**Numerische Mathematik für StudentInnen**
**mit Ziel Diplom in Mathematik**

Gerhard Opfer WS 1997/1998 (1. Teil) SS 1998 (2. Teil)

---

## Bogen 16: Übungen am 5. Mai 1998

### Lösungshinweise

---

**Aufgabe P46.** Das Verfahren von Doolittle liefert dieselbe Zerlegung wie das Gaußsche Eliminationsverfahren, aber in einer anderen Reihenfolge. Gauß rechnet $l_{j1}, r_{jk}, j = 2, 3, \ldots, n, \quad l_{j2}, r_{jk}, j = 3, 4, \ldots, n$, etc. Bei Doolittle werden zuerst die $r'$s in der ersten Zeile, dann die $l'$s in der ersten (Rest)Spalte, etc. ausgerechnet.

**Aufgabe P47.** Man beachte, daß keine expliziten Summen auftreten. Alle Summen sind als Skalarprodukte definiert.

**Aufgabe P48.** Man setze $\mathbf{A}_j \mathbf{u}_j = \mathbf{u}_{j-1}$ mit $\mathbf{u}_0 = \mathbf{b}$ und löse die Gleichungssysteme in der Reihenfolge $j = 1, 2, \ldots, k$. Die Lösung ist dann $\mathbf{x} = \mathbf{u}_k$.

**Aufgabe 31.** Unter den angegebenen Voraussetzungen lautet das Verfahren von Doolittle:

$$\begin{aligned} r_{11} &= a_{11}, \\ l_{j+1j} &= a_{j+1j}/r_{jj}, \\ r_{j+1j+1} &= a_{j+1j+1} - l_{j+1j}a_{jj+1}, \\ r_{jj+1} &= a_{jj+1}, \quad j = 1, 2, \ldots, n-1. \end{aligned}$$

Wegen $|a_{11}| > |a_{21}|$ ist $|a_{11}| = |r_{11}| > 0$. Weiter ist $|l_{21}| = |a_{21}/a_{11}| \le |a_{11}/a_{11}| = 1$. Wir zeigen $r_{jj} \neq 0$ und $|l_{j+1j}| \le 1$ durch vollständige Induktion. Für $j = 1$ haben wir den Nachweis bereits geführt. Dann ist $|r_{j+1j+1}| = |a_{j+1j+1} - l_{j+1j}a_{jj+1}| \ge |a_{j+1j+1}| - |a_{jj+1}| > 0$, und $|l_{j+2j+1}| = |a_{j+2j+1}/r_{j+1j+1}| = |a_{j+2j+1}|/|a_{j+1j+1} - l_{j+1j}a_{jj+1}| \le |a_{j+2j+1}|/(|a_{j+1j+1}| - |a_{jj+1}|) \le 1$. Frage: Läßt sich der Satz ($r_{jj} \neq 0$ und $|l_{j,k}| \le 1$, $j > k$) beim allgemeinen Verfahren von Doolittle auf beliebige, spalten-diagonaldominate Matrizen ausdehnen?

**Aufgabe 32.** Ein matlab-Programm kann so aussehen:

```
%Aufgabe 32, Einzel- (ESV) und Gesamtschrittverfahren (GSV), SS98
%ESV am Schlu x_7  =1.0e-06(-0.0885 -0.2525  0.1466  0.0050  0.0044)'
%GSV am Schlu x_11 =1.0e-05(-0.9095  0.5480  0.5824 -0.2576  0.5424)'
format compact
M=5;
A=[10 -8 0  0  0
    3   7 4  0  0
    0   1 3 -1  0
    0   0 1  4 -1
    0   0 0  8 -9]; %Achtung: A ist zeilen- aber nicht spaltendiagonaldominant!
b=[0 0 0 0 0]';

%Gesamtschrittverfahren. Die neu berechneten Gren
%werden erst nach Ende einer Iteration weiterbenutzt.
%Das erm\"oglicht ein schnelles Programm.
disp('Gesamtschrittverfahren');
x=[1 1 1 1 1]';
y=x;
step=0;
Dh=diag(A);D=diag(Dh);Dh=1./Dh;D1=diag(Dh); %D1=D^-1
while norm(b-A*x,inf)>=10^-5 & step <=50
  y=D1*(b-(A-D)*x); %Faktor 10 schneller als Schleife
  x=y;
  step=step+1;
end; %while
[step,norm(b-A*x,inf)]

%Einzelschrittverfahren. Die neu berechneten Gren
%werden sofort weiterbenutzt. Das Programm ist aber langsam
```

```
disp('Einzelschrittverfahren');
x=[1 1 1 1 1]';
step=0;
while norm(b-A*x,inf)>=10^-5 & step <=50
  for i=1:M
    x(i)=(-A(i,1:i-1)*x(1:i-1)-A(i,i+1:M)*x(i+1:M)+b(i))/A(i,i);
  end; %for i
  step=step+1;%step,x
end; %while
[step,norm(b-A*x,inf)]

%Jetzt SOR-Verfahren. Die neu berechneten Gren
%werden sofort und gewichtet weiterbenutzt.
for omega=1.1:0.01:1.2
disp(['SOR-Verfahren mit omega  = ',num2str(omega)]);
x=[1 1 1 1 1]';
step=0;
while norm(b-A*x,inf)>=10^-5 & step <=50
  for i=1:M
    z(i)=(-A(i,1:i-1)*x(1:i-1)-A(i,i+1:M)*x(i+1:M)+b(i))/A(i,i);
    x(i)=omega*z(i)+(1-omega)*x(i);
  end; %for i
  step=step+1;
end; %while
[step,norm(b-A*x,inf)]
end; %for omega
%Ist beim obigen Beispiel A am besten fr omega=1 (ESV)
```