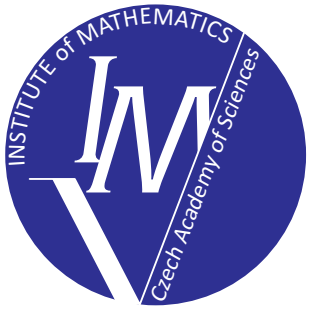
# Programs and Algorithms of Numerical Mathematics 22

Hejnice, June 23–28, 2024

## Proceedings of Seminar

Edited by
J. Chleboun, J. Papež, K. Segeth, J. Šístek, T. Vejchodský




Institute of Mathematics
Czech Academy of Sciences
Prague 2025

ISBN 978-80-85823-74-5
Matematický ústav AV ČR, v. v. i.
Praha 2025

**Contents**

# Preface

These proceedings comprise peer-reviewed papers based on the invited lectures, short communications, and poster presentations from the 22nd seminar *Programs and Algorithms of Numerical Mathematics* (PANM), held at Hejnice Monastery, Czech Republic, from June 23 to 28, 2024.

The seminar was organized by the Institute of Mathematics of the Czech Academy of Sciences under the auspices of EU-MATHS-IN.CZ, the Czech Network for Mathematics in Industry, with financial support from the RSJ Foundation. Continuing the tradition of its predecessors, PANM 2024 followed a long-standing series of biennial (with one exception) seminars on mathematical software and numerical methods, held in various locations — including Alšovice, Bratříkov, Janov nad Nisou, Kořenov, Lázně Libverda, Dolní Maxov, Prague, Hejnice, and Jablonec nad Nisou — since its inception in 1983. The primary objective of these seminars is to provide a platform for discussing advanced topics in numerical analysis, the implementation of numerical algorithms, novel approaches to mathematical modelling, and computational methods for both single- and multi-processor applications.

The seminar welcomed 45 participants, primarily from Czech universities and institutes of the Czech Academy of Sciences, several also from abroad. We particularly value the participation of young scientists, PhD candidates, and undergraduate students. We hope that those attending PANM for the first time found the atmosphere of the seminar both welcoming and stimulating and that they will continue to be part of the PANM community in the future.

The conference photo was taken in front of the Hejnice Monastery that hosted the seminar. We are grateful for the opportunity to return to these inspiring premises after the break in 2022.

The organizing committee comprised Jan Chleboun, Jan Papež, Miro Rozložník, Karel Segeth, Jakub Šístek, and Tomáš Vejchodský. We also extend our sincere thanks to Ms. Hana Bílková for preparing the manuscripts for both the electronic and print versions of these proceedings.

Finally, the editors and organizers wish to thank to all participants for their valuable contributions and, in particular, to the scientists who dedicated their time to reviewing the submitted manuscripts.

*Editors*

*Programs and Algorithms of Numerical Mathematics 22*
J. Chleboun, J. Papež, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics CAS, Prague 2025

# AERODYNAMIC DECELERATION AT VELOCITIES NEAR THE ESCAPE VELOCITY

Stanislav Bartoň

Opole University of Technology
Prószkowska Street 76, 45-758 Opole, Poland
s.barton@po.edu.pl

**Abstract:** This article presents basic procedures for calculating the trajectory of a spaceship that uses only the Earth's atmosphere to reduce its speed, allowing it to land on the Earth's surface successfully. The first flight of the `ARTEMIS` program, which took place from November 16 to December 11 2022, was used as a template for the calculations. All calculations are performed in the symbolic algebra program Maple. To simplify the calculations, forces that have a less significant impact on the shape of the trajectory, such as the gravitational influence of the Sun and Moon, the rotation of the Earth, and its non-spherical shape, were neglected. To conserve space, only the essential components of the solution are shown, given the intensive calculations involved. The commands used to produce the graphics are not included.

**Keywords:** Newton's equations of motion, aerodynamic drag, atmospheric density, gravitational field, iteration method, Maple

**MSC:** 34A34, 68W30, 76-04

## 1. Introduction

The Artemis I mission inspired this article, which expands on the mathematical models the author developed in 2002, see [3].

Spacecraft landings are among the most demanding phases of space missions. Lunar missions necessitate precise trajectories, mandating frequent course corrections. Altering the spacecraft's direction and velocity during lunar orbit insertion and Earth return is also crucial. Fuel consumption is directly linked to the spacecraft's overall mass, meaning more fuel is required for final maneuvers if more is needed for mid-course corrections. Given the limited launch mass of rockets, fuel for course corrections is also limited. To conserve fuel during atmospheric reentry, spacecraft utilize aerodynamic drag, a process that doesn't consume fuel.

All input variables for the forthcoming calculations are expressed numerically using SI base or derived units. To conserve space, only numerical values will be

DOI: 10.21136/panm.2024.01

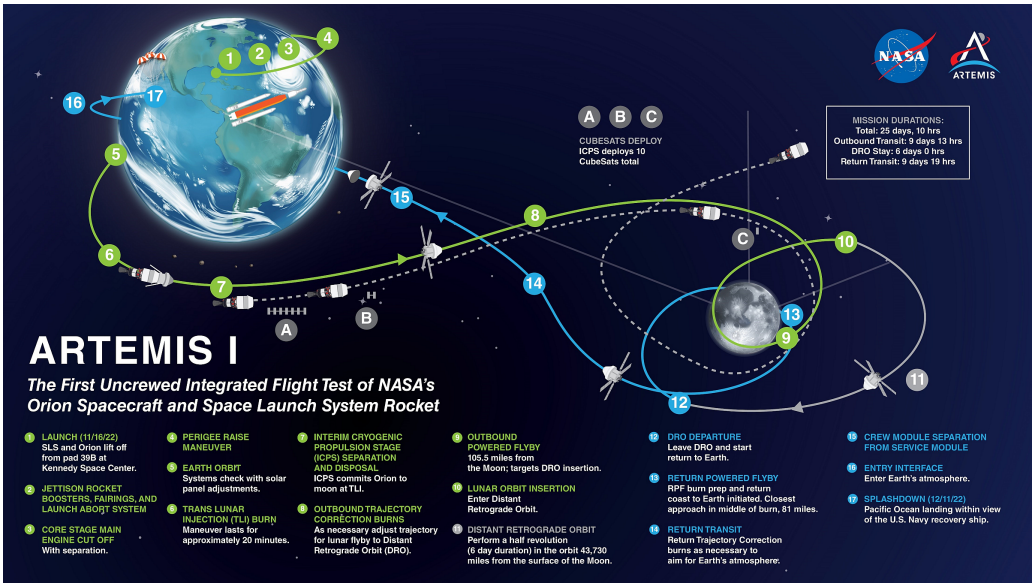



Figure 1: `ARTEMIS I` Mission Map, [1].

presented. For spacecraft returning from lunar trajectories, reentry speeds approach the escape cosmic velocity, $\approx 11.2\text{e}3$. Martian return speeds are significantly higher, around $\approx 21\text{e}3$. The Orion spacecraft during the ARTEMIS 1 mission reentered at $W = 10.7\text{e}3$, detailed in [2] and see Fig. 1.

## 2. Derivation of equations of motion

We start with the fundamental form of Newton's equations of motion:

$$\left[\frac{d^2\,x(t)}{dt^2}, \frac{d^2\,y(t)}{dt^2}\right] = \frac{1}{M}\,\left[F_x\left(x(t), y(t)\right), F_y\left(x(t), y(t)\right)\right], \tag{1}$$

where $x$ and $y$ represent rectangular coordinates with the origin at the Earth's center. The positive $x$-axis points towards the initial point of the landing trajectory, located at $[x_0, 0]\,, x_0 = 1\text{e}6$, where the spacecraft is at time $t = 0$ and $\vec{F}$ is the vector representing the total force acting on the spacecraft, which has a mass of $M = 1.00375\text{e}5$, see [5].

The primary forces acting on a descending spacecraft are the gravitational force $\vec{G} \equiv [G_x, G_y]$ and and aerodynamic drag $\vec{D} \equiv [D_x, D_y]$

$$|\vec{G}| = \frac{\kappa\,Me\,M}{r^2}, \qquad |\vec{D}| = \frac{C_x\,\rho(h)\,S\,V^2}{2}. \tag{2}$$

In these equations: $Me = 5.97\text{e}24$ is the mass of the Earth, $\kappa = 6.67\text{e}-11$, $r$ is the distance of the spacecraft from the Earth's center, $C_x = 1.5$ is the spacecraft's drag coefficient, see [6], $S = 19.6$ is the spacecraft's frontal area, see [5] and $V$ is the spacecraft's velocity.

The function describing the variation of atmospheric density $\rho$ with altitude $h$ above the Earth's surface is

$$\rho(h) = \mathrm{e}^{\left(\left(c_1\,h^2 - c_2\,h + c_3\right)\mathrm{He}(c_4 - h) - \frac{(c_5\,h - c_6)\mathrm{He}(h - c_4)}{h + c_7}\right)}, \text{ where } \begin{array}{ll} c_1 = 6.392146930\text{e}{-11} & c_5 = 3.502764072\text{e}1 \\ c_2 = 1.447577359\text{e}{-4} & c_6 = 1.41258792\text{e}6 \\ c_3 = 3.316213319\text{e}{-1} & c_7 = 5.494654461\text{e}4 \\ c_4 = 1.044139387\text{e}5 & \mathrm{He} = \text{Heaviside function} \end{array} . \quad (3)$$

The function $\rho(h)$, defined by equation (3), is a generalization of atmospheric density relationships found in [7]. The coefficients $c_1, \ldots, c_7$ were computed in Maple using a least squares fit to a nonlinear, piecewise defined regression model for atmospheric density, based on tabulated values in [4].

The following substitutions can now be used:

$$\begin{gathered} F_x = -G_x - D_x,\ F_y = -G_y - D_y,\ r = \sqrt{x(t)^2 + y(t)^2},\ h = r - RE, \\ V = \sqrt{\left(\frac{d\,x(t)}{dt}\right)^2 + \left(\frac{d\,y(t)}{dt}\right)^2},\ G_x = |G|\frac{x(t)}{r},\ G_y = |G|\frac{y(t)}{r},\ D_x = \frac{|D|}{V}\frac{d\,x(t)}{dt},\ D_y = \frac{|D|}{V}\frac{d\,y(t)}{dt} \end{gathered} , \quad (4)$$

where $RE = 6.378\text{e}6$ represents the Earth's radius. These substitutions, along with equations (2), (3) and the provided numerical values, can then be used in equation (1). Because of space limitations, we are unable to show the final form of these equations.

## 3. Calculation of the return trajectory in Maple

A description of the derivation of the equations of motion in Maple would be very lengthy and formally identical to the previous chapter. Therefore, we will assume that the equations of motion have already been derived in Maple from the substitutions (4), (3) and (2) into equation (1) and that this vector equation has been divided in Maple into two equations describing the motion in the $x$ and $y$ axes. These equations have been named `EQx` and `EQy` in Maple. Both equations form a system of ordinary nonlinear second-order differential equations. Their numerical solution and the search for the optimal return trajectory of the spacecraft are the subject of this part.

By the optimal return trajectory, we mean a trajectory that minimizes deceleration caused by aerodynamic drag. To quantify this deceleration, we introduce a new variable, `Ag` $= \frac{\sqrt{Dx^2 + Dy^2}}{g}$, representing the aerodynamic load factor in multiples of the standard gravitational acceleration, $g = 9.81$. A commonly used term for this quantity is g-force. Here, `Dx` and `Dy` represent the components of the aerodynamic drag force in the $x$ and $y$ directions, respectively.

If the g-force is maximal, then the condition for the existence of an extremum of a continuous function must hold: $\frac{d\,Ag(t)}{dt} = 0$ and the corresponding time $t$ for which this condition is fulfilled can be found using the Newton-Raphson method. However, the complication lies in the very complex form of the variable `Ag`. It contains multiple occurrences of both $\frac{d\,x(t)}{dt}$ and $\frac{d\,y(t)}{dt}$ always within the arguments of nonlinear functions. Given that the use of the Newton-Raphson method requires

the computation of `Agt` $= \frac{d\,Ag(t)}{dt}$ and `Agtt` $= \frac{d^2\,Ag(t)}{dt^2}$ an analytical expression of these variables would be possible but practically unusable. The analytical expression of `Agtt` alone contains 66200 characters and occupies over 300 MB in Maple's memory.

However, we can still proceed with a numerical solution. To do this, we must first define the following initial conditions: `W:=1.07e4: Alpha:=78.0: RE:=6.378e6: x0:=RE+1e6: y0:=0`. These values represent the spacecraft's initial position at time $t = 0$ which is $[x0, y0]$, and its initial velocity `W`, which makes an angle of `Alpha` degrees with the direction towards the center of the Earth.

Once the initial conditions `Ini` are defined, we can numerically solve the system of equations `EQx` and `EQy` to obtain a solution `Ns`. This solution provides the values of the coordinates and their corresponding velocities at any given time, as demonstrated in the last line of the following code.

```
> alpha:=evalf(convert(Alpha*degrees,radians));
> Ini:=x(0)=x0,y(0)=0,D(x)(0)=-W*cos(alpha),D(y)(0)=W*sin(alpha):
> Ns:=dsolve({Ini,EQx,EQy},{x(t),y(t)},numeric):
> Ns(500.);
```

$$\left[t=500.0, x(t)=5249645.296, \frac{d\,x(t)}{dt}=-121.112, y(t)=3646265.675, \frac{d\,y(t)}{dt}=-73.279\right]$$

A slightly modified `Ns(tf)` procedure allows for the numerical computation of the third time derivatives of coordinates $x$ and $y$ at time `tf`. The resulting values are then substituted into variables `Agtt` and `Agt` with the aim of applying the Newton-Raphsson method to determine the maximum g-force and the corresponding time.

After deriving the equations of motion `EQx` and `EQy`, the general analytical expressions for the third time derivatives of the coordinates, `Xttt:=diff(rhs(EQx),t):` and `Yttt:=diff(rhs(EQx),t):`, must be obtained. Before initiating the iteration, the first element `SUNs:= Ns(tau)[2..-1]:` needs to be extracted from the output of `Ns`.

Now, the iterative procedure `AGmax` can be used to determine the exact time at which the maximum overload occurs. The input parameter for this procedure is an estimate of the time when we expect the maximum g-force to happen.

```
AGmax := proc(tau) global tau, SUNs;
   local dt, Xtts, Ytts, Xttts, Yttts, Agtts, Agts;
   dt:=1: SUNs:= Ns(tau)[2..-1]:
   while abs(dt)>1e-6 do:
      Xtts:=diff(x(t),t,t)=evalf(subs(SUNs,rhs(EQx)));
      Ytts:=diff(y(t),t,t)=evalf(subs(SUNs,rhs(EQy)));
      Xttts:=diff(x(t),t,t,t)=evalf(subs(Xtts,Ytts,SUNs,Xttt));
      Yttts:=diff(y(t),t,t,t)=evalf(subs(Xtts,Ytts,SUNs,Yttt));
      Agtts:=evalf(subs(Xttts,Yttts,Xtts,Ytts,SUNs,Agtt));
      Agts:=evalf(subs(Xtts,Ytts,SUNs,Agt));dt:=-Agts/Agtts;tau:=tau+dt;
   end do:
end proc:
```

Similarly, the landing time, i.e., the time at which the altitude $h = 0$, can be determined. The calculation is performed by the procedure `Tau`, again using the Newton-Raphson method. The input parameter of the procedure is the estimated landing time.

```
Tau := proc(T) local dt, tau;
   dt := 1.0; tau := T;
   while .1e-3 < abs(dt) do
      dt := subs(Ns(tau),-h/diff(h,t)); tau := tau+dt end do;
   tau
end proc;
```

Given the specified initial conditions, the trajectory can now be computed. The landing time, denoted by `t0 - #1`, is initially determined using the subroutine `Tau`. The trajectory is then visualized using the `odeplot` command and saved as `TR - #2`. Similarly, a plot of the overload versus time is generated and saved as `GT - #3`. The coordinates of the data points on this plot, represented as $[t, Ag]$, are extracted and stored in the matrix `MG - #4`. From this matrix, approximate values of the maximum g-forces and their corresponding times are determined and stored in `MMG - #5`. Considering the selected entry angle, multiple maxima, denoted by `nu - #6`, may exist.

A loop spanning lines `#7 - #11` processes the approximate value of each maximum. The subroutine `AGmax - #8` is employed to calculate the precise times corresponding to these maxima, and the exact g-force values are stored in `Agf - #9`. The ordered pairs [*time of maximum, maximum g-force*] are then collected into the list `AGF - #10`.

```
> t0:=Tau(t0);                                                    #1
> TR:=display(odeplot(Ns,[x(t)/RE,y(t)/RE],0..t0,numpoints=1000)): #2
> GT:=odeplot(Ns,[t,Ag],0..t0,numpoints=1000):                    #3
> MG:=convert(op(1,op(1,GT)),Matrix);                             #4
> MMG:=[seq(`if`(MG[i-1,2]<MG[i,2] and MG[i,2]>MG[i+1,2],
      [MG[i,1],MG[i,2]],NULL),i=2..999)];                         #5
> nu:=nops(MMG); AGF:=[];                                         #6
> for k from 1 by 1 to nu do                                      #7
>   tau:=MMG[k][1]; AGmax(tau);                                   #8
>   Agf:=evalf(subs(SUNs,Ag));                                    #9
>   AGF:=[AGF[],[tau,Agf]];                                       #10
> end do:                                                         #11
```

## 4. Calculation of the optimal trajectory

The optimal trajectory is designed to minimize the overload during aerodynamic braking while also considering the descent time. For this reason, it is advantageous to select an entry angle into the atmosphere that results in two overload peaks of equal magnitude. It is possible to find return trajectories with three or more g-force peaks, but following these trajectories results in circumnavigating the entire Earth

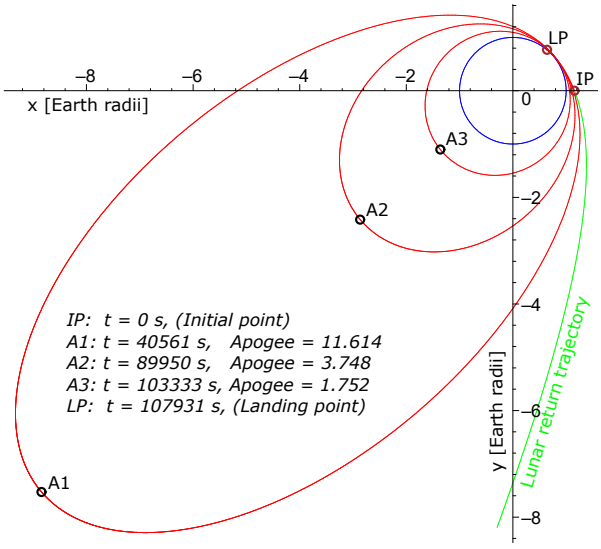



Figure 2: Trajectory with 3 g-force peaks.

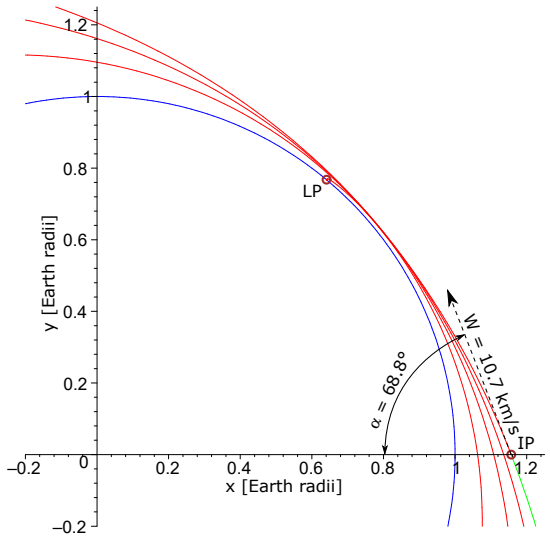



Figure 3: Detail.

on a high-apogee orbits, see Figures 2 and 3. This significantly increases the landing time. Therefore, these trajectories were rejected.

By following the procedure outlined in Section 3 and appropriately selecting the interval of the angle `Alpha` and the step size `dAlpha`, such an angle can be found very quickly, see Figure 4.

The optimal value is $\alpha = 68^\circ\,37'\,8.08'' \pm 0.04''$. Landing occurs at $t_0 = 847.93 \pm 0.01\,[s]$ from the moment the spacecraft was at the initial point. The maximum g-gorce is $Ag = 6.430 \pm 0.001\,[g]$ and the g-force peaks occur at times $t_1 = 443.33 \pm 0.01\,[s]$ and $t_2 = 550.94 \pm 0.01\,[s]$, see Figure 5. Figure 6 illustrates the dependence of a spacecraft's flight altitude on time, while Figure 7 depicts the time evolution of the spacecraft's velocity.

Figure 8 presents the dynamics of the final landing maneuver for the optimal angle as a 3D curve $[Ag(t), V(t), h(t)]$, along with its projections onto the $[Ag(t), V(t)]$, $[Ag(t), h(t)]$, and $[V(t), h(t)]$ planes.

## 5. Conclusion

The presented calculations underscore the critical role of precise navigation in spacecraft reentry. The range of angles $\alpha$ that guarantee a safe landing is exceptionally narrow. If the spacecraft deviates from the optimal value of $\alpha = 68^\circ\,37'\,8.08'' \pm 0.04''$ by $-3'\,50''$, the landing g-force will exceed 10g. A deviation of $-17'\,46''$ will result in a g-force exceeding 20g, which could have fatal consequences. Additionally, at low entry angles, the heat shield may not provide sufficient protection, as the rate of conversion of kinetic energy to thermal energy can be very high.

If the value of angle $\alpha$ increases by $10'\,21''$, the landing will occur after one day, or 86 400 seconds, after passing the initial point. If the deviation is $11'\,50''$, the

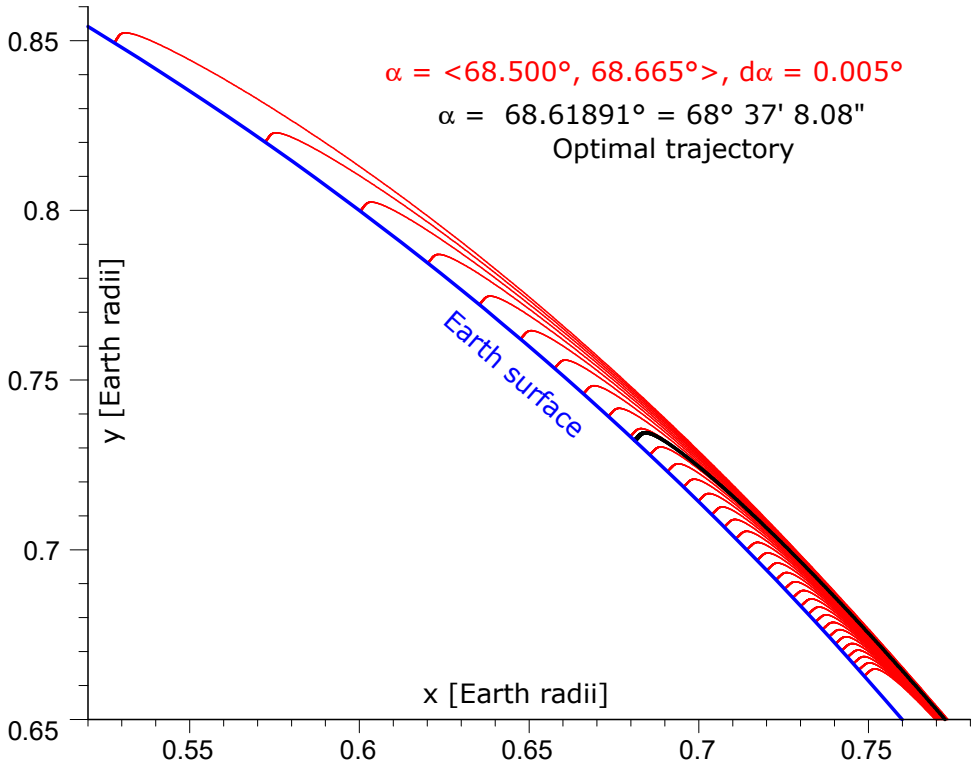



Figure 4: Landing trajectories.

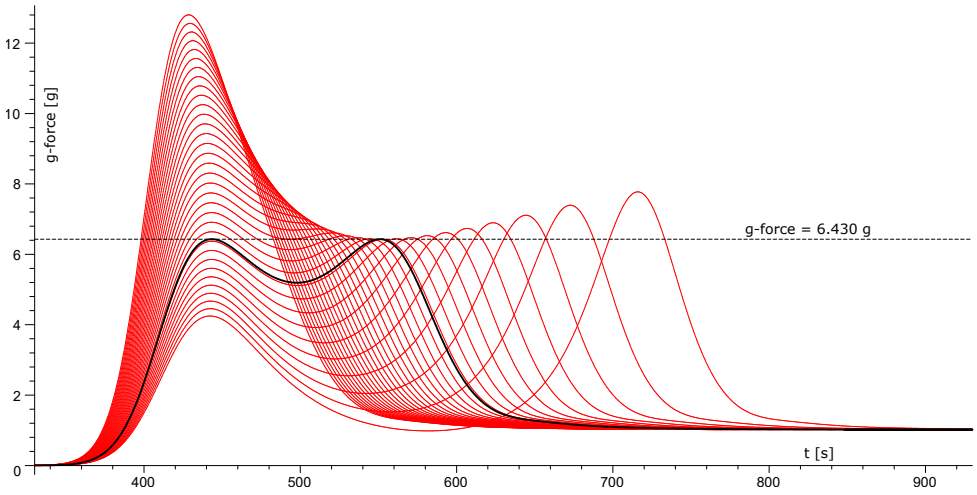



Figure 5: g-force as a function of the time.

landing will occur after two days. This means that a difference of one arcminute and twenty-nine arcseconds results in a full day extension of the flight time. This could be a significant complication for the spacecraft crew after separation from the service

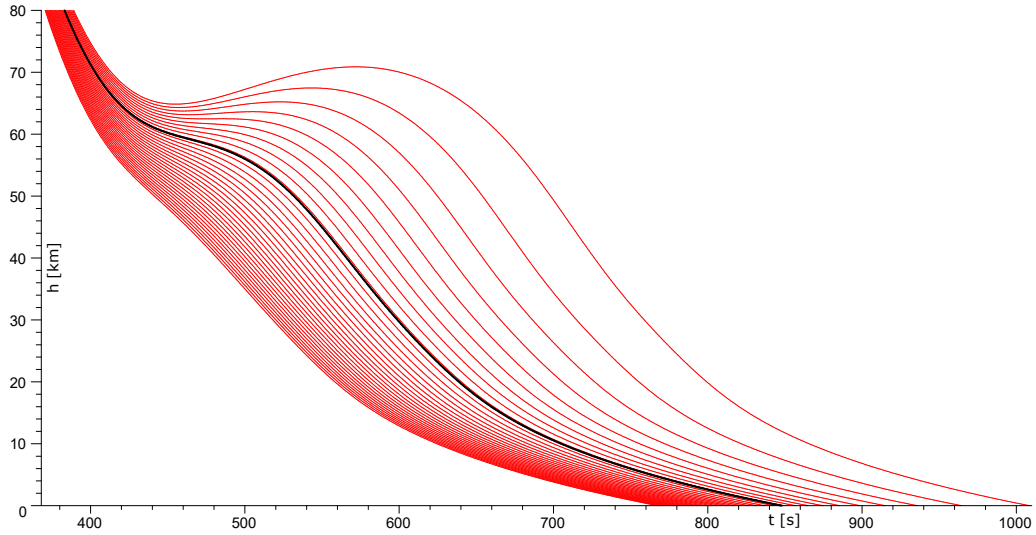



Figure 6: Flight height a function of the time.

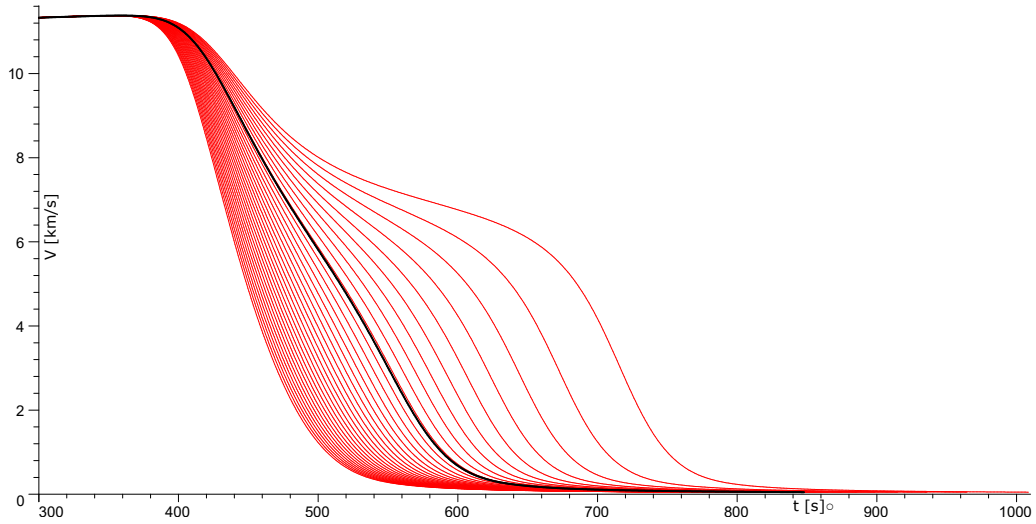



Figure 7: Flight velocity a function of the time.

module. Furthermore, increasing the deviation from the optimal angle leads to an exponential increase in landing time. If the spacecraft were to pass the initial point at a speed higher than the escape velocity, it would enter a solar orbit and never return to Earth. This is the case for spacecraft returning from interplanetary missions.

This means that the range of atmospheric entry angles is very narrow, approximately one quarter of a degree. Therefore, the accuracy and quality of mathematical modeling play a crucial role in solving this problem.

A Maple worksheet with all commands, including the generation of graphics, will be posted on the Maple application center in the near future.

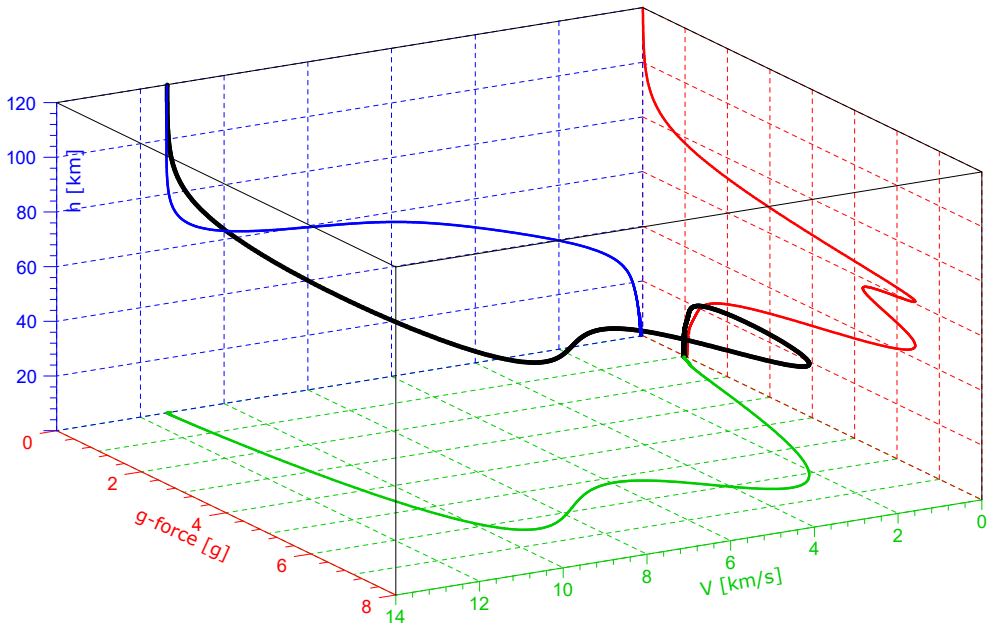



Figure 8: Landing dynamics.

## References


[1] Artemis I.: Online, `https://en.wikipedia.org/wiki/Artemis_I`, last access 14/09/2024.

[2] Artemis I Mission map. Online, `https://www.nasa.gov/mission/artemis-i/`, last access 14/09/2024.

[3] Barton, S.: Modeling of atmospheric motion and astrodynamics. Online, `https://www.maplesoft.com/Applications/Detail.aspx?id=4309`, last access 14/09/2024.

[4] CRC handbook of chemistry and physics. D. R. Linde, (Ed.) Tables 14–13 and 14–14. CRC Press, 72nd edition, 1991.

[5] Owens, D. B. and Aubuchon, V. V.: Overview of Orion crew module and launch abort vehicle dynamic stability. American Institute of Aeronautics and Astronautics. Online, `https://ntrs.nasa.gov/api/citations/20110015371/downloads/20110015371.pdf`, last access 14/09/2024.

[6] Raj, C. A. S., Narasimhavaradhan, M., Vaishnavi, N., Arunvinthan, S., Alarjani, A. A., and Pillai S. N.: Aerodynamics of ducted re-entry vehicles. Chinese Journal of Aeronautics **33** (2020), 1837–1849, doi.org/10.1016/j.cja.2020.02.019.

[7] *U.S. standard atmosphere 1976.* Online, `https://ntrs.nasa.gov/api/citations/19770009539/downloads/19770009539.pdf`,
last access 14/09/2024.

*Programs and Algorithms of Numerical Mathematics 22*
J. Chleboun, J. Papež, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics CAS, Prague 2025

# CONTINUOUS ADJOINT APPROACH TO SHAPE OPTIMIZATION WITH RESPECT TO 2D INCOMPRESSIBLE FLUID FLOW

Marek Brandner, Jiří Egermaier, Hana Kopincová

University of West Bohemia
Univerzitní 2732/8, 301 00, Pilsen, Czech Republic
brandner@kma.zcu.cz, jirieggy@kma.zcu.cz, kopincov@kma.zcu.cz

**Abstract:** The aim of this article is to briefly introduce the procedure for optimizing water turbine blades, which can lead to an innovative blade design and, consequently, an improvement in the desired properties of the water turbine, such as efficiency or the preferred pressure distribution on the blade. The computational method is based on formulating an objective function under certain constraint conditions, which are governed by the Navier-Stokes equations. This formulation enables the use of the Lagrange multiplier method, which incorporates the constraints into the augmented objective function. We derive the so-called adjoint problem, allowing us to simplify the gradient formulation for the chosen gradient-based optimization method.

**Keywords:** shape optimization, continuous adjoint

**MSC:** 49Q10, 49M41

## 1. Introduction

The problem of shape optimization, i.e., optimization where we try to find the optimal shape of a domain or part of it (e.g., water turbine blades), is a constrained optimization problem. It is necessary to prescribe an objective function (usually in integral form), constraint conditions (in our case, the equations describing the fluid flow), and a set of design parameters describing the optimized shape. Furthermore, for the optimization computational process itself, it is essential to determine the gradient of the objective function (required for any gradient-based optimization method), which includes the so-called shape derivative. For gradient computation, the continuous adjoint method is used, i.e., the adjoint problem is derived at first and then it is discretized. The derivation of the method and of all principles and ideas will be illustrated by a simplified two-dimensional model with laminar flow. The selected solver for solving the state and adjoint problems is described in detail in [3].

DOI: 10.21136/panm.2024.02

Consider the following optimization problem:

$$\min_{\mathbf{u},p,\Omega} F(\mathbf{u}, p, \Omega) \tag{1}$$

subject to incompressible steady-state Navier–Stokes equations (so-called primal or state equations)

$$R_i^u = -\frac{\partial \tau_{ij}}{\partial x_j} + u_j \frac{\partial u_i}{\partial x_j} + \frac{\partial p}{\partial x_i} = 0, \quad i = 1, 2 \qquad \mathbf{x} \in \Omega \subset \mathbb{R}^2, \tag{2}$$

$$R^p = \frac{\partial u_j}{\partial x_j} = 0, \qquad \mathbf{x} \in \Omega \subset \mathbb{R}^2, \tag{3}$$

where $u_i$ is a component of the velocity vector, $p := \frac{p}{\rho}$ is static pressure divided by the constant density of the liquid, and constant kinematic viscosity $\nu$ is considered in the stress tensor $\tau_{ij} = \nu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)$. The Lipschitz domain boundary $\partial\Omega := \Gamma$ consists of several disjoint parts: inflow $\Gamma_{\text{in}}$, outflow $\Gamma_{\text{out}}$, periodic parts $\Gamma_1$, $\Gamma_2$ and optimized (changing) part of the boundary $\Gamma_{\text{opt}}$ with the following boundary conditions:

$$\mathbf{u} = \mathbf{u}_{\text{in}}, \qquad \mathbf{x} \in \Gamma_{\text{in}}, \tag{4}$$

$$\mathbf{u} = \mathbf{0}, \qquad \mathbf{x} \in \Gamma_{opt}, \tag{5}$$

$$\mathbf{u}|_{\Gamma_1} = \mathbf{u}|_{\Gamma_2}, \qquad \mathbf{x} \in \Gamma_1, \Gamma_2 \tag{6}$$

$$p|_{\Gamma_1} = p|_{\Gamma_2}, \qquad \mathbf{x} \in \Gamma_1, \Gamma_2$$

$$\frac{\partial \mathbf{u}}{\partial \mathbf{n}}|_{\Gamma_1} = \frac{\partial \mathbf{u}}{\partial \mathbf{n}}|_{\Gamma_2}, \qquad \mathbf{x} \in \Gamma_1, \Gamma_2$$

$$\left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) n_j = 0, \quad i = 1, 2 \qquad \mathbf{x} \in \Gamma_{\text{out}}, \tag{7}$$

$$p = p_{\text{out}}, \qquad \mathbf{x} \in \Gamma_{\text{out}},$$

where $n_j$ is the $j$th component of the outward unit normal vector to the corresponding part of the boundary. $\mathbf{u}_{\text{in}}$ and $p_{\text{out}}$ are given functions and the Einstein convention, where repeated indices imply summation, is used.

The next approach is based on the method of Céa, see [4]. For optimization problems with equality constraints, it is appropriate to formulate the Lagrange function

$$L = F + \int_\Omega \lambda_i R_i^u \, \mathrm{d}\Omega + \int_\Omega \lambda_p R^p \, \mathrm{d}\Omega, \tag{8}$$

where for each flow (or state) variable $u_i$, $i = 1, 2$, and $p$ we define the so-called adjoint variables $\lambda_i$, $i = 1, 2$, and $\lambda_p$. Function $F$ will be described in Section 2.

Next, it is necessary to choose design variables $\mathbf{q} \in \mathbb{R}^{n_q}$. Complex shapes, such as a turbine blade, are suitably described by B-splines. This description is a linear combination of B-spline basis functions with coefficients known as control points, see [3]. Given the selected solver, we choose the set of the control points (more

precisely, coordinates of the control points) as our design parameters. Without loss of generality, we assume in the following text that the vector $\mathbf{q}$ has only one component, i.e., $\mathbf{q} = q$. To determine the shape gradient using the parametric approach, it is necessary to compute the total (or material) derivative of the Lagrange function (8) with respect to the chosen design variables

$$\frac{\mathrm{d}L}{\mathrm{d}q} = \frac{\mathrm{d}F}{\mathrm{d}q} + \frac{\mathrm{d}}{\mathrm{d}q}\int_\Omega \lambda_i R_i^u \,\mathrm{d}\Omega + \frac{\mathrm{d}}{\mathrm{d}q}\int_\Omega \lambda_p R^p \,\mathrm{d}\Omega. \tag{9}$$

Let us briefly summarize the relations between total (material) and partial derivatives both in the domain $\Omega$ and on the boundary $\Gamma$, for details see [1]. For an arbitrary quantity $I = I(\mathbf{u}, p)$ defined in $\Omega$ it holds (after using the Leibniz theorem)

$$\frac{\mathrm{d}}{\mathrm{d}q}\int_\Omega I \,\mathrm{d}\Omega = \int_\Omega \frac{\partial I}{\partial q}\,\mathrm{d}\Omega + \int_\Gamma I\,\frac{\mathrm{d}x_i}{\mathrm{d}q} n_i \,\mathrm{d}\Gamma, \tag{10}$$

where the partial derivative can be expressed by the chain rule $\frac{\partial I}{\partial q} = \frac{\partial I}{\partial u_i}\frac{\partial u_i}{\partial q} + \frac{\partial I}{\partial p}\frac{\partial p}{\partial q}$ in the first integral. The last boundary integral can be split into a sum over individual segments, but the term $\frac{\mathrm{d}x_i}{\mathrm{d}q}$ is zero everywhere except the optimized moving boundary, leaving only the integral over $\Gamma_{\mathrm{opt}}$. For an arbitrary quantity $J = J(\mathbf{u}, p)$ defined on the boundary $\Gamma$, it holds

$$\frac{\mathrm{d}}{\mathrm{d}q}\int_\Gamma J \,\mathrm{d}\Gamma = \int_\Gamma \frac{\mathrm{d}J}{\mathrm{d}q}\,\mathrm{d}\Gamma + \int_\Gamma J\,\frac{\mathrm{d}\,(\mathrm{d}\Gamma)}{\mathrm{d}q}, \tag{11}$$

$$\frac{\mathrm{d}J}{\mathrm{d}q} = \frac{\partial J}{\partial q} + \frac{\partial J}{\partial x_i} n_i \frac{\mathrm{d}x_j}{\mathrm{d}q} n_j \quad \text{and} \quad \frac{\mathrm{d}\,(\mathrm{d}\Gamma)}{\mathrm{d}q} = -\kappa \frac{\mathrm{d}x_i}{\mathrm{d}q} n_i \,\mathrm{d}\Gamma, \tag{12}$$

where $\kappa$ denotes the mean curvature of $\Gamma$ (can be derived using differential geometry, see [2]). We assume that the changes of design variables that produce changes of $\Gamma_{opt}$ in the tangent direction do not change the shape of the domain $\Omega$, therefore we consider only the normal component of the surface deformation. After the substitution of (12) into (11) we get

$$\frac{\mathrm{d}}{\mathrm{d}q}\int_\Gamma J \,\mathrm{d}\Gamma = \int_\Gamma \left(\frac{\partial J}{\partial u_i}\frac{\partial u_i}{\partial q} + \frac{\partial J}{\partial p}\frac{\partial p}{\partial q}\right)\,\mathrm{d}\Gamma + \int_\Gamma \frac{\partial J}{\partial x_j} n_j \frac{\mathrm{d}x_i}{\mathrm{d}q} n_i \,\mathrm{d}\Gamma - \int_\Gamma \kappa\, J \,\frac{\mathrm{d}x_i}{\mathrm{d}q} n_i \,\mathrm{d}\Gamma \tag{13}$$

and again the last two boundary integrals are nonzero only on $\Gamma_{\mathrm{opt}}$ and again we used the chain rule in the first integral.

Now we continue with (9) and after using (10) with $I = \lambda R$ and since the adjoint variables are independent of the flow variables, we arrive at

$$\frac{\mathrm{d}L}{\mathrm{d}q} = \frac{\mathrm{d}F}{\mathrm{d}q} + \int_\Omega \lambda_i \frac{\partial R_i^u}{\partial q}\,\mathrm{d}\Omega + \int_\Omega \lambda_p \frac{\partial R^p}{\partial q}\,\mathrm{d}\Omega + \int_{\Gamma_{\mathrm{opt}}} \left(\lambda_j R_j^u + \lambda_p R^p\right)\,\frac{\mathrm{d}x_i}{\mathrm{d}q} n_i \,\mathrm{d}\Gamma. \tag{14}$$

The next step is to determine the total derivative of the objective function.

## 2. Objective function

The overall objective function, which is mentioned in (1), can be considered as an appropriate weighted combination of multiple components. In this text, we will introduce four components of the objective function $F_1$, $F_2$, $F_3$, $F_4$, so that:

$$F = w_1 F_1 + w_2 F_2 + w_3 F_3 + w_4 F_4. \tag{15}$$

1. The function $F_1$ quantifies the effect of the head. By optimizing this function, we achieve a minimal difference between the target head $H_{\text{tar}}$ and the actual head $H$. The function $F_1$ is prescribed on the inflow and outflow part of the boundary, $\Gamma_{\text{in}}$ and $\Gamma_{\text{out}}$. It is defined as follows:

$$F_1 = \frac{1}{2}\left(\frac{H - H_{\text{tar}}}{H_{\text{tar}}}\right)^2, \tag{16}$$

where the head $H$ is defined as follows:

$$H = \frac{1}{\rho g S_{\text{in}}} \int_{\Gamma_{\text{in}}} p_{\text{tot,in}} \,\mathrm{d}\Gamma - \frac{1}{\rho g S_{\text{out}}} \int_{\Gamma_{\text{out}}} p_{\text{tot,out}} \,\mathrm{d}\Gamma, \tag{17}$$

$$\text{for} \quad p_{\text{tot}} = p_{\text{stat}} + \frac{1}{2}\rho v^2,\ p_{\text{stat}} = \rho\, p,\ v = \frac{Q}{S} = \text{const.},\ Q = \int_{\Gamma_{\text{in}}} \mathbf{u}_{\text{in}} \cdot \mathbf{n} \,\mathrm{d}\Gamma, \tag{18}$$

where $p_{\text{stat}}$ is static pressure and $p$ kinematic pressure, further $\rho$ denotes the density of the liquid, $g$ gravitational acceleration, $Q$ is the flow rate and $S$ is the length of the respective boundary segment.

After some easy manipulations we can see that the only flow variable on which this term depends is the pressure $p$. Thus, after a straightforward differentiation of the composite function and using (13), we get

$$\frac{\mathrm{d}F_1}{\mathrm{d}q} = \left(\frac{H}{H_{\text{tar}}} - 1\right) \frac{1}{H_{\text{tar}}} \left[ \int_{\Gamma_{\text{in}}} \frac{\partial f_{1,\text{in}}(p)}{\partial p} \frac{\partial p}{\partial q} \,\mathrm{d}\Gamma - \int_{\Gamma_{\text{out}}} \frac{\partial f_{1,\text{out}}(p)}{\partial p} \frac{\partial p}{\partial q} \,\mathrm{d}\Gamma \right], \tag{19}$$

where

$$f_{1,\text{in(out)}}(p) = \frac{1}{\rho g S_{\text{in(out)}}} \left(p + \frac{1}{2}\rho v^2\right), \quad \frac{\partial f_{1,\text{in(out)}}(p)}{\partial p} = \frac{1}{\rho g S_{\text{in(out)}}}. \tag{20}$$

2. The function $F_2$ is related to the efficiency of the water turbine. The ideal state is 100% efficiency, and therefore, we will minimize the deviation from this ideal state. Thus, we define the function $F_2$ as follows:

$$F_2 = 1 - \frac{M\omega}{Q\rho g H}, \tag{21}$$

where $\omega = \text{const.}$ denotes the angular velocity, and its value is prescribed by the real situation. The torque $M$, which acts on the turbine blade, i.e. $\Gamma_{\text{opt}}$, is defined as follows:

$$M = N \int_{\Gamma_{\text{opt}}} \mathbf{M} \cdot \mathbf{e} \, \mathrm{d}\Gamma, \quad \text{where} \quad \mathbf{M} = \mathbf{r} \times \mathbf{F}, \quad \mathbf{F} = \mathbf{n} \, p_{\text{stat}} \tag{22}$$

and $\mathbf{e}$ is the direction of the axis of rotation, $N$ is the number of blades, $\mathbf{F}$ denotes the force acting on the blade, $\mathbf{r}$ is the position vector perpendicular to the axis of rotation, and $\mathbf{n}$ is the normal vector pointing outward from the suction side. The above formulas are valid for 3D calculations. For our simplified 2D model, we choose $\mathbf{e} = (1, 0, 0)$, $\mathbf{r} = (0, 0, 1)$, $\omega = 1$ and $N = 5$.

If we substitute the formulas (22) and (17) into (21), then again the resulting expression depends only on the pressure $p$, but this time there are integrals over $\Gamma_{\text{in}}$, $\Gamma_{\text{out}}$ and $\Gamma_{\text{opt}}$. Thus, after differentiation of the quotient and using (13) we obtain

$$\begin{aligned} \frac{\mathrm{d}F_2}{\mathrm{d}q} = &- \frac{N\omega}{Q\rho g H} \left[ \int_{\Gamma_{\text{opt}}} \frac{\partial f_{2,\text{opt}}(p)}{\partial p} \frac{\partial p}{\partial q} \, \mathrm{d}\Gamma + \int_{\Gamma_{\text{opt}}} \left( \frac{\partial f_{2,\text{opt}}}{\partial x_j} n_j - \kappa_{\text{opt}} f_{2,\text{opt}} \right) \frac{\mathrm{d}x_i}{\mathrm{d}q} n_i \, \mathrm{d}\Gamma \right] \\ &+ \frac{N\omega}{Q\rho g H^2} \int_{\Gamma_{\text{opt}}} f_{2,\text{opt}}(p) \, \mathrm{d}\Gamma \left[ \int_{\Gamma_{\text{in}}} \frac{\partial f_{1,\text{in}}(p)}{\partial p} \frac{\partial p}{\partial q} \, \mathrm{d}\Gamma - \int_{\Gamma_{\text{out}}} \frac{\partial f_{1,\text{out}}(p)}{\partial q} \frac{\partial p}{\partial q} \, \mathrm{d}\Gamma \right], \end{aligned} \tag{23}$$

where

$$f_{2,\text{opt}}(p) = (\mathbf{r} \times \mathbf{n}) \cdot \mathbf{e} \, p = n_2 p, \quad \frac{\partial f_{2,\text{opt}}(p)}{\partial p} = (\mathbf{r} \times \mathbf{n}) \cdot \mathbf{e} = n_2. \tag{24}$$

3. The function $F_3$ represents the pressure distribution on the blade. The optimization aims to match this distribution as closely as possible to the target pressure, $p_{\text{tar}}$. Hence, $F_3$ is defined as

$$F_3 = \frac{1}{2} \int_{\Gamma_{\text{opt}}} \frac{(p - p_{\text{tar}})^2}{p_{\text{tar}}^2} \, \mathrm{d}\Gamma. \tag{25}$$

In this function, the dependence on pressure is clear, so the total derivative with respect to $q$ is determined using the same procedure as before and we get

$$\frac{\mathrm{d}F_3}{\mathrm{d}q} = \int_{\Gamma_{\text{opt}}} \frac{\partial f_{3,\text{opt}}(p)}{\partial p} \frac{\partial p}{\partial q} \, \mathrm{d}\Gamma + \int_{\Gamma_{\text{opt}}} \left( \frac{\partial f_{3,\text{opt}}(p)}{\partial x_j} n_j - \kappa_{\text{opt}} f_{3,\text{opt}}(p) \right) \frac{\mathrm{d}x_i}{\mathrm{d}q} n_i \, \mathrm{d}\Gamma, \tag{26}$$

where

$$f_{3,\text{opt}}(p) = \frac{1}{2} \frac{(p - p_{\text{tar}})^2}{p_{\text{tar}}^2}, \quad \frac{\partial f_{3,\text{opt}}(p)}{\partial p} = \frac{(p - p_{\text{tar}})}{p_{\text{tar}}^2}. \tag{27}$$

4. The final part of the objective function, $F_4$, minimizes the difference between the outflow boundary velocity and a given target outflow velocity $\mathbf{u}_{\text{tar}}$. This prevents undesirable turbulence behind the runner, thereby improving overall efficiency. Thus, $F_4$ is defined as:

$$F_4 = \frac{1}{2} \int_{\Gamma_{\text{out}}} \frac{\|\mathbf{u} - \mathbf{u}_{\text{tar}}\|^2}{\|\mathbf{u}_{\text{tar}}\|} \, \mathrm{d}\Gamma. \tag{28}$$

This function is the only one dependent on the flow variables $u_i$. Using the same procedure as before we get

$$\frac{\mathrm{d}F_4}{\mathrm{d}q} = \int_{\Gamma_{\text{out}}} \frac{\partial f_{4,\text{out}}(\mathbf{u})}{\partial u_i} \frac{\partial u_i}{\partial q} \, \mathrm{d}\Gamma, \tag{29}$$

where

$$f_{4,\text{out}}(\mathbf{u}) = \frac{1}{2} \frac{\|\mathbf{u} - \mathbf{u}_{\text{tar}}\|^2}{\|\mathbf{u}_{\text{tar}}\|}, \quad \frac{\partial f_{4,\text{out}}(\mathbf{u})}{\partial u_i} = \frac{u_i - u_{i,\text{tar}}}{\|\mathbf{u}_{\text{tar}}\|}. \tag{30}$$

## 3. Adjoint problem derivation

For the derivation of the adjoint problem, the expressions (19), (23), (26), (29), (2) and (3) are substituted into (14), the interchangeability of derivatives is used, i.e. $\frac{\partial}{\partial q}\frac{\partial J}{\partial x_i} = \frac{\partial}{\partial x_i}\frac{\partial J}{\partial q}$ for any function $J$, and the Green-Gauss theorem is applied. After appropriate term rearranging and relabeling to simplify the formulas, and noting that $\tau_{ij}^a = \nu\left(\frac{\partial \lambda_i}{\partial x_j} + \frac{\partial \lambda_j}{\partial x_i}\right)$ is representing the adjoint stress tensor and again the Einstein convention is used, we arrive at

$$\begin{aligned}
\frac{\mathrm{d}L}{\mathrm{d}q} &= \underbrace{\left(\frac{H}{H_{\text{tar}}} - 1\right)\frac{w_1}{H_{\text{tar}}}}_{\mathrm{C}_1} \left[\int_{\Gamma_{\text{in}}} \frac{\partial f_{1,\text{in}}(p)}{\partial p}\frac{\partial p}{\partial q}\,\mathrm{d}\Gamma - \int_{\Gamma_{\text{out}}} \frac{\partial f_{1,\text{out}}(p)}{\partial p}\frac{\partial p}{\partial q}\,\mathrm{d}\Gamma\right] \\
&- \underbrace{\frac{w_2 N\omega}{Q\rho g H}}_{\mathrm{C}_2} \left[\int_{\Gamma_{\text{opt}}} \frac{\partial f_{2,\text{opt}}(p)}{\partial p}\frac{\partial p}{\partial q}\,\mathrm{d}\Gamma + \int_{\Gamma_{\text{opt}}} \left(\frac{\partial f_{2,\text{opt}}}{\partial x_j} n_j - \kappa_{\text{opt}} f_{2,\text{opt}}\right)\frac{\mathrm{d}x_i}{\mathrm{d}q} n_i \,\mathrm{d}\Gamma\right] \\
&+ \underbrace{\frac{w_2 N\omega}{Q\rho g H^2}\int_{\Gamma_{\text{opt}}} f_{2,\text{opt}}(p)\,\mathrm{d}\Gamma}_{\mathrm{C}_3} \left[\int_{\Gamma_{\text{in}}} \frac{\partial f_{1,\text{in}}(p)}{\partial p}\frac{\partial p}{\partial q}\,\mathrm{d}\Gamma - \int_{\Gamma_{\text{out}}} \frac{\partial f_{1,\text{out}}(p)}{\partial q}\frac{\partial p'}{\partial q},\mathrm{d}\Gamma\right] \\
&+ w_3 \int_{\Gamma_{\text{opt}}} \frac{\partial f_{3,\text{opt}}(p)}{\partial p}\frac{\partial p}{\partial q}\,\mathrm{d}\Gamma + w_3 \int_{\Gamma_{\text{opt}}} \left(\frac{\partial f_{3,\text{opt}}(p)}{\partial x_j} n_j - \kappa_{\text{opt}} f_{3,\text{opt}}(p)\right)\frac{\mathrm{d}x_i}{\mathrm{d}q} n_i \,\mathrm{d}\Gamma \\
&+ w_4 \int_{\Gamma_{\text{out}}} \frac{\partial f_{4,\text{out}}(\mathbf{u})}{\partial u_i}\frac{\partial u_i}{\partial q}\,\mathrm{d}\Gamma + \int_{\Omega} \frac{\partial \tau_{ij}^a}{\partial x_j}\frac{\partial u_i}{\partial q}\,\mathrm{d}\Omega + \int_{\Omega} \lambda_j \frac{\partial u_j}{\partial x_i}\frac{\partial u_i}{\partial q}\,\mathrm{d}\Omega
\end{aligned}$$

$$
\begin{aligned}
&- \int_\Omega u_j \frac{\partial \lambda_i}{\partial x_j} \frac{\partial u_i}{\partial q} \,\mathrm{d}\Omega - \int_\Omega \frac{\partial \lambda_p}{\partial x_i} \frac{\partial u_i}{\partial q} \,\mathrm{d}\Omega - \int_\Omega \frac{\partial \lambda_j}{\partial x_j} \frac{\partial p}{\partial q} \,\mathrm{d}\Omega + \int_\Gamma \tau_{ij}^a n_j \frac{\partial u_i}{\partial q} \,\mathrm{d}\Gamma \\
&+ \int_\Gamma u_j n_j \lambda_i \frac{\partial u_i}{\partial q} \,\mathrm{d}\Gamma + \int_\Gamma \lambda_p n_i \frac{\partial u_i}{\partial q} \,\mathrm{d}\Gamma - \int_\Gamma \frac{\partial \tau_{ij}}{\partial q} n_j \lambda_i \,\mathrm{d}\Gamma + \int_\Gamma \lambda_i n_i \frac{\partial p}{\partial q} \,\mathrm{d}\Gamma \\
&+ \int_{\Gamma_{\mathrm{opt}}} \left(\lambda_j R_j^u + \lambda_p R^p\right) \frac{\mathrm{d}x_i}{\mathrm{d}q} n_i \,\mathrm{d}\Gamma. \qquad (31)
\end{aligned}
$$

First of all, we will focus on the volume integrals in (31). It is useful to avoid calculations the derivatives of the flow variables with respect to the design parameters, i.e., $\frac{\partial u_i}{\partial q}$ and $\frac{\partial p}{\partial q}$. This can be achieved by setting all the terms that involve these derivatives to zero. This leads to the adjoint set of equations

$$
R_i^\lambda = -\frac{\partial \tau_{ij}^a}{\partial x_j} + \lambda_j \frac{\partial u_j}{\partial x_i} - u_j \frac{\partial \lambda_i}{\partial x_j} - \frac{\partial \lambda_p}{\partial x_i} = 0, \quad i = 1, 2, \qquad \mathbf{x} \in \Omega \subset \mathbb{R}^2, \qquad (32)
$$

$$
R^{\lambda_p} = \frac{\partial \lambda_j}{\partial x_j} = 0, \qquad \mathbf{x} \in \Omega \subset \mathbb{R}^2. \qquad (33)
$$

Thus, only the boundary integrals remain and it is necessary to set the correct boundary conditions in order to reduce the number of integrals as much as possible.

### 3.1. Boundary conditions for the adjoint problem

Recall that $\Gamma = \Gamma_{\mathrm{in}} \cup \Gamma_1 \cup \Gamma_2 \cup \Gamma_{\mathrm{opt}} \cup \Gamma_{\mathrm{out}}$, i.e. the boundary integral over the entire boundary is a sum of integrals over the individual parts of the boundary:

1. $\Gamma_{\mathrm{in}}$: For the inlet boundary we set (4) and, therefore, it is easy to see that $\frac{\partial u_i}{\partial q} = 0$ and $\frac{\partial \tau_{ij}}{\partial q} = 0$ holds. Thus, only the following nonzero integrals over inlet boundary remain in (31) (again after appropriate term rearranging and factoring out)

$$
\int_{\Gamma_{\mathrm{in}}} \left[(C_1 + C_3) \frac{\partial f_{1,\mathrm{in}}(p)}{\partial p} + \lambda_i n_i\right] \frac{\partial p}{\partial q} \,\mathrm{d}\Gamma. \qquad (34)
$$

   So we set

$$
\lambda_i n_i = -(C_1 + C_3) \frac{\partial f_{1,\mathrm{in}}(p)}{\partial p}, \quad \lambda_i t_i = 0, \quad \mathbf{x} \in \Gamma_{\mathrm{in}}, \qquad (35)
$$

   to set the integral (34) to zero.

2. $\Gamma_{\mathrm{out}}$: We set the conditions (7) for the flow variables, thus for differentiation w.r.t. design parameters at the outflow boundary it holds

$\left(\frac{\partial}{\partial x_j}\frac{\partial u_i}{\partial q} + \frac{\partial}{\partial x_i}\frac{\partial u_j}{\partial q}\right) n_j = 0$, $i = 1, 2$ and $\frac{\partial p}{\partial q} = 0$. Thus, it is necessary to handle only the following nonzero integrals in (31) (again after appropriate term rearranging and factoring out)

$$\int_{\Gamma_{\text{out}}} \left( w_4 \frac{\partial f_{4,\text{out}}(\mathbf{u})}{\partial u_i} + \tau_{ij}^a n_j + u_j n_j \lambda_i + \lambda_p n_i \right) \frac{\partial u_i}{\partial q} \, \mathrm{d}\Gamma. \tag{36}$$

So if we set

$$\tau_{ij}^a n_j + u_j n_j \lambda_i + \lambda_p n_i = -w_4 \frac{\partial f_{4,\text{out}}(\mathbf{u})}{\partial u_i}, \quad i = 1, 2, \quad \mathbf{x} \in \Gamma_{\text{out}}, \tag{37}$$

all the integrals over the output boundary vanish from (31).

3. $\Gamma_1$, $\Gamma_2$: For the periodic boundaries, none of the boundary integrals is equal to zero, so we obtain

$$\begin{aligned} &\int_{\Gamma_1} \left( \tau_{ij}^a n_j \frac{\partial u_i}{\partial q} + u_j n_j \lambda_i \frac{\partial u_i}{\partial q} + \lambda_p n_i \frac{\partial u_i}{\partial q} - \frac{\partial \tau_{ij}}{\partial q} n_j \lambda_i + \lambda_i n_i \frac{\partial p}{\partial q} \right) \mathrm{d}\Gamma + \\ &\int_{\Gamma_2} \left( \tau_{ij}^a n_j \frac{\partial u_i}{\partial q} + u_j n_j \lambda_i \frac{\partial u_i}{\partial q} + \lambda_p n_i \frac{\partial u_i}{\partial q} - \frac{\partial \tau_{ij}}{\partial q} n_j \lambda_i + \lambda_i n_i \frac{\partial p}{\partial q} \right) \mathrm{d}\Gamma. \end{aligned} \tag{38}$$

For periodic boundary it holds that $\Gamma_2 = T(\Gamma_1)$ is a translational copy of $\Gamma_1$ under a map $T$ with the opposite normal vector with respect to $\Gamma_1$ at the corresponding points of both boundaries. Each pair of integrals for $\Gamma_1$ and the same one for $\Gamma_2$ vanishes if we set

$$\begin{aligned} \tau_{ij}^a(\mathbf{x}) &= \tau_{ij}^a(T(\mathbf{x})), && \mathbf{x} \in \Gamma_1, \\ \lambda_i(\mathbf{x}) &= \lambda_i(T(\mathbf{x})), \quad i = 1, 2, && \mathbf{x} \in \Gamma_1, \\ \lambda_p(\mathbf{x}) &= \lambda_p(T(\mathbf{x})), && \mathbf{x} \in \Gamma_1. \end{aligned} \tag{39}$$

4. $\Gamma_{\text{opt}}$: Optimized boundary is the only one which is assumed to be moving. For velocity, we set homogeneous boundary condition (5), so total derivation w.r.t. $q$ is equal to zero. Thus, using (12) we obtain the following expression for the partial derivative w.r.t. $q$

$$\frac{\partial u_i}{\partial q} = -\frac{\partial u_i}{\partial x_k} n_k \frac{\mathrm{d}x_l}{\mathrm{d}q} n_l, \quad i = 1, 2. \tag{40}$$

Since no terms vanish on this boundary, thus in (31) we can take care only for the following integral

$$\int_{\Gamma_{\text{opt}}} \left( -C_2 \frac{\partial f_{2,\text{opt}}(p)}{\partial p} + w_3 \frac{\partial f_{3,\text{opt}}(p)}{\partial p} + \lambda_i n_i \right) \frac{\partial p}{\partial q} \, \mathrm{d}\Gamma. \tag{41}$$

For vanishing of (41), we set

$$\lambda_i n_i = C_2 \frac{\partial f_{2,\text{opt}}(p)}{\partial p} - w_3 \frac{\partial f_{3,\text{opt}}(p)}{\partial p}, \quad \lambda_i t_i = 0, \quad \mathbf{x} \in \Gamma_{\text{opt}}. \tag{42}$$

## 4. Gradient

After setting the boundary conditions as described above and substituting (40) into (31), we obtain the expression for the gradient in the form

$$\begin{aligned}\frac{\mathrm{d}L}{\mathrm{d}q} &= C_2 \int_{\Gamma_{\mathrm{opt}}} \left( \frac{\partial f_{2,\mathrm{opt}}}{\partial x_j} n_j - \kappa_{\mathrm{opt}} f_{2,\mathrm{opt}} \right) \frac{\mathrm{d}x_i}{\mathrm{d}q} n_i \,\mathrm{d}\Gamma + w_3 \int_{\Gamma_{\mathrm{opt}}} \left( \frac{\partial f_{3,\mathrm{opt}}(p)}{\partial x_j} n_j - \right. \\ &- \left. \kappa_{\mathrm{opt}} f_{3,\mathrm{opt}}(p) \right) \frac{\mathrm{d}x_i}{\mathrm{d}q} n_i \,\mathrm{d}\Gamma - \int_{\Gamma_{\mathrm{opt}}} \left( \tau_{ij}^a n_j + u_j n_j \lambda_i + \lambda_p n_i \right) \frac{\partial u_i}{\partial x_k} n_k \frac{\mathrm{d}x_l}{\mathrm{d}q} n_l \,\mathrm{d}\Gamma \\ &- \int_{\Gamma_{\mathrm{opt}}} \frac{\partial \tau_{ij}}{\partial q} n_j \lambda_i \,\mathrm{d}\Gamma + \int_{\Gamma_{\mathrm{opt}}} \left( \lambda_j R_j^u + \lambda_p R^p \right) \frac{\mathrm{d}x_i}{\mathrm{d}q} n_i \,\mathrm{d}\Gamma, \end{aligned} \tag{43}$$

where the term $\frac{\partial \tau_{ij}}{\partial q} n_j \lambda_i$ can be rewritten as follows (after substituting (40) into the stress tensor and considering that the boundary conditions (42) were set for $\lambda_i$ on $\Gamma_{\mathrm{opt}}$ boundary and tedious computation)

$$\frac{\partial \tau_{ij}}{\partial q} n_j \lambda_i = \lambda_i n_i \left( \tau_{ij} \frac{\mathrm{d}(n_i n_j)}{\mathrm{d}q} + \frac{\partial \tau_{ij}}{\partial x_m} n_m \frac{\mathrm{d}x_k}{\mathrm{d}q} n_k n_i n_j \right). \tag{44}$$

The numerical computation proceeds as follows: we set the initial shape of the blade, i.e., the boundary $\Gamma_{\mathrm{opt}}$, and solve the primal problem (2) and (3) with the boundary conditions (4), (5), (6), (7). This provides the state variables $u_1$, $u_2$ and $p$. The adjoint quantities $\lambda_1$, $\lambda_2$ and $\lambda_p$ are obtained by solving the adjoint problem (32) and (33) with the boundary conditions (35), (37), (39) and (42). Then, the gradient is computed by using the equation (43) and (44) and the shape of the blade is adjusted by using any gradient-based method (here, for simplicity, the steepest descent method is used).

## 5. Numerical experiment

We test our optimization approach on the simplified problem of flow in a domain which is a part of a 2D blade cascade. This cascade is obtained by unfolding a cylindrical cross-section of the turbine and it is illustrated in Figure 1 (left). The computational domain is then a passage between two blade profiles, see Figure 1 (right). The domain consists of three B-spline patches of degree 3. $\Gamma_{\mathrm{opt}}$ corresponds to the upper (suction side) and lower (pressure side) boundaries of the middle patch which form the blade profile. Left and right patches are bounded by periodic boundaries $\Gamma_1$ and $\Gamma_2$ and inlet boundary (the left-most) $\Gamma_{in}$ and outlet boundary (the right-most) $\Gamma_{\mathrm{out}}$. We use $\mathbf{u}_{in} = [7.76, -0.28]$ on the inlet and kinematic viscosity $\nu = 0.01$ in this example.

The objective function is defined by its components and corresponding weights in (15). In this example, we use the following weights

$$w_1 = 1, \quad w_2 = 1.8, \quad w_3 = 1, \quad w_4 = 0.2,$$

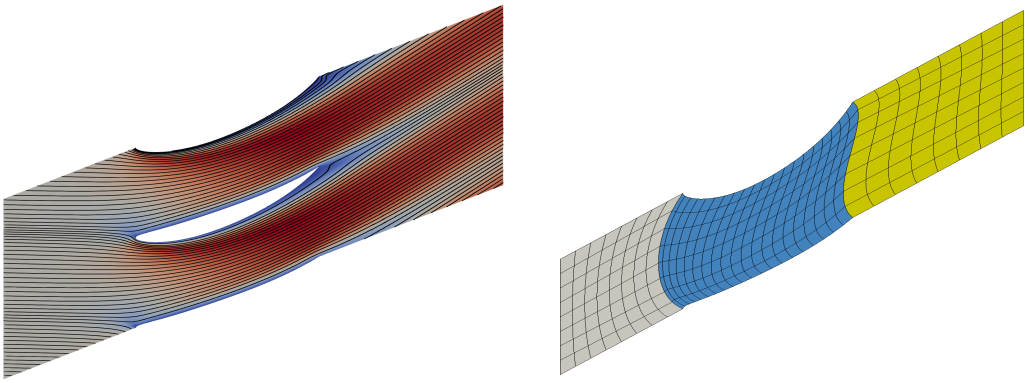


Figure 1: Illustration of the flow in the blade raw (left). Computational domain (right).

which prefers the efficiency component. The target values of pressure ($p_{\text{tar}}$) on pressure and suction sides of the blade profile and velocity ($\mathbf{u}_{\text{tar}}$) at the output are equal to the integral mean values of particular quantities over the corresponding boundary for the initial geometry. For simplicity we use steepest descent method with constant step $\gamma = 5 \cdot 10^{-4}$. Therefore the control points of B-spline curves describing both parts of $\Gamma_{\text{opt}}$ are updated during the iteration process by the formula

$$\mathbf{q}^{\text{new}} = \mathbf{q} - \gamma \nabla_{\mathbf{q}} L, \tag{45}$$

where the length of the vector $\mathbf{q}$ is 42.The optimization is stopped after 40 iterations.

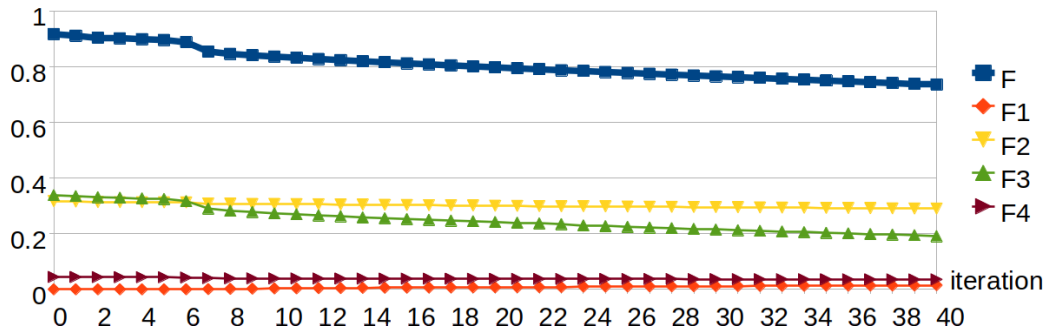



Figure 2: Evolution of the objective function and its components.

The values of objective function and its components are shown in Figure 2. We can see that the objective function as well as its individual components are decreasing, except for $F_1$. That is obvious, because $H_{\text{tar}}$ is defined as the head of the problem with the initial geometry and therefore $F_1 = 0$ for the initial geometry.

The comparison of the initial blade profile and the optimized one is shown in

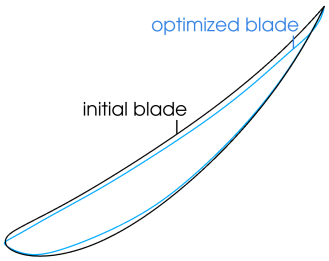



Figure 3: Initial and optimized blade.

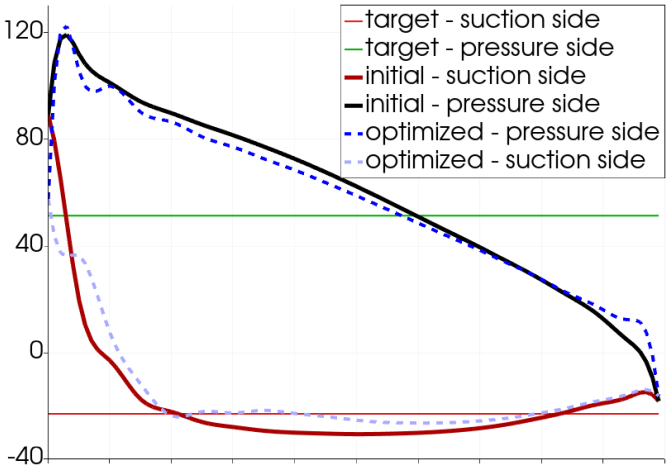



Figure 4: Pressure distribution over the blade profile.

Figure 3. The initial and optimized pressure distribution over the profile is shown in Figure 4 together with the values of the pressure targets.

## 6. Conclusion

In conclusion, this method shows great promise for further development (especially into 3D and turbulent flow models), as even in the presented simple 2D model with laminar flow and a basic optimization method, it was able to reduce the objective function and adjust the blade in a meaningful way.

## Acknowledgements

This work was supported by Technology Agency of the Czech Republic (TA CR) grant No. TK04020250.

## References


[1] Papoutsis-Kiachagias, E. M. and Giannakoglou, K. C.: *Continuous adjoint methods for turbulent flows, applied to shape and topology optimization: industrial applications*. Arch. Comput. Methods in Eng., Vol. 23, 2016, pp. 255–299. https://doi.org/10.1007/s11831-014-9141-9.

[2] Grinfeld, P.: Hadamard's formula inside and out. J. Optim. Theory Appl. **146** (2010), 654–690. https://doi.org/10.1007/s10957-010-9681-6.

[3] Bastl, B., Brandner, M., Egermaier, J., Michálková, M., Turnerová, E.: IgA-Based Solver for turbulence modelling on multipatch geometries. Adv. Eng. Softw. **113** (2017), 7–18.
https://doi.org/10.1016/j.advengsoft.2017.06.012.

[4] Allaire, G., Dapogny, Ch., Jouve, F.: *Chapter 1 - Shape and topology optimization*. Handbook of Numerical Analysis: Geometric partial differential equations - part II, Volume 22, Elsevier, 2021, pp. 1–132.
https://doi.org/10.1016/bs.hna.2020.10.004.

*Programs and Algorithms of Numerical Mathematics 22*
J. Chleboun, J. Papež, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics CAS, Prague 2025

# PERFORMANCE OF PARALLEL QR FACTORIZATION METHODS ON THE NVIDIA GRACE CPU SUPERCHIP

Vít Břichňáč[1], Jakub Šístek[1,2]

[1] Czech Technical University in Prague
Jugoslávských partyzánů 1580/3, 160 00 Prague 6 - Dejvice, Czech Republic
brichvit@cvut.cz
[2] Institute of Mathematics of the Czech Academy of Sciences
Žitná 25, 115 67 Prague 1 - Nové Město, Czech Republic
sistek@math.cas.cz

**Abstract:** This article studies several algorithms for QR factorization based on hierarchical Householder reflectors organized into elimination trees, which are particularly suited for tall-and-skinny matrices and allow parallelization. We examine the effect of various parameters on the performance of the tree-based algorithms. The work is accompanied with a custom implementation that utilizes a task-based runtime system (OpenMP or StarPU). The same algorithm is implemented in the PLASMA library. The performance evaluation is done on the recent NVIDIA Grace CPU Superchip.

**Keywords:** QR factorization, task-based programming, NVIDIA Grace CPU
**MSC:** 65F05

## 1. Introduction

The need for computing the QR factorization of dense matrices with substantially more rows than columns (so-called tall-and-skinny matrices) arises in a number of applications, for example, when solving overdetermined systems of linear equations by the least-squares method or as a preprocessing step for the SVD algorithm used in reduced order modeling.

Modern algorithms for computing the QR factorization of a matrix using orthogonal triangularization by Householder reflectors split the matrix into blocks and then perform operations on those blocks. Importantly, the parallel TSQR and CAQR algorithms of [7] opened the way to parallelizing the panel factorization and hence for deriving parallel algorithms for tall-and-skinny matrices. These algorithms are implemented for example in the ScaLAPACK[1] [3] and PLASMA[2] [5] libraries. Other recent approaches include numerically stable variants of triangular orthogonalization using Cholesky QR [11, 12] or randomized QR factorization methods, see, e.g., [13].

DOI: 10.21136/panm.2024.03
[1] http://www.netlib.org/scalapack
[2] https://icl.utk.edu/plasma

The algorithms work with several parameters (e.g., the block size, inner block size, etc.) that do not influence the result but have an impact on the computation time [4, 10]. In our paper [6], we present a new version of the algorithm for QR factorization based on tasks implemented in the OpenMP version of PLASMA [9] and perform a study of the effect of the main algorithmic parameters on performance on several multicore CPU architectures by Intel, AMD, and Arm. In light of [7], the algorithm can be seen as a combination of the parallel and sequential versions of the Communication-avoiding QR (CAQR) algorithm, with the latter performed on the leaves of the tree arising from the former.

The main purpose of the present article is to complement the experiments from [6] with performance measurements on the NVIDIA Grace CPU Superchip, another recent multicore chip based on the Arm architecture. The reader is referred to [6] for a more detailed description of the algorithm.

## 2. QR factorization

QR factorization is a matrix decomposition of a matrix $A \in \mathbb{R}^{m,n}$ into a product $QR$, where $Q \in \mathbb{R}^{m,m}$ is an orthogonal matrix and $R^{m,n}$ is an upper trapezoidal matrix. Many different methods may be used for computing the QR factorization of a matrix, but for developing parallel algorithms for QR computation, the Householder reflector method is of particular interest.

It works by applying a series of orthogonal transformations $Q_1, Q_2, \ldots, Q_k$ for $k = \min(m, n)$ on an arbitrary matrix $A \in \mathbb{R}^{m,n}$, where each of the transformations $Q_i$:

- zeros out the vector $A_{(i+1):m,i}$ using the entry $A_{i,i}$ by a reflection in a subspace corresponding to the last $m - i + 1$ rows of $A$, and
- functions as the identity transformation in the subspace corresponding to the first $m - 1$ rows.

As a result, the matrix $R = Q_k Q_{k-1} \ldots Q_1 A$ is upper triangular, and the QR decomposition of $A$ can be formed as $A = QR$, where $Q = Q_1^T Q_2^T \ldots Q_k^T$.

## 3. Elimination schemes

### 3.1. Column block Householder reflector algorithm

To promote BLAS Level 3 operations in the application of the Householder reflectors, matrix columns can be grouped into column blocks as in the LAPACK library[3] [1]. This column-blocking also opens a way to parallelize the algorithm, since each block column of the updated matrix can be updated independently. In particular, the algorithm performs the following steps for each column block:

1. Factorize the column block into upper triangular form using a **block Householder reflector**.
2. Apply the calculated reflector to subsequent column blocks (potentially in parallel).

[3] http://www.netlib.org/lapack

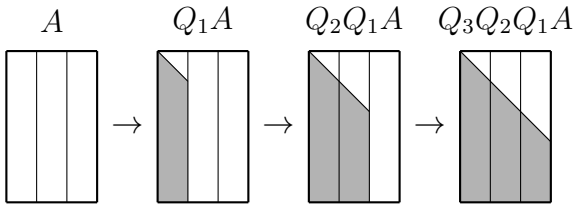



Figure 1: Example of the column block algorithm for a matrix with 3 column blocks.

The algorithm is visualized in Fig. 1.

In LAPACK, the routine for factorization using block Householder reflectors is named **GEQRT**. The routine that applies the calculated factor Q on an arbitrary matrix of appropriate size is labeled **GEMQRT**. These two routines will be referred to as **general QR kernels** in the rest of this article.

If a multithreaded implementation of the BLAS library is employed, parallelism is exploited in the second step of the algorithm. This approach, however, only offers enough parallelism if the matrix has a sufficient number of block columns, as only the update in Step 2 can be parallelized. Hence, for matrices $A \in \mathbb{R}^{m,n}$ with $m >> n$, this algorithm exploits parallelism insufficiently and offers subpar performance.

### 3.2. TS kernels & TS flat tree elimination scheme

In order to develop a parallel algorithm with good performance for tall-and-skinny matrices, it is necessary to split the matrix into row blocks as well as column blocks.

For a blocked matrix $A = \begin{pmatrix} A_1^T & A_2^T \end{pmatrix}^T$, where the block $A_1$ has a QR factorization $A_1 = Q_1 R_1$ and the matrix $\begin{pmatrix} R_1^T & A_2^T \end{pmatrix}^T$ has a QR decomposition $\widehat{Q}R$, it holds (see [7]) that

$$A = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix} = \begin{pmatrix} Q_1 R_1 \\ A_2 \end{pmatrix} = \begin{pmatrix} Q_1 & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} R_1 \\ A_2 \end{pmatrix} = \underbrace{\begin{pmatrix} Q_1 & 0 \\ 0 & I \end{pmatrix} \widehat{Q}}_{\bar{Q}} R = \bar{Q}R.$$

Since $\bar{Q}$ is a product of two orthogonal matrices, it is orthogonal as well. As a result, QR factorizations of the matrices $A$ and $\begin{pmatrix} R_1^T & A_2^T \end{pmatrix}^T$ have the same factor $R$.

Consequently, we can calculate the QR factorization of $A$ by factorizing its block $A_1$ using GEQRT (so that the matrix $A_1$ gets replaced with $R_1$) followed by factorizing the triangle-on-top-of-square matrix $\begin{pmatrix} R_1^T & A_2^T \end{pmatrix}^T$. The factorization and subsequent $Q$ application of triangle-on-top-of square matrices is performed using the so-called **TS kernels**:

**TSQRT** performs factorization of a triangle-on-top-of-square matrix
**TSMQR** applies the transformation from **TSQRT** on an arbitrary block matrix made up of two row/column blocks

A scheme of these functions is visualized in Fig. 2.

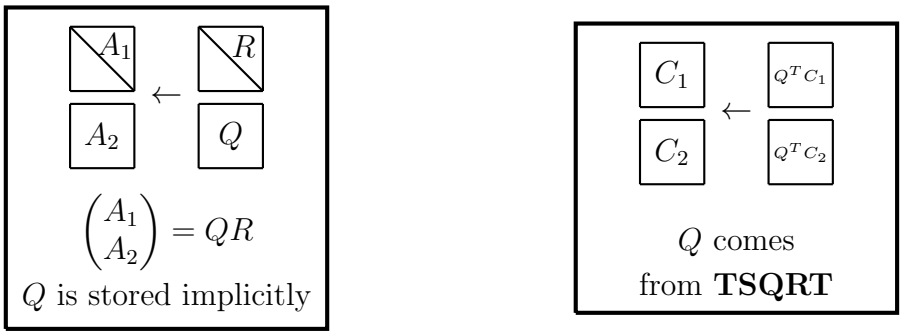



Figure 2: Scheme of the **TSQRT** kernel (left) and of the **TSMQR** kernel (right) for parameter values `SIDE='R'` and `TRANS='T'`.

Expanding on the observations made above, we can calculate the QR factorization of A using the **TS flat tree** elimination scheme:

1. Factorize the diagonal block using general QR kernels
2. Use it to eliminate the blocks below the main diagonal with TS kernels.

Similarly to the column block algorithm, only the $Q$ application kernels can be parallelized in this procedure. This algorithm is called sequential CAQR in [7].

### 3.3. TT kernels & TT binary tree elimination scheme

We now further examine the QR factorization of a blocked matrix $A = \begin{pmatrix} A_1^T & A_2^T \end{pmatrix}^T$. Let $A_1 = Q_1 R_1$ and $A_2 = Q_2 R_2$ be the QR factorizations of $A_1$ and $A_2$, in their respective order. Let then $\widehat{Q}R$ denote the QR factorization of the blocked matrix $\begin{pmatrix} R_1^T & R_2^T \end{pmatrix}^T$. Then (cf. [7]):

$$A = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix} = \begin{pmatrix} Q_1 R_1 \\ Q_2 R_2 \end{pmatrix} = \begin{pmatrix} Q_1 & 1 \\ 0 & Q_2 \end{pmatrix} \begin{pmatrix} R_1 \\ R_2 \end{pmatrix} = \begin{pmatrix} Q_1 & 0 \\ 0 & Q_2 \end{pmatrix} \widehat{Q} R.$$

Instead of using the GEQRT kernel on the upper block followed by the TSQRT kernel to factorize the blocked matrix $A = \begin{pmatrix} A_1^T & A_2^T \end{pmatrix}^T$, we could first factorize both blocks using the GEQRT kernel (so that the upper triangular parts of the blocks get replaced with $R_1$ and $R_2$) and then factorize the obtained triangle-on-top-of-triangle matrix. The last step is done using the **TTQRT factorization kernel**, whose scheme in visualized in Fig. 3.

Based on the procedure presented above, we can define the **TT binary tree scheme** for decomposing a general blocked matrix $A$:

1. Factorize all blocks on & below the main diagonal using `GEQRT`.
2. Eliminate blocks below the (block) diagonal using `TTQRT` in a binary tree fashion.

The application kernels TTMQR can be parallelized, but now the factorization kernels in both steps can be performed in parallel, too (provided that they occur on the same level of the binary tree). This approach is called parallel CAQR in [7].

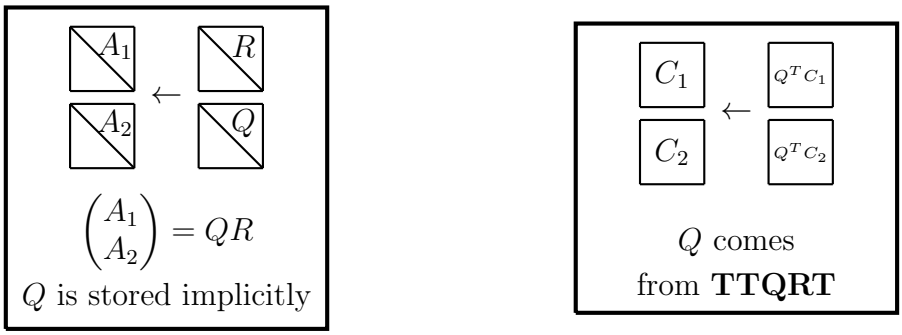



Figure 3: Scheme of the **TTQRT** kernel (left) and of the **TTMQR** kernel (right) for parameter values `SIDE='R'` and `TRANS='T'`.

### 3.4. Superblock-based elimination schemes

By comparing the TS flat tree and the TT binary tree elimination schemes, we can see that:

- The TS flat tree scheme requires fewer kernel calls with only the application kernels being parallelizable.
- The TT binary tree scheme requires more kernel calls with both the factorization and $Q$ application kernels being parallelizable.

In this respect, the two schemes can be seen as block versions of the Householder reflector and the Givens rotation methods, respectively.

To balance out the effects of the two schemes, we may divide each block column into **superblocks**, where all superblocks in each column contain the same number of blocks (with the possible exception of the last superblock in a column). In other words, each superblock is composed of a fixed number of subsequent blocks. This number of blocks is called the **superblock size** $b$. We then factorize each column block of the matrix in the following manner:

1. Eliminate all blocks in an individual superblock using the TS flat tree scheme.
2. Eliminate first blocks of all superblocks in this column block using the TT binary tree scheme.

To select the superblock size $b$, we may utilize the formula [6]

$$b = \frac{m_t(n_t^2/2 + n_t/2)}{\gamma p}, \tag{1}$$

where

- $m_t$ is the number of block rows,
- $n_t$ is the number of block columns,
- $p$ is the number of available threads,
- $\gamma$ is a scaling factor (the default selection is $\gamma = 4$ as in the PLASMA library).

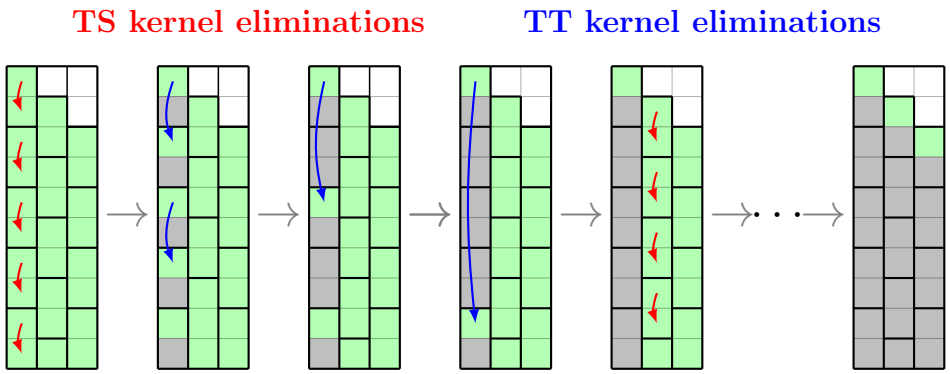



Figure 4: Example of the binary tree elimination scheme for 10 row blocks and 3 column blocks.

Formula (1) takes into account both the shape of the matrix and the number of CPU cores available for parallelization. As such, we obtain elimination schemes similar to the TT binary tree scheme for tall-and-skinny matrices, while for square-like matrices, we obtain elimination schemes very similar to the TS flat tree scheme.

This parameterized scheme is called the **superblock binary tree** scheme, see Fig. 4 for an example on a matrix with 10×3 blocks. By selecting a different scheme for eliminating the first blocks of each superblock in step 2, we may create different superblock-based schemes (other examples include the **superblock greedy** and **superblock Fibonacci** schemes [10], which are later user in Fig. 9). This idea can be seen as composing a hierarchical elimination tree [8] with different elimination trees on the top level and flat trees on the leaves (bottom level within superblocks).

## 4. Task-based runtime systems

The data dependencies between individual kernels can be represented by a **directed acyclic graph** (DAG), see an example in Fig. 5. From the DAG, we can see that the amount of available parallelism varies throughout the computation. As a result, **dynamic scheduling** is a powerful approach to implement a parallel TS flat tree scheme as well as the other schemes presented.

To ease the implementation, we use a **task-based runtime system**. These are systems that let us split the code into sections called **tasks**, and then execute the tasks in parallel while making sure that the data dependencies of the tasks are satisfied.

The runtime systems used in the implementation are **OpenMP** and **StarPU**. OpenMP[4] is a widely used standard for shared memory multicore programming, while StarPU [2] is a library created mainly with the intention of being used in heterogeneous systems (systems with multiple types of computing units), mainly targeting GPU accelerators.

[4] https://www.openmp.org/

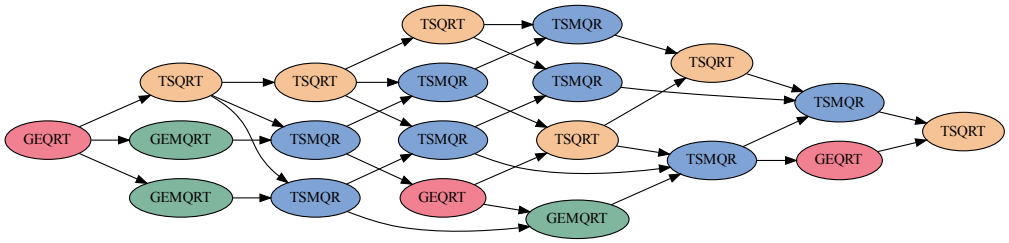



Figure 5: An example of the directed acyclic graph (DAG) of data dependencies. The TS flat tree scheme was used on a matrix with 4 row blocks and 3 column blocks.

## 5. Results

In this section, we evaluate the effect of different parameters on performance of the algorithm. We also compare the performance with the Arm Performance Libraries (ArmPL).

In each of the following figures, every point represents the best performance out of five consecutive runs. All experiments were performed using $m \times n$ matrices of sizes $m = 90000$ and $n \in \{250, 500, \ldots, 2750, 3000, 4000, \ldots, 15000\}$.

The performance was evaluated at the IT4Innovations National Supercomputing Center on a node with the following specifications:

- CPU: 1× NVIDIA Grace CPU Superchip
- CPU architecture: Arm64
- CPU cores: 144
- Base CPU frequency: 3.1 GHz
- All-code SIMD frequency: 3.0 GHz
- Instruction set extensions: Scalable Vector Extension 2 (SVE2)

The following library versions were used during the evaluation process:

- GCC 11.3.0
- Hwloc 2.7.1
- StarPU 1.3.10
- Arm Performance Libraries 22.1
- PLASMA 22.9.29

### 5.1. Block size comparison

In this section, we compare the effect of varying block sizes ($nb$) on the computational performance. In PLASMA, the matrix is first copied from column-major format to the tile layout before the QR factorization starts. After the end, the matrix

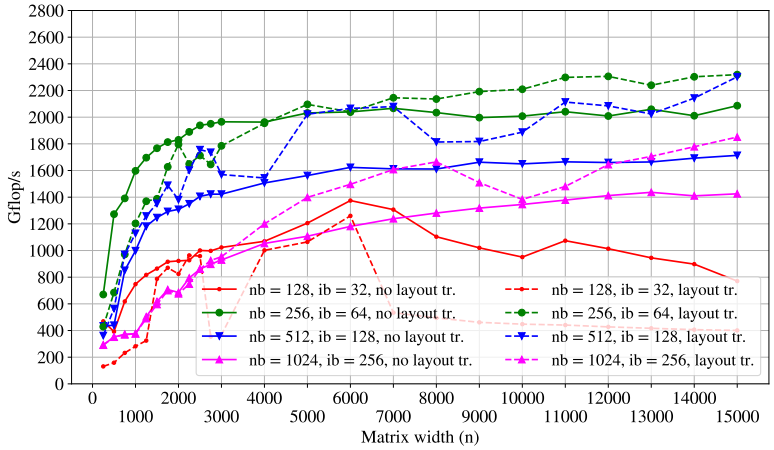



Figure 6: Performance for different block sizes ($nb$). SuperblockGreedy scheme, inner block size ($ib$) equal to $nb/4$, and $\gamma = 4$.

is copied back to the column-major format. We refer to these pre-/post-processing steps as *layout translation*. While the cost of the layout translation becomes outweighed by the faster processing of the matrix in the tile layout for wider matrices, we show in [6] that for very skinny matrices, it can be considerably faster to avoid the layout translation. Hence, for each tested block size in this section, we consider two variants – with and without layout translation.

We can see from Fig. 6 that the block size of 256 offers the best performance on the NVIDIA Grace node. Layout translation can boost the performance for wider matrices, while it can hinder the performance for skinnier matrices.

### 5.2. Inner block size comparison

In this section, we take a look at the effects of different inner block ($ib$) size values. The square blocks of size $nb$ are divided into smaller block columns to perform the block-local operations by column-block oriented functions. Hence, the inner block size $ib$ (the number of columns within these inner blocks) is always less than or equal to the selected block size; details can be found again in [6].

As can be seen from Fig. 7, the inner block size choice of 64 is best for the examined compute node.

### 5.3. Elimination schemes comparison

In this section, we compare the performance of the different elimination schemes presented earlier. We also include a performance curve for the Arm Performance Libraries (ArmPL) for comparison. The results are shown in Fig. 8.

In accordance with the observations in Section 3, the TsFlatTree scheme delivers a better performance for wider matrices, but it gets outperformed by the TtGreedy scheme for skinnier matrices. The SuperblockGreedy scheme combines the advan-

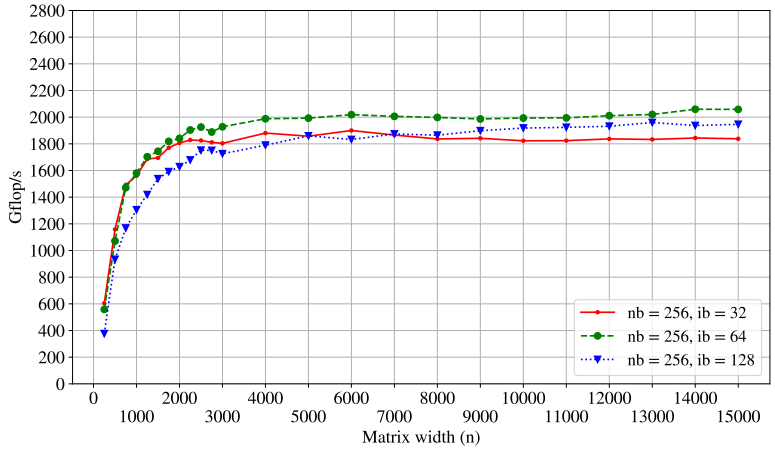



Figure 7: Performance for different inner block sizes. Tile size $nb = 256$, Superblock-Greedy scheme, $\gamma = 4$, and layout translation disabled.

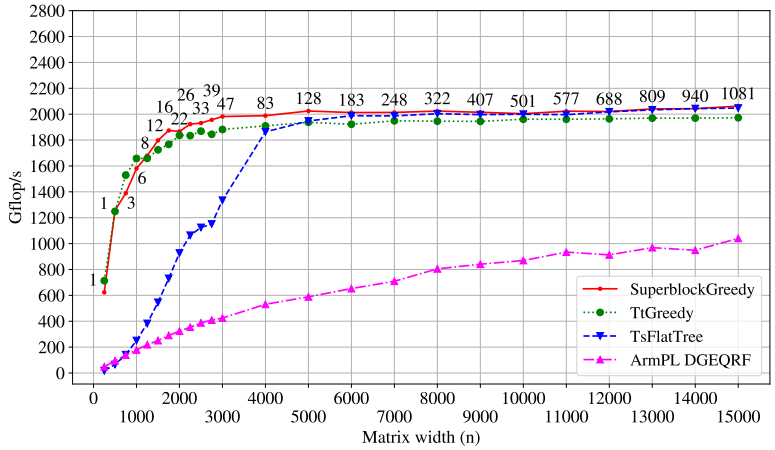



Figure 8: Performance of different elimination schemes. Block size $nb = 256$, inner blocks of size $ib = 64$, layout translation disabled, and $\gamma = 4$ (for the Superblock-Greedy elimination scheme). The numbers in the plot denote the used superblock sizes.

tages of both schemes to provide a good performance for both skinny and wide matrices. Interestingly, the performance of the TtGreedy scheme is only marginally lower on this architecture. All the schemes outperform the ArmPL implementation for matrices with more than 500 columns, while the latter slightly outperforms the TsGreedy scheme for skinnier matrices.

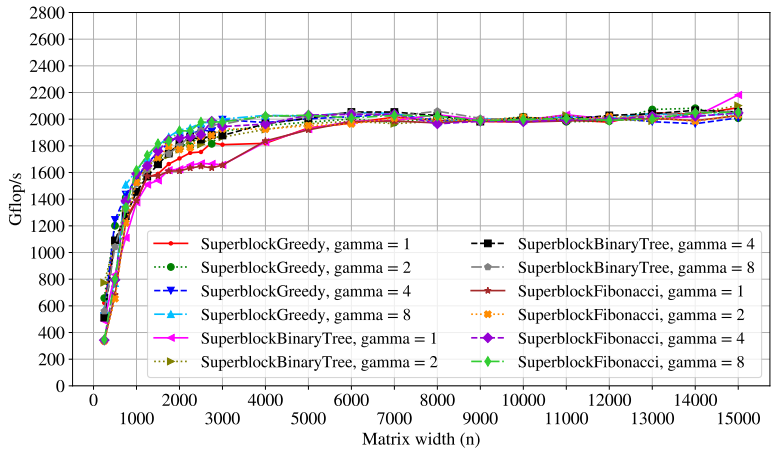



Figure 9: Performance of various superblock size factors ($\gamma$ values). Block size $nb = 256$, inner block size $ib = 64$, and layout translation disabled.

### 5.4. Superblock size factor comparison

Next, we visualize the effects of different choices of the values for the $\gamma$ parameter presented in Section 3.4. We compare four different values of $\gamma \in \{1, 2, 4, 8\}$ for three different superblock-based elimination schemes (SuperblockGreedy, SuperblockBinaryTree and SuperblockFibonacci).

In Fig. 9, we can see similar performances exhibited for all elimination trees and all $\gamma$ values except for $\gamma = 1$. In the case of $\gamma = 1$, the SuperblockGreedy scheme performs better than the other two schemes for matrices with 1750-3000 columns, despite still not reaching the performance of the other tested $\gamma$ values.

### 5.5. Runtime systems comparison

Finally, we examine the differences between the two presented runtime systems. Figure 10 shows that the performance of both runtime systems is very similar for wider matrices, while the OpenMP runtime system performs slightly better for skinnier matrices for both tested parameter sets.

## 6. Conclusions

The results of experiments with the NVIDIA Grace CPU Superchip bring us mostly to similar conclusions as the results from nodes tested in [6]. Nevertheless, the Grace node results have a few distinct features:

- The performance drop of the TtGreedy scheme for wider matrices is much less significant.
- There is a difference in performances of the SuperblockGreedy scheme and the other two superblock-based schemes for $\gamma = 1$. More specifically, the

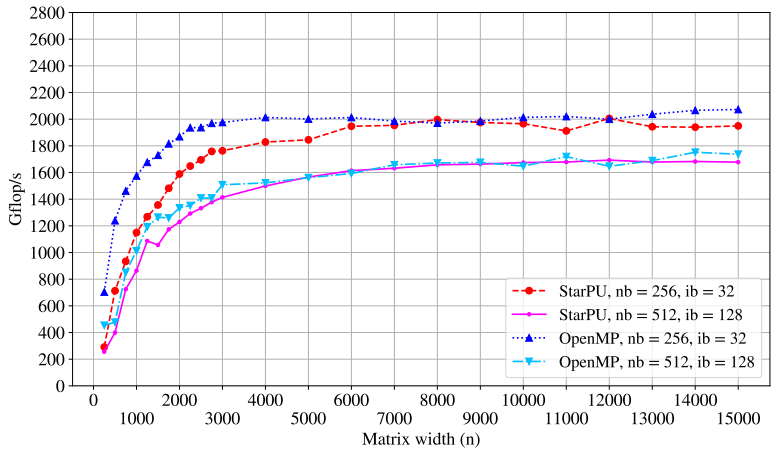



Figure 10: Performance of the OpenMP and StarPU runtime systems. The SuperblockGreedy scheme with inner blocks of size 32 and 128 (for block sizes of 256 and 512, respectively), layout translation disabled, and $\gamma = 4$.

SuperblockGreedy scheme performs better for certain matrix sizes. Hence, $\gamma \geq 2$ can be recommended also for this architecture.

- The difference between the OpenMP and StarPU runtime systems is small for skinny matrices and even smaller for wider matrices. Nevertheless, OpenMP still seems as a good choice for implementing this algorithm.

Details of the algorithm and results for different architectures will appear in [6].

## Acknowledgments

This work was supported by the Student Summer Research Program 2023 of FIT CTU in Prague, by the Czech Science Foundation under project GA ČR 23-06159S, and by the Institute of Mathematics of the Czech Academy of Sciences (RVO:67985840). The computational time on the systems at IT4Innovations was provided thanks to the support by the Ministry of Education, Youth and Sports of the Czech Republic through the e-INFRA CZ (ID:90254).

## References

[1] Anderson, E. et al.: *LAPACK User's Guide*. Society for Industrial and Applied Mathematics, Philadelphia, 1999, Third edn.

[2] Augonnet, C., Thibault, S., Namyst, R., and Wacrenier, P.A.: StarPU: a unified platform for task scheduling on heterogeneous multicore architectures. Concurrency and Computation: Practice and Experience **23** (2011), 187–198.

[3] Blackford, S.L. et al.: *ScaLAPACK User's Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1997.

[4] Bouwmeester, H., Jacquelin, M., Langou, J., and Robert, Y.: Tiled QR factorization algorithms. In: *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*. SC '11, Association for Computing Machinery, 2011 .

[5] Buttari, A., Langou, J., Kurzak, J., and Dongarra, J.: A class of parallel tiled linear algebra algorithms for multicore architectures. Parallel Computing **35** (2009), 38–53.

[6] Břichňáč, V., Šístek, J., and Langou, J.: Effect of different elimination schemes on task-based implementation of qr factorization for multicore architectures. In preparation, 2025.

[7] Demmel, J., Grigori, L., Hoemmen, M., and Langou, J.: Communication-optimal parallel and sequential QR and LU factorizations. SIAM Journal on Scientific Computing **34** (2012), A206–A239.

[8] Dongarra, J. et al.: Hierarchical QR factorization algorithms for multi-core clusters. Parallel Computing **39** (2013), 212–232.

[9] Dongarra, J. et al.: PLASMA: Parallel linear algebra software for multicore using OpenMP. ACM Trans. Math. Softw. **45** (2019), 16:1–16:35.

[10] Faverge, M., Langou, J., Robert, Y., and Dongarra, J.: Bidiagonalization with parallel tiled algorithms. Tech. Rep. R-8969, INRIA, 2016.

[11] Fukaya, T., Kannan, R., Nakatsukasa, Y., Yamamoto, Y., and Yanagisawa, Y.: Shifted Cholesky QR for computing the QR factorization of ill-conditioned matrices. SIAM Journal on Scientific Computing **42** (2020), A477–A503.

[12] Fukaya, T., Nakatsukasa, Y., Yanagisawa, Y., and Yamamoto, Y.: CholeskyQR2: A simple and communication-avoiding algorithm for computing a tall-skinny QR factorization on a large-scale parallel system. In: *2014 5th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems*. 2014 pp. 31–38.

[13] Higgins, A.J., Szyld, D.B., Boman, E.G., and Yamazaki, I.: Analysis of randomized Householder-Cholesky QR factorization with multisketching. arXiv:2309.05868v2, 2024.

*Programs and Algorithms of Numerical Mathematics 22*
J. Chleboun, J. Papež, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics CAS, Prague 2025

# NON-STOCHASTIC UNCERTAINTY QUANTIFICATION OF A MULTI-MODEL RESPONSE

Jan Chleboun

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

**Abstract:** The focus is put on the application of fuzzy sets and Dempster-Shafer theory in assessing the nature and extent of uncertainty in the response of $M$ models that model the same phenomenon and depend on fuzzy input data. Dempster-Shafer theory uses a weighted family of fixed sets called the focal elements to evaluate the relationship between an arbitrarily chosen set and the focal elements. It is proposed to create at least $M$ weighted focal elements on the basis of 1) the responses to fuzzy inputs to the models, and 2) the weights associated with the models. Four variants of this approach are illustrated by academic examples.

**Keywords:** fuzzy sets, evidence theory, uncertainty

**MSC:** 03E75, 90C90

## 1. Introduction

In this contribution, the following situation is addressed: Let one phenomenon be modeled by several models whose input parameters are uncertain. How can the combined responses of the individual models be assessed and their trustworthiness evaluated? In other words, what sort of uncertainty quantification can be applied to the synergy of responses that originates from various models?

An uncertainty analysis applied to one model with uncertain inputs is quite common. Although the above multi-model situation is not frequent, it is not exceptional. Take, for instance, 1D models of elastic beams. One can choose the Euler-Bernoulli beam model, the Timoshenko(-Ehrenfest) model, or the less known nonlinear Gao beam model [6, 8], see also [9]. The 1D models can always be confronted with 3D models or, under special circumstances, with 2D models.

A large variety of models with uncertain input data offers the modeling of a long-term behavior of concrete. They include a number of internationally recognized models, national codes, and models proposed in academia, see [3].

DOI: 10.21136/panm.2024.04

## 2. Elements of fuzzy set theory and evidence theory

Let us recall the three key concepts of fuzzy sets and their applications, namely the membership function $\mu_A$ of a fuzzy interval $A$, the $\alpha$-cut $A^\alpha$ of a fuzzy set $A$, and Zadeh's extension principle.

### 2.1. Fuzzy sets, membership functions, $\alpha$-cuts

Let the membership function $\mu_A$ be a continuous and concave function that maps a closed interval $A = [a, b]$ onto the interval $[0, 1]$. For computational purposes, let us limit ourselves to trapezoidal membership functions, i.e., piecewise linear functions identifiable with ordered 4-tuples $(a, c_1, c_2, b) \in \mathbb{R}^4$, where $\mu_A(a) = 0 = \mu_A(b)$, $\mu_A(c_1) = 1 = \mu_A(c_2)$, and $\mathbb{R}$ stands for the set of real numbers. A special, i.e., triangular case is obtained if $c_1 = c_2$.

The subsets of $A$ defined through

$$A^\alpha = \{x \in A |\ \mu_A(x) \geq \alpha\}, \tag{1}$$

where $\alpha \in [0, 1]$, are called the $\alpha$-cuts of $A$.

*Remark:* The abovementioned concept of membership functions is simple and restrictive, but it is tailored to our future computational needs. Another advantage lies in the fact that the existence of extremes is guaranteed, see (5) and (6), and that we can replace suprema and infima by maxima and minima in the theory of fuzzy sets. Nevertheless, a more general concept of fuzzy sets is common, see [5, 13], for example.

Fuzzy intervals can easily be generalized to fuzzy $n$-dimensional rectangular parallelepiped $A = A_1 \times A_2 \times \cdots \times A_n \subset \mathbb{R}^n$ where each interval $A_i$ is associated with a membership function $\mu_{A_i}$ and the fuzzy variables are mutually independent. Then for each $x = (x_1, x_2, \ldots, x_n) \in A$, we can define

$$\mu_A(x) = \min\{\mu_{A_1}(x_1), \mu_{A_2}(x_2), \ldots, \mu_{A_n}(x_n)\}. \tag{2}$$

We also observe that

$$\forall \alpha \in [0, 1] \quad A^\alpha = A_1^\alpha \times A_2^\alpha \times \cdots \times A_n^\alpha. \tag{3}$$

Let $g$ be a continuous function defined on a fuzzy set $A$ (either $A \subset \mathbb{R}$ or a parallelepiped $A \subset \mathbb{R}^n$) and mapping $A$ to a range $R_{g,A}$. Zadeh's extension principle defines the way how to transfer the membership degree from $x \in A$ to $g(x) \in R_{g,A}$. In detail [13],

$$\forall y \in R_{g,A} \quad \mu_{R_{g,A}}(y) = \max_{\{x \in A |\ g(x) = y\}} \mu_A(x). \tag{4}$$

The original definition (4) is not computation-friendly. This is why we will use an equivalent approach based on the fact that if the $\alpha$-cuts $R_{g,A}^\alpha$ are known for all $\alpha \in [0, 1]$ and if $y \in R_{g,A}$, then

$$\mu_{R_{g,A}}(y) = \max\{\alpha \in [0, 1] |\ y \in R_{g,A}^\alpha\}. \tag{5}$$

It is not difficult to infer, see [10] or elsewhere, that

$$\forall \alpha \in [0,1] \quad R^{\alpha}_{g,A} = \left[ \min_{x \in A^{\alpha}} g(x), \max_{x \in A^{\alpha}} g(x) \right]. \tag{6}$$

In other words, to obtain $R^{\alpha}_{g,A}$, we have to solve worst-case and best-case scenario problems (6).

### 2.2. Evidence theory, focal elements, Belief, Plausibility

The origin of the Dempster-Shafer theory of evidence [4, 11, 1, 12] can be traced back to considerations about lower and upper bounds of probabilities. In our approach, we interpret the weights forming the basic probability assignment as the amounts of trustworthiness assigned to fixed significant sets called focal elements, see the next paragraphs.

To this end, we assume that a set $\mathcal{S}$ of chosen intervals $I_1, I_2, \ldots, I_s$ is given together with the weight map $w\colon I \mapsto (0,1]$ where $I \in \mathcal{S}$ and $\sum_{i=1}^{s} w(I_i) = 1$. In the evidence theory, the intervals and the map are called the focal elements and the basic probability assignment, respectively.

Two values can be associated with an arbitrary subset $B \subset \mathbb{R}$, namely *Belief* and *Plausibility*

$$Bel(B) = \sum_{\{I \in \mathcal{S}\colon\, I \subseteq B\}} w(I) \quad \text{and} \quad Pla(B) = \sum_{\{I \in \mathcal{S}\colon\, I \cap B \neq \emptyset\}} w(I). \tag{7}$$

We observe that $Bel(B)$ collects the weights of those focal elements that are fully covered by $B$. That is, if these focal elements are outputs of some weighted models, then $B$ fully represents all of these outputs. In contrast, $Pla(B)$ is less strict as it allows for both full (subset) and partial (nonempty intersection) representation.

## 3. Uncertainty quantification in multi-modeling

The background idea is not new. It associates $\alpha$-cuts of a fuzzy set with focal elements [2]. A rather straightforward modification leads to an application to responses of several models. The method will be explained and illustrated on a particular example.

Let us consider $M = 3$ models represented by the following respective functions

$$m_1(p) = 7.3 + 0.02 p_3 (p_1 p_2)^{(p_3+p_4)}, \qquad m_2(p) = 7.3 + 0.02 p_2 (p_1 + p_3),$$
$$m_3(p) = 6 + 0.4 \frac{p_2 p_3 p_4}{p_1},$$

where $N_p = 4$ parameters form the vector $p = (p_1, p_2, p_3, p_4)$. If $\widehat{p} = (1.2, 2.1, 1.5, 1.2)$, then the response of all three models is roughly equal to 7.5 as is also indicated in Figure 1, the details of which will be given later.

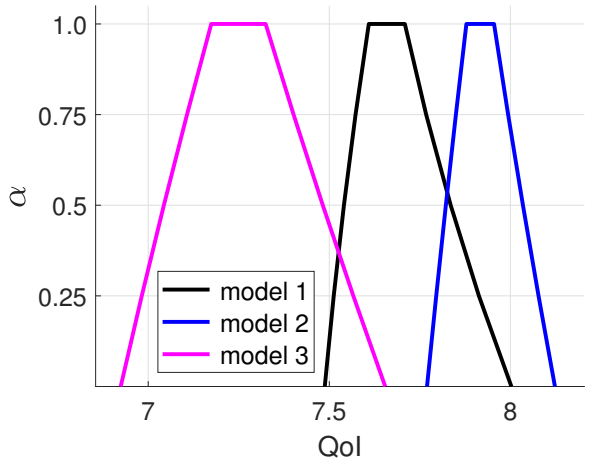



Figure 1: Membership functions of fuzzy responses.

Let the input parameters $p_i$, $i = 1, \ldots, N_p$, belong to intervals $A_i \equiv A_i^0$ provided with membership functions $\mu_i$. The product of the intervals forms the set $A = A_1 \times \cdots \times A_{N_p}$.

Next, let each model be associated with a positive weight $w_i \in \mathbb{R}$ such that $\sum_{j=1}^{M} w_j = 1$.

In the following numerical examples, we use these membership functions

$$\mu_1 = \widehat{p_1}(0.95, 0.99, 1.01, 1.05), \qquad \mu_2 = \widehat{p_2}(0.9, 0.98, 1.02, 1.1),$$
$$\mu_3 = \widehat{p_3}(0.92, 0.97, 1.01, 1.08), \qquad \mu_4 = \widehat{p_4}(0.93, 0.99, 1.01, 1.05);$$

and the basic probability assignment defined as $w_1 = 0.25$, $w_2 = 0.4$, and $w_3 = 0.35$.

The partial derivatives of $m_i$ allow us to conclude that the functions $m_i$ are monotone in each $p_j$ on the supports $A_i$ of the membership functions. As a consequence, the extremes of $m_i$ are attained at the ends of the interval $A_i^\alpha$, thus solving (6) for various values of $\alpha$ is easy. Based on (6) with $\alpha = \alpha_\ell = \ell/N_\alpha$, $\ell = 0, 1, \ldots, N_\alpha$, $N_\alpha = 4$, the approximate piecewise linear membership functions of the ranges of the models' responses are depicted in Figure 1. The value of the quantity of interest (QoI) is simply the scalar response of the models to the input data $p \in A$.

We are ready to introduce **Algorithm 1**:

**Step 1:** *Fix $\alpha \in [0, 1]$ and infer the $\alpha$-cut $A^\alpha$ by using (3) and the $\alpha$-cuts $A_i^\alpha$, $i = 1, \ldots, N_p$.*

**Step 2:** *By setting $g = m_j$ and using (6), calculate $R^\alpha_{m_j,A}$, $j = 1, \ldots, M$.*

**Step 3:** *Interpret the intervals $R^\alpha_{m_j,A}$, $j = 1, \ldots, M$ as focal elements with the respective weights $w_j$.*

**Step 4:** *Choose an interval $B \subset \mathbb{R}$ and calculate $Bel(B)$ and $Pla(B)$ by using (7) where $\mathcal{S} = \{R^\alpha_{m_j,A}\}_{j=1}^M$.*

**Step 5:** *Repeat Step 4 several times with the aim to increase Bel and Pla and to identify a set $B$ that satisfactorily represents the joined responses of the models on the level $\alpha$.*

The algorithm needs some comments. First, the goal of Step 2 can be quite challenging if the models (unlike our case) are not trivial. It may happen, for instance, that $p_i$ are parameters of a problem driven by differential equations whose solution is then post-processed to obtain a value of $m_i(p)$, a quantity of interest. As a consequence, the minimization and maximization in (6) can be a difficult task.

Second, obtaining the weights $w_i$ is a delicate matter. Although measurement-based approaches can be available, see [7] aiming at stochastic uncertainty, expert opinion can often be a substantial, if not sole, source of information.

Third, the goal of Step 4 and Step 5 is to find an interval $B$ that best characterizes the ensemble of output intervals $R^{\alpha}_{m_j,A}$. It commonly happens that there is no such "best" interval available. By taking a sufficiently large and appropriately positioned interval $B$, we can obtain $Bel(B) = 1 = Pla(B)$. The interval, however, might be so large that its practical value as a representative of key models' responses is questionable. Although it shows the total extent of uncertain responses, it does not indicate the subsets where the responses overlap, that is, the responses of at least some models are not too distinct from each other. To identify such intervals, shorter intervals $B$ must also be tested by the focal elements. Again, the results can prevent an unequivocal conclusion. Take, for instance, $Bel(B_1) < Bel(B_2)$ and $Pla(B_1) > Pla(B_2)$ for some two intervals $B_1$ and $B_2$ of the same length.

If the number of the output intervals $R^{\alpha}_{m_j,A}$ (i.e., output focal elements) is small, then the analysis of their intersections and unions can lead to the sets maximizing *Belief* and *Plausibility*. Such analysis is more and more challenging if the number of output focal elements increases. *Bel* and *Pla* values calculated for a family of intervals is then an option that offers both a general view and sufficiently accurate information on the synergy of joint responses. This approach will be in the focus of the next paragraphs.

We define intervals $B^d_{s,k} = (a + ks, a + ks + d)$ of the length $d > 0$. The position of $B^d_{s,k}$ is controlled by the fixed parameters $a \in \mathbb{R}$ and $s \in \mathbb{R}$ as well as by the parameter $k = 0, 1, \ldots, K$. The intervals $B^d_{s,k}$ play the role of $B$ in Algorithm 1. Some results are depicted in Figure 2 where the points $[a + ks, Y]$ represent the values $Y = Bel(B^d_{s,k})$ and $Y = Pla(B^d_{s,k})$. The parameters $\alpha$ and $a$ are fixed to 0.5 and 6.8, respectively.

In the left graph, we observe that $k = 15$ and $k = 19, 20, 21$ indicate the intervals that are worth attention. Although $Pla([7.45, 7.825]) = 1$, $Bel([7.45, 7.825]) = 0$ might suggests that the intervals with nonzero *Belief* could be a better representation of the combined responses since their *Bel* and *Pla* values are more balanced. Similar ambiguity shows the right graph. The analyst can choose either the maximum of *Pla* with a rather low *Bel* value or the maximum of *Bel* accompanied by a decreased *Pla* value. The interval $B^d_{s,15} = [7.5, 8.05]$ shows a balanced assessment in both respects. Naturally, the use of longer intervals ($d = 0.55$) increases the *Bel* value and increases the number of positions where *Pla* is equal to one.

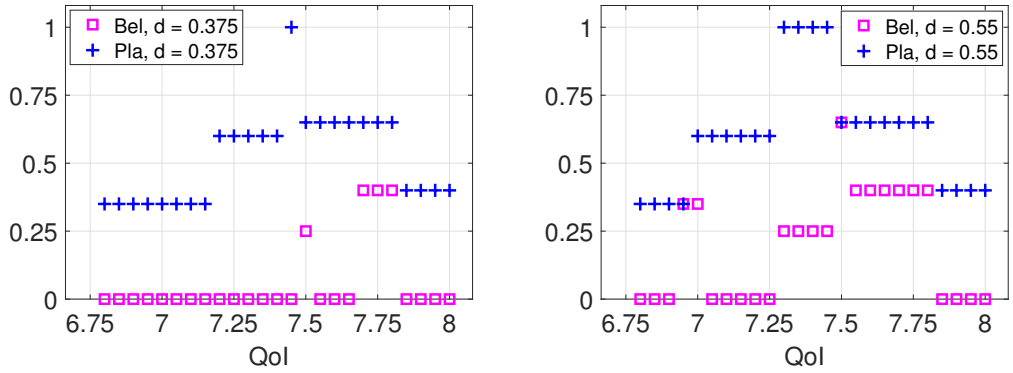



Figure 2: Algorithm 1. $Bel(B^d_{s,k})$ and $Pla(B^d_{s,k})$ for $s = 0.05$, $d = 0.375$ (left) and $s = 0.05$, $d = 0.55$ (right).

### 3.1. Modifications of Algorithm 1

The standard definition (7) shows a shortcoming that becomes more visible especially in our application where we wish to assess the extent of joint responses of the models. In (7), there is no difference between a very short intersection $I \cap B$ and a full set intersection; both cases are evaluated by the full weight $w(I)$.

To take into account the relative extent of intersection, let us redefine $Pla$ in (7) as $Pla^{\text{new}}$

$$Bel(B) = \sum_{\{I \in \mathcal{S}:\ I \subseteq B\}} w(I) \quad \text{and} \quad Pla^{\text{new}}(B) = \sum_{\{I \in \mathcal{S}:\ I \cap B \neq \emptyset\}} w(I) \frac{\text{meas}_1(I \cap B)}{\text{meas}_1 I} \tag{8}$$

where $\text{meas}_1$ stands for the one-dimensional Lebesgue measure, which turns into the length of intervals in our calculations.

**Algorithm 2** coincides with Algorithm 1 except for
**Step 4:** *Choose an interval $B \subset \mathbb{R}$ and calculate $Bel(B)$ and $Pla^{\text{new}}(B)$ by using (8) where $\mathcal{S} = \{R^{\alpha}_{m_j,A}\}_{j=1}^{M}$.*

We observe in Figure 3 that if $d = 0.55$, then the interval $B^s_{d,15} = [7.5, 8.05]$ is the best representation of the joint model response on the uncertainty level $\alpha = 0.5$. For $d = 0.375$, the analyst would see the interval $[7.7, 8.075]$ as the best representative though its $Pla^{\text{new}}$ does not reach the maximum. However, any increase in $Pla^{\text{new}}$ is paid for by the zero $Bel$ value.

Both algorithms focus on uncertainty quantification in model responses restricted to a fixed input uncertainty level, that is, $\alpha = 0.5$ in our examples. By taking into account all the $\alpha$-cuts of the fuzzy inputs and by modifying the standard transformation [2] of one membership function to a set of focal elements, we arrive at an extended set of focal elements with an associated basic probability assignment.

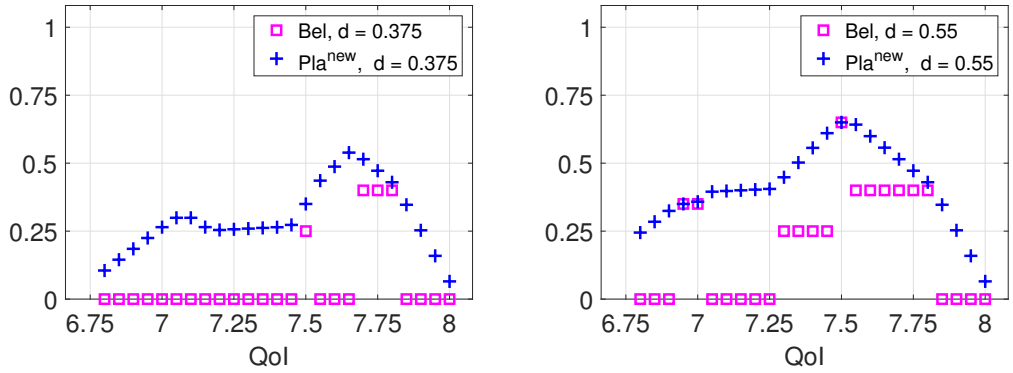



Figure 3: Algorithm 2. $Bel(B^d_{s,k})$ and $Pla^{\text{new}}(B^d_{s,k})$ for $s = 0.05$, $d = 0.375$ (left) and $s = 0.05$, $d = 0.55$ (right).

**Algorithm 3**:
**Step 1:** *For $\alpha_\ell$, $\ell = 0, 1, \ldots, N_\alpha$, infer the $\alpha_\ell$-cut $A^{\alpha_\ell}$ by using (3) and the $\alpha_\ell$-cuts $A_i^{\alpha_\ell}$, $\ell = 0, 1, \ldots, N_\alpha$.*
**Step 2:** *By setting $g = m_j$ and using (6), calculate $R^{\alpha_\ell}_{m_j,A}$ for $j = 1, \ldots, M$ and $\ell = 0, 1, \ldots, N_\alpha$.*
**Step 3:** *Interpret the intervals $R^{\alpha_\ell}_{m_j,A}$ as focal elements with the respective weights $w_j/N_\alpha$.*
**Step 4:** *Choose an interval $B \subset \mathbb{R}$ and calculate $Bel(B)$ and $Pla(B)$ by using (7) where $\mathcal{S} = \{R^{\alpha_\ell}_{m_j,A}\}_{j=1,\ldots,M;\,\ell=0,\ldots,N_\alpha}$.*
**Step 5:** *Repeat Step 4 several times with the aim to increase Bel and Pla and to identify an interval $B$ that satisfactorily represents the joined responses of the models.*

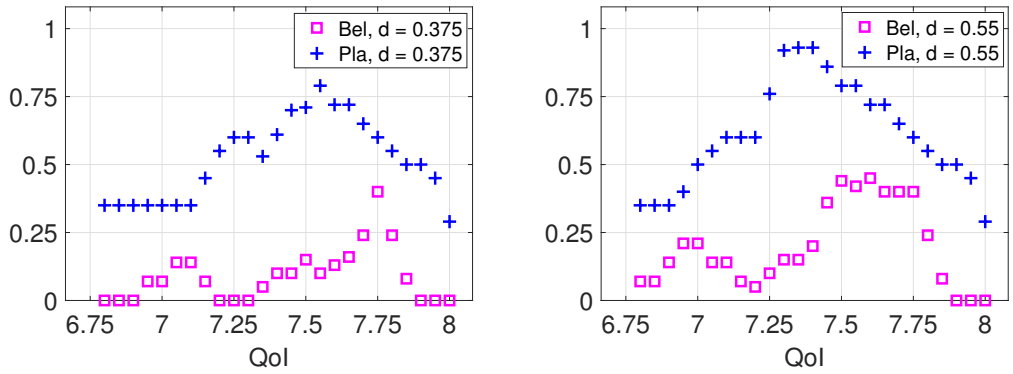



Figure 4: Algorithm 3. $Bel(B^d_{s,k})$ and $Pla(B^d_{s,k})$ for $s = 0.05$, $d = 0.375$ (left) and $s = 0.05$, $d = 0.55$ (right).

The output of Algorithm 3 is depicted in Figure 4. Although more information on fuzzy inputs was taken into account, i.e., more focal elements entered the calculations, the graphs do not offer a definite identification of the intervals that best characterize the join models' outputs. Owing to a rather strong gain in *Bel* and a not bad *Pla* level, one would probably prefer $[7.75, 8.125]$ over the other intervals in the $d = 0.375$ family. If $d = 0.55$, then $[7.45, 8]$ and $[7.5, 8.05]$ seem to be equal candidates because the loss in *Bel* is compensated by the gain in *Pla* and vice versa.

Finally, we can modify Algorithm 3 to get **Algorithm 4**. To this end, we refer to (8) instead to (7) in Step 4. The output of Algorithm 4 is depicted in Figure 5.

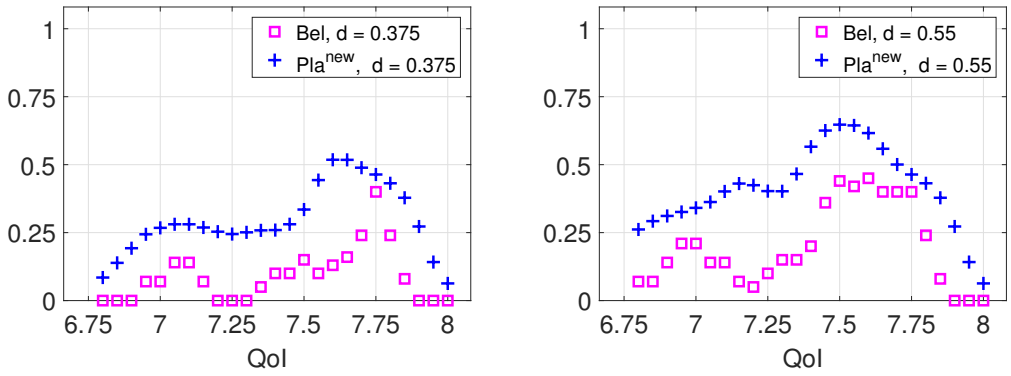



Figure 5: Algorithm 4. $Bel(B^d_{s,k})$ and $Pla^{\text{new}}(B^d_{s,k})$ for $s = 0.05$, $d = 0.375$ (left) and $s = 0.05$, $d = 0.55$ (right).

Now, clearer conclusions can be made than in the case of Figure 4. The intervals $[7.75, 8.125]$ and $[7.5, 8.05]$ seem to guarantee the strongest combination of the *Bel* and *Pla* assessments within the two sequences of intervals.

## 4. Comments and conclusions

The advantage of Algorithm 1 and Algorithm 3 is not only computational (they use the lowest number of focal elements) but also analytical because the uncertainty analysis is limited to a particular $\alpha$-cut of input data. Although Algorithm 2 and Algorithm 4 make use of a richer family of focal elements, the picture of a multi-model synergy might not be clearer. Take, for instance, a high value of $Bel(B)$ for some interval $B$. Then, the questions arise: What is the cause? Does $B$ cover a significant number of focal elements originating in several models, or does $B$ cover a high number of focal elements belonging to only one model? Remember, that the focal elements associated with one model $m_j$, i.e., $j$ fixed, form a chain of intervals for which $R^{\alpha_1}_{m_j,A} \subset R^{\alpha_2}_{m_j,A}$ if $\alpha_2 < \alpha_1$.

The probabilistic background of the evidence theory has been neglected in our exposition. Nevertheless, $Pla^{\text{new}}$ in (8) could be interpreted as the probability that the crisp model response uniformly distributed in the interval $I$ also falls into the interval $B$.

The reader might propose a modification of Algorithm 2 and Algorithm 4: to infer the focal elements of the models' responses, reduce the range of alphas and use, for instance, $\alpha = 0.5$, $\alpha = 0.75$, and $\alpha = 1$. This would certainly be possible, but we can get the same effect by reshaping the membership functions and considering $\alpha = 0$, $\alpha = 0.5$, and $\alpha = 1$. In this way, we obtain the standard Algorithm 2 and Algorithm 4.

What final conclusions can be made? To identify the intervals that most agree with multi-model responses, it is advisable to apply Algorithm 2 for various but individual values of $\alpha$, and then Algorithm 4. Sufficiently rich and fine sequences of intervals determined by various values of $s$ and $d$ should be used in the analysis.

## Acknowledgments

This work is part of 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 author is grateful to Dr. Richard (Dick) Haas for fruitful discussions.

## References

[1] Ayyub, B. M. and Klir, G. J.: *Uncertainty Modeling and Analysis in Engineering and Sciences*. Boca Raton, FL: Chapman & Hall/CRC, 2006.

[2] Bernardini, A. and Tonon, F.: *Bounding Uncertainty in Civil Engineering*. Springer, Berlin, 2010.

[3] Chleboun, J. and Dohnalová, L.: Uncertainty quantification in multi-modeling (submitted).

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

[5] Dubois, D. and Prade, H.: *Fuzzy Sets and Systems: Theory and Applications (with a foreword by Lotfi A. Zadeh)*, *Mathematics in Science and Engineering*, vol. 144. Academic Press, Inc. (Harcourt Brace Jovanovich, Publishers), New York-London, 1980.

[6] Gao, D. Y.: Nonlinear elastic beam theory with application in contact problems and variational approaches. Mech. Res. Commun. **23** (1996), 11–17.

[7] Jin, S. S., Cha, S. L., and Jung, H. J.: Improvement of concrete creep prediction with probabilistic forecasting method under model uncertainty. Constr. Build. Mater. **184** (2018), 617–633.

[8] Machalová, J. and Netuka, H.: Control variational method approach to bending and contact problems for Gao beam. Appl. Math., Praha **62** (2017), 661–677.

[9] Meier, C., Popp, A., and Wall, W. A.: Geometrically exact finite element formulations for slender beams: Kirchhoff-Love theory versus Simo-Reissner theory. Arch. Computat. Methods. Eng. **26** (2019), 163–243.

[10] Nguyen, H. T.: A note on the extension principle for fuzzy sets. J. Math. Anal. Appl. **64** (1978), 369–380.

[11] Shafer, G.: *A Mathematical Theory of Evidence*. Princeton University Press, Princeton, 1976.

[12] Yager, R. R., Kacprzyk, J., and Fedrizzi, M. (Eds.): *Advances in the Dempster-Shafer theory of evidence*. Wiley, Chichester, 1994.

[13] Zimmermann, H. J.: *Fuzzy Set Theory—and Its Applications (with a foreword by L. A. Zadeh)*. Kluwer Academic Publishers, Boston, MA, 2001, fourth edn.

*Programs and Algorithms of Numerical Mathematics 22*
J. Chleboun, J. Papež, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics CAS, Prague 2025

# GALERKIN-TYPE SOLUTION OF NON-STATIONARY AEROELASTIC STOCHASTIC PROBLEMS

Cyril Fischer, Jiří Náprstek

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

**Abstract:** The assessment of vibration characteristics in slender engineering structures, influenced by both deterministic harmonic and stochastic excitation, poses a challenging problem. Due to its complexity, transverse vibration of the structure (relative to the wind direction) is typically modelled using the single-degree-of-freedom van der Pol-type equation. Determining the response probability density function comprises solving the Fokker-Planck equation, a task that generally necessitates the use of approximate numerical methods. Some of these methods rely on Galerkin-type approximation employing orthogonal polynomial or exponential-polynomial basis functions. This contribution reviews available techniques for stationary and non-stationary cases and proposes some modifications while highlighting unresolved questions in the field.

**Keywords:** van der Pol equation, random vibration, stochastic differential equation, quasiperiodic response, Fokker-Planck equation, Galerkin method

**MSC:** 35R60, 37A50, 65C30, 65M60

## 1. Introduction

Exploring the nonlinear dynamic response on random excitation is an important research subject. There are many analytical, semi-analytical, and numerical methods available to obtain stationary probability density functions (PDF) or statistical moments, particularly focusing on systems influenced by Gaussian white noise. However, the non-stationary case remains the subject of intensive research.

The non-linear van der Pol type single-degree-of-freedom (SDOF) oscillator is often used to represent transverse wind-induced vibrations under additive excitation, including deterministic and random components. This particular type of an oscillator is known and used for the so called *lock-in* or *frequency entrainment* effect, where the response frequency, i.e., vibration frequency of the structure, does not follow the dominant frequency present in the excitation but locks onto the natural frequency

DOI: 10.21136/panm.2024.05

of the system. This effect appears in a certain neighbourhood of the frequency of the stable limit cycle. Consequently, the oscillator produces very stable frequency output even with noisy harmonic input, provided the driving frequency remains within a certain proximity to the limit cycle frequency. Conversely, the response may attain various types of non-stationary response, including the cyclo-stationary or chaotic type when the driving frequency is far from the natural one.

The literature rarely addresses the van der Pol oscillator subjected to combined harmonic and random excitations. The stationary response case has been explored in [2], where the stochastic averaging method [5] and the equivalent linearization method are used in conjunction. The authors in [7] investigated a similar scenario, providing an explicit solution for the averaged equations in the resonant case. A more general yet stationary case has been outlined by the authors using the Galerkin method [6] to solve the nonlinear Fokker-Planck equation (FPE). The non-stationary case has been presented only recently, [11], where the probabilistic solution of the non-stationary responses is expressed as an exponential function of polynomial with time-variant coefficients and then the FPE is solved approximately.

This contribution reviews several approaches for determining both stationary and non-stationary response characteristics. For the stationary case, a method that refines the analytical solution available under exact resonance conditions is outlined, with a focus on the numerical integration procedure. In the non-stationary case, two approaches based on the Galerkin method are discussed: one utilizes a time-dependent linear combination of Hermite polynomials, while the other is based on exponential polynomials.

## 2. Mathematical model

Wind-induced vibration due to vortex shedding in slender engineering structures, such as bridge decks, towers, masts, high-rise buildings, or cables, is usually modelled using van der Pol equation. Its self-excitation due to the negative damping closely describes the state when the structure draws energy from the ambient flow. Mathematically,

$$\begin{aligned} \dot{u} &= v\,, \\ \dot{v} &= (\eta - \nu u^2)v - \omega_0^2 u + P\omega^2 \cos\omega t + h\xi(t)\,, \end{aligned} \tag{1}$$

where time differentiation is indicated by a dot above the symbol and the system parameters are:

- $u, v$ – the displacement $[m]$ and velocity $[ms^{-1}]$;
- $\eta, \nu$ – the linear and quadratic damping $[s^{-1}, s^{-1}m^{-2}]$;
- $\omega_0, \omega$ – the eigen-frequency of the linear SDOF system and frequency of the vortex shedding $[s^{-1}]$;

and the external excitation is described with: $f(t) = P\omega^2 \cos\omega t + h\xi(t)$, where:

- $P\omega^2$ – amplitude of the harmonic excitation $[ms^{-2}]$;
- $\xi(t)$ – the non-dimensional broadband Gaussian random process;
- $h$ – multiplicative constant $[ms^{-2}]$.

In the deterministic case, there are four basic configurations that characterize the solution in terms of frequency content and system solvability:

(i) The resonant case, where the excitation frequency is equal to the natural frequency $\omega_0 = \omega$. In this case the response of the model is periodic and, with random additive excitation, there exists an explicit expression for the stationary probability density of the response amplitude and phase shift [7].

(ii) When the frequency of the harmonic part of the right-hand side is close to the model's natural frequency, a lock-in effect occurs. The amplitudes of the deterministic solution are constant, and the response in the presence of stationary random disturbance remains stationary, [2, 6]. The width of the lock-in interval depends on system parameters.

(iii) Just beyond the boundary of the lock-in interval, in the deterministic case, a series of frequencies $\omega_i$ emerge in the frequency content of the response in addition to the natural frequency $\omega_0$. The new frequencies move away from the natural frequency $\omega_0$, depending on the distance of the excitation frequency from the boundary of the lock-in interval, approximately following the relationship $\omega_i = \omega_0 \pm \beta_i \left(\omega - \gamma^+\right)^{d_i}$ where $\gamma^+$ is the upper boundary of the lock-in interval, and $\beta_i, d_i$ are coefficients characteristic to the new frequencies. The presence of nearby frequencies in the response process results in the emergence of long-period beats at a frequency $|\omega_i - \omega_0|$, which give the response a quasiperiodic character. The analytic examination of this effect using the multiple scales method was recently published, [1].

This phenomenon causes ill conditioning of the behaviour of the van der Pol equation, where small errors in the excitation frequency lead to large changes in the nature of the solution. This effect is amplified in the presence of stochastic noise.

(iv) When the frequency of beats and the excitation frequency are comparable and/or the influence of self-excitation diminishes, the system's response is primarily characterized by the harmonic component of the excitation (forced vibrations). The response is periodic in the deterministic case and stationary in the stochastic case.

## 3. Stationary case

For weakly nonlinear systems subjected to weak excitations, the *stochastic averaging method* [9] is commonly employed. This method involves replacing fast variables with statistically equivalent stochastic processes to analyse variables evolving on a slower time-scale. The underlying assumption is that the response process can be uniformly approximated over a given time interval.

Using the Itô stochastic calculus, the response PDF of the original differential system Eq. (1) is governed by the *Fokker-Planck Equation*:

$$\frac{\partial p(\mathbf{x},t)}{\partial t} = -\sum_{j=1}^{N} \frac{\partial}{\partial x_j}\left(\kappa_j(\mathbf{x},t)p(\mathbf{x},t)\right) + \frac{1}{2}\sum_{j,k=1}^{N} \frac{\partial^2}{\partial x_j \partial x_k}(\kappa_{jk}(\mathbf{x},t)p(\mathbf{x},t)), \qquad (2)$$

where $\mathbf{x} = (x_1, x_2) = (u, v)$, $N = 2$. The drift coefficients $\kappa_j(\mathbf{x},t)$ correspond to the first moment of the derivative, while the diffusion coefficients $\kappa_{jk}(\mathbf{x},t)$ correspond to

the second moment. In the case of a stationary response process, $p(\mathbf{x},t)=p(\mathbf{x})$ and the left-hand side of Eq. (1) vanishes. The resulting equation is referred to as the *reduced Fokker-Planck equation*.

In the stochastic average method, the expressions for the displacement and velocity $u(t), v(t)$ are written in trigonometric form:

$$u(t)=a_c\cos\omega t+a_s\sin\omega t, \qquad v(t)=-a_c\omega\sin\omega t+a_s\omega\cos\omega t, \tag{3a}$$

where partial amplitudes $a_c, a_s$ comply with the additional condition

$$\dot{a}_c\cos\omega t+\dot{a}_s\sin\omega t=0. \tag{3b}$$

In the general case, $a_c(\tau), a_s(\tau)$ are functions of the slow time $\tau=\varepsilon t$, where $\varepsilon$ is a small parameter, and may represent non-stationary processes. In the lock-in region (i.e., in cases (i) and (ii) in the previous section), the response process is stationary, and the partial amplitudes $a_c$ and $a_s$ can be assumed stationary.

Based on the approximation Eq. (3), the original stochastic system Eq. (1) can be transformed using the time-averaging operator into the averaged Itô system:

$$\mathsf{d}a_c=\frac{\pi}{\omega}\left[\ \eta a_c+2\Delta a_s-\frac{1}{4}\nu\cdot a_c(a_c^2+a_s^2)\right]\mathsf{d}t+\left(\frac{\pi}{\omega}\Phi_{\xi\xi}\right)^{\frac{1}{2}}\mathsf{d}B_c, \tag{4a}$$

$$\mathsf{d}a_s=\frac{\pi}{\omega}\left[-2\Delta a_c+\eta a_s-\frac{1}{4}\nu\cdot a_s(a_c^2+a_s^2)\right]\mathsf{d}t+\frac{\pi}{\omega}P\omega\ \mathsf{d}t+\left(\frac{\pi}{\omega}\Phi_{\xi\xi}\right)^{\frac{1}{2}}\mathsf{d}B_s. \tag{4b}$$

Here $\Phi_{\xi\xi}(\omega)$ is the spectral density of the process $\xi(t)$ at frequency $\omega$, $B_{c,s}(t)$ stands for the Wiener process corresponding to input excitation $\xi(t)$ and $\Delta=(\omega_0^2-\omega^2)/(2\omega)$ is the frequency detuning.

The stationary PDF of $a_c$, $a_s$ follows from the reduced FPE:

$$\begin{aligned}&\frac{\partial}{\partial a_c}\left(\left[\eta a_c+2\Delta a_s-\frac{1}{4}\nu\cdot a_c(a_c^2+a_s^2)\right]p\right)-\frac{1}{2\omega}\Phi_{\xi\xi}(\omega)\frac{\partial^2 p}{\partial a_c^2}\\&+\frac{\partial}{\partial a_s}\left(\left[\eta a_s-2\Delta a_c-\frac{1}{4}\nu\cdot a_s(a_c^2+a_s^2)+P\omega\right]p\right)-\frac{1}{2\omega}\Phi_{\xi\xi}(\omega)\frac{\partial^2 p}{\partial a_s^2}=0,\end{aligned} \tag{5}$$

with boundary conditions assuring vanishing $p(a_c,a_s)$ for $|a_c|+|a_s|\to\infty$. The differential system Eq. (5) admits a closed-form solution under zero detuning (see [6] and Eq. (7)). The existence of such a solution depends on the existence of a probability density potential, which occurs only when $\Delta=0$.

### 3.1. Galerkin method

For non-zero detuning, but with a stationary response within the lock-in frequency range, a solution to the reduced, stationary Fokker-Planck equation for partial amplitudes can be sought in the form of a Galerkin approximation:

$$p(a_c,a_s)=p_0(a_c,a_s)\sum_{k,l=0}^{M,k}q_{kl}a_c^{k-l}a_s^l, \tag{6}$$

where $M$ is the upper limit of stochastic moments included into the analysis. In Eq. (6), $p_0(a_c, a_s)$ represents the weight function and is selected in the form of the solution to the stationary FPE when $\Delta = 0$:

$$p_0(a_c, a_s) \;=\; C \cdot \exp\left(\frac{\eta}{2S}\left[\left(a_s + \frac{P\omega}{\eta}\right)^2 + a_c^2 - \frac{\nu}{8\eta}\left(a_c^2 + a_s^2\right)^2\right]\right), \tag{7}$$

where $S = \Phi_{\xi\xi}(\omega)/(2\omega)$ and the normalizing factor $C$ is to be determined numerically.

When a harmonic component is present in the excitation, $P \neq 0$, the unsymmetric weight function fails to ensure the orthogonality of Hermite polynomials. Thus, for simplicity, standard polynomial basis and test functions are used, which, by virtue of the weight function $p_0$, satisfy the zero boundary conditions at infinity. For individual values of $k$, the terms in the sum in Eq. (6) represent the $k$-th stochastic moment and act as correction terms to the analytic solution for $\Delta = 0$.

### 3.2. Numerical integration

Integration in the Galerkin method takes place over the entire space $\mathbb{R}^2$, and the coefficients $q_{k,l}$ for $k, l = 0, \ldots, M; k + l \leq M$ are determined from the linear system obtained by substituting Eq. (6) into the FPE (5), followed by several steps of integration by parts and the application of homogeneous boundary conditions, where the specific forms of the partial derivatives of $p_0(a_c, a_s)$ were also taken into account:

$$\begin{aligned} 0 = \iint_{\mathbb{R}\times\mathbb{R}} \Bigg\{ & \left[a_c^{\sigma-2} a_s^{s-2}\left(\sigma(\sigma-1)a_s^2 - s(s-1)a_c^2\right) S + \Delta a_c a_s\left(\sigma a_s^2 - s a_c^2\right)\right] \sum_{k,l=0}^{M,k} q_{kl} a_c^{k-l} a_s^l \\ & - S\left[s\frac{\mathsf{d}}{\mathsf{d}a_s}\left(a_c^{\sigma} a_s^{s-1}\sum_{k,l=0}^{M,k} q_{kl} a_c^{k-l} a_s^l\right) - \sigma\frac{\mathsf{d}}{\mathsf{d}a_c}\left(a_c^{\sigma-1} a_s^{s}\sum_{k,l=0}^{M,k} q_{kl} a_c^{k-l} a_s^l\right)\right]\Bigg\} p_0 \,\mathsf{d}a_c \mathsf{d}a_s. \end{aligned} \tag{8}$$

where $\sigma = (r - s)$, $p_0 = p_0(a_c, a_s)$.

Basis functions in the form of polynomials have poor numerical properties because the corresponding Gram matrix is usually ill-conditioned. However, for low values of $M$ and with careful handling of the numerical integration, constructing the system matrix is feasible, especially when the following considerations are taken into account: Due to symmetry properties, terms involving odd powers of $a_c$ do not contribute to the total value of the integral and should be skipped during integration to avoid numerical cancellation. Additionally, the integral should be computed over the half-plane $a_c > 0$, with the result doubled. It is also convenient to transform the variables into polar coordinates centred at the maximum value of the weight function. In this way, the decrease of the integrand in the radial direction becomes roughly uniform.

The numerical integration in Eq. (6) involves a large number of terms of the form $z_{kl} = p_0(a_c, a_s) a_c^k a_s^l$; each of them approximately bounded from above on a logarith-

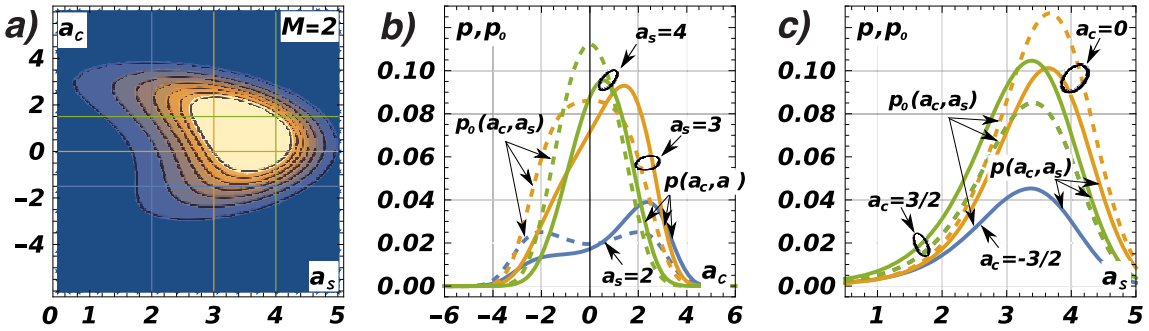



Figure 1: The Galerkin approximation of the stationary PDF for $M = 2$ and detuning value $\Delta = 0.10$. a) Contour plot of the PDF. b) "Vertical" sections of the PDF; $a_s = \{2,\ 3,\ 4\}$. c) "Horizontal" sections of the PDF; $a_c = \{-3/2, 0,\ 3/2\}$. In plots b,c: dashed is analytical solution $p_0(a_c, a_s)$, solid is Galerkin solution $p(a_c, a_s)$.

mic scale by the following estimate

$$\begin{aligned} \log|z_{kl}| \leq & \frac{1}{2S}\left(\eta\varrho^2 - \frac{\nu}{8}\left(\frac{P\omega}{\eta} - \varrho\right)^4\right) + l\log\left(\varrho - \frac{P\omega}{\eta}\right) + k\log(\varrho); \\ a_c = & \varrho\cos\varphi, \quad a_c = \varrho\sin\varphi - \frac{P\omega}{\eta}. \end{aligned} \tag{9}$$

The estimate Eq. (9) is useful for determining the required integration radius $\varrho$ and identifying the terms that contribute to the total value of the integral.

### 3.3. Numerical example

The PDF of the stochastic van der Pol oscillator response with respect to partial amplitudes $a_c$, $a_s$ is shown for $M = 2$ in Figure 1. The value of detuning $\delta = 0.10$ still represents the lock-in response. The contour plot of the estimated cross-PDF $p(a_c, a_s)$ is shown on the left. Plot b) depicts the sections of the PDF for fixed values $a_s = \{2,\ 3,\ 4\}$ and plot c) show sections for $a_c = \{-3/2,\ 0,\ 3/2\}$. The sections and the corresponding colors are indicated as horizontal/vertical lines in the left-hand plots. The dashed curves show the basic analytical solution which is valid for the case $\delta = 0$, i.e., no detuning is assumed. The estimates including the $M = 2$ Galerkin approximations are shown in solid.

## 4. Non-stationary response case

When studying the non-stationary case, the dependence on the original time coordinate must be retained. The FPE reflecting the original stochastic problem Eq. (1) in the original coordinates reads:

$$\frac{\partial p(\mathbf{x},t)}{\partial t} = -\sum_{j=1}^{2}\frac{\partial}{\partial x_j}\left(\kappa_j(\mathbf{x},t)p(\mathbf{x},t)\right) + \frac{1}{2}\sum_{j,k=1}^{2}\frac{\partial^2}{\partial x_j \partial x_k}\left(\kappa_{jk}(\mathbf{x},t)p(\mathbf{x},t)\right), \tag{10}$$

where $\mathbf{x} = (u, v)$; $x_1 = u$, $x_2 = v$. The input random process $\xi(t)$ is considered stationary and ergodic and the drift and diffusion coefficients can be written in a form:

$$\kappa_j(\mathbf{x}_t, t) = f_j(\mathbf{x}_t, t)\,, \quad \kappa_{jk}(\mathbf{x}_t, t) = \sum_{r=1}^{2} g_{jr}(\mathbf{x}_t, t) \int_{-\infty}^{\infty} g_{kr}(\mathbf{x}_{t+\tau}, t+\tau) R(\tau) \mathsf{d}\tau, \quad j, k = 1, 2, \tag{11}$$

where $R(\tau)$ is the auto-correlation function of $\xi(t)$.

Assuming that the detuning $\Delta \sim \varepsilon$ and the terms $(\eta - \nu u^2)\dot{u}$ and $P\omega^2$ are of a small order $\varepsilon$, and $h\xi(t)$ is of order $\varepsilon^{1/2}$. In such a case the FPE can be constructed for the SDE Eq. (1). It holds obviously:

$$\begin{aligned} &\kappa_1 = v\,, && \kappa_2 = (\eta - \nu u^2)v - \omega_0^2 u - P\omega^2 \cos\omega t, \\ &g_{11} = g_{12} = g_{21} = 0\,, && g_{22} = h, \\ &\kappa_{11} = \kappa_{12} = \kappa_{21} = 0\,, && \kappa_{22} = g_{22} \int_{-\infty}^{\infty} g_{22} R_{vv}(\tau) \mathsf{d}\tau = h^2 \sigma_{\xi\xi}^2 = h^2\, S, \end{aligned} \tag{12}$$

where $S$ is the variance of the process $\xi(t)$. Take a note that $\kappa_{22} = h^2 S$ is valid independently from a particular shape of the input process spectral density and formally it corresponds to the special case of $\xi$, which is the white noise ($\delta$ correlated), provided the excitation is a non-modulated additive stationary ergodic process. Anyway, the FPE can be readily written out as follows:

$$\frac{\partial p}{\partial t} = -\frac{\partial}{\partial u}(v\, p) - \frac{\partial}{\partial v}\left(\left((\eta - \nu u^2)v - \omega_0^2 u - P\omega^2 \cos\omega t\right) p\right) + \frac{1}{2} h^2 S \frac{\partial^2 p}{\partial v^2}, \tag{13}$$

together with initial and boundary conditions:

$$\lim_{u,v \to \pm\infty} p(u, v, t) = 0, \qquad p(u, v, 0) = \delta(u, v). \tag{14}$$

Near the boundary of the lock-in interval, the solution exhibits a quasi-periodic nature, which can be identified using a Galerkin-series-based solution in a form:

$$p(u, v, t) = p_0(u, v) \sum_{k=0}^{M} \sum_{l=0}^{k} q_{kl}(u, v, t). \tag{15}$$

The series Eq. (15) represents a weak solution to the FPE in the probabilistic sense. Choices of the weight function $p_0(u, v)$ and an approximation scheme used for terms $q_{kl}$ classify the available methods.

### 4.1. Galerkin solution based on Hermite polynomials

The challenges associated with numerical integration, discussed in the preceding section, have motivated the use of Hermite polynomials as basis functions which approximate the residuum between the weight function in the Galerkin method and

the solution of the FPE. However, the weight function must be adjusted to maintain the orthogonality property of the Hermite polynomials.

The elements $q_{kl}$ are formulated as follows:

$$q_{kl}(u,v,t) = q_{kl}(t)L_{k-l}(\alpha u)L_l(\beta v), \qquad \alpha^2 = \frac{\eta\omega_0^2}{h^2 S}, \quad \beta^2 = \frac{\eta}{h^2 S}\,, \tag{16}$$

where $L_k(x)$ are Hermite polynomials.

The weight function $p_0(u,v)$ is adopted in a form of the Boltzmann's solution to a related problem without damping and external excitation, [3]. In particular:

$$p_0(u,v) = C\,\exp\left(-\frac{2\eta}{h^2 S}H(u,v)\right), \tag{17}$$

where $C$ is the dimensionless normalizing constant, which can be put for now $C = 1$. $H(u,v)$ represents the Hamiltonian function of the basic system:

$$H(u,v) = \frac{1}{2}\omega_0^2 u^2 + \frac{1}{2}v^2, \tag{18}$$

which implicates $p_0(u,v) = p_u(u)p_v(v)$, so that $u, v$ are stochastically independent Gaussian processes on a level of the zero-th approximation.

The unknown functions $q_{kl}(u,v,t)$ in Eq. (16) are determined using the generalized method of stochastic moments [8]. The expression from Eq. (15) is substituted into Eq. (13), and both sides are multiplied by the test functions $\Phi_{rs}(u,v)$, which has the same formal expression as Eq. (16):

$$\Phi_{rs}(u,v) = L_{r-s}(\alpha u)L_s(\beta v), \quad r = 0,\ldots,M;\ s = 0,\ldots,r. \tag{19}$$

Subsequently, applying the expectation operator (which, in fact, involves integration over $\mathbb{R}^2$) to all permutations of the subscripts $r$ and $s$ establishes a sufficient number of ordinary differential equations for the unknown functions $q_{kl}(u,v,t)$.

Employing Hermite polynomials reduces computational cost and associated numerical errors. However, empirical evidence suggests that the convergence is relatively slow and, moreover, these basis functions do not guarantee the non-negativity of the computed PDF estimates, which can pose a substantial problem.

### 4.2. Exponential-polynomial-closure method

The issue of negative PDF estimates does not arise when using the exponential-polynomial-closure method (EPC), [4]. In the original stationary setting, it assumes the sought PDF of an approximate solution in the form of an exponential polynomial:

$$\tilde{p}(u,v;\mathbf{c}) = C\exp\left(Q_n(u,v;\mathbf{c})\right). \tag{20}$$

Here, $\mathbf{c}$ is the unknown parameter vector, and $Q_n(u,v;\mathbf{c})$ is a polynomial function. The algebraic system for unknown parameters $\mathbf{c}$ results from the Galerkin approximation with respect to basis functions $h_k(u,v) = u^r v^s f_N(u,v)$, where $k = r+s$ and $f_N$ is the PDF solution of the linearised Eq. (1) assuming the Gaussian response.

Multiple variants of the EPC method have been proposed for different settings of the stationary PDF solutions of nonlinear stochastic oscillators. Modifications for the non-linear, non-stationary case have only recently emerged, implicitly allowing for non-Gaussian excitation, [10]. The solution is assumed in an evolutionary form:

$$\tilde{p}(u,v,t;\mathbf{c}) = C\exp\left(Q_n(u,v,t;\mathbf{c})\right), \qquad Q_n(u,v,t;\mathbf{c}) = \sum_{i=1}^{n}\sum_{j=1}^{i} c_{ij}(t)u^{i-j}v^{j}. \quad (21)$$

Denoting by $\Delta(u,v,t;\mathbf{c})$ the residuum obtained by substitution Eq. (21) into the FPE (13), a set of ODEs for unknown parameters $\mathbf{c}(t) = \{c_{ij}(t)\}$ result from

$$\iint_{\mathbb{R}\times\mathbb{R}} \Delta(u,v,t;\mathbf{c})h_k(u,v)\mathsf{d}u\mathsf{d}v = 0, \quad k = 1\ldots M, \quad (22)$$

where $M$ indicates number of stochastic moments included into the solution.

## 5. Conclusions

The solution to the stochastic van der Pol equation is generally non-stationary and non-Gaussian, making its characterization a significant challenge. This paper reviews several approaches for determining both stationary and non-stationary response characteristics.

For the stationary case, the presented method is based the stochastic averaging method. The PDF for non-resonant configurations is approximated using the Galerkin method, where improper integrals are evaluated numerically. For this case, some new remarks regarding numerical integration were presented. However, due to the limitations of numerical integration for higher-degree polynomials, alternative basis functions are essential for exploiting higher stochastic moments.

Determining the non-stationary response relies on the Galerkin method, which must account for the time-dependence of the probability density. The paper explores two implementations. One approach utilizes a Boltzmann-type solution as the weight function and Hermite polynomials as basis and test functions in the Galerkin approximations. However, Hermite polynomials do not guarantee the non-negativity of the estimated PDF. As an alternative, the exponential-polynomial closure method is reviewed. It employs a Gaussian-closure solution of the linearised system as the weight function and exponential polynomials as basis and test functions. Based on existing literature, the EPC method is expected to outperform the previous approach. A comparative analysis of these implementations will be addressed in future work.

## Acknowledgment

The kind support of Czech Science Foundation project No. 24-13061S and of the RVO 68378297 institutional support are gratefully acknowledged.

## References


[1] Afzali, F., Kharazmi, E., and Feeny, B. F.: Resonances of a forced van der Pol equation with parametric damping. Nonlinear Dynam. **111** (2023), 5269–5285.

[2] Anh, N., Zakovorotny, V., and Hao, D.: Response analysis of van der pol oscillator subjected to harmonic and random excitations. Probabilist. Eng. Mech. **37** (2014), 51–59.

[3] Cai, G. and Lin, Y.: On exact stationary solutions of equivalent non-linear stochastic systems. Int. J. Non-Linear Mech. **23** (1988), 315–325.

[4] Er, G.-K.: Multi-Gaussian closure method for randomly excited non-linear systems. Int. J. Non-Linear Mech. **33** (1998), 201–214.

[5] Krylov, N. M. and Bogoliubov, N. N.: *Introduction to Non-Linear Mechanics*, *Annals of Mathematics Studies*, vol. 11. Princeton University Press, Princeton, 1947.

[6] Náprstek, J. and Fischer, C.: Averaging-based characteristics of the response induced by combined random and harmonic excitation. In: Sassi, S., Biswas, P., and Náprstek, J. (Eds.), *Proceedings of the 15th International Conference on Vibration Problems. ICOVP 2023*. Lecture Notes in Mechanical Engineering. Springer, Singapore, 2024, pp. 191–202.

[7] Náprstek, J., Fischer, C., Pospíšil, S., and Trush, A.: Modeling of the quasi-periodic galloping response type under combined harmonic and random excitation. Comput. Struct. **247** (2021), 106 478.

[8] Pugachev, V. S. and Sinitsyn, I. N.: *Stochastic differential systems — Analysis and filtering*. J. Willey, Chichester, 1987.

[9] Roberts, J. B. and Spanos, P. D.: Stochastic averaging: An approximate method of solving random vibration problems. Int. J. Non-Linear Mech. **21** (1986), 111–134.

[10] Wang, K., Wang, J., Jia, S., Zhu, Z., Yu, Z., and Xu, L.: Non-stationary nonzero mean probabilistic solutions of nonlinear stochastic oscillators subjected to both additive and multiplicative excitations. Chinese J. Phys. **81** (2023), 64–77.

[11] Zhu, Z., Gong, W., Yu, Z., and Wang, K.: Investigation on the EPC method in analyzing the nonlinear oscillators under both harmonic and Gaussian white noise excitations. J. Vib. Control **29** (2023), 2935–2949.

*Programs and Algorithms of Numerical Mathematics 22*
J. Chleboun, J. Papež, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics CAS, Prague 2025

# NUMERICAL STUDY OF TWO-LEVEL ADDITIVE SCHWARZ PRECONDITIONER FOR DISCONTINUOUS GALERKIN METHOD SOLVING ELLIPTIC PROBLEMS

Tomáš Hammerbauer, Vít Dolejší

Faculty of Mathematics and Physics, Charles University
Ke Karlovu 2027/3, 121 16 Praha, Czech Republic
hammerbt@karlin.mff.cuni.cz, dolejsi@karlin.mff.cuni.cz

**Abstract:** The paper deals with the analysis and numerical study of the domain decomposition based preconditioner for algebraic systems arising from the discontinuous Galerkin (DG) discretization of the linear elliptic problems. We introduce the DG discretization of the model problem and present the spectral *hp*-bound of the corresponding linear algebraic systems. Moreover, we present the two-level additive Schwarz preconditioner together with the theoretical result related to the estimate of the condition number. Finally, we present the numerical experiments supporting the theoretical results and demonstrate the efficiency of this approach for the solution of nonlinear problems.

**Keywords:** domain decomposition, elliptic partial differential equation, two-level additive Schwarz preconditioner

**MSC:** 65N15, 65M15, 65F08

## 1. Introduction

Discontinuous Galerkin method (DGM) became a very popular method for solving partial differential equations, cf. [5]. DGM is based on a piecewise polynomial but discontinuous approximation where the inter-element continuity is replaced by special terms. The DGM exhibits a very robust, accurate, and efficient technique for various problems. On the other side the DG discretization leads to large sparse algebraic systems, whose solution usually exhibits the most time-consuming part of the whole computational process.

The domain decomposition techniques exhibit a powerful strategy, which allows to split the computational work and employ the parallel power of modern supercomputers. One possibility is to split the given problem in several smaller sub-problems with suitably chosen interface conditions, and solve them iteratively to coordinate the solution between neighbouring subdomains, cf. monographs [3, 11]. However,

DOI: 10.21136/panm.2024.06

more frequent is to use the domain decomposition methods as preconditioners for Krylov subspace iterative methods, such as the conjugate gradient method. In this paper, we focus on the two-level additive Schwarz (AS) preconditioner (cf. [2, 9]). In particular, we present theoretical results related to the condition number of the preconditioned system and several numerical examples demonstrating the efficiency of this technique.

In Section 2, we introduce the discretization of the linear model problem by *hp*-variant of the discontinuous Galerkin method (DGM), and present the *hp*-bound of the condition number of the corresponding algebraic system. In Section 3, we formulate the two-level additive Schwarz preconditioner and present the bounds of the condition number of the preconditioned system arising from DGM. In Section 4, we introduce the results of numerical experiments performed to support the analysis. These experiments are the main contribution since they show that the approach also works for nonlinear cases and leads to improved computational time when used with the parallel computations. Several concluding remarks are given in Section 5.

## 2. Discontinuous Galerkin method

We are dealing with the following symmetric linear elliptic problem

$$\begin{aligned} -\mathrm{div}(\boldsymbol{K}\nabla u) &= f \quad \text{in } \Omega \\ u &= 0 \quad \text{on } \partial\Omega, \end{aligned} \tag{1}$$

where $\Omega \in \mathbb{R}^d$, $d = 2, 3$ is a bounded domain with polygonal Lipschitz boundary $\partial\Omega$ and $\boldsymbol{K} = \boldsymbol{K}(x)$ is a symmetric positive definite matrix in $\mathbb{R}^{d\times d}$. We assume that $\exists k_0, k_1 > 0$, independent of $x \in \Omega$, such that $k_0|\xi| \leq |\boldsymbol{K}\xi| \leq k_1|\xi|$ $\forall \xi \in \mathbb{R}^d$. For simplicity, we assume the homogeneous Dirichlet boundary condition, however the results can be easily extended to other boundary conditions. Finally, we use the notation $L^2(M)$ for the Lebesgue space of square-integrable functions over $M \subset \mathbb{R}^d$, $d = 2, 3$ and we denote by $(\cdot, \cdot)_\Omega$ the standard inner product in $L^2(\Omega)$.

### 2.1. Discretization of domain $\Omega$

Let $\mathcal{T}_h$, $h > 0$ be a partition of the domain $\bar{\Omega}$ into non-overlapping triangles $K$ such that $\bigcup_{K\in\mathcal{T}_h} \overline{K} = \overline{\Omega}$. We set $h = \max_{K\in\mathcal{T}_h} h_K$, where $h_K$ is the diameter of the element $K$, $K \in \mathcal{T}_h$, and we denote by $\partial K$ the boundary of $K \in \mathcal{T}_h$.

In addition, let $\mathcal{F}_h$ be the set of all faces $\gamma$ of $\mathcal{T}_h$ and we put

$$\mathcal{F}_h^B = \{\gamma \in \mathcal{F}_h \colon \gamma \subset \partial\Omega\} \qquad \text{and} \qquad \mathcal{F}_h^I = \mathcal{F}_h \setminus \mathcal{F}_h^B$$

for boundary and interior edges, respectively. For each $\gamma \in \mathcal{F}_h^I$ we consider a unit normal vector $\boldsymbol{n}_\gamma$ whose orientation can be arbitrarily chosen. If $\gamma \in \mathcal{F}_h^B$, the unit normal $\boldsymbol{n}_\gamma$ is outer to $\partial\Omega$.

Let $\boldsymbol{p} := \{p_K \colon K \in \mathcal{T}_h\}$ be a set of integers that assigns to each triangular element its polynomial degree of approximation. We assume that the ratio of polynomial approximation degrees of any two neighboring elements is bounded.

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

$$S_{hp} := \{v \in L^2(\Omega) \colon v|_K \in P_{p_K}(K)\, \forall K \in \mathcal{T}_h\},$$

where $P_{p_K}(K)$ denotes the space of polynomials of degree less or equal than $p_K$ on $K$.

By $v|^+$ and $v|^-$ we denote the traces of function $v \in S_{hp}$ on $\gamma \in \mathcal{F}_h^I$ in the direction of $\boldsymbol{n}_\gamma$ and opposite the direction of $\boldsymbol{n}_\gamma$, respectively. Using this notation we define the jump $[\cdot]_\gamma$ and the mean value $\langle\cdot\rangle_\gamma$ of $v \in S_{hp}$ by

$$[v]_\gamma = v|^+ \boldsymbol{n}_\gamma - v|^- \boldsymbol{n}_\gamma \quad \text{and} \quad \langle v\rangle_\gamma = \frac{1}{2}(v|^+ + v|^-), \quad \gamma \in \mathcal{F}_h^I, \tag{2}$$

respectively. For $\gamma \in \mathcal{F}_h^B$, we set $[v]_\gamma = v\boldsymbol{n}_\gamma$ and $\langle v\rangle_\gamma = v$. Usually, we drop the subscript $\gamma$.

Finally, we assume that the mesh is *shape-regular* and *quasi-uniform*. Then we set the edge size by

$$h_\gamma := \max(h_K, h_{K'}) \qquad \gamma \subset \partial K \cap \partial K'.$$

More details can be found in [5, Chapter 2.3].

### 2.2. Primal formulation of DGM

We introduce the approximate solution of our problem, more details can be found, e.g., in [5, Chapter 2.4]. Using (2), we define the billinear form $\mathcal{A}_h(u, v)$ by

$$\begin{aligned}\mathcal{A}_h(u,v) := &\sum_{K\in\mathcal{T}_h} \int_K \boldsymbol{K}\nabla u \cdot \nabla v \,\mathrm{dx} - \sum_{\gamma\in\mathcal{F}_h^I} \int_\gamma (\langle \boldsymbol{K}\nabla u\rangle \cdot [v] + \langle \boldsymbol{K}\nabla v\rangle \cdot [u]) \,\mathrm{dS} \\ &+ \sum_{\gamma\in\mathcal{F}_h^I} \int_\gamma \sigma\,[u]\,[v] \,\mathrm{dS}, \qquad u, v \in S_{hp}.\end{aligned}$$

The last term is called the interior penalty term and is supposed to mimic the continuity of the approximate solution at the interior edges. The penalty parameter $\sigma$ is given by

$$\sigma|_\gamma = \sigma_\gamma = \alpha \frac{k_0 p_\gamma^2}{h_\gamma}, \quad \gamma \in \mathcal{F}_h^I,$$

where the constant $\alpha$ is chosen such that we have guaranteed the coercivity of the form $\mathcal{A}_h$, see [5, Chapter 2.6.3].

**Definition 1.** The function $u_h \in S_{hp}$ is called the *approximate solution* of (1) if

$$\mathcal{A}_h(u_h, v) = (f, v)_\Omega \quad \forall v \in S_{hp}. \tag{3}$$

This scheme is called the *symmetric interior penalty Galerkin* (SIPG) method.

The discrete problem (3) is equivalent to the system of linear algebraic equations

$$\boldsymbol{A}\boldsymbol{u} = \boldsymbol{f}, \tag{4}$$

where $\boldsymbol{A}$ is the matrix having the size $n$ equal to dimension of $S_{hp}$ and the entries of $\boldsymbol{A}$ are given by $\mathcal{A}_h(\phi_j, \phi_i)$, where $\{\phi_i,\ i = 1, \ldots, n\}$ is a basis of $S_{hp}$. If the size of $\boldsymbol{A}$ is large, the use of iterative solvers is advantageous. Very efficient are methods based on Krylov subspaces, among them the *conjugate gradient* (CG) method is very popular for symmetric problems. The rate of convergence of CG can be estimated by the condition number, cf. [10, Chapter 6.11]. In [2, Section 2.4] and [8, Section 2], the following estimate of the condition number of $\boldsymbol{A}$ from (4) was derived

$$\kappa(\boldsymbol{A}) \leq C \frac{k_1}{k_0} p^4 h^{-2} \tag{5}$$

for uniform grids having mesh step $h$ and constant polynomial approximation degree $p$. We aim to use the domain decomposition to construct suitable preconditioner for the algebraic system (4), such that it decreases its condition number and can be performed in parallel setting.

## 3. Additive Schwarz preconditioner

We start with the partition of the computational domain $\Omega$ into smaller non-overlapping subdomains $\Omega_i$ such that $\overline{\Omega} = \bigcup_{i=1}^{N} \overline{\Omega_i}$. We assume that the subdomains $\Omega_i$ are the union of elements of $\mathcal{T}_h$. We employ two-level method, hence we define a coarse mesh $\mathcal{T}_H$ such that $\mathcal{K} \in \mathcal{T}_H$ lies in one subdomain $\Omega_i$. We assume that the partitions are *nested*, i.e. the elements from a coarser mesh are the union of elements of finer mesh, these elements can be non-convex, see Figure 1 for two examples.

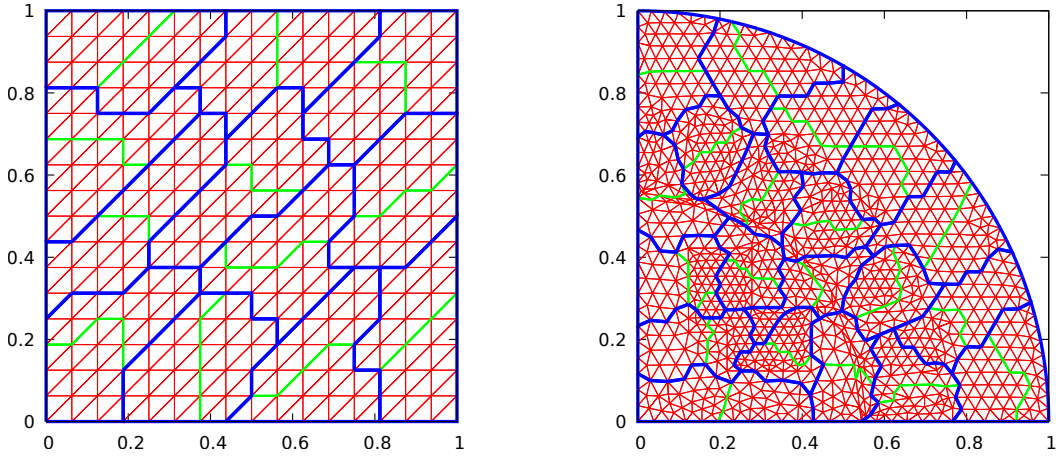



Figure 1: Examples of two fine meshes $\mathcal{T}_h$ (red, thin), subdomains $\Omega_i$ (blue, thick) and coarse meshes $\mathcal{T}_H$ (green, thin).

In the following, we introduce the local bilinear forms corresponding to the restriction of $\mathcal{A}_h$ on the subdomains $\Omega_i$, $i = 1, \dots, N$ and the coarse (global) form corresponding to the restriction of $\mathcal{A}_h$ on the coarse mesh $\mathcal{T}_H$. The forms build the projection operators which are used for the definition of the two-level additive Schwarz preconditioner. For more details, we refer to, e.g, [1, 2].

### 3.1. Local forms

We consider a restriction of the space $S_{hp}$ onto each sub-domain $\Omega_i$, $i = 1, \dots, N$, i.e.

$$S_{hp}^i = \{u \in L^2(\Omega_i) \colon u|_K \in P_{p_K},\ K \in \mathcal{T}_h,\ K \subset \Omega_i\}, \qquad i = 1, \dots, N.$$

We define the *prolongation operators* $R_i^T \colon S_{hp}^i \to S_{hp}$ by

$$R_i^T u_i = \begin{cases} u_i & \text{on } \Omega_i, \\ 0 & \text{on } \Omega \setminus \Omega_i, \end{cases} \qquad u_i \in S_{hp}^i.$$

The corresponding (dual) *restriction operators* $R_i \colon S_{hp} \to S_{hp}^i$ are given by $R_i u = u|_{\Omega_i}$, $i = 1, \dots, N$. Then, we introduce the local bilinear forms $\mathcal{A}_{h,i}$

$$\mathcal{A}_{h,i}(u_i, v_i) := \mathcal{A}_h(R_i^T u_i, R_i^T v_i), \qquad u_i, v_i \in S_{hp}^i,\ i = 1, \dots N.$$

Using the prolongation operators, we can express functions from the space $S_{hp}$ as a linear combination of functions from the local spaces.

### 3.2. Coarse form

In order to increase the speed of the transfer of the information among the subdomains, we formulate the problem on the coarse space $S_{hp}^0$ corresponding to the mesh $\mathcal{T}_H$. To deal with the inconsistency of the polynomial degree, we introduce the quantity $q_{\mathcal{K}}$, $\mathcal{K} \in \mathcal{T}_H$ defined by

$$0 \le q_{\mathcal{K}} \le \min_{K \subset \mathcal{K}} p_K.$$

The definition of the coarse space $S_{hp}^0$ is done similary as in the local space case, i.e.

$$S_{Hp}^0 := \{v \in L^2(\Omega) \colon v|_{\mathcal{K}} \in P_{q_{\mathcal{K}}}(\mathcal{K}),\ \mathcal{K} \in \mathcal{T}_H\}$$

Moreover, we define the prolongation operator $R_0^T \colon S_{Hp}^0 \to S_{hp}$ as a classical injection of the space $S_{Hp}^0$ in $S_{hp}$, and *restriction operator* $R_0 \colon S_{Hp} \to S_{hp}^0$ as its dual. Then, we set

$$\mathcal{A}_{h,0}(u_0, v_0) := \mathcal{A}_h(R_0^T u_0, R_0^T v_0), \qquad u_0, v_0 \in S_{Hp}^0.$$

### 3.3. Projection and preconditioned operators

Finally we define the local projection operators $\tilde{P}_i$, $i = 0, \dots, N$ which project the function onto the space $S^i_{hp}$ using the local forms $\mathcal{A}_{h,i}$. Namely,

$$\tilde{P}_i \colon S_{hp} \to S^i_{hp} \qquad \mathcal{A}_{h,i}(\tilde{P}_i u, v_i) = \mathcal{A}_h(u, R_i^T v_i) \quad \forall v_i \in S^i_{hp}, \quad i = 0, \dots, N.$$

For the projector on the space $S_{hp}$ we use the definition

$$P_i := R_i^T \tilde{P}_i \; : \; S_{hp} \to S_{hp}, \quad i = 0, \dots, N.$$

Finally, the *two-level additive Schwarz operator* reads

$$P_{ad} := \sum_{i=0}^{N} P_i. \tag{6}$$

### 3.4. Algebraic representation

We introduce the algebraic representation of the local bilinear forms $\mathcal{A}_{h,i}$ and the projector operators $\tilde{P}_i$ and $P_i$, $i = 0, \dots, N$ from previous paragraphs. Let $n = \dim(S_{hp})$, $n_i = \dim(S^i_{hp})$, $i = 1, \dots, N$, and $n_0 = \dim(S^0_{Hp})$. Let $\boldsymbol{R}_i^T \in \mathbb{R}^{n \times n_i}$, $i = 0, \dots, N$ be the matrices corresponding to the prolongation operators $R_i^T$ with respect to the used basis of $S_{hp}$. Their construction is simple since $S^i_{hp} \subset S_{hp}$, $i = 1, \dots, N$ and $S^0_{Hp} \subset S_{hp}$. Then the algebraic representations of the restriction operators $R_i$, $i = 0, \dots, N$ are just the transposed matrices $\boldsymbol{R}_i = (\boldsymbol{R}_i^T)^T$.

Moreover, the algebraic representation of the local bilinear forms $\mathcal{A}_{h,i}$ are matrices $\boldsymbol{A}_i = \boldsymbol{R}_i \boldsymbol{A} \boldsymbol{R}_i^T \in \mathbb{R}^{n_i \times n_i}$, $i = 0, \dots, N$. Consequently, the matrix representation of projection operators $\tilde{P}_i$ and $P_i$ reads

$$\tilde{\boldsymbol{P}}_i = \boldsymbol{A}_i^{-1} \boldsymbol{R}_i \boldsymbol{A} \quad \text{and} \quad \boldsymbol{P}_i = \boldsymbol{R}_i^T \boldsymbol{A}_i^{-1} \boldsymbol{R}_i \boldsymbol{A}, \qquad i = 0, \dots, N,$$

respectively. Finally, the matrix representation of the additive Schwarz operator is given by

$$\boldsymbol{P}_{ad} = \sum_{i=0}^{N} \boldsymbol{P}_i = \sum_{i=0}^{N} \boldsymbol{R}_i^T \boldsymbol{A}_i^{-1} \boldsymbol{R}_i \boldsymbol{A} =: \boldsymbol{M}_{ad}^{-1} \boldsymbol{A}. \tag{7}$$

Hence, the matrix $\boldsymbol{M}_{ad}^{-1}$ is a preconditioner of system (4) arising from DG discretization. Therefore, we replace (4) by the equivalent problem

$$\boldsymbol{M}_{ad}^{-1} \boldsymbol{A} \boldsymbol{u} = \boldsymbol{M}_{ad}^{-1} \boldsymbol{f}, \tag{8}$$

where the application of $\boldsymbol{M}_{ad}^{-1}$ exhibits a solution of small algebraic systems which can be done in a parallel way. For the solution of (8), we use standard Krylov iterative solver, namely the conjugate gradient (CG) method. More details on the solver can be found in [10, Chapter 6].

### 3.5. Analysis of the preconditioner

In this section, we present the upper bound of the condition number of the matrix $\boldsymbol{P}_{ad} = \boldsymbol{M}_{ad}^{-1}\boldsymbol{A}$ using the abstract technique from [11, Chapter 2], which is based on the following three assumptions.

**Assumption 1** (Stable decomposition) There exists a constant $C_0 > 0$ such that $\forall u \in S_{hp}$ we have the decomposition $u = \sum_{i=0}^{N} R_i^T u_i$, with $u_0 \in S_{Hp}^0$, $u_i \in S_{hp}^i$, $i = 1, \ldots, N$, that satisfies $\sum_{i=0}^{N} \mathcal{A}_{h,i}(u_i, u_i) \leq C_0^2 \mathcal{A}_h(u, u)$.

**Assumption 2** (Local stability) There exists a constant $\omega$, $0 \leq \omega \leq 2$, such that

$$\mathcal{A}_h(R_i^T u_i, R_i^T u_i) \leq \omega \mathcal{A}_{h,i}(u_i, u_i) \quad \forall u_i \in S_{hp}^i,\ i = 1, \ldots, N,$$
$$\mathcal{A}_h(R_0^T u_0, R_0^T u_0) \leq \omega \mathcal{A}_{h,0}(u_0, u_0) \quad \forall u_0 \in S_{Hp}^0.$$

**Assumption 3** (Strengthened Cauchy-Schwarz inequalities) There exist constants $0 \leq \epsilon_{ij} \leq 1$, $i, j = 1, \ldots, N$, such that

$$|\mathcal{A}_h(R_i^T u_i, R_j^T u_j)| \leq \epsilon_{ij} \mathcal{A}_h(R_i^T u_i, R_i^T u_i)^{\frac{1}{2}} \mathcal{A}_h(R_j^T u_j, R_j^T u_j)^{\frac{1}{2}}, \quad i, j = 1, \ldots N,$$

for all $u_i \in S_{hp}^i$, $u_j \in S_{hp}^j$. By $\rho(\boldsymbol{\varepsilon})$ we denote the spectral radius of $\boldsymbol{\varepsilon} = \{\epsilon_{ij}\}_{i,j=0}^{N}$

Using [11, Theorem 2.7], we have the following results.

**Theorem 1.** *Let Assumptions 1–3 be satisfied. Then the condition number of the two-level additive Schwarz operator can be bounded by*

$$\kappa(P_{ad}) \leq C_0^2\, \omega\, (\rho(\boldsymbol{\varepsilon}) + 1).$$

Verifying Assumptions 1–3 for the presented additive Schwarz formulation and using Theorem 1, cf. [2, 9], we get the bound

$$\kappa(\boldsymbol{P}_{ad}) \leq C\alpha \frac{p^2 H k_1}{q h k_0}, \tag{9}$$

where $\boldsymbol{P}_{ad}$ is given by (7).

## 4. Numerical study

The objective of this section is to numerically compute the bounds (5) and (9) and to demonstrate their accuracy. We focus on the experiments dealing with the condition number of the non-preconditioned systems and also the preconditioned systems. In the end, we show that the application of this approach can be used to solve non-linear problems.

All experiments were performed using the ADGFEM code [4] for the generation of the system matrices and then exported to MATLAB, where we used the function `eigs` to compute the approximations of the largest and smallest eigenvalues of $\boldsymbol{A}$ and $\boldsymbol{M}_{ad}^{-1}\boldsymbol{A}$. Then we set the condition number as the ratio of the largest and smallest eigenvalues. This approach is valid since we are dealing with symmetric positive definite matrices. We investigate the dependence of condition number $\kappa(\boldsymbol{A})$ and $\kappa(\boldsymbol{M}_{ad}^{-1}\boldsymbol{A})$ on the parameters $h$, $H$, $p$ and the ratio of $k_1/k_0$ as we have seen.

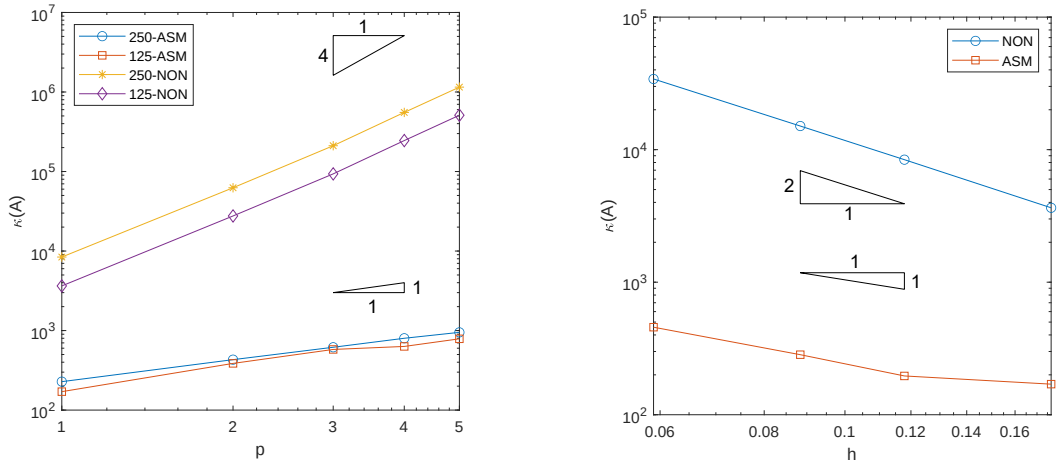



Figure 2: The dependence of $\kappa(\boldsymbol{A})$ (NON) and $\kappa(\boldsymbol{M}_{ad}^{-1}\boldsymbol{A})$ (ASM) on the polynomial degree $p$ (left) and on the mesh size $h$ using $p = 1$ (right).

It is important to say that similar numerical examples were performed in [8], but there we had not quite correct implementation of system matrix generation and the condition number was computed using the function `condest`, which computes different type of condition number.

### 4.1. Laplace equation

We consider the problem (1) with $\boldsymbol{K} = \boldsymbol{I}$, where $\boldsymbol{I}$ is the identity matrix and with $\Omega = (0, 1)^2$. The corresponding mesh is on the left of Figure 1. Since $k_0 = k_1 = 1$ the results (5) and (9) depend only on $h$, $H$ and $p$. Similarly as in [2], we plot the dependence of $\kappa(\boldsymbol{A})$ in logarithmic scale to see the slope.

- First, we investigate the dependence of $\kappa(\boldsymbol{A})$ and $\kappa(\boldsymbol{M}_{ad}^{-1}\boldsymbol{A})$ on $p$ for two uniform meshes having (approximately) 125 and 250 elements. We set $N = 12$, each $\Omega_i$ is one coarse element, and the coarse polynomial degree is set $q = p$.
- Moreover, we investigate the dependence of $\kappa(\boldsymbol{A})$ on $h$ for $p = 1$, where we use meshes having 128, 288, 512 and 1152 mesh elements.

Figure 2, left shows that $\kappa(\boldsymbol{A})$ behaves as $O(p^4)$ and $\kappa(\boldsymbol{M}_{ad}^{-1}\boldsymbol{A})$ behaves as $O(p)$ which is in agreement with (5) and (9). Moreover, Figure 2, right shows that $\kappa(\boldsymbol{A})$ behaves as $O(h^2)$ and $O(h)$, which is again expected based on the result (5) and (9), respectively.

### 4.2. Symmetric linear elliptic equation

Furthermore, we deal with a linearization of the example from [7, Section 5.4]. This corresponds to a simulation of the magnetostatic field in the alternator. Due to symmetry, we consider only a quarter of the alternator. The domain $\Omega$ is divided into $\Omega_s$ (Stator), $\Omega_r$ (Rotor), and $\Omega_a$ (Air) (geometry can be seen in [8, Figure 3.6]).

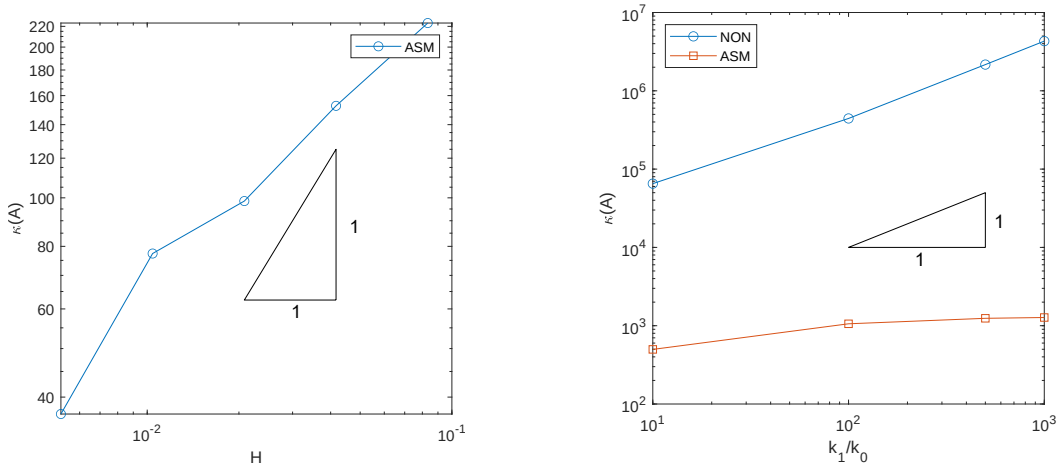



Figure 3: The dependence of the condition number of the preconditioned system (ASM) on the coarse mesh size (left) and the dependence of the condition number of the preconditioned and non-preconditioned system on the ration of $k_1/k_0$ for $p = 1$ and $p = 2$ (right).

The corresponding mesh can be seen on the right of Figure 1. The formulation in terms of the magnetic potential $u$ reads:

$$-\mathrm{div}(\nu(x)\nabla u(x)) = f \quad \text{in } \Omega, \tag{10}$$

where $\nu$ is in the form $\nu(x) = \begin{cases} \frac{1}{\mu_0} & \text{for } x \in \Omega_a, \\ \frac{100}{\mu_0} & \text{for } x \in \Omega_s \cup \Omega_a, \end{cases}$ where $\mu_0 = 1.256 \cdot 10^{-6}$.

We use the same technique as described above to generate system matrices and compute the condition number. We focus on the following.

- We investigate the dependence of $\kappa(\boldsymbol{M}_{ad}^{-1}\boldsymbol{A})$ on the coarse mesh size $H$ with $p = 1$ and $N = 12$ and the division of the subdomains $\Omega_i$ into 1,2,4,8 and 12 coarse elements.
- We investigate the dependence of $\kappa(\boldsymbol{A})$ and $\kappa(\boldsymbol{M}_{ad}^{-1}\boldsymbol{A})$ on the ratio of $k_1/k_0$ for $p = 1$.

Figure 3, left supports the theoretical result $\kappa(\boldsymbol{M}_{ad}^{-1}\boldsymbol{A}) = O(H)$, at least asymptotically. Figure 3, right gives that the dependency on the ratio of $k_1/k_0$ is also in agreement with the result (5) and (9), in which we see that $\kappa(\boldsymbol{M}_{ad}^{-1}\boldsymbol{A})$ and $\kappa(\boldsymbol{A})$ behaves as $O(k_1/k_0)$. We can see that we are getting slightly better result for the preconditioned system than we expected.

### 4.3. Symmetric nonlinear elliptic equation

Finally we present numerical result for the nonlinear variant of the alternator equation (10), namely

$$-\mathrm{div}(\nu(x, |\nabla u(x)|^2)\nabla u(x)) = f \quad \text{in } \Omega,$$

| $N$ | non-linear iter | linear iter | time on 1 processor | theoretical $\frac{time \times 2}{\#\Omega_i}$ |
|---|---|---|---|---|
| 4 | 81 | 7049 | 158 s | 79 s |
| 8 | 71 | 8073 | 158 s | 39 s |
| 16 | 68 | 8939 | 156 s | 19 s |
| 32 | 71 | 10493 | 194 s | 12 s |
| 64 | 70 | 11649 | 230 s | 7 s |

Table 1: Number of iterations and the computational time for increasing number of subdomains.

where the function $\nu$ is a strongly nonlinear function in $|\nabla u(x)|$, see [7, Section 5.4] for the explicit form of $\nu$. The nonlinear problem is solved as a sequence of linear ones, namely

$$-\mathrm{div}(\nu(x, |\nabla u_{k-1}|)\nabla u_k) = f \quad \text{in } \Omega, \qquad k = 1, 2, \dots$$

where $k$ is the index of nonlinear iterations. In every non-linear iteration we compute 100 linear iterations. As the linear solver we used conjugate gradient method with the two-level additive Schwarz preconditioner (7). The stopping criterion was the ratio of algebraic residual error estimator over the space residual error estimator, cf. [6].

We investigate the speed of convergence for increasing number of subdomains $N = \#\Omega_i$, each $\Omega_i$ is just one coarse element $\mathcal{K} \in \mathcal{T}_H$. The number of conjugate gradient iterations and computational time in seconds (using one processor) is shown in Table 1. Although the computations were performed using one processor, we present in the last column of Table 1 the theoretical computational time using an ideal parallelization, i.e., one processor for one subdomain (excluding overheads). We observe an almost optimal speed up of the computation.

## 5. Conclusion

We presented the outline of the theory used for the condition number bounds of the two-level additive Schwarz preconditioner for the solution of partial differential equations using DGM. The main part of our work was the numerical study done on a more complex example and also the application of the method for the non-linear problem. We have shown that the method has potential for non-linear problems and can be further investigated.

## References

[1] Antonietti, P., Giani, S., and Houston, P.: Domain decomposition preconditioners for discontinuous Galerkin methods for elliptic problems on complicated domains. J. Sci. Comput. **60** (2014), 203–227.

[2] Antonietti, P. and Houston, P.: A class of domain decomposition preconditioners for *hp*-discontinuous Galerkin finite element methods. J. Sci. Comput. **46** (2011).

[3] Dolean, V., Jolivet, P., and Nataf, F.: *An Introduction to Domain Decomposition Methods*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2015.

[4] Dolejší, V.: *ADGFEM – Adaptive discontinuous Galerkin finite element method, in-house code*. Charles University, Prague, Faculty of Mathematics and Physics, 2020. https://msekce.karlin.mff.cuni.cz/~dolejsi/adgfem/index.html.

[5] Dolejší, V. and Feistauer, M.: *Discontinuous Galerkin Method: Analysis and Applications to Compressible Flow*. Springer Series in Computational Mathematics, Springer International Publishing, 2015.

[6] Dolejší, V., Roskovec, F., and Vlasák, M.: Residual based error estimates for the space-time discontinuous Galerkin method applied to the compressible flows. Comput. Fluids **117** (2015), 304–324.

[7] Dolejší, V. and Congreve, S.: Goal-oriented error analysis of iterative Galerkin discretizations for nonlinear problems including linearization and algebraic errors. J. Comput. Appl. Math. **427** (2023), 115 134.

[8] Hammerbauer, T.: *Domain decomposition methods for the solution of partial differential equations using discontinuous Galerkin method*. Master's thesis, Charles University, Prague, CZ, 2024.

[9] Krzyżanowski, P.: On a nonoverlapping additive Schwarz method for *h-p* discontinuous Galerkin discretization of elliptic problems. Numer. Methods Partial Differ. Equations **32** (2016), 1572–1590.

[10] Saad, Y.: *Iterative methods for sparse linear systems*. The PWS series in computer science, PWS, Boston, 1996.

[11] Toselli, A. and Widlund, O.: *Domain decomposition methods-algorithms and theory*, vol. 34. Springer Science & Business Media, 2004.

*Programs and Algorithms of Numerical Mathematics 22*
J. Chleboun, J. Papež, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics CAS, Prague 2025

# ON THE POSSIBILITIES OF COMPUTATIONAL MODELLING OF INTERACTION OF A STRUCTURE WITH SUBSOIL

Michal Jedlička[1,2], Ivan Němec[2], Jiří Vala[3,4]

[1] Brno University in Technology, Faculty of Civil Engineering, Institute of Structural Mechanics, 602 00 Brno, Veveří 95, Czech Republic
Michal.Jedlicka@vut.cz
[2] FEM consulting Ltd., 602 00 Brno, Veveří 95, Czech Republic
nemec@fem.cz
[3] Brno University in Technology, Faculty of Civil Engineering, Institute of Mathematics and Descriptive Geometry, 602 00 Brno, Veveří 95, Czech Republic
Jiri.Vala@vut.cz
[4] Software Engineering, Zemědělská 10, 613 00 Brno, Czech Republic

**Abstract:** The following possibilities of reduction of dimension in the computational analysis of strain and stresses transferred to the subsoil massive are available: i) coming from the effective subsoil model by Kolář & Němec (1989), based on the assumptions of the Pasternak's model (1954), where the pair of material parameters of a surface model is evaluated from the energy equivalence, ii) reducing a large sparse matrix of soil massive stiffness to a smaller one, using Schur's complement technique. In both cases i), ii) the steady-state analysis is decisive: inclusion of more complicated combination of loads can be performed without repeated computations.

**Keywords:** structure-soil interaction, computational modelling, finite element method

**MSC:** 74M15, 74S05, 74L10

## 1. Introduction

Evaluation of the soil-rock mass interaction represents a key element in geotechnical engineering, which deals with analysing geological conditions, soil composition, and the physical properties of the subsoil. This information constitutes essential input data for numerical modelling, allowing the simulation of soil-rock mass behaviour under various conditions. When modelling soil-rock masses, it is important to take various factors into account, such as rock types, soil composition, groundwater levels, and other geotechnical parameters. This information enables us to create a realistic representation of the subsoil, and its response to external influences, such as loading from building structures.

DOI: 10.21136/panm.2024.07

Numerical modelling of the subsoil is essential for the proper design of building structures, especially of their foundations. In the early years of numerical modelling, the subsoil and the structure were treated separately, with no mutual influence, due to the computational complexity and the division between two design teams (geotechnics and structural engineering). This approach worked for simple structures and subsoil conditions (the soil environment under the foundations). For more complex structures, such as dams, high-rise buildings, tunnels, or large underground constructions, a model of the subsoil in interaction with the superstructure is needed, including deformations, tilts and stability considerations, to support the adaptation of foundation design to specific conditions of the given area and the particular structure. However, such advanced modelling techniques require higher computational power and more detailed input data, which must be gathered through geological surveys. Therefore, an essential engineering requirement is to consider the complexity of numerical modelling for various classes of structures reliably and economically.

After this brief motivation (1st section), we shall demonstrate the possibility of modelling the soil-rock mass in interaction with the structure, starting with the physical and mathematical background, including some historical remarks (2nd section). Then the computational design and software implementation (3rd section) is presented, supplied by an illustrative example (4th section) and followed by the sketch of possible generalizations, related to the research priorities for the near future (5th section).

## 2. Physical and mathematical background

Numerous theories for the modelling of a structure together with its subsoil can be classified by their characterization of subsoil properties and their approach to structure-subsoil interaction. Unlike simple (semi-)analytical historical formulae, such theories can handle viscoelastic and / or viscoplastic behaviour including damage to both a structure and its subsoil due to the class of rather general constitutive models, as presented by [22] and [33] and implemented into the RFEM software package (developed in collaboration with FEM consulting Brno with Dlubal Software Tiefenbach). In this short paper, we shall pay attention to the effective subsoil incorporation into the design of structures. The (quasi-)static approach will be preferred for simplicity; for its modification required by dynamic calculations see [29], for the extensive review of traditional and advanced computational techniques cf. [12].

### 2.1. Classical theories

The classical analytical models can be derived from the Boussinesq's theory [4], which focuses on the behaviour of subsoil under a single isolated force. A homogeneous isotropic subsoil which is defined by two key parameters: Young's modulus of elasticity $E$ [Pa] and Poisson's ratio $\nu$ [–]. Consequently a full 3-dimensional model in the Cartesian coordinate system $(x, y, z)$ can be formulated, using displace-

ments (non-zero in general, related to the initial configuration) $u(x, y, z)$, $v(x, y, z)$, $w(x, y, z)$. Its later modification, well-known as the Westergaard's theory [36], focuses on the non-uniform distribution of pressure on the foundation surface, adding corrections for wider foundations; the zero-valued $u$ and $v$ are considered, for more details in the modern formulation see [8]. Another significant contribution is the Mindlin's theory [21] where a closed-form solution for the displacement field caused by horizontal and vertical forces acting at any point in an elastic half-space can be found.

Biot's theory [3] describes the interactions in a porous elastic medium filled with an incompressible fluid; so-called Terzaghi-Wegmann's model [32] can be seen as an application of this theory in engineering practice, namely for the analysis of the influence of foundation geometry on subsoil stresses. Skempton's model [31] is frequently employed for analyzing subsoil deformations induced by shrinkage. Seed-Idriss's theory [30] contributes to the analysis of the behaviour of a cohesive subsoil under dynamic loading, namely in seismically active areas. Vesić's theory [35] describes the behaviour of soft subsoil under a foundation: the pore pressure within the subsoil is regarded as the combined effect of foundation-induced stress and hydrostatic pressure. Janbu-Meyerhof's approach [14] addresses the analysis of slope stability, accounting for the influence of the foundation on a plastic subsoil.

### 2.2. Winkler's and Pasternak's models

Simple (but still frequently used) Winkler's subsoil model [37] needs only one parameter $C_1$ [N/m$^3$], the vertical modulus of compressibility (coefficient of support). Since the displacements $u$ and $v$ are supposed to be negligible in comparison to $w$, we can take $w(x, y, z) = \widetilde{w}(x, y)\psi(z)$, $\psi$ is a prescribed function. The stress $p$ under a foundation structure (and also the subsoil reaction) can be expressed as $p(x, y) = C_1\widetilde{w}(x, y)$. The disadvantage of this model is the omission of shear stresses, which can lead to a sudden change in deformation immediately at the edge of the foundation structure where the deformation is zero, see Fig. 1, part A).

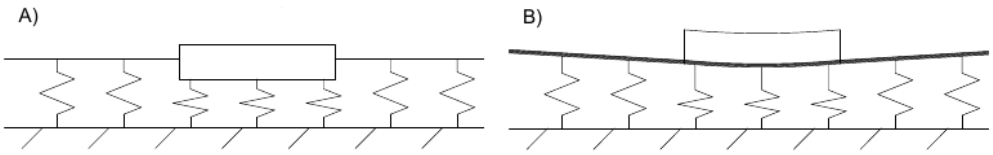



Figure 1: Subsoil models: A) 1 parameter by Winkler, B) 2 parameters by Pasternak.

Later studies try to suppress such a disadvantage, cf. [11]. In Pasternak's model [24], Winkler's model is extended by the parameter $C_2$ [N/m], which takes the effect of both normal and shear stresses into account, i. e. (under the assumption of isotropic behaviour of a subsoil, for simplicity here) $p(x, y) = C_1\widetilde{w}(x, y) - C_2\Delta\widetilde{w}(x, y)$, utilizing the Laplace operator $\Delta = \partial^2/\partial x^2 + \partial^2/\partial y^2$; cf. Fig. 1, part B). A more general class of foundation models of this type, involving additional parameters typically is introduced in [16]; for their detailed classification and numerous historical remarks cf. [19], [10] and [34].

### 2.3. Full 3-dimensional models

The 3-dimensional modelling requires the computational analysis of boundary value problems for partial differential equations, for a building structure and its subsoil separately, rarely reducible to those well-known from linear elasticity, together with a detailed analysis of all corresponding interfaces, thus rare (semi-)analytical solutions are available and an appropriate numerical approach is needed, based on the finite element techniques by [2] typically, as [22], [23] and [15]; another approach [1] relies on the boundary element method and certain integrals transforms.

Such approaches enable us to perform various specialized studies, such as stability analysis, permeability analysis of the subsoil, slope stability analysis, and seismic behaviour and response analysis of the subsoil. The proper analysis of complex shapes of subsoil layers and intricate structures, as well as their mutual interactions, can be seen as the principal advantage of this approach. Nevertheless, its evident disadvantages must be mentioned, too: at least i) rather high hardware and software requirements, ii) a tricky choice of the appropriate size of subsoil area, iii strong dependency of the reliability of all results on the correct choice of parameters and model validation, which must be provided by the user everywhere for ii), cf. [20]. In particular, in [17] the subsoil model is extended to a distance of approximately 115 m from the structure.

From the point of view of ii), the linear elastic model can be appreciated as simple and effective, allowing easy simulation of soil deformations under low stresses. Beyond this simplification, higher stresses lead to irreversible plastic (or viscous, etc.) deformations, as evaluable from historical Mohr-Coulomb's model [6], upgraded by [25], from later Drucker-Prager's [9] or Hoek-Brown's [13] ones, or from those developed especially from the soil analysis, referenced as Cam-Clay [26], working with a relation between stress, strain, and porosity, and Hardening Soil [27] for cyclic loading.

## 3. Computational approach

The subsoil can be modelled using the computational approaches mentioned in the 2nd section. The implementation into RFEM software makes it possible to perform a wide range of mechanical analyses, simplifying the analysis of the subsoil and its interaction with the structure. This is especially important in such tasks where only some specific loading cases influence the subsoil significantly.

### 3.1. Stress in the subsoil

In the analysis of stress within the subsoil, various types of stress arising from loading and site conditions can play a crucial role. The vertical (normal) stress is determined using the following formula $\sigma_z = \gamma h$ where $h$ is the depth of the layer in the soil and $\gamma$ is the unit weight of the soil. The horizontal (lateral) stress can be expressed as $\sigma_x = K_b \sigma_z$ where the lateral dimensionless pressure coefficient $K_b$ depends on the type of soil: i) for cohesive soils it is considered as $K_b = \nu/(1-\nu)$

(cf. Subsection 2.1), whereas ii) for granular (cohesion-free) soils the relation $K_b = 1 - \sin\phi$ is used, $\phi$ is the angle of internal friction. In the case that the groundwater level occurs, the total stress is usually expressed as $\sigma_{\text{tot}} = \sigma_{\text{eff}} + u$ where the pore pressure $u = \gamma_u \text{h}$ depends on one additional constant $\gamma_u$ and the effective stress $\sigma_{\text{eff}} = \gamma_{\text{su}} h$, $\gamma_{\text{su}}$ is the unit weight of dry soil.

Namely the stress at a depth $z$ in an elastic, homogeneous, and isotropic soil caused by a single point load $P$ with $\nu = 0$ can be determined by Boussinesq as

$$\sigma_z = \frac{3P}{2\pi z^2 \left(1 + (r/z)^2\right)^{5/2}} .$$

By Westergaard, infinitely thin soil layers are assumed, together with $0 \leq \nu < 1$, which results

$$\sigma_z = \frac{P(1-2\nu)(2-2\nu)}{2\pi z^2 \left((1-2\nu)/(2-\nu) + (r/z)^2\right)^{3/2}} .$$

The depth of the deformation zone is defined according to the technical standards CSN EN 1997-1 (731000) and Eurocode 7: Design of Geotechnical Structures – Part 1: General Rules, obligatory in the Czech Republic. These methods determine the depth below the foundation where the substantial increase in vertical stress occurs. The first method is the primary stress limitation method, which is expressed by the formula $\sigma_z = p\sigma_{\text{or}}$ where $\sigma_{\text{or}}$ represents the original geostatic stress and $p$ is its considered percentage. The second method refers to the structural strength theory with the (formally similar) result $\sigma_z = m\sigma_{\text{or}}$, $m$ is the structural strength coefficient. The stress distributions and the deformation depths for both methods are illustrated by Fig. 2.

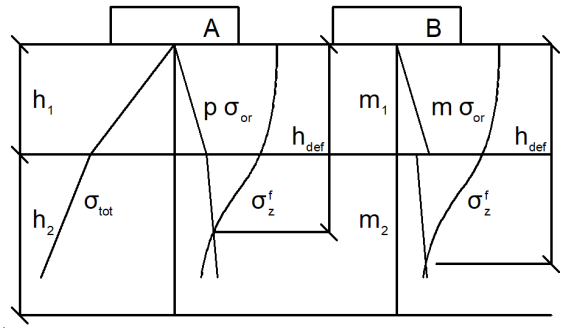



Figure 2: Evaluation of the deformation depth.

### 3.2. Effective subsoil approach

The approach working with dimension reduction comes from Pasternak's model, allows us to develop a relatively accurate model of the structure-subsoil interaction using only the foundation plane and its boundaries. However, an iterative evaluation of the deformation depth (zone) is necessary. All needed relations between the

parameters of the surface and the spacial model for individual layers are compatible with [19]. We receive (as introduced in Subsection 2.2)

$$C_1 = \int_0^h E_z \left(\frac{\partial f(z)}{\partial z}\right)^2 \mathrm{d}z, \qquad C_{2x} = \int_0^h G_{xz} f^2(z)\,\mathrm{d}z, \qquad C_{2y} = \int_0^h G_{yz} f^2(z)\,\mathrm{d}z$$

for certain function $f(z)$; $E_z$ is the deformation modulus, $G_{xz}$ and $G_{yz}$ are the shear moduli in the respective axes. In a non-isotropic environment, the parameters $C_{2x}$ and $C_{2y}$ can be different; in this presentation, we shall assume a homogeneous and isotropic environment for simplicity, which allows us to work with one constant value $C_2$, similarly to the case of $C_1$.

On the boundary line, the spring constants $k_w$ and $k_\varphi$ for the displacement $w$ and the rotation $\varphi$ can be derived in the form $k_w = \sqrt{C_1 C_2}$, $k_\varphi = \frac{1}{2} C_2 \sqrt{C_2/C_1}$. For corner nodes or changes in the curvature of the line (e. g. on a polygonal boundary) a nodal spring constant $\mathcal{K} = \frac{1}{2} C_2 \alpha \mathfrak{K}(\alpha)$ must be added where $\alpha$ denotes the angle measured between normals of adjacent curves and $\mathfrak{K}(\alpha)$ is a certain additional function taking further geometric properties into account. In particular, for $\alpha = \pi/2$ the spring constant $\mathcal{K} \approx \frac{1}{2} C_2$ can be considered; for the justification see [18], p. 60. For multiple subsoil layers, it is possible to calculate $C_1$ and $C_2$ from $n$ parameters $C_{1i}$ with $i \in \{1, \ldots, n\}$ in the form

$$C_{1i} = \frac{E_i\,(1-\nu_i)}{h_i\,(1+\nu_i)\,(1-2\nu_i)}, \qquad C_1 = 1/\sum_{i=1}^{n} (1/C_{1i})\,,$$

$$C_2 = \frac{1}{6} C_1^2 \sum_{i=1}^{n} \left( \frac{E_i h_i}{1+\nu_i} \left[ \left( \sum_{j=i}^{n} \frac{1}{C_{1j}} \right)^2 + \left( \sum_{j=i}^{n} \frac{1}{C_{1j}} \right) \left( \sum_{j=i+1}^{n} \frac{1}{C_{1j}} \right) + \left( \sum_{j=i+1}^{n} \frac{1}{C_{1j}} \right)^2 \right] \right).$$

### 3.3. Stiffness matrix reduction

As discussed in Subsection 2.3, the finite element (or similar) techniques are needed for the full 3-dimensional modelling, with the result of the solution of large systems of linear algebraic equations (frequently iterative, handling various nonlinearities, as mentioned at the beginning of the 2nd section), with sparse or banded stiffness matrices as system ones. From the physical point of view, in certain cases where the effects and behaviour of the soil mass have been precisely calculated for the critical loading conditions, these stiffness values can be used for subsequent states that are less significant for the behaviour of the soil mass. This complexity can be reduced by using Schur's complements by [28] and [7]; for their effective applications in numerical analysis see [5]. Schur's complement technique involves partitioning the large square stiffness matrix into particular blocks. Namely the stiffness matrix

$$K = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

contains 4 matrices $A, B, C, D$: $A$ corresponds to internal degrees of freedom (DOFs), $D$ to boundary DOFs, $B$, $C$ are cross-reference terms connecting internal and boundary DOFs. The reduction of a matrix $K$ can be then introduced (if $D$ is invertible) as $K/D = A - BD^{-1}C$, $K/A = C - AD^{-1}B$ which implies

$$K^{-1} = \begin{bmatrix} (K/D)^{-1} & -(K/D)^{-1}BD^{-1} \\ -D^{-1}C(K/D)^{-1} & D^{-1} + D^{-1}C(K/D)^{-1}BD^{-1} \end{bmatrix}.$$

Some well-known properties of Schur's complements help to simplify our practical calculations, namely $\mathrm{rank}(K) = \mathrm{rank}(D) + \mathrm{rank}(K/D)$ (rank additivity formula), $A/B = (A/C)/(B/C)$ (quotient identity), etc. Even in the case that $A$ or $D$ is singular, the generalized inverse (pseudoinverse) instead of the standard one on $K/A$ and $K/D$ yields generalized Schur's complement.

## 4. Illustrative example

A simple example was prepared to verify the implemented formulas for the parameters $C_1, C_2$. The problem involves a plate of size $10 \times 10 \times 0.3$ m, subjected to the uniform perpendicular load 40 kPa. The material characteristics are $E = 25$ MPa, $\nu = 0.28$, $\gamma = 17$ kPa, the total height is $h = 8$ m.

Three variants of computational modelling cover: i) one single layer with $h = 8$ m, ii) three layers with $h_1 = 5$ m, $h_2 = 2$ m, $h_3 = 1$ m, iii) stress-based calculations (cf. Subsection 3.2). The parameters $C_1$ and $C_2$ based on the effective subsoil model were identical in variants i) and ii), with $C_1 = 2.604167$ MPa/m and $C_2 = 3.995028$ MN/m. For the approach iii) using the deformation zone, all results were computed for each finite element at the centroid and the stress distribution below this point was obtained. It was found that the formulae for calculating the stress $\sigma_z$ are not quite suitable because they approach infinity near the surface (ground level). Therefore it is better to use modified formulas for distributed loads where this effect is eliminated. Implementing such additional formulas shortly is feasible; it requires adjusting the calculation of $\sigma_z$ to obtain a more accurate distribution only, as shown at Fig. 3 (with non-constant values of $C_1$) and Fig. 4.

For the variant of reducing the stiffness matrix using Schur's complements, the results are still too large to be displayed for this example. The original stiffness matrix had 7 497 nodes with 6 degrees of freedom with a total number of columns of 44 982, and the total number of non-zero elements was 3 304 413, indicating that a significantly large and sparse system is being solved. For the reduced system, from the original 7 497 nodes to a surface with 441 nodes, the number of columns decreased to 2 646, and the total number of non-zero elements was 1 750 329. We can see that the total size of the matrix decreased 17 times, but the number of non-zero elements decreased approximately 1.9 times. This is therefore advantageous for us, but it is necessary to continue the development and implementation, as well as the use of this reduction in subsequent analyses.

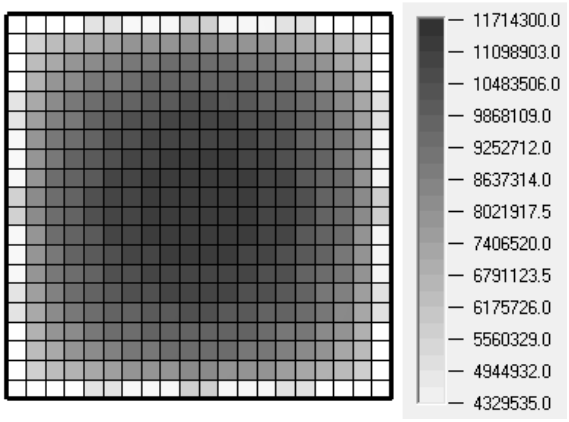



Figure 3: Distribution of the parameter $C_1$ for variant iii).

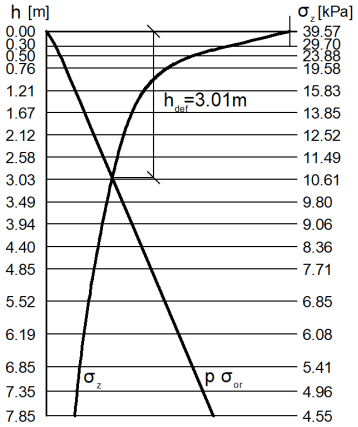



Figure 4: Stress distribution under the finite element.

## 5. Conclusions

This paper aims to demonstrate the possibility of modelling of the soil-rock mass and its reduction. For surface models, the implementation of an effective subsoil model was presented, including the calculation of parameters $C_1$ and $C_2$ for multiple layers of subsoil. The stress calculation $\sigma_z$ in the subsoil and the determination of the deformation depth were also included. This solution is already fully implemented and can be used in RFEM software in collaboration with FEM consulting.

For advanced modelling using 3D objects, a reduction method via stiffness matrix condensation was proposed. Schur's complement technique was applied to reduce the stiffness matrix, resulting in a smaller system of equations. This solution is not yet fully implemented in the program and cannot be used routinely due to the fact that it is not a fully general solution and can only be applied with the correct setup of the calculated analyses on the computational core side. Therefore, the software user cannot control this part, this can be seen as a major challenge for us in generalizing this approach and releasing it in the relevant software as soon as possible. This reduction is especially significant for subsequent calculations where it is no longer necessary to analyze the soil-rock mass in detail, but rather the superstructure, for example in dynamic problems.

Based on the above findings, our plans focus on further development and improvement of methods for modelling the soil-rock massif and its reduction. Our priority is to complete the implementation into the RFEM software and enable the creation of more test cases more easily. Above all, to determine the appropriateness of using individual variants and identify their limitations. This will require a large number of test and benchmark examples to validate and compare these approaches. In addition, the reduction of the stiffness matrix using Schur's complement can be used in other analyses, not only within the soil-rock massif, which increases the im-

portance of this implementation. So our drive remains unchanged to create the most sophisticated tools possible to support structural engineers in solving increasingly complex problems, both in terms of computational speed and accuracy.

**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] Aji, H. D. B., Wuttke, F., and Dineva, P.: 3D structure-soil-structure interaction in an arbitrary layered half-space. Soil Dyn. Earthquake Eng. **159** (2022) 107352/1–22.

[2] Bathe, K.-J.: *Finite Element Procedures*. Prentice Hall, Hoboken, 2006.

[3] Biot, M. A.: Theory of elastic waves in a fluid-saturated porous solid. J. Acoust. Soc. Am. **28** (1956), 168–191.

[4] Boussinesq, J.: *Application des potentiels à l'étude de l'équilibre et du mouvement des solides élastiques*. Gauthier-Villars, Paris, 1885. (In French.)

[5] Brezinski, C.: Schur complements and applications in numerical analysis. In: Zhang, F. (Ed.), *The Schur Complement and Its Applications*, Chap. 4. Springer, Boston, 2005.

[6] Coulomb, C. A.: Essai sur une application des règles de maximis & minimis à quelques problèmes de statique: relatifs à l'architecture. Memoires de Mathematique de l'Academie Royale de Science **7** (1776), 343–387. (In French.)

[7] Crabtree, D. and Haynsworth, E. V.: The Schur complement and its applications. Proc. Am. Math. Soc. **22** (1969), 364–366.

[8] Das, B. M.: *Principles of Foundation Engineering*. Cengage Learning, London, 2010.

[9] Drucker, D. C. and Prager, W.: Soil mechanics and plastic analysis or limit design. Q. Appl. Math. **10** (1952), 157–165.

[10] Dutta, S. Ch. and Roy, R.: A critical review on idealization and modeling for interaction among soil–foundation–structure system. Comput. Struct. **80** (2002), 1579–1594.

[11] Filonenko-Borodich, M. M.: Some approximate theories of elastic foundations. Uchenye zapiski Moskovskogo gosudarstvennogo universiteta: Mekhanika **46** (1940), 3–18.

[12] Gallese, D., Gorini, D. N., and Callisto, L.: Modelling nonlinear static analysis for soil-structure interaction problems. F. Di Trapani, C. Demartino, G. C. Marano, and G. Monti (Eds.), *Proc. 2022 Eurasian OpenSees Days*, pp. 377–387. Lecture Notes in Civil Engineering 326, Springer, Cham, 2023.

[13] Hoek, E. and Brown, E. T.: Empirical strength criterion for rock masses. J. Geotech. Eng. Div. **106** (1980), 1013–1035.

[14] Janbu, N.: Slope stability computation. R. C. Hirschfeld and S. J. Polulos (Eds.), *Embankment-Dam Engineering*, *Casagrande Volume*. Krieger, London, 1987.

[15] Kant, L. and Samanta, A.: Nonlinear analysis of building structures resting on soft soil considering soil–structure interaction. J. Inst. Eng. India A **105** (2024), 475–485.

[16] Kerr, A. D.: Elastic and viscoelastic foundation models. J. Appl. Mech. **31** (1964), 491–498.

[17] Kelezi, L.: Local transmitting boundaries for transient elastic analysis. Soil Dynamics and Earthquake Engineering **19** (2000), 533–547.

[18] Kolář, V. and Němec, I.: *Studie nového modelu podloží staveb*. Academia, Prague, 1986. (In Czech.)

[19] Kolář, V. and Němec, I.: *Modelling of Soil-Structure Interaction*. Elsevier, Amsterdam, 1989.

[20] Labudková, J. and Čajka, R.: Experimental measurements of subsoil-structure interaction and 3D numerical models. Perspect. Sci. **7** (2016), 240–246.

[21] Mindlin, R. D.: Influence of rotatory inertia and shear on flexural motions of isotropic, elastic plates. ASME J. Appl. Mech. **18** (1951), 31–38.

[22] Němec, I., Trcala, M., and Rek, V.: *Nelineární mechanika*. VUTIUM, Brno, 2018. (In Czech.)

[23] Nepelski, K.: 3D FEM analysis of the subsoil-building interaction. Appl. Sci. **12** (2022), 10700/ 1–25.

[24] Pasternak, P. L.: *Osnovy novogo metoda rascheta fundamentov na uprugom osnovanii pri pomoshchi dvukh koeffitsientov posteli*. Gosstroĭizdat, Moscow, 1954. (In Russian.)

[25] Powrie, W.: *Soil Mechanics: Concepts and Applications*. CRC Press, Boca Raton, 1996.

[26] Roscoe, K. H., Schofield, A. N., and Wroth, C. P.: On the yielding of soils. Géotechnique **8**, (1958), 22–53.

[27] Schanz, T., Vermeer, P. A., and Bonnier, P. G.: The hardening soil model: formulation and verification. R. B. J. Brinkgreve (Ed.), *Beyond 2000 in Computational Geotechnics*, *Part Education and Research, Chap. 4*. Routledge, London, 1999.

[28] Schur, I.: Über Potenzreihen, die im Innern des Einheitskreises beschränkt sind. Journal für die reine und angewandte Mathematik **147** (1917), 205–232. (In German.)

[29] Shehata, O. E., Faris, A. F., and Rashed, Y. F.: A suggested dynamic soil – structure interaction analysis. J. Eng. Appl. Sci. **70** (2023), 63/ 1–16.

[30] Seed, H. B. and Idriss, I. M.: Simplified procedure for evaluating soil liquefaction potential. J. Soil Mech. Found. Div. **97** (1971), 1249–1273.

[31] Skempton, A. W.: The pore-pressure coefficients $A$ and $B$. Géotechnique **4** (1954), 143–147.

[32] Terzaghi, K.: *Theoretical Soil Mechanics*. J. Wiley & Sons, New York, 1943.

[33] Trcala, M., Suchomelová, P., Bošanský, M., Hokeš, F., and Němec, I.: The generalized Kelvin chain-based model for an orthotropic viscoelastic material. Mechanics of Time-Dependent Materials **28** (2024), 1639–1659.

[34] Vala, J., Němec, I., and Vaněčková, A.: Exact solution of a thick beam on Pasternak subsoil in finite element calculations. Math. Comput. Simul. **189** (2021), 36–54.

[35] Vesić, A. S.: Expansion of cavities in infinite soil mass. J. Soil Mech. Found. Div. **98** (1972), 265–290.

[36] Westergaard, H. M.: Bearing pressures and cracks: bearing pressures through a slightly waved surface or through a nearly flat part of a cylinder, and related problems of cracks. J. Appl. Mech. **6** (1939), A49–A53.

[37] Winkler, E.: *Die Lehre von der Elasticitaet und Festigkeit*. H. Dominicus, Prague, 1867. (In German.)

*Programs and Algorithms of Numerical Mathematics 22*
J. Chleboun, J. Papež, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics CAS, Prague 2025

# OPTIMAL ERROR ESTIMATES FOR FINITE ELEMENTS ON MESHES CONTAINING BANDS OF CAPS

Václav Kučera, Jiří Szotkowski

Faculty of Mathematics and Physics, Charles University
Sokolovská 83, Praha 8, 186 75, Czech Republic
kucera@karlin.mff.cuni.cz, szotkowski.jiri@email.cz

**Abstract:** In this short note we provide an optimal analysis of finite element convergence on meshes containing a so-called band of caps. These structures consist of a zig-zag arrangement of 'degenerating' triangles which violate the maximum angle condition. A necessary condition on the geometry of such a structure for various $H^1$-convergence rates was previously given by Kučera. Here we prove that the condition is also sufficient, providing an optimal analysis of this special case of meshes. In the special case of optimal $O(h)$-convergence of finite elements, the analysis states that such optimal convergence is possible if and only if the height of the band of caps is at least $Ch^2$ for some constant $C$. Numerical experiments confirm this result.

**Keywords:** Finite element method, error estimates, maximum angle condition

**MSC:** 65N30, 65N15, 65N50

## 1. Introduction

The finite element method is the golden standard of current methods for partial differential equations. Much work has been devoted over the past 60 years to develop various error estimates for this method applied a wide range of problems. It may therefore seem surprising that the simplest basic question remains unanswered to this day: What is a necessary and sufficient condition on triangular meshes for piecewise linear finite elements to converge? Even in the simplest of all settings – Poisson's problem and estimates in the corresponding $H^1(\Omega)$ energy norm, this is still an open problem.

The basic textbook result is that if the meshes satisfy the *minimum angle condition*, then finite elements will exhibit optimal $O(h)$ convergence in the energy norm. This condition requires that all angles of all elements in the mesh(es) are uniformly bounded away from zero. A slightly more advanced result is that $O(h)$ convergence occurs under the more general *maximum angle condition*, which requires that the

DOI: 10.21136/panm.2024.08

maximal angles of all triangles are uniformly bounded away from $\pi$. This sufficient condition was generally assumed to also be necessary – the confusion was caused by the misleading title "The maximum angle condition is essential" from the original paper [1]. The title refers to a counterexample provided in the paper, where finite elements do not converge on a special mesh consisting only of 'degenerating' elements. As it turns out, the maximum angle condition is not necessary for $O(h)$ convergence of the finite element method, cf. [3]. Since then, another counterexample was analyzed in the paper [6], where a single structure, a so-called band of caps, contained in the mesh destroys finite element convergence. The analysis leads to conditions on the proportions and geometry of the band of caps that is necessary for $O(h)$ convergence, and more generally $O(h^\alpha)$ convergence for some $\alpha \in [0, 1]$.

The purpose of this short note is to show that the condition on the band of caps derived in [6] is optimal, i.e. both necessary and sufficient for $O(h^\alpha)$ convergence. Although the question of a general necessary and sufficient condition for the convergence of the finite element method still remains open, at least there is a second special case that can be analyzed optimally. The main result of the analysis is that $O(h)$ convergence of finite elements occurs if and only if the height of the band of caps is at least $Ch^2$ for some constant $C$. This is important, as a band of caps is a natural triangulation of a (straight) interface. In 2D, an interface is a 1D object, and it is natural to approximate it using very flat triangles in a mesh. The theorem states that the triangles approximating the interface can be flatter and flatter as we refine the mesh, as long as their height is at least $Ch^2$. We present numerical experiments that confirm this result, and also indicate that a height of at least $Ch^2$ is also necessary and sufficient for $O(h^2)$-convergence in the $L^2$-norm, a result that we are unable to prove rigorously.

## 2. Finite element method

As a model problem, we will be focused on Poisson's problem in $\mathbb{R}^2$. Let $\Omega \subset \mathbb{R}^2$ be a polygonal domain with Lipschitz boundary $\partial\Omega$. We solve the problem

$$-\Delta u = f \text{ on } \Omega, \quad u|_{\partial\Omega} = 0 \tag{1}$$

with the weak form: Find $u \in H_0^1(\Omega)$ such that

$$\int_\Omega \nabla u \cdot \nabla v \, \mathrm{d}x = (f, v), \quad \forall v \in H_0^1(\Omega), \tag{2}$$

where $H_0^1(\Omega)$ is the standard Sobolev space of functions with square integrable derivatives and a zero trace on $\partial\Omega$, while $(f, v) = \int_\Omega fv \, \mathrm{d}x$ is the $L^2$ scalar product.

In the finite element method, we consider a conforming triangulation $\mathcal{T}_h$ of $\Omega$, i.e. a partition into triangles (elements) with mutually disjoint interiors such that the intersection of two neighboring elements is either a single vertex or a whole edge.

Here $h$ denotes the length of the longest edge in the triangulation. This partition defines the continuous piecewise linear finite element space

$$V_h = \{v_h \in C(\overline{\Omega}); v_h|_K \in P^1(K) \text{ for all } K \in \mathcal{T}_h\}, \tag{3}$$

where $P^1(K)$ is the space of linear functions on the triangular element $K \in \mathcal{T}_h$.

The finite element method is then defined as follows: Find $u_h \in V_h$ such that

$$\int_\Omega \nabla u_h \cdot \nabla v_h \, \mathrm{d}x = (f, v_h), \quad \forall v_h \in V_h. \tag{4}$$

It is desirable to obtain estimates for the error $u - u_h$. To this end, Céa's lemma, cf. [2], gives us an estimate in the $H^1(\Omega)$-seminorm:

$$|u - u_h|_{H^1(\Omega)} = \inf_{v_h \in V_h} |u - v_h|_{H^1(\Omega)}, \tag{5}$$

where $|u|_{H^1(\Omega)} = \sqrt{\int_\Omega |\nabla u|^2 \mathrm{d}x}$. We note that for other problems, one can expect an inequality in (5) and a problem-dependent constant in the upper bound.

Standard finite element estimates are typically derived by taking the piecewise linear Lagrange interpolation $\Pi_h u$ as $v_h$ in (5). This is defined element-wise: on each element $K \in \mathcal{T}_h$ the function $\Pi_h u|_K = \Pi_K u \in P^1(K)$ coincides with $u$ at the vertices of $K$. Such a locally defined function naturally gives a globally continuous piecewise linear function in $V_h$.

For triangles, there is an *optimal* estimate for the interpolation error $u - \Pi_K u$ in seminorms in the general Sobolev space $W^{1,p}(\Omega)$. We will need this estimate only in the special case of $p = \infty$. Consider an arbitrary triangle $K \subset \mathbb{R}^2$. Denote the length of its longest edge as $h_K$ and its height perpendicular to this edge as $\overline{h}_K$. Finally, define $R_K$ as the circumradius of $K$, i.e. the radius of the circumscribed circle to $K$. We have the following optimal estimate, cf. [4], [5].

**Lemma 1** (Circumradius estimate)**.** *Let $K \subset \mathbb{R}^2$ be an arbitrary triangle. Let $u \in W^{2,p}(K)$, $1 \le p \le \infty$, and let $\Pi_K u$ be the linear Lagrange interpolation of $u$ on $K$. Then there exists a constant $C_c$ independent of $u$ and $K$ such that*

$$|u - \Pi_K u|_{W^{1,p}(K)} \le C_c R_K |u|_{W^{2,p}(K)} \le C_c \frac{h_K^2}{\overline{h}_K} |u|_{W^{2,p}(K)}. \tag{6}$$

One is especially interested in optimal convergence results of the order $O(h)$ in the $H^1(\Omega)$-seminorm, via (5). A sufficient (but not necessary!) condition for this to happen is when $R_K \le \widetilde{C} h$ for all $K \in \mathcal{T}_h$ with some constant $\widetilde{C}$ independent of $h$. Geometrically, this is equivalent to satisfying the *maximum angle condition*. This condition requires that all maximal angles $\alpha_K$ of all triangles $K \in \mathcal{T}_h$ are smaller than some $\alpha_0 < \pi$. Then we have the following element-wise estimate, which can then be applied in (5).

**Lemma 2** (Maximum-angle condition)**.** *Let $K \subset \mathbb{R}^2$ be a triangle satisfying the maximum angle condition: $\alpha_K \leq \alpha_0 < \pi$ for some fixed $\alpha_0$. Let $u \in H^2(K)$ and let $\Pi_K u$ be the linear Lagrange interpolation of $u$ on $K$. Then there exists a constant $C_I$ depending only on $\alpha_0$ such that*

$$|u - \Pi_K u|_{H^1(K)} \leq C_I h |u|_{H^2(K)}. \tag{7}$$

By taking the piecewise linear element-wise Lagrange interpolation in Céa's lemma (5) one immediately obtains the following error estimate from Lemma (2).

**Theorem 3** (Basic error estimate)**.** *Let $u \in H^2(\Omega)$ be the solution of (2) and $u_h \in V_h$ the finite element solution of (4). If $\alpha_K \leq \alpha_0 < \pi$ for all $K \in \mathcal{T}_h$, we have*

$$|u - u_h|_{H^1(\Omega)} \leq C_I h |u|_{H^2(\Omega)}, \tag{8}$$

*where $C_I$ is the constant from Lemma 2.*

The maximum angle condition has a long and complicated history, being discovered independently by several groups, e.g. [1]. In [3] it was proven that this condition is not necessary for $O(h)$ convergence. In fact $\mathcal{T}_h$ can contain many 'bad' triangles violating the maximum angle condition while still exhibiting optimal $O(h)$ convergence. In other words, the finite element method can converge optimally even when the Lagrange interpolation error goes to infinity. This is especially important when we have a sequence of meshes obtained e.g. by refinement and let $h \to 0$. In this situation one usually considers a set of triangulations $\mathcal{T}_h$, $h \in (0, h_0)$ for some $h_0 > 0$.

Apart from the paper [3], paper [6] has dealt with sufficient as well as necessary conditions for $O(h)$ convergence, or more generally, for $O(h^\alpha)$ estimates with $0 \leq \alpha \leq 1$. Specifically, the so-called *band of caps* has been identified as the basic (but not only) villain preventing optimal convergence of the finite element method. The band of caps consists of triangles in a zigzag pattern, cf. Figure 1, where all of the elements violate the maximum angle condition with the given $\alpha_0$. Specifically, we shall consider such a band of length $L$ and height $\overline{h}$ consisting of identical isosceles triangles with diameters $h$, cf. Figure 1. We assume that every $\mathcal{T}_h$ we consider contains one such band, while all other elements satisfy the maximum angle condition with a fixed maximal angle $\alpha_0$. It is important to note that the length $L$ of the band can also depend on $h$ (e.g. $L \sim \sqrt{h}$, etc.), although the most important case in our situation is that $L \sim 1$ is independent of $h$.


Figure 1: Band of caps of length $L$ and height $\overline{h}$.

The band of caps is important as a model for an approximated interface within the mesh $\mathcal{T}_h$. This is because it is an essentially 1D object (as an interface in 2D

would be) with some nonzero thickness $\overline{h}$. It is then desirable to have the thickness of the approximate interface as small as possible without affecting the convergence rate of the finite element method. Due to the regular structure, the finite element error can be analyzed on meshes containing these bands of caps. Specifically, what are conditions on the geometry parameters $L$ and $\overline{h}$ in order to preserve $O(h)$ convergence, or more generally $O(h^\alpha)$ convergence for some $\alpha \in [0,1]$. In [6], the following result is proved as a special case of the main theorem of the paper dealing with a band of general elements (cf. estimate (64) in the cited paper).

**Theorem 4.** *Let $u \in W^{2,\infty}(\Omega)$ and let $\alpha \in [0,1]$. Let $\mathcal{T}_h, h \in (0, h_0]$ each contain a band of caps $\mathcal{B}$ of length $L$ and height $\overline{h}$. Let $L \geq C_L h^{2\alpha/5}$, where $C_L$ is a sufficiently large constant. Then a necessary condition for the estimate*

$$|u - u_h|_{H^1(\Omega)} \leq \widehat{C} h^\alpha \tag{9}$$

*to hold with some $\widehat{C}$ independent of $h$, is*

$$\overline{h} \geq \widetilde{C} h^{4-2\alpha} L \tag{10}$$

*for some $\widetilde{C} > 0$.*

In the special case of a band of caps of length $L \sim 1$, the condition says that for $O(h)$ convergence of the finite element method, we must necessarily have $\overline{h} \geq \widetilde{C} h^2$ for some $\widetilde{C} > 0$. And for (even arbitrarily slow) convergence of the finite element method, i.e. the limiting case of $\alpha = 0$, we must necessarily have $\overline{h} \geq \widetilde{C} h^4$ for some $\widetilde{C} > 0$. In the next section, we will show that these conditions are both necessary and sufficient.

## 3. Optimal error estimate for a band of caps

Here we will show that the condition (10) on $\overline{h}$ from Theorem 4 is not only necessary for $O(h^\alpha)$-convergence, but also sufficient. It turns out that unlike the lengthy technical proof of Theorem 4, this is a simple application of the circumradius estimate. In the following, $C$ will be a generic constant independent of $u$ and $h$.

**Theorem 5.** *Let $u \in W^{2,\infty}(\Omega)$ and let $\alpha \in [0,1]$. Let $\mathcal{T}_h$ contain a band of caps $\mathcal{B}$ of length $L$ and height $\overline{h}$, while all other elements in $\mathcal{T}_h$ satisfy the maximum angle condition with some $\alpha_0$. Let there exist $\widetilde{C} > 0$ such that*

$$\overline{h} \geq \widetilde{C} h^{4-2\alpha} L. \tag{11}$$

*Then there exists a constant $C$ independent of $u$ and $h$, such that*

$$|u - u_h|_{H^1(\Omega)} \leq C h^\alpha |u|_{W^{2,\infty}(\Omega)}. \tag{12}$$

*Proof.* From Céa's lemma we have

$$|u-u_h|^2_{H^1(\Omega)} \leq |u-\Pi_h u|^2_{H^1(\Omega)} = |u-\Pi_h u|^2_{H^1(\Omega\setminus\mathcal{B})} + |u-\Pi_h u|^2_{H^1(\mathcal{B})}, \tag{13}$$

due to additivity of integrals. The first term in (13) uses standard estimates (all elements of $\Omega \setminus \mathcal{B}$ satisfy the maximum angle condition):

$$|u-\Pi_h u|^2_{H^1(\Omega\setminus\mathcal{B})} \leq Ch^2|u|^2_{H^2(\Omega)} \leq Ch^2|\Omega||u|^2_{W^{2,\infty}(\Omega)}. \tag{14}$$

The second term in (13) is estimated using the circumradius estimate (6):

$$\begin{aligned} |u-\Pi_h u|^2_{H^1(\mathcal{B})} &= \int_{\mathcal{B}} |\nabla u - \nabla\Pi_h u|^2 \,\mathrm{d}x \leq |u-\Pi_h u|^2_{W^{1,\infty}(\mathcal{B})}|\mathcal{B}| \\ &\leq C\left(\frac{h^2}{\overline{h}}\right)^2 |u|^2_{W^{2,\infty}(\mathcal{B})}|\mathcal{B}| \leq C\frac{h^4}{\overline{h}}L|u|^2_{W^{2,\infty}(\Omega)}, \end{aligned} \tag{15}$$

since $|\mathcal{B}| \leq \overline{h}L$. Using assumption (11) on $\overline{h}$ in the right-hand side of (15), we get

$$|u-\Pi_h u|^2_{H^1(\mathcal{B})} \leq C\frac{h^4}{\widetilde{C}h^{4-2\alpha}L}L|u|^2_{W^{2,\infty}(\Omega)} = Ch^{2\alpha}|u|^2_{W^{2,\infty}(\Omega)}. \tag{16}$$

Combining estimates (13), (14), and (16), and taking the square root gives us the desired estimate. □

If we are specifically interested in the most interesting case of $L \sim 1$ independent of $h$, and $\alpha = 1$ (i.e. $O(h)$-convergence), we get the following theorem. It states that the height $\overline{h}$ of the band of caps can in fact go to zero as fast as $h^2$ without influencing the $O(h)$ convergence rate of the finite element method. For the simulation of interfaces this is good news, since it allows for a finer resolution of the interface (which technically has zero height).

**Theorem 6.** *Let $u \in W^{2,\infty}(\Omega)$. Let $\mathcal{T}_h$ contain a band of caps $\mathcal{B}$ of length $L \sim 1$ and height $\overline{h}$, while all other elements in $\mathcal{T}_h$ satisfy the maximum angle condition with some $\alpha_0$. Let there exist $\widetilde{C} > 0$ such that*

$$\overline{h} \geq \widetilde{C}h^2. \tag{17}$$

*Then there exists a constant $C$ independent of $h$ and $u$, such that*

$$|u-u_h|_{H^1(\Omega)} \leq Ch|u|_{W^{2,\infty}(\Omega)}. \tag{18}$$

## 4. Numerical experiments

In this section we use numerical experiments to confirm that condition (17) is necessary and sufficient for $O(h)$-convergence. Namely we consider problem (2) on $\Omega = [-1,1]^2$ with the manufactured solution $u(x,y) = \cos(\pi x)\sin(\pi y)$. We consider

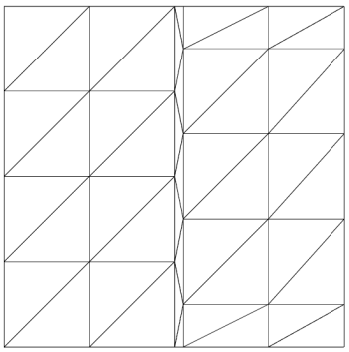


Figure 2: Test mesh containing a vertical band of caps.

meshes with a single vertical band of caps in the center of the domain which spans the whole height of $\Omega$, cf. Figure 2. The mesh outside of the band of caps is very regular, consisting of right-angled triangles. We construct a series of 11 such meshes with decreasing $h$. We also construct reference meshes which do not contain a band of caps but only identical right-angled triangles throughout the entire mesh.

To test the sharpness of condition (17), we consider meshes containing bands of caps with height $\overline{h} = h^k$ for $k = 2, 2.5$, and 3, where $k = 2$ corresponds to the case of $\overline{h} = h^2$ from Theorem 6. According to the theorem, higher exponents than $k = 2$ should result in slower than $O(h)$-convergence of the finite element method.

The $H^1$-errors are plotted in Figure 3. We observe that the convergence curves on the 'nice' reference meshes and on the meshes with a band of caps with height $\overline{h} = h^2$ are essentially indistinguishable. On the other hand, the curve corresponding to $k = 3$, i.e. $\overline{h} = h^3$, clearly exhibits a slower convergence rate as $h \to 0$. For $k = 2.5$ the curve also exhibits a decrease in convergence rate, although not as dramatic as $k = 3$. Testing exponents even closer to $k = 2$, e.g. $k = 2.1$ would require extremely fine meshes to observe the slow-down of convergence. Nevertheless, we view Figure 3 as a confirmation of Theorem 6.

Finally, we have also tested convergence in the $L^2(\Omega)$-norm. The results are in Figure 4. In the $L^2(\Omega)$-norm, we expect the error to be $O(h^2)$ under ideal circumstances (e.g. provided the maximum angle condition). This convergence rate can be seen in the convergence curve for the reference meshes. Although we are unable to prove this, we get this optimal convergence rate also in the presence of bands of caps of height $\overline{h} = h^2$ (i.e. the case satisfying Theorem 6). Again the two curves are essentially identical. And as for the $H^1$-seminorm, the convergence rate in the $L^2$-norm decreases for the higher exponents $k = 3$ and $k = 2.5$.

We note that we are currently unable to prove the optimal convergence rates in $L^2$ in the presence of bands of caps, since the proof of Theorems 6 and 5 is based essentially on $L^\infty(\Omega)$ estimates of the gradients and second derivatives. Proving these optimal convergence rates would require the application of the Aubin-Nitsche duality argument in $L^\infty$-based norms, which leads to technical issues that we were so-far unable to circumvent.

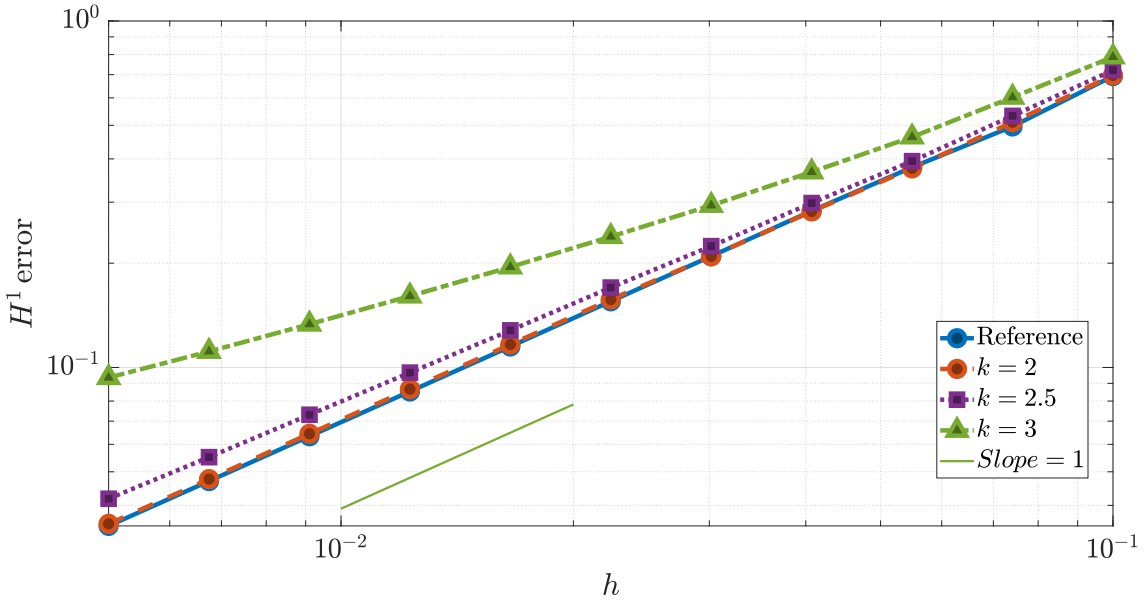



Figure 3: Convergence of the finite element method in the $H^1$-seminorm. Convergence on a regular reference mesh and convergence on meshes containing bands of caps with height $\overline{h} = h^k$ with $k = 2, 2.5, 3$.

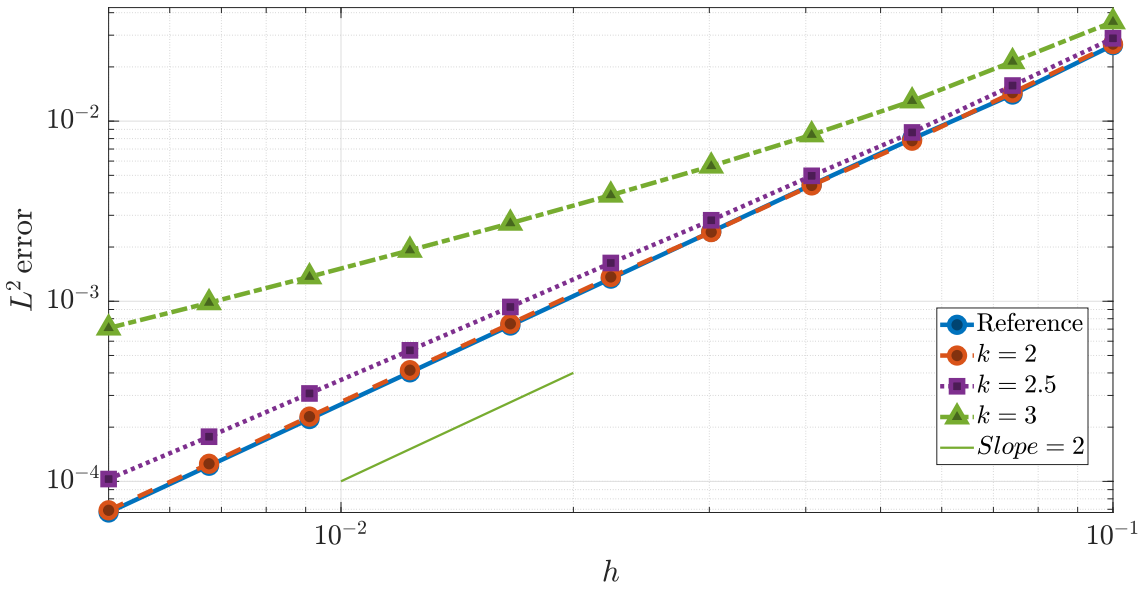



Figure 4: Convergence of the finite element method in the $L^2$-norm. Convergence on a regular reference mesh and convergence on meshes containing bands of caps with height $\overline{h} = h^k$ with $k = 2, 2.5, 3$.

**Acknowledgements**

This research is supported by the European Research Council (project X-MESH, ERC-2022-SyG-101071255). The authors thank Jean-François Remacle, Nicolas Moës, Jonathan Lambrechts and Antoine Quiriny for their kind support within the X-MESH project.

**References**

[1] Babuška, I. and Aziz, A K.: On the angle condition in the finite element method. SIAM J. Numer. Anal. **13**, 2 (1976), 214–226.

[2] Ciarlet, P. G.: *The finite element method for elliptic problems*. North-Holland, Amsterdam, 1978.

[3] Hannukainen, A., Korotov, S., and Křížek, M.: The maximum angle condition is not necessary for convergence of the finite element method. Numer. Math. **120**, 1 (2012), 78–88.

[4] Kobayashi, K. and Tsuchiya, T.: On the circumradius condition for piecewise linear triangular elements. Japan J. Ind. Appl. Math. **32** (2015), 65-76.

[5] Kučera, V.: Several notes on the circumradius condition, Appl. Math. **61**, 3 (2016), 287-298.

[6] Kučera, V.: On necessary and sufficient conditions for finite element convergence, `http://arxiv.org/abs/1601.02942` (preprint), Numer. Math. (submitted).

[7] Oswald, P.: Divergence of the FEM: Babuška-Aziz triangulations revisited. Appl. Math. **60**, 5 (2015), 473–484.

*Programs and Algorithms of Numerical Mathematics 22*
J. Chleboun, J. Papež, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics CAS, Prague 2025

# FINDING A HAMILTONIAN CYCLE USING THE CHEBYSHEV POLYNOMIALS

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 present an algorithm of finding the Hamiltonian cycle in a general undirected graph by minimization of an appropriately chosen functional. This functional depends on the characteristic polynomial of the graph Laplacian matrix and attains its minimum at the characteristic polynomial of the Laplacian matrix of the Hamiltonian cycle.

**Keywords:** Hamiltonian cycle, Chebyshev polynomial, minimization of functional

**MSC:** 65N15, 65M15, 65F08

## 1. Introduction

The problem of finding a Hamiltonian cycle in a general undirected graph is one of the basic optimization tasks and has a wide application not only in logistics, but also in some modern fields, such as computer graphics or microchip construction [3]. However, it belongs to the so-called NP-complete problems [2] and finding an algorithm that could solve NP-complete problems in polynomial time is one of the seven Millennium Prize Problems [1]. In graph theory, there exists a number of sufficient conditions guaranteeing that a given graph is Hamiltonian (i.e. contains a Hamiltonian cycle). These conditions are most often based on some properties of the graph, such as the sum of degrees of non-adjacent vertices or the minimum degree of the graph [4]. In this contribution we apply a different (numerical) approach: The characteristic polynomial of the Laplacian matrix (one may also choose the adjacency matrix) of an undirected graph formed by a single Hamiltonian cycle is related to some Chebyshev polynomial of the first kind. Whereas linearly constrained minimization problem have already been employed for finding a Hamiltonian cycle (e.g. [5]) we use the properties of Chebyshev polynomials and present the algorithm consisting in finding a Hamiltonian cycle by minimization of an appropriately chosen nonlinear functional.

DOI: 10.21136/panm.2024.09

## 2. Graph, its representation and basic properties

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 $\vec{w} \in \{0,1\}^{m \times 1}$, which is a characteristic vector of the set $W \subseteq E$ satisfying $w_i = 1$ if $e_i \in X$ and $w_i = 0$ otherwise.

Using this notation we may define the *vertex-disjoint cycle cover* $\vec{w}$ of the graph $G$ being any set of edges satisfying

$$\vec{w} \in \{0,1\}^{m \times 1}, \tag{1}$$
$$1_m^T \vec{w} = n, \tag{2}$$
$$B\vec{w} = 2 \cdot 1_n, \tag{3}$$

where $1_n = (1, 1, \ldots, 1)^T \in \mathbb{R}^n$. While the second condition ensures the cycle cover contains $n$ edges, the third one guarantees that each vertex coincides with exactly 2 edges.

Further, let $W \subseteq E$ be any set of edges and let $\vec{w} \in \{0,1\}^{m \times 1}$ be its representation. If we denote by $\mathrm{diag}(\vec{w}) \in \{0,1\}^{m \times m}$ a diagonal matrix with the vector $\vec{w}$ on its main diagonal, then

$$L(\vec{w}) = 4I - B\,\mathrm{diag}(\vec{w})\,B^T \tag{4}$$

is the Laplacian matrix of the graph induced by the set $W$. Consequently, if $\mathrm{diag}(\vec{w}) = I$, then $L = 4I - BB^T$ is the Laplacian matrix of the graph $G$.

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

$$\vec{w}_{LS} = B^\dagger(2 \cdot 1_n) = 2 \cdot B^\dagger 1_n. \tag{5}$$

The following lemma provides a characterization of the distribution of all vertex-disjoint cycle covers: they all lie on the same sphere with the center at $\vec{w}_{LS}$ and radius equal to $\sqrt{n - \|\vec{w}_{LS}\|^2}$.

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

$$\|\vec{w} - \vec{w}_{LS}\|^2 = n - \|\vec{w}_{LS}\|^2. \tag{6}$$

*Proof.* One can find the proof in [6]. □

## 3. Definition of the solution space

In this chapter we describe how we chose the solution space in which the minimum of the functional will be searched. At first let us consider any undirected graph containing two different Hamiltonian cycles (cf. Figure 1).

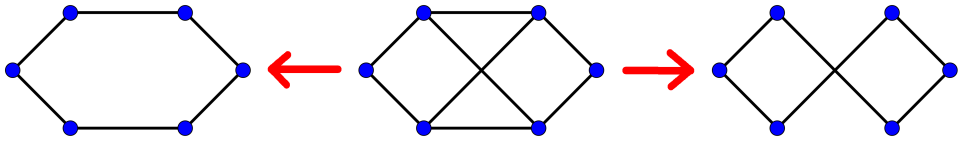


Figure 1: Graph with two Hamiltonian cycles

Consequently, the Laplacian matrices of the subgraphs induced by these Hamiltonian cycles have the following form

$$L_A = \begin{pmatrix} 2 & -1 & 0 & 0 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & 0 & -1 & 2 \end{pmatrix}, \qquad L_B = \begin{pmatrix} 2 & -1 & 0 & 0 & 0 & -1 \\ -1 & 2 & 0 & 0 & -1 & 0 \\ 0 & 0 & 2 & -1 & 0 & -1 \\ 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & -1 & 0 & -1 & 2 & 0 \\ -1 & 0 & -1 & 0 & 0 & 2 \end{pmatrix}.$$

We observe that we can obtain one matrix from the other one simply by simultaneous permutation of columns and rows, i.e. $L_A = PL_BP^T$ for some permutation matrix $P$. Hence, both matrices share a common characteristic polynomial. In this case it has a form

$$p_A(x) \;=\; p_B(x) \;=\; x^6 - 12x^5 + 54x^4 - 112x^3 + 105x^2 - 36x. \tag{7}$$

If the Laplacian matrix of the $n$-cycle is tridiagonal with another two $-1$ in the corners, we call it in the *standard form* (cf. matrix $L_A$).

**Lemma 2.** *Let $L_n$ be the Laplacian matrix of the $n$-cycle in the standard form and $j = n/2$ for $n$ even or $j = (n+1)/2$ for $n$ odd. Then the eigenvectors and eigenvalues of the matrix $L_n$ have the following form:*

$$\vec{u}_{k,1} \;=\; \left[\cos\left(1\,\frac{k\pi}{n}\right), \cos\left(3\,\frac{k\pi}{n}\right), \ldots, \cos\left((2n-1)\,\frac{k\pi}{n}\right)\right]^T, \quad k = 0, 1, \ldots, j-1,$$

$$\vec{u}_{k,2} \;=\; \left[\sin\left(1\,\frac{k\pi}{n}\right), \sin\left(3\,\frac{k\pi}{n}\right), \ldots, \sin\left((2n-1)\,\frac{k\pi}{n}\right)\right]^T, \quad k = 1, 2, \ldots, n-j,$$

$$\lambda_k \;=\; 4\sin^2\left(\frac{k\pi}{n}\right), \quad k = 0, 1, \ldots, n-j.$$

*In this notation vectors $\vec{u}_{k,1}$ and $\vec{u}_{k,2}$ are two linearly independent eigenvectors that correspond to the eigenvalue $\lambda_k$, for all suitable $k$.*

*Proof.* The proof results immediately from the identities

$$-\cos\left((i-2)\frac{k\pi}{n}\right) + 2\cos\left(i\,\frac{k\pi}{n}\right) - \cos\left((i+2)\frac{k\pi}{n}\right) \\ = \; 4\sin^2\left(\frac{k\pi}{n}\right)\cdot\cos\left(i\,\frac{k\pi}{n}\right), \quad (8)$$

$$-\sin\left((i-2)\frac{k\pi}{n}\right) + 2\sin\left(i\,\frac{k\pi}{n}\right) - \sin\left((i+2)\frac{k\pi}{n}\right) \\ = \; 4\sin^2\left(\frac{k\pi}{n}\right)\cdot\sin\left(i\,\frac{k\pi}{n}\right). \quad (9)$$

□

**Remark 3.** *In Lemma 2 for $\lambda_0 = 0$ we obtain a single eigenvector $\vec{u}_{0,1} = 1_n$. When $n$ is even and $j = n/2$ then since $\vec{u}_{j,1} = [0, 0, \ldots, 0]^T$ for eigenvalue $\lambda_j = 4$ only a single eigenvector $\vec{u}_{j,2} = [1, -1, 1, \ldots, -1]^T$ is obtained as well. If we want to consider all eigenvalues $\lambda_k$ with their multiplicities then instead of the upper bound $k = n - j$ we simply take $k = n - 1$ and use the fact that $\lambda_k = \lambda_{n-k}$.*

For given $n \geq 3$ the following lemma provides an expression for the characteristic polynomial of the Laplacian matrix of $n$-cycle (c.f. Table 1).

**Lemma 4.** *Let $n \in \mathbb{N}, n \geq 3$ be given, then the characteristic polynomial of the Laplacian matrix of $n$-cycle has a form*

$$S_n(x) \;=\; 2\cdot\left(T_n\left(\frac{x}{2}-1\right) - (-1)^n\right), \quad (10)$$

*where $T_n(x)$ is the Chebyshev polynomial of the first kind.*

*Proof.* We show that $\lambda_k$, $k = 0, 1, \ldots, n-1$, from Lemma 2 (with their multiplicity) are roots of $S_n$. Hence, let us evaluate $S_n(\lambda_k)$ for $k = 0, 1, \ldots, n-1$:

$$\begin{aligned} S_n(\lambda_k) \;&=\; 2\left(T_n\left(\frac{\lambda_k}{2}-1\right) - (-1)^n\right) \;=\; 2\left(T_n\left(2\sin^2\left(\frac{k\pi}{n}\right)-1\right) - (-1)^n\right) \\ &=\; 2\left(T_n\left(-\cos\left(2\frac{k\pi}{n}\right)\right) - (-1)^n\right) \;=\; 2(-1)^n\left(T_n\left(\cos\left(2\frac{k\pi}{n}\right)\right) - 1\right) \\ &=\; 2(-1)^n\Big(\cos(2k\pi) - 1\Big) \;=\; 2(-1)^n\Big(1-1\Big) \;=\; 0, \end{aligned} \quad (11)$$

where we used the property $T_n(\cos\alpha) = \cos(n\alpha)$ and the parity of the Chebyshev polynomials (cf. [7]).

We shall also evaluate $S_n'(\lambda_k)$ for $k = 0, 1, \ldots, n-1$:

$$\begin{aligned} S_n'(\lambda_k) &= 2T_n'\left(\frac{\lambda_k}{2} - 1\right) = 2nU_{n-1}\left(\frac{\lambda_k}{2} - 1\right) = 2nU_{n-1}\left(-\cos\left(2\frac{k\pi}{n}\right)\right) \\ &= 2n(-1)^{n-1}U_{n-1}\left(\cos\left(2\frac{k\pi}{n}\right)\right) = 2n(-1)^{n-1}\frac{\sin(2k\pi)}{\sin 2\frac{k\pi}{n}} = 0, \quad (12) \end{aligned}$$

for all considered $k$ except $k = 0$ and $k = n/2$. For $\lambda_0 = 0$ and $\lambda_{n/2} = 4$ (for $n$ even) we obtain $S_n'(0) = 2nU_{n-1}(-1) = 2n^2(-1)^{n-1}$ and $S_n'(4) = 2nU_{n-1}(1) = 2n^2$. This corresponds to Lemma 2, since the multiplicity of the root $\lambda_0 = 0$ and $\lambda_{n/2} = 4$ (for $n$ even) is always equal to one. Here we have also employed the properties of the Chebyshev polynomials of the second kind: $T_n'(x) = nU_{n-1}(x)$ and $U_n(\cos\alpha) = \frac{\sin(n+1)\alpha}{\sin\alpha}$ (cf. [7]). □

| $n$ | $T_n(x)$ | $S_n(x)$ |
|---|---|---|
| 3 | $4x^3 - 3x$ | $x^3 - 6x^2 + 9x$ |
| 4 | $8x^4 - 8x^2 + 1$ | $x^4 - 8x^3 + 20x^2 - 16x$ |
| 5 | $16x^5 - 20x^3 + 5x$ | $x^5 - 10x^4 + 35x^3 - 50x^2 + 25x$ |
| 6 | $32x^6 - 48x^4 + 18x^2 - 1$ | $x^6 - 12x^5 + 54x^4 - 112x^3 + 105x^2 - 36x$ |
| 7 | $64x^7 - 112x^5 + 56x^3 - 7x$ | $x^7 - 14x^6 + 77x^5 - 210x^4 + 294x^3 - 196x^2 + 49x$ |
| ⋮ | ⋮ | ⋮ |

Table 1: Comparison of polynomials $T_n(x)$ and $S_n(x)$.

Since the equation (3) has in general infinitely many solutions (e.g. 2-factors) we denote by $\mathcal{H} \subset \mathbb{R}^{m\times 1}$ the set of all its solutions. Each vector $\vec{z} \in \mathcal{H}$ can then be expressed in a form

$$\vec{z} = \vec{z}_0 + \sum_{j=1}^{m-n} \beta_j \cdot \vec{z}_j, \quad (13)$$

where $\vec{z}_0$ is any solution of equation (3) and $\mathrm{span}(\{\vec{z}_j\}_{j=1}^{m-n})$ is the nullspace of $B$.

**Remark 5.** *We consider $\vec{z}_0 = x_{LS}$ being the least-square solution of the equation (3) and $\{\vec{z}_j\}_{j=1}^{m-n}$ form the orthonormal basis.*

Let $\vec{z} \in \mathcal{H}$ be any vector, then in virtue of (4) we define the matrices:

$$L(\vec{z}) = 4 \cdot I - B \cdot \mathrm{diag}(\vec{z}) \cdot B^T. \quad (14)$$

The set of matrices $\mathcal{L} = \{L(\vec{z}),\ \vec{z} \in \mathcal{H}\}$ then has the same dimension as $\mathcal{H}$ and any matrix $L \in \mathcal{L}$ can be expressed in a form $L = L_0 + \sum_{i=1}^{m-n} \beta_i \cdot L_i$ with $L_i = -B \cdot \mathrm{diag}(\vec{z}_i) \cdot B^T$ and $L_0 = 4 \cdot I - B \cdot \mathrm{diag}(\vec{z}_0) \cdot B^T$.

For each $L \in \mathcal{L}$ we shall compute its characteristic polynomial

$$p_L(x) \ = \ \det(x \cdot I - L) \tag{15}$$

and obtain the set of (admissible) polynomials $\mathcal{P} = \{p_L(x),\ L \in \mathcal{L}\}$. Consequently, if the graph $G$ contains the Hamiltonian cycle, then $S_n(x) \in \mathcal{P}$. Hence, we shall try to find a functional $F\colon \mathcal{P} \to \mathbb{R}$ so that there holds:

$$S_n \ = \ \arg\min_{p \in \mathcal{P}} F(p). \tag{16}$$

Thus, the whole problem is reduced to finding the minimum of the functional $F$. Moreover, since

$$\min_{p \in \mathcal{P}} F(p) \ = \ \min_{L \in \mathcal{L}} F(p_L(x)) \ = \ \min_{L \in \mathcal{L}} F(\det(x \cdot I - L)), \tag{17}$$

it suffices to find a proper matrix $L = L_0 + \sum_{i=1}^{m-n} \beta_i \cdot L_i$, i.e. a proper coefficients $\beta_i$, $i = 1, 2, \ldots, m-n$.

## 4. Definition of functionals and their derivatives

### 4.1. Coordinate functional

One possible choice of the functional consists in expressing any polynomial $p \in \mathcal{P}$ in the basis formed by polynomials $S_i$, $1 \le i \le n$, i.e. $p = \sum_{i=1}^{n} \alpha_i \cdot S_i$. Since the minimum of the desired functional should be reached at $p = S_n$, i.e. $\alpha_i = 0$ for $i = 1, 2, \ldots, n-1$, we choose

$$F_c(p) \ = \ \sum_{i=1}^{n-1} \alpha_i^2 \ = \ \sum_{i=1}^{n-1} \alpha_i^2(p). \tag{18}$$

In what follows, we would like to find out, how the coefficients $\alpha_i$ depend on the polynomial $p$. We apply the discrete orthogonality of the Chebyshev polynomials (cf. [7]):

$$\sum_{k=0}^{n-1} T_i(x_k) T_j(x_k) = \begin{cases} 0 & \text{if } i \neq j, \\ n & \text{if } i = j = 0, \\ n/2 & \text{if } i = j \neq 0, \end{cases} \tag{19}$$

where $0 \le i, j < n$ and $x_k$ are Chebyshev's nodes of $T_n(x)$. Then there holds

$$\begin{aligned}
\sum_{k=0}^{n-1} p(y_k)T_j(x_k) &= \sum_{k=0}^{n-1}\left(\sum_{i=1}^{n}\alpha_i S_i(y_k)\right)T_j(x_k)\\
&= 2\sum_{k=0}^{n-1}\sum_{i=1}^{n}\alpha_i\left(T_i\left(\frac{y_k}{2}-1\right)-(-1)^i\right)T_j(x_k)\\
&= 2\sum_{i=1}^{n}\alpha_i\sum_{k=0}^{n-1}T_i(x_k)T_j(x_k) - 2\sum_{i=1}^{n}\alpha_i(-1)^i\sum_{k=0}^{n-1}T_j(x_k)\\
&= 2\sum_{i=1}^{n}\alpha_i\left(\frac{n}{2}\delta_{ij}\right) - 2\sum_{i=1}^{n}\alpha_i(-1)^i\cdot 0 \;=\; n\cdot\alpha_j, \qquad (20)
\end{aligned}$$

where $y_k = 2(x_k+1)$. Hence $\alpha_j = \frac{1}{n}\sum_{k=0}^{n-1} p(y_k)T_j(x_k)$.

Since all polynomials $p$ depend on the choice of the vector $(\beta_1, \beta_2, \ldots, \beta_{m-n})$, we need to compute the derivative of $F$ with respect to $\beta_i$:

$$\frac{\partial F_c}{\partial \beta_i} = \frac{\partial}{\partial\beta_i}\sum_{j=1}^{n-1}\alpha_j^2 = 2\sum_{j=1}^{n-1}\alpha_j\cdot\frac{\partial\alpha_j}{\partial\beta_i}, \qquad (21)$$

$$\frac{\partial\alpha_j}{\partial\beta_i} = \frac{1}{n}\sum_{k=0}^{n-1}T_j(x_k)\,\frac{\partial}{\partial\beta_i}\,p(y_k), \qquad (22)$$

$$\frac{\partial p(y_k)}{\partial\beta_i} = \frac{\partial}{\partial\beta_i}\,\det\Big(y_k I - L_0 - \sum_{j=1}^{m-n}\beta_j\cdot L_j\Big). \qquad (23)$$

If we now denote $R_k = y_k I - L_0 - \sum_{j=1}^{m-n}\beta_j\cdot L_j$, we may apply Jacobi's formula $\big(\det A(x)\big)' = \operatorname{tr}\big(\operatorname{adj} A(x)\cdot A'(x)\big)$ and obtain:

$$\frac{\partial p(y_k)}{\partial\beta_i} = \operatorname{tr}\big(\operatorname{adj} R_k\cdot(-L_i)\big) = -\det R_k\cdot\operatorname{tr}\big(R_k^{-1}\cdot L_i\big), \qquad (24)$$

for nonsingular matrix $R_k$.

### 4.2. Integral functional

Since the polynomials $S_n$ are defined using Chebyshev's polynomials, they solve the following (Chebyshev's) differential equation (cf. [7]):

$$x(4-x)y'' + (2-x)y' + n^2 y = -2n^2(-1)^n, \qquad (25)$$

$$y(0) = 0, \qquad y(4) = 2(1-(-1)^n). \qquad (26)$$

If we transfer the boundary condition and transform the equation into the divergent form we obtain the following quadratic functional:

$$F_{in}(p) \;=\; \int_0^4 \sqrt{x(4-x)}\,(p')^2 - \frac{n^2}{\sqrt{x(4-x)}}\,p^2 + 2\,f_n(x)\,p\,\mathrm{d}x, \tag{27}$$

where $f_n(x) = \frac{(2-x)(n^2-1)}{\sqrt{x(4-x)}}$ for $n$ odd and $f_n(x) = \frac{-2n^2}{\sqrt{x(4-x)}}$ for $n$ even.

**Remark 6.** *To evaluate integrals containing $\sqrt{1-x^2}$ on the interval $(-1,1)$ one can apply the Chebyshev-Gauss quadrature rules:*

$$\int_{-1}^{1} \frac{f(x)}{\sqrt{1-x^2}}\,\mathrm{d}x \;\approx\; \sum_{k=1}^{n} w_k\,f(x_k), \qquad \int_{-1}^{1} g(x)\,\sqrt{1-x^2}\,\mathrm{d}x \;\approx\; \sum_{k=1}^{n} \widehat{w}_k\,g(\widehat{x}_k),$$

*where $x_k$ are roots of $T_n(x)$ and $w_k = \frac{\pi}{n}$, while $\widehat{x}_k$ are roots of $T'_{n+1}(x)$ and $\widehat{w}_k = \frac{\pi}{n+1}\sin^2\left(\frac{\pi\,k}{n+1}\right)$. These formulas are exact for polynomials up to order $2n-1$ and when we transform them on the interval $[0,4]$, we obtain the following expressions:*

$$\int_0^4 \frac{p^2(y)}{\sqrt{y(4-y)}}\,\mathrm{d}y \;=\; 2\pi\left[\sum_{i=1}^{n}\alpha_i^2 + 2\left(\sum_{i=1}^{n}\alpha_i(-1)^i\right)^2\right], \tag{28}$$

$$\int_0^4 \sqrt{y(4-y)}(p'(y))^2\,\mathrm{d}y \;=\; 2\pi\sum_{i=1}^{n} i^2\alpha_i^2, \tag{29}$$

$$\int_0^4 f_n(y)p(y)\,\mathrm{d}y \;=\; 4\pi n^2\sum_{i=1}^{n}(-1)^i\alpha_i, \text{ for } n \text{ even}, \tag{30}$$

$$\int_0^4 f_n(y)p(y)\,\mathrm{d}y \;=\; -2\pi(n^2-1)\alpha_1, \text{ for } n \text{ odd}, \tag{31}$$

*providing $p(y) = \sum_{i=1}^{n}\alpha_i S_i(y)$. Consequently, for the functional $F_{in}$ there holds:*

$$F_{in}(p) \overset{n \text{ even}}{=} -2\pi\left[\sum_{i=1}^{n}(n^2-i^2)\alpha_i^2 - 2n^2\left(1-\sum_{i=1}^{n}\alpha_i(-1)^i\right)\sum_{i=1}^{n}\alpha_i(-1)^i\right], \tag{32}$$

$$F_{in}(p) \overset{n \text{ odd}}{=} -2\pi\left[\sum_{i=1}^{n}(n^2-i^2)\alpha_i^2 + 2n^2\left(\sum_{i=1}^{n}\alpha_i(-1)^i\right)^2 + (n^2-1)\alpha_1\right]. \tag{33}$$

Unfortunately, these functionals failed to be positive and, hence, they do not attain their minimum in $p = S_n$. Therefore, together with the functional $F_c$ (cf. (18)) we consider only functional (28) (functional $F_{in,1}$) and (29) (functional $F_{in,2}$) with the sums ending at $i = n-1$.

## 5. Numerical experiments

For the minimization we employ the gradient descent method with the backtracking line search driven by the Armijo condition (cf. Figure 2). We consider random graphs with 16 vertices and 18–32 edges containing Hamiltonian cycle. For each kind of graph and for each functional we generate 100 random graphs. The results in Table 2 show numbers of graphs for which the algorithm successfully ended and found the Hamiltonian cycle. If the algorithm failed, it was due to finding a local extremum or exceeding the maximum number of iterations.

| functional | 16/18 | 16/20 | 16/22 | 16/24 | 16/26 | 16/28 | 16/30 | 16/32 |
|---|---|---|---|---|---|---|---|---|
| $F_c$ | 93 | 80 | 82 | 70 | 58 | 55 | 54 | 50 |
| $F_{in,1}$ | 77 | 61 | 63 | 54 | 48 | 35 | 37 | 31 |
| $F_{in,2}$ | 96 | 88 | 80 | 66 | 63 | 71 | 63 | 62 |

Table 2: Numerical results for all considered functionals.

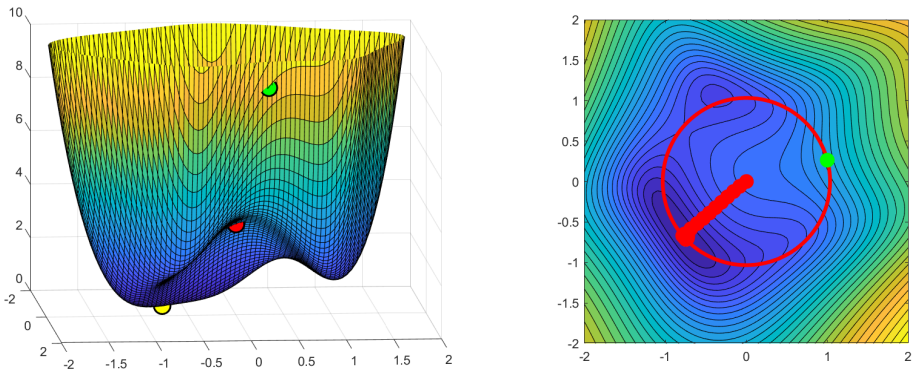



Figure 2: An example of minimization algorithm for the functional $F_c$ and a graph with 7 vertices and 9 edges. The minimum lies on the circle with the center in $x_{LS}$. The other point on the circle corresponds to the 2-factor of the graph considered.

## 6. Conclusion

Numerical experiments show that all three functionals contain unwanted local extrema which cause problems during minimization process. It also results from the Table 2 that the more edges a graph has, the more complicated it is to reach the global minimum. Of these three algorithms, algorithm $F_{in,1}$ provided the worst results, probably due to the presence of the oscillation term $\left(\sum_{i=1}^{n-1} \alpha_i(-1)^i\right)^2$.

The construction of another (hopefully convex) functional, as well as improvements to the minimization process and different choice of the null space basis of the incidence matrix $B$ will be the subject of the future research.

**Acknowledgements**

This work was supported by the Grant Agency of the Czech Technical University in Prague, grant No. SGS24/002/OHK1/1T/11.

**References**

[1] Carlson J. et al.: *The Millennium Prize Problems*. American Mathematical Society, For The Clay Mathematics Institute, 2006.

[2] Fortnow L.: The status of the P versus NP problem. Commun. ACM **52** (9) (2009), 78–86.

[3] Girard P.: Survey of low-power testing of VLSI circuits. IEEE Design & Test of Computers **19** (3) (2002), 82–92.

[4] Gould R. J.: Advances on the Hamiltonian problem - A survey. Graphs Comb. **19** (1) (2003), 7–52.

[5] Haythorpe M. and Murray W.: Finding a Hamiltonian cycle by finding the global minimizer of a linearly constrained problem. Comput. Optim. Appl. **81** (1) (2022), 309–336.

[6] Lamač J. and Vlasák M.: Finding vertex-disjoint cycle cover of undirected graph using the least-squares method. Proceedings of PANM 21, Institute of Mathematics CAS, pp. 97–106, 2023.

[7] Mason J. C., Handscomb D. C.: *Chebyshev Polynomials*. Chapman and Hall/CRC Press Company, New York, 2003.

[8] Penrose, R.: A generalized inverse for matrices. Mathematical Proceedings of the Cambridge Philosophical Society, **51**(3) (1955), 406–413.

*Programs and Algorithms of Numerical Mathematics 22*
J. Chleboun, J. Papež, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics CAS, Prague 2025

# NONSMOOTH EQUATION METHOD FOR NONLINEAR NONCONVEX OPTIMIZATION

Ladislav Lukšan, Ctirad Matonoha, Jan Vlček

Institute of Computer Science, Czech Academy of Sciences,
Pod vodárenskou věží 2, 182 00 Prague 8, Czech Republic
luksan@cs.cas.cz, matonoha@cs.cas.cz

**Abstract:** The contribution deals with the description of two nonsmooth equation methods for inequality constrained mathematical programming problems. Three algorithms are presented and their efficiency is demonstrated by numerical experiments.

**Keywords:** nonlinear programming, nonsmooth analysis, semismooth equations, KKT systems, algorithms

**MSC:** 90C30, 49M05, 49M37, 65K05

## 1. Introduction

In this contribution, we are concerned with a nonlinear programming problem (NP): Find the minimum of a function $F(x)$ on the set given by constraints $c(x) \leq 0$, where $F\colon R^n \to R$, $c\colon R^n \to R^m$ are twice continuously differentiable mappings ($c(x) \leq 0$ is considered by elements).

Necessary conditions (the KKT conditions) for the solution of problem (NP) (if the gradients of active constraints are linearly independent) have the following form

$$g(x,u) = 0, \quad c(x) \leq 0, \quad u \geq 0, \quad UC(x)e = 0, \tag{1}$$

where

$$g(x,u) = \nabla F(x) + \sum_{k=1}^{m} u_k \nabla c_k(x) = \nabla F(x) + A(x)u$$

and $A(x) = [\nabla c_k(x)\colon 1 \leq k \leq m]$. Here $u \in R^m$ are the vectors of Lagrange multipliers, $U = \operatorname{diag}(u_k\colon 1 \leq k \leq m)$, $C(x) = \operatorname{diag}(c_k(x)\colon 1 \leq k \leq m)$ and $e$ is the vector with unit elements.

Nonlinear programming problems are frequently solved by three types of methods:

DOI: 10.21136/panm.2024.10

- Sequential quadratic programming (SQP) methods: In this case, the quadratic programming subproblem

$$\text{Minimize} \quad Q(d) = \frac{1}{2} d^T B d + g^T d, \quad \text{where} \quad A^T d + c \leq 0,$$

  is solved in every iteration.

- Interior points (IP) methods: In this case, we solve the sequence of equality constrained problems

$$\text{Minimize} \quad F(x) - \mu e^T \log(S) e, \quad \text{where} \quad c(x) + s = 0,$$

  where $S = \text{diag}(s_k \colon 1 \leq k \leq m) > 0$ and $\mu \to 0$. The constraints $s \geq 0$ are satisfied algorithmically using the bounds for stepsizes.

- Nonsmooth equation (NE) methods: In this case, we solve the equality constrained problem

$$\text{Minimize} \quad F(x), \quad \text{where} \quad h(x, u) = 0,$$

  in every iteration. The set of equations $h(x, u) = 0$ is usually nonsmooth.

SQP methods require an efficient solution of the quadratic programming subproblem. In the large scale case it usually consumes a large computational time. IP and NE methods, which transform inequality constrained problems to equality constrained ones, are very efficient.

## 2. Nonsmooth equation methods

Inequalities in (1), so called complementarity conditions, can be transformed to equations using the Fischer-Burmeister function [2]

$$\psi(a, b) = \sqrt{a^2 + b^2} - (a + b),$$

which is zero if and only if $a \geq 0$, $b \geq 0$ and $ab = 0$. The Fischer-Burmeister function $\psi(a, b)$ is continuously differentiable if $|a| + |b| \neq 0$ and semismooth if $|a| + |b| = 0$. Moreover, function $\psi^2(a, b)$ is continuously differentiable everywhere. The gradient and the Clarke subdifferential of the Fischer-Burmeister function are given by the formulas

$$\nabla \psi(a, b) = \begin{bmatrix} \frac{a}{\sqrt{a^2+b^2}} - 1 \\ \frac{b}{\sqrt{a^2+b^2}} - 1 \end{bmatrix}, \quad |a| + |b| \neq 0, \tag{2}$$

$$\partial \psi(0, 0) = \text{conv} \bigcup_{\phi \in [0, 2\pi]} \begin{bmatrix} \cos \phi - 1 \\ \sin \phi - 1 \end{bmatrix}. \tag{3}$$

Formula (3) implies that $[-1,-1]^T \in \partial\psi(0,0)$. Therefore, setting $r(a,b) = \sqrt{a^2+b^2}$ for $|a|+|b| \neq 0$ and $r(a,b) = 1$ for $|a|+|b| = 0$ we obtain

$$\begin{bmatrix} \frac{a}{r(a,b)} - 1 \\ \frac{b}{r(a,b)} - 1 \end{bmatrix} \in \partial\psi(a,b). \tag{4}$$

Complementarity conditions in (1) are satisfied if and only if $\psi(u_k, -c_k(x)) = 0$, $1 \leq k \leq m$, so (1) can be replaced by the system of nonlinear equations

$$f(z) = f(x,u) = \begin{bmatrix} g(x,u) \\ h(x,u) \end{bmatrix} = 0, \tag{5}$$

where $h(x,u) = [\psi(u_k, -c_k(x)) : 1 \leq k \leq m]^T$. The mapping $f(z)$ is semismooth at every point $z \in R^{n+m}$. Therefore

$$f'(z,d) = Jd + o(\|d\|) \quad \text{if} \quad \|d\| \to 0 \quad \text{and} \quad J \in \partial f(z+d)$$

and

$$f(z+d) - f(z) = f'(z,d) + o(\|d\|) = Jd + o(\|d\|). \tag{6}$$

Linearizing system (5) by using (6), we obtain a step of the Newton method

$$x_+ = x + d_x, \qquad u_+ = u + d_u,$$

where

$$\begin{bmatrix} B & A \\ (R+C)R^{-1}A^T & -(R-U)R^{-1} \end{bmatrix} \begin{bmatrix} d_x \\ d_u \end{bmatrix} = - \begin{bmatrix} g(x,u) \\ h(x,u) \end{bmatrix}, \tag{7}$$

and where

$$B \approx G(x,u) = \nabla^2 F(x) + \sum_{k=1}^{m} u_k \nabla^2 c_k(x),$$

$A = A(x)$, $C = C(x)$, $U = U(x)$, $R = \operatorname{diag}(r_k : 1 \leq k \leq m)$, $r_k = \sqrt{c_k(x)^2 + u_k^2}$.

The algorithm of a nonsmooth equation method can be roughly described in the following way. For given vectors $x \in R^n$, $u \in R^m$ we determine direction vectors $d_x$, $d_u$ by solving a linear system equivalent to (7). Furthermore, we choose new vectors $x_{i+1}$, $u_{i+1}$ by using a suitable merit function (Section 4) or by using a combined filter (Section 5).

## 3. Determination of a direction vector

Linear system (7) is not suitable for iterative solvers in general since it is nonsymmetric and can have unsuitable diagonal elements. A symmetric linear system can be obtained by multiplying the second row of (7) by the matrix $(R+C)^{-1}R$. Then

$$\begin{bmatrix} B & A \\ A^T & -M \end{bmatrix} \begin{bmatrix} d_x \\ d_u \end{bmatrix} = - \begin{bmatrix} g(x,u) \\ (R+C)^{-1}R\, h(x,u) \end{bmatrix},$$

where $M = (R + C)^{-1}(R - U)$ is a diagonal positive definite matrix. Diagonal elements of $M$ can be very large in general. Therefore, we eliminate direction vectors corresponding to inactive constraints.

**Definition 1.** *A constraint with index $k$ is active if*

$$-\frac{\partial \psi_k}{\partial u_k} \le \hat{\varepsilon}\,\frac{\partial \psi_k}{\partial c_k} \quad \Longleftrightarrow \quad r_k - u_k \le \hat{\varepsilon}(r_k + c_k),$$

*where $\psi_k = \psi(u_k, -c_k)$ and $\hat{\varepsilon} > 0$ (usually $0.01 \le \hat{\varepsilon} \le 1$). Active quantities are denoted by $\hat{c}_k$, $\hat{u}_k$, $\hat{r}_k$, $\hat{M}$ and inactive quantities are denoted by $\check{c}_k$, $\check{u}_k$, $\check{r}_k$, $\check{M}$.*

Eliminating inactive directions we obtain

$$\check{d}_u = \check{M}^{-1}(\check{A}^T d_x + \check{c}) - \check{u}, \tag{8}$$

$$\begin{bmatrix} \hat{B} & \hat{A} \\ \hat{A}^T & -\hat{M} \end{bmatrix} \begin{bmatrix} d_x \\ \hat{d}_u \end{bmatrix} + \begin{bmatrix} \hat{g}(x,u) \\ (\hat{R} + \hat{C})^{-1}\hat{R}\hat{h}(x,u) \end{bmatrix} = \begin{bmatrix} r_x \\ \hat{r}_u \end{bmatrix}, \tag{9}$$

where

$$\hat{B} = B + \check{A}\check{M}^{-1}\check{A}^T, \qquad \hat{g}(x,u) = g(x,u) + \check{A}\check{M}^{-1}\check{c}.$$

To obtain direction vectors $d_x$, $\hat{d}_u$, we solve linear equations (9) with sufficient precisions $r_x$, $\hat{r}_u$ and compute $\check{d}_u$ by (8). Note that $\|\hat{M}\| \le \hat{\varepsilon}$, $\|\check{M}^{-1}\| < 1/\hat{\varepsilon}$ and

$$\|\hat{M}\| \to 0, \quad \|\check{M}^{-1}\| \to 0 \quad \text{if} \quad \hat{g}(x.u) \to 0, \quad \hat{h}(x,u) \to 0.$$

Symmetric matrix $\hat{B}$ has a bounded norm and is positive definite if $B$ is positive definite. For this reason we use a positive definite matrix $B = G + E$ obtained by using the Gill-Murray decomposition [3] of $G = G(x,u)$ ($B$ is positive definite if it is obtained by the quasi-Newton method).

Nonsmooth equation methods for nonlinear programming problems are realized by the following algorithm.

**Algorithm 1.** *Line search method.*

**Data:** *Parameter for active constraint determination $\hat{\varepsilon}$. Precisions $0 < \overline{\omega}_x < 1$, $0 < \overline{\omega}_u < 1$. Maximum stepsize $\overline{\Delta} > 0$.*

**Input:** *Initial approximation of a KKT point $x$.*

**Step 1:** Initiation. *Choose initial Lagrange multipliers $u_k$, $1 \le k \le m$, such that $u_k \ne 0$. Compute value $F(x)$ and vector $c(x)$. If a filter is used, set $n_F = 1$ and $\mathcal{F} = \{F(x), \Phi(x,u)\}$. Set $i := 0$.*

**Step 2:** Termination. *Compute matrix $A := A(x)$ and vector $g := g(x,u)$. If (8) holds with a required precision, terminate computation, else set $i := i + 1$.*

**Step 3:** Hessian matrix approximation. *Determine positive definite matrix $B$ as an approximation of the Hessian matrix $G(x,u)$.*

**Step 4:** Determination of direction vectors. *Divide constraints into active and inactive parts using parameter $\hat{\varepsilon}$ to obtain system (9). Determine vectors $d_x$, $\hat{d}_u$ as approximate solutions of (9) with precisions $r_x$, $\hat{r}_u$ and compute vector $\check{d}_u$ by (8). If a merit function is used, determine value $\sigma \geq 0$ by (12) and compute derivative $\varphi'(0)$ by (11).*

**Step 5:** Stepsize selection. *Determine stepsize $t > 0$ using Algorithm 2 or Algorithm 3 and set $x := x + td_x$, $u := u + td_u$. Compute value $F(x)$, vector $c(x)$ and go to Step 2.*

## 4. Line search with a merit function

After obtaining direction vectors $d_x$, $d_u$, we seek a stepsize $t$ to decrease the value of the merit function

$$\varphi(t) = F_j(x + td_x) + \sigma P_j(x + td_x, u + td_u), \quad \sigma \geq 0, \quad j = 1, 2,$$

where

$$\begin{aligned} F_1(x + td_x) &= F(x + td_x), \\ F_2(x + td_x) &= F(x + td_x) + (u + d_u)^T c(x + td_x), \\ P_1(x + td_x, u + td_u) &= \|h(x + td_x, u + td_u)\|_1, \\ P_2(x + td_x, u + td_u) &= \frac{1}{2}\|h(x + td_x, u + td_u)\|^2. \end{aligned}$$

It is necessary that $\varphi'(0) < 0$ holds and that the stepsize $t$ satisfies the Armijo condition

$$\varphi(t) - \varphi(0) \leq \varepsilon_1 t\varphi'(0), \quad \text{where} \quad 0 < \varepsilon_1 < 1/2. \tag{10}$$

For subsequent investigations, we use the notation

$$\begin{aligned} F_1: &\quad \chi(r) = d_x^T r_x - (\hat{u} + \hat{d}_u)^T \hat{r}_u, \\ F_1: &\quad \gamma_0 = (u + d_u)^T M(u + d_u) - (u + d_u)^T c, \\ P_1: &\quad \gamma_1 = \|h\|_1 - \|(\hat{R} + \hat{C})\hat{R}^{-1}\hat{r}_u\|_1, \\ \\ F_2: &\quad \chi(r) = d_x^T r_x, \\ F_2: &\quad \gamma_0 = 0, \\ P_2: &\quad \gamma_1 = \|h\|^2 - \hat{h}^T(\hat{R} + \hat{C})\hat{R}^{-1}\hat{r}_u. \end{aligned}$$

It is necessary that $\gamma_1 > 0$ holds, which is satisfied if

$$P_1: \ \|\hat{r}_u\|_1 \leq \frac{\overline{\omega}_u}{2}\|h\|_1, \quad P_2: \ \|\hat{r}_u\| \leq \frac{\overline{\omega}_u}{2}\|h\|, \quad \text{where} \quad 0 \leq \overline{\omega}_u < 1.$$

**Theorem 1** ([4]). *Let vectors $d_x$, $\hat{d}_u$ be obtained as an approximate solution of (9) and vector $\check{d}_u$ be obtained by (8). Then*

$$\varphi'(0) = -d_x^T B d_x - \gamma_0 - \gamma_1 \sigma + \chi(r). \tag{11}$$

*If $\gamma_1 > 0$,*

$$\sigma \geq \underline{\sigma} > -\frac{d_x^T B d_x + \gamma_0}{\gamma_1}, \tag{12}$$

*and if system (9) is solved with the precision*

$$\chi(r) < d_x^T B d_x + \gamma_0 + \gamma_1 \sigma, \tag{13}$$

*then $\varphi'(0) < 0$.*

**Algorithm 2.** *Line search with a merit function.*

**Data:** *Parameters $0 < \beta < 1$, $0 < \varepsilon_1 < 1/2$, minimum stepsize $0 < \underline{t} < 1$. Derivative $\varphi'(0)$ obtained from (11)*

**Input:** *Pair $(x, u)$, values $F(x)$, $c(x)$ and direction pair $(d_x, d_u)$ obtained as a solution of equations (8)–(9).*

**Step 1:** *Choose initial stepsize $t > 0$ (usually $t = 1$). If $\varphi'(0) \geq 0$ go to Step 5.*

**Step 2:** *If $t < \underline{t}$, go to Step 5, else compute new values $F(x + td_x)$ and $c_k(x + td_x)$, $1 \leq k \leq m$.*

**Step 3:** Minimization of the objective function. *If the Armijo condition (7) is satisfied, go to Step 6.*

**Step 4:** *Set $t := \beta t$ and go to Step 2.*

**Step 5:** Restart. *Choose well positive diagonal matrix $D$ (usually $D = I$). Solve precisely equations (8)–(9) with $B$ replaced by $D$. Set $\sigma = 0$ and compute derivative $\varphi'(0) < 0$ from (11). Find stepsize $0 < t < 1$ such that $F(x + td_x) < F(x)$.*

**Step 6:** *Terminate stepsize selection ($t > 0$ is an obtained stepsize).*

The line search methods with a merit function are very efficient, namely if we use the Lagrangian function $F_2(x, u)$ and if the penalty parameter can decrease. Unfortunately, in this case the global convergence cannot be proved.

## 5. Line search with a filter

Denote for simplicity $z = (x, u)$, $\Phi(z) = (1/2)\|h(x, u)\|^2$ and $g(z) = g(x, u)$. At the same time, although $F$ does not depend on $u$, let for consistency $F(z) = F(x)$.

**Definition 2.** *Let $F(z_1) \leq F(z_2)$ and $\Phi(z_1) \leq \Phi(z_2)$. Then we say that the pair $(F(z_2), \Phi(z_2))$ is dominated by the pair $(F(z_1), \Phi(z_1))$. A filter $\mathcal{F} = \{(F_j, \Phi_j) \colon 1 \leq j \leq n_F\}$ is a set of pairs where no pair is dominated by another pair ($n_F$ is a number of pairs in the filter).*

The line search with a filter procedure uses three strategies for obtaining new trial points. If $t < \underline{t}$, where $\underline{t} > 0$ is a computed lower bound, we use a feasibility restoration phase. In this case, we determine a new vector $d_z \in R^{n+m}$ and a suitable stepsize $t > 0$ by minimizing $\Phi(z)$ to satisfy (17). If $t \geq \underline{t}$, we first check whether

$$F(z + td_z) < F_j \quad \text{or} \quad \Phi(z + td_z) < \Phi_j \tag{14}$$

holds for $1 \leq j \leq n_F$ (otherwise, the stepsize is shortened). If $t \geq \underline{t}$ and

$$d_z^T \nabla F(z) < 0, \qquad -d_z^T \nabla F(z) t > \delta_3 \Phi^\nu(z), \tag{15}$$

where $\delta_3 > 0$ a $\nu > 1$, the stepsize selection is terminated if

$$F(z + td_z) - F(z) \leq \varepsilon_1 t d_z^T \nabla F(z), \tag{16}$$

where $0 < \varepsilon_1 < 1/2$ (the Armijo condition). If $t \geq \underline{t}$ and (15) does not hold, the stepsize selection is terminated if

$$F(z + td_z) < F(z) - \delta_1 \Phi(z) \quad \text{or} \quad \Phi(z + td_z) < \Phi(z) - \delta_2 \Phi(z), \tag{17}$$

where $0 < \delta_1 < 1$ and $0 < \delta_2 < 1$ (the filter condition).

**Algorithm 3.** *Line search with a filter.*

**Data:** *Parameters* $0 < \beta < 1$, $0 < \varepsilon_1 < 1/2$, $0 < \delta_1 < 1$, $0 < \delta_2 < 1$, $\delta_3 > 0$, $0 < \delta_4 < 1$, *size of filter* $n_F \geq 1$, *maximum size of filter* $m_F > 1$, *filter* $\mathcal{F} = \{(F_j, \Phi_j) \colon 1 \leq j \leq n_F\}$ *(usually* $n_F = 1$ *and* $\mathcal{F} = \{F(z), \Phi(z)\}$*).*

**Input:** *Pair* $z = (x, u)$, *values* $F(z)$, $\Phi(z)$ *and direction vector* $d_z = (d_x, d_u)$ *obtained as a solution of equations* (8)–(9).

**Step 1:** *Compute minimum stepsize* $\underline{t} > 0$ *by* (18). *Choose initial stepsize* $t > 0$ *(usually* $t = 1$*).*

**Step 2:** *If* $t < \underline{t}$, *go to Step* 6. *If* $t \geq \underline{t}$, *compute new values* $F := F(z + td_z)$ *and* $\Phi := \Phi(z + td_z)$. *If* $(F, \Phi) \in \mathcal{F}$ *(i.e.,* (14) *does not hold), go to Step* 5.

**Step 3:** Minimization of the objective function. *If* (15) *holds and Armijo condition* (16) *is satisfied, go to Step* 8. *If* (15) *holds and Armijo condition* (16) *is not satisfied, go to Step* 5.

**Step 4:** Utilization of the filter. *If* (15) *does not hold and condition* (17) *is satisfied, go to Step* 7. *If* (15) *does not hold and condition* (17) *is not satisfied, go to Step* 5.

**Step 5:** *Set* $t := \beta t$ *and go to Step* 2.

**Step 6:** Feasibility restoration. *Find a new direction vector* $d_z$ *and a suitable stepsize* $t > 0$ *in such a way that the values* $F := F(z + td_z)$, $\Phi := \Phi(z + td_z)$ *satisfy conditions* $(F, \Phi) \notin \mathcal{F}$ *and* $\Phi < \Phi(z) - \delta_2' \Phi(z)$, *where* $\delta_2' > 0$.

**Step 7:** Filter update. *Compute values* $F = F(z) - \delta_1 \Phi(z)$, $\Phi = \Phi(z) - \delta_2 \Phi(z)$. *Remove from the filter pairs* $(F_j, \Phi_j)$ *dominated by* $(F, \Phi)$ *and add* $(F, \Phi)$ *into the filter.*

**Step 8:** *Terminate stepsize selection* ($t > 0$ *is an obtained stepsize).*

The minimum stepsize is computed by the rule [8]

$$\begin{aligned} \underline{t} &= \delta_4 \min\left(\varepsilon_0, \frac{\delta_1 \Phi(z)}{|d_z^T \nabla F(z)|}, \frac{\delta_3 \Phi^\nu(z)}{|d_z^T \nabla F(z)|}\right), \qquad && d_z^T \nabla F(z) < 0, \\ \underline{t} &= \delta_4 \varepsilon_0, && d_z^T \nabla F(z) \geq 0, \end{aligned} \tag{18}$$

where $0 < \delta_4 < 1$.

The line search method with a filter is globally convergent (i.e, the process, started from an arbitrary point, converges to the KKT point) if the following standard assumptions are satisfied:

- Functions $F(x)$ and $c_k(x)$, $1 \leq k \leq m$, are twice continuously differentiable. Function values and derivatives are uniformly bounded.
- Matrices appearing in (9) are uniformly nonsingular.
- Matrices $B$ in (7) are uniformly bounded and uniformly positive definite.
- Conditions $|u_k| + |c_k(x)| \geq \varepsilon$ (strict complementarity) and $r_k + c_k(x) \neq 0$ are satisfied.

**Theorem 2** ([8])**.** *Consider a nonsmooth equation line search method realized by Algorithm 1 and Algorithm 2. If standard assumptions for global convergence are satisfied, then* $\|h(z)\| \to 0$.

**Theorem 3** ([7])**.** *Consider a nonsmooth equation method, where equations* (8)–(9) *are solved with the precisions*

$$d_x^T r_x \leq \overline{\omega}_x d_x^T \hat{B} d_x, \qquad \|\hat{r}_u\| \leq \overline{\omega}_u \|\hat{c}(x)\|,$$

*where* $0 \leq \overline{\omega}_x < 1$, $0 \leq \overline{\omega}_u < 1$. *Let the stepsizes be determined by Algorithm 3. If standard assumptions for global convergence are satisfied, then the method is globally convergent.*

## 6. Computational experiments

The computational comparisons were preformed using the system for universal functional optimization UFO [5] on the collection of test problems TEST21. This collection contains 18 problems with 1000 variables and is a modification of the collection TEST20 described in [6]. The comparisons were made using the performance

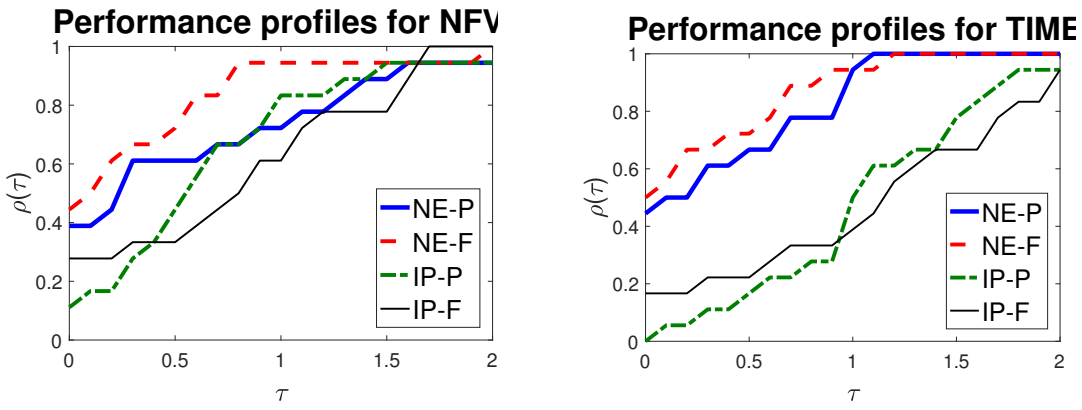



Figure 1: Comparison of Newton's methods.

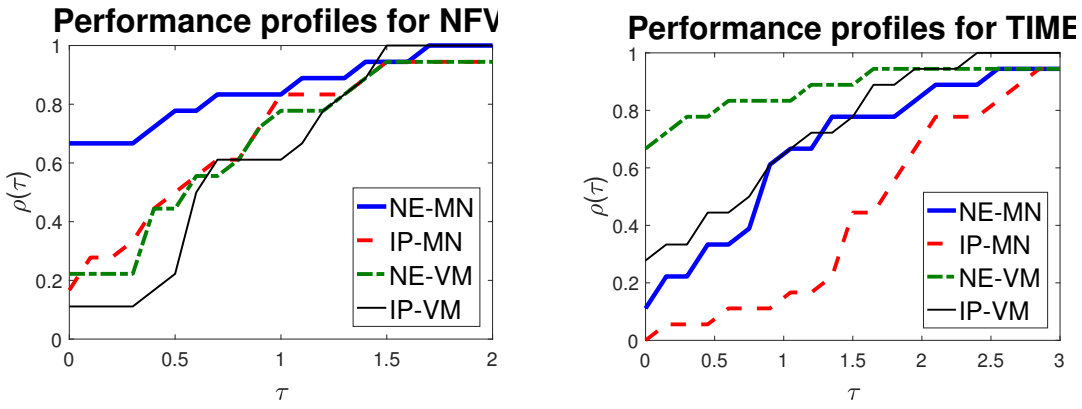



Figure 2: Comparison of variable metric methods.

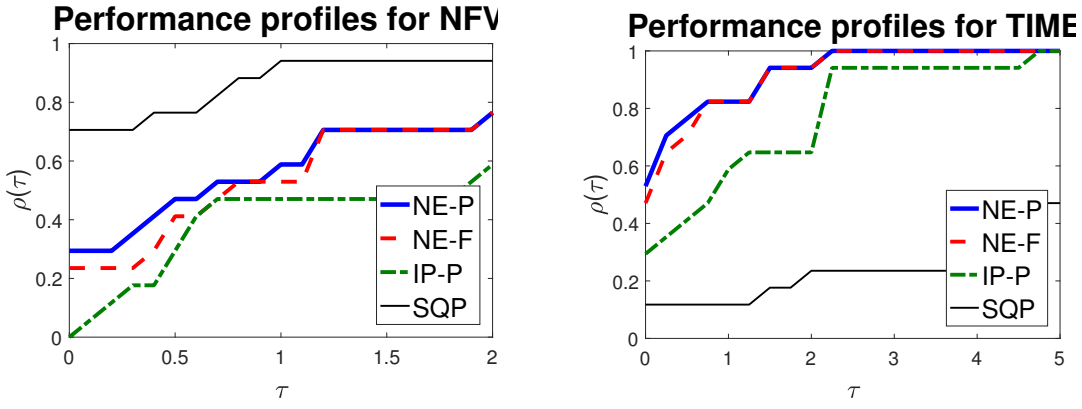



Figure 3: Comparison with the sequential quadratic programming method (SQP).

profiles for the number of function evaluations (NFV) and for the total computational time (TIME). The details about performance profiles as well as the meaning of $\tau$ and $\rho(\tau)$ used in Figures 1–3 can be found in [1]. The following notation is used:

| | |
|---|---|
| NE – nonsmooth equation methods, | IP – interior point methods, |
| P – merit function, | F – filter, |
| MN – Newton methods, | VM – variable metric methods. |

**Acknowledgements**

This work was supported by the long-term strategic development financing of the Institute of Computer Science (RVO:67985807).

**References**

[1] Dolan, E. D. and Moré, J. J.: Benchmarking optimization software with performance profiles. Math. Program. **91** (2002), 201–213.

[2] Fischer, A.: A special Newton-type optimization method. Optimization **24** (1992), 269–284.

[3] Gill, P. E. and Murray, W.: Newton type methods for unconstrained and linearly constrained optimization. Math. Program. **7** (1974), 311–350.

[4] Lukšan, L., Matonoha, C., and Vlček, J.: Nonsmooth equation method for nonlinear nonconvex optimization. In: M. Křížek, P. Neittaanmäki, R. Glovinski, S. Korotov (eds.), *Conjugate Gradient Algorithms and Finite Element Methods*, pp. 131–145. Springer Verlag, Berlin, 2004.

[5] Lukšan, L., Tůma, M., Matonoha, C., Vlček J., Ramešová, N., Šiška, M., and Hartman, J.: UFO 2024. Interactive System for Universal Functional Optimization. Technical Report V-1289. Prague, ICS AS CR, 2024.

[6] Lukšan, L. and Vlček, J.: Sparse and partially separable test problems for unconstrained and equality constrained optimization. Technical Report V-767. Prague, ICS AS CR 1999.

[7] Stuchlý, J.: Řešení rozsáhlých úloh nelineárního programování metodami vnitřního bodu. Diplomová práce, Univerzita Karlova, Praha 2004.

[8] Wächter, A. and Biegler, L. T.: Line search filter methods for nonlinear programming - motivation and global convergence. SIAM J. on Computation **16** (2005), 1–31.

*Programs and Algorithms of Numerical Mathematics 22*
J. Chleboun, J. Papež, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics CAS, Prague 2025

# NUMERICAL APPROXIMATION OF AEROACOUSTICS INDUCED BY FLOW OVER A SQUARE CYLINDER

Tomáš Marhan, Petr Sváček

Department of Technical Mathematics, Faculty of Mechanical Engineering,
Czech Technical University in Prague
Technická 4, 166 01 Prague 6, Czechia
tomas.marhan@fs.cvut.cz, petr.svacek@fs.cvut.cz

**Abstract:** This study investigates the generation of aeroacoustic sound resulting from the interaction of flow with a square cylinder at a Reynolds number of 150 and a Mach number of 0.2. The analysis combines the Finite Volume Method (FVM) for fluid dynamics using the OpenFOAM framework with the Finite Element Method (FEM) for acoustics implemented via the FEniCS Python library.

**Keywords:** finite element method, finite element method, aeroacoustic, OpenFOAM, FEniCS, CFD, CAA

**MSC:** 76Q05

## 1. Introduction

For many years, Computational Fluid Dynamics (CFD) has been widely used across various scientific and industrial fields. With recent advancements in computational resources, it has become feasible to study also flow-induced noise from bluff bodies, such as the noise generated by aircraft landing gear or car side mirrors.

Aeroacoustics, the study of noise generated and propagated by fluid flows, poses a unique challenge because the sound pressure is much smaller than the atmospheric pressure. Moreover, as the Mach number decreases, the disparity between the fluid length scale and the acoustic length scale (wavelength) increases. Consequently, the mesh size required to resolve fluid length scales becomes significantly smaller than that needed for acoustic length scales. To address this, a variety of Computational Aeroacoustics (CAA) methodologies have been adopted, many of which separate the flow field from the acoustic computation in a hybrid approach (see [7] for an overview). The aim is to derive the equations that describe the generation of sound waves propagating into the acoustic field, separately from those that define fluid motion in the unsteady flow. The hybrid approach has been successfully applied in cases like low Mach airframe noise in [4] and human phonation in [8].

DOI: 10.21136/panm.2024.11

This study focuses on low Mach number laminar air flow over a square cylinder, a classical problem in fluid mechanics with practical relevance to building design. The interaction between the flow and the body leads to vortex shedding, forming a von Kármán vortex street that generates acoustic waves. The investigation of acoustic emissions is conducted using a combination of two open-source tools, OpenFOAM and FEniCS. The integration of these tools is detailed in Chapter 5.

## 2. The mathematical model

We consider the conservation of mass equation and the conservation of momentum equation given by

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho\, \mathbf{u}) = 0\,, \tag{1}$$

$$\frac{\partial (\rho\, \mathbf{u})}{\partial t} + \nabla \cdot (\rho\, \mathbf{u} \otimes \mathbf{u} - \boldsymbol{\sigma}) = 0\,, \tag{2}$$

where $\mathbf{u}$ denotes the fluid velocity, $\rho$ fluid density and $t$ time. For fluid, the stress tensor $\boldsymbol{\sigma}$ is defined as

$$\boldsymbol{\sigma} = -p\, \mathbf{I} + \boldsymbol{\tau}\,, \tag{3}$$

where $p$ is static pressure, $\boldsymbol{\tau}$ denotes the viscous (shear) stress tensor and $\mathbf{I}$ is the unit tensor. Since air is a Newtonian fluid, the constitutive relation between the viscous stress tensor and the rate of strain tensor is expressed as

$$\boldsymbol{\tau} = \mu \left( \nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) - \frac{2}{3} \mu \nabla \cdot \mathbf{u}\,, \tag{4}$$

where $\mu$ is dynamic viscosity of the fluid. At low Mach numbers, the fluid is assumed to be nearly incompressible, implying that the density remains constant and the velocity field is divergence-free.

In order to obtain unique solution for eq. (1) and (2) we have to consider bounded domain $\Omega_1 \subset \mathbb{R}^2$ with boundary conditions defined on Lipschitz boundary $\partial\Omega_1$. The boundary $\partial\Omega_1$ is further subdivided as $\partial\Omega_1 = \Gamma_\mathrm{b} \cup \Gamma_1$ and $\Gamma_1 = \Gamma_\mathrm{in} \cup \Gamma_\mathrm{out} \cup \Gamma_\mathrm{slip}$, as illustrated in Fig. 1.

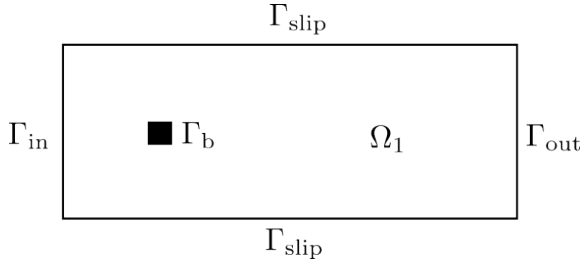



Figure 1: Fluid computational domain.

The initial-boundary value problem for incompressible fluid is then formulated as: for $t \in (0,T]$ find $\mathbf{u}(\mathbf{x},t) : \Omega_1 \times (0,T] \to \mathbb{R}^2$ and $p(\mathbf{x},t) : \Omega_1 \times (0,T] \to \mathbb{R}$ such that

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) - \nabla \cdot (\nu \nabla \mathbf{u}) + \frac{1}{\rho_0} \nabla p = \mathbf{0} \quad & \text{in } \Omega_1 \times (0,T] \,, \\ \nabla \cdot \mathbf{u} = 0 \quad & \text{in } \Omega_1 \times (0,T] \,, \end{aligned} \tag{5}$$

where $\nu$ is the kinematic viscosity (dynamic viscosity divided by density) of the fluid and $\rho_0$ is the freestream density. The boundary and initial conditions are prescribed as follows

$$\begin{aligned} \mathbf{u} &= \mathbf{0} && \text{on } \Gamma_\mathrm{b} && \times (0,T] \,, \\ \mathbf{u} &= (U_\infty, 0) && \text{on } \Gamma_\mathrm{in} && \times (0,T] \,, \\ -\nu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} + \frac{p}{\rho_0} \mathbf{n} &= \mathbf{0} && \text{on } \Gamma_\mathrm{out} && \times (0,T] \,, \\ \mathbf{u} \cdot \mathbf{n} &= 0 && \text{on } \Gamma_\mathrm{slip} && \times (0,T] \,, \\ \frac{\partial p}{\partial \mathbf{n}} &= 0 && \text{on } \Gamma_1 \setminus \Gamma_\mathrm{out} \cup \Gamma_\mathrm{b} && \times (0,T] \,, \\ \mathbf{u}(\mathbf{x},0) &= (U_\infty, 0) && \text{for } \mathbf{x} \in \Omega_1 \,, \end{aligned} \tag{6}$$

where $U_\infty$ is the freestream velocity and $\mathbf{n}$ is the outward unit vector to $\Gamma_\mathrm{b}$ and $\Gamma_1$.

**Aeroacoustics**

The most widely used CAA formulation is Lighthill's aeroacoustic analogy, where the governing equations (1) and (2) are reformulated into a wave-like equation, as detailed in [5]. In this approach, acoustic noise is radiated from a localized region of fluctuating flow embedded within an infinite homogeneous fluid, see Fig. 2. In the surrounding fluid, the speed of sound $c_0$, the density $\rho_0$ and the pressure $p_0$ are constants and the density fluctuations $\rho' = \rho - \rho_0$ are governed by the standard homogeneous acoustic wave equation. Within the fluctuating region, the Lighthill's aeroacoustic equation is derived by taking the time derivate of the continuity equation (1) and subtracting the divergence of the momentum equation (2), which yields

$$\frac{\partial^2 (\rho - \rho_0)}{\partial t^2} = \nabla \cdot \nabla \cdot \left[ \rho \, \mathbf{u} \otimes \mathbf{u} + (p - p_0) \, \mathbf{I} - \boldsymbol{\tau} \right] , \tag{7}$$

where $p' = p - p_0$ are the pressure perturbations. Further substracting the term $c_0^2 \Delta (\rho - \rho_0)$ from both sides of eq. (7), we retrieve the desired inhomogeneous wave equation

$$\left( \frac{\partial^2}{\partial t^2} - c_0^2 \Delta \right) (\rho - \rho_0) = \nabla \cdot \nabla \cdot \mathbf{T} \,, \tag{8}$$

where the Lighthill's tensor $\mathbf{T}$ is introduced as

$$\mathbf{T} = \rho \, \mathbf{u} \otimes \mathbf{u} + \left[ (p - p_0) - c_0^2 (\rho - \rho_0) \right] \mathbf{I} - \boldsymbol{\tau} \,. \tag{9}$$

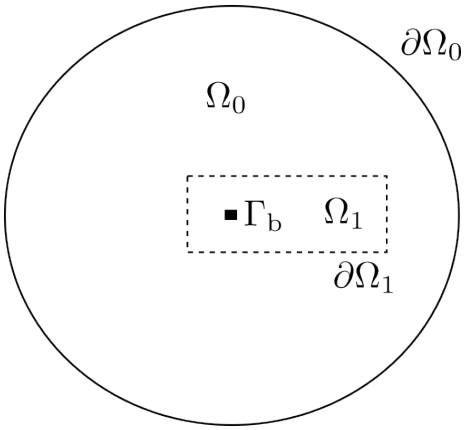



Figure 2: Aeroacoustic computational domain.

The Lighthill's tensor in eq. (9) consists of three terms. The viscous source term $\boldsymbol{\tau}$ is significant only at low Reynolds numbers and over sufficiently long distances, so it is often neglected. Additionally, for low Mach numbers flows and no heat effects, the fluid can be considered isentropic, meaning the relation $p' = c_0^2 \rho'$ holds and density can be approximated by the density of the resting media $\rho_0$. Under these conditions, the Lighthill's tensor reduces to $\mathbf{T} \approx \rho_0 \, \mathbf{u} \otimes \mathbf{u}$ and eq. (8) results in the following inhomogeneous wave equation

$$\frac{1}{c_0^2} \frac{\partial^2 p'}{\partial t^2} - \Delta p' = \nabla \cdot \nabla \cdot (\rho_0 \, \mathbf{u} \otimes \mathbf{u}) \ . \tag{10}$$

In order to solve eq. (10) we consider the homogeneous fluid region to be finite and bounded. We denote the aeroacoustic computational domain as $\Omega_0 \subset \mathbb{R}^2$ with Lipschitz boundary $\partial\Omega_0$, such that $\Omega_1 \subset \Omega_0$, see Fig. 2. The boundary $\partial\Omega_0$ is further subdivided as $\partial\Omega_0 = \Gamma_\mathrm{b} \cup \Gamma_0$. The initial-boundary value problem then reads as: for $t \in (0, T]$ find $p'(\mathbf{x}, t) \colon \Omega_0 \times (0, T] \to \mathbb{R}$ such that

$$\frac{1}{c_0^2} \frac{\partial^2 p'}{\partial t^2} - \Delta p' = \begin{cases} \nabla \cdot \nabla \cdot (\rho_0 \, \mathbf{u} \otimes \mathbf{u}) & \text{in } \Omega_1 \times (0, T] \ , \\ 0 & \text{in } \Omega_0 \setminus \Omega_1 \times (0, T] \ , \end{cases} \tag{11}$$

and which satisfies the following boundary and initial conditions

$$\begin{aligned} \frac{\partial p'}{\partial \mathbf{n}} &= 0 && \text{on } \Gamma_\mathrm{b} \times (0, T] \, , \\ \frac{\partial p'}{\partial \mathbf{n}} &= -\frac{1}{c_0} \frac{\partial p'}{\partial t} && \text{on } \Gamma_0 \times (0, T] \ , \\ p'(\mathbf{x}, 0) &= 0 && \text{for } \mathbf{x} \in \Omega_0 \, , \\ \frac{\partial p'}{\partial t}(\mathbf{x}, 0) &= 0 && \text{for } \mathbf{x} \in \Omega_0 \, , \end{aligned} \tag{12}$$

where $\mathbf{n}$ is the outward unit vector to $\Gamma_\mathrm{b}$ and $\Gamma_0$.

The acoustic sources $\nabla \cdot \nabla \cdot (\rho_0 \, \mathbf{u} \otimes \mathbf{u})$ within the near-field domain $\Omega_1$ are evaluated using the fluid velocity $\mathbf{u}$ obtained from the Navier-Stokes equations for incompressible fluid, see eq. (5). Non-reflective boundary condition is prescribed in eq. (12) on the boundary $\Gamma_0$ to mitigate acoustic reflections.

## 3. Finite volume discretization

The discretization of eq. (5) involves subdivision of the domain $\Omega_1$ into a finite number of closed, non-overlapping polygonal cells $V_k$ (with volume $|V_k|$), such that $\Omega_1 = \bigcup_{k \in \mathcal{J}} V_k$, where $\mathcal{J}$ is an index set. Integrating eq. (5) over an arbitrary polygon $V_k$ yields

$$\begin{aligned} \int_{V_k} \frac{\partial \mathbf{u}}{\partial t} \, \mathrm{d}V + \int_{V_k} \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) \, \mathrm{d}V - \int_{V_k} \nabla \cdot (\nu \nabla \mathbf{u}) \, \mathrm{d}V + \int_{V_k} \frac{1}{\rho_0} \nabla p \, \mathrm{d}V = \mathbf{0} \,, \\ \int_{V_k} \nabla \cdot \mathbf{u} \, \mathrm{d}V = 0 \,. \end{aligned} \tag{13}$$

The solution of (13) is approximated by piecewise constant functions $\mathbf{u}_k, p_k$ given as

$$\mathbf{u}_k \approx \frac{1}{|V_k|} \int_{V_k} \mathbf{u} \, \mathrm{d}V \,, \quad p_k \approx \frac{1}{|V_k|} \int_{V_k} p \, \mathrm{d}V \,.$$

Considering $V_k$ remains constant over time, the time derivative of velocity in eq. (13) can be cast in form $\int_{V_k} \partial \mathbf{u}/\partial t \, \mathrm{d}V \approx |V_k| \, \mathrm{d}\mathbf{u}_k/\mathrm{d}t$. For the time discretization, we first divide the temporal interval $(0, T]$ into $N$ subintervals, such that $T = N\Delta t$, setting $t^n = n\Delta t$, with $n = 0, \ldots, N$, where $\Delta t$ denotes constant time step. The Crank-Nicolson scheme is used for the temporal dicretization in the form

$$\frac{\mathrm{d}\mathbf{u}_k}{\mathrm{d}t} \approx \left[ \frac{1}{1 + c_{\mathrm{oc}}} \left( \frac{\mathbf{u}_k^{n+1} - \mathbf{u}_k^n}{\Delta t} \right) - \frac{c_{\mathrm{oc}}}{1 + c_{\mathrm{oc}}} \left( \frac{\mathbf{u}_k^n - \mathbf{u}_k^{n-1}}{\Delta t} \right) \right], \tag{14}$$

where $c_{\mathrm{oc}}$ is an off-centering coefficient, see [6]. For $c_{\mathrm{oc}} = 0$ the scheme results in the implicit Euler scheme, whereas for $c_{\mathrm{oc}} = 1$ the central scheme is obtained. In the following work, $c_{\mathrm{oc}} = 0.9$ is used.

For other terms in eq. (13), we employ Gauss's theorem and approximate them using the midpoint quadrature rule on the face $f \in S_k$, where $S_k$ is the set of all faces of the cell $V_k$ and $|S_f|$ denotes the surface of face $f$, as

$$\int_{V_k} \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) \, \mathrm{d}V = \oint_{\partial V_k} \mathbf{u} \, (\mathbf{u} \cdot \mathbf{n}) \, \mathrm{d}S \approx \sum_{f \in S_k} \mathbf{u}_f \, (\mathbf{u}_f \cdot \mathbf{s}_f) = \sum_{f \in S_k} \mathbf{u}_f \, \phi_f \,, \tag{15}$$

$$\int_{V_k} \nu \Delta \mathbf{u} \, \mathrm{d}V = \oint_{\partial V_k} \nu \, (\nabla \mathbf{u}) \cdot \mathbf{n} \, \mathrm{d}S \approx \nu \sum_{f \in S_k} (\nabla \mathbf{u})_f \cdot \mathbf{s}_f \,, \tag{16}$$

$$\int_{V_k} \frac{1}{\rho_0} \nabla p \, \mathrm{d}V = \oint_{\partial V_k} \frac{1}{\rho_0} (p \, \mathbf{I}) \cdot \mathbf{n} \, \mathrm{d}S \approx \frac{1}{\rho_0} \sum_{f \in S_k} (p_f \, \mathbf{I}) \cdot \mathbf{s}_f \,, \tag{17}$$

where $\mathbf{s}_f = \mathbf{n}\,|S_f|$ and $\phi_f = \mathbf{u}_f \cdot \mathbf{s}_f$ represents the volumetric flux at face $f$. The continuity equation in (13) enforces the sum of fluxes across all faces to be zero, i.e. $\sum_{f \in S_k} \phi_f = 0$ in order to satisfy the divergence-free condition. Concerning the discretization of the fluxes in (15)–(17) and the gradient reconstruction of velocity in eq. (16) we refer to [6].

For solving the discretized NSE for incompressible fluid, the PIMPLE algorithm is used as a combination of SIMPLE (semi-implicit method for pressure-linked equations) and PISO (pressure-implicit algorithm with the splitting of the operator). We begin by discretizing the momentum equation (13). Let $a_C^{\mathbf{u}}$ and $a_N^{\mathbf{u}}$ represent the coefficients in the resulting algebraic equations, where $C$ and $N$ refer to the central and neighboring cells, respectively. The discretized momentum equation then reads

$$a_C^{\mathbf{u}}\mathbf{u}_C + \sum_{f \in S_k} a_N^{\mathbf{u}}\mathbf{u}_N = \mathbf{r} - \frac{1}{\rho_0}(\nabla p)_C\,, \tag{18}$$

where vector $\mathbf{r}$ represents contributions from previous time steps. Next we introduce operator $\mathbf{H}(\mathbf{u}) = \mathbf{r} - \sum_{f \in S_k} a_N^{\mathbf{u}}\mathbf{u}_N$ such that

$$\mathbf{u}_C = (a_C^{\mathbf{u}})^{-1}\left[\mathbf{H}(\mathbf{u}) - \frac{1}{\rho_0}(\nabla p)_C\right]\,. \tag{19}$$

We substitute eq. (19) into the continuity equation to obtain a pressure equation

$$\nabla \cdot \left[(a_C^{\mathbf{u}})^{-1}(\nabla p)_C\right] = \rho_0 \nabla \cdot \left[(a_C^{\mathbf{u}})^{-1}\mathbf{H}(\mathbf{u})\right]\,. \tag{20}$$

The PIMPLE algorithm is based on a predictor and corrector step. In the predictor step, we solve eq. (18) using an intermediate pressure to obtain predicted velocity, which does not yet satisfy the continuity equation. We follow with the corrector step, in which we solve eq. (20) to obtain corrected pressure, and subsequently divergence-free velocity is obtained from eq. (19). We repeat these inner and outer loops until the pressure and velocity fields converge, see [6] for further reference. Additionally, under-relaxation can be used in each time step to smooth convergence.

## 4. Finite element discretization

In order to approximate the inhomogeneous wave equation (11) using FEM, the equation is multiplied by a test function $w \in \mathcal{V} \subset H^1(\Omega_0)$ and integrated over the entire acoustic domain $\Omega_0$. This yields

$$\frac{1}{c_0^2}\left(\frac{\partial^2 p'}{\partial t^2}, w\right)_{\Omega_0} - (\Delta p', w)_{\Omega_0} = (\nabla \cdot \nabla \cdot (\rho_0\,\mathbf{u} \otimes \mathbf{u})\,, w)_{\Omega_0}\,, \tag{21}$$

where by $(\cdot,\cdot)_D$ the dot product in $L_2(D)$ is denoted. After applying Green's integration theorem to the second spatial derivate of $p'$ as well as to the acoustic source

term on the right-hand side, the eq. (21) can be rearranged into

$$\begin{aligned}\frac{1}{c_0^2}\left(\frac{\partial^2 p'}{\partial t^2}, w\right)_{\Omega_0} - \left(\frac{\partial p'}{\partial \mathbf{n}}, w\right)_{\partial\Omega_0} + (\nabla p', \nabla w)_{\Omega_0} \\ = (\nabla\cdot(\rho_0\,\mathbf{u}\otimes\mathbf{u})\cdot\mathbf{n}, w)_{\partial\Omega_1} - (\nabla\cdot(\rho_0\,\mathbf{u}\otimes\mathbf{u}), \nabla w)_{\Omega_1}\,.\end{aligned} \tag{22}$$

Boundary conditions are applied to each boundary term. For the source term in eq. (22), this leads to the condition $(\nabla\cdot(\rho_0\,\mathbf{u}\otimes\mathbf{u})\cdot\mathbf{n}, w)_{\partial\Omega_1} = 0$ since

$$\begin{aligned}(\nabla\cdot(\rho_0\,\mathbf{u}\otimes\mathbf{u})\cdot\mathbf{n}, w)_{\Gamma_1} &= 0\,,\\ (\nabla\cdot(\rho_0\,\mathbf{u}\otimes\mathbf{u})\cdot\mathbf{n}, w)_{\Gamma_b} &= 0\,,\end{aligned} \tag{23}$$

for details, see [2]. This leads to the variational (weak) formulation of Lighthill's aeroacoustic equation, which may be stated as: find $p' \in \mathcal{V}$ such that

$$\frac{1}{c_0^2}\left(\frac{\partial^2 p'}{\partial t^2}, w\right)_{\Omega_0} + \frac{1}{c_0}\left(\frac{\partial p'}{\partial t}, w\right)_{\Gamma_0} + \left(\frac{\partial p'}{\partial x_i}, \frac{\partial w}{\partial x_i}\right)_{\Omega_0} = -(\nabla\cdot(\rho_0\,\mathbf{u}\otimes\mathbf{u}), \nabla w)_{\Omega_1}\,, \tag{24}$$

is fulfilled for all $w \in \mathcal{V}$. The source term in eq. (24) can be further simplified for incompressible fluid flows as follows

$$(\nabla\cdot(\rho_0\,\mathbf{u}\otimes\mathbf{u}), \nabla w)_{\Omega_1} = \rho_0\,(\mathbf{u}\cdot\nabla\mathbf{u}, \nabla w)_{\Omega_1}\,. \tag{25}$$

The semi-discrete Galerkin formulation is obtained from the weak formulation (24) after discretization of the domain and the introduction of finite element spaces. A finite-dimensional finite element space $V_h \subset \mathcal{V}$ with dimension $n$ is chosen and the solution $p' \in \mathcal{V}$ is approximated by $p_h \in V_h$ written as a time-dependant linear combination of coefficients $p_j(t)$ and basis functions $\varphi_j(x) \in V_h$, i.e.

$$p'(x,t) \approx p_h(t,x) = \sum_{j=1}^{n} p_j(t)\varphi_j(x)\,. \tag{26}$$

Using relation (26) in eq. (24) with $w_h = \varphi_i$ for $i = 1, \ldots, n$ leads to the second-order system of ODEs for an unknown vector $\mathbf{p}(t) = \{p_j\}_{j=1}^n$ in the matrix form

$$\frac{1}{c_0^2}\mathbb{M}\ddot{\mathbf{p}}(t) + \frac{1}{c_0}\mathbb{D}\dot{\mathbf{p}}(t) + \mathbb{K}\mathbf{p}(t) = \mathbf{b}(t)\,, \tag{27}$$

where the matrices $\mathbb{M} = \{m_{ij}\}_{i,j=1}^n, \mathbb{D} = \{d_{ij}\}_{i,j=1}^n$ $\mathbb{K} = \{k_{ij}\}_{i,j=1}^n$ and the vector $\mathbf{b} = \{b_i\}_{i=1}^n$ are computed as follows

$$m_{ij} = (\varphi_j, \varphi_i)_{\Omega_0}\,, \quad d_{ij} = (\varphi_j, \varphi_i)_{\Gamma_0}\,, \quad k_{ij} = \left(\frac{\partial\varphi_j}{\partial x_l}, \frac{\partial\varphi_i}{\partial x_l}\right)_{\Omega_0}\,,$$

$$b_i = -\rho_0\left(u_l\frac{\partial u_j}{\partial x_l}, \frac{\partial\varphi_i}{\partial x_j}\right)_{\Omega_1}\,.$$

The problem described in eq. (27) is discretized in time with the aid of the Newmark method. This method is formally realized by using approximations

$$\begin{aligned} \mathbf{p}_{n+1} &= \mathbf{p}_n + \dot{\mathbf{p}}_n \, \Delta t + ((1-2\beta)\ddot{\mathbf{p}}_n + 2\beta\ddot{\mathbf{p}}_{n+1}) \, \frac{\Delta t^2}{2} \,, \\ \dot{\mathbf{p}}_{n+1} &= \dot{\mathbf{p}}_n + ((1-\gamma)\ddot{\mathbf{p}}_n + \gamma\ddot{\mathbf{p}}_{n+1}) \, \Delta t \,, \end{aligned} \tag{28}$$

in eq. (27), which is solved for $\ddot{\mathbf{p}}_{n+1}$. Values $\beta$ and $\gamma$ are taken as $\beta = 0.25$, $\gamma = 0.5$.

## 5. Implementation

The finite volume approach available within the OpenFOAM library has been adopted for space and time discretization of the NSE for incompressible fluid. Tab. 1 briefly describes the basic directory structure for OpenFOAM case that is required to run the simulation. The exception is the acousticMesh and funcObjects folders, which include the neccessary data for evaluation of the acoustic sources within the fluid domain and their subsequent interpolation onto the acoustic domain.

| | |
|---|---|
| 0 | → Initial and boundary conditions for fields |
| ⌞ U | |
| ⌞ p | |
| constant | |
| ⌞ transportProperties | → Physical properties of the fluid |
| ⌞ turbulenceProperties | → Type of fluid flow |
| ⌞ polyMesh | → Mesh data for the fluid domain |
| ⌞ acousticMesh | → Mesh data for the acoustic domain |
| system | |
| ⌞ controlDict | → Simulation's control parameters |
| ⌞ fvSchemes | → Numerical schemes used for discretizing |
| ⌞ fvSolution | → Solver settings and relaxation factors |
| ⌞ funcObjects | → Custom functions to be applied during |
| ⌞ ... | simulation |

Table 1: OpenFOAM folder structure.

Once the fluid simulation is finished and the acoustic sources have been interpolated for the desired time period, the Lighthill's aeroacoustic equation is solved using the FEM framework implemented in the FEniCS Python library (see [1]).

After importing the acoustic mesh in *.msh* format, a finite element function space V is created. The trial function p and test function w are then defined, followed by the initialization of fem.Function, which stores the coefficients for the solution.

```
from dolfinx import fem, import ufl
V = fem.functionspace(msh, ("Lagrange", 1))
p, w = ufl.TrialFunction(V), ufl.TestFunction(V)
p_h = fem.Function(V)
```

Similarly, we initalize fem.Function for the divergence of Lighthill's tensor.

```
V_divT = fem.functionspace(msh, ("Lagrange", 1, (msh.geometry.dim,)))
divT = fem.Function(V_divT)
```

We also prepare data structures for the Newmark method.

```
p_0, pdot_0, pddot_0 = fem.Function(V), fem.Function(V), fem.Function(V)
pddot = p
pdot = pdot_0 + ((1-gamma)*pddot_0 + gamma*pddot) * dt
p_ = p_0 + pdot_0*dt + ((1-2*beta)*pddot_0 + 2*beta*pddot) * dt**2/2
```

The integration measures are defined to substitute for the different subdomain cells and their boundary faces.

```
ds = ufl.Measure("ds", domain=msh, subdomain_data=boundary_tags)
dx = ufl.Measure("dx", domain=msh, subdomain_data=subdomain_tags)
```

With all the data structures in place, we define the variational formulation.

```
F = 1 / c_0**2 * ufl.inner(pddot, w) * dx(0) \
    + 1 / c_0 * ufl.inner(pdot, w) * ds(0) \
    + ufl.inner(ufl.grad(p_), ufl.grad(w)) * dx(0) \
    + ufl.inner(divT, ufl.grad(w)) * dx(1)
a, L = ufl.system(F)
```

Using the finite element variational problem formulation, the class dolfinx.fem.petsc.LinearProblem is created for solution of the variational problem. This class utilizes PETSc as the linear algebra backend and a direct solver (LU-factorization) is employed to solve the linear system.

```
import dolfinx.fem.petsc
problem = dolfinx.fem.petsc.LinearProblem(a, L, u=p_h, bcs=[], petsc_options)
```

Finally, the problem is solved repeatedly in time in order to obtain the evolution of the acoustic pressure field.

```
t = t_start
while t < t_end + dt:
    divT = get_interpolated_OpenFOAM_field(divT, msh, t)
    p_h = problem.solve()
    p_0, pdot_0, pddot_0 = evaluate_Newmark_fields(p_h, p_0, pdot_0, pddot_0)
    write_results(p_0, t)
    t += dt
```

## 6. Numerical results

A laminar, two-dimensional simulation of an incompressible fluid flow over a square cylinder is performed. When a rigid square cylinder is immersed in a uniform flow, it generates strong vortex shedding. The resulting fluctuating forces on the cylinder induce acoustic waves, which are the focus of this study.

By $L_{\text{cyl}} = 3.28 \cdot 10^{-5}$ m we denote the dimension of the cylinder. The dimensions of the fluid computational domain $\Omega_1$ are then (-$30L_{\text{cyl}}$, $100L_{\text{cyl}}$) $\times$ ($-25L_{\text{cyl}}$, $25L_{\text{cyl}}$), with a blockage ratio[1] of $\beta = L_{\text{cyl}}/50L_{\text{cyl}}$. The acoustic domain $\Omega_0$ is a circle with a radius of $150L_{\text{cyl}}$. The mesh within the fluid domain is roughly three times finer than the mesh for the acoustic simulation, as can be seen in Fig. 3. The flow properties and the setup of the simulation are listed in Tab 2. The fluid flow

[1]The ratio of the square cylinder's frontal area to the domain's cross-sectional area in the flow direction.

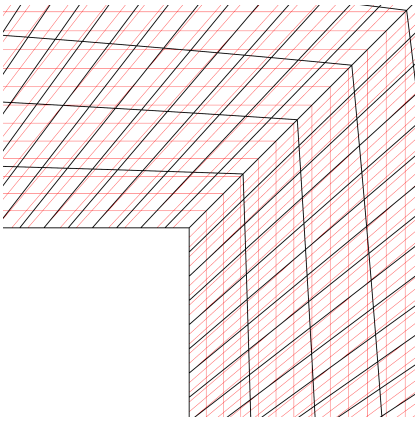


Figure 3: Acoustic (black) and fluid (red) mesh near cylinder.

| Setup of the simulation | | |
|---|---|---|
| $U_\infty$ | = | 68.7 m $s^{-1}$ |
| $L_{\text{cyl}}$ | = | $3.28 \cdot 10^{-5}$ m |
| $\nu$ | = | $1.5 \cdot 10^{-5}$ $m^2 s^{-1}$ |
| $c_0$ | = | 343 m $s^{-1}$ |
| $\rho_0$ | = | 1.2 kg $m^{-3}$ |
| Re | = | 150 |
| Ma | = | 0.2 |

Table 2: Setup of the simulation.

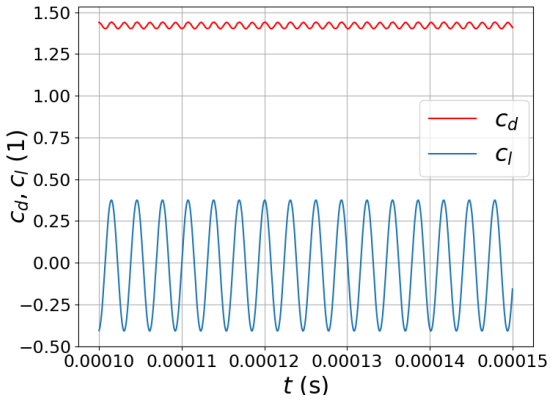



Figure 4: Lift and drag coefficients in time.

| | $\overline{c_D}$ | St |
|---|---|---|
| present study | 1.42 | 0.153 |
| Doolan [3] | 1.44 | 0.156 |

Table 3: Comparison of the mean drag coeffcient and Strouhal number with reference values.

solution is sampled after the full vortex street developed from $t_0^{\text{a}} = 1 \cdot 10^{-4}$ s every 10 fluid time steps, i.e. $\Delta t^{\text{a}} = 10^{-8}$ s and $\Delta t^{\text{f}} = 10^{-9}$ s, until the end of simulation $t_{\text{end}} = 1.5 \cdot 10^{-4}$ s. The whole time of acoustic simulation is $0.5 \cdot 10^{-4}$ s. The lift and drag coefficients are plotted in time in Fig. 4. The mean drag coefficient $\overline{c_D}$ and Strouhal number St based on vortex shedding frequency are evaluated and compared to reference values with good agreement, see Tab. 3. The acoustic pressure values are monitored at three different observer locations. The first two observers are positioned downstream along the $x$-axis and along the fringe of the cylinder, respectively. Significantly lower values are anticipated at the third observer location in the far-field region, see Tab. 4. Fig. 5 presents the acoustic field (scaled by dynamic pressure) at final time, in which dipole pattern can be seen. The acoustic pressure values at three observer locations are shown in Fig. 6 in the time and frequency domain. The main frequency component for observer 2 and 3 corresponds to the vortex shedding frequency. On the other hand, the main frequency component for observer 1, located along the $x$-axis is twice as high. This fact can be associated with the combination of the upper and lower vortices.

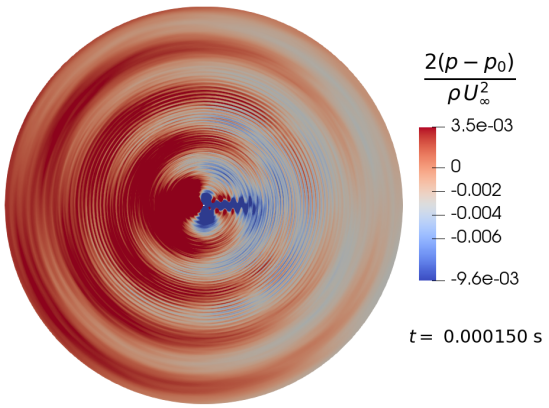



Figure 5: Acoustic pressure field.

| Observers | |
|---|---|
| 1: | $(1 \cdot 10^{-4}, 0, 0)$ m |
| 2: | $(1 \cdot 10^{-4}, 2 \cdot 10^{-5}, 0)$ m |
| 3: | (0.002, -0.003, 0) m |

Table 4: Positions of the observers with respect to the origin of the coordinate system located in the center of the square cylinder.

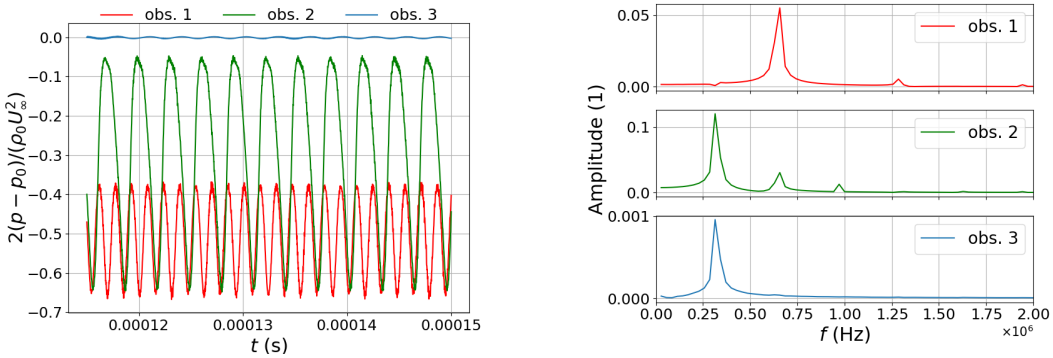



Figure 6: Acoustic pressure at three observer locations in: (a) the time domain and (b) the frequency domain.

## 7. Conclusion

In this study we have adopted a hybrid method for CAA that establishes a foundation for future aeroacoustic investigations. Our primary focus was on a 2D square cylinder placed within a laminar flow with Reynolds number 150 and Mach number 0.2. The presence of the square cylinder resulted in the formation of strong vortices in the downstream region, which induced acoustic waves. The resulting acoustic pressure obtained by Lighthill's aeroacoustic analogy was analyzed both in the near-field and far-field acoutic region. The dominant frequencies for selected observers correspond with expectations.

## Acknowledgements

This work was supported by the Czech Technical University in Prague under the grant No. SGS24/120/OHK2/3T/12 and grant No. SGS22/148/OHK2/3T/12. The authors also gratefully acknowledge the Center of Advanced Aerospace Technology (CZ.02.1.01/0.0/0.0/16_019/0000826) at the Czech Technical University in Prague for awarding the access to computing facilities.

## References


[1] Baratta, I. A. et al.: DOLFINx: The next generation FEniCS problem solving environment, 2023.

[2] Caro, S., Ploumhans, P., and Gallez, X.: Implementation of lighthill's acoustic analogy in a finite/infinite elements framework. 10th AIAA/CEAS Aeroacoustics Conference, (2004).

[3] Doolan, C. J.: Flat-plate interaction with the near wake of a square cylinder. AIAA Journal **47** (2009), 475–479.

[4] Ewert, R. and Schröder, W.: Acoustic perturbation equations based on flow decomposition via source filtering. J. Comput. Phys. **188** (2003), 365–398.

[5] Lighthill, M. J.: On sound generated aerodynamically. i. general theory. Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences **211** (1952), 564–587.

[6] Moukalled, F., Mangani, L., and Darwish, M.: *The finite volume method in computational fluid dynamics: an advanced introduction with OpenFOAM and Matlab*. Springer International Publishing, Cham, 2016.

[7] Schoder, S. and Kaltenbacher, M.: Hybrid aeroacoustic computations: State of art and new achievements. J. Theor. Comput. Acoust. **27** (2019).

[8] Valášek, J. and Sváček, P.: Aeroacoustic simulation of human phonation based on the flow-induced vocal fold vibrations including their contact. Adv. Eng. Softw. **194** (2024).

*Programs and Algorithms of Numerical Mathematics 22*
J. Chleboun, J. Papež, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics CAS, Prague 2025

# A NOTE ON THE OD-QSSA AND BOHL–MAREK METHODS APPLIED TO A CLASS OF MATHEMATICAL 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 00 Prague 8, Czech Republic
matonoha@cs.cas.cz

**Abstract:** The complex (bio)chemical reaction systems, frequently possess fast/slow phenomena, represent both difficulties and challenges for numerical simulation. We develop and test an enhancement of the classical QSSA (quasi-steady-state approximation) model reduction method applied to a system of chemical reactions. The novel model reduction method, the so-called delayed quasi-steady-state approximation method, proposed by Vejchodský (2014) and further developed by Papáček (2021) and Matonoha (2022), is extensively presented on a case study based on Michaelis–Menten enzymatic reaction completed with the substrate transport. Eventually, an innovative approach called the Bohl–Marek method is shown on the same numerical example.

**Keywords:** mathematical modelling, chemical kinetic systems, model reduction, quasi-steady-state approximation, M-Matrix, quasi-linear formulation

**MSC:** 92C45, 34A34, 65F60, 65K10

## 1. Introduction

Since Briggs and Haldane's application of the quasi-steady-state (QSS) assumption, e.g., [11], the idea of reducing complex chemical networks persists in the field of large-scale (bio)chemical systems modeling, see [12] and references therein. On the other side of control theory (cooperative biochemical systems), there are inspiring works of Bohl and Marek [1, 2, 8].

This study presents the development and application of one special model order reduction technique further called the delayed quasi-steady-state approximation method (D-QSSA), first proposed by Vejchodský in 2014 [13, 14] and further developed by our group in [9, 10]. We continue in the direction of papers devoted to the analysis of fast/slow phenomena arising in biology and chemistry, and more

DOI: 10.21136/panm.2024.12

precisely to the problem of parameter estimation for mathematical models describing the drug-induced enzyme production networks [3] aiming to develop biologically meaningful models, which can be used for drug delivery analysis and optimization. Although our ultimate goal is to develop a reliable method for fitting the model parameters of large biochemical networks to given experimental data, here we study certain numerical issues within the framework of efficient computations of inverse problems involving numerical optimization.

The paper is organized as follows. In Section 2, different numerical methods are presented. Then, in Section 3, we employ an illustrative case study to comprehensively account the pros and cons of each of the analyzed techniques. Section 4 concludes the work and outlines the future work. Finally, Appendix A presents a straightforward method for setting up the governing ODE system, while Appendix B provides the reformulation of nonlinear ODEs to the quasi-linear form.

## 2. Model and methods

This section further introduces the necessary theoretical background and notations used throughout this study, concerning mainly the fast/slow dynamical systems [15] and singular perturbation methods (SPM) with delays [6]. Let us consider the following system of ordinary differential equations (ODE) representing a general class of mathematical models describing (bio)chemical systems

$$\dot{x}(t) = Ax(t) + b(t, x(t)), \tag{1}$$

for $t \in [0, T]$ with $T > 0$, where $x(t) \in \mathcal{R}^n$, constant matrix $A \in \mathcal{R}^{n \times n}$ represents a linear part of the system, and $b(t, x(t)) \in \mathcal{R}^n$ contains nonlinear, time-varying and constant parts of the system. The ODE system (1) is further completed by suitable initial conditions, such that $x(0) = x_0$, defining the initial value problem (IVP). In the following subsections, we introduce the so-called optimal delayed quasi-steady-state approximation method (OD-QSSA) and an innovative approach here called the Bohl–Marek (BM) method.

### 2.1. Order reduction methods for the fast/slow dynamical systems

Suppose the existence of the fast and slow variables $x_F \in \mathcal{R}^{n_F}$ and $x_S \in \mathcal{R}^{n_S}$ and let $x(t) = \begin{pmatrix} x_F^T(t) & x_S^T(t) \end{pmatrix}^T$ be the partitioning of $x(t)$, where $n_F + n_S = n$. Then for a general fast/slow ODE system it holds

$$\begin{array}{rcl} \varepsilon\, \dot{x}_F & = & f_F(x_S, x_F; \varepsilon), \\ \dot{x}_S & = & f_S(x_S, x_F; \varepsilon), \end{array} \tag{2}$$

when $0 < \varepsilon \ll 1$, and suitable initial conditions are set. Then, the ODE system (2) can be approximated by a simpler algebro-differential system (an associated slow subsystem)

$$\begin{array}{rcl} 0 & = & f_F(x_S, x_F; 0), \\ \dot{x}_S & = & f_S(x_S, x_F; 0). \end{array} \tag{3}$$

Equations (3) are called singularly perturbed in the singular perturbation theory, whereas, in the chemical literature, such a model reduction is called a (standard) quasi-steady-state approximation (QSSA) when the underlying assumption ($0 < \varepsilon \ll 1$) assuring small approximation error, i.e., the validity of the standard QSSA is often referred to as the reactant-stationary assumption [4]. Several mathematical studies are dedicated to quantifying the accuracy of different QSSA methods applied to enzyme kinetics. Identification of a presumably small parameter $\varepsilon$, see (2), is common to these efforts, which quantifies the timescale separation. This explicit identification of a suitable $\varepsilon$ for every system and operating condition requires nontrivial mathematical operations. Consequently, when one tries to omit such analysis, the non-justified use of the QSSA method frequently occurs, which in fact represents the QSSA method's abuse [5].

Our solution to the difficulties mentioned above dwells in the relatively novel extension of the D-QSSA method, being the delayed QSSA with the optimal constant delay introduced by Matonoha et al. [9] for a class of chemical networks with the mass conservation property and a wide timescale separation.

For completeness, we provide the main theorem concerning the existence of an optimal constant delay. The proof and detailed description can be found in [9].

**Theorem 1.** *Let $\bar{x}(t)$ be a solution of the (full) system (2). Choose arbitrary numbers $0 < \underline{\tau} \le \overline{\tau} < T$ and a fixed constant delay $\tau \in [\underline{\tau}, \overline{\tau}]$. Let $x_F^{cdqss}(t,\tau)$ be a constant delay QSS approximation of $x_F(t)$ with this $\tau$. Let $x_S^{cdqss}(t,\tau)$ be a solution of the reduced delayed ODE system, continuous for $t \in [0,T]$. Denote $x^{cdqss}(t,\tau) = \left( \begin{array}{cc} x_F^{cdqss^T}(t,\tau) & x_S^{cdqss^T}(t,\tau) \end{array} \right)^T$. Then there exists at least one value $\tau^* \in [\underline{\tau}, \overline{\tau}]$ minimizing the error between $\bar{x}(t)$ and $x^{cdqss}(t,\tau)$, i.e.,*

$$\tau^* = \arg\min_{\tau} \|\bar{x}(t) - x^{cdqss}(t,\tau)\|^2, \tag{4}$$

*subject to* $0 < \underline{\tau} \le \tau \le \overline{\tau} < T$, *where* $\|.\|$ *denotes the vector* $L^2[0,T]$*-norm.*

### 2.2. Bohl–Marek method (and a quasi-linear M-matrix formulation)

QSSA may increase the nonlinearity of the model, see, e.g., the Michaelis–Menten equation for enzyme kinetics [11]. While the ODEs describing enzyme kinetics are mildly nonlinear (only quadratic through terms containing products of two reactants), the Michaelis–Menten equation represents a rational function in an involved reactant. Conversely, the Bohl–Marek (BM) method, makes the model quasi-linear because the ODE system (1) with conservation properties containing the original mass action kinetics terms can be described using the quasi-linear formulation (5). As far as we know, the first appearance of this approach can be found in the works of Erich Bohl and Ivo Marek, see, e.g., [1, 2, 8], where the principle of total mass conservation was employed to prove the existence of positive solutions and stationary states. The details about the BM method applied to our case study problem are

described in Appendix B, here we state that (under some assumptions) the ODE system (1) for a modified state variable vector $\tilde{x}(t)$ can be formulated as a quasi-linear system

$$\frac{\mathrm{d}\tilde{x}(t)}{\mathrm{d}t} = M(x)\tilde{x}(t), \tag{5}$$

with the block diagonal system matrix $M(x)$ of a special form of a negative M-matrix with some elements containing components of a system variable $x$. The advantages of this formulation reside in the computational speedup and precision and shall be highlighted in the next Section 3.

## 3. Case study

As a case study, we take the paradigmatic example consisting of the Michaelis–Menten kinetics with a simple transport process described in Tab. 1.

| **Description of the related process** | **Chem. notation** |
|---|---|
| Substrate $X_{ext}$ dosing | $\emptyset \to X_{ext}$ |
| $R_1$: Substrate transport through a membrane, $k_0 = 10^{-1}$ | $X_{ext} \rightleftharpoons X_{int}$ |
| $R_2$: Enzyme $E$ binds to substrate, complex $C$ formation, $k_1 = 10^{6}$ | $X_{int} + E \rightleftharpoons C$ |
| $R_3$: Reverse reaction to $R_2$, $k_{-1} = 10^{-4}$ | |
| $R_4$: Complex breaks down into $E$ plus a product $P$, $k_2 = 10^{-1}$ | $C \to E + P$ |

Table 1: Transport and reaction processes defining the network, parameter values taken from [7].

Introducing a new notation for state variables, i.e., an $n$-size (here $n = 5$) vector $x$ according to

$$x(t) = \begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \end{pmatrix}^T \equiv \begin{pmatrix} X_{ext} & X_{in} & E & C & P \end{pmatrix}^T,$$

the ODE system describing the process under study can be written either in the usual form (1), i.e., $\dot{x}(t) = Ax(t) + b(t, x(t))$, see Appendix A or in the quasi-linear Bohl–Marek formulation, see, e.g., [2] and Appendix B, for this special case study.

Equipped by the initial conditions

$$x(0) = \begin{pmatrix} u_0 & 0 & e_0 & 0 & 0 \end{pmatrix}^T = \begin{pmatrix} 5 \cdot 10^{-7} & 0 & 2 \cdot 10^{-7} & 0 & 0 \end{pmatrix}^T, \tag{6}$$

we compare the numerical results obtained from the full (non-reduced) problem (1), (6) with those obtained using different models corresponding to different reduction methods. The state variables $x_1$ and $x_4$ can be considered as fast variables $x_F$, since they satisfy all assumptions for fast variables mentioned in [13]. Thus we use the notations QSSA1, QSSA4, QSSA14, etc. Besides, we compare the results with those obtained from the quasi-linear BM formulation (5), (6).

It is well known that the QSS approximation is derived for larger times (to enable the fast variable to reach its steady state) and hence it may not satisfy the original initial condition. This happens if $x_1$ is considered as a fast variable yielding $x_1^{qss}(t) = x_2(t)$. This conflicts initial conditions $x_1(0) = u_0 > 0$ and $x_2(0) = 0$ (it cannot hold $x_1^{qss}(0) = x_2(0)$). Therefore, we introduce a parameter $t_Q$, $0 < t_Q < T$, and derive the QSS approximation for $t > t_Q$, only.

For our numerical experiments, we used parameters given in Tab. 1, $T = 120$, and the time step $\Delta t = 10^{-3}$ for solving the respective ODEs by the backward Euler method. The value $m = \frac{T}{\Delta t}$ denotes the total number of steps. To compare the quality of approximate solutions $x^A(t)$ with a solution $\bar{x}(t)$ of the original non-reduced model (full system) (1),(6), for each of the five state variables we used the error metrics $\delta_i$ and the total error $\delta$ as follows

$$\delta = \frac{1}{n}\sum_{i=1}^{n}\delta_i, \qquad \delta_i = \sqrt{\frac{4}{m}\sum_{j=0}^{m}\left[\frac{\bar{x}_i(t_j) - x_i^A(t_j)}{\bar{x}_i(t_j) + x_i^A(t_j)}\right]^2}, \quad i = 1, \ldots, n. \tag{7}$$

In (7), the exact solution $\bar{x}_i(t_j)$, $j = 0, 1, \ldots, m$, is supposed to be the solution computed using the non-reduced model (full system) (1),(6). The values $x_i^A(t_j)$, $j = 0, 1, \ldots, m$, $i = 1, \ldots, n$, are approximate solutions computed from the models QSSAk (i.e., $x^{qss}(t_j)$), D-QSSAk (i.e., $x^{dqss}(t_j)$ with the delay $\tau(t) = 1/g(t)$), OD-QSSAk (i.e., $x^{odqss}(t_j)$ with an optimal constant delay $\tau^*$ in the sense of optimization problem (4), see Theorem 1), k = 1, 4, 14, and from the BM formulation. The nonconstant delays in models D-QSSAk are $\tau_1(t) = 1/g_1(t) = 1/k_0 = 10$ and $\tau_4(t) = 1/g_4(t) = 1/(k_{-1} + k_2 + k_1 x_1(t))$, respectively. Note that $\tau_1$ is constant because the function $g(t) = k_0$ is constant.

A schematic description of the studied models with obtained optimal values $t_Q$ and optimal constant delays $\tau_1^*$, $\tau_4^*$ are given in Tab. 2. Other columns give the total error metric $\delta$, see (7), and the computational time obtained for 1000 simulations with exactly the same parameter values. The last column shows the speedup obtained as the ratio of computational times between individual models and the full non-reduced model.

Fig. 1 shows the behaviour of state variables $x_1$ and $x_4$ for different models QSSAk, D-QSSAk, OD-QSSAk, k =1,4, and BM. The left picture shows the value $t_Q = 10.77$, from which the quasi-steady-state solutions are considered. Different approaches $(x^{qss}(t), x^{dqss}(t), x^{odqss}(t))$ give different solutions. The right picture shows the optimal constant delay $\tau_4^* = 4.897$ which gives zero quasi-steady-state solution $x_4(t) = 0$, $t \in [0, \tau_4^*]$. Note that the nonconstant delay $\tau_4(t) = 1/g(t)$ for a D-QSS approximation is for small $t$ nearly the same as the optimal constant value $\tau_4^* = 4.897$. Besides, notice that the BM quasi-linear solution is almost the same as the solution of non-reduced model (1), (6).

Resuming: It can be seen that although it is possible to find optimal values of constant delays $\tau^*$ that can significantly speed up the computation when $x_1$ and $x_4$ are fast (we are solving small ODE systems), it is more efficient to convert the

| **model** | **description** | $t_Q$ | **delay** $\tau$ | **total** $\delta$ | **time** | **speedup** |
|---|---|---|---|---|---|---|
| non-reduced | system (1),(6) | - | - | - | 21.94 | 1.00 |
| QSSA1 | $x_1$ fast | opt. | - | 1.0408 | 18.18 | 0.83 |
| QSSA4 | $x_4$ fast | - | - | 0.2736 | 18.28 | 0.83 |
| QSSA14 | $x_1, x_4$ fast | opt. | - | 1.1524 | 5.78 | 0.26 |
| D-QSSA1 | $x_1$ fast | opt. | $\tau_1 = 1/g_1(t)$ | 0.2960 | 21.58 | 0.98 |
| D-QSSA4 | $x_4$ fast | - | $\tau_4 = 1/g_4(t)$ | 0.1896 | 20.34 | 0.93 |
| D-QSSA14 | $x_1, x_4$ fast | opt. | $\tau_i = 1/g_i(t)$ | 0.3237 | 9.27 | 0.42 |
| OD-QSSA1 | $x_1$ fast | 10.77 | $\tau_1^* = 12.753$ | 0.1634 | 21.58 | 0.98 |
| OD-QSSA4 | $x_4$ fast | - | $\tau_4^* = 4.897$ | 0.1952 | 17.44 | 0.79 |
| OD-QSSA14 | $x_1, x_4$ fast | 12.54 | $\tau_1^* = 12.417$ | 0.1563 | 6.03 | 0.28 |
| | | | $\tau_4^* = 11.426$ | | | |
| BM | system (5),(6) | - | - | 0.0006 | 5.30 | 0.24 |

Table 2: Comparison of the studied models: (i) Schematic description, (ii) Computed and used optimal values $t_Q$ and delay $\tau^*$, (iii) Computed total error $\delta$, (iv) Computational times and the speedup.

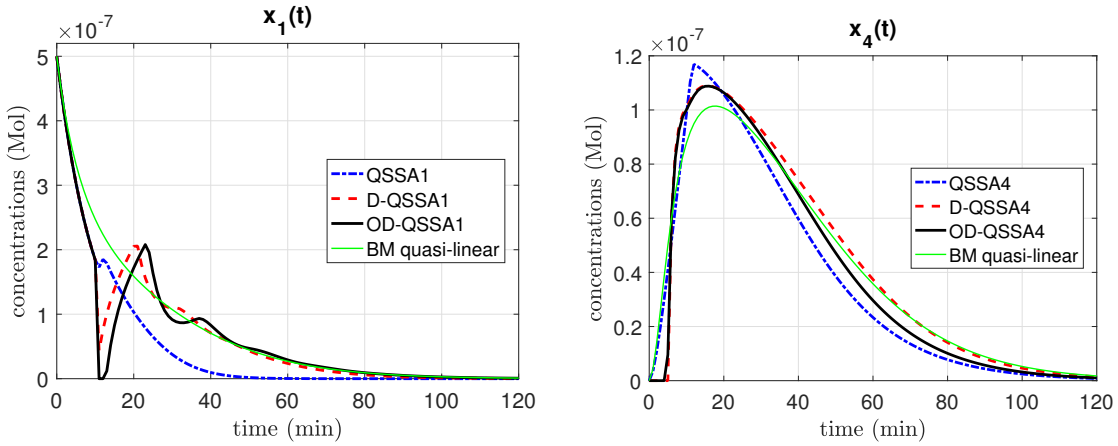



Figure 1: Comparison of $x_1(t)$ and $x_4(t)$ obtained using different models.

problem to the BM quasi-linear form, obviously, if and only if all the corresponding requirements are met (especially the conservation properties).

## 4. Contribution and Outline

We presented one relatively unknown model reduction technique for a class of (bio)chemical reaction networks proposed first by Vejchodský in [13]. The assumptions for this, the so-called D-QSSA approximation are not too restrictive and D-QSSA applies to the majority of (bio)chemical systems based on the law of mass action. While the standard QSSA ignores the time-fast variables needed to reach

their steady states, the advantage of D-QSSA (and its variant OD-QSSA) is the possibility of a time delay introduction to improve the accuracy. This general conclusion was supported by the example presented in Section 3, where we used the case study of enzyme-catalyzed reactions with a substrate transport chain, see [9] for further details. Moreover, we performed a preliminary comparison of numerical computations for two equivalent formulations of governing (non-reduced) ODEs, i.e., for the classical formulation (1) and the quasi-linear Bohl–Marek formulation (5), showing the considerable speedup for the latter. It is due to eliminating the nonlinear part $b(t, x(t))$ from the system which causes a numerical burden when solving ODEs. Rigorous analysis of numerical issues related to both approaches is the subject of our ongoing work.

**Acknowledgments**

The work of Ctirad Matonoha was supported by the long-term strategic development financing of the Institute of Computer Science (RVO:67985807).

**References**

[1] Bohl, E. and Marek, I.: Existence and uniqueness results for nonlinear cooperative systems. In: I. Gohberg and H. Langer (Eds.), *Linear Operators and Matrices*. Birkhäuser Basel, Basel, 2002 pp. 153–170.

[2] Bohl, E. and Marek, I.: Input-output systems in biology and chemistry and a class of mathematical models describing them. Applications of Mathematics **50** (2005), 219–245. `https://doi.org/10.1007/s10492-005-0015-1`.

[3] Duintjer Tebbens, J., Matonoha, C., Matthios, A., and Papáček, Š.: On parameter estimation in an *in vitro* compartmental model for drug-induced enzyme production in pharmacotherapy. Applications of Mathematics **64** (2019), 253–277.

[4] Eilertsen, J. and Schnell, S.: The quasi-steady-state approximations revisited: Timescales, small parameters, singularities, and normal forms in enzyme kinetics. Mathematical Biosciences **325** (2020). Cited by: 15; All Open Access, Green Open Access.

[5] Flach, E. H. and Schnell, S.: Use and abuse of the quasi-steady-state approximation. Syst. Biol. **4** (2006), 187–91.

[6] Glizer, V. Y.: *Controllability of Singularly Perturbed Linear Time Delay Systems*. Birkhäuser Cham, 2022.

[7] Higham, D. J.: Modeling and simulating chemical reactions. SIAM Review **50** (2008), 347–368.

[8] Marek, I.: On a class of stochastic models of cell biology: Periodicity and controllability. In: *Positive Systems*, pp. 359–367. Springer Berlin Heidelberg, 2009.

[9] Matonoha, C., Papáček, Š., and Lynnyk, V.: On an optimal setting of constant delays for the D-QSSA model reduction method applied to a class of chemical reaction networks. Applications of Mathematics **50** (2022), 831–857. http://eudml.org/doc/298520.

[10] Papáček, Š. and Lynnyk, V.: Quasi-steady state assumption vs. delayed quasi-steady state assumption: Model reduction tools for biochemical processes. In: *2021 23rd International Conference on Process Control (PC)*. 2021 pp. 278–283.

[11] Segel, L. A. and Slemrod, M.: The quasi-steady-state assumption: A case study in perturbation. SIAM Review **31** (1989), 446–477.

[12] Snowden, T. J., van der Graaf, P. H., and Tindall, M. J.: Methods of model reduction for large-scale biological systems: A survey of current methods and trends. Bulletin of Mathematical Biology **79** (2017), 1449–1486.

[13] Vejchodský, T.: Accurate reduction of a model of circadian rhythms by delayed quasi-steady state assumptions. Mathematica Bohemica **139** (2014), 577–585. http://hdl.handle.net/10338.dmlcz/144135.

[14] Vejchodský, T., Erban, R., and Maini, P. K.: Reduction of chemical systems by delayed quasi-steady state assumptions, arXiv:1406.4424, 2014.

[15] Witelski, T. and Bowen, M.: *Fast/slow Dynamical Systems*, pp. 201–213. Springer International Publishing, Cham, 2015.

## Appendix A

### Matrix $A$ of constant coefficients and vector of nonlinear terms $b(t, x(t))$

The system of differential equations (1) describing the processes under study can be systematically derived using the vector of reaction rates and the so-called stoichiometric matrix $S \in \mathcal{R}^{n\times q}$, where $q$ is the number of reactions (including the transport of species). Generally, for chemical reaction networks, the governing ODE system, i.e., the vector of changes in species concentrations $x \in \mathcal{R}^n$, is described as a linear transformation (imposed by the matrix $S$) of the reaction rate vector $\nu \in \mathcal{R}^q$ (depending on corresponding states $x$ and a model parameter vector $p$). For our case study $x \in \mathcal{R}^5$, $q = 4$ (see Tab. 1 in Section 3), and it holds:

$$\dot{x}(t) = S\ \nu(x, p), \qquad \text{where } p = (k_0, k_1, k_{-1}, k_2)^T\,, \tag{8}$$

$$S = \begin{array}{c} \begin{matrix} R_1 & R_2 & R_3 & R_4 \end{matrix} \\ \begin{pmatrix} -1 & 0 & 0 & 0 \\ 1 & -1 & 1 & 0 \\ 0 & -1 & 1 & 1 \\ 0 & 1 & -1 & -1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array}, \qquad \nu = \begin{pmatrix} k_0\ (x_1 - x_2) \\ k_1\ x_2\ x_3 \\ k_{-1}\ x_4 \\ k_2\ x_4 \end{pmatrix}. \tag{9}$$

Thus, the ODE system in the usual form (1), i.e., $\dot{x}(t) = Ax(t) + b(t, x(t))$, has the constant matrix (the linear part of the system)

$$A = \begin{pmatrix} -k_0 & k_0 & 0 & 0 & 0 \\ k_0 & -k_0 & 0 & k_{-1} & 0 \\ 0 & 0 & 0 & k_{-1} + k_2 & 0 \\ 0 & 0 & 0 & -(k_{-1} + k_2) & 0 \\ 0 & 0 & 0 & k_2 & 0 \end{pmatrix} \tag{10}$$

and the vector representing the nonlinear part

$$b(t, x(t)) = \begin{pmatrix} 0 \\ -k_1 \cdot x_2(t) \cdot x_3(t) \\ -k_1 \cdot x_2(t) \cdot x_3(t) \\ k_1 \cdot x_2(t) \cdot x_3(t) \\ 0 \end{pmatrix}. \tag{11}$$

**Remark 2.** *Reaction networks frequently possess subsets of reactants that remain constant at all times, i.e., they are referred to as conserved species. Generally, there exists a conservation matrix* $\Gamma$ *(of dimension* $h \times n$*), where the rows represent the linear combination of species (reactants) that are constant in time. It can be solved explicitly for large systems (*$0 = \Gamma\, S$*). For our case of* $S$ *in form* (9)*, the conservation property reads*

$$x_3 + x_4 = e_0, \quad x_1 + x_2 + x_4 + x_5 = u_0. \tag{12}$$

*Consequently, here*

$$\Gamma = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \end{pmatrix}. \tag{13}$$

The existence of two relations (12) signifies not only the possibility to reduce the number of state variables, but also induces the reformulation of the governing equations for species concentration using negative M-matrices, see Appendix B.

## Appendix B

### Matrix $M$ and Bohl–Marek formulation

Based on the mass conservation properties, the non-linear ODEs (1) can be represented as a linear system with the system matrix of a special form, a negative M-matrix. To the best of our knowledge, this approach was first proposed by Erich Bohl and Ivo Marek [1, 2] and further extended into the framework of control theory in [8].

For the case study defined by Tab. 1, the state variables can be listed in two subsets $\{x_3, x_4\}$, $\{x_1, x_2, x_4, x_5\}$ and the non-linear ODEs (1) can be represented as a linear system with the system matrix of a special form, a negative M-matrix whose

column sums are zero.[1] These two subsets of state variables can be assembled and merged as follows:

$$\tilde{x}(t) = \left( {x^{(1)}}^T(t),\ {x^{(2)}}^T(t) \right)^T,$$

where

$$x^{(1)}(t) = \begin{pmatrix} x_3(t) \\ x_4(t) \end{pmatrix}, \quad x^{(2)}(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \\ x_4(t) \\ x_5(t) \end{pmatrix}. \tag{14}$$

Then the ODE system for a modified state variable vector $\tilde{x}(t)$ gets the form which was already announced in (5):

$$\frac{\mathrm{d}\tilde{x}(t)}{\mathrm{d}t} = M\tilde{x}(t). \tag{15}$$

For our case study problem, the block diagonal system matrix $M = M(x(t))$ is of a special form

$$M = \begin{pmatrix} M_1 & 0 \\ 0 & M_2 \end{pmatrix}, \tag{16}$$

where

$$M_1 = \begin{pmatrix} -k_1 \cdot x_2 & k_{-1} + k_2 \\ k_1 \cdot x_2 & -(k_{-1} + k_2) \end{pmatrix}, \tag{17}$$

$$M_2 = \begin{pmatrix} -k_0 & k_0 & 0 & 0 \\ k_0 & -k_0 - k_1 \cdot x_3 & k_{-1} & 0 \\ 0 & k_1 \cdot x_3 & -(k_{-1} + k_2) & 0 \\ 0 & 0 & k_2 & 0 \end{pmatrix}. \tag{18}$$

[1] This property in fact assures the conservation of the sum of all components of the (new) state variable vector $\tilde{x}$.

*Programs and Algorithms of Numerical Mathematics 22*
J. Chleboun, J. Papež, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics CAS, Prague 2025

# COMPARISON OF PRECONDITIONING AND DEFLATION TECHNIQUES OF FETI METHODS FOR PROBLEM OF 2D LINEAR ELASTICITY

Adam Růžička[1,2], David Horák[1,2]

[1] Department of Applied Mathematics, FEECS, VŠB-TUO
17. listopadu 2172/15, Ostrava, Czech Republic
adam.ruzicka.st@vsb.cz, david.horak@vsb.cz
[2] Institute of Geonics, Czech Academy of Sciences
Studentská 1768/9, Ostrava, Czech Republic

**Abstract:** This paper deals with the basic preconditioning and deflation variants of the FETI-1 and TFETI-1 methods, with (T)FETI-1 with deflation being called (T)FETI-2. It also presents the results of numerical experiments performed on a simple benchmark 2D problem of linear elasticity to compare the computational efficiency of FETI-1 and TFETI-1 and each variant of their preconditioning or deflation in terms of number of executed CG iterations.

**Keywords:** FETI, domain decomposition, preconditioning, deflation

**MSC:** 65F08, 65F10

## 1. Introduction

The Finite element tearing and interconnecting (FETI) methods are probably the most commonly used domain decomposition methods for a parallel numerical solution of PDEs. In Section 2, the mathematical formulations of basic FETI methods: FETI-1 and TFETI-1 are presented. The basic ways of their preconditioning are introduced in Section 3. In Section 4, the principle of the deflated conjugate gradient method is presented. In Section 5, mathematical formulations of applied methods of deflation are introduced. In Section 6, the results of numerical experiments performed on the simple benchmark 2D FEM-discretized problem of linear elasticity are presented.

## 2. FETI-1 and TFETI-1 methods

In all methods of the FETI-type, the global problem of linear elasticity discretized by FEM, defined on discretized linear elastic domain, is decomposed into several local problems defined on non-overlapping subdomains which are then glued via conditions of displacements continuity across their mutual interfaces, which leads to the constrained minimization problem of quadratic programming [1]:

DOI: 10.21136/panm.2024.13

$$\min\,(1/2)\,u^T K u - u^T f \quad \text{s.t. } Bu = o \tag{1}$$

$$K = \operatorname{diag}\left(K_1 \;\; \cdots \;\; K_i \;\; \cdots \;\; K_{N_s}\right), \quad K \in \mathbb{R}^{n\times n} \tag{2}$$

$$u = \left[\, u_1^T \;\cdots\; u_i^T \;\cdots\; u_{N_s}^T \,\right]^T;\; f = \left[\, f_1^T \;\cdots\; f_i^T \;\cdots\; f_{N_s}^T \,\right]^T;\; u \in \mathbb{R}^{n\times 1}; f \in \mathbb{R}^{n\times 1} \tag{3}$$

where blocks $K_i, u_i, f_i$ are blocks associated with $i$th subdomain denoting its stiffness matrix, vector of deformation parameters of nodes of the subdomain, and vector of external loading concentrated into the subdomains nodes.

Equality conditions: $Bu = o$, $B \in \mathbb{R}^{m\times n}$ ensure the continuity of node displacements by gluing its subdomains on their interfaces. The Dirichlet boundary conditions (BCs) are prescribed by modifying $K$ and $f$ in corresponding columns and rows (FETI-1), or by adding Dirichlet BCs to the problem constraints expressed by $Bu = o$ (TFETI-1). The constraints $Bu = o$ are then enforced by the vector of Lagrange multipliers $\lambda$, where $\lambda \in \mathbb{R}^{m\times 1}$.

Problem (1) can be expressed as the following saddle-point problem:

$$\begin{bmatrix} K & B^T \\ B & O \end{bmatrix} \begin{bmatrix} u \\ \lambda \end{bmatrix} = \begin{bmatrix} f \\ o \end{bmatrix}. \tag{4}$$

The vector of solution $u$ can be expressed from the first equation in (4)

$$u = u_{\mathrm{Im}K} + u_{\mathrm{Ker}K} = K^+(f - B^T\lambda) + R\alpha, R \in \mathbb{R}^{n\times r}, \alpha \in \mathbb{R}^{r\times 1} \tag{5}$$

where $K^+$ is some form of a generalized inverse of $K$, and $R$ is the matrix whose columns are the basis of $\mathrm{Ker}K$, so it should also hold: $R^T(f - B^T\lambda) = o$.

Dualizing this problem and using the standard FETI notation [2]:

$$F = BK^+B^T; d = BK^+f; G = -R^TB^T; e = -R^Tf, \tag{6}$$

the following problem is obtained:

$$\begin{bmatrix} F & G^T \\ G & O \end{bmatrix} \begin{bmatrix} \lambda \\ \alpha \end{bmatrix} = \begin{bmatrix} d \\ e \end{bmatrix}. \tag{7}$$

After homogenizing: $G\lambda = e$ using $\lambda_0 = G^T(GG^T)^{-1}e, \lambda_0 \in \mathrm{Im}G^T$, the remaining part of $\lambda$ is $\mu$ in $\mathrm{Ker}G$, and the following minimization problem is obtained [2]:

$$\min\,(1/2)\,\mu^T F\mu - \mu^T(d - F\lambda_0) \text{ s.t. } G\mu = o. \tag{8}$$

The equality constraint can be enforced by dual penalty or more efficiently by the orthogonal projector $P$ onto $\mathrm{Ker}\,G$, $P \in \mathbb{R}^{m\times m}$ [2]:

$$P = I - G^T(GG^T)^{-1}G, \tag{9}$$

so that the minimization problem is equivalent to the problem of finding the solution $\bar{x}$ of the system of linear equations

$$Ax = b; A = PFP; x = \mu; b = P(d - F\lambda_0), \tag{10}$$

which is solved iteratively, typically by the conjugate gradients (CG).

The primal solution $\bar{u}$ can be reconstructed as follows: [2]

$$\overline{\alpha} = (GG^T)^{-1}G\,(d - F(\lambda_0 + \overline{x}));\; \overline{u} = K^+(f - B^T(\lambda_0 + \overline{x})) + R\overline{\alpha}. \tag{11}$$

## 3. Preconditiong and preconditioned conjugate gradients (PCG) method

There exist two basic FETI preconditioners for FETI-1 and TFETI-1, both approximating the inverse of the matrix $F$. To assemble these preconditioners, the stiffness matrix $K$ has to be divided into 4 blocks [3]:

$$K = \begin{bmatrix} K_{ii} & K_{ib} \\ K_{ib}^T & K_{bb} \end{bmatrix}, \tag{12}$$

where $K_{ii}$, and $K_{bb}$ are composed of elements of $K$ associated with subdomains' internal, respectively boundary nodes, etc. Likewise, the gluing matrix $B$ should be divided into blocks $B_i$ and $B_b$: $B = \begin{bmatrix} B_i & B_b \end{bmatrix}$. The block $B_i$, associated with internal nodes of subdomains, is always a zero matrix, since the conditions of equality of the deformation parameters, respectively Dirichlet BCs of the problem, are expressed only between or for boundary nodes of the subdomains.

### 3.1. Dirichlet preconditioner (DP)

The Dirichlet preconditioner is expressed as follows [3]:

$$F_I^{D^{-1}} = \begin{bmatrix} B_i & B_b \end{bmatrix} \begin{bmatrix} O & O \\ O & S_{bb} \end{bmatrix} \begin{bmatrix} B_i^T \\ B_b^T \end{bmatrix} = B_b S_{bb} B_b^T, \; S_{bb} = K_{bb} - K_{ib}^T K_{ii}^{-1} K_{ib}, \tag{13}$$

where $S_{bb}$ is the Schur complement of the block $K_{ii}$.

### 3.2. Lumped preconditioner (LP)

The matrix of the lumped preconditioner is an approximation of the Dirichlet one with only the first term in the relation for computation of $S_{bb}$ used [3]:

$$F_I^{L^{-1}} = \begin{bmatrix} B_i & B_b \end{bmatrix} \begin{bmatrix} O & O \\ O & K_{bb} \end{bmatrix} \begin{bmatrix} B_i^T \\ B_b^T \end{bmatrix} = \begin{bmatrix} B_i & B_b \end{bmatrix} \begin{bmatrix} K_{ii} & K_{ib} \\ K_{ib}^T & K_{bb} \end{bmatrix} \begin{bmatrix} B_i^T \\ B_b^T \end{bmatrix} = BKB^T. \tag{14}$$

The lumped preconditioner is less accurate and less optimal approximation of the inverse of $F$, so its effect as a preconditioner on improving the spectral properties of the system matrix and reducing the number of PCG iterations is smaller than with the Dirichlet preconditioner, but its computation is significantly cheaper [3].

## 4. Deflation and deflated conjugate gradient (DCG) method

When solving the system of equations $Ax = b$ using the CG method, the $k$th approximation $x_k$ of the solution vector is found as the minimizer of quadratic function $f(x) = \frac{1}{2}x^T Ax - x^T b$ over the $k$th Krylov subspace $\mathcal{K}_k(A, r_0)$.

The basic idea of the DCG method is to enrich the Krylov subspace $\mathcal{K}_k$ by some subspace $\mathcal{W}$, the so-called deflation subspace. If $\mathcal{W}$ is defined conveniently, a faster convergence of the CG method, solving the system, can be anticipated [4], [5].

Let the subspace $\mathcal{W}$ be spanned by column vectors $w_j$ forming the matrix $W$:

$$W = \begin{bmatrix} w_1 & \dots & w_j & \dots & w_m \end{bmatrix} \tag{15}$$

then the projector $P_D$ on $A$-conjugate complement of the deflation subspace $\mathcal{W}$ can be formulated as follows [4], [5]:

$$P_D = I - QA = I - W(W^T AW)^{-1} W^T A. \tag{16}$$

In the DCG method, the process of a solution can be split into 2 parts: solution on the deflation subspace $\mathcal{W}$ and solution on its A-conjugate complement. It is achieved using the fact that in a classical CG method it holds that the vector $r_k$ of the residual in the $k$th iteration is orthogonal to $k$th Krylov subspace $\mathcal{K}_k(A, r_0)$, over which the quadratic functional $f(x)$ is minimized in the $k$th iteration [4], [5].

If some arbitrary initial guess $x_{-1}$ is given, then the corresponding vector of residual is $r_{-1} = b - Ax_{-1}$, and the correction $x_0$ of the initial guess in the deflation subspace $\mathcal{W}$ is then computed as follows [4], [5]:

$$x_0 = x_{-1} + Qr_{-1} = x_{-1} + W(W^T AW)^{-1} W^T r_{-1}. \tag{17}$$

If the last equation is multiplied by $W^T A$ from the left, then

$$W^T Ax_0 = W^T Ax_{-1} + W^T AW(W^T AW)^{-1} W^T (b - Ax_{-1}) \tag{18}$$

$$W^T b - W^T Ax_0 = W^T r_o = o, \tag{19}$$

so that the vector $r_0$ corresponding to $x_0$ satisfies the condition of its orthogonality to $\mathcal{W}$, i.e., it has no components in $\mathcal{W}$, and thus $x_0$ is the exact solution in $\mathcal{W}$.

If columns of the deflation matrix $W$ are exact eigenvectors of the system matrix $A$ computed in exact arithmetics, it holds: $W^T A = \Lambda W^T$, where $\Lambda$ is a diagonal matrix with eigenvalues of $A$, with the $k$th entry corresponding to the $k$th column of $W$, i.e., to the $k$th of the chosen eigenvectors. Thus, in such case also the $k$th Krylov subspace $\mathcal{K}_k(A, r_0)$ is orthogonal to $\mathcal{W}$ since $W^T A^{k-1} = \Lambda^{k-1} W^T$.

Since the residual $r_k$ in the $k$th iteration of the CG method belongs to $\mathcal{K}_{k+1}(A, r_0)$, $k = 0, 1, \ldots$, then $r_k$ is orthogonal to and thus has no components in $\mathcal{W}$.

However, since the computations in reality cannot be performed in exact arithmetic and $\mathcal{W}$ generally does not consist of exact eigenvectors of $A$, the residual $r_k$ is not $A$-conjugate to $\mathcal{W}$, in general. Thus, conjugate directions $p_k$ are generally not $A$-conjugate to $\mathcal{W}$ since in standard CG it holds: $p_{k+1} = r_{k+1} + \beta_{k+1} p_k$, which is a problem since the approximations $x_k$ of a solution are searched only on the $A$-conjugate complement of $\mathcal{W}$. Thus, some sort of correction has to be performed in the iteration process to make the vectors $p_k$ $A$-conjugate to $\mathcal{W}$, so that the approximations $x_k$ are searched only in the $A$-conjugate complement of $\mathcal{W}$.

The correction is performed in a way that in relation used to compute the vector $p_k$ in the standard CG method, the vector of residual $r_k$ is projected onto the $A$-conjugate complement of $\mathcal{W}$, by multiplying it by the projector $P_D$ defined in (16), so the relations for computing $p_0$ and $p_{k+1}$, $k = 0, 1, \ldots$ get the following form:

$$p_0 = P_D\, r_0;\ p_{k+1} = P_D\, r_{k+1} + \beta_{k+1} p_k. \tag{20}$$

By ensuring that the approximations $x_k$ of the solution during the iterative process of DCG are searched only on the $A$-conjugate complement of the deflation subspace $\mathcal{W}$, the required splitting of the solution into components on $\mathcal{W}$ (in the form of correction of the initial guess) and on its $A$-conjugate complement is achieved. The CG, PCG, and DCG algorithms are presented in Table 1.

| CG | PCG | DCG |
|---|---|---|
| Input : $A, b, x_0, k = 0$ | Input : $A, b, x_0, M^{-1}, k = 0$ | Input : $A, b, x_{-1}, W, k = 0$ |
| | | $Q = W(W^T AW)^{-1}W^T$ |
| | | $P_D = I - QA$ |
| | | $r_{-1} = b - Ax_{-1}$ |
| | | $x_0 = x_{-1} + Qr_{-1}$ |
| $r_0 = b - Ax_0$ | $r_0 = b - Ax_0$ | $r_0 = b - Ax_0$ |
| | $z_0 = M^{-1}r_0$ | |
| $p_0 = r_0$ | $p_0 = z_0$ | $p_0 = P_D\, r_0$ |
| while (some ending criterium) | while (some ending criterium) | while (some ending criterium) |
| $s = Ap_k$ | $s = Ap_k$ | $s = Ap_k$ |
| $\alpha_k = (r_k^T r_k)/(s^T p_k)$ | $\alpha_k = (r_k^T z_k)/(s^T p_k)$ | $\alpha_k = (r_k^T r_k)/(s^T p_k)$ |
| $x_{k+1} = x_k + \alpha_k p_k$ | $x_{k+1} = x_k + \alpha_k p_k$ | $x_{k+1} = x_k + \alpha_k p_k$ |
| $r_{k+1} = r_k - \alpha_k s$ | $r_{k+1} = r_k - \alpha_k s$ | $r_{k+1} = r_k - \alpha_k s$ |
| | $z_{k+1} = M^{-1}r_{k+1}$ | |
| $\beta_{k+1} = (r_{k+1}^T r_{k+1})/(r_k^T r_k)$ | $\beta_{k+1} = (r_{k+1}^T z_{k+1})/(r_k^T z_k)$ | $\beta_{k+1} = (r_{k+1}^T r_{k+1})/(r_k^T r_k)$ |
| $p_{k+1} = r_{k+1} + \beta_{k+1} p_k$ | $p_{k+1} = z_{k+1} + \beta_{k+1} p_k$ | $p_{k+1} = P_D\, r_{k+1} + \beta_{k+1} p_k$ |
| Output : $x_k$ | Output : $x_k$ | Output : $x_k$ |

Table 1: CG, PCG and DCG algorithms

## 5. (T)FETI-2 – deflated variant of (T)FETI-1

In this section, it is considered that deflation is applied on the CG method solving the final system of equations obtained by decomposition of the FEM-discretized problem of 2D linear elasticity by (T)FETI-1. It is also presumed that both the discretization and decomposition of the analyzed linear elastic domain are conforming.

### 5.1. Deflation by equality of displacements in corner nodes (CE)

Equation $B_C u = o$ expresses the equality conditions of the corresponding displacement components of mutually corresponding corner nodes on the interfaces of neighbouring subdomains in two perpendicular directions $x$ and $y$.

Since conditions $B_C u = o$ are already included in conditions $Bu = o$ using the matrix $B$, the matrix $B_C$ can be obtained by splitting $B$ into two parts as follows $B = \begin{bmatrix} B_C^T & B_R^T \end{bmatrix}^T$, with $B_R$ expressing the equality conditions of displacements of the remaining nodes by $B_R u = o$, which are not in corners of subdomains.

The deflation matrix $W$ is: $W = BB_C^T = \begin{bmatrix} B_C^T & B_R^T \end{bmatrix}^T B_C^T = \begin{bmatrix} B_C B_C^T & B_C B_R^T \end{bmatrix}^T = \begin{bmatrix} B_C B_C^T & O^T \end{bmatrix}^T$, where in case of orthonormal rows of $B$ it holds: $W = \begin{bmatrix} I & O^T \end{bmatrix}^T$.

### 5.2. Deflation by equality of the displacement averages and by moment equilibrium of gluing forces on subdomains' interfaces

In this method of defining the deflation subspace $\mathcal{W}$, at first the matrix $B_A$ has to be defined. This matrix will be divided into two vertical blocks $B_{A\_A}$ and $B_{A\_M}$, i.e. $B_A = \begin{bmatrix} B_{A\_A}^T & B_{A\_M}^T \end{bmatrix}^T$ for purposes of following formulations.

The block $B_{A\_A}$ in the relation $B_{A\_A}u = o$ expresses the conditions enforcing the equality of the averages of values of displacement components of each node along opposite sides of each corresponding interface of 2 subdomains.

The block $B_{A\_M}$ is used to express the conditions of moment equilibrium of the force system of solitary forces of contributions of the notional total gluing force, acting on the interfaces of two subdomains, distributed continuously and uniformly along their length, as contributions concentrated in each corresponding node on either side of the interface, with the total force distributed uniformly (averaged) among the contributions, in terms of their magnitude and direction. The moments of the forces of contributions concentrated in corresponding nodes are all related to the same reference point, here always the point with the global coordinates [0,0].

The moment of the corresponding component, in direction of axis $x$, or $y$, of the solitary force of corresponding contribution of the total gluing force, concentrated into the node on interface, denoted as 12, of two subdomains: 1 and 2, related to the point [0,0] is equal to the corresponding element of the vector of the product of transpose of the corresponding row of matrix $B_{A\_M}$, with the corresponding element, denoted e.g. as $\lambda_{A\_M_{12}}$ of the vector $\lambda_{A\_M}$, where sum of all the elements of the product equals zero, i.e., the moment equilibrium of the force system is ensured.

The deflation matrix $W$ is computed as $W = BB_A^T$.

### 5.2.1. Conditions of averages equality (AE)

The formulation of the conditions of equality of averages of displacement components in two directions $x$ and $y$, of nodes lying along opposite sides of the interface, denoted as 12, of two subdomains: 1 and 2, of decomposed domain, is following:

$$x : \frac{1}{n}\sum_{k=1}^{n} u_{1_{k\_x}} - u_{2_{k\_x}} = 0, y : \frac{1}{n}\sum_{k=1}^{n} u_{1_{k\_y}} - u_{2_{k\_y}} = 0, \tag{21}$$

where $n$ is the number of nodes on side of the interface and $u_{1(2)_{k\_x(y)}}$ is the displacement component of the $k$-th node in the $x(y)$ direction. The structure of the two corresponding rows of the $B_{A\_A}$ matrix expressing these two conditions is then:

$$\frac{1}{n}\begin{bmatrix} & 1_{1\_x} & 1_{1\_y} & \cdots & 1_{k\_x} & 1_{k\_y} & \cdots & 1_{n\_x} & 1_{n\_y} & & 2_{1\_x} & 2_{1\_y} & \cdots & 2_{k\_x} & 2_{k\_y} & \cdots & 2_{n\_x} & 2_{n\_y} & \\ O & 1 & 0 & \cdots & 1 & 0 & \cdots & 1 & 0 & O & -1 & 0 & \cdots & -1 & 0 & \cdots & -1 & 0 & O \\ O & 0 & 1 & \cdots & 0 & 1 & \cdots & 0 & 1 & O & 0 & -1 & \cdots & 0 & -1 & \cdots & 0 & -1 & O \end{bmatrix} \tag{22}$$

The conditions of equality of displacement averages on the interface are implicitly also the conditions of equilibrium of the force system of solitary forces of discrete contributions of the total gluing force into corresponding nodes on that interface.

### 5.2.2. Conditions of moment equilibrium (ME)

The condition of moment equilibrium of the discrete contributions of the total gluing force acting along entire length of the interface 12 of two subdomains: 1 and 2, concentrated and uniformly distributed in each node on either corresponding side of the interface, with moments all related to the reference point [0,0], is enforced using the corresponding row of $B_{A\_M}$ of the following structure:

$$\left[\begin{array}{cccccccccccccccc} & 1_{1\_x} & 1_{1\_y} & \cdots & 1_{k\_x} & 1_{k\_y} & \cdots & 1_{n\_x} & 1_{n\_y} & & 2_{1\_x} & 2_{1\_y} & \cdots & 2_{k\_x} & 2_{k\_y} & \cdots & 2_{n\_x} & 2_{n\_y} & \\ O & -y_1 & x_1 & \cdots & -y_k & x_k & \cdots & -y_n & x_n & O & y_1 & -x_1 & \cdots & y_k & -x_k & \cdots & y_n & -x_n & O \end{array}\right], \quad (23)$$

where, $x(y)_1(k,n)$ is the $x(y)$-coordinate, in the global coordinate system, belonging to the first ($k$-th, last) node on each side of the interface.

### 5.3. Deflation by the eigenvectors of the system matrix (EIG)

As it was mentioned in Section 4 concerning the DCG method, the deflation matrix $W$ should be in an ideal case composed of the (exact) eigenvectors of the system matrix $A$. If the columns $W$ are exact eigenvectors of the system matrix $A$ computed in exact arithmetic, then $W^TA = \Lambda_M W^T$, where $\Lambda_M$ is a diagonal matrix with eigenvalues of $A$, where the $k$th diagonal entry ($k$th eigenvalue) corresponds to the $k$th column of $W$.

To achieve the desired effect of deflation in significantly improving the spectral properties of the spectral operator $P_DA$ in the iterative process of the DCG method, and thus speeding up convergence of the iterative process, the eigenvectors of the matrix $A$ that slow down convergence the most should be deflated, which are usually those corresponding to the extremal, usually the lowest, eigenvalues of $A$. If the eigenvectors of $A$, which form $W$, are favourably selected, then the desired effect of deflation can be reached for a relatively small number of eigenvectors of $A$, which leads to a small matrix $W^TAW$ of the coarse problem (CP) in the DCG method and thus to a computationally cheap solution of CP [4], [5].

However, the process of obtaining the eigenvalues and eigenvectors is generally very costly, and thus the solution of the system of linear algebraic equations $Ax = b$ by the CG method with a good preconditioner is often faster, in terms of the total time needed for the assembly of the preconditioner, or the deflation matrix, and the subsequent solution of the system using the PCG, or DCG method.

### 5.4. Deflation by discrete wavelet transform (DWT)

In the following text, it is considered that the discrete Haar wavelet, as the structurally simplest, is applied during the DWT. The Haar wavelet has two filters, the "low-frequency" and "high-frequency", which are used to obtain the components of some signal corresponding to the low/high frequencies.

The process of splitting 1D signal, represented by vector $x$, into its low- and high-frequency components, is in $k$th level of forward DWT represented by decomposition of the vector $a_{k-1}$, with $a_0 = x$, lying in so-called $(k-1)$th discretized scaling subspace $V_{k-1}$, on its so-called approximation coefficients $a_k$ (corresponding to the lower frequencies), lying in $k$th discretized scaling subspace $V_k$, and detailed coefficients $d_k$ (higher frequencies), in so-called $k$th discretized wavelet subspace $W_k$ as orthogonal complement of $V_k$ in $V_{k-1}$, is carried out by the gradual application of corresponding orthogonal projectors $H_k$ (from $V_{k-1}$ onto $V_k$) and $G_k$ (from $V_{k-1}$ onto $W_k$).

This means that in the $k$-th level, where $k = 1, \ldots, M$, with $M$ being the given chosen total number of levels of DWT performed, it holds:

$$a_k = H_k a_{k-1},\ d_k = G_1 a_{k-1},\ \begin{bmatrix} a_k \\ d_k \end{bmatrix} = \begin{bmatrix} H_k \\ G_k \end{bmatrix} a_{k-1}. \tag{24}$$

Thus the vector $a_M$ obtained after $M$ compressions ($M$ levels of forward DWT) applied on the vector $x$, of length equal to $N/2^M$, or to its closest higher or lower integer to $N/2^M$, of original signal $x$ (of length $N$) can be expressed using the matrix $H$ of the total projector from the space $V_0 = l^2(N)$ to the $V_M$ in the following way:

$$a_M = H_M H_{M-1} \dots H_k \dots H_2 H_1 x = Hx. \tag{25}$$

The inverse DWT of $a_M$ is then given by multiplication of $a_M$ by transpose of $H$, i.e. by $H^T$, where the vector obtained by applying $M$ levels of inverse DWT using $H^T$ on the vector obtained by $M$ levels of DWT on $x$ using matrix $H$, only the trend part of the signal represented by $x$ is preserved.

If DWT is applied to square matrices $A$ in order to obtain only its components corresponding to its lower eigenvalues, the projector $H$ is applied to the columns and its transpose $H^T$ on transformed rows, so that the matrix obtained by 2D FDWT of $A$ is a matrix $A_T = HAH^T$. The deflation matrix $W$ is obtained as $W = H^T$.

The matrix $H_k$ of the orthogonal projector from $V_{k-1}$ onto $V_k$ has structure:

$$H_k = \frac{1}{\sqrt{2}} \begin{bmatrix} \ddots & 1 & 1 & 0 & 0 & \\ & 0 & 0 & 1 & 1 & \ddots \end{bmatrix}. \tag{26}$$

The vector (matrix) on which the projector $H_k$, without modification, is applied at the $k$th level of DWT must have the length (dimensions) divisible by 2. If it does not hold, then some adjustment of the structure of $H_k$ has to be performed; see [6].

In numerical experiments, the case where the structure of matrix of orthogonal projector $H_k$ was adjusted with regard to the fact that 2D decomposed discretized problem of linear elasticity is solved in a following way (27), was also tested:

$$H_{k,2D} = \frac{1}{\sqrt{2}} \begin{bmatrix} \ddots & 1 & 0 & 1 & 0 & \\ & 0 & 1 & 0 & 1 & \ddots \end{bmatrix}, \tag{27}$$

so that $H_{2D} = H_{M,2D} H_{M-1,2D} \dots H_{1,2D}$ and $W_{2D} = H_{2D}^T$.

### 5.5. Deflation by discrete Fourier transform (DFT)

DFT of some vector $x \in l^2(N)$ is in fact the computation of its coordinate vector $c$ in complex orthonormal discrete Fourier basis of the vector space $l^2(N)$. The $k$th component $c_k$, of the vector $c$ as the DFT of the vector $x$ can be computed as the complex inner product of $x$, with the $k$th Fourier basis vector having the structure:

$$F_k = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & (e^{2\pi i/N})^k & \cdots & (e^{2\pi i/N})^{nk} & \cdots & (e^{2\pi i/N})^{(N-1)k} \end{bmatrix}^T. \tag{28}$$

The deflation matrix $W$ is then composed of the first $M$ vectors $F_k$, of a discrete Fourier basis, where $k = 0, 1, \dots, M-1$ as follows:

$$W = \begin{bmatrix} F_0 & F_1 & \cdots & F_k & \cdots & F_{M-1} \end{bmatrix}. \tag{29}$$

The deflation matrix $W$ can again have its structure adjusted with regard to solution of 2D elasticity problem, where the block $F_{k,2D}$ of the deflation matrix $W$, replacing the $k$th discrete Fourier basis vector, is constructed using $k$th discrete Fourier basis vector of the vector space $l^2(N/2)$, and it has the following structure:

$$F_{k,2D} = \frac{2}{\sqrt{N}} \begin{bmatrix} 1 & 0 & \cdots & e^{\frac{2\pi i(nk)}{N/2}} & 0 & \cdots & e^{\frac{2\pi i(N/2-1)k}{N/2}} & 0 \\ 0 & 1 & \cdots & 0 & e^{\frac{2\pi i(nk)}{N/2}} & \cdots & 0 & e^{\frac{2\pi i(N/2-1)k}{N/2}} \end{bmatrix}^T. \quad (30)$$

The deflation matrix, with structure adjusted with regard to solving the 2D decomposed discretized problem of linear elasticity, denoted as $W_{2D}$, is defined as follows:

$$W_{2D} = \begin{bmatrix} F_{0,2D} & F_{1,2D} & \cdots & F_{k,2D} & \cdots & F_{M-1,2D} \end{bmatrix}. \quad (31)$$

### 5.6. Deflation by discrete cosine transform (DCT)

DCT works on similar principle as DFT, only the complex discrete Fourier basis is replaced by real discrete cosine basis, whose $k$th vector has following structure:

$$C_k = \sqrt{\frac{2-\delta_{k,0}}{N}} \begin{bmatrix} \cos\frac{(1/2)k\pi}{N} & \cdots & \cos\frac{(n+1/2)k\pi}{N} & \cdots & \cos\frac{(N-1+1/2)k\pi}{N} \end{bmatrix}^T, \quad (32)$$

and the deflation matrix $W$ is then composed of the first $M$ vectors of this discrete cosine basis $C_k$, $k = 0, 1, ..., M-1$ :

$$W = \begin{bmatrix} C_0 & C_1 & \cdots & C_k & \cdots & C_{M-1} \end{bmatrix}. \quad (33)$$

The structure of the deflation matrix $W$, respectively of the vectors $C_k$ as the columns of $W$ can be again adjusted with regard to solving 2D problem of elasticity

$$C_{k,2D} = \sqrt{\frac{2-\delta_{k,0}}{N/2}} \begin{bmatrix} \cdots & \cos\frac{(n+1/2)k\pi}{N/2} & 0 & \cdots \\ \cdots & 0 & \cos\frac{(n+1/2)k\pi}{N/2} & \cdots \end{bmatrix}^T, \; n = 0, 1, \ldots, \frac{N}{2} - 1, \quad (34)$$

resulting in $W_{2D}$ with the structure:

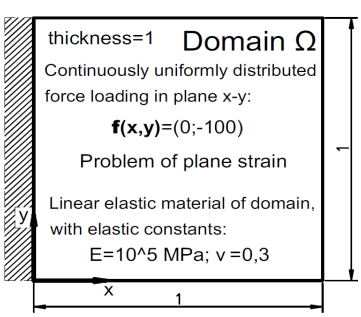
$$W_{2D} = \begin{bmatrix} C_{0,2D} & C_{1,2D} & \cdots & C_{k,2D} & \cdots & C_{M-1,2D} \end{bmatrix}. \quad (35)$$

## 6. Numerical experiments

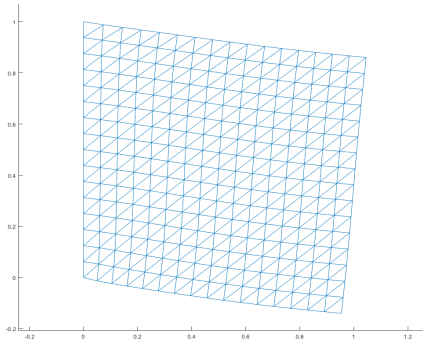
As the benchmark problem used in numerical experiments, model 2D linear elasticity problem, see Fig. 1, defined on 2D linear elastic domain, decomposed into $(1/H)^2$ identical square non-overlapping subdomains with edge lengths $H = 1/2, 1/4, 1/8, 1/16$, was chosen. The subdomains were discretized by 2D identical finite elements of shape of isoscleses right-angled triangle with length $h$. There are presented the results of numerical experiments only for value of the ratio $H/h = 8$.

All algorithms were implemented and numerical experiments were performed in Matlab, as the stopping criteria was used: $||r_k|| \leq \epsilon||b||$, where always: $\epsilon = 10^{-6}$, and the initial guess $x_0$ was always zero vector.

Table 2 presents the corresponding dimensions of the problems for given $H$ using FETI-1/TFETI-1. The dimensions of the deflation subspace $\mathcal{W}$ for each tested variant of deflation are presented in Table 3 .




(a) Definition of the problem



(b) Solution of the problem

Figure 1: The benchmark: 2D problem of linear elasticity

| $H$ | 1/2 | 1/4 | 1/8 | 1/16 |
|---|---|---|---|---|
| Primal dimension | 648 | 2592 | 10368 | 41472 |
| Dual dimension ($\dim(A)$) | 70/104 | 414/480 | 1918/2048 | 8190/8448 |
| $\dim(\mathrm{Ker}K) = (\dim(\mathrm{Ker}A)$ | 6/12 | 36/48 | 168/192 | 720/768 |
| $\mathrm{rank}(A) = \dim(A) - \dim(\mathrm{Ker}A)$ | 64/92 | 378/432 | 1750/1856 | 7470/7680 |

Table 2: Problems dimensions ($A = PFP$, $H/h = 8$)

| | $H$ | 1/2 | 1/4 | 1/8 | 1/16 |
|---|---|---|---|---|---|
| FETI-2 | CE1/CE2 | 6/14 | 54/78 | 294/350 | 1350/1470 |
| | AE/AE+ME | 8/12 | 48/72 | 224/336 | 960/1440 |
| | EIG | 1/2/4/.../32/64 | 1/2/.../256/378 | 1/2/.../1024/1750 | 1/2/.../2048/4096 |
| | DWT1 | 36/18/10/6 | 208/104/52/26 | 960/480/240/120 | 4096/2048/1024/512 |
| | DWT2 | 35/18/9/5 | 207/104/52/26 | 959/480/240/120 | 4095/2048/1024 |
| | DFT1 | 2/4/8/.../32/64 | 2/4/.../256/378 | 2/4/.../1024/1734 | 2/4/.../2048/4096 |
| | DFT2 | 1/2/4/.../32/64 | 1/2/.../256/376 | 1/2/.../1024/1728 | 1/2/.../2048/4096 |
| | DCT1 | 2/4/8/.../32/64 | 2/4/.../256/378 | 2/4/.../1024/1738 | 2/4/.../2048/4096 |
| | DCT2 | 1/2/4/.../32/64 | 1/2/.../256/376 | 1/2/.../1024/1730 | 1/2/.../2048/4096 |
| TFETI-2 | CE1/CE2/CE3 | 6/14/16 | 54/78/84 | 294/350/364 | 1350/1470/1500 |
| | AE/AE+ME | 8/12 | 48/72 | 224/336 | 960/1440 |
| | EIG | 1/2/4/.../64/92 | 1/2/.../256/432 | 1/2/.../1024/1856 | 1/2/.../2048/4096 |
| | DWT1 | 52/26/14/8 | 240/120/60/30 | 1024/512/256/128 | 4224/2112/1056/528 |
| | DWT2 | 52/26/13/7 | 240/120/60/30 | 1024/512/256/128 | 4224/2112/1056/528 |
| | DFT1 | 2/4/8/.../64/92 | 2/4/.../256/432 | 2/4/.../1024/1854 | 2/4/.../2048/4096 |
| | DFT2 | 1/2/4/.../64/92 | 1/2/.../256/432 | 1/2/.../1024/1854 | 1/2/.../2048/4096 |
| | DCT1 | 2/4/8/.../64/92 | 2/4/.../256/432 | 2/4/.../1024/1854 | 2/4/.../2048/4096 |
| | DCT2 | 1/2/4/.../64/92 | 1/2/.../256/432 | 1/2/.../1024/1855 | 1/2/.../2048/4096 |

Table 3: Dimensions of the deflation subspaces

The numbers of performed iterations of (P)CG method with no, lumped and Dirichlet preconditioners, and of the DCG method for each tested variant of deflation, solving the system of equations obtained by (T)FETI-1, are depicted in Table 4.

| | $H$ | 1/2 | 1/4 | 1/8 | 1/16 |
|---|---|---|---|---|---|
| FETI-1 (NO/LP/DP) | | 23/14/8 | 37/20/13 | 45/24/17 | 56/29/25 |
| TFETI-1 (NO/LP/DP) | | 25/14/8 | 34/16/8 | 34/16/11 | 33/16/11 |
| FETI-2 | CE1/CE2 | 18/15 | 22/20 | 25/24 | 26/26 |
| | AE/AE+ME | 16/14 | 24/20 | 25/21 | 26/22 |
| | EIG | 28/23/19/ 16/12/7/0 | 46/46/47/32/27/ 22/15/9/6/0 | 56/56/56/58/44/29/ 27/22/15/10/6/0 | 72/70/70/70/62/52/ 29/28/27/22/16/11/7 |
| | DWT1 | 25/21/13/9 | 53/36/19/11 | 67/49/27/15 | 89/61/34/16 |
| | DWT2 | 22/18/15/– | 38/29/26/– | 49/36/32/– | 62/48/35/– |
| | DFT1 | 27/26/19/ 15/13/0 | 62/5954/2262/1079 /215/49/23/16/0 | 74/7186/–/8522/9267 /–/2285/148/46/23/5 | 95/–/–/–/–/–/–/ –/–/231/61/33 |
| | DFT2 | 28/1029/640 /57/24/17/0 | 58/5183/–/6069/5471 /3946/335/54/14/2 | 67/7443/7904/8719/9251/ 9190/8404/–/1193/134/25/7 | 84/7788/9757/7437/9456/ 9979/–/–/–/–/2799/308/33 |
| | DCT1 | 29/23/19/ 13/8/0 | 59/80/102/66/ 44/29/16/8/0 | 74/98/213/221/236/ 238/116/50/25/13/4 | 93/140/231/242/238/280 /289/336/343/70/33/15 |
| | DCT2 | 27/34/32/ 23/19/15/0 | 57/67/114/103/110 /75/49/30/19/2 | 66/85/148/148/156/155 /165/191/102/47/24/7 | 87/108/155/165/165/182/ 198/211/231/261/169/69/30 |
| TFETI-2 | CE1/CE2/CE3 | 23/19/18 | 26/22/20 | 28/24/20 | 28/25/20 |
| | AE/AE+ME | 22/21 | 30/27 | 31/28 | 31/29 |
| | EIG | 25/24/22/19 /16/11/7/0 | 33/33/32/31/29/ 24/17/12/8/0 | 34/34/33/33/33/32/ 30/25/17/12/8/0 | 33/33/33/33/33/33/ 33/32/30/25/18/13/8 |
| | DWT1 | 22/19/15/10 | 31/25/17/11 | 32/26/18/11 | 32/28/18/11 |
| | DWT2 | 22/19/17/– | 29/26/25/– | 31/28/27/– | 31/29/27/– |
| | DFT1 | 25/25/21/ 19/16/13/0 | 33/34/33/33/ 31/23/21/19/0 | 34/34/34/34/34/ 34/33/24/22/20/2 | 34/34/34/34/34/34/ 34/34/33/28/23/22 |
| | DFT2 | 25/24/23/22 /21/18/10/0 | 33/33/33/33/32/ 29/29/27/16/0 | 34/34/34/34/34/34/ 33/30/30/30/20/2 | 33/33/33/33/33/33/ 33/33/33/30/30/30/25 |
| | DCT1 | 25/23/21/ 18/12/7/0 | 33/33/33/32/ 30/22/14/9/0 | 34/34/34/33/33/ 33/32/25/15/10/2 | 34/33/34/34/34/34/ 33/33/33/29/18/10 |
| | DCT2 | 25/24/23/21 /18/17/12/0 | 33/33/33/33/31/ 28/25/24/18/0 | 34/34/34/34/33/33/ 33/30/27/25/22/1 | 33/33/33/33/33/33/ 33/33/33/31/29/26/22 |

Table 4: Number of performed iterations of the (P)CG and DCG methods

In numerical experiments the following variants of deflation were tested:

- CE1 (CE with displacement equality conditions between corner nodes on the boundary of the domain not included in $W$), CE2 (CE with conditions between corner nodes on the domain boundary included), CE3 (only TFETI-2 – CE2 + Dirichlet BCs assigned in corner nodes on the domain boundary),
- AE (with displacement components of no corner nodes on the interface included into the computation of averages on the interface), AE+ME ((T)FETI-2 – with no corner nodes of interface included into averages, solitary forces of contributions of total gluing force concentrated into the corner nodes on the domain boundary in case of FETI-2, inside the domain in case of TFETI-2, not included)
- EIG1 (deflation by a given number of eigenvectors of the system matrix $A$),
- DWT1 and DWT2 (4/3/2/1 levels of 2D DWT applied on $A$ with and without the modification of $W$ with regard to solving 2D problem of elasticity),

- DFT1, DFT2 (deflation by first $M$ vectors of discrete Fourier basis with and without the modification of $W$ with regard to solving 2D problem of elastictity),
- DCT1, DCT2 (deflation by first $M$ vectors of discrete cosine basis with and without the modification of $W$ with regard to solving 2D problem of elastictity),

## 7. Conclusion

This paper provides experimental evidence of an effect of standard FETI-1 and TFETI-1 preconditioners and various types of deflation resulting in FETI-2 and TFETI-2 variants for a model 2D linear elasticity problem. This effect considering the numbers of iterations should always be taken into account with its costs. A detailed analysis in parallel environment is work in progress.

It should be mentioned that the benchmark 2D plane strain linear elasticity problem discretized by FEM on which the numerical experiments were performed was well-conditioned and thus the effect of the deflation was not that significant. If deflation were applied, for example, to a decomposed problem of linear elasticity with plates or shells, or to a decomposed problem without dualization, the effect of the deflation would be even more considerable. A more significant effect of deflation could also appear in the case of nonconforming and irregular subdomains' meshes resulting in a worse conditioned system matrix.

## Acknowledgements

This work was supported by the SGS grant No. SP2024/067 of VSB-Technical University of Ostrava.

## References

[1] Dostál, Z., Horák, D. and Kučera, R.: Total FETI - an easier implementable variant of the FETI method for numerical solution of elliptic PDE Commun. Numer. Meth. Engng **22** (2006), 1155–1162.

[2] Dostál, Z.: Rozděl a slep aneb jak řešit soustavu s bilionem lineárních rovnic. Pokroky matematiky, fyziky a astronomie **63** (2018), 28–40.

[3] Farhat, C., Mandel, J., and Roux, F.X.: Optimal convergence properties of the FETI domain decomposition method. Comput. Methods Appl. Mech. Engrg. **115** (1994), 365–385.

[4] Kružík, J.: *Implementation of the deflated variants of the conjugate gradient method*. Diploma thesis, VSB - Technical University of Ostrava, 2018.

[5] 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.

[6] Taswell, C., McGill, K.C., Algorithm 735: Wavelet transform algorithms for finite duration discrete-time signals ACM Trans. Math. Softw. **20** (1994), 398–412.

*Programs and Algorithms of Numerical Mathematics 22*
J. Chleboun, J. Papež, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics CAS, Prague 2025

# SPHERICAL RBF INTERPOLATION EMPLOYING PARTICULAR GEODESIC METRICS AND TREND FUNCTIONS

Karel Segeth

Institute of Mathematics, Czech Academy of Sciences
Prague, Czech Republic
segeth@math.cas.cz

**Abstract:** The paper is concerned with spherical radial basis function (SRBF) interpolation. We introduce particular SRBF interpolants employing several different geodesic metrics and a single trend function. Interpolation on a sphere is an important tool serving to processing data measured on the Earth's surface by satellites. Nevertheless, our model physical quantity is the magnetic susceptibility of rock measured in different directions. We construct a general SRBF formula and prove conditions sufficient for its existence. Particular formulae with specified geodesic metrics, trend and SRBFs are then constructed and tested on a series of magnetic susceptibility examples. The results show that this interpolation is sufficiently robust in general.

**Keywords:** radial basis function, spherical interpolation, spherical radial basis function, geodesic metric, trend, multiquadric, magnetic susceptibility

**MSC:** 65D05, 65D12, 65Z05

## 1. Introduction

In many geophysical applications there is a demand to compute an approximate representation of data measured on the sphere. We introduce a radial basis function (RBF) or spherical radial basis function (SRBF) interpolant in a real Euclidean space $\mathbb{R}^d$ for data measured at nodes on the $(d-1)$-dimensional surface of the unit sphere in $\mathbb{R}^d$ ([2], [10], [14]) in Section 2. Further we present sufficient conditions for the existence of such an interpolation formula.

Physical quantities measured on a sphere have brought an increasing interest with very advanced satellite technology of acquiring such data on the Earth surface. In the paper, the model physical quantity, having extensive applications, is different. It is concerned with the laboratory determined scalar physical data, the values of magnetic susceptibility of rock measured in different directions.

We introduce the spherical data interpolation formula and give sufficient conditions for its existence in Section 2. We describe the ways of approximating raw data

DOI: 10.21136/panm.2024.14

starting from the primary statistical treatment, important for the choice of the trend of the interpolation formula, in Section 3, see, e.g., [15]. We use a single trend in the formula, the second order polynomial in three Cartesian variables, that follows from these considerations and fits the data measured as well as possible.

Several geodesic metrics, functions necessary for the construction of spherical radial basis function interpolation, are considered in the paper, cf. [9], [10], [11]. We employ only one SRBF in the experiments presented, the multiquadric $\psi(r) = \sqrt{r^2 + c^2}$, see Sections 4 and 5. Further RBFs often used can be found in [2], [10], [12] etc.

The choice of a grid for the measurements performed is an important part of interpolation [1], [5], [6], [7]. An apparent drawback of the simplest grid equidistant in the spherical coordinates $\varphi$ and $\vartheta$ is considered in Section 7. In this section, numerical experiments employ as input the exact data given by the formula for trend, but perturbed randomly. The results given in Sections 6 and 7 show that the interpolation considered is sufficiently reliable.

## 2. Spherical data interpolation

We start with the notation necessary for introducing spherical data interpolation. Let $d$ be the dimension of a real Euclidean space $\mathbb{R}^d$. Then $S^{d-1} = \{x \in \mathbb{R}^d \mid \|x\| = 1\}$, where the norm $\|\cdot\|$ is Euclidean, is the $(d-1)$-*dimensional surface of unit sphere* in the $d$-dimensional space $\mathbb{R}^d$.

Further, a function $\sigma(x, y)$ of two variables $x, y \in \mathbb{R}^d$ is called *radial* if there exists a function $\tau(r)$, $r \geq 0$, such that $\sigma(x, y) = \tau(r)$, where $r \in \mathbb{R}$ is usually the Euclidean distance between $x$ and $y$ in case of non-spherical data.

Let $N$ and $M$ be integers, $N > 0$, $M \geq 0$, $N \geq M$, and $X = \{x_j\}_{j=1}^N$ be a set of mutually distinct *interpolation nodes* $x_j = (x_{j1}, x_{j2}, \ldots, x_{jd})$ on $S^{d-1}$. The real *spherical interpolant* $v$ for $x \in S^{d-1}$ is constructed as

$$v(x) = \sum_{j=1}^{N} a_j \psi(g(x, x_j)) + \sum_{k=1}^{M} b_k p_k(x), \tag{1}$$

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 is empty.

In the interpolant, $g$ is a nonnegative function called the *geodesic metric*, usually $g\colon S^{d-1} \times S^{d-1} \to [0, 1]$ is based on the angle between the radius vectors corresponding to the two arguments of $g$, see Section 3. Examples are given in Section 4. Further, $\psi\colon [0, 1] \to \mathbb{R}$ is a continuous real function, called the *radial basis function* (RBF) or *spherical radial basis function* (SRBF), and $p_k$ is a polynomial from $\Pi_t(\mathbb{R}^d)$, where $\Pi_t(\mathbb{R}^d)$ is the set of all polynomials (*trends*) $p\colon \mathbb{R}^d \to \mathbb{R}$ with real coefficients and of total degree less than or equal to some nonnegative integer $t$.

Let us formulate the interpolation problem to be solved. Given a continuous real *target function* $f\colon S^{d-1} \to \mathbb{R}$, find the *spherical interpolant*, i.e., a continuous function $v\colon S^{d-1} \to \mathbb{R}$ that satisfies the *interpolation conditions*

$$v(x_i) = f(x_i), \quad i = 1, \ldots, N, \tag{2}$$

where $f(x_i)$ is the value measured at the node $x_i$. Multiple measurements in a single direction $x_i$ with different results lead to a singular linear algebraic system for coefficients of the interpolation formula.

We confine ourselves only to real-valued functions and real data to make the exposition clearer. Substitute $x_i,\ i = 1, \ldots, N$, for $x$ in the formula (1) for $v$ to get

$$v(x_i) = \sum_{j=1}^{N} a_j \psi(g(x_i, x_j)) + \sum_{k=1}^{M} b_k p_k(x_i), \quad i = 1, \ldots, N,$$

and replace the left hand parts $v(x_i)$ of the interpolation conditions (2) with these expressions.

In the matrix notation, introduce an $N \times N$ symmetric square matrix $\Psi$ with the entries $\psi_{ij} = \psi(g(x_i, x_j)),\ i, j = 1, \ldots, N$, and an $N \times M$ matrix $P$ with the entries $p_{jk} = p_k(x_j),\ j = 1, \ldots, N,\ k = 1, \ldots, M$. Let $a \in \mathbb{R}^N$, $b \in \mathbb{R}^M$, and $f \in \mathbb{R}^N$ be the vectors of the unknowns and the vector of the right hand parts $f(x_i)$ of the interpolation conditions (2).

Note that if $M > 0$ then we have only $N$ interpolation conditions for $N+M$ interpolation coefficients $a_j$ and $b_k$ of the interpolant. Thus we can impose $M$ additional linear constraints for the individual trends $p_k$,

$$\sum_{j=1}^{N} a_j p_k(x_j) = \sum_{j=1}^{N} a_j p_{jk} = 0, \quad k = 1, \ldots, M.$$

Now the system of linear algebraic equations to be solved for the unknown vectors $a$ and $b$ reads

$$Q \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix}, \quad \text{where } Q = \begin{bmatrix} \Psi & P \\ P^{\mathrm{T}} & 0 \end{bmatrix} \tag{3}$$

is a symmetric $(N + M) \times (N + M)$ matrix of the system.

**Theorem 1.** *Let the $N \times N$ principal submatrix $\Psi$ of the $(N+M) \times (N+M)$ matrix $Q$ be positive definite and let* rank $P = M$. *Then the matrix $Q$ is nonsingular.*

*Proof.* The proof follows from Theorem 1 of [13]. □

In Theorem 1, we use the hypothesis that the matrix $\Psi$ is positive definite and rank $P = M$. However, in Micchelli [12] and many other sources, the condition that the spherical basis function $\psi$ is *conditionally (strictly) positive definite* is employed to prove that the matrix $Q$ is nonsingular.

A problem similar to data interpolation is *data smoothing (fitting)* but we are not concerned with that problem in this contribution.

## 3. Model problem

For a model problem, we have chosen the laboratory determination of raw susceptibility data, see, e.g., [8], [15]. The 3D rock sample rotates in magnetic field and the scalar data items $s_i$ measured in a set of selected directions $u_i$ are of the form

$$s_i = u_i^{\mathrm{T}} K u_i + e_i, \tag{4}$$

where $u_i$ is a unit vector in Cartesian coordinates in $\mathbb{R}^3$, whose initial point is at the origin and whose end point is on the unit sphere at $x_i$. $K$ is a tensor, and $e_i$ are deviations from the theoretical tensor model. Assuming the equation (4), we carry out linear regression and find an estimation of the tensor $K$. Then an appropriate rotation of the coordinate system can make the tensor $K$ diagonal with the *principal susceptibilities* $K_1, K_2, K_3$ on the diagonal.

We call the graphical representation of the directional susceptibilities the *lemniscate surface*, see Figure 1. Two-dimensional surfaces in $\mathbb{R}^3$ are depicted as endpoints of the corresponding vectors $s_i u_i$, as usual.

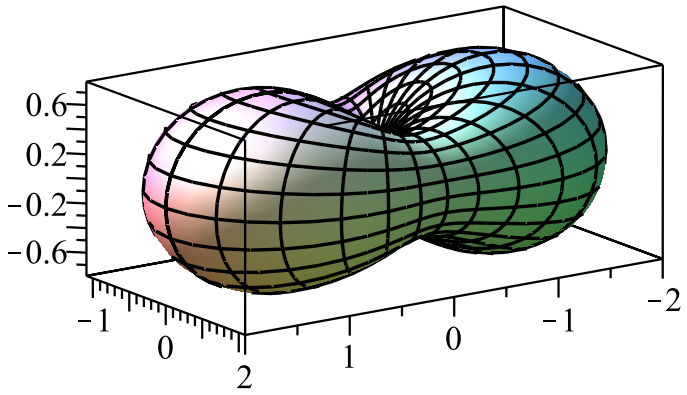



Figure 1: Model problem. Lemniscate surface $v$ obtained by the interpolation formula (10) (with interpolation conditions (2)) employing the theoretical values $f(x_i) = s_i$ given by (4) with principal susceptibilities $K_1 = 2.00$, $K_2 = 1.00$, $K_3 = 0.10$, and $e_i = 0$.

The magnitude of directional susceptibility in the $i$th direction $z_i$ is given by the distance between the origin and the surface measured along the vector $z_i$. The polynomial $s(z) = z^{\mathrm{T}} K z$ determined by the tensor $K$ is taken for the only trend in our further considerations, see Section 5.

## 4. Geodesic metric

Employing a RBF in the interpolation formula, we are supposed to define the distance between two nodes (i.e., between two unit vectors) $x$ and $y$ on the unit sphere $S^{d-1}$. The angle $\alpha$ of these two vectors is given as

$$\cos\alpha = x \cdot y,$$

where $x \cdot y$ is the inner product of two vectors from $\mathbb{R}^d$. Since $\cos\alpha = \cos(2\pi - \alpha)$ we can choose for computation either the angle $\alpha$ or its complement to $2\pi$, i.e. $2\pi - \alpha$. Only few geodesic metrics $g$ are used in practice. They usually satisfy $g\colon S^{d-1} \times S^{d-1} \to [0, 1]$.

The simplest geodesic metric is the angle $\alpha$ itself,

$$g_0(x, y) = \alpha/(2\pi) = \cos^{-1}(x \cdot y)/(2\pi). \tag{5}$$

Further two geodesic metrics, $g_1$ and $g_2$, are based on $\cos\alpha$, $\alpha \in [0, 2\pi]$. We put

$$g_1(x, y) = \sqrt{1 - \cos^2\alpha} = |\sin\alpha|. \tag{6}$$

Central symmetry of the data measured is expected when we apply the geodesic metric $g_1$. Every unit vector $x$ is considered as a part of an axis coming through the center of the sphere and from its two possible directions no direction is prescribed. Our quantity measured (magnetic susceptibility of rock) is just of this kind. If the angle $\alpha$ of two vectors $x$ and $y$ equals $\pi$ (i.e., $y = -x$) then the values measured on the sphere at $x$ and $y$ should be identical since the nodes $x$ and $-x$ of interpolation are not distinguished.

Therefore, in what follows, when using $g_1$, 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 $i, j = 1, \ldots, N$. The geodesic metric $g_1$ is periodic in $\alpha$ with the period $\pi$, and it holds $g_1(x, y) = 0$ for $\alpha = 0, \pi, 2\pi$.

The next geodesic metric considered is

$$g_2(x, y) = \sqrt{\tfrac{1}{2}(1 - \cos\alpha)}. \tag{7}$$

No symmetry of data measured is supposed when we employ the geodesic metric $g_2$. Apparently, $g_2$ is periodic in $\alpha$ with the period $2\pi$, and it holds $g_2(x, y) = 0$ for $\alpha = 0, 2\pi$.

## 5. A particular trend function

Let us turn back to our 3D problem introduced in Sec. 3. We take the second degree polynomial corresponding to (4), i.e.

$$s(z) = K_1 z_1^2 + K_2 z_2^2 + K_3 z_3^2, \quad z = (z_1, z_2, z_3) \in S^2, \tag{8}$$

where $K_1$, $K_2$, $K_3$ are known positive constants, for the only trend, i.e. $M = 1$.

Notice that the single argument of the SRBF function $\psi$ is from the interval $[0, 1]$ due to the geodesic metric, while the argument $z$ of the trend $s$ is from $S^2$.

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, in our case from (4). Description of the ideal field is then fitted by the trend part of the formula and the corrections resulting from the first, spherical part of the formula are only small.

## 6. The SRBF formula employed

We put $d = 3$ in our model problem, then $S^2$ is the usual two-dimensional unit sphere surface in the three-dimensional Euclidean space $\mathbb{R}^3$. Choose a fixed positive integer $N$ and put $M = 1$.

We take the *multiquadric*

$$\psi(r) = \sqrt{r^2 + c^2} \tag{9}$$

for the spherical radial basis function, where $r \in [0, 1]$ (the range of the geodesic function) and $c$ is a positive shape parameter.

Apparently, the trend $s$ given by (8) belongs to $\Pi_2(\mathbb{R}^3)$, which is the set of all polynomials $p\colon \mathbb{R}^3 \to \mathbb{R}$ of three variables with real coefficients and of total degree less then or equal to 2.

Consider the interpolation formula (1) in the form

$$v(x) = \sum_{j=1}^{N} a_j \psi(g(x, x_j)) + bs(x), \tag{10}$$

where $x, x_j \in S^2$, i.e., in the interpolation system (3), $P$ is a single column $N$-vector and $b$ and 0 are scalars.

We add a single constraint

$$\sum_{j=1}^{N} a_j s(x_j) = 0$$

to the interpolation conditions.

The following theorem is a particular case of Theorem 1 that covers our model problem.

**Theorem 2.** *Let the linear algebraic system* (3) *correspond to the interpolation formula* (10). *Let the block $P$ in the block matrix $Q$ have rank* 1. *Then the interpolation problem has the unique solution $a_j$, $j = 1, \ldots, N$, and $b$.*

*Proof.* It is known that the principal submatrix $\Psi$ of the block matrix $Q$ of the linear algebraic system (3) is positive definite when $\psi$ is an inverse multiquadric (Micchelli [12]). On the assumption that rank $P = 1$, the matrix $Q$ is nonsingular by Theorem 1 and the system has the unique solution $a_j$, $j = 1, \ldots, N$, and $b$. □

**Remark 1.** $P$ 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 $p_k = s(x_k)$ is nonzero.

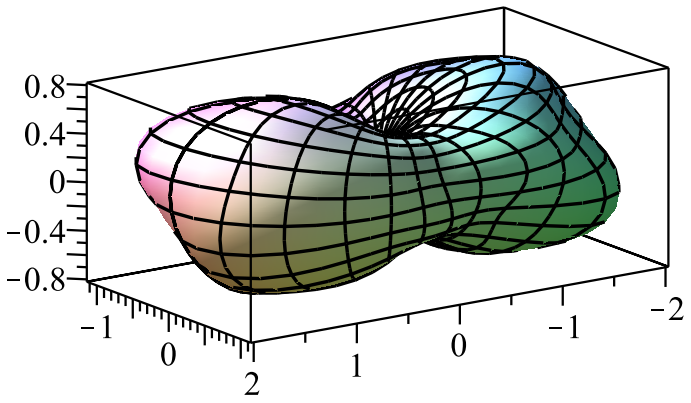



Figure 2: Lemniscate surface with $K_1 = 2.00, K_2 = 1.00, K_3 = 0.10$. The exact values given by (4) were multiplied by a random factor $\sigma$ from the range $[0.9, 1.1]$ resulting in the corresponding $e_i$. The geodesic metric $g_0$, $N = 74$, $c^2 = 0.25$.

## 7. Computational experiments

We have accomplished several series of computational experiments with the SRBF interpolation of the theoretical as well as perturbed theoretical lemniscate surfaces in the model problem with $d = 3$, where $S^2$ is the usual two-dimensional unit sphere surface in the three-dimensional Euclidean space. We have employed the SRBF interpolation formula (10) and different grids, geodesic metrics $g_0, g_1, g_2$, and several SRBF functions. See Figures 2, 3, 4.

The simplest grid used on a unit sphere is the grid equidistant in both the spherical coordinates $\varphi$ and $\vartheta$. The drawback of this grid is the fact that its nodes are dense in the vicinity of poles and sparse around the equator. For $g_1$, the interpolation nodes should satisfy the condition $x_j \neq \pm x_i$ mentioned above. The results presented in this paper have been computed in such grids.

Grids on a unit sphere are often used also for numerical integration. For interpolation, we have tried three such systems of grids: Bažant grids [1], Fibonacci grids [4], [6], and triangular grids stemming from an icosahedron [7], but we have found that they bring no significant advantage. A general treatment of data sampling on a unit sphere is provided in [5].

In literature (see, e.g., [2], [10], [12]), one can find several SRBFs $\psi$ known to provide a positive definite matrix $\Psi$ of (3). For example, the (direct) multiquadric (9), inverse multiquadric $1/\sqrt{r^2 + c^2}$, Gaussian function $\exp(-cr^2)$ or thin plate spline [3]. The results presented in this contribution have been computed with the direct multiquadric $\psi$ with a positive parameter $c$. The results may strongly depend on the constant $c$.

The resulting linear algebraic system (3) for the coefficients of the formula can be easily solved by the LU decomposition method for $N$ of order tens. For higher $N$, the system may be very ill-conditioned and special solution methods should be used. We apply, e.g., the Gauss-Jordan method.

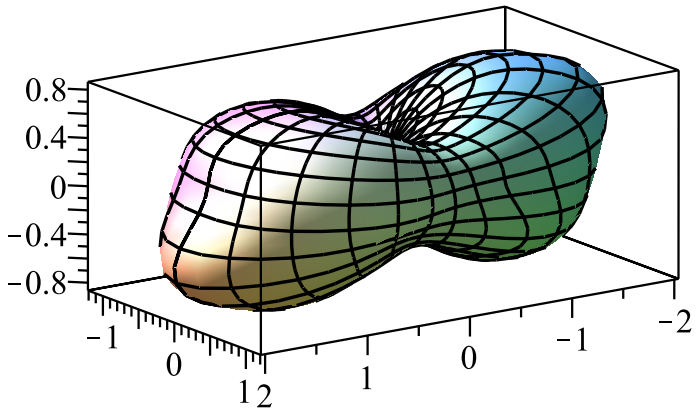



Figure 3: Lemniscate surface with $K_1 = 2.00, K_2 = 1.00, K_3 = 0.10$. The exact values given by (4) were multiplied by a random factor $\sigma$ from the range $[0.9, 1.1]$ resulting in the corresponding $e_i$. The geodesic metric $g_1$, symmetric grid and data, $N = 40$, $c^2 = 0.25$.

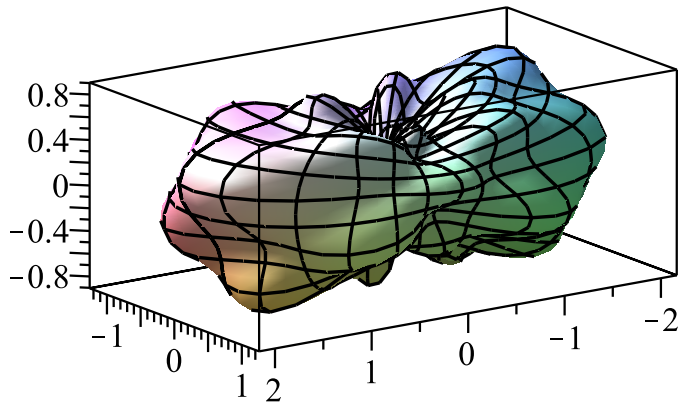



Figure 4: Lemniscate surface with $K_1 = 2.00, K_2 = 1.00, K_3 = 0.10$. The exact values given by (4) were multiplied by a random factor $\sigma$ from the range $[0.999, 1.001]$ resulting in the corresponding $e_i$. The geodesic metric $g_2$, $N = 74$, $c^2 = 0.25$.

## 8. Conclusions

We have carried out numerical tests with the interpolation formula (10), three geodesic metrics (5), (6) and (7), and SRBF (9). The formula performs efficiently and the results exhibit dependence on the parameter $c$. Further research shall provide a comparison of results obtained using various other SRBFs, e.g. thin plate splines, inverse multiquadrics, or the Gaussian function.

**Acknowledgements**

The author wishes to thank Professor Josef Ježek from the Faculty of Science of Charles University in Prague for all the interesting suggestions concerned with interpolation problems on sphere.

This work has been supported by the Institute of Mathematics of the Czech Academy of Sciences in Prague (RVO 67985840).

**References**

[1] Bažant, Z. P. and Oh, B. H.: Efficient numerical integration on the surface of a sphere. Z. Angew. Math. Mech. **66** (1986), 37–49.

[2] Buhmann, M. D.: *Radial basis functions.* Cambridge University Press, Cambridge, 2003.

[3] 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.

[4] Ehret, A. E., Itskov, M., and Schmid, J.: Numerical integration on the sphere and its effect on the material symmetry of constructive equations—A comparative study. Internat. J. Numer. Methods Engrg. **81** (2010), 189–206.

[5] Freeden, W., Nashed, M. Z., and Schreiner, M.: *Spherical sampling.* Birkhäuser, Basel, 2018.

[6] Hannay, J. H. and Nye, J. F.: Fibonacci numerical integration on a sphere. J. Phys. A **37** (2004), 11591–11601.

[7] Heikes, R. P., Randall, D. A., and Konor, C. S.: Optimal icosahedral grids: Performance of finite-difference operators and multigrid solver. Monthly Weather Rev. **141** (2013), 4450–4469.

[8] 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.

[9] Hubbert, S. and Baxter, B.: Radial basis functions for the sphere. In: W. Haussmann, K. Jetter, and M. Reimer (Eds.), *Recent Progress in Multivariate Approximation. Proc. of 4th International Conference 2000, International Series of Numerical Mathematics*, vol. 137, pp. 33–51. Birkhäuser, Basel, 2001.

[10] Hubbert, S., Lê Gia, Q. T., and Morton, T. M.: *Spherical radial basis functions, theory and applications.* Springer, Cham, 2015.

[11] Levesley, J., Luo, Z., and Sun, X.: Norm estimates of interpolation matrices and their inverses associated with strictly positive definite functions. Proc. Amer. Math. Soc. **127** (1999), 2127–2134.

[12] Micchelli, C. A.: Interpolation of scattered data: distance matrices and conditionally positive definite functions. Constr. Approx. **2** (1986), 11–22.

[13] Segeth, K.: Some computational aspects of smooth approximation. Computing **95** (Suppl. 1) (2013), 695–708.

[14] Segeth, K.: Spherical radial basis function approximation of some physical quantities measured. J. Comput. Appl. Math. **427** (2023), 115128.

[15] Tarling, D. H. and Hrouda, F.: *The magnetic anisotropy of rocks.* Chapman and Hall, London, 1993.

*Programs and Algorithms of Numerical Mathematics 22*
J. Chleboun, J. Papež, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics CAS, Prague 2025

# ON FLUID STRUCTURE INTERACTION PROBLEMS OF THE HEATED CYLINDER APPROXIMATED BY THE FINITE ELEMENT METHOD

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 study addresses the problem of the flow around circular cylinders with mixed convection. The focus is on suppressing the vortex-induced vibration (VIV) of the cylinder through heating. The problem is mathematically described using the arbitrary Lagrangian-Eulerian (ALE) method and Boussinesq approximation for simulating fluid flow and heat transfer. The fluid flow is modeled via incompressible Navier-Stokes equations in the ALE formulation with source term, which represent the density variation due to the change of temperature. The temperature is driven by the additional governing transport equation. The equations are numerically discretized by the finite element (FEM) method, where for the velocity-pressure couple the Taylor-Hood (TH) finite element is used and the temperature is approximated by the quadratic elements. The proposed solver is tested on benchmark problems.

**Keywords:** finite element method, Taylor-Hood element, arbitrary Lagrangian-Eulerian method, heated cylinder

**MSC:** 65N15, 65M15, 65F08

## 1. Introduction

The problem of flow around circular cylinders with mixed convection is of considerable importance in various engineering applications, such as flow in tubes, heat exchangers, nuclear reactor fuel rods, chimney stacks, cooling towers, etc. These applications involve critical engineering design parameters related to fluid flow, heat transfer, and vibration, which must be carefully considered, see [4].

This paper focuses on the suppression of vortex-induced vibration (VIV) of the cylinder by its heating. Over the years, numerous numerical and experimental studies have focused on investigating homogeneous or uniform flow around a circular cylinder that is movable in a vertical direction (see, e.g., [1,3]). In these studies, flow

DOI: 10.21136/panm.2024.15

behavior is primarily characterized by the Reynolds number ($Re$), and structure motion is always non-dimensional, where its stiffness is characterized by the reduced velocity $U_r$. The response of the system and its resonance is dependent only on these two variables, see [1]. However, if one considers the buoyancy forces, there is another non-dimensional parameter, called the Grashof number ($Gr$), which can be used for controlling the fluid flow and the structural response. For example, in [10] it is shown that for $Re = 100$ and for the $Gr \geq 1500$ the vortex shedding is stopped and the flow becomes steady state. An increase in the $Gr$ number also leads to an increase in the drag coefficient. Similar results were found in [11] where for $Re = 200$ the critical value $Gr = 12000$ was determined. The results of the heated movable cylinder can be found, e.g., in [14], where the critical $Gr$ number was defined to be dependent also on the reduced velocity.

This paper focuses on a numerical simulation of the VIV problems of the cylinder leading to suppression of the vibrations, a description of such strategies can be found in [4]. A simplified model of incompressible fluid with buoyancy forces is considered, however, for such a model still several numerical challenges, such as managing the incompressibility constraint, nonlinear convective terms and coupling between the additional transport equation of the temperature with the momentum equations need to be addressed (see, e.g., [10]). The model needs to treat the time-dependent computational fluid domain, which is usually handled using the arbitrary Lagrangian-Eulerian (ALE) method, see e.g., [13]. To describe the fluid flow influenced by the heat transfer, the Boussinesq approximation is used. The mathematical model consists of the incompressible Navier-Stokes equations with a right-hand side term depending on the temperature. The temperature is described by an additional transport equation. For the approximation of the system of incompressible Navier-Stokes equations in the ALE formulation, the Taylor-Hood (TH) finite element is used. This choice of the velocity-pressure pair satisfies the Babuška-Brezzi (BB) inf-sup condition, which guarantees the stability of the numerical scheme, see [8]. The temperature is approximated by continuous piecewise quadratic functions.

The proposed method is tested on two benchmark problems. The first involves the flow around a fixed heated cylinder, where the critical Grashof number and mean drag coefficient are compared with the data from [10]. In the second test case, the suppression of vibration of a moving cylinder is addressed by its heating, the response is compared with the findings of [14].

## 2. Governing equations

In this section, the mathematical model of the fluid flow around the heated moving cylinder is given, where the density changes due to the temperature described by the Boussinesq approximation. The model consists of the incompressible Navier-Stokes equations in the ALE formulation coupled with the convection-diffusion equation for the temperature.

Let $\Omega_t \subset \mathbb{R}^2$ be a bounded computational time-dependent fluid domain with

a continuous Lipschitz boundary, which is composed of three disjoint segments: $\partial\Omega = \Gamma_D \cup \Gamma_O \cup \Gamma_{W_t}$. The domain $\Omega_t$ is assumed to be polygonal, and completely filled with fluid at any time $t \in (0, T_\infty)$. The movability of the domain is treated via the Arbitrary Lagrangian-Eulerian (ALE) formulation. The ALE method uses a mapping $A_t$ that transforms the reference domain $\Omega_0$ on the current domain $\Omega_t$, i.e.,

$$A_t \colon \Omega_{\text{ref}} \to \Omega_t, \quad X \mapsto x(X,t) = A_t(X), \quad x \in \Omega_{\text{ref}},\ t \in (0, T_\infty]),$$

moreover transforms the reference interface $\Gamma_{W_0}$ on the current interface $\Gamma_{W_t}$ based on the movement of the cylinder, while the other boundaries remain stationary. For further details, see [13].

For computation, the non-dimensional Navier-Stokes (NS) equations for incompressible flow and the thermal equation in the ALE formulation are used. Firstly, all lengths are characterized by the cylinder diameter $D$, the flow velocities $\boldsymbol{u} = (u_1, u_2)$ are scaled by the free stream velocity $U_{\text{ref}}$, the time is scaled by the factor $D/U_{\text{ref}}$, and the kinematic pressure is scaled by $\rho U_{\text{ref}}^2$, where $\rho$ is the fluid density. In addition, the non-dimensional temperature is given by $\theta = (T - T_{\text{ref}})/(T_s - T_{\text{ref}})$, where $T$ represents fluid temperature, $T_{\text{ref}}$ is the temperature of the free stream, and $T_s$ is the temperature of the cylinder. For simplicity, in the rest of the paper, all of the quantities are dimensionless. The nondimensional form of the NS equations with the transport temperature equation read: Find the velocity $\boldsymbol{u}(x,t) \colon \Omega_t \to \mathbb{R}^2$, the pressure $p(x,t) \colon \Omega_t \to \mathbb{R}$, and the temperature $\theta(x,t) \colon \Omega_t \to \mathbb{R}$ which satisfy

$$\begin{aligned} \frac{D^A}{Dt}\boldsymbol{u} + [(\boldsymbol{u} - \boldsymbol{w}) \cdot \boldsymbol{\nabla}]\boldsymbol{u} - \frac{1}{Re}\boldsymbol{\Delta}\boldsymbol{u} + \boldsymbol{\nabla} p &= \frac{Gr}{Re^2}\theta && \text{in } \Omega_t, t \in (0, T_\infty], \\ \boldsymbol{\nabla} \cdot \boldsymbol{u} &= 0 && \text{in } \Omega_t, t \in (0, T_\infty], \\ \frac{D^A}{Dt}\theta + [(\boldsymbol{u} - \boldsymbol{w}) \cdot \boldsymbol{\nabla}]\theta - \frac{1}{RePr}\boldsymbol{\Delta}\theta &= 0 && \text{in } \Omega_t, t \in (0, T_\infty], \end{aligned} \tag{1}$$

where $\frac{D^A}{Dt}$ denotes the ALE derivative, and $\boldsymbol{w} = \partial A^t / \partial t$ represents the domain velocity, see [2,13]. The $Re$, $Pr$, and $Gr$ are the Reynolds, Prandtl and Grashof numbers respectively, given as $Re = U_{\text{ref}} D/\nu$, $\text{Pr} = \nu/\kappa$, and $Gr = g\beta\Delta T D^3/\nu^2$, where $\nu$ is the kinematic viscosity, $\kappa$ is the thermal diffusivity, $\Delta T$ is the temperature difference ($\Delta T = T_s - T_{\text{ref}}$), $\beta$ means the thermal expansion coefficient and $g$ is the gravitational acceleration (in this paper acting in the horizontal direction), see [14]. This approximation of the flow problem around the heated cylinder is valid for approximately $\beta\Delta T \leq 0.01$, see [10].

To close problem (1), the following conditions are added: initial condition

$$\boldsymbol{u}(x,0) = \boldsymbol{u}_0, \quad \theta(x,0) = \theta_0 \quad \text{in } \Omega_0, \tag{2}$$

and the boundary conditions

$$\boldsymbol{u}(x,t) = \boldsymbol{g}(x,t), \quad \theta(x,t) = q(x,t) \quad \text{on } \Gamma_D \times (0, T_\infty], \tag{3a}$$

$$\boldsymbol{u}(x,t) = \boldsymbol{w}(x,t), \quad \theta(x,t) = q(x,t) \quad \text{on } \Gamma_{W_t},\ t \in (0, T_\infty], \tag{3b}$$

$$\frac{\partial \theta}{\partial n} = 0 \quad \text{on } \Gamma_O \times (0, T_\infty], \tag{3c}$$

$$-(p - p_{\text{ref}})\boldsymbol{n} + \frac{1}{Re}\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}} = 0 \quad \text{on } \Gamma_O \times (0, T_\infty], \tag{3d}$$

where $\boldsymbol{n}$ represents the unit outward normal vector to $\partial\Omega_t$ and $p_{\text{ref}}$ represents a reference pressure value at the outlet. Here, Eq. (3a) is the no-slip condition, (3b) reflects the assumption that the fluid remains attached to the cylinder, (3c) is the Neumann condition, and (3d) is the so-called do-nothing condition, see [6].

### 2.1. Motion of the cylinder

In this paper, a simplified model can be used as the cylinder is movable only in the vertical direction. Therefore, the ordinary differential equation (ODE) for the displacement $Y$, its velocity $\dot{Y}$ and acceleration $\ddot{Y}$ in non-dimensional form are

$$\ddot{Y} + \left(\frac{4\pi\xi}{U_r}\right)\dot{Y} + \left(\frac{4\pi^2}{U_r^2}\right)Y = \frac{C_l}{2M^*}, \tag{4}$$

where $\xi$ symbolizes the structural damping ratio, $U_r = \frac{U_\infty}{f_n D}$ is the reduced velocity of the cylinder (with $f_n$ representing the natural frequency of the cylinder), $M^*$ indicates for the reduced mass of the rigid cylinder ($M^* = \frac{m}{\rho D^2}$), and $C_l = \frac{L}{1/2\rho U_\infty^2 A}$ is the lift coefficient (here $L$ represents the lift force), see [1, 14].

## 3. Discretization of the fluid flow problem

In order to discretize problem (1) by the finite element method (FEM), the weak formulation has to be introduced. First, a constant time step $\Delta t > 0$ is taken, and the time interval $(0, T_\infty)$ is equidistantly divided into time intervals $(t_n, t_{n+1})$ with $t_n = n\Delta t$. Further, the velocity, pressure and the temperature are approximated at time step $t_n \in (0, T_\infty]$ by $\boldsymbol{u}^n(x) \approx \boldsymbol{u}(x, t_n)$ for $x \in \Omega_{t_n}$, $p^n(x) \approx p(x, t_n)$ for $x \in \Omega_{t_n}$, and $\theta^n(x) \approx \theta(x, t_n)$ for $x \in \Omega_{t_n}$. The velocity of the domain at the instant $t_{n+1}$ is approximated by $\boldsymbol{w}^{n+1}(x) \approx \boldsymbol{w}(x, t_{n+1})$ for $x \in \Omega_{t_{n+1}}$ and the ALE derivative is approximated at fixed time instance $t_{n+1}$ by the second-order two-step backward difference formula (BDF2). Hence, the implicit scheme is given

$$\begin{aligned}
\frac{3\boldsymbol{u}^{n+1} - 4\tilde{\boldsymbol{u}}^n + \tilde{\boldsymbol{u}}^{n-1}}{2\Delta t} + ((\boldsymbol{u}^{n+1} - \boldsymbol{w}^{n+1}) \cdot \boldsymbol{\nabla})\boldsymbol{u}^{n+1} - \frac{1}{\text{Re}}\boldsymbol{\Delta}\boldsymbol{u}^{n+1} + \boldsymbol{\nabla} p^{n+1} &= \frac{Gr}{Re^2}\theta, \\
\boldsymbol{\nabla} \cdot \boldsymbol{u}^{n+1} &= 0, \\
\frac{3\theta^{n+1} - 4\tilde{\theta}^n + \tilde{\theta}^{n-1}}{2\Delta t} + ((\boldsymbol{u}^{n+1} - \boldsymbol{w}^{n+1}) \cdot \boldsymbol{\nabla})\theta^{n+1} - \frac{1}{\text{RePr}}\boldsymbol{\Delta}\theta^{n+1} &= 0,
\end{aligned}$$

where $\tilde{\boldsymbol{u}}^i$ and $\tilde{\theta}^i$ denotes the transformation of $u^i$ and $\theta^i$ from $\Omega_i$ onto $\Omega_{n+1}$, i.e., $\tilde{\boldsymbol{u}}^i = \boldsymbol{u}^i \circ A_{t_i} \circ A_{t_{n+1}}^{-1}$.

### 3.1. Spatial discretization by the FEM

In this section, the FEM discretization of the semi-discrete problem (5) is introduced in the standard way. Firstly, a weak formulation is provided. Let us assume the fixed time instance $t_{n+1}$, and present a simplified notation: $\boldsymbol{u} = \boldsymbol{u}^{n+1}$, $\boldsymbol{w} = \boldsymbol{w}^{n+1}$, $p = p^{n+1}$, and $\Omega = \Omega_{t_{n+1}}$.

Furthermore, the velocity test space $\boldsymbol{\mathcal{V}}$, the pressure test space $\mathcal{Q}$ and the temperature test space $\mathscr{T}$ are defined as

$$\begin{aligned}\boldsymbol{\mathcal{V}} &= \left\{\boldsymbol{\varphi} \in \boldsymbol{H}^1(\Omega)\, \boldsymbol{\varphi}(x) = 0\ \forall x \in \Gamma_D \cup \Gamma_W\right\}, \quad \mathcal{Q} = L^2(\Omega),\\ \mathscr{T} &= \left\{\varphi \in H^1(\Omega)\, \varphi(x) = 0\ \forall x \in \Gamma_D \cup \Gamma_W\right\},\end{aligned}$$

where $\boldsymbol{H}^1(\Omega) = [H^1(\Omega)]^2$ is the vector Sobolev space and $L^2(\Omega)$ is the Lebesgue space, see [9].

Using some mathematical operation and proposing the notation of the scalar product $(\boldsymbol{u}, \boldsymbol{v})_\Omega = \int_\Omega \boldsymbol{u} \cdot \boldsymbol{v}\, \mathrm{d}x$ in $L^2(\Omega)$ and of the trilinear form $c(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}) = \int_\Omega [(\boldsymbol{u} \cdot \nabla)\boldsymbol{v}] \cdot \boldsymbol{w}\ \mathrm{d}x$, the weak formulation reads: Find $U = (\boldsymbol{u}, p.\theta) \in \boldsymbol{\mathcal{V}} \times \mathscr{Q} \times \mathscr{T}$ such that the equation

$$a\left(\boldsymbol{u}, U, V\right) + a_\theta\left(\boldsymbol{u}, U, V\right) = F(V) + F_\theta(V), \tag{5}$$

holds for any test function $V = (\boldsymbol{v}, q, \zeta) \in \boldsymbol{\mathcal{V}} \times \mathscr{Q} \times \mathscr{T}$, where

$$\begin{aligned} a\left(U, V\right) &= \frac{3}{2\Delta t}(\boldsymbol{u}, \boldsymbol{v})_\Omega + \frac{1}{\mathrm{Re}}(\nabla \boldsymbol{u}, \nabla \boldsymbol{v})_\Omega + c(\boldsymbol{u} - \boldsymbol{w}, \boldsymbol{u}, \boldsymbol{v}) - (p, \nabla \cdot \boldsymbol{v})_\Omega - (\nabla \cdot \boldsymbol{u}, q)_\Omega,\\ F\left(V\right) &= \frac{1}{2\Delta t}(4\tilde{\boldsymbol{u}}^n - \tilde{\boldsymbol{u}}^{n-1}, \boldsymbol{v})_\Omega + \frac{Gr}{Re^2}(\theta, \boldsymbol{v})_\Omega, \end{aligned} \tag{6}$$

and

$$\begin{aligned} a_\theta\left(\boldsymbol{u}, \theta, \zeta\right) &= \frac{3}{2\Delta t}(\theta, \zeta)_\Omega + \frac{1}{RePr}(\nabla\theta, \nabla\zeta)_\Omega + ((\boldsymbol{u} \cdot \nabla)\theta, \zeta)_\Omega,\\ F_\theta(\zeta) &= \frac{1}{2\Delta t}(4\tilde{\theta}^n - \tilde{\theta}^{n-1}, \zeta)_\Omega. \end{aligned} \tag{7}$$

For a more detailed description see [8].

In addition, the admissible triangulation $\tau_h$ of the domain $\Omega$ is considered (see [5]) and in this triangulation, the following finite element (FE) subspaces are used: $\boldsymbol{\mathcal{V}}_h \subset \boldsymbol{\mathcal{V}}$ as the velocity subspace, $\mathcal{Q}_h \subset \mathcal{Q}$ as the pressure subspace, and $\mathscr{T}_h \subset \mathscr{T}$ as the temperature subspace. Generally, finite element subspaces consist of piecewise polynomial functions. In this paper, the velocity and the pressure are discretized by the so-called Taylor-Hood element which leads to the following function spaces

$$\boldsymbol{\mathcal{V}}_h = \left\{\boldsymbol{\varphi} \in \mathbf{C}(\overline{\Omega})\, (\boldsymbol{\varphi}\big|_K \in P_2(K), \forall K \in \tau_h)\right\} \cap \boldsymbol{\mathcal{V}}, \tag{8}$$

$$\mathcal{Q}_h = \left\{\varphi \in C(\overline{\Omega})\, (\varphi\big|_K \in P_1(K), \forall K \in \tau_h)\right\}. \tag{9}$$

The temperature is discretized by the piecewise quadratic functions

$$\mathscr{T}_h = \left\{\varphi \in \mathbf{C}(\overline{\Omega})\,(\varphi\big|_K \in P_2(K), \forall K \in \tau_h)\right\} \cap \mathscr{T}. \tag{10}$$

Then, the discrete problem reads: Find $U_h = (\boldsymbol{u}_h, p_h, \theta_h) \in \boldsymbol{\mathcal{V}}_h \times \mathscr{Q}_h \times \mathscr{T}_h$ such that the equations

$$a_h\,(U_h, V_h) + a_\theta\,(\boldsymbol{u}_h, \theta_h, \zeta_h) = F(V_h, \theta_h) + F_{\theta_h}(\zeta_h) \tag{11}$$

hold for any test function $V_h = (\boldsymbol{v}_h, q_h, \zeta_h) \in \boldsymbol{\mathcal{V}}_h \times \mathscr{Q}_h \times \mathscr{T}_h$ and satisfies the boundary conditions (3a)–(3c).

## 4. Numerical simulations

In this section, the results of numerical simulations are discussed, such as the problem of flow around the fixed heated cylinder and flow around the heated cylinder with one degree of freedom in the cross direction. The domain of the problem is shown in Figure 1. The fluid flow around the heated cylinder is modeled using Eqs. (1), which are incompressible Navier-Stokes equations, incorporating the Boussinesq approximation to account for temperature variations. For the fixed case the $\Gamma_{W_t}$ remains stationary while in the problem with vibrations, it can move in the vertical direction.

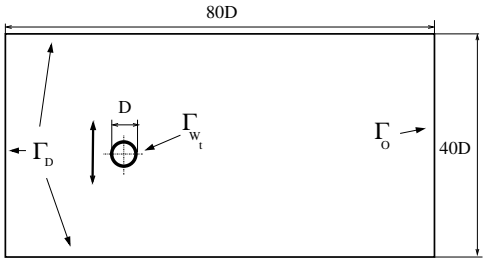



Figure 1: Domain of the flow around the cylinder.

### 4.1. Flow around the heated cylinder

The first test case is the flow around the fixed cylinder. The boundary conditions include the Dirichlet boundary conditions on $\Gamma_D$. The $\Gamma_{W_t}$ incorporates the movable surface (for a simple case without moving, the surface is fixed). At the outlet $\Gamma_O$, the so-called do-nothing condition is used for the velocity and pressure (see [12]). The temperature is subject to Dirichlet boundary conditions in the free stream $\Gamma_{D,1}$ and at the $\Gamma_{W_t}$, while a Neumann boundary condition is applied at the outlet $\Gamma_O$. The domain size was selected based on [11], and the size of the mesh was limited by the solver, which the UMFPACK library provides, and it performs efficiently up to 200000 DoFs.

Calculations were performed for various scenarios involving different Grashof numbers, with Reynolds numbers $Re = 100$ and $Re = 200$. The critical $Gr$ number

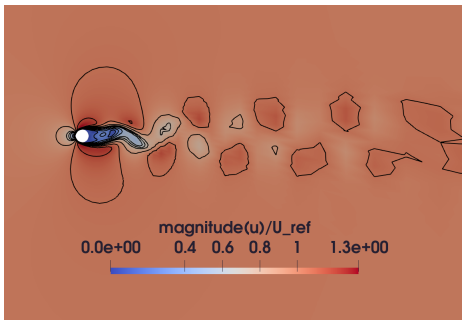



Figure 2: Magnitude of the velocity ($||\boldsymbol{u}||_\infty$) for the $Gr = 1000$.

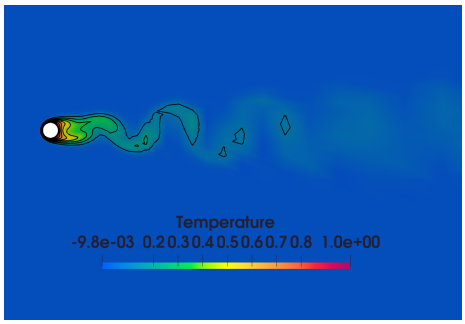



Figure 3: Temperature field $\theta$ for the $Gr = 1000$.

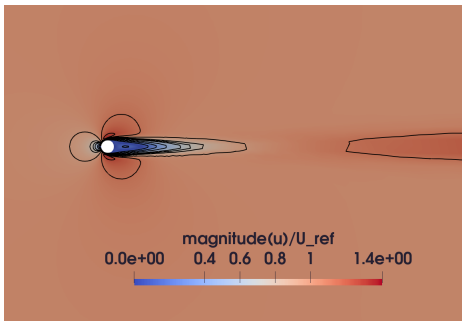



Figure 4: Magnitude of the velocity ($||\boldsymbol{u}||_\infty$) for the $Gr = 1500$.

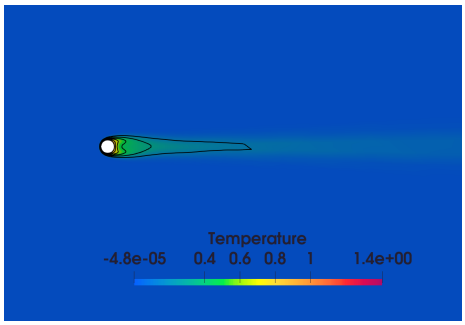


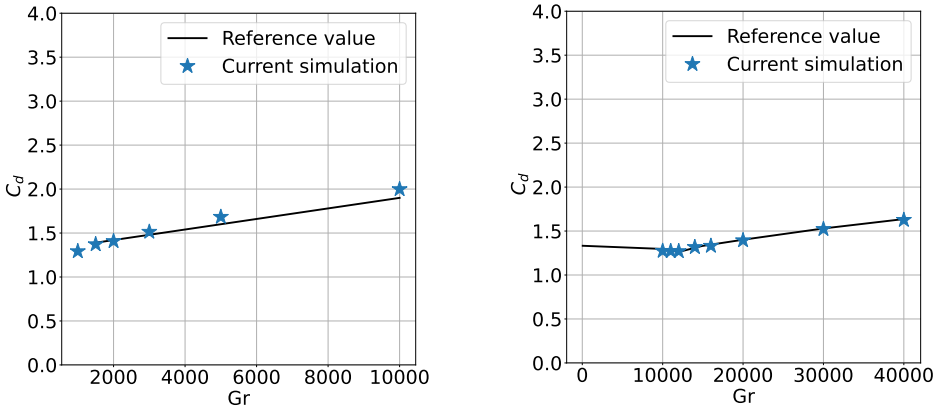

Figure 5: Temperature field $\theta$ for the $Gr = 1500$.

(the lowest number where the flow becomes steady) is compared with [10] and the drag coefficients for various scenarios are also compared. Figures 2–5 show the results for four cases with $Re = 100$. As the Grashof number increases, the flow gradually stabilizes until it reaches a critical Grashof number ($Gr = 1500$), after which the flow is nearly steady. This corresponds to [10]. Similar trends were observed for $Re = 200$, although the critical Grashof number is higher ($Gr = 15000$), probably due to the insufficient quality of the mesh. Despite this, the mean drag coefficient for both cases is aligned well with the reference data [10, 11], see Figure 6.

### 4.2. Flow around the movable heated cylinder

The initial state of the domain, denoted as $\Omega_t$, is in Figure 1 with heated cylinder. The boundary conditions are similar to the previous problem, and due to the movement of the cylinder, the Dirichlet boundary condition is $\boldsymbol{u} = \boldsymbol{w}$. Its position is obtained by solving the problem (4) using the 4-th order Runge-Kutta method. The coupling procedure between the cylinder and the fluid flow is performed using a strong coupling algorithm, which is well described in [7]. The mesh movement is realized by the pseudo-elastic approach, which is described, e.g., in [7].

The flow problem around the cylinder is characterized by the Reynolds number Re = 150, aligned with the reference data from [14]. The model of a movable cylinder




a) b)

Figure 6: Drag coefficient for different $Gr$ numbers for $Re = 100$ a), and for $Re = 200$ b), compared to the reference data from [10, 11].

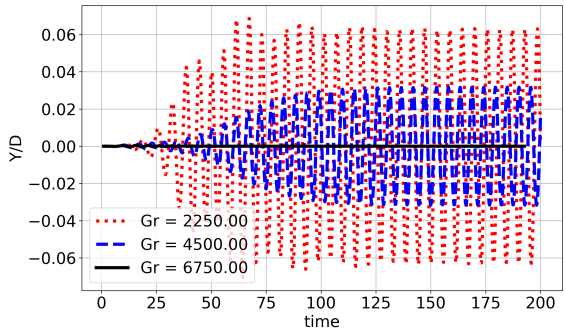



Figure 7: Displacement $Y/D$ in dependent of time for the case $U_r = 8$ and $Re = 150$.

without heating is well described in [1]. A vortex street forms in the wake of the cylinder, leading to oscillations in aerodynamic forces, which in turn induce the vibration of the cylinder. The interval in which the resonance occurs is $U_r \in [4, 8]$. The highest amplitude is for $U_r = 4$ and then the amplitude decreases with increasing $U_r$, see [1].

Our goal is to suppress the vibration by heating and stabilizing the flow. In Figure 7, the displacement $Y$ is given for $U_r = 8$, with zero damping $\xi = 0$, and $M^* = 2$.

It can be observed, that as the $Gr$ is increased (we add more and more heating), the vortex shedding is stabilized until we reach the $Gr = 6750$ and we reach an almost steady state. This result corresponds to [14].

## 5. Conclusion

In this paper, the problem of the interaction between incompressible flow and a heated cylinder with one degree of freedom is analyzed using numerical simulations. The main goal was to suppress the flow-induced vibrations (VIV) by its heating.

The problem was mathematically described as the incompressible fluid which is approximated by the incompressible Navier-Stokes (NS) equations, where to take into account the density dependence on the temperature, the Boussinesq approximation was used. As a result, a source term depending on the temperature is included in the NS equations, where the temperature is modeled by an additional transport equation. For time discretization the backward-difference formula of second order (BDF2) is used, whereas for space discretization the finite element method (FEM) is utilized. The velocity and pressure are discretized by the Taylor-Hood (TH) element, while the temperature is discretized by the piecewise quadratic functions. The numerical results of the developed in-house solver are presented and compared with the reference data of [10, 14].

For the first case of the flow around the fixed cylinder, it was confirmed that the stability of the flow is dependent in addition to the Reynolds number ($Re$) also on the Grashof number ($Gr$). It was observed that for $Re = 100$ the critical $Gr$ number is $Gr = 1500$, which is in agreement with [10]. For the case, $Re = 200$, the obtained critical $Gr$ number ($Gr = 15000$) is larger than the value $Gr = 12000$ found in [11]. This is probably due to the use of a not sufficiently refined mesh, where the applied solver is limited by the number of unknowns from the UMFPACK library. In addition, the dependence of the drag coefficient on the Grashof number was compared with the reference data from [10, 11]. It was shown that with an increase in the $Gr$ number, the drag coefficient also increases. Our simulations slightly overestimated the drag for $Re = 200$, it might again be attributed to insufficient quality of the mesh.

The second case was the flow around a vibrating cylinder, whose vibrations are described using one degree of freedom (vertical displacement). The structural movement is characterized by the reduced velocity $U_r$. It is shown that for one case of reduced velocity (i.e., $U_r = 8$) the amplitude of the response is lowered with an increase of the $Gr$ number. Such a decrease of vibrations continues with further increase up to the critical $Gr$ number $Gr = 6750$, for which an almost steady state is obtained. This is also in agreement with the findings in [14].

It was shown that the presented results of the developed in-house numerical solver agree with the reference data. Further, the numerical results showed that the heating of the cylinder can lead to the suppression of the VIV of the cylinder. The main limitation of the presented solver is that it can solve only small systems due to the UMFPACK library used as a solver. This problem can be addressed, e.g., by domain decomposition.

## Acknowledgements

The authors acknowledge the support by the Grant Agency of the Czech Technical University in Prague, grant No. SGS SGS22/148/OHK2/3T/12, and grant No. SGS SGS24/120/OHK2/3T/12. Karel Vacek has also been supported by the Czech Science Foundation (GAČR) project 22-01591S. The Institute of Mathematics of the CAS 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. J. Comput. Phys. **219** (2006), 671–696.

[2] Alsabery, A., Sheremet, M., Ghalambaz, M., Chamkha, A., and Hashim, I.: Fluid-structure interaction in natural convection heat transfer in an oblique cavity with a flexible oscillating fin and partial heating. Applied Thermal Engineering **145** (2018), 80–97.

[3] 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. J. Fluids Struct. **35** (2012), 50–75.

[4] Chen, W. L., Huang, Y., Chen, C., Yu, H., and Gao, D.: Review of active control of circular cylinder flow. Ocean Engineering **258** (2022), 111 840.

[5] Ciarlet, P. G.: *The finite element method for elliptic problems*. Society for Industrial and Applied Mathematics, 2002.

[6] Feistauer, M.: *Mathematical methods in fluid dynamics*. 67, Chapman and Hall/CRC, 1993.

[7] Feistauer, M., Horáček, J., Růžička, M., and Sváček, P.: Numerical analysis of flow-induced nonlinear vibrations of an airfoil with three degrees of freedom. Comput. Fluids **49** (2011), 110–127.

[8] Girault, V. and Raviart, P.: *Finite Element Methods for Navier-Stokes Equations: Theory and Algorithms*. Computational Mathematics Series, Springer-Verlag, 1986.

[9] Kufner, A., John, O., and Fucik, S.: *Function spaces*. Mechanics: Analysis, Springer Netherlands, 1977.

[10] Patnaik, B. V., Narayana, P. A., and Seetharamu, K.: Numerical simulation of vortex shedding past a circular cylinder under the influence of buoyancy. Int. J. Heat Mass Transfer **42** (1999), 3495–3507.

[11] Salimipour, E.: A numerical study on the fluid flow and heat transfer from a horizontal circular cylinder under mixed convection. Int. J. Heat and Mass Transfer **131** (2019), 365–374.

[12] Sváček, P., Feistauer, M., and Horáček, J.: Numerical simulation of flow induced airfoil vibrations with large amplitudes. J. Fluids Struct. **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. Comput. Methods in Appl. Mech. Eng. **95** (1992), 115–138.

[14] Wan, H. and Patnaik, S. S.: Suppression of vortex-induced vibration of a circular cylinder using thermal effects. Phys. Fluids **28** (2016).

*Programs and Algorithms of Numerical Mathematics 22*
J. Chleboun, J. Papež, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics CAS, Prague 2025

# SIMPLIFIED MATHEMATICAL MODELS OF FLUID-STRUCTURE-ACOUSTIC INTERACTION PROBLEM MOTIVATED BY HUMAN PHONATION PROCESS

Jan Valášek[1,2], 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:** Human phonation process represents an interesting and complex problem of fluid-structure-acoustic interaction, where the deformation of the vocal folds (elastic body) are interplaying with the fluid flow (air stream) and the acoustics. Due to its high complexity, two simplified mathematical models are described – the fluid-structure interaction (FSI) problem describing the self-induced vibrations of the vocal folds, and the fluid-structure-acoustic interaction (FSAI) problem, which also involves aeroacoustic phenomena. The FSI model is based on the incompressible Navier-Stokes equations in the ALE formulation coupled with the linear elasticity model. Both the fluid and structural models are approximated using finite element methods, and the influence of different inlet boundary conditions is discussed in detail. For the FSAI model, an aeroacoustic hybrid approach is used, incorporating the Lighthill analogy or the perturbed convective wave equation. The acoustic results strongly depend on the proper choice of the computational acoustic domain (i.e. vocal tract model). Further, the spatial and frequency distributions of sound sources computed from the FSI solution are compared for both used approaches. The final frequency spectra of acoustic pressure at the mouth position are also analyzed for both approaches.

**Keywords:** human phonation, flow-induced vibrations, Navier-Stokes equations, aeroacoustic analogy, flutter instability, finite element method.

**MSC:** 65M60, 74F10, 76Q05.

## 1. Introduction

The basic sound of human phonation is created by an airstream (fluid flow) pouring through a channel constricted by vibrating elastic vocal folds (VFs), naturally

DOI: 10.21136/panm.2024.16

leading to fluid-structure interaction (FSI) problem, see [17]. Moreover, both the involved physical fields also interact with the acoustic field and we speak in general about fluid-structure-acoustic interaction (FSAI) problem, [7, 15], see Figure 1. The acoustic interaction occurs in two primary ways: the resonant frequencies of the acoustic domain dominate the output signal as other frequencies are less distinct, [17, 14], and acoustic waves can influence VF vibration patterns (as the major sound source mechanism). This can occur particularly under high sound pressure levels (SPL), as observed in loud singing [23], or during phonation into a length-adjusted tube used in voice therapy [4].

In this paper the modelling of human phonation during the normal speech is considered, where the source-filter theory ([17, 14]) can be utilized (due to low acoustic SPL) neglecting any acoustic influence on the VF vibration. This allows us to decouple the acoustic problem from the FSI problem and to use the hybrid approach of aeroacoustic analogies, see Figure 1 on the right. The acoustic problem, treated as a post-processing task after the FSI simulation, can be solved with a different solver, offering many advantages, see [7]. This simplified FSAI model, [13], is the primary focus here.

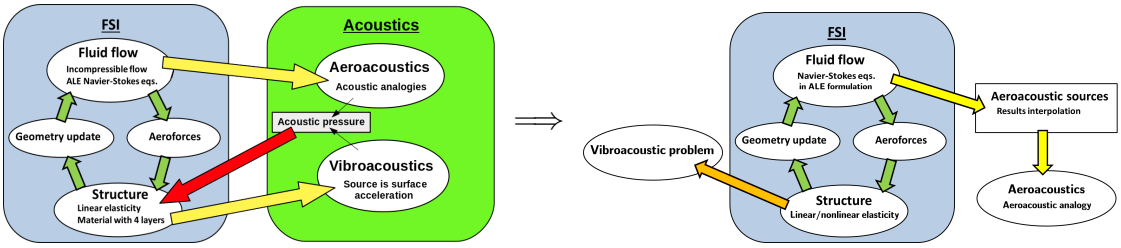



Figure 1: Dependence of physical fields in FSAI problem and its possible simplification, where the acoustic influence on the FSI problem is neglected.

Another mechanism of sound production, aside from the aeroacoustic one, is the vibrations of VFs, see e.g. [7]. This contribution is often modelled as a simplified vibro-acoustic problem, neglecting the influence of the acoustic field on VF vibration, see Fig. 1 right. This problem is sometimes overlooked due to anticipated prominence of aerodynamically produced sound, [13], for a more detailed discussion see e.g. [23].

The typical healthy VF vibration regime is characterized by flutter instability, making modelling and numerical approaches highly demanding, [17], [15]. During the flutter regime the structural displacements exponentially grow until – for the case of healthy phonation – the both VFs reach contact and impact each other. Mathematical modelling of contact problems is challenging on its own and highly demanding to be included in the already complex FSAI problem [13]. Although some promising results emerged, e.g. a low-order model comprising a three-mass system coupled with 1D Euler equations and Hertz contact theory applied, [5], or more recently a simplified contact treatment in the continuum settings [20], the contact modelling is completely omitted here. On the other hand, the novelty of the

present study lies in the detailed analysis of the energy balance between the flow and the structure based on the pressure-gap curve [4], as well as in the improved acoustic results compared to those previously published in [18], where the final results were affected by the improper implementation of a perfectly matched layer (PML).

The FSI problem is modelled here by a linear elasticity model for the vocal folds and the incompressible Navier-Stokes equations for the air flow. The arbitrary Lagrangian-Eulerian (ALE) method addresses the time-dependent fluid domain, see [2], offering simplicity in description and implementation, [16], but requiring remeshing or additional modifications for topological changes, such as the omitted contact phenomenon, [8, 20]. The numerical discretization by the finite element method (FEM) is performed and the stabilization of the convection-dominated airflow is applied. Finally, two aeroacoustic approaches are presented: the classical Lighthill (LH) analogy and the perturbed convective wave equation (PCWE) based on a careful separation of acoustic from other fluid components, [1, 7]. The analysis of computed sound sources is important for validating the computation procedure and identifying the origin and location of the generated sound, [13]. Subsequently, the time propagation problem is solved in the selected acoustic domain representing vocal tract geometry, which can strongly influence acoustic results, cf. [23] and [13]. The PML technique models acoustically open boundaries by effectively absorbing outgoing acoustic waves, surpassing other methods limited to specific angles, [7].

The structure of the paper is as follows. The next section is devoted to the FSI problem formulation including also description of numerical approximation and details of the FSI simulation. The third section presents (two) aeroacoustic models and the analysis of sound sources based on the FSI simulation. Finally, a short conclusion closes the paper.

## 2. FSI model

First, the geometrical configuration is showed. Further, the mathematical description of the FSI problem and the FEM discretization procedure is given. Some characteristic results of the flow-induced vibrations of VFs are shown.

### 2.1. Geometry

The schematic figure of larynx anatomy including VF position without an airways space is shown at Figure 2 followed by a considered idealized two-dimensional geometrical set-up of the FSI problem. For the description of the elastic structure deformation the reference coordinates are utilized, i.e. computational domain $\Omega^s = \Omega^s_t = \Omega^s_{\mathrm{ref}} \subset \mathbb{R}^2$ at arbitrary time $t$ is used. In the case of fluid flow we distinguish between the reference fluid domain $\Omega^f_{\mathrm{ref}} \subset \mathbb{R}^2$, i.e. the domain occupied by fluid at time instant $t = 0$ with the common interface $\Gamma_{\mathrm{W_{ref}}} = \Gamma_{\mathrm{W_0}}$, and the domain $\Omega^f_t \subset \mathbb{R}^2$ occupied by fluid at any time instant $t \in (0, \mathrm{T})$, which is determined by the motion of the elastic structure (particularly by the position of the interface $\Gamma_{\mathrm{W_t}}$).

### 2.2. Mathematical model

We start with the description of the ALE method which allow us to treat relatively easy time-dependency of fluid domain $\Omega_{\mathrm{t}}^f$.

**ALE method.** This method is based on a diffeomorphic and smooth mapping $A_t$ of any reference point $X \in \Omega_{ref}^f$ on the point of deformed domain $x = A_t(X) \in \Omega_t^f$, particularly the interface can only evolve in time (according to the structural displacement) as $\Gamma_{\mathrm{W_t}} = A_t(\Gamma_{\mathrm{W_{ref}}})$, while the other boundaries remain static $A_t(\partial\Omega_{\mathrm{ref}}^f \backslash \Gamma_{\mathrm{W_{ref}}}) = \partial\Omega_{\mathrm{ref}}^f \backslash \Gamma_{\mathrm{W_{ref}}}$. Further, the ALE domain velocity $\mathbf{w}_D$ representing the velocity of a point $x$ with a given reference $X \in \Omega_{\mathrm{ref}}^f$ is defined by

$$\mathbf{w}_D(x,t) = \hat{\mathbf{w}}_D(A_t^{-1}(x),t), \quad \text{where } x = A_t(X) \in \Omega_t^f, \tag{1}$$

and $\hat{\mathbf{w}}_D(X,t) = \frac{\partial}{\partial t}A_t(X)$, for $t \in (0,\mathrm{T})$ and $X \in \Omega_{\mathrm{ref}}^f$. Finally, the ALE derivative, i.e. the time derivative with respect to a fixed reference $X \in \Omega_{\mathrm{ref}}^f$, satisfies (see [2])

$$\frac{D^A}{Dt}f(x,t) = \frac{\partial f}{\partial t}(x,t) + \mathbf{w}_D(x,t)\cdot\nabla f(x,t). \tag{2}$$

**Fluid flow.** The flow of a viscous incompressible fluid in the time-dependent domain $\Omega_t^f$ is modelled using the Navier-Stokes equations in the ALE form (for details see [2])

$$\frac{D^A\mathbf{v}}{Dt} + ((\mathbf{v}-\mathbf{w}_D)\cdot\nabla)\mathbf{v} - \nu^f\Delta\mathbf{v} + \nabla p = \mathbf{0}, \quad \text{div } \mathbf{v} = 0 \quad \text{in } \Omega_t^f, \tag{3}$$

where $\mathbf{v}(x,t)$ is the fluid velocity, $p$ denotes the kinematic pressure and $\nu^f$ is the kinematic fluid viscosity.

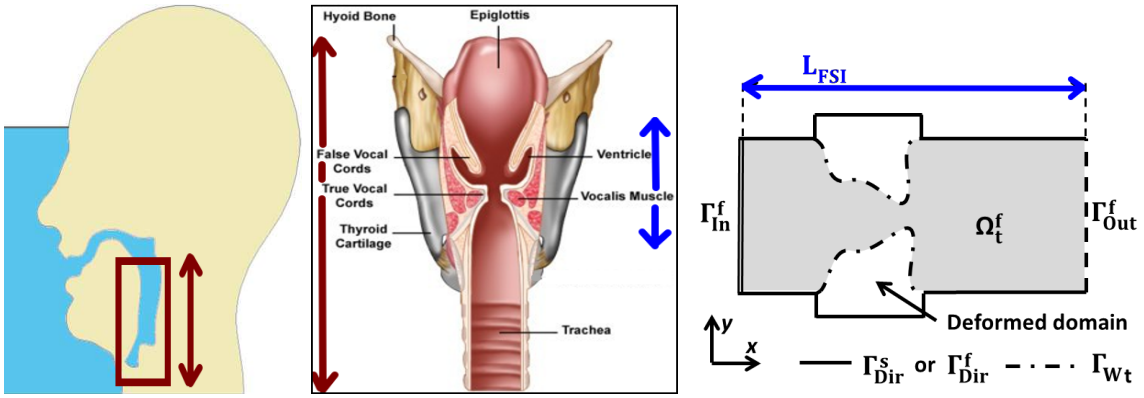



Figure 2: **Left:** Schematic picture of upper human airways. **Middle:** Frontal cut of the larynx reveals the position and a complicated physiological structure of VFs. Arrows denotes approximate scaling with respect to the left and to the right figure. **Right:** Considered simplified FSI geometry undergoing a VFs deformation and the marked boundaries are: inlet $\Gamma_{\mathrm{In}}^f$, outlet $\Gamma_{\mathrm{Out}}^f$, walls $\Gamma_{\mathrm{Dir}}^f$, $\Gamma_{\mathrm{Dir}}^s$ and interface $\Gamma_{\mathrm{W_t}}$.

We impose the zero initial condition and the following boundary conditions (BCs) alongside equations (3)

$$\begin{aligned}
&\text{a)} && \mathbf{v}(x,t) = \mathbf{w}_D(x,t) && \text{for } x \in \Gamma^f_{\mathrm{Dir}} \cup \Gamma_{\mathrm{W_t}},\\
&\text{b)} && (p - p_{\mathrm{ref}})\mathbf{n}^f - \nu^f \frac{\partial \mathbf{v}}{\partial \mathbf{n}^f} = -\frac{1}{2}\mathbf{v}(\mathbf{v}\cdot\mathbf{n}^f)^- && \text{on } \Gamma^f_{\mathrm{Out}},\\
&\text{c)} && (p - p_{\mathrm{ref}})\mathbf{n}^f - \nu^f \frac{\partial \mathbf{v}}{\partial \mathbf{n}^f} = -\frac{1}{2}\mathbf{v}(\mathbf{v}\cdot\mathbf{n}^f)^- + \frac{1}{\epsilon}(\mathbf{v}-\mathbf{v}_{\mathrm{in}}) && \text{on } \Gamma^f_{\mathrm{In}},
\end{aligned} \tag{4}$$

where the vector $\mathbf{n}^f = (n^f_j)$ denotes the outward unit normal to the boundary $\partial\Omega^f$, $p_{\mathrm{ref}}$ denotes a reference pressure and by $(\alpha)^-$ the negative part of real number $\alpha \in \mathbb{R}$ is denoted, i.e. $(\alpha)^- = \min\{0, \alpha\}$. Condition (4 b) is the so-called directional do-nothing boundary condition, which increases the stability in the case of a backward inlet through the outlet boundary, see [11]. Condition (4 c) is the penalization inlet boundary condition, a generalization of the Dirichlet (for $\epsilon \to 0$) and the Neumann BC (for $\epsilon \to +\infty$), see [21]. For suitably chosen penalization parameter $\epsilon$ its behaviour is favourable, as it allows maintaining the maximal subglottic pressure within a physiological range during the channel closing phase, [16, 21].

**Elastic structure.** The structure deformation represented by displacement $\mathbf{u}(X,t) = (u_1, u_2)$ of any point $X \in \Omega^s$ is described by partial differential equations

$$\rho^s \frac{\partial^2 u_i}{\partial t^2} - \frac{\partial \tau^s_{ij}}{\partial X_j} = 0, \qquad \text{in } \Omega^s \times (0,\mathrm{T}), \quad (i = 1,2), \tag{5}$$

where $\rho^s$ is the structure density and $\tau_{ij}$ are the components of the Cauchy stress tensor. The stress tensor components assuming the isotropic body can be expressed as

$$\tau^s_{ij} = \lambda^s \mathrm{div}\ \mathbf{u}\, \delta_{ij} + 2\mu^s e^s_{ij}(\mathbf{u}), \tag{6}$$

where $\delta_{ij}$ denotes Kronecker's delta and $e^s_{ij}(\mathbf{u}) = \frac{1}{2}\left(\frac{\partial u_j}{\partial X_i} + \frac{\partial u_i}{\partial X_j}\right)$ is the small strain tensor. Parameters $\lambda^s, \mu^s$ are the Lamé coefficients, see e.g. [2]. Problem (5) is equipped with the zero initial conditions and the following BCs

$$\begin{aligned}
&\text{a)} && \mathbf{u}(X,t) = \mathbf{u}_{\mathrm{Dir}}(X,t) && \text{for } X \in \Gamma^s_{\mathrm{Dir}},\\
&\text{b)} && \tau^s_{ij}(X,t)\, n^s_j(X) = q^s_i(X,t), && \text{for } X \in \Gamma_{\mathrm{W_{ref}}},
\end{aligned} \tag{7}$$

where the $\Gamma_{\mathrm{W_{ref}}}, \Gamma^s_{\mathrm{Dir}}$ are disjoint parts of the boundary $\partial\Omega^s$ and $n^s_j(X)$ are the components of the outward unit normal to $\partial\Omega^s$, see Figure 2.

**Coupling conditions.** The fluid and structure problems are coupled together with the aid of the interface boundary conditions prescribed at the interface $\Gamma_{\mathrm{W_t}}$ whose position is unknown and it is determined implicitly through the structural displacement $\mathbf{u}$

$$\Gamma_{\mathrm{W_t}} = \left\{ x \in \mathbb{R}^2 \,|\, x = X + \mathbf{u}(X,t),\ X \in \Gamma_{\mathrm{W_{ref}}} \right\}, \forall t \in (0,\mathrm{T}). \tag{8}$$

Further, the kinematic BC representing continuity of velocities across the interface is prescribed for the fluid flow problem in the form of equation (4 a).
The dynamic BC enforcing stress continuity in normal direction at the interface $\Gamma_{\mathrm{W_{ref}}}$ has the form of equation (7 b), where the components $q_i^s$ of the vector of acting aerodynamic forces $\mathbf{q}^s$ are given by

$$q_i^s = \sum_{j=1}^{2} \rho^f \left( p\delta_{ij} - \nu^f \left( \frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \right) n_j^f(x). \tag{9}$$

### 2.3. Numerical approximation

The FEM is used for spatial discretization of considered subproblems (5) and (3). For the purpose of time discretization the time interval $[0, \mathrm{T}]$ is divided into $N$ equidistant parts of length $\Delta t$, i.e. $t_n = n\Delta t, \Delta t = \frac{T}{N}$, where $n = \{0, 1, \ldots, N\}$.

**Elastic structure.** The FEM discretization of elasticity problem (5) is standard and it leads to the system of ordinary differential equations of the second order

$$\mathbb{M}\ddot{\boldsymbol{\alpha}} + \mathbb{C}\dot{\boldsymbol{\alpha}} + \mathbb{K}\boldsymbol{\alpha} = \mathbf{b}(t), \tag{10}$$

for definitions and further details see [21]. The system (10) is then numerically solved by the Newmark method.

**Fluid flow.** First, the ALE derivative is discretized by the backward difference formula of second order (BDF2), see [2].

In order to formulate problem (3) weakly, we start with the definition of function spaces involved. The function space for velocity test functions $\mathbf{X} = X \times X$ is defined as follows $X = \{f \in H^1(\Omega^f) | \, f = 0 \text{ on } \Gamma^f_{\mathrm{Dir}} \cup \Gamma^f_{\mathrm{W}_{t_{n+1}}}\}$ and $M = L^2(\Omega^f)$. Then the fluid flow problem can be formulated abstractly in weak form as searching for unknown $V = (\mathbf{v}, p) \in \mathbf{H}^1(\Omega^f) \times M$, which approximately satisfies boundary condition (4a) and

$$a(V, \Phi) + c(V; V, \Phi) + \frac{1}{2}((\mathbf{v}\cdot\mathbf{n})^+\mathbf{v}, \boldsymbol{\varphi})_{\Gamma^f_{\mathrm{In}}} + \frac{1}{\epsilon}(\mathbf{v}, \boldsymbol{\varphi})_{\Gamma^f_{\mathrm{In}}} = f(\Phi) + \frac{1}{\epsilon}(\mathbf{v}_{\mathrm{Dir}}, \boldsymbol{\varphi})_{\Gamma^f_{\mathrm{In}}} \tag{11}$$

is fulfilled for any test function $\Phi = (\boldsymbol{\varphi}, q) \in \mathbf{X} \times M$, where

$$\begin{aligned}
a(V, \Phi) &= \left( \frac{3\mathbf{v}}{2\Delta t}, \boldsymbol{\varphi} \right)_{\Omega^f} + \nu^f (\nabla\mathbf{v}, \nabla\boldsymbol{\varphi})_{\Omega^f} - (p, \operatorname{div}\boldsymbol{\varphi})_{\Omega^f} + (q, \operatorname{div}\mathbf{v})_{\Omega^f}, \\
c(V^*; V, \Phi) &= \frac{1}{2}\left( (((\mathbf{v}^* - 2\mathbf{w}_D)\cdot\nabla)\mathbf{v}, \boldsymbol{\varphi})_{\Omega^f} - ((\mathbf{v}^*\cdot\nabla)\boldsymbol{\varphi}, \mathbf{v})_{\Omega^f} + ((\mathbf{v}^*\cdot\mathbf{n})^+\mathbf{v}, \boldsymbol{\varphi})_{\Gamma^f_{\mathrm{Out}}} \right), \\
f(\Phi) &= \frac{1}{2\Delta t}\left( 4\overline{\mathbf{v}}^n - \overline{\mathbf{v}}^{n-1}, \boldsymbol{\varphi} \right)_{\Omega^f},
\end{aligned} \tag{12}$$

and by $(\alpha)^+$ the positive part of real number $\alpha \in \mathbb{R}$ is denoted, i.e. $(\alpha)^+ = \max\{0, \alpha\}$. The bilinear form $a(\cdot,\cdot)$ and functional $f(\cdot)$ is the standard weak formulation of

Stokes problem. The trilinear form $c(\cdot;\cdot,\cdot)$ represents the skew-symmetric form of the convection, which gives us the directional do-nothing BC (4b), see [11]. The realization of penalization inlet BC (4 c) introduces additional terms $\frac{1}{2}((\mathbf{v}\cdot\mathbf{n})^{+}\mathbf{v},\boldsymbol{\varphi})_{\Gamma^f_{\mathrm{In}}} + \frac{1}{\epsilon}(\mathbf{v},\boldsymbol{\varphi})_{\Gamma^f_{\mathrm{In}}}$ and $\frac{1}{\epsilon}(\mathbf{v}_{\mathrm{Dir}},\boldsymbol{\varphi})_{\Gamma^f_{\mathrm{In}}}$ in the weakly formulated fluid flow problem, see [21].

The derived weak formulation (12) is discretized by the stabilized FEM, see [21].

Finally, the strongly coupled partitioned approach is selected for the FSI numerical solution, i.e. the convergence of aerodynamic forces in each inner iteration cycle is checked and the fluid flow and the elasticity approximative solutions are iterated in every time step until the difference of aerodynamic forces is smaller than $10^{-5}$, see [21].

### 2.4. Numerical results of the FSI problems

In this part the FSI problem is solved in the full channel with vocal fold model MALE having parabolic shape, see e.g. [5, 16]. All material parameters are the same as in [21, 19], particularly the initial gap is set to $g_{\mathrm{init}} = 0.8\,\mathrm{mm}$ and time step $\Delta t = 2.5\cdot 10^{-5}\,\mathrm{s}$. Then four cases with different inlet BCs are compared:

1) case DIR: the Dirichlet boundary condition $\mathbf{v} = \mathbf{v}_{\mathrm{Dir}}$ with the given constant inlet velocity $\mathbf{v}_{\mathrm{Dir}} = (2.1, 0)\,\mathrm{m/s}$.

2) case PRES: the pressure difference (between the inlet and the outlet) in the form of $p_{\mathrm{ref}} = 400\,\mathrm{Pa}$ is prescribed in condition (4b) on the inlet $\Gamma^f_{\mathrm{In}}$. The choice of pressure drop ensures that the airflow rates in cases PRES and DIR are comparable.

3) case PEN-W: the penalization BC (4c) is applied with the given velocity $\mathbf{v}_{\mathrm{Dir}}$ and the penalization parameter $\epsilon = 5\cdot 10^{-4}\,\mathrm{s/m}$.

4) case PEN-S: the penalization BC (4c) is applied with the given velocity $\mathbf{v}_{\mathrm{Dir}}$ and the penalization parameter $\epsilon = 1\cdot 10^{-5}\,\mathrm{s/m}$.

First, two snapshots from the PEN-S simulation are shown in Figures 3 and 4, illustrating the typical change in VFs position as it alternates between convergent and divergent states. Further, the increasing intensity of glottal jet during opening phase followed by intensity fading for fully open glottis and again the rise of fluid velocity at the glottis up to the maximal values during VF closing phase can be observed. The large vortices formed downstream from the glottis (only the first one is visible in the snapshots) are slowly decaying into smaller ones. The very similar character of the flow field was obtained e.g. in [8].

The given selection of $\mathbf{v}_{\mathrm{Dir}}$ and $p_{\mathrm{ref}}$ is above critical one and it leads in all cases to flutter instability phenomenon, simulations were terminated by a solver failure due too large structure vibration amplitudes and therefore too much deformed computational fluid mesh. Such behaviour is here documented by the inlet flow velocity, the pressure drop and the (whole) gap width displayed in Figures 5 and 6. We can notice that the inlet velocity is either constant or heavily oscillating in cases of DIR and PRES, respectively. Similarly the pressure drop - if prescribed - remains almost constant, while for the DIR case it grows fast to unphysically high values. This

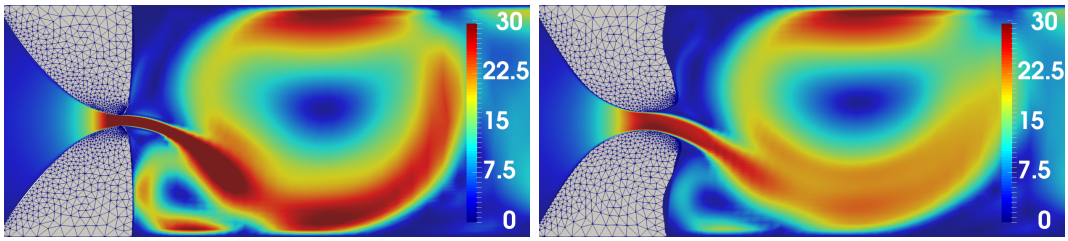



Figure 3: Airflow velocity magnitude in *PEN-S* case at moments of the most closed and the most opened channel. The domain $\Omega_t^f$ is in figures truncated.

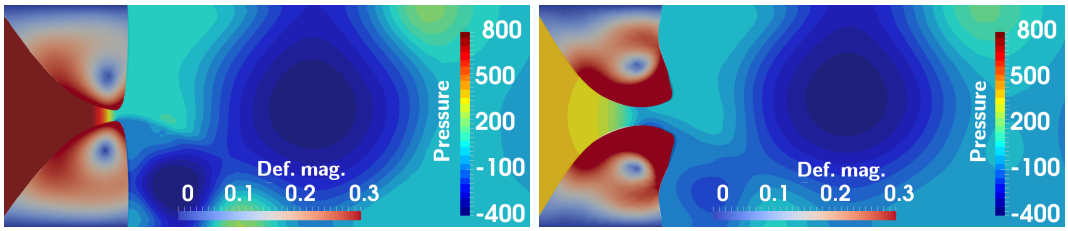



Figure 4: Airflow pressure and magnitude of the VF displacement in mm shown in *PEN-S* case as in Fig. 3.

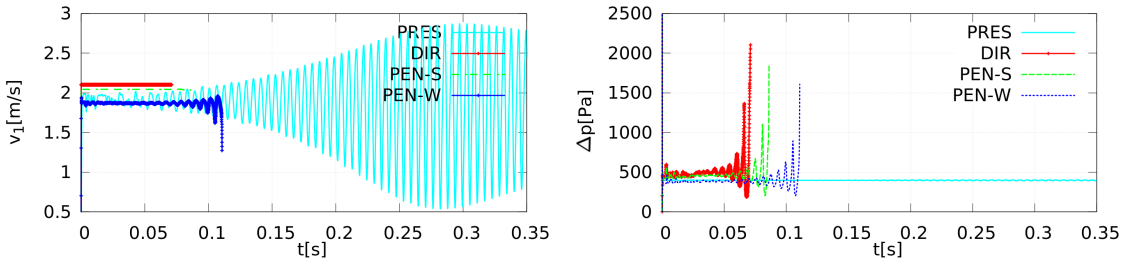



Figure 5: The inlet airflow velocity (left) and pressure difference between the inlet and the outlet of the channel (right) for cases *DIR*, *PEN-S*, *PEN-W* and *PRES*.

behaviour is expected as theoretically the pressure drop in the DIR case would reach infinity as the channel approaching closure.

The behaviour of the PEN-S and PEN-W cases, i.e. a generalization of both previous BCs with switching controlled by parameter $\epsilon$, provides a combination of the aforementioned. The inlet velocity can a little oscillate and the pressure drop gradually rises as the gap between VFs starts to close more and more, see Figure 6. Nevertheless, the maximal value of the pressure drop is obviously controlled by the value of $\epsilon$.

Further, the VF vibration pattern can be illustrated on the phase portraits of point S (the top point of the bottom VF), see Figure 7. The phase portraits of cases DIR and PEN-S indicate a much faster development of the flutter phenomenon than in case PRES. The phase portrait of case PRES moreover differs in the motion of point S, the different motion pattern(s) is evidently excited.

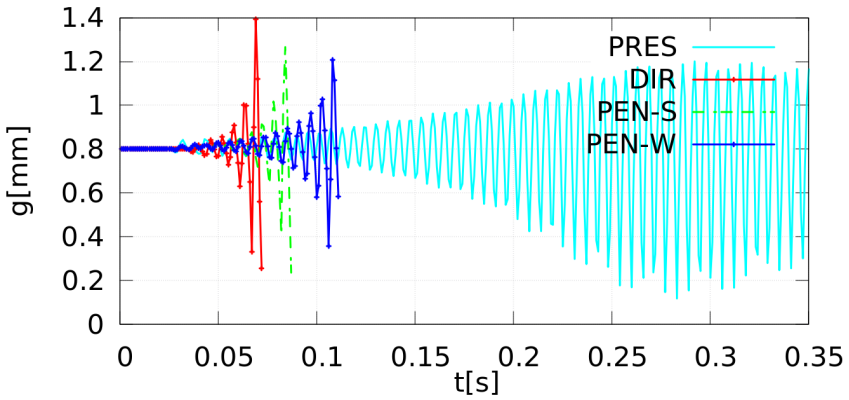



Figure 6: Time development of the gap in cases *DIR*, *PEN-S*, *PEN-W* and *PRES*.

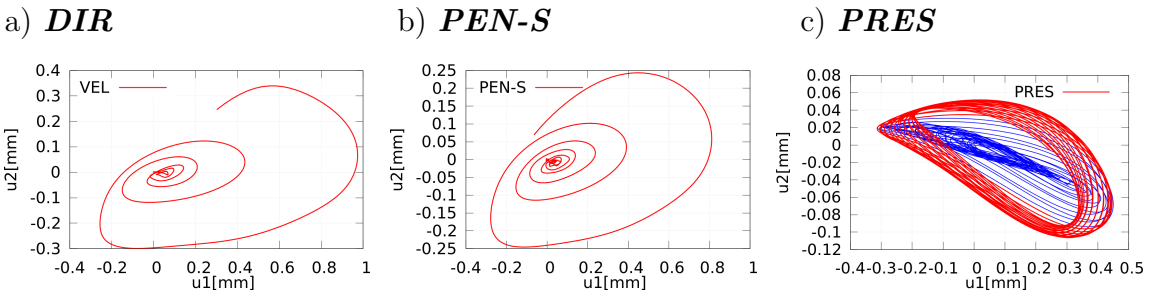



Figure 7: Trajectory of point S in the X–Y plane. The blue curve in the *PRES* case shows the initial development, while the red one marks the developed VF vibrations.

Additionally, the dependence of the transglottal pressure on the gap can be constructed from Figures 5 and 6 by time elimination, see Figure 8. The pressure-gap curve is a rough estimate of the transferred energy from airflow to VF vibration provided by means of an area $A$ closed inside, [4], and it is usually a good metric in the case of laboratory experiments, although the transferred energy can be precisely computed for the case of numerical simulation, see e.g. [19].

The pressure-gap curves in Fig. 8 capture the flutter regime and they are not closed as the regular periodic VF vibration cycle has not emerged yet (typically connected with VF mutual contact). Nevertheless, it is obvious that the pressure drop associated with reaching a certain minimal gap value is much lower for case PEN-S (and also for PEN-W) compared to the DIR case, and it still remains within the physiological range, i.e. below circa 3 kPa, [17]. The orientation of the curves in all cases is anticlockwise, which is interestingly in a contradiction with laboratory results of [4].

## 3. Aeroacoustic models

First, the considered two-dimensional geometry is shown. Then two diffferent aeroacoustic analogies are described. Finally, the acoustic sources and corresponding results of simplified FSAI simulation are shown.

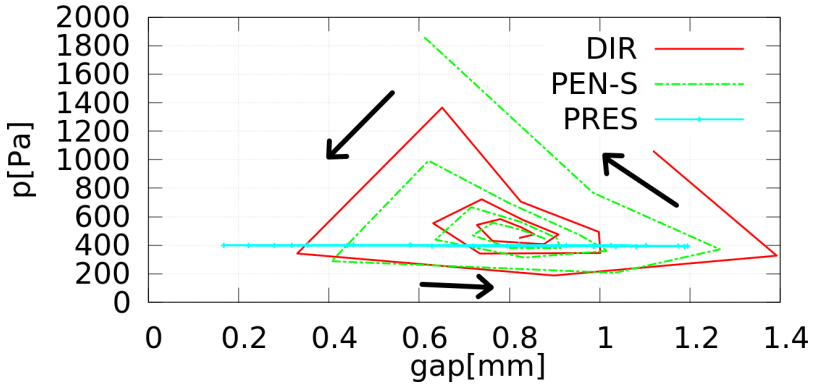



Figure 8: Dependence of the transglottal pressure on the gap for three simulations of the FSI problem: *DIR*, *PEN-S* and *PRES*. The graph depicts only last four incomplete oscillation cycles and it is undulated due to too low sampling rate of the data saving. Arrows show the orientation of the curves (i.e. time progression).

### 3.1. Geometry configuration

The acoustic domain $\Omega^a$, where the acoustic problem is solved, is depicted in Figure 9, compare it with Figure 2. It is composed of three parts, i.e. $\overline{\Omega^a} = \overline{\Omega^a_{\text{src}}} \cup \overline{\Omega^a_{\text{air}}} \cup \overline{\Omega^a_{\text{pml}}}$. The acoustic sources are calculated from the known flow field exclusively in the domain $\Omega^a_{\text{src}}$, which is the same as the reference fluid domain, i.e. $\Omega^a_{\text{src}} = \Omega^f_{\text{ref}}$.[1] The domain $\Omega^a_{\text{air}}$ represents a part of the vocal tract behind the glottis up to the mouth (indicated by arrow $L_{\text{tract}}$ in Fig. 9) including a far field region (arrow $L_{\text{free}}$), i.e. the outer space. The PML domain $\Omega^a_{\text{pml}}$ (see arrow $L_{\text{PML}}$) closes both the aforementioned domains in order to damp the outgoing sound waves.

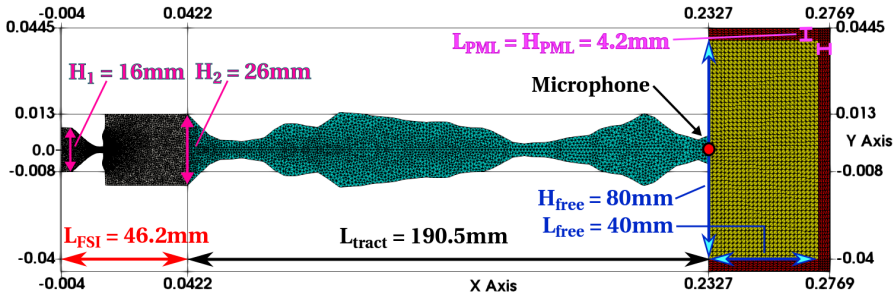



Figure 9: Computational acoustic domain $\Omega^a$ with vocal tract model M1 described later and its dimensions. Microphone is placed in the mouth opening.

### 3.2. Mathematical models of the aeroacoustic problem

Aeroacoustics studies sound generated by aerodynamic processes, typically sound generated by flow around obstacles or by turbulence, see e.g. [7], [1]. The compress-

[1] The change of domain $\Omega^a_{\text{src}}$ in time is neglected. Sound sources outside domain $\Omega^a_{\text{src}}$ are omitted.

ible Navier-Stokes equations in general describe all aspects of fluid flow including acoustics. However, acoustic pressure is usually a tiny part of the total pressure, often comparable to numerical errors. Additional challenges arise from length scale disparities or unwanted dispersion and dissipation properties of numerical schemes, see [7]. To address these challenges, hybrid acoustic analogies, which decouple fluid flow and acoustic problems, provide an effective and practical solution by allowing the use of problem-specific solvers.

### 3.2.1. Lighthill acoustic analogy

The Lighthill analogy was derived from compressible Navier-Stokes equations under the assumption that acoustic waves with origin in a small source region propagate through a surrounding medium in rest state characterized by $\mathbf{v}_0 = \mathbf{0}, p_0$ and rest fluid density $\rho_0^f$. The Lighthill analogy has the final form of inhomogenous wave equation for unknown pressure fluctuation $p' = p - p_0$

$$\frac{1}{c_0^2}\frac{\partial^2 p'}{\partial t^2} - \frac{\partial^2 p'}{\partial x_i^2} = \frac{\partial^2 T_{ij}}{\partial x_i \partial x_j}, \tag{13}$$

with a given speed of sound $c_0$ and known values of the Lighthill tensor $\mathbf{T} = (T_{ij})$, which double divergence plays role of effective sound source term. The components of the Lighthill tensor $T_{ij}$ are given by

$$T_{ij} = \rho^f v_i v_j + ((p - p_0) - c_0^2(\rho^f - \rho_0^f))\delta_{ij} - \tau_{ij}^f \approx \rho_0^f v_i v_j, \tag{14}$$

where $\tau_{ij}^f$ is the fluid viscous stress tensor and the subsequent approximation of the Lighthill tensor by neglecting the viscous stress $\tau_{ij}^f$ and the stresses connected with the non-isentropic processes $(p' - c^2\rho')\delta_{ij}$ are applied according to [9], [1].

The disadvantage of the Lighthill analogy is that pressure fluctuation $p'$ can be regarded as the acoustic pressure $p^a$ only outside the flow domain because inside the source region it represents a superposition of acoustic and hydrodynamic pressures, see [7], [1].

### 3.2.2. Perturbed convective wave equation

Another suitable choice from many other acoustic analogies is the PCWE, see [6, 7]. Its aim is to describe more precisely the behaviour of purely acoustic components. It is based on splitting of physical quantities into mean and fluctuating parts. The fluctuating variables consists of acoustic parts $\mathbf{v}^a, p^a$ and non-acoustic components $\mathbf{v}^{ic}$, $p^{ic}$, (i.e. incompressible parts)

$$p = \overline{p} + p^{ic} + p^a, \qquad \mathbf{v} = \overline{\mathbf{v}} + \mathbf{v}^{ic} + \mathbf{v}^a, \tag{15}$$

see [7]. Assuming incompressible homoentropic flow the splitting leads to the following partial differential equation for unknown $\mathbf{v}^a$ and $p^a$

$$\frac{\partial p^a}{\partial t} + \overline{\mathbf{v}} \cdot \nabla p^a + \rho_0^f c_0^2 \nabla \cdot \mathbf{v}^a = -\frac{Dp^{ic}}{Dt}, \qquad \frac{\partial \mathbf{v}^a}{\partial t} + \nabla(\overline{\mathbf{v}} \cdot \mathbf{v}^a) + \frac{1}{\rho_0^f}\nabla p^a = \mathbf{0}, \tag{16}$$

where the substantial derivative $\frac{D}{Dt}$ equals $\frac{D}{Dt} = \frac{\partial}{\partial t} + \overline{\mathbf{v}} \cdot \nabla$. These equations can be rewritten into scalar one, denoted as PCWE, with the help of acoustic potential $\psi^a$, which is related to the acoustic particle velocity as $\mathbf{v}^a = -\nabla\psi^a$ (since the acoustic velocity field is irrotational)

$$\frac{1}{c_0^2}\frac{D^2\psi^a}{Dt^2} - \Delta\psi^a = -\frac{1}{\rho_0^f c_0^2}\frac{Dp^{ic}}{Dt}. \tag{17}$$

Moreover, for low velocities, we can simplify (17) by disregarding the convection effect and setting $\overline{\mathbf{v}} = 0$, see [18]. A relatively big advantage comparing (17) with (13) is only one and just the time derivative of right hand source term. The numerical computation of the time derivative is usually less sensitive to numerical errors, [7, 1], and also it is usually well resolved in time.

The equations (13) or (17) are equipped with the zero initial conditions and the boundary of acoustic domain $\partial\Omega^a$ with the outer normal $\mathbf{n}^a$ is considered as fully reflecting (called also sound hard)

$$\frac{\partial P}{\partial \mathbf{n}^a}(x,t) = 0 \qquad \text{for } x \in \partial\Omega^a,\ t \in (0,\mathrm{T}), \tag{18}$$

where $P$ denotes the appropriate acoustic unknown.

**PML.** In order to mimic the open-boundary problem of radiation acoustic waves outside the human head the PML technique is used. The key of this technique is to add a new PML subdomain on the boundary. The proper choice of complex values of sound speed and density governed by the set of artificial equations inside the PML domain leads to exponential wave damping inside PML and to eliminating any reflection of acoustic waves on the interface between the propagation domain and the PML. We further refer to [7].

### 3.3. Numerical approximation

For the numerical solution the FEM is again used, see e.g. [18]. The interpolation of aeroacoustic sources from the computational fluid to the acoustic mesh is performed with the help of the program CFSDat, see [7].

### 3.4. Numerical results of the simplified FSAI problem

This part contains acoustic results corresponding to proper choice of acoustic domains characterized by their resonant frequencies, computation and analysis of sound sources and finally the transient computation providing the frequency spectra of phonation of vowel [u:].

#### 3.4.1. Resonant frequencies of acoustic domains

Two variants of acoustic domain $\Omega^a$ are analyzed here in order to find their acoustic resonant frequencies, usually called formants. In both cases the acoustic domains

differ only in the portion of inclusion of domain $\Omega^a_{\mathrm{src}}$. The first variant is labeled as M1 (model 1) and the second as M2 (model 2), which has removed the subglottal and the glottal regions, see Figure 10. The part of domain $\Omega^a$ representing the VT model for the vowel [u:] based on vocal tract cross-section MRI segmentation [14] is for M1 and M2 models the same, see Figure 9 and [22].

The formants of vocal tract are determined by the transfer function approach due to inclusion of the PML layer prohibiting a natural choice of modal analysis. In this approach, the ratio of the output to the input (unit) signal $\hat{F}$ is evaluated based on the Helmholtz equation (wave equation in frequency domain), see [7, 22],

$$-\left(\frac{\omega^2}{c_0^2}+\Delta\right)\hat{p}=\hat{F}, \tag{19}$$

where the speed of sound $c_0 = 343\,\mathrm{m/s}$, $\omega$ denotes the angular frequency and $\hat{p}(x,\omega)$ is the Fourier transform of $p(x,t)$. As output is regarded $\hat{p}$ at the microphone position in the investigated frequency range $50-3000\,\mathrm{Hz}$.

The transfer functions computed for models M1 and M2 are shown in Figure 10 on the right and the found formants are listed in Table 1. Both models M1 and M2 have four formants in the range $50-2500$ Hz, M1 having an additional formant F5 at 2638 Hz due to the subglottal part of the VT model, see [22]. The occurrence of F3 at frequency 1432 Hz contrary to Story's results [14] is probably caused by the longer acoustic domain (the length of approx. 23 cm compared to Story's length of 18.25 cm). The formant frequency F4 of both models lies in the vicinity of Story's reference F3, however the M2 model is chosen for further simulations due to a higher similarity with results of [14].

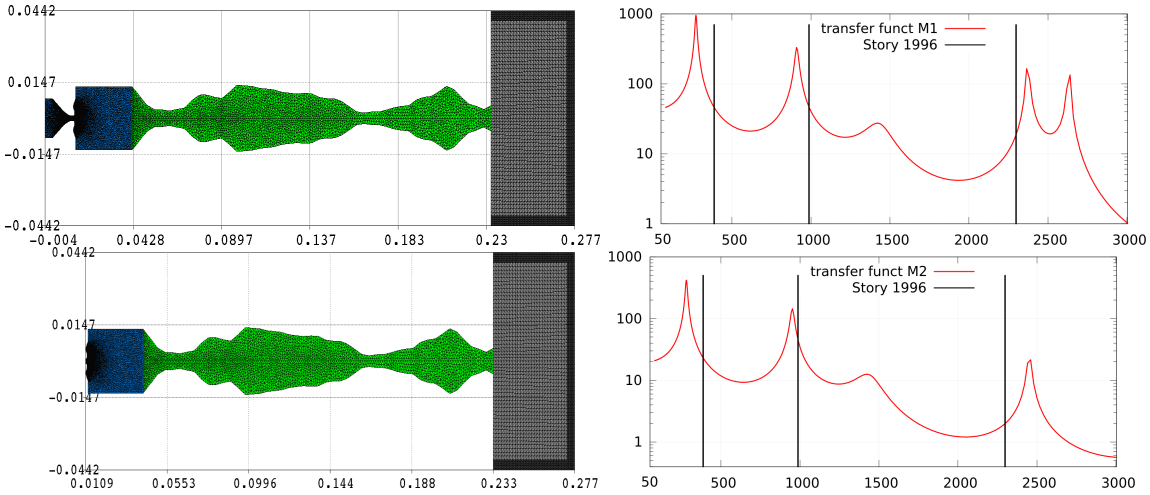



Figure 10: **Left:** Acoustic models M1 and M2. **Right:** Computed transfer functions for given cases. The formants of vowel [u:] from [14] are highlighted by vertical lines.

| | $F_1$ | $F_2$ | $F_3$ | $F_4$ |
|---|---|---|---|---|
| M1 | 271 | 909 | 1432 | 2365 |
| M2 | 280 | 952 | 1432 | 2440 |
| Story | 389 | 987 | 2299 | – |

Table 1: Computed formant frequencies (in Hz) of the vocal tract models M1 and M2. The measured (Story) results for vowel [u:] are from reference [14].

### 3.4.2. Sound sources

The aeroacoustic results are based on the FSI results obtained with four-layered VF of shape denoted by us as ZORNER and inlet pressure difference of 800 Pa, see the detailed settings and the results of fluid flow in [18]. The sound sources computed from the FSI results are analyzed to get a spatial distribution and frequency content. Finally, the sound source propagation in the chosen acoustic model M2 of both aeroacoustic approaches – LH and PCWE, are compared.

**Spatial distribution of sound sources for different aeroacoustic approaches.** The sound sources computed for both different approaches according to (14) and (17) are displayed in Figure 11. In the LH case the sound sources are primarily associated with the velocity gradients and in the current simulation they are greatly distributed downstream of the glottis, where the glottal jet creates strong shear layers as it enters the supraglottal spaces, and also in the vicinity of the VF boundary, where the glottal jet separates from the VF surface.

The dominant sound sources in the cases of the PCWE approach are connected with pressure time changes, which local extremes are located primarily in the vortex centers. The vortices are formed by a complex decay of the glottal jet downstream the glottis. The sound source structure is similar as in phase-locked PIV measurements [10] or in the numerical simulations [12].

**Frequency content.** The frequency content of the sound sources is investigated with the Fourier transform applied on the time signal at each point of the sound sources. The power spectral densities (PSD) of the sound sources at two representative frequencies for both aeroacoustic approaches are shown in Figure 12. The frequencies 232 Hz and 2486 Hz are the local spectral maxima representing one of the dominant VF vibration frequencies and an (higher) non-harmonic frequency, respectively. The quantitative comparison of sound sources PSD values is here irrelevant as in all cases a different acoustic quantity is depicted.

The location of main sound sources for frequency 232 Hz for all considered cases is inside the glottis and having dipole character. The LH sources located before the tip of VFs are less prominent than the quadrupole-like structure formed downstream from the narrowest part of the channel. In the PCWE case the dipole clearly dominates. These findings coincide very well with the results of [13].

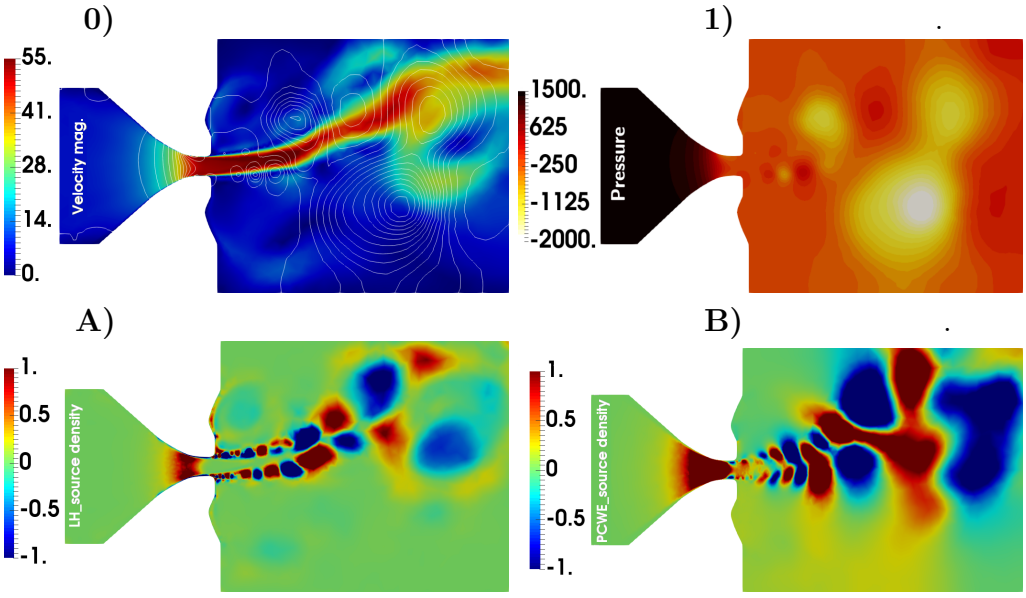



Figure 11: Comparison of (normalized) instant sound densities for different aeroacoustic approaches at chosen time instant shown together with the flow field. **0)** The magnitude of airflow velocity. **1)** The pressure distribution. Below instant sound densities are shown for: **A)** the LH analogy and **B)** the PCWE approach.

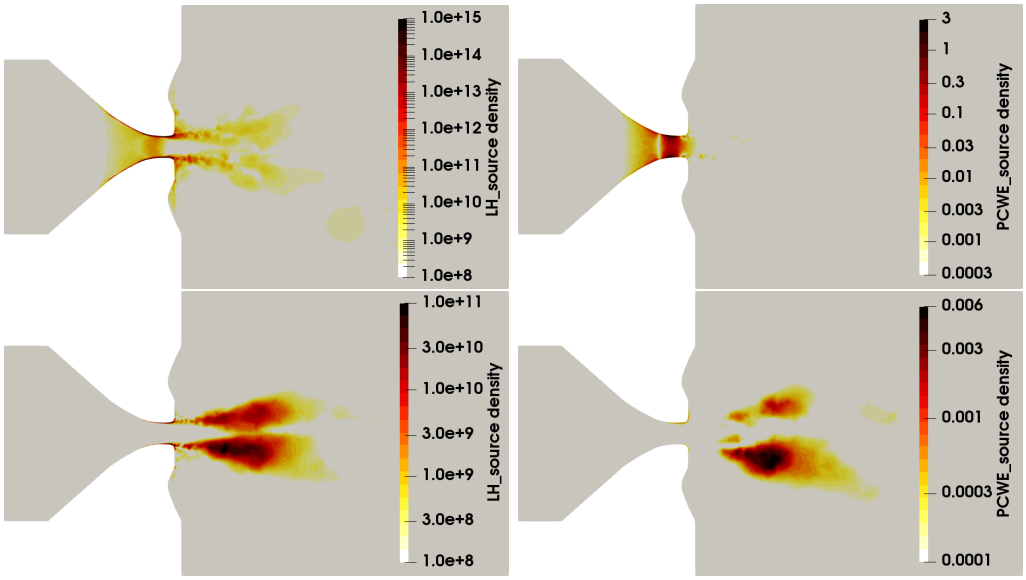



Figure 12: Computed power spectral densities of sound sources at **232** Hz (top) and **2486** Hz (bottom) for the LH (left) and PCWE approach (right). The color scale is logarithmic, and it is different for each figure.

The higher frequency sources like e.g. at 2486 Hz are mainly located in the supraglottal channel, see Figure 12 bottom. These sound sources can be associated with the free jet pouring out of an opening (glottis). In the LH case the sound sources at 2486 Hz are located along boundaries of the glottal jet, cp. [10]. The PCWE sound sources are situated in the supraglottal area typically following periodic series of vortices centers, nevertheless in this case the PSD graph is dominated by the merged spatial maxima of the first four vortices.

### 3.4.3. Sound propagation in the chosen acoustic domain

The sound sources of the Lighthill (LH) analogy and the simplified PCWE (sPCWE) approach, where the convection effects are disregarded on the left-hand side of (17) while keeping the full right-hand sound sources of (17), similar as in [18]. The computed sound sources are then used for their time propagation in the chosen acoustic domain M2 and the resulting acoustic pressure is observed in the microphone position. Its sound pressure levels at frequency domain up to 3 kHz are shown in Figure 13. Both approaches detect four frequency peaks matching very well the first four formants of the vocal tract model M2, but there are substantial differences in the SPL maxima. For the LH case the first frequency of 278 Hz reaches the highest SPL of circa 135 dB followed by frequency peaks 942 Hz and 2421 Hz, each gradually lowered by approximately 20 dB. The sPCWE approach is able to predict all four formants with more equal distribution of SPL, where the most significant peak with circa 110 dB is located at the frequency of $F_2$ contrary to the LH case. This is in agreement with [10] stating clear domination of the first frequency peak of the LH simulation, see also [13] and cf. [12]. Our previous results of [18] were spoiled by a wrong setting of PML contrary to the latest one, see [20, 15].

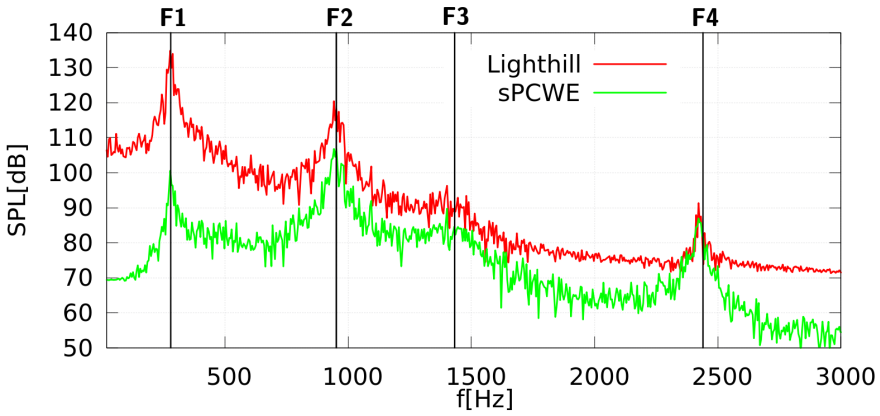



Figure 13: Sound pressure levels of acoustic pressure in the frequency domain, obtained by the LH analogy and the sPCWE approach at the microphone position (see Fig. 9). The black vertical lines mark the formants of acoustic domain M2, see Table 1.

The high values of SPL (comparable to a loud singing) are probably caused, first, by a relatively high prescribed pressure drop and the position of the microphone directly at the mouth opening, cf. [23]. Second, there is a generally different 2D fluid flow dynamics contrary to more complex 3D fluid flow dynamics (having impact on the aerodynamical sound sources). Finally, in agreement with [13], we regard the SPL results of the LH analogy as overestimated due to the absence of acoustic/hydrodynamic splitting, which leads to the superimposition of hydrodynamic quantities in the sound sources.

## 4. Conclusion

This article presents a complex problem of fluid-structure-acoustic interaction, motivated by human phonation. To simulate normal speech, a suitable approach is to use the fluid-structure interaction model to describe flow-induced VF vibrations as the main phonation mechanism, along with the application of acoustic analogies to separately solve the aeroacoustic problem. The both aforementioned problems are mathematically described and numerically approximated using FEM-based solvers.

The FSI numerical results compare flow characteristics for three inlet boundary conditions, showing that the penalization BC effectively controls maximal pressure difference during the channel closing phase. The simulation of flutter regime is documented by phase portraits of the selected point and by the curve plotting the dependence of the transglottal pressure on the gap.

In the acoustic results, the resonant acoustic frequencies of different acoustic domains are first investigated. Then the sound source analysis reveals the major sound source distribution at the glottis for low frequencies connected to VF vibration, while the majority of high-frequency sources is located at the supraglottal area. Finally, the acoustic pressure at the mouth position is obtained by the propagation of sound sources in time. Its SPL shows that the formant frequencies are the most dominant ones, as expected for the simulation without VF contact. The results of the Lighthill analogy obviously overestimates SPL, while the sPCWE results seem promising.

## Acknowledgements

The work was supported by grant No. SGS24/120/OHK2/3T/12 of CTU in Prague and from Premium Academiae of Prof. Nečasová. This work was supported by the Institute of Mathematics of the Czech Academy of Sciences (RVO:67985840). The authors gratefully acknowledge the Center of Advanced Aerospace Technology (CZ.02.1.01/0.0/0.0/16_019/0000826) at the Czech Technical University in Prague for awarding the access to computing facilities.

## References


[1] Delfs, J.: *Basics of Aeroacoustics*. Technische Universitaet Braunschweig, 2016.

[2] Feistauer, M., Sváček, P., and Horáček, J.: Numerical simulation of fluid-structure interaction problems with applications to flow in vocal folds. In: T. Bodnár, G. P. Galdi, and S. Nečasová (Eds.), *Fluid-structure Interaction and Biomedical Applications*, Birkhauser, 2014.

[3] Girault, V. and Raviart, P. A.: *Finite element methods for Navier-Stokes equations*. Springer-Verlag, 1986.

[4] Horáček, J., Radolf, V., and Laukkanen, A. M.: Experimental and computational modeling of the effects of voice therapy using tubes. J. Speech Lang. Hear. Res., 2019 pp. 1–18.

[5] Horáček, J., Šidlof, P., and Švec, J.: Numerical simulation of self-oscillations of human vocal folds with Hertz model of impact forces. J. Fluids Struct. **20** (2005), 853–869.

[6] Hüppe, A. and Kaltenbacher, M.: Spectral finite elements for computational aeroacoustics using acoustic perturbation equations. J. Comput. Acoust. **20** (2012), 1240 005.

[7] Kaltenbacher, M.: *Numerical simulation of mechatronic sensors and actuators: finite elements for computational multiphysics*. Springer, 2015.

[8] Kosík, A., Feistauer, M., Hadrava, M., and Horáček, J.: Numerical simulation of the interaction between a nonlinear elastic structure and compressible flow by the discontinuous Galerkin method. Appl. Math. Comput. **267** (2015), 382–396.

[9] Lighthill, M. J.: On sound generated aerodynamically. I. General theory. In: *Proceedings of the Royal Society of London*, vol. 211. The Royal Society, 1952 pp. 564–587.

[10] Lodermeyer, A. et al.: Aeroacoustic analysis of the human phonation process based on a hybrid acoustic PIV approach. Experiments in Fluids **59** (2018).

[11] Neustupa, T.: Existence of a steady flow through a rotating radial turbine with an arbitrarily large inflow and an artificial boundary condition on the outflow. J. Appl. Math. Mech. **103** (2023).

[12] Schoder, S. et al.: Aeroacoustic sound source characterization of the human voice production-perturbed convective wave equation. Appl. Sci. **11** (2021), 2614.

[13] Šidlof, P., Zörner, S., and Hüppe, A.: A hybrid approach to the computational aeroacoustics of human voice production. Biomechanics and Modeling in Mechanobiology **14** (2014), 473–488.

[14] Story, B. H., Titze, I. R., and Hoffman, E. A.: Vocal tract area functions from magnetic resonance imaging. JASA **100** (1996), 537–554.

[15] Sváček, P. and Valášek, J.: Numerical Simulation of Fluid-Structure-Acoustic Interactions Models of Human Phonation Process. In: Bodnár, T., Galdi, G.P., Nečasová, Š. (Eds.), *Fluids Under Control. Advances in Mathematical Fluid Mechanics*, Birkhäuser, Cham, 2023.

[16] Sváček, P. and Horáček, J.: Finite element approximation of flow induced vibrations of human vocal folds model: Effects of inflow boundary conditions and the length of subglottal and supraglottal channel on phonation onset. Appl. Math. Comput. **319** (2018), 178–194.

[17] Titze, I. R.: *Principles of voice production*. Prentice Hall, 1994.

[18] Valášek, J., Kaltenbacher, M., and Sváček, P.: On the application of acoustic analogies in the numerical simulation of human phonation process. Flow Turbul. Combust. **102** (2019), 129–143.

[19] Valášek, J. and Sváček, P.: On aerodynamic force computation in fluid-structure interaction problems - comparison of different approaches. J. Comput. Appl. Math. **429** (2023), 115–208.

[20] Valášek, J. and Sváček, P.: Aeroacoustic simulation of human phonation based on the flow-induced vocal fold vibrations including their contact. Adv. Eng. Softw. **194** (2024).

[21] Valášek, J., Sváček, P., and Horáček, J.: On suitable inlet boundary conditions for fluid-structure interaction problems in a channel. Applications of Mathematics **64** (2019), 225–251.

[22] Valášek, J., Sváček, P., and Horáček, J.: The influence of different geometries of human vocal tract model on resonant frequencies. In: D. Šimurda and T. Bodnár (Eds.), *Topical problems of fluid mechanics 2018*. Institute of Thermomechanics, AS CR, 2018 pp. 307–314.

[23] Zörner, S. and Kaltenbacher, M.: Fluid-structure-acoustic interaction algorithms and implementations using the finite element method. In: *Eccomas*, Vol. 2010. 2010 p. 28.

# LIST OF PARTICIPANTS

**Monika Balázsová**, balazsova@math.cas.cz
Matematický ústav AV ČR, v. v. i., Praha

**Stanislav Bartoň**, s.barton@po.opole.pl
Opole Polytechnic University, Opole

**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

**Vít Břichňáč**, brichvit@fit.cvut.cz
Fakulta informačních technologií, ČVUT v Praze

**Jana Burkotová**, jana.burkotova@upol.cz
Katedra matematické analýzy a aplikací matematiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci

**Vít Dolejší**, dolejsi@karlin.mff.cuni.cz
Katedra numerické matematiky, Matematicko-fyzikální fakulta UK, Praha

**Cyril Fischer**, fischerc@itam.cas.cz
Ústav teoretické a aplikované mechaniky AV ČR, v. v. i., Praha

**Barbora Halfarová**, barbora.halfarova.st@vsb.cz
Vysoká škola báňská – Technická univerzita Ostrava, Ústav geoniky AV ČR, v. v. i., Ostrava

**Tomáš Hammerbauer**, tom.hamm@seznam.cz
Katedra numerické matematiky, Matematicko-fyzikální fakulta UK, Praha

**Hana Honnerová**, hhornik@ntis.zcu.cz
Katedra matematiky, Fakulta aplikovaných věd, Západočeská univerzita v Plzni

**Jan Chleboun**, jan.chleboun@cvut.cz
Katedra matematiky, Fakulta stavební ČVUT v Praze

**Lukáš Kapera**, lukas.kapera.st@vsb.cz
Katedra aplikované matematiky, Fakulta elektrotechniky a informatiky, Vysoká škola báňská – Technická univerzita Ostrava

**Radka Keslerová**, Radka.Keslerova@fs.cvut.cz
Ústav technické matematiky, Fakulta strojní ČVUT v Praze

**Michal Kočvara**, m.kocvara@bham.ac.uk
School of Mathematics, University of Birmingham

**Alexej Kolcun**, alexej.kolcun@ugn.cas.cz
Ústav geoniky AV ČR, v. v. i., Ostrava

**Radek Kučera**, radek.kucera@vsb.cz
Vysoká škola báňská – Technická univerzita Ostrava,

**Václav Kučera**, kucera@karlin.mff.cuni.cz
Katedra numerické matematiky, Matematicko-fyzikální fakulta UK, Praha

**Jan Lamač**, jan.lamac@cvut.cz
Katedra matematiky, Fakulta stavební ČVUT v Praze

**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

**Ladislav Lukšan**, luksan@cs.cas.cz
Ústav informatiky AV ČR, v. v. i., Praha

**Zbyšek Machaczek**, zbysek.machaczek@vsb.cz
Vysoká škola báňská – Technická univerzita Ostrava

**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

**Tomáš Marhan**, tomas.marhan@fs.cvut.cz
Ústav technické matematiky, Fakulta strojní ČVUT v Praze

**Ctirad Matonoha**, matonoha@cs.cas.cz
Ústav informatiky AV ČR, v. v. i., Praha

**Luděk Nechvátal**, nechvatal@fme.vutbr.cz
Fakulta strojního inženýrství, Vysoké učení technické v Brně

**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

**Martin Plešinger**, martin.plesinger@tul.cz
Katedra matematiky a didaktiky matematiky, Fakulta přírodovědně-humanitní a pedagogická, Technická univerzita v Liberci

**Stefano Pozza**, pozza@karlin.mff.cuni.cz
Katedra numerické matematiky, Matematicko-fyzikální fakulta UK, Praha

**Adam Růžička**, adam.ruzicka.st@vsb.cz
Ústav geoniky AV ČR, v. v. i., Vysoká škola báňská – Technická univerzita Ostrava

**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., Praha

**Jan Stebel**, jan.stebel@tul.cz
Fakulta mechatroniky, informatiky a mezioborových studií, Technická univerzita v Liberci

**Petr Sváček**, petr.svacek@fs.cvut.cz
České vysoké učení technické v Praze, Fakulta strojní

**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

**Jakub Šístek**, sistek@math.cas.cz
Matematický ústav AV ČR, v. v. i., 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**, radek.varga@vsb.cz
Vysoká škola báňská – Technická univerzita Ostrava

**Tomáš Vejchodský**, vejchod@math.cas.cz
Matematický ústav AV ČR, v. v. i., Praha