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

Wintersemester 2018

Praktikum zu Numerik 1
Blatt 1
Abgabe: 2. November 2018

Heron, Collatz, LU

Aufgabe 1 (4 Punkte). *Heron-Methode*
Die Wurzel $\phi(x) = \sqrt{x}$ einer Zahl $x > 0$ ist gegeben als Grenzwert der Folge $\{z_n\} : z_{n+1} = (z_n + x/z_n)/2$ (für beliebiges $z_0 > 0$). Benutzen Sie diesen Algorithmus um $\sqrt{2}$ zu approximieren.

[Als Abbruchkriterium sollten Sie $|z_{n+1} - z_n| < 10^{-6}$ nehmen.]

Aufgabe 2 (4 Punkte). *Collatz-Vermutung*
Die berühmt/berüchtigte Collatz-Vermutung bezieht sich auf die folgende Konstruktion von Zahlenfolgen:

- Beginne mit einer beliebigen natürlichen Zahl $n > 0$.
- Ist n gerade, so nimm als nächstes $n/2$.
- Ist n ungerade, so nimm als nächstes $3n + 1$.
- Wiederhole die Vorgehensweise mit der erhaltenen Zahl, usw.

Die Vermutung lautet:
Jede so konstruierte Zahlenfolge mündet in den Zyklus $4, 2, 1$*, egal, mit welcher natürlichen Zahl* $n > 0$ *man beginnt.*
Generieren und plotten Sie einige solchen Folgen mit einem Programm; der Algorithmus sollte terminieren wenn der $4, 2, 1$-Zyklus erreicht wird.

Aufgabe 3 (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,

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

Aufgabe 4 (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 bestimmte 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 $b = (1, 2, 3, 4)^T$.

Abgabe: Die Ubungsblätter müssen bis 18:00 Uhr am jeweils angegebenen Abgabedatum abgegeben werden.