Albert-Ludwigs-Universität Freiburg
Abteilung für Angewandte Mathematik
Prof. Dr. Patrick Dondl
Dr. Keith Anguige

Wintersemester 2016

Praktikum zu Numerik 1

Blatt 2

(Abgabe: 23. November 2016)

LU-Zerlegung

Aufgabe 5 (4 Punkte). *Vorwärts/Rückwärtssubstitution*

Für eine untere Dreiecksmatrix, L, wird das lineare System $Lx = b$ mit der Vorwärtssubstitutionsformel

$$x_j = \left(b_j - \sum_{k=1}^{j-1} l_{jk} x_k \right) / l_{jj}$$

gelöst. Der folgende MATLAB-Code implementiert diese Formel für festgelegte L und b.

```
% Vorwaertssubstitution:
% Definition einer Matrix L
L = [
      3  0  0  0;
      2  3  0  0;
      1  2  3  0;
     -1 -1  2 10 ];
% und eines Vektors b.
% (es gibt natuerlich noch andere Moeglichkeiten,
% Matrizen und Vektoren in MATLAB zu definieren).
b = [ 1;
      2;
      3;
      4 ];
% Nun berechnen wir x, so dass Lx = b.
n = length(b); % n*n-Matrix, n ist auch die Laenge des Vektors b.

x = zeros(n,1); % wir initialisieren den Vektor x.
for j = 1:n

    sum = 0;  % berechne die Summe in der Formel
    for k=1:j-1
% fuer j=1 ist das in MATLAB automatisch eine 'leere' Summe.
        sum = sum + L(j,k)*x(k);
    end
    x(j) = (b(j) - sum)/L(j,j); % und den Wert fuer x.
end

resid = norm(L*x-b,2); % das Residuum ist die 2-Norm
fprintf('Residuum: %g\n', resid);
```

Schreiben Sie den Code so um, dass er nun Systeme mit *oberer* Dreiecksmatrix behandelt – lösen Sie insbesondere die Gleichung $Ux = b$ für $U = L^T$ (L wie oben) und noch einmal $b = (1,2,3,4)^T$.

Aufgabe 6 (4 Punkte)**.** *LU-Faktorisierung*

Man kann die LU-Faktorisierung einer gegebenen LU-Zerlegbaren Matrix $A = (a_{ij})$ mit folgendem Algorithmus von Crout berechnen:

$$u_{ik} = a_{ik} - \sum_{j=1}^{i-1} l_{ij} u_{jk}, \quad l_{ki} = \left(a_{ki} - \sum_{j=1}^{i-1} l_{kj} u_{ji} \right) / u_{ii}.$$

Implementieren Sie diesen Algorithmus, und so berechnen Sie die LU-Zerlegung von

$$A_1 = \begin{pmatrix} 5 & 3 & 1 \\ 10 & 8 & 8 \\ 15 & 11 & 10 \end{pmatrix}$$

und, für verschiedene sehr kleine Werte von ϵ,

$$A_2 = \begin{pmatrix} \epsilon & 1 \\ 1 & 0 \end{pmatrix}. \tag{1}$$

Aufgabe 7 (4 Punkte)**.** *Tridiagonalmatrizen*

Betrachten Sie die LU-Faktorisierung einer Tridiagonalen Matrix A und vergewissern Sie sich, dass L und U je nur zwei nichttriviale Diagonalen besitzen: d.h. $L_{ij} = 0$ für $i > j+1$ und $U_{ij} = 0$ für $j > i+1$.
Nutzen Sie diese Einsicht aus, um einen viel effizienteren Algorithmus für die LU-Zerlegung in diesem Fall zu entwickeln, und testen Sie Ihre Methode am $(n \times n)$-Beispiel $A_{ii} = 4, A_{i,i+1} = A_{i-1,i} = -1$ für verschiedene n.

Aufgabe 8 (4 Punkte)**.** *Matrixnormen*

Schreiben Sie ein Programm, das die Operatornorm $\| \cdot \|_\infty$ einer Matrix $A \in \mathbb{R}^{m \times n}$ berechnet. Messen Sie (z.B. mit Hilfe des Matlab-Befehls 'cputime') für die Hilbert-Matrix $H \in \mathbb{R}^{n \times n}$ mit Einträgen $h_{ij} = 1/(i+j-1)$, $1 \le i, j \le n$ die Laufzeit des Programms für $n = 10^k$, $k = 1,2,3,4$.

Abgabe der Übungen nach Absprache mit dem Tutor bis zum angegebenen Datum.