Allgemeine Informationen: Dieses Aufgabenblatt enthält schriftliche und/oder Programmieraufgaben. Bitte kombinieren Sie alle Lösungen zu den schriftlichen Aufgaben zu einem einzelnen PDF Dokument, welches Sie nach folgendem Schema benennen:`{lastname}-written.pdf`. Sie können Ihre Lösungen auch scannen oder fotografieren. Achten Sie in diesem Fall auf die Lesbarkeit. Es werden JPEG/PNG Bilddateien akzeptiert welche wie folgt benannt werden müssen:`{exercisenumber}-{lastname}-written.{jpeg/png}`. Stellen Sie sicher, dass alle Rechenschritte nachvollziehbar sind und kombinieren Sie nicht zu viele kleine Schritte zu einem einzelnen. Die Programmieraufgaben müssen in *Julia* gelöst sein und Ihr Quellcode sollte nach folgendem Schema benannt sein: `{exercisenumber}-{lastname}.jl`.

(1) (2 Punkte) Gegeben ist die folgende 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) Berechne mit Gauß'scher Eliminierung ihre LU-Zerlegung von Hand. (Lass Brüche als solche stehen.)

b) Beschreibe ihre Eigenschaften.

c) Berechne die Cholesky-Zerlegung, falls anwendbar.

d) Berechne $Mx = b$ von Hand, wobei $b = (1,1,1,1,1)^T$. (Verwende keine Inversen.)

e) Wie kann man die Inverse der Matrix M leicht berechnen?

(2) (3 Punkte)

a) Berechne die erste Iteration der Jacobi-Methode von Hand für die in der Aufgabe (1) gegebenen Matrix M. Nimm als Ausgangsvektor $x_0 = (1,1,1,1,1)^T$ und ein $b = (1,1,1,1,1)^T$.

b) Implementiere die Jacobi- und Gauß-Seidel-Iterationsvorschriften in `iterative_solvers.jl`.

c) Um eine frühzeitige Beendigung zu ermöglichen und um die Konvergenz des Solvers zu verfolgen, sollte die Berechnung der euklidischen Norm des Residuums implementiert werden:

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

mit der aktuellen Schätzung $\mathbf{x}^{(\mathbf{k})}$ – siehe Funktion `errorNorm`.

d) Die Implementierung für die konjugierte Gradientenmethode wird zur Verfügung gestellt. Vergleiche die Methoden. Beschreibe kurz die Algorithmen und gib die Kernideen an, beschreiben sie die Konvergenzverhalten, vergleiche die Anzahl der erforderlichen Schritte, erkläre, warum einige der Methoden schneller sind als die anderen, usw.

(3) (2 Punkte) Wir wollen eine Lösung der stationären Wärmegleichung (Laplace-Gleichung) in einem Quadrat D finden, welche bestimmten Randbedingungen unterliegt:

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

Die Temperatur an den Heizkörpern, der Wand und dem Fenster wird auf $30°C$, $18°C$ bzw. $10°C$ eingestellt.

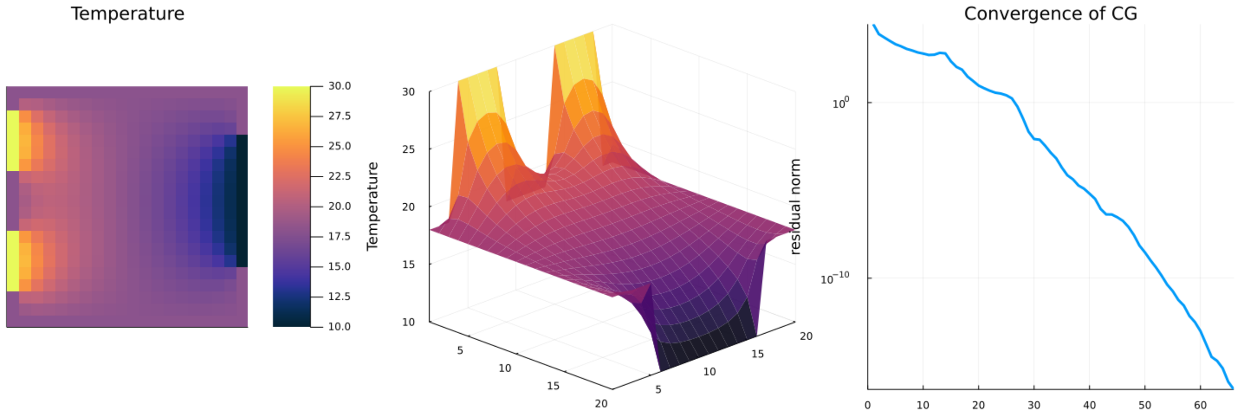

Abbildung 1: Ergebnisse von *Übung 3* für die Methode der konjugierten Gradienten. Die gelbe Farbe im Diagramm sind zwei Heizkörper (warm) und die schwarze Farbe ist das Fenster (kalt). Der Rest sind die Wände im Raum.

a) Approximiere die partiellen Ableitungen auf einem einheitlichen Gitter ($\Delta x = \Delta\, y \;=\; h$). Verwende die Approximation der ersten Ableitungen bei i und $i-1$ (j und $j-1$) mit Vorwärts- und Rückwärtsdifferenzen, um die Approximation zweiter Ordnung zu erhalten. Wie lautet das resultierende Differenzschema? (Du solltest ein bekanntes Schema erhalten.)

b) Verwende die Methoden aus der vorherigen Übung, um die diskretisierte Gleichung von Laplace zu lösen, indem Du das Template `steady_state_heat.jl` anpasst. (Falls Du auf Rechenprobleme stoßt, verringe die Anzahl der Gitterpunkte N)

c) Löse die stationäre Wärmegleichung (bei verschiedenen Auflösungen N) mit allen Solvern und beobachte die Konvergenzrate – speichern Sie die Plots als PNGs und füge sie zusammen mit Erklärungen der schriftliche Lösung hinzu.