

Übungsblatt 4: Lineare Gleichungssysteme II

11. November 2016

Die benötigten Matrizen sind auch als Textdateien zur Verfügung gestellt, sodass sie auch eingelesen werden können (Im Zweifel stimmen die Werte aus den Dateien).

Aufgabe 4.1: Berechnung von Determinanten (5 Punkte)

Eine Matrix \mathbf{A} kann dargestellt werden als $\mathbf{A} = \mathbf{L} \cdot \mathbf{R}$ (LR-Zerlegung), wobei erstere eine untere und letztere eine obere Dreiecksmatrix ist. Hierbei können die Diagonalelemente von \mathbf{L} als 1 gewählt werden.

Hat man diese Zerlegung erreicht, ist es einfach über den Determinantenproduktsatz die Determinante der Matrix \mathbf{A} zu berechnen.

- a) (3 Punkte) Implementieren Sie eine Funktion, die mittels LR-Zerlegung die Determinante einer ihr übergebenen Matrix berechnet .
- b) (2 Punkte) Berechnen Sie die Determinante der folgenden zwei Matrizen:

$$\mathbf{A}_1 = \begin{pmatrix} 3 & 0 & -6 & -2 & -5 \\ 8 & 6 & 9 & -1 & -9 \\ -2 & -1 & 4 & 1 & -8 \\ -7 & -8 & -6 & 8 & -2 \\ -3 & 9 & -9 & 9 & 3 \end{pmatrix}$$

$$\mathbf{A}_2 = \begin{pmatrix} 5 & -4 & -6 & -6 & -4 & 3 & -1 & -10 \\ -1 & 0 & -9 & 5 & 3 & 6 & -5 & -9 \\ -6 & -9 & 3 & 5 & -1 & -7 & 0 & -8 \\ 3 & -2 & -9 & -2 & -4 & -7 & -10 & -9 \\ 4 & 2 & 2 & -8 & 5 & 1 & -8 & -8 \\ -10 & -4 & 1 & 3 & -1 & 3 & -10 & 4 \\ 7 & 7 & -1 & 7 & -1 & -9 & 1 & -6 \\ -3 & 9 & 8 & 5 & -4 & -8 & 9 & 2 \end{pmatrix}$$

Zur Kontrolle Ihrer Ergebnisse können Sie `numpy.linalg.det` verwenden.

Aufgabe 4.2: Iterative Lösung von Gleichungssystemen (15 Punkte)

Wir betrachten im Folgenden ein lineares Gleichungssystem der Form $\mathbf{Ax} = \mathbf{b}$. Zerlegt man die Matrix $\mathbf{A} = \mathbf{D} + \mathbf{L} + \mathbf{R}$ in eine Diagonalmatrix \mathbf{D} sowie eine untere und eine obere Dreiecksmatrix (\mathbf{L} und \mathbf{R}), so kann man ein Fixpunktverfahren für dessen Lösung verwenden.

Beim Jacobi Verfahren fängt man mit einem beliebiger Startvektor $\mathbf{x}^{(0)}$ an und löst dann iterativ

$$\mathbf{x}^{(k+1)} = -\mathbf{D}^{-1} [(\mathbf{L} + \mathbf{R}) \mathbf{x}^{(k)} - \mathbf{b}]. \quad (1)$$

Als Konvergenzkriterium kann der Abstand zweier aufeinanderfolgender Vektoren gewählt werden $\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|^2 < \epsilon^2$, wobei ϵ die gewünschte Genauigkeit ist (Verwenden Sie die euklidische Norm).

Bemerkung: Wir verzichten in dieser Aufgabe darauf, eine Pivotisierung durchzuführen.

Das Gauss-Seidel Verfahren ähnelt dem Jacobi Verfahren. Allerdings nutzt man nun die Tatsache, dass beim komponentenweisen Lösen der Zeile j im Schritt $k+1$ schon $j-1$ Komponenten von $\mathbf{x}^{(k+1)}$ bestimmt worden sind. Die Iterationsvorschrift lautet

$$\mathbf{x}^{(k+1)} = -(\mathbf{D} + \mathbf{L})^{-1} \mathbf{R} \mathbf{x}^{(k)} + (\mathbf{D} + \mathbf{L})^{-1} \mathbf{b} \quad (2)$$

Um außerdem noch die Information über die gewonnene Verbesserung im letzten Iterationsschritt auszunutzen wird in Relaxationsverfahren ein Zwischenschritt eingeführt: Erst wird ein Schritt nach Gauss-Seidel gerechnet und der erhaltene aktualisierte Vektor $\tilde{\mathbf{x}}^{(k+1)}$, mit dem vorherigen nach $\mathbf{x}^{(k+1)} = \omega \tilde{\mathbf{x}}^{(k+1)} + (1 - \omega) \tilde{\mathbf{x}}^{(k)}$ gewichtet. $\omega \geq 1$ bezeichnet man als Relaxationsparameter.

Die Iterationsvorschrift lautet dann

$$\mathbf{x}^{(k+1)} = (\mathbf{D} + \omega \mathbf{L})^{-1} [(1 - \omega) \mathbf{D} - \omega \mathbf{R}] \mathbf{x}^{(k)} + \omega (\mathbf{D} + \omega \mathbf{L})^{-1} \mathbf{b} \quad (3)$$

a) (4 Punkte) Implementieren Sie eine Funktion für das Jacobi-Verfahren. Ihre Funktion sollte als Argumente \mathbf{A} , \mathbf{b} nehmen, sowie als optionale Argumente $\mathbf{x}^{(0)}$, ϵ und eine maximale Anzahl an

Iterationsschritten $N_{\text{iter,max}}$. Verwenden Sie $\mathbf{x}^{(0)} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$, $\epsilon = 10^{-9}$, $N_{\text{iter,max}} = 100$ als Standardwerte für die optionalen Argumente. Die Funktion soll den Lösungsvektor sowie die benötigte Anzahl an Iterationsschritten zurückgeben. Wurde keine Konvergenz in $N_{\text{iter,max}}$ Iterationsschritten erzielt, soll eine Fehlermeldung ausgegeben werden.

Führen Sie in jedem Iterationsschritt eine a priori und eine a posteriori-Abschätzung des Restfehlers nach ($\bar{\mathbf{x}}$ bezeichnet die exakte Lösung)

$$\|\mathbf{x}^{(k)} - \bar{\mathbf{x}}\| \leq \frac{\|\mathbf{B}\|^k}{1 - \|\mathbf{B}\|} \|\mathbf{x}^{(1)} - \mathbf{x}^{(0)}\| \quad \text{a-priori} \quad (4)$$

und

$$\|\mathbf{x}^{(k)} - \bar{\mathbf{x}}\| \leq \frac{\|\mathbf{B}\|}{1 - \|\mathbf{B}\|} \|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\| \quad \text{a-posteriori} \quad (5)$$

durch und plotten Sie den Restfehler (**B ist die Iterationsmatrix, die je nach Verfahren anders aussieht. Benutzen Sie als Norm den Spektral-Radius $\text{spr}(B)$. Den können Sie mit `np.real(max(np.linalg.eig(B)))` bestimmen**).

- b) (3 Punkte) Implementieren Sie analog zur vorherigen Aufgabe eine Funktion für das Gauss-Seidel Verfahren, mit denselben Argumenten, Standard- und Rückgabewerten wie in 4.2a), sowie auch der Fehlerabschätzung.
- c) (3 Punkte) Erweitern Sie die Funktion für das Gauss-Seidel Verfahren oder schreiben Sie eine neue Funktion, so dass Sie Überrelaxation nutzen können. Verwenden Sie dieselben Standard- und Rückgabewerten wie in 4.2a) und 4.2b) und der Fehlerabschätzung, aber ergänzen Sie die Argumente um den Relaxationsfaktor ω .
- d) (5 Punkte) Lösen Sie das lineare Gleichungssystem $\mathbf{A}\mathbf{x} = \mathbf{b}$ für $(\mathbf{A}_3, \mathbf{b}_3)$, $(\mathbf{A}_4, \mathbf{b}_4)$, $(\mathbf{A}_5, \mathbf{b}_5)$:

$$\mathbf{A}_3 = \begin{pmatrix} 25 & 2 & 3 & 4 & -5 \\ 2 & 10 & 4 & -5 & 6 \\ 3 & 4 & 13 & 6 & -7 \\ 4 & 5 & 6 & -11 & 8 \\ 5 & 6 & 7 & -8 & 14 \end{pmatrix} \mathbf{b}_3 = \begin{pmatrix} 29 \\ 44 \\ 39 \\ 28 \\ 76 \end{pmatrix} \quad (6)$$

$$\mathbf{A}_4 = \begin{pmatrix} 25 & 2 & 3 & 4 & 5 \\ 2 & 10 & 4 & 5 & 6 \\ 3 & 4 & 13 & 6 & 7 \\ 4 & 5 & 6 & 11 & 8 \\ 5 & 6 & 7 & 8 & 14 \end{pmatrix} \mathbf{b}_4 = \begin{pmatrix} 79 \\ 84 \\ 109 \\ 116 \\ 140 \end{pmatrix} \quad (7)$$

$$\mathbf{A}_5 = \begin{pmatrix} 20 & 3 & 2 & 4 & 5 \\ 2 & 5 & 4 & 3 & 6 \\ 3 & 4 & 5 & 6 & 7 \\ 4 & 7 & 6 & 5 & 8 \\ 5 & 7 & 6 & 8 & 9 \end{pmatrix} \mathbf{b}_5 = \begin{pmatrix} 73 \\ 66 \\ 85 \\ 96 \\ 114 \end{pmatrix} \quad (8)$$

mithilfe Ihrer selbst geschriebenen Funktionen für das Jacobi-Verfahren, das Gauss-Seidel-Verfahren ohne und mit Überrelaxation für $\omega = 1., 1.2, \dots, 2.0.$, und vergleichen Sie ihre Lösungsvektoren mit dem Ergebnis von `numpy.linalg.solve`.

Welches der selbstimplementierten Verfahren benötigt die wenigsten Iterationsschritte bzw. konvergiert in der vorgegebenen Anzahl an Iterations-schritten?

Wie verhalten sich die Restfehler gemäß Ihren Abschätzungen?