

Lösungen zur Vorlesung
„Numerische Mathematik für Studierende der Wirtschaftsmathematik, der
Lehrämter und der Naturwissenschaften“

Blatt 1

Abgabetermin: 11.04.2006

Aufgabe 1 (2 Punkte)

Zur Berechnung von F^{-1} betrachten wir n lineare Gleichungssysteme der Form $Fx^k = e^k$, wobei e^k der k -te Einheitsvektor ist, $k = 1, \dots, n$.

Die Gleichungssysteme sind eindeutig lösbar, da F regulär ist ($\det(F) = 1$).

1. Sei $k < j$: Aus der i -ten Gleichung mit $i \leq j$ und $i \neq k$ folgt $x_i^k = 0$ und für $i = k$ folgt $x_k^k = 1$.

Damit gilt insbesondere $x_j^k = 0$ womit unmittelbar $x_i^k = 0$ für $i > j$ folgt. Also gilt $x^k = e^k$.

2. Sei $k = j$. Aus der i -ten Gleichung mit $i < j$ folgt $x_i^k = 0$ und für $i = k = j$ folgt $x_k^k = 1$.

Für $i > j$ hat die i -te Gleichung die Form $f_{ij}x_j^k + x_i^k = 0$, woraus sich wegen $x_k^k = 1$ unmittelbar $x_i^k = -f_{ij}$ ergibt.

3. Sei $k > j$: Aus der i -ten Gleichung mit $i \leq j$ folgt $x_i^k = 0$.

Damit gilt insbesondere $x_j^k = 0$ womit unmittelbar $x_i^k = 0$ für $i > j$, $i \neq k$ folgt sowie $x_k^k = 1$ für $i = k$. Also gilt wieder $x^k = e^k$.

Insgesamt ergibt sich:

$$F^{-1} = \begin{pmatrix} 1 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & 1 & \ddots & & & & & \vdots \\ \vdots & \ddots & \ddots & \ddots & & & & \vdots \\ \vdots & & 0 & 1 & \ddots & & & \vdots \\ \vdots & & \vdots & -f_{j+1,j} & 1 & \ddots & & \vdots \\ \vdots & & \vdots & \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & & \vdots & \vdots & \vdots & \ddots & 1 & 0 \\ 0 & \cdots & 0 & -f_{n,j} & 0 & \cdots & 0 & 1 \end{pmatrix}.$$

Aufgabe 2 (4 Punkte (2+2))

Betrachten wir einen Schritt des Gauß Algorithmus' und vertauschen Zeile 1 mit Zeile $m+1$. Die neue erste Zeile ist erste Zeile von R , so dass R mindestens die Bandbreite $2m$ besitzt.

Elimination mit der neuen ersten Zeile erzeugt in den Untermatrizen höchstens Bandbreite $2m$. Die erste Spalte von L enthält höchstens $m + 1$ Nichtnullelemente, weil die Faktoren l_{j1} höchstens für $j = 1, \dots, m$ von Null verschieden sind. Die Argumentation in den nachfolgenden Eliminationsschritten verläuft vollkommen analog.

$$PA = \begin{pmatrix} a_{m+1,1} & \cdots & a_{m+1,2m+1} & 0 & \cdots \\ \vdots & & & & \\ a_{11} & \cdots & & & \\ \vdots & & & & \\ 0 & & & & \\ & & \ddots & & \end{pmatrix}.$$

c) Induktionsschritt von n nach $n + 1$:

$$LL^H = \begin{pmatrix} L_n & 0 \\ c^H & \alpha \end{pmatrix} \begin{pmatrix} L_n^H & c \\ 0 & \alpha \end{pmatrix} = \begin{pmatrix} A_n & b \\ b^H & a_{n+1,n+1} \end{pmatrix} = A$$

mit $b \in \mathbb{C}^n$ und A, A_n symmetrisch und pos.def. Daraus folgt u.A.

$$L_n c = b.$$

Da in b nur die letzten $m - 1$ Elemente nicht-0 sind und L untere Dreiecksmatrix ist mit einer Bandbreite $\leq m$ (laut Induktionsannahme), können in c auch nur die letzten $m - 1$ Elemente nicht-0 sein.

Aufgabe 3 (4 Punkte (2+2))

(a) Wegen $A = LL^T$ gilt $a_{i,i} = \sum_{s=1}^i l_{i,s}^2$, woraus folgt $a_{i,i} \geq l_{i,s}^2, s = 1, \dots, i$, d.h. $|l_{i,s}| \leq \sqrt{a_{i,i}}$.

(b) Algorithmus der LR Zerlegung mit partieller Pivotisierung

Do $j = 1, n$

$i_p = \text{pivot}(a_{jj}, \dots, a_{nj}) = \arg\max_{j \leq l \leq n} |a_{lj}|$

if ($i_p == 0$) Stop, A singular

vertausche Zeilen i_p und j in A und b_{i_p} und b_j

Do $i = j + 1, n$

$l_{ij} = a_{ij} / a_{jj}$

$b_i = b_i - l_{ij} * b_j$

do $k = j, n$

$a_{ik} = a_{ik} - l_{ij} * a_{jk}$

enddo

enddo

enddo

Sei $M_m = \max |a_{ij}^m|$ das betragsgrößte Element nach dem m-ten Aufdatierungsschritt (Sepziell $M = M_0 = \max |a_{ij}|$ das der Originalmatrix) R wird auf dem Platz von A

gespeichert, am Ende steht in A also R . Wegen der Pivotwahl und Vertauschung ist $|a_{jj}^m| \geq |a_{ij}^m|$ $i > j$ also $|l_{ij}| \leq 1$ damit gilt in innerste Schleife $|a_{ik}^{m+1}| \leq |a_{ik}^m| + 1M_m$ d.h. in jedem Schritt wird der Betrag von a_{ik} um höchstens M_m größer. Da $|a_{ik}^m| < M_m$ gilt, ist $|a_{ik}^{m+1}| \leq 2M_m \forall i, k$ also auch $M_{m+1} = \max |a_{ik}^{m+1}| \leq 2M_m$. In den beiden innersten Schleifen wird ein a_{ik} nur (höchstens) einmal aufdatiert. Die i Schleife wird nur für $j = 1, n-1$ ausgeführt als $n-1$ mal. Damit gilt $|r_{ik}| \leq 2^{n-1}M$. Die Schranke ist scharf, siehe Kielbasinski & Schwetlick, Numerische lineare Algebra, Übungsaufgabe 5.2.2.

Aufgabe 4 (6 Punkte)

Matlab-Funktion lr.m

```
function [L,R] = lr(A)

n = size(A,1);

eps = 1e-14;

L = eye(n);

for k = 1:n-1

    for i = k+1:n

        if abs(A(k,k)) < eps*abs(A(i,k))

            fprintf(1,'Warnung: Sehr kleines Pivotelement; Pivotsuche waere sinnvoll'\n)

        end

        L(i,k) = A(i,k)/A(k,k);

        for j = k+1:n

            A(i,j) = A(i,j)-L(i,k)*A(k,j);

        end

        A(i,k) = 0;

    end

end
```

Matlab-Funktion lrp.m

```
function [L,R,p] = lrp(A)

amax = max(max(abs(A)));

n = size(A,1);

eps = 1e-14;

p = [1:n]';

L = eye(n);

for k = 1:n-1

    absakk = abs(A(p(k),k));

    s = k;

    for i = k+1:n

        if (abs(A(p(i),k))>absakk)

            s = i;

            absakk = abs(A(p(i),k));

        end

    end

    if absakk < eps*amax

        fprintf(1,'Warnung: Matrix fast singulaer'\n)

    end

    ps = p(s);

    p(s) = p(k);
```

```

p(k) = ps;

akk = A(p(k),k);

for i = k+1:n

    lik = A(i,k)/akk;

    for j = k+1:n

        A(p(i),j) = A(p(i),j)-lik*A(p(k),j);

    end

    A(p(i),k) = lik;

end

end

R = triu(A(p),:);

L = eye(n) + tril(A(p,:),-1);

```

Matlab-Funktion solve.m

```

function x = solve(L,R,p,b)

n = size(b,1);

y = zeros(n,1);

x = zeros(n,1);

% berechne y = L-1*b(p);

y(1) = b(p(1))/L(1,1); % eigentlich L(k,k) = 1

for k = 2:n

    y(k) = (b(p(k))-L(k,1:k-1)*y(1:k-1))/L(k,k);

end

```

Aufgabe	(i)	(ii)	(iii)
x aus lr.m		$(0.0608, 0, 0.3313)^T$	$1e5 * (-0.0000, -2.4900, 2.4900)^T$
x aus lrp.m	$(1, 1)^T$	$(-5.2644, 1.3313, 0.3313)^T$	$1e5 * (-0.0000, -2.4900, 2.4900)^T$
Defekt $Ax - b$ aus lr.m		$(0, 0, -1.3313)^T$	$1e - 4 * (0, -0.0002, -0.3356)^T$
Defekt $Ax - b$ aus lrp.m	$(0, 0)^T$	$1e - 15 * (0.6613, 0, 0)^T$	$1e - 10 * (0.5821, 0, -0.0004)^T$

Tabelle 1: Zu Aufgabe 4. Gleichungssysteme (i) – (iii) mit **lr.m** bzw. **lrp.m**.

```
% berechne  $x = R^{-1} * y$ 

x(n) = y(n)/R(n,n);

for k = n-1:-1:1

    x(k) = (y(k)-R(k,k+1:n)*x(k+1:n))/R(k,k);

end
```

Im ersten Fall (i) ist LR -Zerlegung nicht durchführbar. Matlab gibt die Warnung 'Warning: divide by zero' aus.

Im zweiten und dritten Fall liefert sie ungenaue Lösungen.

In allen drei Fällen liefert LR -Zerlegung mit Spaltenpivotsuche genaue Lösungen der Gleichungssysteme.

Die Lösungen in (iii) von **lr.m** bzw. **lrp.m** unterscheiden sich frühestens in der 8. Nachkommastelle

($\sim 10^{-3}$). Die Unterschiede im Defekt sind aber gravierend.