General Information: This assignment contains written and/or programming tasks. Combine all the answers to the written tasks in a single PDF document, named `{lastname}-written.pdf`. You can also scan or take pictures of (readable) handwritten papers. JPEG/PNG image files are accepted in this case and they should be named `{exercisenumber}-{lastname}-written.{jpeg/png}`. Make sure that we can follow the manual calculations. Do not combine too many small steps into one. The programming tasks have to be solved in *Julia* and the source code files have to be submitted using the following naming scheme: `{exercisenumber}-{lastname}.jl`.

(1) (2 points) Given is the following matrix:

$$M = \begin{pmatrix} 1 & 3 & 1 & 2 & 5 \\ 3 & 13 & 7 & 8 & 17 \\ 1 & 7 & 21 & 8 & 15 \\ 2 & 8 & 8 & 7 & 16 \\ 5 & 17 & 15 & 16 & 40 \end{pmatrix}$$

a) Compute by hand with Gaussian elimination its LU-decomposition. (Leave fractions as fractions.)

b) Describe its properties.

c) Compute Cholesky decomposition, if applicable.

d) Compute by hand $Mx = b$, where $b = (1, 1, 1, 1, 1)^T$. (Do not use inverses.)

e) How would you easily compute the inverse of matrix M?

(2) (3 points)

a) Compute by hand the first iteration of Jacobi method for the matrix given in the first exercise. Take as a starting vector $x_0 = (1, 1, 1, 1, 1)^T$ and a $b = (1, 1, 1, 1, 1)^T$.

b) In the provided code `iterative_solvers.jl` write the code for Jacobi and Gauß-Seidel iterations.

c) In order to allow for early termination, implement the computation of the Euclidean norm of the residual to track the solvers convergence:

$$||\mathbf{b} - \mathbf{A}\mathbf{x}^{(\mathbf{k})}||,$$

with the current estimate $\mathbf{x}^{(\mathbf{k})}$ – see function `errorNorm`.

d) Compare the methods. Moreover, for further comparison is the code for the conjugate gradient method provided. Briefly describe their algorithms and state main ideas, discuss their convergence, compare the number of steps necessary, explain why some of the methods are faster than the others, etc.

(3) (2 points) We want to find a solution to the steady state heat equation (Laplace's equation) in a square D, subject to certain boundary conditions:

$$\nabla^2 u(x, y) = \frac{\partial^2 u(x, y)}{\partial x^2} + \frac{\partial^2 u(x, y)}{\partial y^2} = 0, \quad (x, y) \in D$$
$$u(x, y) = F(x, y), \quad (x, y) \in \partial D$$

The temperature at the heaters, wall and window is set to be $30°C, 18°C$, and $10°C$, respectively.

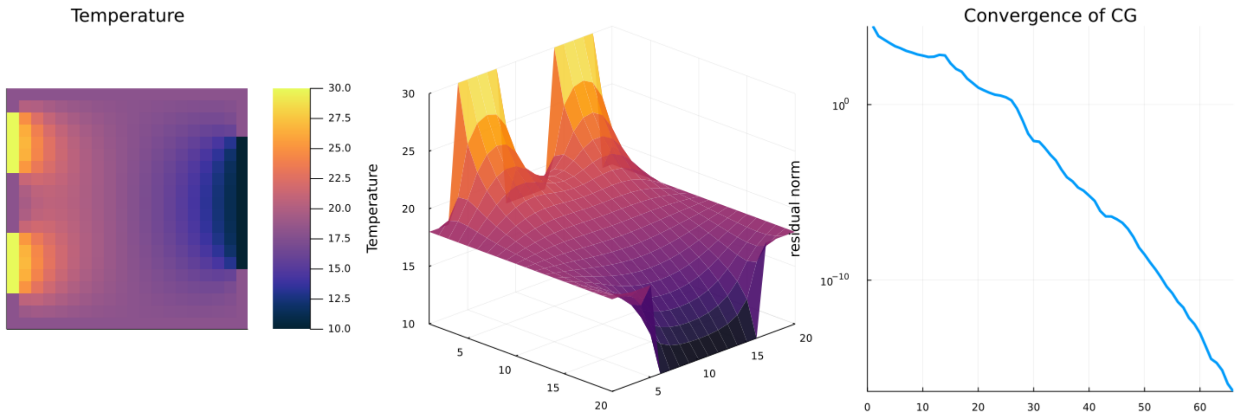

Figure 1: Results of *Exercise 3* for conjugate gradients method. The yellow color in the plot are two heaters (warm) and black color is the window (cold). The rest are walls in the room.

a) Approximate the partial derivatives on an uniform grid ($\Delta x = \Delta y = h$). Use the approximation of the first derivatives at i and $i-1$ (j and $j-1$) with forward and backward differences, to get the approximation of the second order. What is the resulting difference scheme? (You should get a known scheme.)

b) Use the methods from the previous exercise to solve the discretized Laplace's equation by adjusting the template file `steady_state_heat.jl`. (In case you run into computational issues decrease the number of grid points N.)

c) Solve the steady state heat equation (at different resolutions N) with all solvers and observe the convergence rate – save the plots as PNGs and add them together with the discussion to your written solution.