Musterlösung Aufgabe 24

Aufgabe 24: (LR-Faktorisierung von Bandmatrizen)

Das $n \times n$-Gleichungssystem $\begin{pmatrix} 2 & -1 & & 0 \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ 0 & & -1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ \vdots \\ b_n \end{pmatrix}$ ist zu lösen.

a) Finde die (exakten) Daten der LR-Zerlegung $A = LR$.

Anleitung: die die Bänder von L und R definierende Rekursion ist leicht lösbar.

b) Berechne (exakt) den Lösungsvektor $\vec{x}$ für $\vec{b} = (1, \ldots, 1)^T$.

Anleitung: die Rekursionen der Rücksubstitutionen sind für dieses $\vec{b}$ leicht lösbar.

c) Berechne numerisch die Komponente $x_{10\,000}$ für $\vec{b} = (\sin(1), \sin(2), \ldots, \sin(n))^T$, $n = 500\,000$.

Welcher Speicherplatzbedarf fällt an? Notiere die Laufzeit, Quelltext des Programmes mitabgeben!

Achtung! Das Gleichungssystem ist schlecht konditioniert, relative Rundungsfehler werden um Faktoren der Größenordnung n^2 verstärkt!

Musterlösung: a) Ansatz

$$LR = \begin{pmatrix} 1 & & & \\ l_1 & \ddots & & \\ & \ddots & \ddots & \\ & & l_{n-1} & 1 \end{pmatrix} \begin{pmatrix} r_1 & -1 & & \\ & \ddots & \ddots & \\ & & \ddots & -1 \\ & & & r_n \end{pmatrix}$$

$$= \begin{pmatrix} r_1 & -1 & & & \\ l_1 r_1 & -l_1 + r_2 & -1 & & \\ & l_2 r_2 & -l_2 + r_3 & -1 & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & -1 \\ & & & & l_{n-1} r_{n-1} & -l_{n-1} + r_n \end{pmatrix} .$$

Durch Vergleich mit A folgt

$$r_1 = 2 \ , \quad l_i \, r_i = -1 \ (i = 1, \ldots, n-1) \ , \quad -l_i + r_{i+1} = 2 \ (i = 2, \ldots, n-1) \ ,$$

also $l_i = -1/r_i$ und

$$r_{i+1} = 2 + l_i = 2 - \frac{1}{r_i} \ , \quad r_1 = 2 \ .$$

Diese Rekursion liefert

$$r_1 = 2 \ , \quad r_2 = 2 - \frac{1}{2} = \frac{3}{2} \ , \quad r_3 = 2 - \frac{2}{3} = \frac{4}{3} \ , \quad \ldots \ .$$

In der Tat ist

$$r_i = \frac{i+1}{i} \ , \quad i = 1, \ldots, n \ , \quad l_i = -\frac{1}{r_i} = -\frac{i}{i+1}$$

die Lösung der Rekursion:

$$r_{i+1} = \frac{i+2}{i+1} \stackrel{!}{=} 2 - \frac{1}{r_i} = 2 - \frac{i}{i+1} = \frac{i+2}{i+1} \ .$$

b) Rücksubstitutionen $LR\vec{x} = \vec{b}$. Zunächst $L\vec{y} = \vec{b}$:

$$\begin{pmatrix} 1 & & & \\ l_1 & \ddots & & \\ & \ddots & \ddots & \\ & & l_{n-1} & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ \vdots \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ \vdots \\ b_n \end{pmatrix}$$

$$\Rightarrow \quad y_1 = b_1 \ , \quad y_i = b_i - l_{i-1}\, y_{i-1} = b_i + \frac{i-1}{i}\ y_{i-1} \ (i = 2, \ldots, n) \ . \tag{1}$$

Mit $b_i = 1$ ergeben sich aus

$$y_1 = 1 \ , \quad y_i = 1 + \frac{i-1}{i}\ y_{i-1} \ (i = 2, \ldots, n)$$

die Werte

$$y_1 = 1 \ , \quad y_2 = 1 + \frac{1}{2} = \frac{3}{2} \ , \quad y_3 = 1 + \frac{2}{3}\frac{3}{2} = \frac{4}{2} \ , \quad y_4 = 1 + \frac{3}{4}\frac{4}{2} = \frac{5}{2} \ , \quad \cdots \ .$$

Die offensichtliche Lösung der Rekursion ist $y_i = (i+1)/2$:

$$y_i = \frac{i+1}{2} \stackrel{!}{=} 1 + \frac{i-1}{i}\ y_{i-1} = 1 + \frac{i-1}{i}\ \frac{i}{2} = \frac{i+1}{2} \ .$$

Zweite Rücksubstitution $R\vec{x} = \vec{y}$:

$$\begin{pmatrix} r_1 & -1 & & \\ & \ddots & \ddots & \\ & & \ddots & -1 \\ & & & r_n \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} y_1 \\ \vdots \\ \vdots \\ y_n \end{pmatrix}$$

liefert

$$x_n = \frac{y_n}{r_n} = \frac{n}{n+1}\, y_n \ , \quad x_i = \frac{y_i + x_{i+1}}{r_i} = \frac{i}{i+1}\, (y_i + x_{i+1}) \ (i = n-1, \ldots, 1) \ , \qquad (2)$$

also

$$x_n = \frac{n+1}{2}\, \frac{n}{n+1} = \frac{n}{2} \ , \quad x_i = \left(\frac{i+1}{2} + x_{i+1}\right) \frac{i}{i+1} = \frac{i}{2} + \frac{i}{i+1}\, x_{i+1} \ .$$

Für $z_i := x_i/i$ folgt die einfachere Rekursion

$$z_n = \frac{1}{2} \ , \quad z_i = \frac{1}{2} + z_{i+1} \ (i = n-1, \ldots, 1)$$

mit der offensichtlichen Lösung

$$z_i = \frac{n+1-i}{2} \ .$$

Endergebnis:

$$x_i = i\, z_i = \frac{i\,(n+1-i)}{2} \ .$$

c) Für allgemeines $\vec{b}$ wird die Lösung mittels (1) und (2) durch

$$y_1 := b_1 \ , \quad \mathit{for}\ i = 2..n\ \mathit{do}\ y_i := b_i + \frac{i-1}{i}\, y_{i-1} \ ;$$

$$x_n := \frac{n}{n+1}\, y_n \ , \quad \mathit{for}\ i = n-1..1\ \mathit{do}\ x_i := \frac{i}{i+1}\, (y_i + x_{i+1})$$

geliefert. Speichert man erst $\vec{y}$, dann $\vec{x}$ in $\vec{b}$, so liefert

> **for** $i = 2 \,..\, n$ **do** $b[i] := b[i] + (i-1)/i * b[i-1]$;
>
> $b[n] := n/(n+1) * b[n]$;
>
> **for** $i = n-1 \,..\, 1$ **do** $b[i] := i/(i+1) * (b[i] + b[i+1])$;

die Lösung im Vektor $\vec{b}$. Der Speicherplatzbedarf besteht aus einem Vektor $\vec{b}$. Bei 64-bit(=8-Byte)-floats (16 Dezimalstellen) mit $n = 500\,000$ sind dies 4 MByte.

MuPAD-Lösung mit verschiedenen Werten von n und verschiedenen Maschinengenauigkeiten:

```
   *----*    MuPAD 1.4.0 --- Multi Processing Algebra Data Tool
  /|   /|
 *----* |    Copyright (c) 1992-97 by B. Fuchssteiner, Automath
 | *--|-*    University of Paderborn. All rights reserved.
 |/   |/
 *----*      ----------- Developers NSB Version -----25.11.97--
```

```
>> for n in [5000,50000,500000] do
&>  for digits in [7,16,32] do
&>    DIGITS:=digits;
&>    st:=time():
&>    b:=array(1..n,[ float(sin(i)) $ hold(i)=1..n] ):
&>    for i from 2 to n do b[i]:=b[i]+(i-1)/i*b[i-1]: end_for:
&>    b[n]:=n/(n+1)*b[n];
&>    for i from n-1 downto 1 do b[i]:=i/(i+1)*(b[i]+b[i+1]); end_for:
&>    DIGITS:=10;
&>    print("n=".n,"DIGITS=".digits,"Ergebnis", b[n/50],"Zeit", (time()-st)*sec/1000.0);
&>  end_for;
&> end_for:

 "n=5000",   "DIGITS= 7", "Ergebnis", -0.5418536113, "Zeit",    7.56sec
 "n=5000",   "DIGITS=16", "Ergebnis", -0.5419803388, "Zeit",    8.29sec
 "n=5000",   "DIGITS=32", "Ergebnis", -0.5419803388, "Zeit",   10.54sec

 "n=50000",  "DIGITS= 7", "Ergebnis",  0.8984039626, "Zeit",   75.37sec
 "n=50000",  "DIGITS=16", "Ergebnis",  0.9114517129, "Zeit",   84.07sec
 "n=50000",  "DIGITS=32", "Ergebnis",  0.9114517129, "Zeit",  106.51sec

 "n=500000", "DIGITS= 7", "Ergebnis",  0.08038886289,"Zeit",  756.83sec
 "n=500000", "DIGITS=16", "Ergebnis", -0.3164850488, "Zeit",  829.65sec
 "n=500000", "DIGITS=32", "Ergebnis", -0.3164850491, "Zeit", 1051.66sec
```

C-Lösung mit double-Variablen (16-Stellen-Arithmetik):

```
#include <stdio.h>
#include <math.h>
#define  n 500000
main()
{ double b[n+1];
  int    i;
  for (i=1; i<=n; i++)   b[i]= sin( (double) i);
  for (i=2; i<=n; i++)   b[i]= b[i]+((i-1)*b[i-1])/i;
  b[n] = (n*b[n])/(n+1);
  for (i=n-1; i>=1; i--) b[i]= (i*(b[i]+b[i+1]))/(i+1);
  printf("%+.10f\n",b[10000]);
}

Ergebnis: -0.3164850492     Zeit:  1.46u 0.06s 0:01.55 98.0%
```