Numerics of incompressible flows 1 (SS 25)
Homework 1

(Prof. Dr. Eberhard Bänsch, M.Sc. Nick Schneider)

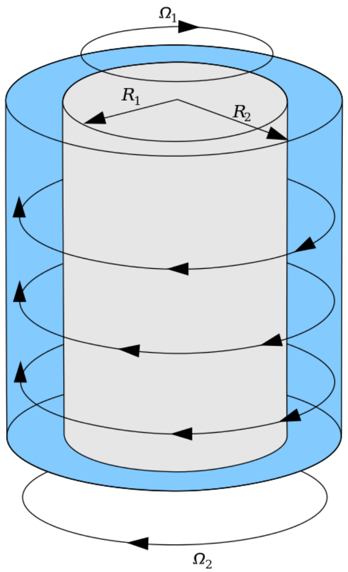
Deadline for submission: May 14th, 2025
Department of Mathematics, FAU Erlangen-Nuremberg

Note: Hand in your assignments in pairs of two or three, either on paper at the start of the exercise class or online via StudOn. Ensure that your names are written on your assignments.

Exercises are optional, but if you finish the course with at least 50% of the attainable points, you will be awarded 1 grade better in the final exam (provided you pass with at least a 4.0). Similarly, at least 75% of the attainable points will earn you 2 grades better.

Exercise 1: Couette Flow I (6+4)

The goal of this exercise sheet and the next is to find an analytical expression for the flow between two concentric cylinders of infinite height with radii $0 < R_1 < R_2$, i.e. in the domain $\Omega := \{(x,y,z) \in \mathbb{R}^3 \mid R_1 < \sqrt{x^2+y^2} < R_2\}$.



a) The Couette flow is more naturally described in cylindrical coordinates

$$\Omega \ni (x,y,z) \mapsto (r,\varphi,z) := \left(\sqrt{x^2+y^2}, \arctan\left(\frac{y}{x}\right), z\right).$$

Let e_r, e_φ, e_z denote the unit cylindrical vector fields

$$e_r(r,\varphi,z) := \begin{pmatrix}\cos(\varphi)\\ \sin(\varphi)\\ 0\end{pmatrix},\ e_\varphi(r,\varphi,z) := \begin{pmatrix}-\sin(\varphi)\\ \cos(\varphi)\\ 0\end{pmatrix},\ e_z := \begin{pmatrix}0\\0\\1\end{pmatrix}.$$

Prove that for a scalar-valued function $f(x,y,z) = \widehat{f}(r,\varphi,z)$ (henceforth also simply denoted by f), we have

$$\nabla f = \partial_r f e_r + \frac{1}{r}\partial_\varphi f e_\varphi + \partial_z f e_z$$

$$\Delta f = \frac{1}{r}\partial_r(r\partial_r f) + \frac{1}{r^2}\partial_{\varphi\varphi} f + \partial_{zz} f$$

Note: You may use without proof that for a vector field $v = v_x e_x + v_y e_y + v_z e_z = v_r e_r + v_\varphi e_\varphi + v_z e_z$ with $e_x := (1,0,0)^T$ and $e_y := (0,1,0)^T$ it holds

$$\text{div } v = \frac{1}{r}\partial_r(r v_r) + \frac{1}{r}\partial_\varphi v_\varphi + \partial_z v_z.$$

b) Prove that for any smooth vector field $v : \Omega \to \mathbb{R}^3$ there holds

$$(v\cdot\nabla)v = \frac{1}{2}\nabla\|v\|^2 - v \times \text{rot } v$$

with rot $v := \nabla \times v$ (the *rotation*, sometimes also denoted by curl v).

Hint: For $a,b,c \in \mathbb{R}^3$, $a\cdot(b\times c) = -b\cdot(a\times c) = b\cdot(c\times a)$ holds true. This is a consequence of the fact that $\det([a,b,c]) = a\cdot(b\times c)$ and the alternating property of the determinant.

Exercise 2: FEniCS (5+5)

As part of this course we will also cover programming exercises in the Python-based FEniCS library. On the computer terminals in the CIP-pool, this is already pre-installed, as well as the visualization program Paraview. If you want to use those on your private computers, you may follow the suggestions further below.

Whichever option you chose, you should spend some time carefully reading the tutorial demo script and the explanations in `https://olddocs.fenicsproject.org/dolfin/latest/python/demos/poisson/demo_poisson.py.html` in order to understand how FEniCS is working.

Hint: The program `demo_poisson.py` that you can download from this site is qualitatively identical to `demo_poisson_adapted.py` given in StudOn, but there I have changed some notation and the style with which the function at the Dirichlet boundary is defined, in order to make adapting the program in Exercise 2b) a little easier for you.

a) Run the given program `demo_poisson_adapted.py` via

```
python3 demo_poisson_adapted.py
```

in the terminal (or `python`, depending on your Python version). You will obtain a file called `poisson.pvd`. With this file, you can visualize the solution with the program Paraview. Opening the data set (Drag-and-drop is possible) will give you a two-dimensional heatmap of the solution. (You may have to click on the closed eye next to `solution.pvd` in the "Pipeline browser"). Now click on the "Apply"-button under "Properties". In the Task bar you find "Filters". There use "Alphabetical/Warp By Scalar" (or if you cannot see / find it, use "Search" and enter "Warp by Scalar"). This will give you a 3D-model of the solution. (Again, you may have to click on the closed eye next to `WarpByScalar1` in the "Pipeline browser". Also possibly clicking the "2D"-button, if it appears, is necessary to change from 2D-mode to 3D-mode.)

Hand in pictures / screenshots of the heatmap and the 3D-plot of the solution for the exercise points.

b) Make a copy of `demo_poisson_adapted.py` that you call `demo_poisson_adapted2.py`. Here, make the appropriate changes such that the new program solves the Poisson problem

$$\begin{cases} -\Delta u = -6 & \text{in } \Omega := (0,1)^2, \\ u = 1 + x^2 + 2y^2 & \text{on } \partial\Omega. \end{cases}$$

Hand in `demo_poisson_adapted2.py` for the exercise points.

Using FEniCS on private computer: In order to install FeniCS directly on your computer, having a version of Python installed is already required. Follow the information for the installation of FEniCS documented on `https://fenicsproject.org/download/archive/`. Beware that you might have to (re)activate the virtual environment corresponding to FEniCS in the command line terminal.

Paraview can be downloaded from `https://www.paraview.org/download/`.