ÜBUNG 3 ADJAZENZMATRIX

Sei I eine Indexmenge und $R \subset I \times I$ eine symmetrische und reflexive Relation mit

$$\max_i |\{j : (i,j) \in R\}| \leqslant K$$

Definiere eine zugehörige Matrix E durch

$$(E)_{ij} := \begin{cases} 1 & (i,j) \in R \\ 0 & \text{sonst} \end{cases}$$

Zeige $||E||_2 \leqslant K$. *5 Punkte*

ÜBUNG 4 OPTIMALER PARAMETER FÜR DAS RICHARDSON-VERFAHREN

Sei A eine symmetrische und positiv definite Matrix. Die Iterationsformel des Richardson-Verfahrens lautet

$$x_{k+1} = x_k + \omega \left(b - Ax_k\right).$$

Der minimale und maximale Eigenwert λ_{min} und λ_{max} von A sei bekannt.

1. Wie kann man die Eigenwerte der Iterationsmatrix beschränken?
2. Bestimmen Sie den optimalen Relaxationsparameter ω und den dazugehörigen Spektralradius.

5 Punkte

ÜBUNG 5 LÖSERKONVERGENZ IN ABHÄNGIGKEIT VON DEN ANFANGSWERTEN

In dieser Aufgabe betrachten wir die Laplace-Gleichung mit homogenen Dirichlet-Randbedingungen

$$\begin{aligned} -\Delta u &= 0 && \text{in } \Omega = (0,1)^2 \subset \mathbb{R}^2, \\ u &= 0 && \text{auf } \partial\Omega. \end{aligned}$$

Diese Gleichung hat offensichtlich die Lösung $u = 0$. Wir nutzen dies aus, um das Konvergenzverhalten verschiedener Löser zu visualisieren, indem wir verschiedene Startvektoren ungleich 0 vorgeben und beobachten, wie sie durch die iterativen Löser gedämpft werden.

Dazu gibt es im Modul `dune-parsolve` die Datei `istl.cc`, in der eine Reihe Löser definiert werden. Das Jacobi- und das Gauß-Seidel-Verfahren werden in der Regel nur als Vorkonditionierer eingesetzt und finden sich daher in ISTL in der Datei `preconditioners.hh`. Man kann sie trotzdem als eigenständige Löser verwenden, dafür gibt es einen trivialen Löser `Loop`, der in jeder Iteration lediglich den Vorkonditionierer anwendet. Der Rest der Löser ist wie erwartet in der Datei `solvers.hh` zu finden.

Modifizieren Sie die Datei `uebung02.cc` so, dass sie analog zu `istl.cc` Löser anlegt und nacheinander VTK-Ausgaben für alle Kombinationen folgender Koomponenten erstellt:

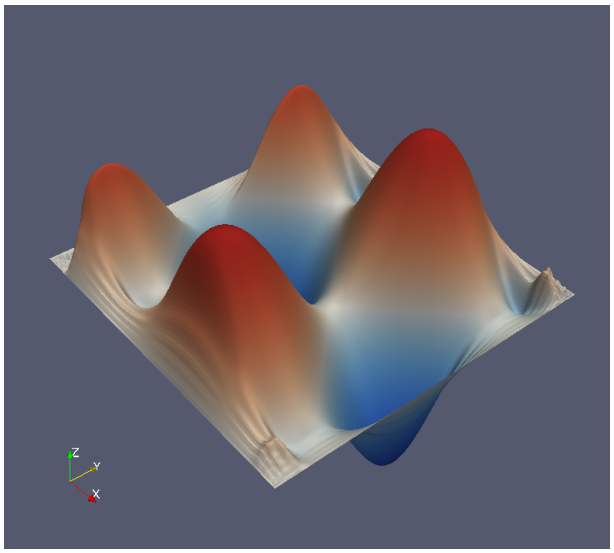
- Löser: Jacobi, Gauß-Seidel, Steepest Descent und CG
- Startwerte: Die Startwert-Funktionen

$$\begin{aligned} u_1(x,y) &= 1, \\ u_2(x,y) &= cos(10x) + sin(10y), \\ u_3(x,y) &= cos(100x) + sin(100y) \end{aligned}$$

- Iterationszahlen: 1, 10 oder 100 Iterationen

In der gegebenen Datei existiert die Matrix, die Sie für den direkten Einsatz der ISTL-Löser brauchen, nur innerhalb der PDELab-Klassen. Überlegen Sie sich, wie Sie diese Matrix extrahieren können (Tip: die GridOperator-Referenz der Assembler-Klasse), oder duplizieren Sie den Code, der innerhalb von PDELab die Matrix erzeugt.

Die entstehende Näherungslösung ist für die obigen Probleme jeweils gleich dem Restfehler des linearen Lösers, da die exakte Finite-Elemente-Lösung ebenfalls identisch verschwindet. Diesen Fehler können wir nun mittels ParaView visualisieren:



Falls Sie Ihr Programm so schreiben, dass es auch für die übrigen Iterationen passende VTK-Dateien schreibt, können Sie sich den Verlauf der Konvergenz noch besser ansehen.

Untersuchen Sie die Konvergenzrate der einzelnen Löser. Hat die Startbedingung einen Einfluss? Welche weiteren Modifikationen der Aufgabenstellung wären möglich, und was würde sich am Konvergenzverhalten ändern? *10 Punkte*