Vertr.-Prof. Dr. Sigrun Ortleb
Dr. Satyvir Singh
Donat Weniger

Applied and Computational Mathematics | RWTH AACHEN UNIVERSITY

Numerical Methods for PDEs | WS 2025/26
Programming Exercise 2 | 05.12.2025

Notes concerning the submission of a programming exercise (read carefully!):

- The programming exercises should be completed in **your Moodle group**.
- You can use whatever programming language you like.
- Please make an appointment for your group attestation using the **Moodle appointment planer**. Attestations will take place 05.12.2025, you can register until 02.12.2025.
- Every group member needs to take an active role during the attestation.

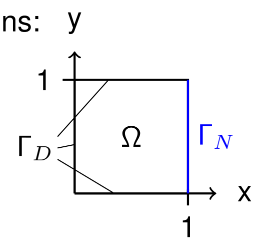
Finite Elements in 2D

In this programming exercise, you will implement the Finite Element method for the Poisson and Stokes equations. We will give you mesh data structures in `python` that will help you to implement the finite element method in 2D (`mesh_ops.py`). We also provide several meshes. In the next two global exercise sessions, an introduction to the provided code will be given. Alternatively, you may also rewrite the provided code in your preferred programming language.

Poisson

We consider the Poisson problem in two dimensions:

$$
\begin{aligned}
-\Delta u &= f && \text{in } \Omega \\
u &= 0 && \text{on } \Gamma_D = \Omega_2 \\
\partial_n u &= 0 && \text{on } \Gamma_N = \Omega_3
\end{aligned}
$$



on the unit square $\Omega \in [0,1] \times [0,1]$. Using the finite element method on unstructured triangular grids, we can determine approximations to the solution u.

a) Solve the Poisson problem using piecewise linear ansatz functions on triangles (P1 elements) with $f = 1$. Check your result on the toy-mesh `unitSquare1.msh`. The solution should look like in Figure 1 with the point value $u(0.5, 0.5) \approx 0.0833$. The A matrix and the b and u_N vectors should have the following values during program execution (dots represent zeros):

   - After assembly, before applying boundary conditions:

$$
A = \begin{pmatrix} 1 & \cdot & \cdot & \cdot & -1 \\ \cdot & 1 & \cdot & \cdot & -1 \\ \cdot & \cdot & 1 & \cdot & -1 \\ \cdot & \cdot & \cdot & 1 & -1 \\ -1 & -1 & -1 & -1 & 4 \end{pmatrix}, \quad b = \begin{pmatrix} 0.1667 \\ 0.1667 \\ 0.1667 \\ 0.1667 \\ 0.3333 \end{pmatrix}
$$

   - After setting the boundary values:

$$
A = \begin{pmatrix} 1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & 1 & \cdot \\ -1 & -1 & -1 & -1 & 4 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.3333 \end{pmatrix}, \quad u_N = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.0833 \end{pmatrix}
$$

b) Solve the problem on the finer mesh `unitSquare2.msh` and for the right-hand side $f(x, y) = \sin(2\pi x)\cos(2\pi y)$.

c) **Bonus** Implement second order finite elements (P2 elements).

d) **Bonus** Implement a Neumann boundary condition that is not zero, e.g. $\partial_n u = g$ on Γ_N with $g = 2$.

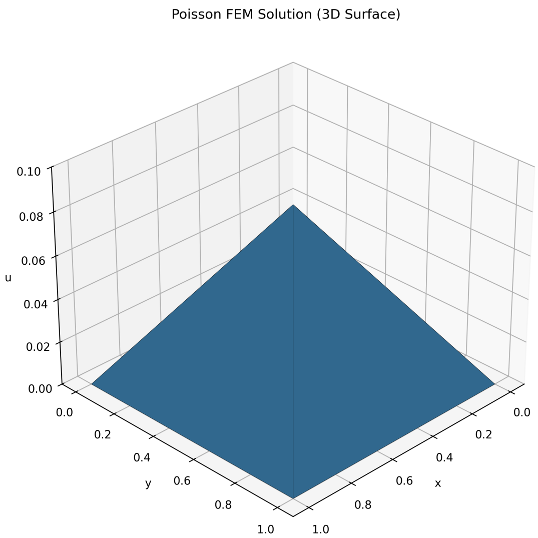
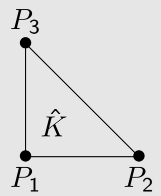


Figure 1: Numerical solution of the Poisson problem with $f(x, y) \equiv 1$ on the mesh `unitSquare1.msh`. Middle point as value $u(0.5, 0.5) \approx 0.0833$.

Hint: In order to create the stiffness matrix, you should proceed element by element and iterate over all triangles. In other words, you should be computing the so-called element stiffness matrices. For each triangle, you should transform the integrals to the reference triangle $\hat{K}$. [a] The integral transformation from an arbitrary triangle K to the reference triangle $\hat{K}$ is given by

$$\int_K \nabla_x \phi_i(\underline{x}) \nabla_x \phi_j(\underline{x})\, dx = \int_{\hat{K}} (D_\xi \underline{\Phi}_K^T)^{-1} \nabla_\xi \underbrace{\phi_i(\underline{x}(\underline{\xi}))}_{\hat{\phi}_i(\underline{\xi})} (D_\xi \underline{\Phi}_K^T)^{-1} \nabla_\xi \underbrace{\phi_j(\underline{x}(\underline{\xi}))}_{\hat{\phi}_j(\underline{\xi})} \,|\det D\underline{\Phi}_K|\, d\underline{\xi},$$

where $\underline{x}(\underline{\xi}) := \underline{\phi}_K(\underline{\xi})$ is the affine map that transforms $\hat{K}$ to K, and $D\underline{\phi}_K$ is the associated Jacobian of this transformation. The integration on the reference triangle should be done numerically. Use the quadrature points and weights provided by the class `MeshOperations` for numerical integration on $\hat{K}$.



Note the numbering of the nodes of the reference triangle!

[a] The reference triangle $\hat{K}$ is determined by the nodes $(0, 0)$, $(1, 0)$ and $(0, 1)$.

Stokes

The Stokes equation reads

$$
\begin{aligned}
-\Delta u + \nabla p = f \quad &\text{in } \Omega, \\
\operatorname{div} u = 0 \quad &\text{in } \Omega,
\end{aligned}
$$

with $u : \Omega \subset \mathbb{R}^2 \to \mathbb{R}^2$ and $p : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$.

We consider the flow around a cylinder in a 2D channel. We have $f = 0$ and the following boundary conditions:

$$
\begin{aligned}
u_1 = (y-1)(y+1), \quad u_2 = 0 \quad &\text{on } \Omega_2, \\
(\nabla \boldsymbol{u} \cdot \boldsymbol{n} - p\boldsymbol{n}) = 0 \quad &\text{on } \Omega_3, \\
u_1 = u_2 = 0 \quad &\text{on } \Omega_4 \cup \Omega_5,
\end{aligned}
$$

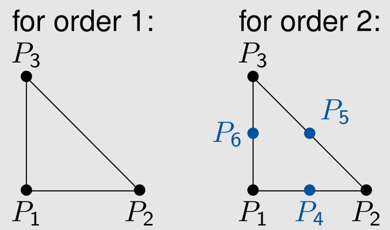
where

$$
\begin{aligned}
\Omega &:= (-1,1) \times (-1,1) \setminus \left\{ (x,y) \in \mathbb{R}^2 | x^2 + y^2 \leq \left(\frac{2}{10}\right)^2 \right\} \\
\Omega_2 &:= \{(x,y) \in [-1,1] \times [-1,1] | x = -1\} \text{ (left inflow)}, \\
\Omega_3 &:= \{(x,y) \in [-1,1] \times [-1,1] | x = 1\} \text{ (right outflow)}, \\
\Omega_4 &:= \left\{ (x,y) \in \mathbb{R}^2 | x^2 + y^2 = \left(\frac{2}{10}\right)^2 \right\} \text{ (cylinder)}, \\
\Omega_5 &:= \{(x,y) \in [-1,1] \times [-1,1] | y = -1 \ \vee \ y = 1\} \text{ (top and bottom)}.
\end{aligned}
$$

Hint: Impose the inflow and wall (top, bottom, and cylinder) boundary conditions strongly by modifying the system matrix and, depending on your implementation, the right-hand side. The outflow *do nothing* boundary condition has its name because we simply leave away the corresponding boundary terms (it is zero). Use pen and paper to make this clear to yourself.

a) Implement the mixed finite element method for the Stokes equations using P2 elements for the velocity and P1 elements for the pressure. We will stick to the mesh `unitSquareStokes.msh` for this exercise.

   Hint: The node numbering for the reference triangle is as follows:



   Without further additions, your system matrix will be singular. Impose a single pressure value to fix this issue (e.g. set $p = 0$ at one node).

b) Plot the velocity components and the pressure field. Check that your boundary conditions are fulfilled.

c) Verify your implementation. The velocity field in x-direction should look like in Figure 2.

d) Change the boundary condition on Ω_2 to $u_1 = y(y-1)(y+1), \quad u_2 = 0$. How does the behavior of the flow change?

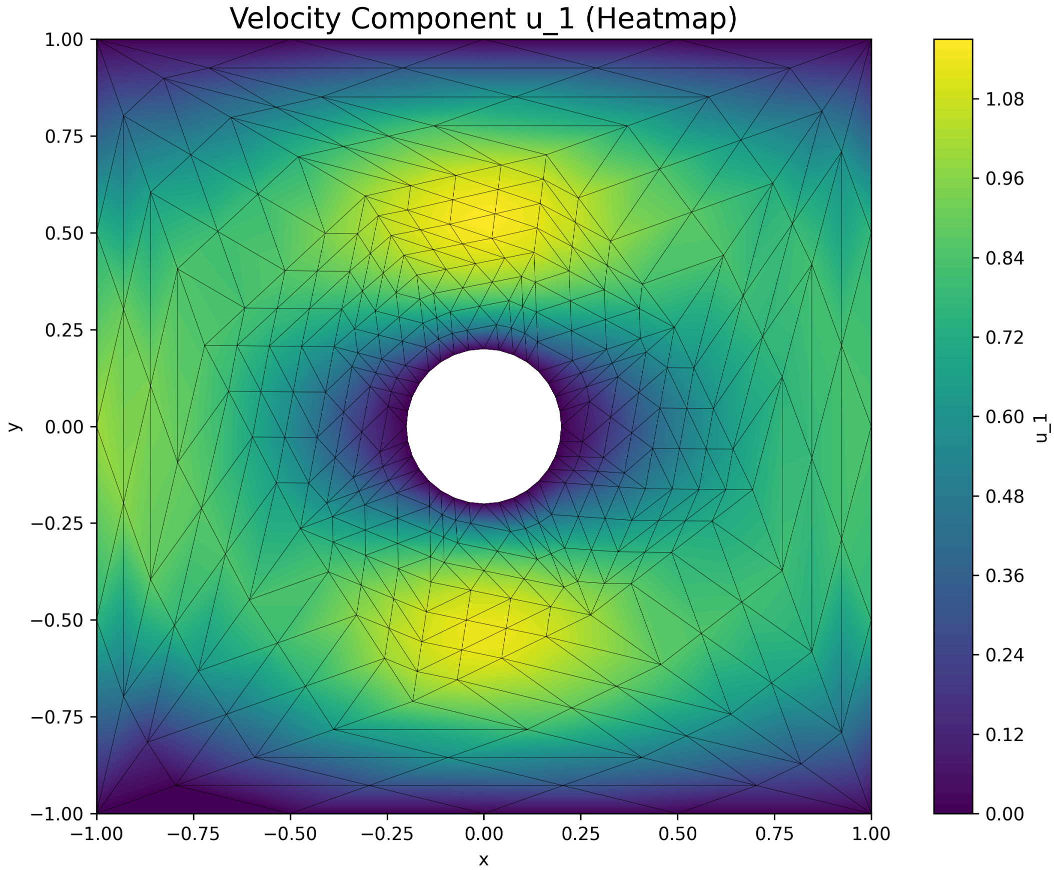


Figure 2: x-direction of the velocity field for the Stokes flow around a cylinder in a channel.

20 + 20 = 40 Points