13. Übung zur Algorithmischen Mathematik und Programmieren

Hinweis: Diese Übung dient nur Wiederholung und muss nicht abgegeben werden.

Aufgabe 1:
Untersuchen Sie das nichtlineare Gleichungssystem $f(x_1, x_2) = (0, 0)^T$ mit

$$f(x_1, x_2) = \begin{pmatrix} 8x_1 + \cos(x_1 + 3x_2) \\ \sin(x_1) + \sin(x_2) - 7x_2 \end{pmatrix}.$$

(i) Überführen Sie das Problem in ein Fixpunktproblem und formulieren Sie die Iterationsvorschrift der Fixpunktiteration.

(ii) Zeigen Sie, dass das Fixpunktproblem eine eindeutige Lösung in $\mathbb{R}^2$ hat.

(iii) Berechnen Sie vom Startvektor $(0, 0)$ ausgehend, wieviele Schritte k des Fixpunktverfahrens mindestens nötig sind, um einen maximalen Fehler $\epsilon_{max} = \frac{2^{-20}}{8}$ garantieren zu können.

(iv) Geben Sie die Iterationsvorschrift des Newtonverfahrens für das Problem $f(x_1, x_2) = (0, 0)^T$ an. Explizites Invertieren ist nicht erforderlich!

Aufgabe 2:
Bestimmen Sie zu

$$A = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 10 \end{bmatrix} \quad , \quad b = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

(i) die LR-Zerlegung mit Spaltenpivotsuche.

(ii) die Lösung von $Ax = b$ mit Hilfe der Vorwärts-Rückwärts-Substitution.

Aufgabe 3:

(i) Geben Sie das lokale Newton-Verfahren in Form eines Pseudo-Codes an. Erläutern Sie kurz die einzelnen Schritte.

(ii) Geben Sie die Gauß-Elimination in der *kij*-Variante in Form eines Pseudo-Codes an. Erläutern Sie kurz die einzelnen Schritte.

(iii) Erläutern Sie die Idee der Globalisierung des Newton-Verfahrens mit Hilfe einer Schrittweitensteuerung.

Aufgabe 4:
Sei $A \in \mathbb{R}^{n \times n}$ symmetrisch positiv definit. Leiten Sie die Komplexitäten der LR-Zerlegung mittels Gauß-Elimination und der Cholesky-Zerlegung her. Vergleichen Sie beide.

Aufgabe 5:
Bestimmen Sie handschriftlich die Cholesky-Zerlegung der Matrix A und bestimmen Sie mit deren Hilfe die Lösung des Gleichungssystems $Ax = b$, wobei

$$A = \begin{pmatrix} 9 & 9 & 9 & 9 \\ 9 & 18 & 9 & 18 \\ 9 & 9 & 18 & 9 \\ 9 & 18 & 9 & 27 \end{pmatrix} \quad , \quad b = \begin{pmatrix} 36 \\ 54 \\ 45 \\ 63 \end{pmatrix}.$$

Aufgabe 6:
Gegeben sei der unten stehende Matlab-Code, der bei gegebener oberer Dreiecksmatrix R und gegebenem Vektor b, die Gleichung $Rx = b$ durch Rückwärts-Substitution löst.

```
function x = bw_subst(R, b)
x = zeros(length(b), 1);
for i = n : -1 : 1
    x(i) = b(i);
    for j = i + 1 : n
        x(i) = x(i) - R(i, j) * x(j)
    end
    x(i) = x(i)/R(i, i)
end
```

a) Ersetzen Sie die innere Schleife durch eine adäquate Vektor-Operation.

b) Um in Matlab die Rückwärts-Substitution weiter zu optimieren, sollte man spaltenorientiert arbeiten. Die innerste Schleife sollte somit in einer Spalte von R arbeiten und nicht in einer Zeile. Formulieren Sie eine solche Variante. Hier reicht auch eine Darstellung in Pseudo-Code; es muss kein ausführbarere Matlab-Code generiert werden. Im optimalen Fall geben Sie auch hier eine vektorisierte Variante an.