Project 3

NUMN21/FMNN25: Advanced Numerical Algorithms in Python

Robert Klöfkorn and Andreas Langer

This describes a programming project in the course, devoted to parallel numerics and MPI.
This assignment has 4 tasks.

Background

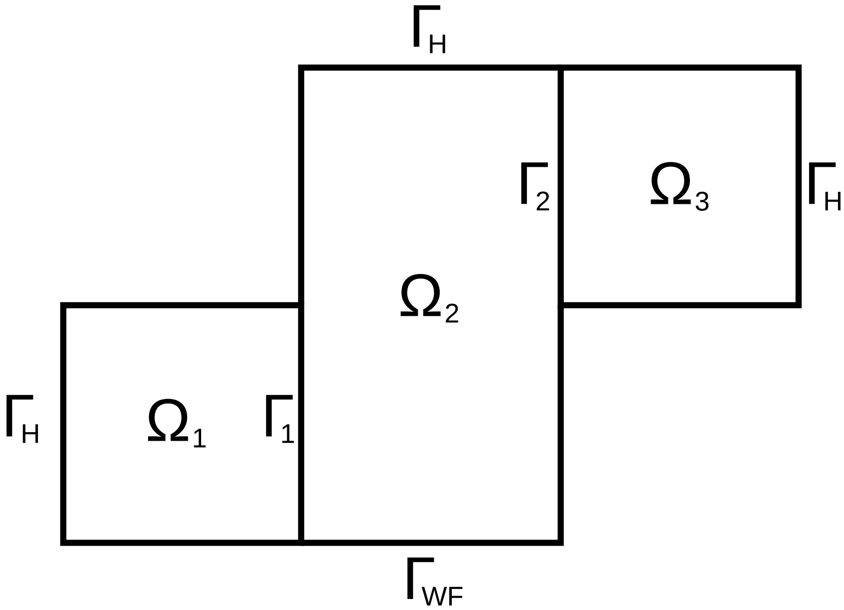

Figure 1: Layout of the 2-room apartment. Rooms 1 and 3 are of size 1 × 1, room 2 is of size 1 × 2

You decided to buy a 2-room apartment. When you see the layout of the apartment, you decide it is time to figure out if it's going to be warm in winter. To model a steady temperature distribution in the apartment, we can use the *Laplace equation*

$$\Delta u(x) = 0, \quad x \in \Omega \subset \mathbb{R}^2$$

for the temperature function $u(x)$. We use three kinds of boundary conditions to model different walls:

- normal wall $u(x) = 15, \quad x \in \partial\Omega \backslash (\Gamma_H \cup \Gamma_{WF})$,
- walls with heater $u(x) = 40, \quad x \in \Gamma_H$,
- wall with a big window $u(x) = 5, \quad x \in \Gamma_{WF}$.

See figure 1 for a specification of the boundaries.
To approximate the solution, use a Cartesian grid with an equidistant mesh width Δx where we define the approximations $u_{i,j} \approx u(x_{i,j})$ on nodes of the mesh. We then use *2nd order central differences*:

$$\Delta u(x_{i,j}) \approx \frac{u_{i,j+1} + u_{i,j-1} - 4u_{ij} + u_{i+1,j} + u_{i-1,j}}{\Delta x^2}.$$

Solve this using MPI in a Python implementation of the *Dirichlet-Neumann method*. To this end, consider each of the rooms as one domain, coupled along an interface with the others. Across these interfaces, prescribe Dirichlet conditions on one side and Neumann conditions for the other. Specifically, the outer rooms will have a Neumann conditions and the inner one a Dirichlet condition. As a solver on the subdomains, use `scipy.linalg.solve`. This gives the following iteration:
Given discrete temperature vectors u_1^k, u_2^k and u_3^k on the three domains, do

1. Determine u_2^{k+1} by solving the problem on Ω_2 with Dirichlet conditions at Γ_1 and Γ_2, given by u_1^k and u_3^k
2. Determine u_1^{k+1} and u_3^{k+1} by solving the problem on Ω_1 and Ω_3 with Neumann conditions at Γ_1 and Γ_2, given by u_2^{k+1}
3. Use relaxation: $u^{k+1} \leftarrow \omega u^{k+1} + (1-\omega)u^k$

Iterate ten times with $\omega = 0.8$ and use $\Delta x = \frac{1}{20}$.

Task 1

Write down the matrices that you get within the Dirichlet-Neumann iteration when you choose a mesh width of $\Delta x = 1/3$.

Task 2

Is the heating in the flat adequate?

Task 3

Plot the temperature distribution.

Task 4

> Optional: Vary the parameters, e.g. the temperature of the heaters, the number of iterations or the relaxation parameter.

Note: For this project you should install mpi4py. Those who use the anaconda distribution of Python can use for installation

```
pip install mpi4py
```

This might also work on windows computers, but we had no opportunity to test it. Check also https://mpi4py.readthedocs.io.