocities is prescribed between the cylinder motion and the airflow. At the outlet, there is the condition (4) which is the so-called do-nothing condition, for more information see [5]. Furthermore, the equations are supplied by an initial condition

$$\mathbf{u}(x,0) = \mathbf{u}^0(x) \quad \text{in } \Omega_0.$$

2.2. Motion of cylinder

We consider the motion of the rigid cylinder with one degree of freedom. This means that the cylinder can move only in vertical directions, as in [1]. Its motion is described using the nondimensionless displacement Y governed by

$$\ddot{Y} + \left(\frac{4\pi\xi}{U_r}\right)\dot{Y} + \left(\frac{4\pi^2}{U_r^2}\right)Y = \frac{C_l}{2M^*}, \tag{5}$$

where $\ddot{Y}$, $\dot{Y}$ are the vertical acceleration and velocity of the rigid cylinder, ξ means the structural damping ratio, $U_r = \frac{U_\infty}{fD}$ represents the reduced velocity of the cylinder (where f denotes the natural frequency of the cylinder) and M^* is the reduced mass of the rigid cylinder ($M^* = \frac{m}{\rho D^2}$). The lift coefficient C_l is computed by

$$C_l = \frac{2}{\rho U_\infty^2 b D} F_l,$$

where b is the depth of the cylinder, U_∞ means free velocity, ρ expresses the density and F_l is the lift force acting on the cylinder of diameter D.

3. Numerical approximation of the Navier-Stokes equations

In order to approximate the problem (1), we start with time discretization. Here, the equidistant division $t_n = n\Delta t$ of the time interval $(0,T)$ is employed with a constant time step $\Delta t > 0$. Further, the velocity approximations at time step $t_n \in (0,T]$ are denoted by

$$\mathbf{u}^n(x) \approx \mathbf{u}(x,t_n) \quad \text{for } x \in \Omega_{t_n},$$

and similarly the pressure approximations are denoted as

$$p^n(x) \approx p(x,t_n) \quad \text{for } x \in \Omega_{t_n}.$$

The domain velocity is at the time instant t_{n+1} approximated by $\mathbf{w}^{n+1}(x) \approx \mathbf{w}(x, t_{n+1})$. The ALE derivative is approximated by implicit Euler (BDF) and we get

$$\begin{aligned} \frac{\mathbf{u}^{n+1} - \tilde{\mathbf{u}}^n}{\Delta t} + ((\mathbf{u}^{n+1} - \mathbf{w}^{n+1}) \cdot \boldsymbol{\nabla})\mathbf{u}^{n+1} - \nu \boldsymbol{\Delta} \mathbf{u}^{n+1} + \boldsymbol{\nabla} p^{n+1} &= 0, \\ \boldsymbol{\nabla} \cdot \mathbf{u}^{n+1} &= 0, \end{aligned} \tag{6}$$

where by $\tilde{\mathbf{u}}^n$ we denote the velocity from time level t^n defined in Ω_{t_n} transformed to $\Omega_{t_{n+1}}$, that is $\tilde{\mathbf{u}}^i := \mathbf{u}^i \circ A_{t_i} \circ A_{t_{n+1}}^{-1}$. Equations (6) are equipped with boundary conditions (2–4).

3.1. Space discretization

For the discretization of problem (6) by using the finite element method, a weak formulation of problem (6) is introduced. First, assuming the fixed time instant t_{n+1}, the simplified notation $\mathbf{u} := \mathbf{u}^{n+1}$, $\mathbf{w} := \mathbf{w}^{n+1}$, $p := p^{n+1}$ and $\Omega := \Omega_{t_{n+1}}$ are considered. Then we define the velocity test space $\boldsymbol{\mathcal{V}}$ and the pressure test space $\mathcal{Q}$ as

$$\boldsymbol{\mathcal{V}} = \left\{ \boldsymbol{\varphi} \in \mathbf{H}^1(\Omega) \colon \boldsymbol{\varphi}(x) = 0 \quad \forall x \in \Gamma_D \cup \Gamma_W \right\},$$

$$\mathcal{Q} = L^2(\Omega),$$

where $\mathbf{H}^1(\Omega) = [H^1(\Omega)]^2$ is the vector Sobolev space and $L^2(\Omega)$ is the Lebesgue space, see [10].

Now, we take a function $\mathbf{v} \in \boldsymbol{\mathcal{V}}$, multiply first of equations (6) and take an arbitrary $q \in \mathcal{Q}$, multiply second of equations (6) by it, integrate over the domain Ω and apply Green's theorem to the pressure gradient $(\boldsymbol{\nabla} p)$ and the viscous term $(-\nu \boldsymbol{\Delta} \mathbf{u})$. Further, the boundary conditions are used. Then the weak formulation reads: Find $\mathbf{u} \in \mathbf{g} + \boldsymbol{\mathcal{V}}$ and $p \in \mathcal{Q}$ such that the equations

$$\frac{1}{\Delta t}(\mathbf{u}, \mathbf{v})_\Omega + \nu(\boldsymbol{\nabla}\mathbf{u}, \boldsymbol{\nabla}\mathbf{v})_\Omega c(\mathbf{u} - \mathbf{w}, \mathbf{u}, \mathbf{v}) - (p, \boldsymbol{\nabla} \cdot \mathbf{v})_\Omega = \frac{1}{\Delta t}(\tilde{\mathbf{u}}^n, \mathbf{v})_\Omega, \tag{7}$$

$$(\boldsymbol{\nabla} \cdot \mathbf{u}, q)_\Omega = 0, \tag{8}$$

hold for any $\mathbf{v} \in \boldsymbol{\mathcal{V}}$ and $q \in \mathcal{Q}$. In these equations, $(\mathbf{u}, \mathbf{v})_\Omega = \int_\Omega \mathbf{u} \cdot \mathbf{v} \mathrm{d}x$ means the scalar product in $\mathbf{L}^2(\Omega)$ and $c(\mathbf{u}, \mathbf{v}, \mathbf{z})$ denotes the trilinear form. This form is defined by $c(\mathbf{u}, \mathbf{v}, \mathbf{z}) = \int_\Omega ((\mathbf{u} \cdot \boldsymbol{\nabla})\mathbf{v}) \cdot \mathbf{z} \mathrm{d}x$ for any $\mathbf{u}, \mathbf{v}, \mathbf{z} \in \boldsymbol{\mathcal{V}}$, for more details see [7].

For the reason of using the finite element method, we define an admissible triangulation τ_h of the domain Ω, see [4]. Now, we assume that the finite element subspaces $\boldsymbol{\mathcal{V}}_h \subset \boldsymbol{\mathcal{V}}$ and $\mathcal{Q}_h \subset \mathcal{Q}$ are approximations of the spaces $\boldsymbol{\mathcal{V}}$ and $\mathcal{Q}$ defined over the triangulation τ_h. These spaces are formed by piecewise polynomial functions. The discrete problem of problem (7) is as follows: Find $\mathbf{u}_h \in \mathbf{g}_h + \boldsymbol{\mathcal{V}}_h$ and $p_h \in \mathcal{Q}_h$ such that equations

$$\frac{1}{\Delta t}(\mathbf{u}_h, \mathbf{v}_h)_\Omega + \nu(\boldsymbol{\nabla}\mathbf{u}_h, \boldsymbol{\nabla}\mathbf{v}_h)_\Omega + c(\mathbf{u}_h - \mathbf{w}_h, \mathbf{u}_h, \mathbf{v}_h) - (p_h, \boldsymbol{\nabla} \cdot \mathbf{v}_h)_\Omega = \frac{1}{\Delta t}(\tilde{\mathbf{u}}_h^n, \mathbf{v}_h)_\Omega,$$

$$(\boldsymbol{\nabla} \cdot \mathbf{u}_h, q_h)_\Omega = 0, \tag{9}$$

hold for any $\mathbf{v}_h \in \boldsymbol{\mathcal{V}}_h$ and $q_h \in \mathcal{Q}_h$. To guarantee stability of the scheme, the couple $\boldsymbol{\mathcal{V}}_h$, $\mathcal{Q}_h$ should satisfy the BB condition, see [7]. In this paper, the well-known Taylor-Hood element and Scott-Vogelius element are used.

Taylor-Hood (P_2/P_1) finite element uses quadratic velocities and linear pressures, i.e. the spaces are defined by

$$\begin{aligned}\boldsymbol{\mathcal{V}}_h =&\left\{\boldsymbol{\varphi} \in \mathbf{C}(\overline{\Omega})\colon (\boldsymbol{\varphi}\big|_K \in P_2(K),\ \forall K \in \tau_h)\right\} \cap \boldsymbol{\mathcal{V}},\\ \mathcal{Q}_h =&\left\{\varphi \in C(\overline{\Omega})\colon (\varphi\big|_K \in P_1(K), \forall K \in \tau_h)\right\}.\end{aligned} \tag{10}$$

Velocity and pressure functions are continuous in the domain Ω, however, the element satisfies the continuity equation only discretely. This is the reason why we use the Scott-Vogelius P_2/P_1^{disc} element, which strongly guarantees divergence-free velocity on each element, see [3].

It has the same space $\boldsymbol{\mathcal{V}}_h$ (10) for velocity as TH element, whereas for the pressure p_h the linear but discontinuous functions are used, i.e.

$$\mathcal{Q}_h^{\text{disc}} = \left\{\varphi\colon \overline{\Omega} \to \mathbb{R}\colon (\varphi\big|_K \in P_1(K), \forall K \in \tau_h)\right\}.$$

In order to satisfy the BB condition, element is constructed over the barycentric refined mesh created from the given regular mesh, see [3]. For both cases the velocity and the pressure can be solved together as both couples satisfy BB condition.

So, the base $\boldsymbol{\Phi}_1, \ldots, \boldsymbol{\Phi}_{N_u}$ of the space $\boldsymbol{\mathcal{V}}_h$, where $N_u = \dim(\boldsymbol{\mathcal{V}}_h)$ is chosen. In addition, a base of the pressure space $\mathcal{Q}_h$ is defined by $\theta_1, \ldots, \theta_{N_p} \in \mathcal{Q}_h$, where $N_p = \dim(\mathcal{Q}_h)$. The approximation of the velocity $\mathbf{u}_h$ can be expressed as a combination of the basis functions of the space $\boldsymbol{\mathcal{V}}_h$

$$\mathbf{u}_h = \sum_{j=1}^{N_u} \alpha_j \boldsymbol{\Phi}_j. \tag{11}$$

and approximation of pressure p_h as a linear combination of the base of space $\mathcal{Q}_h$

$$p_h = \sum_{j=1}^{N_p} \beta_j \theta_j. \tag{12}$$

Equations (11) and (12) are now used in equations (9). Also, the test functions $\mathbf{v}_h$ and q_h in equation (9) are expressed as $\mathbf{v}_h = \boldsymbol{\Phi}_i$, for $i = 1, \ldots, N_u$ and $q_h = \theta_i$, for $i = 1, \ldots, N_p$. Then the system of nonlinear equations is obtained

$$\begin{pmatrix} \frac{1}{\Delta t}\mathbf{M} + \mathbf{A}(\boldsymbol{\alpha}) & \mathbf{B} \\ -\mathbf{B}^T & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{pmatrix} = \begin{pmatrix} \mathbf{f} + \frac{1}{\Delta t}\mathbf{M}\tilde{\mathbf{u}}_h^n \\ 0 \end{pmatrix}, \tag{13}$$

where $\mathbf{M}$ denotes the mass matrix (which depends on the mesh, so it is different in each time step due to ALE formulation), $\mathbf{A}(\boldsymbol{\alpha})$ represents discretization of the nonlinear convective and the viscous terms, $\mathbf{B}$ corresponds to the discrete gradient

and $\mathbf{B}^T$ is the discrete divergence operator. Equations (13) is a system of nonlinear equations which is further to be linearized before it can be solved, see e.g. [9]. In this article the linearization is taken from previous time instant

$$c(\mathbf{u}_h^{n+1}, \mathbf{u}_h^{n+1}, \mathbf{v}_h^{n+1}) \approx c(\mathbf{u}_h^{n}, \mathbf{u}_h^{n+1}, \mathbf{v}_h^{n+1}).$$

Due to this linearization, there is a restriction on the choice of the time step, for more information see [9]. The linearized system of equations can be solved by some iterative methods e.g. GMRES, see [8] or a direct solver such as UMFPACK, MUMPS, MKL, see [11].

4. Numerical results

The benchmark problem of the flow around a movable cylinder [1] is regarded. The numerical results obtained by TH and SV elements are compared to each other and to the reference data. For the numerical solution of the cylinder motion given by equation (5), the Runge-Kutta method of 4th order was used.

The domain Ω_t is shown in Fig. 1 in its initial state. The cylinder has a radius $r = 0.5$ and its center is located at $[x, y] = [19, 20]$. The Dirichlet boundary condition is prescribed ($\mathbf{g} = (1, 0)$) at the inlet $\Gamma_{D,1}$ and at the wall $\Gamma_{D,2}$ zero velocity is given. At the cylinder surface Γ_{W_t} is used Dirichlet boundary condition of the form $\mathbf{u} = \mathbf{w}$. The problem is solved on meshes which are different for each considered finite element. Due to the discontinuity of the SV element, the number of unknowns is much higher than for the TH element. In order to compare both elements, meshes providing a similar number of unknowns are used. The first mesh A for TH leads to solving a system with 89519 unknowns, whereas the use of second mesh B for the SV element results to the system of 90798 unknowns for the SV element.

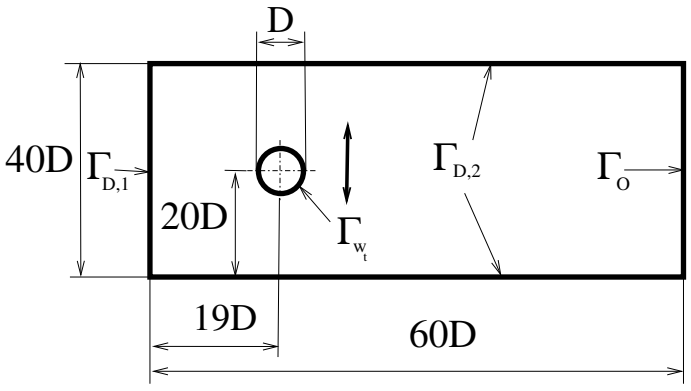


Figure 1: Fluid domain Ω_{ref} of the considered benchmark of flow around a movable cylinder, represented by interface Γ_{W_t}. Boundary Γ_D consists of two parts $\Gamma_{D,1}$ and $\Gamma_{D,2}$, where $\Gamma_{D,1}$ represents inlet and $\Gamma_{D,2}$ represents walls.

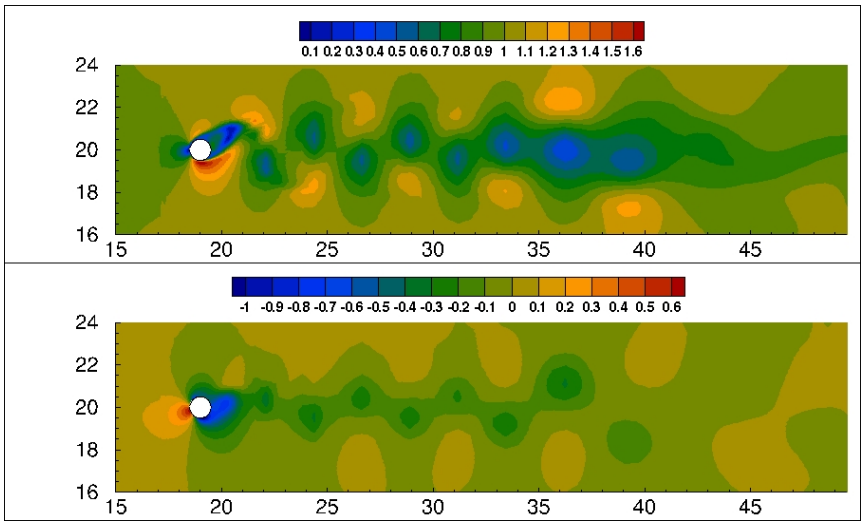


Figure 2: Velocity magnitude $||\mathbf{u}||_2$ (in the upper part) and the pressure p (in the lower part), for Re = 150 and $U_r = 3$ obtained by TH element.

The configuration of the problem is characterized by Reynolds number

$$Re = \frac{U_\infty D}{\nu}, \tag{14}$$

where the free stream velocity is $U_\infty = 1$ and ν expresses the kinematic viscosity. This setup provides us the same $Re = 150$ as the reference data [1]. The computations were done for several cases of different values of natural frequencies of the cylinder (realized by different values of U_r), in all cases the zero damping ratio is considered ($\xi = 0$) and reduced mass as $M^* = 2$, as in [1].

In Fig. 2, the velocity magnitude and pressure field is shown. The Von Karman vortex street is created behind the cylinder and the oscillations of the aerodynamic forces appear leading to the oscillations of the cylinder. For this case of $U_r = 3$, the SV and TH results are almost identical, see Fig. 3a). Further, it can be observed that if the frequency of Von Karman vortex street differs from the natural frequency of the cylinder, there is no resonance. On the other hand for the vortex shadding frequency close to the natural frequency of the cylinder the amplitudes of coefficients C_l, C_d and the amplitude of cylinder vibration are six times higher than for the previous case. Moreover, the peaks of the amplitudes occur in the same time. This phenomenon is called resonance. The resuts obtained by the SV element has slightly higher amplitudes of the displacement than the TH element.

The dependence of amplitude of cylinder oscilation on reduced velocity $U_r \in [3, 8]$ are shown in Fig. 4. The interval where we can see the resonance is the same as in the reference data [1] for both FE discretizations (i.e. $U_r \in [4, 7]$). The maximum amplitude is obtained for the case of $U_r = 4$. Then the amplitude decreases with increasing U_r, and finally for the case $U_r = 8$ there is no resonance.

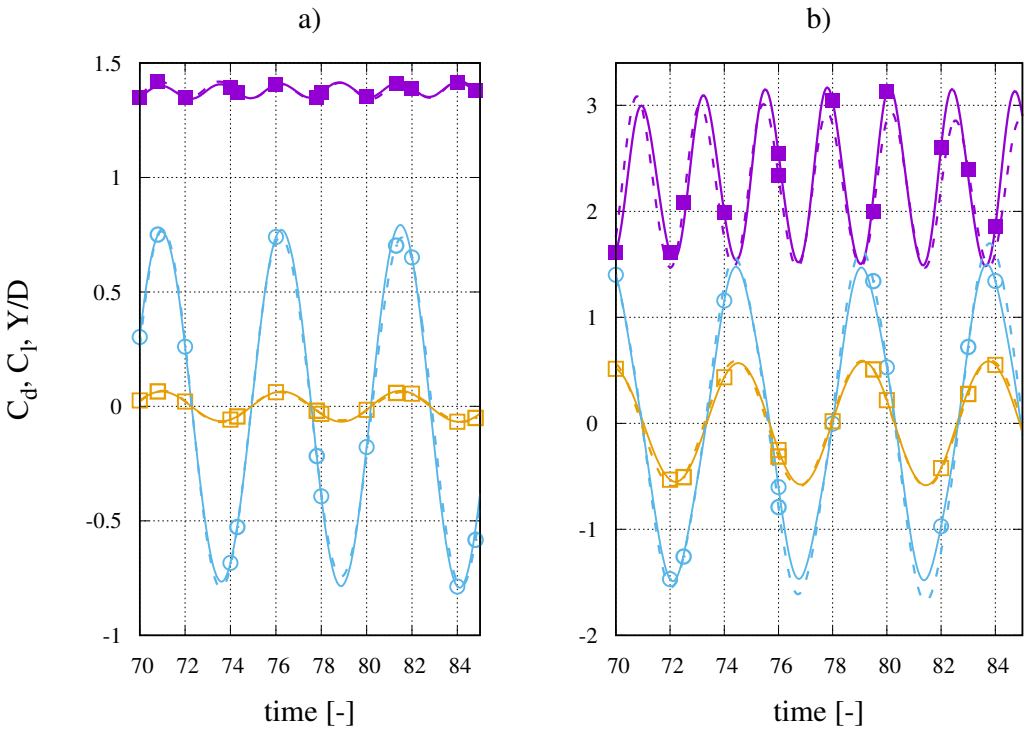


Figure 3: Comparison of lift C_L coefficient (line with empty circle), drag C_D coefficient (line with full square) and position Y/D (line with empty square) over time solved by TH element (full line) and SV element (dashed line) for a) Reynolds number $Re = 150$, $U_r = 3$ and b) Reynolds number $Re = 150$, $U_r = 4$.

5. Conclusion

In this article, the numerical approximation of the interaction of incompressible fluid flow with a movable rigid cylinder is performed. For the fluid flow description the incompressible Navier-Stokes equations in the ALE formulation is used and non-dimensional equation of cylinder motion is utilized. The coupled variables approach is chosen where the Taylor-Hood P_2/P_1 element and the Scott-Vogelius P_2/P_1^{disc} element are compared on the benchmark of movable cylinder in cross-flow, see [1].

The obtained numerical results agree well with the reference data, especially in the resonance occurrence for the considered interval of cylinder reduced velocity. As the maximum amplitudes obtained by the TH and SV elements are practically the same, it shows that the SV element performs well in this case in full agreement with the TH element, which can be considered here as the reference choice.

Although the SV element theoretically provides better results for the considered benchmark test, with similar number of unknowns the TH element has comparable results. The further advantages of the SV element is expected for higher Reynolds numbers, on what we will focus in our future work.

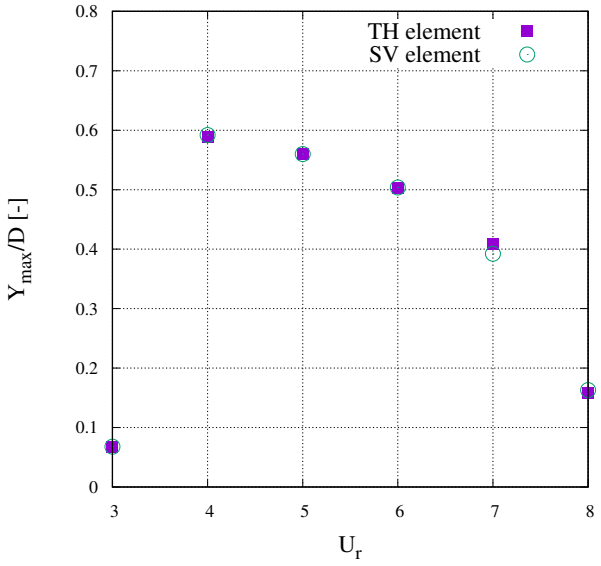


Figure 4: Comparison of maximum amplitudes for different reduced velocities U_r obtined by the TH and the SV elements for $Re = 150$ and $M^* = 2$.

Acknowledgements

Petr Sváček acknowledges the support from the EU Operational Programme Research, Development and Education, and from the Center of Advanced Aerospace Technology (CZ.02.1.01/0.0/0.0/16_019/0000826), Faculty of Mechanical Engineering, Czech Technical University in Prague. The authors also acknowledge the support by the Grant Agency of the Czech Technical University in Prague, grant No. SGS SGS22/148/OHK2/3T/12, and Karel Vacek has also been supported by the Czech Science Foundation (GAČR) project 22-01591S. The Institute of Mathematics of the Czech Academy of Sciences is supported by RVO:67985840.

References

[1] Ahn, H. T. and Kallinderis, Y.: Strongly coupled flow-structure interactions with a geometrically conservative ALE scheme on general hybrid meshes. Journal of Computational Physics **219** (2006), 671–696.

[2] Bao, Y., Huang, C., Zhou, D., Tu, J., and Han, Z.: Two-degree-of-freedom flow-induced vibrations on isolated and tandem cylinders with varying natural frequency ratios. Journal of Fluids and Structures **35** (2012), 50–75.

[3] Case, M. A., Ervin, V. J., Linke, A., and Rebholz, L. G.: A connection between Scott—Vogelius and grad-div stabilized Taylor—Hood FE approximations of the Navier—Stokes equations. SIAM Journal on Numerical Analysis **49** (2011), 1461–1481.

[4] Ciarlet, P. G.: *The Finite Element Method for Elliptic Problems*. Society for Industrial and Applied Mathematics, 2002.

[5] Feistauer, M. and Feistauer, M.: *Mathematical methods in fluid dynamics*. 67, Chapman and Hall/CRC, 1993.

[6] Gauger, N. R., Linke, A., and Schroeder, P. W.: On high-order pressure-robust space discretisations, their advantages for incompressible high Reynolds number generalised Beltrami flows and beyond. The SMAI journal of computational mathematics **5** (2019), 89–129.

[7] Girault, V. and Raviart, P.: *Finite Element Methods for Navier-Stokes Equations: Theory and Algorithms*. Computational Mathematics Series, Springer-Verlag, 1986.

[8] Gupta, P. K. and Pagalthivarthi, K. V.: Application of multifrontal and GMRES solvers for multi-size particulate flow in rotating channels. Progress in Computational Fluid Dynamics, an International Journal **7** (2007), 323–336.

[9] Karniadakis, G. and Sherwin, S.: *Spectral/hp Element Methods for Computational Fluid Dynamics: Second Edition*. Numerical Mathematics and Scientific Computation, OUP Oxford, 2013.

[10] Kufner, A., John, O., and Fučik, S.: *Function Spaces*. Mechanics: Analysis, Springer Netherlands, 1977.

[11] Raju, M. and Khaitan, S.: High performance computing of three-dimensional finite element codes on a 64-bit machine. Journal of Applied Fluid Mechanics **5** (2012).

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

[13] Takashi, N. and Hughes, T. J.: An arbitrary Lagrangian-Eulerian finite element method for interaction of fluid and a rigid body. Computer Methods in Applied Mechanics and Engineering **95** (1992), 115–138.

Programs and Algorithms of Numerical Mathematics 21
J. Chleboun, P. Kůs, J. Papež, M. Rozložník, K. Segeth, J. Šístek (Eds.)
Institute of Mathematics CAS, Prague 2023

ON A COMPUTATIONAL APPROACH TO MULTIPLE CONTACTS / IMPACTS OF ELASTIC BODIES

Jiří Vala[1], Václav Rek[2]

[1] Brno University of Technology, Faculty of Civil Engineering
Institute of Mathematics and Descriptive Geometry
602 00 Brno, Veveří 95, Czech Republic
vala.j@fce.vutbr.cz
[2] Czech Technical University in Prague, Faculty of Transportation Sciences
Department of Mechanics and Materials
110 02 Prague, Konviktská 20, Czech Republic
vaclav.rek@cvut.cz

Abstract: The analysis of dynamic contacts / impacts of several deformable bodies belongs to both theoretically and computationally complicated problems, because of the presence of unpleasant nonlinearities and of the need of effective contact detection. This paper sketches how such difficulties can be overcome, at least for a model problem with several elastic bodies, using i) the explicit time-discretization scheme and ii) the finite element technique adopted to contact evaluations together with iii) the distributed computing platform. These considerations are supported by the references to useful generalizations, motivated by significant engineering applications. Illustrative examples demonstrate this approach on structures assembled from a finite number of shells.

Keywords: contact of elastic bodies, finite element method, finite difference method, distributed computing

MSC: 74M15, 74S05, 74S20, 68Q85

1. Introduction

Reliable computational prediction of the behaviour of deformable bodies under mechanical, thermal, etc. loads belongs to the priorities of both civil and mechanical engineering, due to the development of advanced materials, structures and technologies, whose traditional analysis, coming from long-time experience, certified laboratory measurements and heuristic computational formulae, is not available. Such computational prediction should come from the numerical analysis of initial and boundary value problems for systems of partial differential equations of evolution, based on the principles of classical thermomechanics by [3], namely in the form

DOI: 10.21136/panm.2022.25

of conservation of such scalar quantities as mass, (linear and angular) momentum components and energy, supplied by appropriate constitutive relations, whose parameters have to be identified by experiments. A significant task is the modelling and simulation of the rapid movement of several bodies with potential contacts and impacts, accompanied by their deformation: in addition to the incorporation of various geometrical and physical nonlinearities, the design of an effective algorithm needs e. g. some results from the graph theory and the distributed and parallel computing.

After these motivational comments (*Section* 1) we intend to present a model problem of multiple contacts / impacts of elastic (or viscoelastic) deformable bodies. The overview of physical and mathematical background (*Section* 2) will be followed by some details of the computational approach (*Section* 2), with special attention to the advanced search for potential contacts, using a distributed computing platform (*Section* 3). This will be demonstrated on two illustrative examples (*Section* 4) and supplied by brief concluding remarks with future research priorities (*Section* 5).

2. Physical and mathematical background

As a first model problem, let us consider a deformable body occupying a single domain Ω in the Euclidean space $\mathbb{R}^3$, supplied by a fixed Cartesian coordinate system $x = (x_1, x_2, x_3)$ for simplicity, with the Lipschitz boundary $\partial\Omega$, decomposed to disjoint parts Θ (for homogeneous Dirichlet boundary conditions) and Γ (for Neumann boundary conditions, inhomogeneous in general). The deformation of Ω will be analyzed on a finite time interval $\mathcal{I} = [0, T]$, T being a positive constant, i. e. for any time $t \in \mathcal{I}$. For any appropriate function ϕ we shall write $\phi_{,i}$ instead of $\partial\phi/\partial x_i$ with $i \in \{1, 2, 3\}$ and $\dot{\phi}$ instead of $\partial\phi/\partial t$ for brevity. The unit (formally outward) normal vector $\nu = (\nu_1, \nu_2, \nu_3)$ can be constructed (almost everywhere) on $\partial\Omega$. The standard notation of Lebesgue, Sobolev, Bochner - Sobolev, etc. function spaces following [24, Parts 1 and 7], will be applied here. The basic unknown variable $u(x, t)$, working with $x \in \Omega$ and $t \in \mathcal{I}$, introduced as the displacement of $x \in \Omega$, with possible extensions to Θ and Γ, in time $t \in \mathcal{I}$ related to the initial configuration at $t = 0$, can be considered as an element of $L^p(\mathcal{I}, V)$, with its first time derivative belonging to the same space and the second one (at least) to $L^2(\mathcal{I}, V^*)$. Here $V = \{w \in W^{1,p}(\Omega)^3 : \ w = o \text{ on } \Theta\}$ incorporates all body supports, V^* means the dual to V, o denotes the zero vector from $\mathbb{R}^3$ and $p, q \in [2, \infty)$ are some fixed exponents; satisfying $1/p + 1/q = 1$ (for all linearized formulations always $p = q = 2$). Let us notice that for any $w \in V$ we have (at least) $w \in L^6(\Omega)^3$, thanks to the Sobolev embedding theorem. Let us also introduce $X = L^q(\Omega)^3$ and $Z = L^q(\Gamma)^3$. To avoid technical difficulties, we shall make use of the results of [21, Parts 1.2 and 6.7], for elliptic (purely static) problems, referring to their natural generalization to hyperbolic (general dynamic) problems, thanks to the properties of Rothe sequences by [24, Part 7]. All detailed derivations must be left to the curious reader, due to the limited extent of this paper.

Let the pair of Cauchy initial conditions $u(0,.) = o$ and $\dot{u}(0,.) = \widehat{u}$ be introduced on Ω, $\widehat{u} \in V$ being some prescribed initial displacement rate. Let also the body forces $f \in L^q(\mathcal{I}, X)$ and the surface forces $g \in C_L(\mathcal{I}, Z)$ be given a priori, C_L referring to Lipschitz continuous functions on $\mathcal{I}$ (to avoid the difficulties with the properties of traces from V on $\partial\Omega$). Let $i, j, k \in \{1, 2, 3\}$ be the Einstein summation indices. Then the weak formulation of the conservation of linear momentum reads

$$(w_i, \rho\ddot{u}_i) + (w_{k,i}, \tau_{ik} + \alpha\dot{\tau}_{ik}) = (w_i, f_i) + \langle w_i, g_i\rangle \tag{1}$$

on $\mathcal{I}$ for any test function (virtual displacement) $w \in V$, However, the Piola stress tensor $\tau \in L^q(\Omega)^{3\times3}$ in (1) is still undefined and must be evaluated from an appropriate constitutive relation. Most frequently such relation uses the stress - strain dependence between the symmetric Kirchhoff stress tensor σ (its symmetry can be justified from the conservation of angular momentum, under the usual assumptions on Boltzmann continuum) introduced as $\tau_{ik} = \sigma_{ij}(\delta_{kj} + u_{k,j})$, with the help of the Kronecker symbol δ, and the Almansi strain tensor $\varepsilon_{ik}(u) = (u_{i,k} + u_{k,i} + u_{j,i}u_{j,k})/2$. Moreover, for appropriate functions φ and $\widetilde{\varphi}$, $(\varphi, \widetilde{\varphi})$ in (1) means the Lebesgue integral of $\varphi\widetilde{\varphi}$ over Ω and $\langle\varphi, \widetilde{\varphi}\rangle$ the similar Hausdorff integral over Γ; for $p = q = 2$ we can identify $(\varphi_i, \widetilde{\varphi}_i)$ and $\langle\varphi_i, \widetilde{\varphi}_i\rangle$ just with scalar products on X and Z. In (1) new positive material characteristics occur: $\rho \in L^\infty(\Omega)$ is the material density and $\alpha \in L^\infty(\Omega)$ introduces the structural damping factor, taking certain energy dissipation into account (because no closed physical systems occur in real applications).

The crucial choice for the practical implementation of (1) is the evaluation of σ from ε. Here we shall present only the empirical Hooke law for the isotropic case

$$\sigma_{ij} = \partial\Psi(\varepsilon)/\partial\varepsilon_{ij}\,, \qquad \Psi(\varepsilon) = \lambda_1\varepsilon_{kk}^2/2 + \lambda_2\varepsilon_{ij}\varepsilon_{ij}\,, \tag{2}$$

containing just two positive Lamé factors $\lambda_1, \lambda_2 \in L^\infty(\Omega)$ (or the Young modulus and the Poisson coefficient, derivable from them easily). Admitting the material anisotropy, most of our considerations with a generalized stored-energy function Ψ could be repeated, but with the duty to work with (up to) 21 independent material characteristics on Ω instead of two Lamé factors; the same can be valid even for a wider class of Ψ, introduced carefully, as discussed by [4]. Clearly the positive values of α on Ω in (1) upgrade this formulation to the parallel viscoelastic Kelvin model.

Unfortunately, the full procedure of verification of the existence and uniqueness of u satisfying (1) including (2), due to both Cauchy initial conditions, is not straightforward. For the time steps $t = sh$ with $s \in \{1, \ldots, m\}$, $h = T/m$, with the aim $m \to \infty$ in all convergence considerations, we are allowed to search for some $u_s \in V$ instead of the unknown $u(., sh)$ understanding $\tau(u_s)$ as the approximation of $\tau(u(., sh))$ by (2). Replacing $\dot{u}$ and $\ddot{u}$ by the first and second relative differences $\mathcal{D}u_s = (u_s - u_{s-1})/h$ and $\mathcal{D}^2u_s = (\mathcal{D}u_s - \mathcal{D}u_{s-1})/h$, (1) can be rewritten in the form

$$(w_i, \rho\mathcal{D}^2u_{is}) + (w_{k,i}, \tau_{iks} + \alpha\mathcal{D}\tau_{iks}) = (w_i, f_{is}) + \langle w_i, g_{is}\rangle \tag{3}$$

for any $w \in V$ again; f_{is} and g_{is} can be taken e. g. as the Clément quasi-interpolations of the components of f and g by [24, Part 7], together with $u_0 = o$ and $u_{-1} = -h\widehat{u}$. Thus we come, step by step, to some particular nonlinear elliptic equations, which should be solved iteratively, generating several types of Rothe sequences constructed i) as linear Lagrange splines on $\mathcal{I}$ using the values u_{is} and ii) as simple (piecewise constant) abstract functions using the same values, and iii) as time-retarded modifications of i) and ii) (to cover semi-linearization in iterative processes), whose convergence to u in a reasonable sense can be expected. However, the theoretical analysis of these equations needs some assumptions on polyconvexity (or quasiconvexity, etc.) for Ψ on Ω, together with the guarantee of mutual impenetrability of parts of Γ, which must be seen as nontrivial problems beyond the scope of this paper. Since all Rothe sequences i), ii), iii) are defined in infinite-dimensional function spaces, a finite element (or similar) technique is needed for most numerical evaluations.

As a second model problem, let us consider Ω as a union of a finite number of deformable bodies, whose frictionless contact is allowed now. Therefore three parts of $\partial\Omega$ must be distinguished in any time $t \in \mathcal{I}$, namely Θ, Γ and Λ where $\Lambda \subset \Gamma$ refers to all internal, adaptively activated interfaces; the lower index $_*$ will identify the integration over Λ (instead of Γ), the square brackets will be used for interface jumps of function values on Λ. Consequently (1) gets the form

$$(w_i, \rho\ddot{u}_i) + (w_{k,i}, \tau_{ik} + \alpha\dot{\tau}_{ik}) = (w_i, f_i) + \langle w_i, g_i\rangle + \langle [w_i\nu_i], \mathcal{T}\rangle_* \tag{4}$$

on $\mathcal{I}$ for any $v \in V$, containing the interface tractions $\mathcal{T} \in L^q(\Lambda)^3$ (not prescribed explicitly) replacing g_i on Γ by $g_i + \mathcal{T}\nu_i$ on Λ where $[u_i\nu_i] = 0$ is required. The activation and deactivation of Λ can be explained as the conversion of (4) to certain variational inequality of the Hertz - Signorini - Moreau type, as demonstrated by [33]. In some more details: in practical calculations, working with $u_\nu = u_i\nu_i$, we need $[u_i\nu_i]\,\mathcal{T} \le 0$ for all potential contacts (including both Λ and some adjacent parts of Γ) where always i) $[u_\nu] = 0$ and $\mathcal{T} \le 0$ (on Λ) or ii) $\mathcal{T} = 0$ and $[u_\nu] \le 0$ (outside Λ).

Following still [33], such formulation can be handled without the application of explicit inequalities, using the penalty approach. This approach admits some (sufficiently small) impacts, characterized by a positive part of $[u_\nu]_+$, suppressed by an artificial stiffness $\mathcal{K} \to \infty$ (constant frequently), occurring in one additional constitutive equation $\tau = \mathcal{K}[u_\nu]_+$. However, namely the searching for potential couples for all evaluations $[u_\nu]$ in arbitrary time $t \in \mathcal{I}$ can be seen as a serious numerical problem, exceeding the set of usual numerical methods for the analysis of differential equations, tending to the implementation of an appropriated distributed computing platform. Nevertheless, repeating the approach for a first model problem formally, (2) can be applied without any change and (3) has to be enriched by one right-hand-side additive term $\langle [w_{is}\nu_i], \mathcal{T}_s\rangle_*$ only.

3. Computational approach

Unfortunately the computational algorithm induced by the generalized version of (1) (including its above sketched generalization), applied to the study of convergence of Rothe sequences successfully, is not optimal for practical calculations. Thus we will sketch the derivation of a simple (nearly) explicit concurrent algorithm, up to the evaluation in certain finite-dimensional space. First, let us notice the unpleasant evaluations of sufficiently large discretized approximate values of u related to the reference configuration of Ω at $t = 0$; in this case the simple remedy is some adaptive setting of a new reference configuration after certain number of time steps, using some a posteriori estimates, relevant for the time development of Ω. The following task is then the full discretization of (1). For simplicity, as usual in the finite element method, let us consider some (at least weakly) regular decomposition of Ω to finite elements, using a set of n basis functions (with a small compact support, as derived e. g. from linear 3-dimensional Lagrange splines) $\{\phi_1, \dots, \phi_n\}$ from an n-dimensional space V^n approximating V (in particular, for conforming finite elements, from such subspace of V). We shall use the notation $\mathfrak{D}$ for a norm of such decomposition, e. g. that introduced as the largest diameter of a ball containing all applied finite elements, too; $\mathfrak{D} \to 0$ with $n \to \infty$ is expected.

Let us try to express $u(., t)$ at $t = sh$ by (3), using one more Einstein summation index $r \in \{1, \dots, n\}$ in its form $u_{is} = \mathbb{U}_{irs}\phi_r$ where $i \in \{1, 2, 3\}$ and $s \in \{1, \dots, m\}$ such that $\mathbb{U}_{irs}$ are, for simplicity, just the values approximating $u_i(x_r, sh)$ in some selected points x_r from Ω and Γ (including Λ); thus $\phi_r = 1$ for $x = x_r$, being zero-valued in all remaining cases. Thus the test functions are allowed to be $w_j = \mathbb{W}_{jr}\phi_r$ where $j \in \{1, 2, 3\}$, just with one non-zero value $\mathbb{W}_{jr}$ equal to 1. As the result we can compose the explicit time integration scheme, inspired by [11], in the form of a system of $3n$ seemingly linear algebraic equations

$$\mathbb{MA}_s = h^2\mathbb{F}_s + (h^2/\mathfrak{D})\mathbb{G}_s + (h^2/\mathfrak{D})\widetilde{\mathbb{G}}_s([\mathbb{U}_s])) - (h/\mathfrak{D}^2)\mathbb{C}(\mathbb{V}_s^\times) - (h^2/\mathfrak{D}^2)\mathbb{K}(\mathbb{U}_s)\,, \quad (5)$$

for $s \in \{0, 1, \dots, m\}$ supplied by the auxiliary formulae

$$\mathbb{V}_{s+1/2} = \mathbb{V}_{s-1/2} + h\mathbb{A}_s\,, \qquad \mathbb{V}_s = (\mathbb{V}_{s-1/2} + \mathbb{V}_{s+1/2})/2, \qquad \mathbb{U}_{s+1} = \mathbb{U}_s + h\mathbb{V}_{s+1/2} \quad (6)$$

(for $s = m$ without the last one) where $\mathbb{M}$ is a positive definite real symmetric sparse matrix of order $3n$ (or even a diagonal one, using the well-known lumped mass trick, working with the replacement of $\{\phi_1, \dots, \phi_n\}$ by simple functions where no differentiation is needed). All other symbols (except h and $\mathfrak{D}$) in (5) and (6) refer to vectors from $\mathbb{R}^{3n}$: $\mathbb{A}_s$, $\mathbb{V}_s$ and $\mathbb{U}_s$ approximate $\ddot{u}(., sh)$, $\dot{u}(., sh)$ and $u(., sh)$, $\mathbb{C}(.)$, $\mathbb{K}(.)$, $\mathbb{F}_s$, $\mathbb{G}_s$ and $\widetilde{\mathbb{G}}_s(.)$ are known a priori, $\mathbb{V}_s^\times \approx \mathbb{V}_s$ should be predicted as $\mathbb{V}_s^\times = \mathbb{V}_{s-1/2} + h(\mathbb{V}_{s-1/2} - \mathbb{V}_{s-1})/2$ for the first guess and corrected by iterations (if needed), $\mathbb{U}_0$ is zero-valued, $\mathbb{V}_0$ can be set using $\widehat{v}(x_r)$, $\mathbb{V}_{1/2} = \mathbb{V}_0 + h\mathbb{A}_0/2$ (to avoid undefined $\mathbb{V}_{-1/2}$ in the second formula of (6)).

In numerous papers written by engineers all considerations start with some discrete formulae like (5) and (6), continuing with their various modifications and al-

ternatives, which leads to the risk of misunderstanding with the language of mathematicians. Namely the common form of (5) is $\mathbb{MA}_s = \mathcal{F}_s^I + \mathcal{F}_s^E + \mathcal{F}_s^C$ where, as inherited from *Section* 2, according to the full discretization above, $\mathbb{MA}_s$ (as the complete left-hand side of (5)) represents the inertia forces, $\mathcal{F}_s^I$ the internal forces, expressed by the fourth and fifth right-hand-side additive terms of (5), $\mathcal{F}_s^E$ the external forces, expressed by its first and second additive terms, and $\mathcal{F}_s^C$ the contact forces, expressed by its third additive term, whose effective evaluation is the most delicate task. The following blocks of comments are motivated by the experience with the development of the prototype of the computational tool for the effective simulation of multiple contact of deformable bodies, applicable e. g. to crash testing in the automotive industry.

General approach. Our numerical approach should ensure all computations regardless of their environment, i. e. in sequential, parallel or hybrid manner, on a computer network. Each cluster node, considered in the hybrid form of computation, as suggested by [23], can be represented by some single workstation, which processes computation of a set of associated macro-entities; it can comprise a multi-core CPU capable of executing computational instructions in a fully parallel form. All computational procedures are activated within a global time loop. These computing cluster nodes are called worker nodes. The parallel and hybrid types of computations require the synchronization of CPU threads between individual dependent parts of the computation on each worker node. The synchronization is performed by means of barriers, supplied by some supportive processes. This mainly concerns the functionality focused on data exchange with the central server (master node) used in the hybrid type of computation compatible with [22] and [8]. The procedures themselves are called from another thread, focused purely on communication within the in scope of a computer network.

Contact analysis. The computational platform is assumed to deal with the node-to-segment type of contact in the sense of [33]. Due to its generality, the algorithm should be applicable to all finite elements of a model to find all finite element (FE) nodes suspected from the penetration of a finite element. A naive way to perform contact detection, checking each body against all other ones, ignoring any available information about the distribution of particular bodies in $\mathbb{R}^3$, has the very expensive time complexity $O(\mathcal{N}^2)$, $\mathcal{N}$ being the number of items in a dataset. A more suitable is offered by the nearest neighbour (NN) search, following [26]. The core of such algorithm is defined as a collection of $\mathcal{N}$ objects (FE element nodes); this builds a data structure which provides objects (FEs, their nodes, etc.) in the time as fast as possible, based on the NN query. Two levels of such analysis can be distinguished: i) search for penetration between bounding box volumes encapsulating individual macro-entities, and ii) search for contacts betweens FE nodes and individual FEs, using the node-to-segment approach. Even i) separately (as presented in both examples of *Section* 4) provides underlying support for the analysis of Macro Entity Interaction Multi-graph (MEIM, see lower) regarding the data redistribution within a computer cluster. The *kd*-tree data structure, utilized e. g. for machine learning

and for composition of graphical (gaming) engines, is able to provide an algorithmic support for both i) and ii).

Nearest neighbour search. Let us consider a given set S_p of points p in some high-dimensional real space. Our aim is to construct, to any query point q, a data structure able to find the point in S_p closest to q. Such NN problem belongs to a larger class of proximity problems investigated in computational geometry. Geometric range-searching data structures are constructed by subdividing $\mathbb{R}^3$ into several regions with some predefined properties and recursive generation of a data structure for each such region. Range queries are answered with such a data structure by performing a depth-first search through the resulting recursive space partition. Such data structure is created only once, until the development of situation (as of the FE-based approximate solutions by (5)) does not force its dynamical changes; this algorithm can be useful also for the MEIM analysis. The data structure used here is the k-dimensional tree (kd-tree), designed by [1] as a powerful extension of one-dimensional trees, i. e. the binary tree where the underlying space is partitioned using the value of just one attribute at each level of the tree, instead of all d attributes, unlike the quad-tree, introduced by [27], making such d-tests at each level. The basic analysis of kd-trees can be found in [20]; for its development see [25] and [32]. To compare other multi-dimensional data structures for spatial databases, cf. [19] for R-trees and their mutations, [2] and [14] for X-trees and their mutations and [36] for PH-tree.

Range search. An algorithm working with the kd-tree data structure consist i) of the assembly of a kd-tree map from the appropriate set S_p and ii) of its subsequent usage to obtain a set of nodes falling within the searching range query of any examined node belonging to an appropriate FE. Since the above sketched approach to contact detection can include the topology of the discretized model for the explicit time integration using (5) with (6), no algorithm for deletion of nodes or tree balancing algorithm are needed in i). In ii) we can traverse the kd-tree, but visit only nodes whose region is intersected by the query rectangle. If a region is fully contained in the query rectangle, we can report all the points stored in its sub-tree. When the traversal reaches a leaf, we have to check whether the point stored at the leaf is contained in the query region and, if so, report it.

Explicit integration scheme. Only one special type of FEs will be presented here for simplicity of numerical simulation of a massive impact process, namely the flat shell finite element with co-rotated coordinates of the Reissner - Mindlin type with linear fields for rotations and transverse deflections, developed by [29]. It is very effective in an explicit integration due to a smaller number of operations required for numerical integration (single quadrature point). Their geometrically non-linear behaviour was analyzed in [11] and [34] in details; its rate of convergence is approximately of quadratic order. The concrete form of explicit integration of FE forces $\mathcal{F}_s^I$, $\mathcal{F}_s^E$ and $\mathcal{F}_s^C$ for $s \in \{1, \ldots, m\}$, as required by (5), depends on the implementation of nonlinearities of various types; namely the approach of [34] expects the large rotational kinematics in the small strain regime. Such procedure is performed

by each hardware computational thread of a multi-core CPU; a specific number of FEs is assigned to each thread to optimize the thread load.

Distributed and parallel analysis. Two levels occur in the explicit FE analysis: i) standard process of both parallel integration of all FEs and explicit evaluation of (5), working with parallel integration of internal, external and contact forces, mapping FE ranges on the individual cores of a multi-core CPU, ii) parallel processing of MEIM on computer cluster, representing a distributed computational process able to run on a computer cluster within a cloud environment or some VPN (Virtual Private Network). The TCP/IP protocols enable the interprocess communication within clusters, with difficulties related to the CAP theorem (Consistency, Availability, Partition tolerance) by [5]; for its improvements cf. [6] and [35]. The data distribution for numerical computations is based on the domain decomposition (DD). From this class of methods we need to adopt the FE tearing and interconnecting (FETI), suggested by [10] and developed by [9], [17], [7] and [16] to (5) and (6); for many references (482 items) to particular variants of DD see [30], especially [30, Part 6.3] for the one-level FETI, [30, Part 6.4] for the dual-primal FETI ana [30, Part 8.5] for their applications to elasticity.

Advantages and drawbacks. In our approach each separate discretized domain is able to interact with its surroundings through the contact forces. All domains that come to contact then must be solved together within one worker node in a computer cluster. A large number of moving domains is represented by MEIM, whose edges are related to particular contacts. Such movement of domains is controlled, following [12], by the autonomous character of Lagrangians; this can control even the whole process of data distribution across the computer cluster. Nevertheless, the distributed applications, sketched here, suffer from a number of issues that need to be resolved gradually to reach an optimal model. Serious problems are: i) random application freezing, ii) model data migration between individual worker nodes in a computer cluster at runtime (input structural data, serialized contents of variables), iii) type of transferred data (unstructured vs. structured protocol) and iv) merging of data from individual worker nodes to get the final view on simulation results.

4. Illustrative example

The example presents the announced results for two benchmark problems, referring to the first and second model problem in *Section* 2. These results were obtained from the in-house software at BUT for the type of shells introduced in *Section* 3.

Fig. 1 shows the time development of contacts / impacts of elastic shells in selected time steps: i) for 1 big sphere falling to 1 fixed plane rectangle (3 upper graphs) and ii) for 10 small spheres thrown to 3 moving plane rectangles (3 lower graphs).

5. Conclusions

The aim of this paper was to show the possibility of effective computational analysis of contacts / impacts of deformable bodies for selected model problems, referring

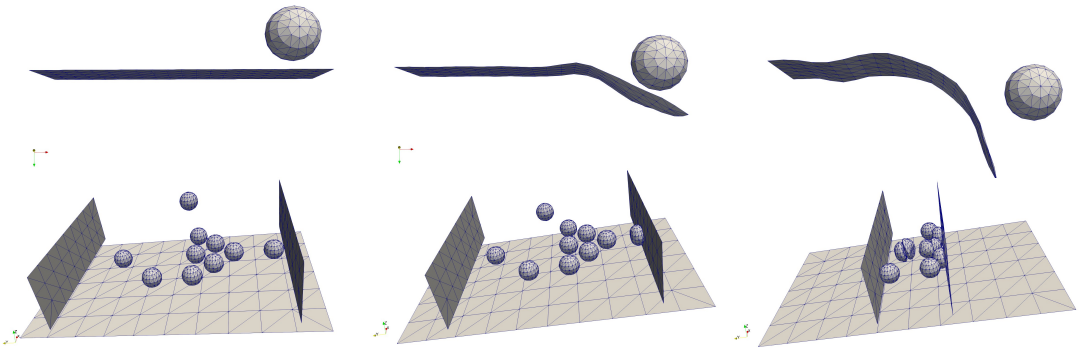


Figure 1: Example of time development of contacts / impacts of some elastic shells.

to still unclosed problems both in mathematical theory and in information science, too. Numerous improvements are required in distributed applications, as summarized at the end of *Section* 3. The upgrade of the explicit calculation scheme, coming from (5) with (6), could be inspired by the recent analyses of [13], [15] and [18].

For real engineering applications the next research step should be the careful revision of physical formulations in the scope of classical thermomechanics, together with the analysis of related mathematical and numerical problems, namely the proper study of energy dissipation on contacts, independently introduced by [28] and [31]. Such dissipation can be accompanied by the formation of plastic or microscopic damage zones, followed by the initiation and development of macroscopic cracks and further phenomena, dangerous for the bearing ability and durability of materials and structures.

Acknowledgements

This work was supported by the project of specific university research No. FAST-S-22-7867 at Brno University of Technology (BUT).

References

[1] Bentley, J. L.: Multidimensional binary search trees used for associative searching. *Commun. ACM* **18** (1975), pp. 509–517.

[2] Berchtold, S., Keim, D. A., and Kreigel, H.-P.: The X-tree: an index structure for high-dimensional data. In: *Proc. 23 VLDB (Int. Conf. on Very Large Databases)* in Bombay (1996), pp. 907–912. VLDB Endowment, Bombay, 1996.

[3] Bermúdez de Castro, A.: *Continuum Thermomechanics*. Birkhäuser, Basel, 2005.

[4] Bonet, J., Gil, A. J., and Ortigosa, R.: A computational framework for polyconvex large strain elasticity. *Comput. Methods Appl. Mech. Eng.* **283** (2015), pp. 1061–1094.

[5] Brewer, E. A.: Towards robust distributed systems. In: *Abstr. 19th PODC (Symposium on Principles of Distributed Computing)* in Portland (Oregon, USA), 2000, p. 7. Assoc. Computing Machinery, New York, 2000.

[6] Brewer, E. A.: CAP twelve years later: how the "rules" have changed. *IEEE Computer Magazine* **45** (2012), pp. 23–29.

[7] Cai, Y., Cui, X., Li, G., and Liu, W.: A parallel finite element procedure for contact-impact problems using edge-based smooth triangular element and GPU. *Int. J. Numer. Methods Eng.* **225** (2018), pp. 47–58.

[8] Celesti, A., Galletta, A., Fazio, M., and Villari, M.: Towards hybrid multi-cloud storage systems: understanding how to perform data transfer. *Big Data Res.* **16** (2019), pp. 1–17.

[9] Dostál, Z., Gomes Neto, F. A. M., and Santos, S. A.: Solution of contact problems by FETI domain decomposition with natural coarse space projections. *Comp. Methods Appl. Mech. Eng.* **190** (2000), pp. 1611–1627.

[10] Farhat, C., and Roux, F. X.: A method of finite element tearing and interconnecting and its parallel solution algorithm. *Int. J. Numer. Methods Eng.* **32** (1991), pp. 1205–1227.

[11] Halquist, J. O., et al.: *LS-DYNA Theoretical Manual*. Livermore Software Technology, Livermore (California), 2006.

[12] Har, J., and Tamma, K. K.: *Advances in Computational Dynamics of Particles, Materials and Structures*. J. Wiley & Sons, Hoboken, 2012.

[13] Jammy, S. P., Jacobs, Ch. T., and Sandham, N. D.: Performance evaluation of explicit finite difference algorithms with varying amounts of computational and memory intensity. *J. Comp. Sci.* **36** (2019), pp. 100565 / 1–7.

[14] Keim, D., Bustos, B., Berchtold, S., and Kriegel, H.-P.: Indexing, X-tree. In: *Encyclopedia of GIS* (Shekhar, S., Xiong, H., and Zhou, X., Eds.), pp. 543–547. Springer, Basel, 2017.

[15] Kim, W., and Reddy, J. N.: Novel explicit time integration schemes for efficient transient analyses of structural problems. *Int. J. Mech. Sci.* **172** (2020), pp. 105429 / 1–16.

[16] Kružík, J., Horák, A., Hapla, V., and Čermák, M.: Comparison of selected FETI coarse space projector implementation strategies. *Parallel Comput.* **93** (2020), pp. 102608 / 1–11.

[17] Kůs, P., and Šístek, J.: Coupling parallel adaptive mesh refinement with a nonoverlapping domain decomposition solver. *Adv. Eng. Softw.* **110** (2014), pp. 34–54.

[18] Liu, W., and Guo, W.: A novel predictor–corrector explicit integration scheme for structural dynamics. *Structures* **34** (2021), pp. 2735–2745.

[19] Manolopoulos, Y., Nanopoulos, A., Papadopoulos, A. N., and Theodoridis, Y.: *R-Trees: Theory and Applications*. Springer, London, 2006.

[20] Mehlhorn, K.: *Data Structures and Algorithms.* Springer, Berlin, 1984.

[21] Nečas, J.: *Introduction to the Theory of Nonlinear Elliptic Equations.* Teubner, Leipzig, 1983.

[22] Pääkkönen, P., and Pakkala, D.: Reference architecture and classification of technologies, products and services for big data systems. *Big Data Res.* **2** (2015), pp. 166–186.

[23] Rek, V., and Vala, J.: On a distributed computing platform for contact / impact of elastic bodies. In: *Proc. SNA (Seminar on Numerical Analysis)* 2021 in Ostrava (virtual), pp. 64–67. Inst. of Geonics CAS, Ostrava, 2021.

[24] Roubíček, T.: *Nonlinear Partial Differential Equations with Applications.* Birkhäuser, Basel, 2005.

[25] Rouquier, J., Alvarez, I., Reuillon, R., and Wuilleminet, P.-H.: A *kd*-tree algorithm to discover the boundary of a black box hypervolume. *Ann. Math. Artif. Intell.* **75** (2015), pp. 335–350.

[26] Sabharwal, Y, Sharma, N., and Sen, S.: Nearest neighbors search using point location in balls with applications to approximate Voronoi decompositions. *J. Comput. Syst. Sci.* **72** (2006), pp. 955–977.

[27] Sperber, M.: Quadtree and octree. In: *Encyclopedia of GIS* (Shekhar, S., Xiong, H., and Zhou, X., Eds.), pp. 1695–1700. Springer, Basel, 2017.

[28] Štekbauer, H., Němec, I., Lang, R., Burkart, D., and Vala, J.: On a new computational algorithm for impacts of elastic bodies. *Appl. Math.* **67** (2022), in print, 28 pp.

[29] Stolarski, H., Belytschko, T., Carpenter, N., and Kennedy, J. M.: A simple triangular curved shell element. *Eng. Comput.* **1** (1984), pp. 210–218.

[30] Toseli, A., and Widlund, O.: *Domain Decomposition Methods – Algorithms and Theory.* Springer, Berlin, 2005.

[31] Wang, G., Liu, C., and Liu, Y.: Energy dissipation analysis for elastoplastic contact and dynamic dashpot models. *Int. J. Mech. Sci.* **221** (2022), pp. 107214 / 1–14.

[32] Wehr, D., and Radkowski, R.: Parallel kd-tree construction on the GPU with an adaptive split and sort strategy. *Int. J. Parallel Program.* **46** (2018), pp. 1139–1156.

[33] Wu, S. R.: A variational principle for dynamic contact with large deformation. *Comput. Methods Appl. Mech. Eng.* **198** (2009), pp. 2009–2015, and **199** (2009), p. 220.

[34] Wu, S. R., and Gu, L.: *Introduction to the Explicit Finite Element Method for Nonlinear Transient Dynamics.* J. Wiley & Sons, Hoboken, 2012.

[35] Yuan, L.-Y., Wu, L., You, J.-H., and Shanghai, Y. Ch.: A demonstration of Rubato DB: a highly scalable NewSQL database system for OLTP and big data applications. In: *Proc. ACM SIGMOD Int. Conf. on Management of Data* in Melbourne (2015), pp. 907–912. Assoc. for Computing Machinery, New York, 2015.

[36] Zäschke, T., Zimmerli, Ch., and Norrie, M. C.: The PH-tree – a space-efficient storage structure and multi-dimensional index. In: *Proc. ACM SIGMOD Int. Conf. on Management of Data* in Snowbird (Utah, 2014), pp. 397–408. Assoc. for Computing Machinery, New York, 2014.

Programs and Algorithms of Numerical Mathematics 21
J. Chleboun, P. Kůs, J. Papež, M. Rozložník, K. Segeth, J. Šístek (Eds.)
Institute of Mathematics CAS, Prague 2023

INTERPOLATION WITH RESTRICTIONS — ROLE OF THE BOUNDARY CONDITIONS AND INDIVIDUAL RESTRICTIONS

Jan Valášek[1], Petr Sváček[2]

[1] Institute of Mathematics, Czech Academy of Sciences
Žitná 25, 115 67 Praha 1, Czech Republic
valasek@math.cas.cz
[2] Faculty of Mechanical Engineering, CTU in Prague
Karlovo nám. 13, Praha 2, 121 35, Czech Republic
petr.svacek@fs.cvut.cz

Abstract: The contribution deals with the remeshing procedure between two computational finite element meshes. The remeshing represented by the interpolation of an approximate solution onto a new mesh is needed in many applications like e.g. in aeroacoustics, here we are particularly interested in the numerical flow simulation of a gradual channel collapse connected with a severe deterioration of the computational mesh quality.
Since the classical Lagrangian projection from one mesh to another is a dissipative method not respecting conservation laws, a conservative interpolation method introducing constraints is described. The constraints have form of Lagrange multipliers enforcing conservation of desired flow quantities, like e.g. total fluid mass, flow kinetic energy or flow potential energy. Then the interpolation problem turns into an error minimization problem, such that the resulting quantities of proposed interpolation satisfy these physical properties while staying as close as possible to the results of Lagrangian interpolation in the L2 norm. The proposed interpolation scheme does not impose any restrictions on mesh generation process and it has a relatively low computational cost. The implementation details are discussed and test cases are shown.

Keywords: interpolation, Lagrange multiplier, Lagrange projection

MSC: 65D05, 65M60

1. Introduction

Interpolation is one of the basic mathematical problems and therefore there are plenty of available methods. Here we consider an interpolation procedure between two 2D computational finite element meshes involved during the remeshing step. This is a typical task in engineering simulations of cutting, forging, casting, welding,

DOI: 10.21136/panm.2022.26

(see e.g. [1]), where material is processed and reshaped, or in multiphysics simulations like geophysics, aeroacoustics or fluid-structure interaction (FSI), see e.g. [7]. Particularly, motivation for this paper is provided by the FSI problem of flow-induced vibrations of vocal folds studied in [9, 8]. The implemented ALE method during large vibrations was not able to provide a computational flow mesh of sufficient quality and thus the remeshing is needed, see [9].

As the base of the available interpolation methods the scattered data interpolations can be regarded. Such approaches are realized in many packages as e.g. Matlab, SciPy. Another possibility available also for higher dimensional cases and unstructed grids is the use of the radial basis function approach, see e.g. [6]. Further, there are methods specially suited for ALE methods, see e.g. [4]. However, they are designed for meshes with the same topology based on the computation of the local fluxes. Another approach is represented by so called supermesh approach, see e.g. [2], where a superior mesh given by mesh intersections is constructed what results in a high computational cost albeit it guarantees a L2 accurate projection. More computationally favourable approach of [1] replaces supermesh approach used together with Galerkin projection by an approximate evaluation of involved integrals, where a relative lack of precise intersection information should be compensated by increase of number of quadrature points. Nevertheless the most suitable method for our purpose is the idea of [5] to combine a cheap interpolation with supplementary restrictions typically chosen such that conservation of quantities from the physical nature of investigated problem is required. Let us call this approach as interpolation with restrictions or Codina & Pont interpolation (CPI). This method has a great advantage of satisfying physical laws (in global meaning) what is a typical disadvantage of other methods which results do not respect physical laws. Disadvantage is that restrictions, i.e. conservation of selected quantities, are not valid locally.

Thus we will further deal only with the interpolation with restrictions, see [5], and we will focus on behaviour of this method near domain boundaries. Our aim is to improve CPI by using further information from boundaries, i.e. we assume that new target FE mesh occupies the same space as the old donor FE mesh and further that the vertex locations of the old and the new mesh on the mesh boundaries are identical. This assumption is motivated by implementation of our in-house FSI solver, [9, 8]. Then two methods of boundaries values treatment are compared and the interpolation error for case of small highly distorted domain contrary to case of larger domain with smaller distortion is calculated (motivated by different settings during construction of ALE mapping).

The structure of the paper is following. First the interpolation with restrictions is described and applied for the case of fluid flow. Further the implementaion details are presented. Finally the errors of different interpolation settings are analyzed and summarized in conclusion.

2. Interpolation with restrictions

In the whole paper we consider two triangulations $\mathcal{T}^{\mathrm{o}}$ and $\mathcal{T}^{\mathrm{n}}$ of the same bounded physical domain Ω of $\mathbb{R}^2$, see Figure 1, which moreover satisfy that their boundary vertices are identical. Here, $\mathcal{T}^{\mathrm{o}}$ is called the old (donor) mesh and $\mathcal{T}^{\mathrm{n}}$ is the new (target) mesh. By $\mathcal{V}_h^{\mathrm{o}}$ and $\mathcal{V}_h^{\mathrm{n}}$ the corresponding FE spaces constructed over the triangulations $\mathcal{T}^{\mathrm{o}}$ and $\mathcal{T}^{\mathrm{n}}$ are denoted, respectively. Further, we denote a FE function from FE space $\mathcal{V}_h^{\mathrm{o}}$ constructed over the FE mesh $\mathcal{T}^{\mathrm{o}}$ by u_h^{o}, i.e. $u_h^{\mathrm{o}}(x) = \sum_j U_j^{\mathrm{o}} \psi_j^{\mathrm{o}}(x)$, where $\psi_j^{\mathrm{o}}(x)$ are basis functions of the FE space $\mathcal{V}_h^{\mathrm{o}}$ and U_j^{o} are corresponding nodal values. Similarly, a function from $\mathcal{V}_h^{\mathrm{n}}$ can be written as $u_h^{\mathrm{n}}(x) = \sum_k U_k^{\mathrm{n}} \psi_k^{\mathrm{n}}(x) \in \mathcal{V}_h^{\mathrm{n}}$.

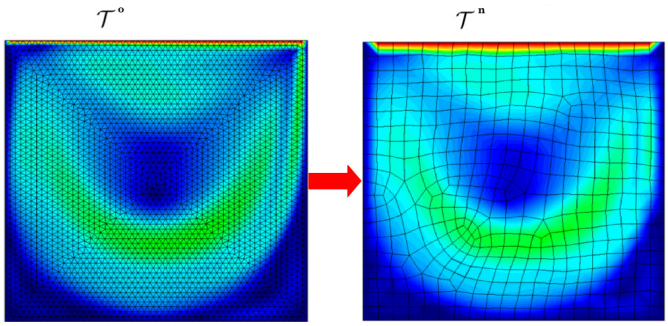


Figure 1: Illustration of interpolation from old to new FE mesh, [5].

2.1. Key idea of the method

The general procedure of interpolation with restrictions, see [5], is based on two steps. During the first step a function $u_h^{\mathrm{o}} \in \mathcal{V}_h^{\mathrm{o}}$ defined on the old mesh $\mathcal{T}^{\mathrm{o}}$ is projected on the new mesh. The commonly used projections are either Lagrange or Galerkin projections, [5]. The first one, the Lagrange projection, is based on the evaluation of the values U_A^{n} given by

$$U_A^{\mathrm{n}} = u_h^{\mathrm{n}}(X_A^{\mathrm{n}}) = \sum_j U_j^{\mathrm{o}} \, \psi_j^{\mathrm{o}}(X_A^{\mathrm{n}}), \tag{1}$$

where X_A^{n} denotes the coordinates of the point associated with the nodal value U_A^{n}. In the second (Galerkin) case, the L2 projection is applied leading to the integral identity

$$\int_\Omega u_h^{\mathrm{n}} \, \psi^{\mathrm{n}} \, \mathrm{d}x = \int_\Omega u_h^{\mathrm{o}} \, \psi^{\mathrm{n}} \, \mathrm{d}x \quad \forall \psi^{\mathrm{n}} \in \mathcal{V}_h^{\mathrm{n}}. \tag{2}$$

In order to precisely fulfill (2) one needs to compute elements intersections. Such a procedure can be computationally demanding and requires additional techniques to be applied as e.g. supermesh approach used in [2]. High computational costs of the Galerkin approach can be reportedly reduced by using numerical quadrature of high orders, see [1]. As this phenomenon was not observed for the considered numerical tests, the use of the Lagrangian interpolation is preferred.

As one of the biggest interpolation problems is the violation of physical nature of interpolated variable, see e.g. [4], in the second step appropriate restrictions are applied as a correction step of the projection. The idea of imposing additional restrictions with the help of Lagrangian multipliers is a key how to conserve quantities of the interest (in global sense). The presented two steps of the CPI algorithm is general and it can be potentially used in many different scenarios, [5]. The disadvantage of CPI is that local conservation of desired quantities is not guaranteed.

2.2. Application to fluid flow problem

The previous general concept is now applied for incompressible fluid flow problem with the constant density ρ. In this context we will use following notation: $\mathbf{v}^{\mathrm{o}} \in \mathbf{V}_h^{\mathrm{o}} = \mathcal{V}_h^{\mathrm{o}} \times \mathcal{V}_h^{\mathrm{o}}$ for the given velocity defined on the old mesh $\mathcal{T}^{\mathrm{o}}$, $\widetilde{\mathbf{v}}^{\mathrm{n}} = \Pi_h \mathbf{v}^{\mathrm{o}} \in \mathbf{V}_h^{\mathrm{n}}$ for the Lagrangian projection of $\mathbf{v}^{\mathrm{o}}$ on the new mesh $\mathcal{T}^{\mathrm{n}}$ and $\mathbf{v}^{\mathrm{n}}$ for the sought interpolation with restrictions on the target mesh $\mathcal{T}^{\mathrm{n}}$. The interpolation procedure is now described.

Based on the nature of the problem we impose conservation of the following quantities: 1) mass (through the conservation of the velocity divergence), 2) linear momenta and 3) kinetic energy. This leads to the following four restrictions:

$$\begin{aligned} &1)\ \int_\Omega \nabla\cdot\mathbf{v}^{\mathrm{o}}\,\mathrm{d}x = \int_\Omega \nabla\cdot\mathbf{v}^{\mathrm{n}}\,\mathrm{d}x, \qquad 2)\ \int_\Omega \rho\mathbf{v}^{\mathrm{o}}\cdot\mathbf{e}_i\,\mathrm{d}x = \int_\Omega \rho\mathbf{v}^{\mathrm{n}}\cdot\mathbf{e}_i\,\mathrm{d}x, \quad i=\{1,2\},\\ &3)\ \frac{1}{2}\int_\Omega \rho|\mathbf{v}^{\mathrm{o}}|^2\,\mathrm{d}x = \frac{1}{2}\int_\Omega \rho|\mathbf{v}^{\mathrm{n}}|^2\,\mathrm{d}x, \end{aligned} \tag{3}$$

where vectors $\mathbf{e}_i$ denotes standard basis. In what follows we set $\rho = 1$.

Then the problem of interpolation with restrictions reads: For the given velocity $\mathbf{v}^{\mathrm{o}} \in \mathbf{V}_h^{\mathrm{o}}$ find

$$[\mathbf{v}^{\mathrm{n}}, \boldsymbol{\lambda}] = \arg \inf_{\mathbf{u}^{\mathrm{n}}\in\mathbf{V}_h^{\mathrm{n}}} \sup_{\boldsymbol{\mu}\in\mathbb{R}^4} L(\mathbf{u}^{\mathrm{n}}, \boldsymbol{\mu}), \tag{4}$$

where $\boldsymbol{\mu}$ are Lagrangian multipliers and $L(\mathbf{u}^{\mathrm{n}}, \boldsymbol{\mu})$ is Lagrangian function defined as

$$\begin{aligned} L(\mathbf{u}^{\mathrm{n}}, \boldsymbol{\mu}) =& \frac{1}{2}\int_\Omega \left(\sum_k (U_k^{\mathrm{n}} - \widetilde{U}_k^{\mathrm{n}})\,\boldsymbol{\psi}_k^{\mathrm{n}}\right)^2 \mathrm{d}x - \mu_1 \int_\Omega \nabla\cdot\left(\sum_k U_k^{\mathrm{n}}\,\boldsymbol{\psi}_k^{\mathrm{n}} - \sum_j U_j^{\mathrm{o}}\,\boldsymbol{\psi}_j^{\mathrm{o}}\right)\mathrm{d}x \\ &- \sum_{l=1}^{2} \mu_l \int_\Omega \left(\sum_k U_k^{\mathrm{n}}\,\boldsymbol{\psi}_k^{\mathrm{n}} - \sum_j U_j^{\mathrm{o}}\,\boldsymbol{\psi}_j^{\mathrm{o}}\right)\cdot\mathbf{e}_l\,\mathrm{d}x \\ &- \frac{\mu_4}{2}\int_\Omega \left(\sum_k U_k^{\mathrm{n}}\,\boldsymbol{\psi}_k^{\mathrm{n}}\right)^2 - \left(\sum_j U_j^{\mathrm{o}}\,\boldsymbol{\psi}_j^{\mathrm{o}}\right)^2 \mathrm{d}x. \end{aligned} \tag{5}$$

The differentiation of the function L with respect to all unknowns U_i^{n} yields

$$\int_\Omega \sum_k U_k^{\mathrm{n}} \boldsymbol{\psi}_k^{\mathrm{n}} \boldsymbol{\psi}_i^{\mathrm{n}} \,\mathrm{d}x - \mu_1 \int_\Omega \nabla \cdot \boldsymbol{\psi}_i^{\mathrm{n}} \,\mathrm{d}x - \sum_{l=1}^{2} \mu_l \int_\Omega \boldsymbol{\psi}_i^{\mathrm{n}} \,\mathrm{d}x \\ - \mu_4 \int_\Omega \sum_k U_k^{\mathrm{n}} \boldsymbol{\psi}_k^{\mathrm{n}} \boldsymbol{\psi}_i^{\mathrm{n}} \,\mathrm{d}x = \int_\Omega \sum_k \widetilde{U}_k^{\mathrm{n}} \boldsymbol{\psi}_k^{\mathrm{n}} \boldsymbol{\psi}_i^{\mathrm{n}} \,\mathrm{d}x, \tag{6}$$

and by the differentiation of L with respect to μ_i together with the condition given by Eq. (4) we get

$$\begin{aligned} \int_\Omega \nabla \cdot \left(\sum_k U_k^{\mathrm{n}} \, \boldsymbol{\psi}_k^{\mathrm{n}} \right) \,\mathrm{d}x &= \int_\Omega \nabla \cdot \left(\sum_j U_j^{\mathrm{o}} \, \boldsymbol{\psi}_j^{\mathrm{o}} \right) \,\mathrm{d}x, \\ \int_\Omega \left(\sum_k U_k^{\mathrm{n}} \, \boldsymbol{\psi}_k^{\mathrm{n}} \cdot \mathbf{e}_l \right) \,\mathrm{d}x &= \int_\Omega \left(\sum_j U_j^{\mathrm{o}} \, \boldsymbol{\psi}_j^{\mathrm{o}} \cdot \mathbf{e}_l \right) \,\mathrm{d}x, \quad l = \{1, 2\}, \\ \int_\Omega \left(\sum_k U_k^{\mathrm{n}} \, \boldsymbol{\psi}_k^{\mathrm{n}} \right)^2 \,\mathrm{d}x &= \int_\Omega \left(\sum_j U_j^{\mathrm{o}} \, \boldsymbol{\psi}_j^{\mathrm{o}} \right)^2 \,\mathrm{d}x. \end{aligned} \tag{7}$$

Previous equations written in the matrix notation reads

$$\begin{pmatrix} \mathbb{M}^{\mathrm{n}} & -R_1 & -R_2 & -R_3 & -\mathbb{M}^{\mathrm{n}}\mathbf{U}^n \\ R_1^T & 0 & 0 & 0 & 0 \\ R_2^T & 0 & 0 & 0 & 0 \\ R_3^T & 0 & 0 & 0 & 0 \\ (\mathbb{M}^{\mathrm{n}}\mathbf{U}^n)^T & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{U}^n \\ \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \end{pmatrix} = \begin{pmatrix} \mathbb{M}^{\mathrm{n}}\widetilde{\mathbf{U}}^n \\ R_1^{\mathrm{o}}\mathbf{U}^{\mathrm{o}} \\ R_2^{\mathrm{o}}\mathbf{U}^{\mathrm{o}} \\ R_3^{\mathrm{o}}\mathbf{U}^{\mathrm{o}} \\ (\mathbf{U}^{\mathrm{o}})^T \, \mathbb{M}^{\mathrm{o}}\mathbf{U}^{\mathrm{o}} \end{pmatrix}, \tag{8}$$

where $\mathbb{M}^{\mathrm{n}}$ denotes mass matrix with components $m_{ij}^{\mathrm{n}} = \int_\Omega \boldsymbol{\psi}_j^{\mathrm{n}} \boldsymbol{\psi}_i^{\mathrm{n}} \,\mathrm{d}x$, $\mathbb{M}^{\mathrm{o}}$ is the mass matrix defined on the old mesh $\mathcal{T}^{\mathrm{o}}$ and vectors R_1, R_2, R_3 are given componentwise by

$$(R_1)_i = \int_\Omega \nabla \cdot \boldsymbol{\psi}_i^{\mathrm{n}} \,\mathrm{d}x, \qquad (R_2)_i = \int_\Omega \boldsymbol{\psi}_i^{\mathrm{n}} \cdot \mathbf{e}_1 \,\mathrm{d}x, \qquad (R_3)_i = \int_\Omega \boldsymbol{\psi}_i^{\mathrm{n}} \cdot \mathbf{e}_2 \,\mathrm{d}x. \tag{9}$$

Vectors R_i^{o}, $i \in \{1, 2, 3\}$ are defined similarly on the old mesh. Since problem (8) is nonlinear the Newton-Rhapson method is used for its numerical solution, see [5].

Pressure. The same concept is also used for the interpolation of the pressure obtained by the solution of the Navier-Stokes equations. In this case only the conservation of its L2 norm is considered.

3. Implementation

Although problem (8) has a saddle point structure the most computationally demanding part is the computation of the Lagrange projection. It is due to the

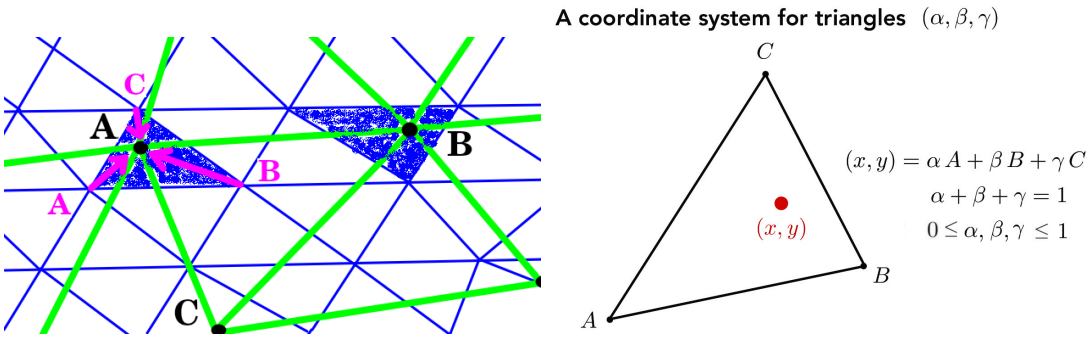


Figure 2: Illustration of Lagrange interpolation from the old FE mesh $\mathcal{T}^{\mathrm{o}}$ (blue) to the new FE mesh $\mathcal{T}^{\mathrm{n}}$ (green) with its vertices plotted in black colour. The filled blue triangles highlight the area, from which the final value in vertices A and B are computed.

necessity to find locations of the vertices X_k^{n} from $\mathcal{T}^{\mathrm{n}}$ in terms of the old mesh $\mathcal{T}^{\mathrm{o}}$ in order to evaluate $\boldsymbol{\psi}^{\mathrm{o}}(X_k^{\mathrm{n}})$ in Eq. (1). The way towards it is to determine at which triangles from $\mathcal{T}^{\mathrm{o}}$ points X_k^{n} lie and to find their barycentric coordinates inside these triangles, see Fig. 2. Then evaluation of Eq. (1) is straightforward.

There are more possible methods how to find such locations. In the work of [5] the octree parallel algorithm was employed, another possibility offers advancing front techniques, see e.g. [3]. Nevertheless here we adopted the procedure based on the computation of barycentric coordinates as it is implemented in software Octave. The algorithm is following: First prepare the list $\mathcal{X}$ of vertices X_k^{n} of $\mathcal{T}^{\mathrm{n}}$. Then in a loop over all triangles $T_i^{\mathrm{o}} \in \mathcal{T}^{\mathrm{o}}$ determine which points from the list $\mathcal{X}$ lie in T_i^{o}:

1. Compute the barycentric coordinates $\alpha_j, \beta_j, \gamma_j$ for each $X_j^{\mathrm{n}} \in \mathcal{X}$ by solving 3x3 matrix system with M right hand vectors, where M is the length of list $\mathcal{X}$.
2. If $0 \le \alpha_j, \beta_j, \gamma_j \le 1$ and $\alpha_j + \beta_j + \gamma_j = 1$ then point X_j^{n} belongs to triangle T_i^{o}. Save its barycentric coordinates and shorten list $\mathcal{X}$.

Complexity of this approach is almost quadratic, on the other hand this procedure can be well parallelized. Further, a division of list $\mathcal{X}$ in short sub-list based e.g. on conditions $x \gtrless x_0,\ y \gtrless 0$ can speed up the algorithm.

4. Numerical simulations

Two tests of the interpolation with restrictions are performed.

4.1. First interpolation test – question of boundary values

The modified interpolation test of [1, 5] shows how an additional information from the boundary can improve the interpolation results. Let have a divergence-free function $\mathbf{F}(x, y)$ with the components $f_1(x, y) = 2x^2(x-1)^2y(y-1)(2y-1),\ f_2(x, y) = -2y^2(y-1)^2x(x-1)(2x-1)$ and the donor and the target triangular meshes of the

domain $\langle 0, 1.1\rangle^2$ with identical vertices at the boundary. Both meshes has the characteristic length $h = 0.033$ and the inner vertices of the target mesh are shifted by $h/2$ to the right. In total 20 pairs of interpolations between these meshes are performed and four different interpolation variants are compared. The first two are the classical Lagrange projection (LAG) and the interpolation with restrictions (CPI). Further two are the CPI modifications: by CPI_m the variant, where the known values at boundary vertices are eliminated from the final matrix system (8), is denoted. The CPI_bv denotes the CPI variant, where the results of system (8) are at the positions related to the boundary vertices overwritten by the known values.

Figure 3 shows the velocity magnitude after all 20 interpolation runs from the original to the target mesh and back. It is evident that the Lagrange projection performs badly and it is too diffusive. The results of the interpolations CPI_m, CPI_bv (not shown) and CPI are very similar each to the other as well to the original data. The behaviour of interpolations along two lines are shown in Figure 4. In the case of the top domain boundary only the CPI results do not correspond to the exact ones because other CPI variants benefit from the additional information at the boundary. The CPI behaviour along the middle line is the same as the CPI_bv and the CPI_m is even slightly closer to the exact values than CPI, the LAG results are the worst.

From the quantitative point of view the L_2 error of the Lagrange projection is higher by 38%, while both the CPI modifications outperforms the original by 13% (CPI_bv) and by 11% (CPI_m), respectively, see Table 1. The L_∞ error is for the considered interpolation methods similar. Nevertheless the disadvantage of the CPI_bv method is the violation of the conservation of the kinetic energy. This happens due to the modification of the CPI solution[1] at the positions related to the boundary vertices contrary to the CPI_m variant, where the matrix system is modified rather than the individual values of interpolation result. Consequently the CPI_m provides a very precise kinetic energy conservation. Thus better choice appears to be the CPI_m than the CPI_bv, the CPI interpolation performs also reasonably well.

method	max $\lvert\mathbf{F}\rvert$	E_{kin}	L_2 error	L_∞ error
exact	$1.650 \cdot 10^{-2}$	$6.657 \cdot 10^{-5}$	0	0
Lagrange int.	$1.650 \cdot 10^{-2}$	$5.306 \cdot 10^{-5}$	$2.343 \cdot 10^{-6}$	$3.489 \cdot 10^{-3}$
CPI	$1.848 \cdot 10^{-2}$	$6.657 \cdot 10^{-5}$	$1.697 \cdot 10^{-6}$	$3.499 \cdot 10^{-3}$
CPI_bv	$1.650 \cdot 10^{-2}$	$6.592 \cdot 10^{-5}$	$1.436 \cdot 10^{-6}$	$3.211 \cdot 10^{-3}$
CPI_m	$1.650 \cdot 10^{-2}$	$6.657 \cdot 10^{-5}$	$1.520 \cdot 10^{-6}$	$3.221 \cdot 10^{-3}$

Table 1: Comparison of interpolation results of the first test.

[1]The CPI interpolation preserves kinetic energy.

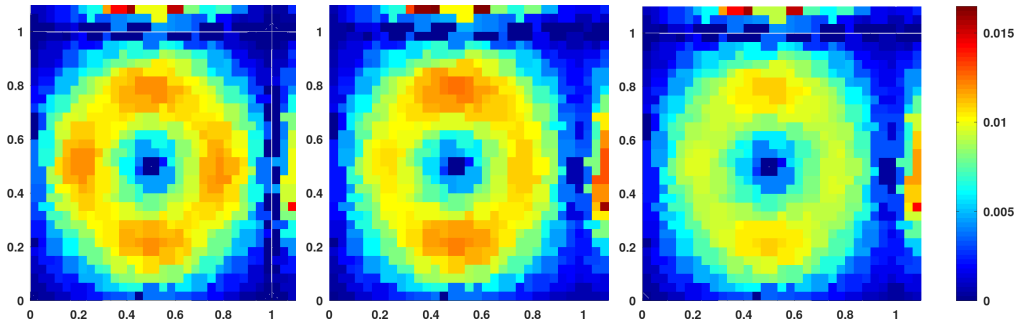


Figure 3: Magnitude of the interpolated vector field on the structured FE mesh after 20 runs. The exact values are shown on the left, the result of the interpolation with restrictions in the middle and the Lagrange interpolation on the right.

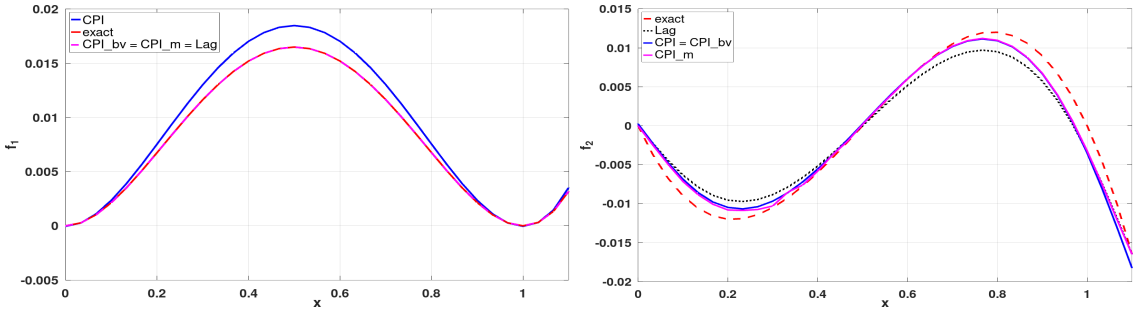


Figure 4: **Left:** Comparison of the interpolation of the first component of the velocity along the top boundary given by $y = 1.1, x \in \langle 0, 1.1 \rangle$. **Right:** Comparison of the second component of the velocity along line given by $y = 0.5, x \in \langle 0, 1.1 \rangle$.

4.2. Second interpolation test – question of interpolation domain

In the second test the interpolation results for different choices of interpolation domain are compared using an additional assumption of the following correspondence between the donor and the target mesh: The difference of the new target against the original mesh is the coarsened middle part around a channel constriction, see Figure 5, where the remaining parts of the target mesh are identical with the original one. Such mesh coarsening is motivated by the usage of our in-house solver FSIfem based on the ALE method (see [9]) in order to avoid deterioration of fluid mesh quality during simulations involving (almost complete) channel closing. Since the target domain of the ALE mapping can be chosen in the FSIfem solver we compare CPI interpolation on the following choices of the interpolation subdomains of the computational domain, see Fig. 5, with the aim to decrease interpolation error:

1. only the middle part of the constriction (CPI_sel)
2. the whole domain, but with the coincident mesh vertices outside of the middle part (CPI_coi)
3. the whole domain (here slightly shifted mesh vertices are used) (CPI_all).

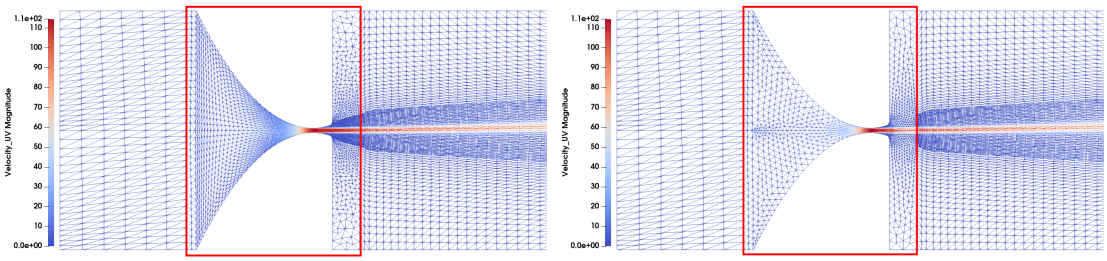


Figure 5: Meshes used in the second test together with initial and interpolated velocity distributions. The donor (original) mesh is shown on left and the target mesh on the right, only the middle part of both meshes differs and it is highlighted by the red square.

Here, the considered velocity field, which is obtained by FSIfem as part of the FSI solution, is once interpolated from the donor to the target mesh and vice versa.

Figure 6 illustrates the distribution of error after one pair of the interpolation runs. The results of the interpolations CPI_sel and CPI_coi are very close, while the error of CPI_all is a little higher. Moreover the error of CPI_all is distributed also significantly in the area right from the channel constriction contrary to the CPI_sel and CPI_coi results. The relative high interpolation error in the boundary layer is caused by the coarse target mesh at the region. In the case with a similarly dense target mesh the interpolation error can be expected to be significantly lower.

Interpolations obtained by CPI_sel and CPI_coi have similar L_2 and L_∞ errors, see Table 2, however slightly smaller L_2 error of CPI_sel is redeemed by the inconsistency in the maximal value and in the total kinetic energy. The CPI_all presents the largest L_2 and L_∞ errors.

method	max $\lvert\mathbf{F}\rvert$	E_{kin}	L_2 error	L_∞ error
exact	$1.142 \cdot 10^2$	$9.644 \cdot 10^{-2}$	0	0
Lagrange int.	$1.141 \cdot 10^2$	$9.580 \cdot 10^{-2}$	$9.914 \cdot 10^{-5}$	$3.512 \cdot 10^1$
CPI_sel	$1.184 \cdot 10^2$	$9.646 \cdot 10^{-2}$	$9.086 \cdot 10^{-5}$	$3.369 \cdot 10^1$
CPI_coi	$1.145 \cdot 10^2$	$9.644 \cdot 10^{-2}$	$9.838 \cdot 10^{-5}$	$3.498 \cdot 10^1$
CPI_all	$1.153 \cdot 10^2$	$9.644 \cdot 10^{-2}$	$1.323 \cdot 10^{-4}$	$3.472 \cdot 10^1$

Table 2: Comparison of interpolation results of the second test.

5. Conclusion

The article presents the general concept of the interpolation between FEM meshes based on paper [5]. The idea of the interpolation with restrictions is to improve the commonly available interpolation procedure by a restriction of the conservation of additional physical quantities. Such approach has the advantage of the relatively

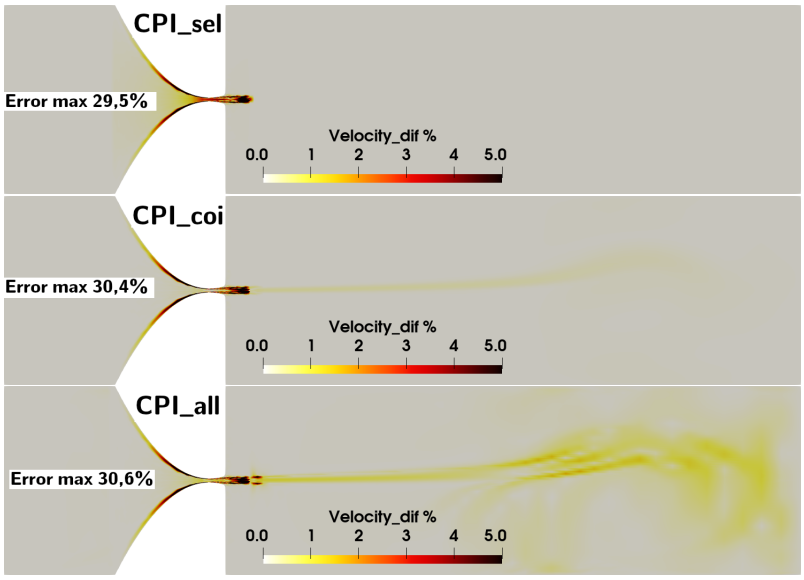


Figure 6: Distributions of relative error magnitudes after one pair of interpolations. Results for CPI_sel on the top, for CPI_coi in the middle and error of the CPI_all interpolation on the bottom. The maximal error is located for all methods similarly in a few elements inside boundary layer (out of color scale).

computationally cheap method which moreover respects physical nature of the problem. Here for the case of the fluid flow the conservation of the linear momenta, the divergence and the kinetic energy is considered. Our implementation based on the barycentric coordinates is described.

Two interpolation tests are performed in order to compare different modifications of this method motivated by the different settings of the FSI solver based on the ALE method. First, the different treatments of the nonzero boundary values are studied. The interpolation results can be slightly improved by inclusion of the information from the boundary into the resulting matrix system (variant CPI_m). Second, the interpolation error is calculated for different choices of the interpolation domain. The best results are obtained for the interpolation in the whole domain where the mesh distortion is highly localized around the channel constriction (variant CPI_coi). The interpolation of only the distorted part of the domain violates the conservation of the total kinetic energy in the whole domain.

Acknowledgements

The work was supported from European Regional Development Fund – Project "Center for Advanced Applied Science" (No.CZ.02.1.01/0.0/0.0/16-019/0000778) and from Premium Academiae of Prof. Nečasová.

References

[1] Bussetta, P., Boman, R., and Ponthot, J.P.: Efficient 3D data transfer operators based on numerical integration. Internat. J. Numer. Methods Engrg. **102** (2015), 892–929.

[2] Farrell, P.E. and Maddison, J.R.: Conservative interpolation between volume meshes by local galerkin projection. Comput. Methods Appl. Mech. Engrg. **200** (2011), 89–100.

[3] Gander, M.J. and Japhet, C.: An algorithm for non-matching grid projections with linear complexity. In: *Proc. of 18th Intern. Conf. on Domain Decomposition Methods*. Springer, 2009.

[4] Klíma, M., Kuchařík, M., and Shashkov, M.: Combined swept region and intersection-based single-material remapping method. Internat. J. Numer. Methods Fluids **85** (2017), 363–382.

[5] Pont, A., Codina, R., and Baiges, J.: Interpolation with restrictions between finite element meshes for flow problems in an ALE setting. Internat. J. Numer. Methods Engrg. **110** (2017), 1203–1226.

[6] Schoder, S., Roppert, K., Weitz, M., Junger, C., and Kaltenbacher, M.: Aeroacoustic source term computation based on radial basis functions. Internat. J. Numer. Methods Engrg. (2019).

[7] Schoder, S. et al.: Application limits of conservative source interpolation methods using a low Mach number hybrid aeroacoustic workflow. J. Theor. Comput. Acoust. **29** (2021), 2050 032.

[8] Sváček, P. and Horáček, J.: FE numerical simulation of incompressible airflow in the glottal channel periodically closed by self-sustained vocal folds vibration. J. Comput. Appl. Math. **393** (2021), 113 529.

[9] Valášek, J., Sváček, P., and Horáček, J.: On suitable inlet boundary conditions for fluid-structure interaction problems in a channel. Appl. Math. **64** (2019), 225–251.

Programs and Algorithms of Numerical Mathematics 21
J. Chleboun, P. Kůs, J. Papež, M. Rozložník, K. Segeth, J. Šístek (Eds.)
Institute of Mathematics CAS, Prague 2023

IMPROVED FLUX RECONSTRUCTIONS IN ONE DIMENSION

Miloslav Vlasák, Jan Lamač

Faculty of Civil Engineering, Czech Technical University
Thakurova 7, 166 29 Prague 6, Czech Republic
miloslav.vlasak@cvut.cz, jan.lamac@cvut.cz

Abstract: We present an improvement to the direct flux reconstruction technique for equilibrated flux a posteriori error estimates for one-dimensional problems. The verification of the suggested reconstruction is provided by numerical experiments.

Keywords: a posteriori error estimates, flux reconstructions, numerical experiments

MSC: 65N15, 65N30

1. Introduction

A posteriori error estimates play an important role in the numerical solution of PDEs. They enable to provide the information about the discretization error for the current choice of discretization parameters and also enable localization of the sources of errors that can be exploited in possible adaptive strategies. For the survey of main a posteriori techniques for PDE discretizations see e.g.[1], [3], [7], [12], [14] and references cited therein.

Important class of approaches for deriving guaranteed a posteriori upper bounds is based on the Hyper-circle theorem, see [11]. This theorem assumes the reconstruction of the fluxes to be in $H(\mathrm{div})$. Such a property can be gained by global procedures that are very accurate but also very expensive, see e.g. [12]. Among the local procedures, the local mixed finite element technique is very popular, since it enables to reconstruct the fluxes based on local, relatively cheap problems. The theoretical results devoted to these mixed finite element reconstructions can be found in [5] and [9]. The paper [15] presents even more simple, more direct and cheaper reconstructions based on the natural degrees of freedom for the Raviart-Thomas spaces inspired by [8], where a similar idea is applied to the discontinuous Galerkin discretizations.

Although a posteriori error estimates based on the direct evaluation presented in [15] are reliable and robust, their accuracy gets slightly worse in some situations, especially for even degree polynomial approximations. The reason behind this

DOI: 10.21136/panm.2022.27

behavior may possibly come from a rather naive choice how to define the flux reconstructions on the boundary of elements. Therefore, we present a suggestion for an improvement of the definition of the flux reconstruction on the boundary of elements for one-dimensional problems. Our suggestion is supported with numerical experiments.

2. Continuous problem and its discretization

2.1. Continuous problem

Let $\Omega \subset \mathbb{R}^d$ be a bounded polyhedral domain with Lipschitz continuous boundary $\partial\Omega$. Most of the presented results hold true in any dimension. Nevertheless, the final result will be presented for one-dimensional problems only, i.e. $d = 1$. We use standard notation for Lebesque and Sobolev spaces, respectively. Let us consider the following boundary value problem: find $u : \Omega \to \mathbb{R}$ such that

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

where $f \in L^2(\Omega)$ and $b \in W^{1,\infty}(\Omega)^d$ such that $\nabla \cdot b = 0$. Let us denote weak derivative of u by u' for $d = 1$.

Let $(.,.)$ and $\|.\|$ be the $L^2(\Omega)$ scalar product and norm, respectively.

Definition 1. *We say that a function $u \in H_0^1(\Omega)$ is a weak solution of (1), if*

$$(\nabla u - bu, \nabla v) = (f, v) \quad \forall v \in H_0^1(\Omega). \tag{2}$$

According to the Lax-Milgram lemma, there exists a unique solution of problem (2).

2.2. Discrete problem

We consider a space partition $\mathcal{T}_h$ consisting of a finite number of closed, d-dimensional simplices K with mutually disjoint interiors and covering $\overline{\Omega}$, i.e., $\overline{\Omega} = \cup_{K\in\mathcal{T}_h} K$. We denote the edges (or faces) by e. In the rest of the paper we speak about boundary objects of co-dimension 1 as about edges, but we mean vertices, edges or faces depending on the dimension d. For each edge e, let $n = n_e$ denote a unit normal vector to e with arbitrary but fixed direction for the inner edges and with outer direction on $\partial\Omega$. The unit outward normal to K will be denoted by n_K. We assume conforming properties of the mesh, i.e., neighbouring elements share an entire edge. We set $h_K = \mathrm{diam}(K)$ and $h = \max_K h_K$. We assume shape regularity of elements, i.e., $h_K/\rho_K \leq C$ for all $K \in \mathcal{T}_h$, where ρ_K is the radius of the largest d-dimensional ball inscribed into K and constant C does not depend on $\mathcal{T}_h$ for $h \in (0, h_0)$. Moreover, we assume the local quasi-uniformity of the mesh, i.e. we assume $h_K \leq Ch_{K'}$ for neighbouring elements K and K' and constant C does not depend on $\mathcal{T}_h$ for $h \in (0, h_0)$ again.

In order to simplify the notation, we set $(.,.)_M$ and $\|.\|_M$ to be the local $L^2(M)$-scalar products and norms, respectively, where $M \subset \overline{\Omega}$ is some union of elements K or edges e. We denote a sum over all elements as $\sum_K$.

We define classical finite element space

$$V_h = \{v \in H_0^1(\Omega) \colon v|_K \in P^p(K)\}, \tag{3}$$

where the space $P^p(K)$ denotes the space of polynomials on K up to the degree $p \geq 1$.

Although the functions from V_h are globally continuous, we will need to work with piece-wise continuous functions as well. We define one-sided values, jumps and mean values on the inner edges

$$\begin{aligned} v(x-) &= \lim_{s \to 0+} v(x - ns), \qquad & v(x+) &= \lim_{s \to 0+} v(x + ns), \\ [v](x) &= v(x-) - v(x+), \qquad & \langle v \rangle(x) &= \frac{1}{2}(v(x-) + v(x+)). \end{aligned} \tag{4}$$

For the boundary edges we define

$$v(x-) = \langle v \rangle(x) = \lim_{s \to 0+} v(x - ns), \qquad [v](x) = 0. \tag{5}$$

Finally, we define the finite element solution of problem (2).

Definition 2. *We say that a function $u_h \in V_h$ is a discrete solution of (2), if*

$$(\nabla u_h - b u_h, \nabla v_h) = (f, v_h) \quad \forall v_h \in V_h. \tag{6}$$

The existence and uniqueness of the discrete solution follows again from the Lax-Milgram lemma.

2.3. Discontinuous Galerkin method

The justification of the presented result is based on the discontinuous Galerkin method. Therefore, we briefly define the interior penalty discontinuous Galerkin discretization of problem (2) using the same notation as in Section 2.2. In order to simplify forthcoming considerations, we assume here purely diffusion problems, i.e. $b = 0$, only. Then the interior penalty discontinuous Galerkin method reads: find $u_h \in X_h$ such that

$$\begin{aligned} \sum_K (\nabla u_h, \nabla v_h)_K - \sum_e (\langle \nabla u_h \rangle \cdot n, [v_h])_e + \theta(\langle \nabla v_h \rangle \cdot n, [u_h])_e \\ + \sum_e (\alpha [u_h], [v_h])_e = (f, v_h) \quad \forall v_h \in X_h, \end{aligned} \tag{7}$$

where $\alpha > 0$ is penalization parameter that should be chosen large enough to ensure positivity of the resulting problem and the space X_h is defined as

$$X_h = \{v \in L^2(\Omega) \colon v|_K \in P^p(K)\}. \tag{8}$$

Parameter θ distinguishes different variants, where the most common variants are symmetric (SIPG, $\theta = 1$), nonsymmetric (NIPG, $\theta = -1$) and incomplete (IIPG, $\theta = 0$). For more details about the discontinuous Galerkin method and its properties see e.g. [6].

The same discretization can be denoted with the aid of the numerical fluxes similarly as in the finite volume method. Then the general discontinuous Galerkin discretization can be expressed as

$$\sum_K (\nabla u_h, \nabla v_h)_K - (\hat{\sigma} \cdot n_K, v_h)_{\partial K} + (\hat{u} - u_h, \nabla v_h \cdot n_K)_{\partial K} = (f, v_h) \quad \forall v_h \in X_h, \quad (9)$$

where the numerical fluxes $\hat{\sigma}$ and $\hat{u}$ approximate ∇u_h and u_h on the edges, respectively. For example, the choice for the numerical fluxes corresponding to IIPG is

$$\hat{u} = u_h, \qquad \hat{\sigma} = \langle \nabla u_h \rangle - \alpha [u_h] n. \quad (10)$$

The connection between the primal discontinuous Galerkin formulations and the formulations with the numerical fluxes is described in [2].

3. A posteriori error bound

3.1. Flux reconstruction

Since the discretization by the finite element method is conforming, the exact solution u as well as the discrete solution u_h belong to common space $H_0^1(\Omega)$. This quality no longer holds for the flux of the solution $\sigma(u) = \nabla u - bu$, since $\sigma(u) \in H(\mathrm{div}, \Omega)$ and $\sigma(u_h) \notin H(\mathrm{div}, \Omega)$ in general. Our aim is to find a suitable reconstruction $\sigma_h = \sigma_h(u_h) \in H(\mathrm{div}, \Omega)$ such that $\sigma_h \approx \sigma(u_h)$.

Let $RT_p(K)$ be the local Raviart-Thomas space of order p for element $K \in \mathcal{T}_h$, i.e. $RT_p(K) = P_p(K)^d + xP_p(K)$. For the details about Raviar-Thomas spaces and about FEM-like spaces for approximation $H(\mathrm{div}, \Omega)$ in general see e.g. [4]. We define the reconstruction σ_h element-wise. We seek $\sigma_h|_K \in RT_p(K)$ such that

$$\begin{aligned} \sigma_h|_e \cdot n &= \phi_e \quad \forall e \subset K, \\ (\sigma_h, z_h)_K &= (\nabla u_h - b u_h, z_h)_K \quad \forall z_h \in P_{p-1}(K)^d, \end{aligned} \quad (11)$$

where $\phi_e \in P_p(e)$ is a suitable function. The conditions in (11) represent the natural degrees of freedom for $RT_p(K)$, see [4, Proposition 2.3.4]. Applying the basis corresponding to these degrees of freedom enables to assemble σ_h directly without the necessity to solve any local linear problems which results in extremely cheap evaluation of the reconstruction σ_h. This property is demonstrated in [15, Lemma 5.1] for $d = 1$.

We point out that the resulting function σ_h has globally continuous normal components and therefore the sum of local contributions of σ_h is globally in $H(\mathrm{div}, \Omega)$.

Important property of σ_h is the orthogonality of $f + \nabla \cdot \sigma_h$ to functions from V_h that follows from the discrete problem formulation (6) and from (11)

$$\begin{aligned}(f + \nabla \cdot \sigma_h, v_h) =& (f, v_h) - (\sigma_h, \nabla v_h) \\ =& (f, v_h) - (\nabla u_h - b u_h, \nabla v_h) = 0 \quad \forall v_h \in V_h. \end{aligned} \tag{12}$$

3.2. Upper bound

We define the error measure as the dual norm of residual

$$\mathrm{Err}(w) = \sup_{0 \neq v \in H_0^1(\Omega)} \frac{(f, v) - (\nabla w - b w, \nabla v)}{\|\nabla v\|}. \tag{13}$$

For the most simple case $b = 0$, the error measure is equivalent to H^1-seminorm, i.e. $\mathrm{Err}(w) = \|\nabla u - \nabla w\|$.

An upper bound to the error measure $\mathrm{Err}(u_h)$ can be derived similarly as in [15]. Here, we present the final result.

Theorem 1. *Let $u_h \in V_h$ be the discrete solution obtained by (6) and σ_h be the reconstruction obtained from u_h by (11). Then*

$$Err(u_h)^2 \leq \eta^2 = \sum_K (\eta_{R,K} + \eta_{F,K})^2, \tag{14}$$

where the local error estimators are

$$\begin{aligned}\eta_{R,K} &= C_P h_K \|f + \nabla \cdot \sigma_h\|_K, \\ \eta_{F,K} &= \|\sigma_h - \sigma(u_h)\|_K = \|\sigma_h - \nabla u_h + b u_h\|_K. \end{aligned} \tag{15}$$

The constant C_P is the Poincare constant and can be bounded by $C_P \leq 1/\pi$, cf. [10]. It shall be pointed out that all the terms in (14) are cheaply computable.

3.3. Choice of ϕ_e

A posteriori error estimate (14) holds regardless of the choice of ϕ_e in (11). On the other hand, the quality of the estimate (14), i.e. how much the estimator η overestimates the error $\mathrm{Err}(u_h)$, depends on the choice of ϕ_e.

The paper [15] discusses the most naive possibility $\phi_e = \langle \nabla u_h \rangle \cdot n$ and the numerical experiments provided in the paper [15] show that this choice is far from optimal in some cases, most importantly for even degree polynomial approximations.

The goal of this paper is to show a suggestion for some more accurate choice of ϕ_e. Since we will only consider one-dimensional problems, we may simplify the domain Ω as the interval $(0, 1)$ and we can denote the partition nodes $0 = e_0 < e_1 < \ldots < e_N = 1$ and the corresponding elements $K_k = [e_{k-1}, e_k]$. Then the suggested choice for ϕ_e is following

$$\begin{aligned}\phi_{e_N} &= -(f, x) - (b u_h, 1) = -\int_0^1 x f(x) + b(x) u_h(x) \mathrm{d}x, \\ \phi_{e_k} &= \phi_{e_{k+1}} + (f, 1)_{K_{k+1}}, \quad k = N - 1, \ldots, 0. \end{aligned} \tag{16}$$

The idea of element-wise flux reconstruction similar to (11) was already applied with success for the interior penalty discontinuous Galerkin a posteriori error estimates, see e.g. [8]. It is possible to find out by careful comparison that the choice for boundary degrees of freedom ϕ_e in [8] corresponds to the numerical fluxes $\hat{\sigma}$, cf. Section 2.3.

Our idea for the choice (16) follows from imitating the discontinuous Galerkin technique, where the finite element method is expressed as a variant of the discontinuous Galerkin method. More precisely, we modify the IIPG numerical flux $\hat{\sigma}$ from (10) in such a way that the resulting IIPG solution with this modified flux is identical to the finite element solution.

Still, there is a work to be done concerning precise numerical analysis, e.g. IIPG error norm justification or IIPG a posteriori error analysis including efficiency analysis.

4. Numerical experiments

The aim of this section is to show how accurate, reliable and robust are a posteriori error estimates based on (11) and (16). The numerical experiments in paper [15], where the naive choice of ϕ_e as $\phi_e = \langle \nabla u_h \rangle \cdot n$ is discussed, show that the estimates are slightly worse in some situations, especially for even polynomial degrees. We want to show that the choice of ϕ_e according to (16) improves this behavior and the resulting estimates are accurate regardless of the situation.

Although the individual error estimator can be computed directly, the evaluation of the error measure can be difficult even in simplified situations, where the exact solution is known, since the defining formula (13) represents the supremum over infinite-dimensional space. Therefore, we approximate the error measure $\mathrm{Err}(w)$ by

$$\mathrm{Err}^+(w) = \sup_{0 \neq v \in V_h^+} \frac{(f, v) - (\nabla w - bw, \nabla v)}{\|\nabla v\|}, \tag{17}$$

where V_h^+ is chosen adaptively and $V_h \subset V_h^+ \subset H_0^1(\Omega)$. The error measure simplifies to $\mathrm{Err}(w) = \|\nabla u - \nabla w\|$ for purely diffusion problems ($b = 0$) and no approximation of the error measure is needed in these situations.

4.1. Purely diffusion problem

We study the error estimate (14) with respect to the mesh refinement and with respect to the changing polynomial degree. We assume the purely diffusion problem ($b = 0$) on the domain $\Omega = (0, 1)$ and we set the right-hand side $f = \pi^2 \sin(\pi x)$.

Since the paper [15] shows that there are two different regimes for odd and even polynomial degrees, we provide the tests with equidistant meshes for refining mesh-size h starting at $h = 1$ and fixed polynomial degrees $p = 2$ and $p = 3$.

We set fixed $h = 0.25$ for the changing polynomial degree tests.

Tables 1–3 show that the estimate (14) provides extremely accurate upper bounds. The estimator η_R converges faster to 0 than the error and the second estimator η_F

$1/h$	$\|u' - u'_h\|$	η	Eff	η_R	η_F
1	$2.6718 - 1$	$3.1054 - 1$	1.16	$5.4235 - 2$	$2.5631 - 1$
2	$1.9719 - 1$	$2.0686 - 1$	1.05	$1.3166 - 2$	$1.9369 - 1$
4	$5.0620 - 2$	$5.1238 - 2$	1.01	$8.4125 - 4$	$5.0396 - 2$
8	$1.2739 - 2$	$1.2778 - 2$	1.00	$5.2868 - 5$	$1.2724 - 2$
16	$3.1900 - 3$	$3.1924 - 3$	1.00	$3.3088 - 6$	$3.1891 - 3$
32	$7.9783 - 4$	$7.9787 - 4$	1.00	$2.0687 - 7$	$7.9777 - 4$
64	$1.9948 - 4$	$1.9949 - 4$	1.00	$1.2930 - 8$	$1.9947 - 4$

Table 1: Global h-performance, diffusion, $p = 2$

$1/h$	$\|u' - u'_h\|$	η	Eff	η_R	η_F
1	$2.6718 - 1$	$3.1054 - 1$	1.16	$5.4235 - 2$	$2.5631 - 1$
2	$2.6332 - 2$	$2.7382 - 2$	1.04	$1.3086 - 3$	$2.6073 - 2$
4	$3.3650 - 3$	$3.3984 - 3$	1.01	$4.1667 - 5$	$3.3567 - 3$
8	$4.2295 - 4$	$4.2400 - 4$	1.00	$1.3082 - 6$	$4.2269 - 4$
16	$5.2941 - 5$	$5.2974 - 8$	1.00	$4.0928 - 8$	$5.2933 - 5$
32	$6.6199 - 6$	$6.6211 - 6$	1.00	$1.4696 - 9$	$6.6197 - 6$
64	$8.2751 - 7$	$8.2778 - 7$	1.00	$3.3135 - 10$	$8.2756 - 7$

Table 2: Global h-performance, diffusion, $p = 3$

as expected. On the other hand, the results show that the estimator η_F is not able to provide upper bound without the correction from the estimator η_R. Moreover, Tables 1–3 show that there is no longer any significant difference between odd and even polynomial degrees, compare with [15].

4.2. Convection-diffusion problem

We study convection-diffusion equation

$$-\epsilon u'' + bu' = f, \tag{18}$$

where $\Omega = (0, 1)$, $b = 1$, $f = 1$ and $\epsilon > 0$ is a constant. For more information about convection-diffusion problems see [13]. We present the performance of the estimate (14) with respect to h for fixed $\epsilon = 0.01$, $p = 1$ and successively refined equidistant meshes starting with $h = 0.1$ and with respect to ϵ for the fixed equidistant mesh with $h = 0.025$ and decreasing parameter ϵ.

Tables 4 and 5 show that the accuracy of the estimate is preserved either for convection or diffusion dominated situation and the estimate is accurate and robust with respect to h as well as ϵ.

Moreover, it is possible to study the local distribution of errors and corresponding estimates. Figure 1 presents the exact solution u and the discrete solution u_h for the convection dominated situation on the equidistant mesh with $h = 0.1$ and $\epsilon = 0.01$. The corresponding distribution of estimates is presented in Figure 2. We can find

p	$\|u' - u'_h\|$	η	Eff	η_R	η_F
1	$4.9851 - 1$	$5.0603 - 1$	1.02	$1.2655 - 2$	$4.9338 - 1$
2	$5.0620 - 2$	$5.1238 - 2$	1.01	$8.4125 - 4$	$5.0396 - 2$
3	$3.3650 - 3$	$3.3984 - 3$	1.01	$4.1667 - 5$	$3.3567 - 3$
4	$1.6667 - 4$	$1.6806 - 4$	1.01	$1.6459 - 6$	$1.6641 - 4$
5	$6.5836 - 6$	$6.6304 - 6$	1.01	$5.3935 - 8$	$6.5765 - 6$
6	$2.1766 - 7$	$2.1911 - 7$	1.01	$4.2163 - 9$	$2.1617 - 7$

Table 3: Global p-performance, diffusion, $h = 0.25$

$1/h$	$\mathrm{Err}^+(u_h)$	η	Eff
10	$2.0665 - 1$	$2.0770 - 1$	1.01
20	$1.0155 - 1$	$1.0206 - 1$	1.01
40	$5.0775 - 2$	$5.1031 - 2$	1.01
80	$2.5388 - 2$	$2.5516 - 2$	1.01
160	$1.2694 - 2$	$1.2758 - 2$	1.01

Table 4: Global h-performance, convection-diffusion, $\epsilon = 0.01$

ϵ	$\mathrm{Err}^+(u_h)$	η	Eff
$1.0 - 0$	$7.4691 - 3$	$7.5067 - 3$	1.01
$1.0 - 1$	$1.6057 - 2$	$1.6138 - 2$	1.01
$1.0 - 2$	$5.0775 - 2$	$5.1031 - 2$	1.00
$1.0 - 3$	$1.6159 - 1$	$1.6164 - 1$	1.00
$1.0 - 4$	$9.1726 - 1$	$9.1727 - 1$	1.00

Table 5: Global ϵ-performance, convection-diffusion, $h = 0.025$

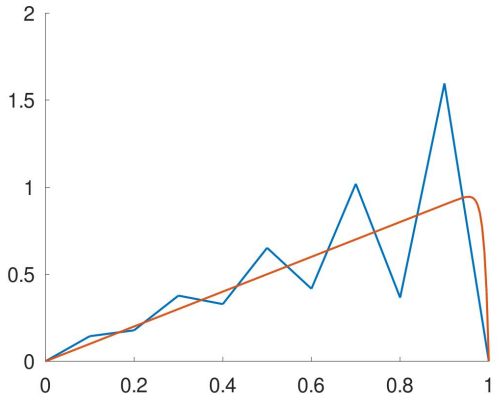


Figure 1: Exact and discrete solution

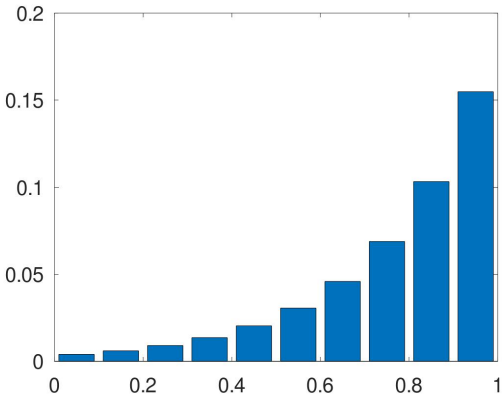


Figure 2: Element-wise error estimates

out comparing Figures 1 and 2 that the distribution of the error matches very well with the distribution of the local error estimates.

5. Conclusion

We suggested an improvement of the flux reconstruction for a posteriori error estimates from [15] for one-dimensional problems and provided numerical experiments verifying the accuracy, robustness and reliability of the suggested reconstruction. The main drawback lies in the fact that it is not obvious how to extend presented result to multi-dimensional problems. Moreover, precise analysis is still missing as well. These topics will be part of the future research.

Acknowledgements

This research was financially supported by the Czech Science Foundation grant 20-14736S. The work was also supported from European Regional Development Fund-Project "Center for Advanced Applied Science"
No. CZ.02.1.01/0.0/0.0/16 019/0000778.

References

[1] Ainsworth, M., Oden, J. T.: A procedure for a posteriori error estimation for hp finite element methods. Comput. Methods Appl. Mech. Engrg. **101** (1992), 73–96.

[2] Arnold, D. N., Brezzi, F., Cockburn, B., and Marini, L. D. : Unified analysis of discontinuous Galerkin methods for elliptic problems. SIAM J. Numer. Anal. **39** (2002), 1749–1779.

[3] Babuška, I., Strouboulis, T.: *The finite element method and its reliability*. Numer. Math. Sci. Comput., Oxford University Press, New York (2001).

[4] Boffi, D., Brezzi, F., and , Fortin, M.: *Mixed finite element methods and applications*. Springer Series in Computational Mathematics 44, Berlin: Springer (2013).

[5] Braess, D., Pillwine, V., Schöberl, J.: Equilibrated residual error estimates are p-robust. Comput. Methods Appl. Mech. Engrg. **198** (2009), 1189–1197.

[6] Dolejší, V. and Feistauer, M.: *Discontinuous Galerkin method. Analysis and applications to compressible flow*. Springer Ser. Comput. Math. 48 Cham: Springer (2015).

[7] Eriksson, K., Estep, D., Hansbo, P., and Jonson, C.: *Computational differential equations.* Cambridge University Press, Cambridge (1996).

[8] Ern, A., Stephansen, A. F., and Vohralík, M.: Guaranteed and robust discontinuous Galerkin a posteriori error estimates for convection-diffusion-reaction problems. J. Comput. Appl. Math. **243** (2010), 114–130.

[9] Ern, A., Vohralík, M.: Polynomial-degree-robust a posteriori estimates in a unified setting for conforming, nonconforming, discontinuous Galerkin, and mixed discretizations. SIAM J. Numer. Anal. **53** (2015), 1058–1081.

[10] Payne, L. E. and Weinberger, H. F.: An optimal Poincaré inequality for convex domains. Arch. Ration. Mech. Anal. **5** (1960), 286–292.

[11] Prager, W. and Synge, J. L.: Approximations in elasticity based on the concept of function space. Quart. Appl. Math. **5** (1947), 241–269.

[12] Repin, S. I.: *A posteriori estimates for partial differential equations.* Radon Ser. Comput. Appl. Math., Walter de Gruiter, Berlin (2008).

[13] Roos, H. G., Stynes, M. and Tobiska, L.: *Robust numerical methods for singularly perturbed differential equations.* Convection-diffusion-reaction and flow problems. 2nd ed., Springer Ser. Comput. Math. 24., Berlin: Springer (2008).

[14] Verfürth, R.: *A posteriori error estimation techniques for finite element methods.* Numer. Math. Sci. Comput., Oxford University Press, Oxford (2013).

[15] Vlasák, M.: On polynomial robustness of flux reconstructions, Appl. Math. **65** (2020), 153–172.

LIST OF PARTICIPANTS

Monika Balázsová, balazmon@fjfi.cvut.cz
Katedra matematiky, Fakulta jaderná a fyzikálně inženýrská, Praha

Stanislav Bartoň, s.barton@po.opole.pl
Faculty of Electrical Engineering, Automatic Control and Informatics, Opole, Polsko

Ondřej Bartoš, ondra.bartosh@seznam.cz
Katedra numerické matematiky, Matematicko-fyzikální fakulta UK, Praha

Michal Béreš, michal.beres@ugn.cas.cz
Ústav geoniky AV ČR, v. v. i., Ostrava

Simona Bérešová, Simona.Beresova@ugn.cas.cz
Ústav geoniky AV ČR, v. v. i., Ostrava

Adam Bílek, adam.bilek@vsb.cz
Vysoká škola báňská – Technická univerzita Ostrava

Hana Bílková, hbilkova@math.cas.cz
Matematický ústav AV ČR, v. v. i., Praha

Marek Brandner, brandner@kma.zcu.cz
Katedra matematiky, Fakulta aplikovaných věd, Západočeská univerzita v Plzni

Jan Březina, jan.brezina@tul.cz
Technická univerzita v Liberci

Jana Burkotová, jana.burkotova@upol.cz
Katedra matematické analýzy a aplikací matematiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci

Martin Čermák, martin.cermak@vsb.cz
Vysoká škola báňská – Technická univerzita Ostrava

Dana Černá, dana.cerna@tul.cz
Katedra matematiky a didaktiky matematiky, Fakulta přírodovědně-humanitní a pedagogická, Technická univerzita v Liberci

Barbora Chlebišová, barborachlebisova@gmail.com
Vysoká škola báňská – Technická univerzita Ostrava

Jan Chleboun, jan.chleboun@cvut.cz
Katedra matematiky, Fakulta stavební ČVUT v Praze

Dagmar Dlouhá, dagmar.dlouha@vsb.cz
Vysoká škola báňská – Technická univerzita Ostrava

Vít Dolejší, dolejsi@karlin.mff.cuni.cz
Katedra numerické matematiky, Matematicko-fyzikální fakulta UK, Praha

Vojtěch Dorňák, vojtech.dornak@vsb.cz
Vysoká škola báňská – Technická univerzita Ostrava

Viktor Dubovský, viktor.dubovsky@vsb.cz
Vysoká škola báňská – Technická univerzita Ostrava

Jurjen Duintjer Tebbens, duintjertebbens@cs.cas.cz
Farmaceutická fakulta v Hradci Králové, Univerzita Karlova

Cyril Fischer, fischerc@itam.cas.cz
Ústav teoretické a aplikované mechaniky AV ČR, v. v. i., Praha

Eva Havelková, Havelkova-Eva@seznam.cz
Katedra numerické matematiky, Matematicko-fyzikální fakulta UK, Praha

David Horák, david.horak@vsb.cz
Ústav geoniky AV ČR, v. v. i., Ostrava

Jiří Hozman, jiri.hozman@tul.cz
Katedra matematiky a didaktiky matematiky, Fakulta přírodovědně-humanitní a pedagogická, Technická univerzita v Liberci

Čeněk Jirsák, cenek.jirsak@tul.cz
Katedra matematiky a didaktiky matematiky, Fakulta přírodovědně-humanitní a pedagogická, Technická univerzita v Liberci

Radka Keslerová, Radka.Keslerova@fs.cvut.cz
Ústav technické matematiky, Fakulta strojní ČVUT v Praze

Štěpán Klapka, klapka.stepan@azd.cz
AŽD Praha s. r. o., Praha

Alexej Kolcun, alexej.kolcun@ugn.cas.cz
Ústav geoniky AV ČR, v. v. i., Ostrava

Vladislav Kozák, kozak.v@fce.vutbr.cz
Ústav matematiky a deskriptivní geometrie, Fakulta stavební VUT v Brně

Jakub Kružík, jakub.kruzik@vsb.cz
Vysoká škola báňská – Technická univerzita Ostrava; Ústav geoniky AV ČR, v. v. i., Ostrava

Václav Kučera, kucera@karlin.mff.cuni.cz
Katedra numerické matematiky, Matematicko-fyzikální fakulta UK, Praha

Michal Kuráž, kuraz@fzp.czu.cz
Česká zemědělská univerzita v Praze

Jan Lamač, jan.lamac@cvut.cz
Katedra matematiky, Fakulta stavební ČVUT v Praze

Martin Lanzendörfer, martin.lanzendorfer@natur.cuni.cz
Přírodovědecká fakulta, Univerzita Karlova

Tomáš Luber, tomas.luber@ugn.cas.cz
Ústav geoniky AV ČR, v. v. i., Ostrava

Dalibor Lukáš, dalibor.lukas@vsb.cz
Vysoká škola báňská – Technická univerzita Ostrava

Volodymyr Lynnyk, voldemar@utia.cas.cz
Ústav teorie informace a automatizace AV ČR, v. v. i., Praha

Jitka Machalová, jitka.machalova@upol.cz
Katedra matematické analýzy a aplikací matematiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci

Josef Malík, josef.malik@ugn.cas.cz
Ústav geoniky AV ČR, v. v. i., Ostrava

Ctirad Matonoha, matonoha@cs.cas.cz
Ústav informatiky AV ČR, v. v. i., Praha

Karol Mikula, karol.mikula@gmail.com
Slovak University of Technology, Bratislava, Slovakia

Ivan Němec, nemec@fem.cz
Ústav matematiky a deskriptivní geometrie, Fakulta stavební VUT v Brně

Eda Oktay, oktay@karlin.mff.cuni.cz
Katedra numerické matematiky, Matematicko-fyzikální fakulta UK, Praha

Štěpán Papáček, spapacek@seznam.cz
Ústav teorie informace a automatizace AV ČR, v. v. i., Praha

Jan Papež, papez@math.cas.cz
Matematický ústav AV ČR, v. v. i., Praha

Marek Pecha, marek.pecha@vsb.cz
Ústav geoniky AV ČR, v. v. i., Ostrava

Martin Plešinger, martin.plesinger@tul.cz
Katedra matematiky a didaktiky matematiky, Fakulta přírodovědně-humanitní a pedagogická, Technická univerzita v Liberci

Lukáš Pospíšil, lukas.pospisil@vsb.cz
Vysoká škola báňská – Technická univerzita Ostrava

Stefano Pozza, pozza@karlin.mff.cuni.cz
Katedra numerické matematiky, Matematicko-fyzikální fakulta UK, Praha

Ivana Pultarová, Ivana.Pultarova@cvut.cz
Katedra matematiky, Fakulta stavební ČVUT v Praze

Jana Radová, jana.radova@upol.cz
Katedra matematické analýzy a aplikací matematiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci

Branislav Rehák, rehakb@utia.cas.cz
Ústav teorie informace a automatizace AV ČR, v. v. i., Praha

Hynek Řezníček, reznicek@cs.cas.cz
Ústav informatiky AV ČR, v. v. i., Praha

Miroslav Rozložník, miro@math.cas.cz
Matematický ústav AV ČR, v. v. i., Praha

Adam Rychtář, rychtar.adam@azd.cz
AŽD Praha s. r. o., Praha

Karel Segeth, segeth@math.cas.cz
Matematický ústav AV ČR, v. v. i.

David Šilhánek, david.silhanek@fsv.cvut.cz
Katedra matematiky, Fakulta stavební ČVUT v Praze

Dorota Šimonová, simonova@karlin.mff.cuni.cz
Katedra numerické matematiky, Matematicko-fyzikální fakulta UK, Praha

Jakub Šístek, sistek@math.cas.cz
Matematický ústav AV ČR, v. v. i., Praha

Radek Svačina, radek.svacina@upol.cz
Katedra matematické analýzy a aplikací matematiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci

Tadeáš Světlík, tadeas.svetlik@vsb.cz
Vysoká škola báňská – Technická univerzita Ostrava

Stanislav Sysala, stanislav.sysala@ugn.cas.cz
Ústav geoniky AV ČR, v. v. i., Ostrava

Petr Tichý, ptichy@karlin.mff.cuni.cz
Katedra numerické matematiky, Matematicko-fyzikální fakulta UK, Praha

Lukáš Vacek, lukas.vacek6@gmail.com
Katedra numerické matematiky, Matematicko-fyzikální fakulta UK, Praha

Karel Vacek, karel.vacek@fs.cvut.cz
Ústav technické matematiky, Fakulta strojní ČVUT v Praze

Jiří Vala, Vala.J@fce.vutbr.cz
Ústav matematiky a deskriptivní geometrie, Fakulta stavební VUT v Brně

Jan Valášek, cvalda.valasek@gmail.com
Matematický ústav AV ČR, v. v. i., Praha

Radek Varga, lukas.pospisil@vsb.cz
Vysoká škola báňská – Technická univerzita Ostrava

Tomáš Vejchodský, vejchod@math.cas.cz
Matematický ústav AV ČR, v. v. i., Praha

Miloslav Vlasák, vlasakmila@gmail.com
Katedra matematiky, Fakulta stavební ČVUT v Praze

Jana Žáková, jana.zakova@tul.cz
Katedra matematiky a didaktiky matematiky, Fakulta přírodovědně-humanitní a pedagogická, Technická univerzita v Liberci