Prof. Dr. Dirk Lebiedz
Dipl.-Math. oec. Klaus Stolle
Institut für Numerische Mathematik
Universität Ulm

Angewandte Numerik 1
SoSe 2020
15.05.2020

Übungsblatt 04

Besprechung in den Tutorien in der Woche vom 25.05.2020 bis 29.05.2020

Für dieses Übungsblatt gibt es 20 Theorie- und 28 Matlab-Punkte.
Die 70-Prozent-Grenzen liegen aktuell (inklusive Blatt 04) bei 76,3 Theorie- und 36,4 Matlabpunkten.

Aufgabe 12 *(Gauß-Elimination ohne Pivotisierung)* (3T+3T+2T Punkte)

In dieser Aufgabe lösen wir mit Hilfe der LR-Zerlegung das lineare Gleichungssystem $Ax = b$ mit

$$A = \begin{pmatrix} 6 & -4 & 7 \\ -12 & 5 & -12 \\ 18 & 0 & 22 \end{pmatrix} \quad \text{und} \quad b = \begin{pmatrix} \frac{41}{12} \\ -\frac{22}{3} \\ \frac{29}{2} \end{pmatrix} = \left(\tfrac{41}{12},\ -\tfrac{22}{3},\ \tfrac{29}{2}\right)^\top.$$

a) Lösen Sie per Hand das lineare Gleichungssystem $Ax = b$ mit Hilfe der Gaußschen Eliminationsmethode (ohne Pivotisierung). Geben Sie dabei alle Faktoren $\ell_{i,j}$ explizit an.

b) Stellen Sie nun die Frobenius-Matrizen L_1 und L_2 auf und geben Sie deren Inverse L_1^{-1} und L_2^{-1} an. Berechnen Sie damit die Matrizen L und R der LR-Zerlegung von A.

c) Lösen Sie jetzt mit Hilfe der LR-Zerlegung von A das Gleichungssystem $Ax = b$ erneut, und zwar durch Vorwärts- und Rückwärtseinsetzen.

Verwenden Sie bei allen Rechnungen ausschließlich Brüche und keine Dezimalzahlen und geben Sie alle Zwischenschritte an.

Aufgabe 13 *(Programmieraufgabe: LR-Zerlegung ohne Pivotisierung)* (4M+2M+3M+2M Punkte)

In dieser Aufgabe implementieren wir die Gaußsche Eliminationsmethode zur Lösung eines linearen Gleichungssystems $Ax = b$.

a) Schreiben Sie eine Matlab-Funktion `[L, R] = gaussLR(A)`, die die LR-Zerlegung einer quadratischen Matrix `A` mit Hilfe der Gaußschen Eliminationsmethode berechnet. Speichern Sie sukzessive die Faktoren $\ell_{i,j}$ an den Stellen der Matrix A, an denen der Gauß-Algorithmus Nullen erzeugt hat. Falls keine LR-Zerlegung der Matrix `A` existiert, soll Ihre Funktion eine Fehlermeldung ausgeben.

b) Schreiben Sie ein Matlab-Skript `testGaussLR`, mit dem Sie Ihre Matlab-Funktion `[L, R] = gaussLR(A)` an verschiedenen Matrizen `A` testen. Testen Sie auch an der Matrix A aus Aufgabe 12.

c) Schreiben Sie eine Matlab-Funktion `x = solveLR(L, R, b)`, die die Lösung des linearen Gleichungssystems $LRx = b$ durch Vorwärts- und Rückwärtseinsetzen berechnet.

d) Schreiben Sie ein Matlab-Skript `testSolveLR`, mit dem Sie Ihre Matlab-Funktionen `[L, R] = gaussLR(A)` und `x = solveLR(L, R, b)` an verschiedenen linearen Gleichungssystemen $Ax = b$ testen. Überprüfen Sie Ihre Matlabfunktionen auch am linearen Gleichungssystem aus Aufgabe 12.

Aufgabe 14 *(LR-Zerlegung mit Skalierung und Spalten-Pivotisierung)* (6T+3T Punkte)

Gegeben seien

$$A = \begin{pmatrix} 2 & 3 & -1 & 0 \\ -6 & -5 & 0 & 2 \\ 2 & -5 & 6 & -6 \\ 4 & 6 & 2 & -3 \end{pmatrix} \quad \text{und} \quad b = \begin{pmatrix} 20 \\ -33 \\ -43 \\ 49 \end{pmatrix}.$$

a) Bestimmen Sie, falls existent, von Hand die Matrizen D, P, L und R der LR-Zerlegung mit Skalierung und Spaltenpivotisierung der Matrix A. Hierbei bezeichne D die Diagonalmatrix der Skalierung und P die Permutationsmatrix.

b) Lösen Sie mit Hilfe dieser Matrizen das Gleichungssystem $Ax = b$.

Verwenden Sie bei allen Rechnungen ausschließlich Brüche und keine Dezimalzahlen und geben Sie alle Zwischenschritte, insbesondere alle Permutations- und alle Frobenius-Matrizen, an.

Aufgabe 15 *(Programmieraufgabe: LR-Zerlegung mit Skalierung und Spalten-Pivotisierung)* (6M+1M+4M+(3T+2M)+2M+2M Punkte)

a) Erweitern Sie Ihre Matlab-Funktion `[L, R] = gaussLR(A)` aus Aufgabe 13 zu einer Matlab-Funktion `[L, R, P, d] = lrPivot(A)` zur Berechnung der LR-Zerlegung mit Skalierung und Spaltenpivotisierung. Ihre Funktion soll die Dreiecks-Matrizen L und R, die volle Permutationsmatrix P und den Vektor d der Diagonalelemente der Diagoonalmatrix D der Skalierung zurück geben.

b) Falls nötig schreiben Sie auch analog zu Ihrer Matlab-Funktion `x = solveLR(L, R, b)` aus Aufgabe 13 c) eine Matlab-Funktion `x = solveLrPivot(L, R, P, d, b)`, die zusammen mit Ihrer Funktion `[L, R, P, d] = lrPivot(A)` ein Gleichungssystem $Ax = b$ mit Skalierung und Spaltenpivotisierung löst.

c) Schreiben Sie ein Matlab-Skript `testSolve`, das die folgenden Gleichungssysteme $Ax = b$ numerisch jeweils ohne Pivotisierung und mit Skalierung und Spaltenpivotisierung löst.

i)

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 1 \end{pmatrix},$$

ii)

$$A = \begin{pmatrix} 11 & 44 & 1 \\ 0.1 & 0.4 & 3 \\ 0 & 1 & -1 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix},$$

iii)

$$A = \begin{pmatrix} 0.001 & 1 & 1 \\ -1 & 0.004 & 0.004 \\ -1000 & 0.004 & 0.000004 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

d) Vergleichen Sie ihre Ergebnisse hinsichtlich der Genauigkeit. Als Referenzlösung können Sie das Ergebnis von `A\b` betrachten.

Gibt es Unterschiede zwischen den Fällen „Gaußsches Eliminationsverfahren ohne Pivotisierung“ und „Gaußsches Eliminationsverfahren mit Skalierung und Spaltenpivotisierung“? Woran liegt das? Worin unterscheiden sich jeweils die Matrizen L und R in diesen Fällen?

Hinweis: Schreiben Sie sich nötigenfalls eine Matlab-Funktion `zeigeMatrix(A)`, mit der Sie diese Matrizen in einem geeigneten Format ausgeben können.

e) Schreiben Sie ein Matlab-Skript `testA14`, das das Gleichungssystem $Ax = b$ aus Aufgabe 14 numerisch mit Skalierung und Spaltenpivotisierung löst. Stimmen Ihre Ergebnisse mit den von Ihnen in Aufgabe 14 berechneten Ergebnissen überein?

f) Berechnen Sie mit den von Ihnen erstellten Funktionen effizient die Inverse der Matrix A, ohne einschlägige matlabinterne Funktionen zu verwenden. Geben Sie den Aufwand des von Ihnen verwendeten Algoritmus in $\mathcal{O}$-Notation an.

Hinweise:
Die Lösungen der Theorieaufgaben und Ihren Text zu den Programmieraufgaben können Sie in LaTeX erstellen oder handschriftlich aufschreiben und einscannen. Die Programmieraufgaben sind in Matlab zu lösen. Der Source Code muss strukturiert und dokumentiert sein.
Falls Sie die Aufgaben im Team lösen, geben Sie bitte auf allen Lösungen alle an der Aufgabe beteiligten Teammitglieder an. Jedes Teammitglied muss eine Lösung abgeben.
Speichern Sie Ihre Lösungen und Ihre Ergebnisse in einem Directory mit dem Namen **Blatt04_*Vorname*_*Nachname*** und verpacken Sie dieses in eine `.zip`-Datei. Laden Sie **spätestens 48 Stunden vor Ihrem Tutorium** diese `.zip`-Datei in Moodle hoch.