Technische Universität München
Institut für Informatik
Prof. Dr. Hans-Joachim Bungartz
Dipl.-Inf. Christoph Riesinger
Dipl.-Math. Jürgen Bräckle

WS 2013/2014

Numerisches Programmieren, Übungen

Musterlösung 8. Übungsblatt: Matrixkondition, LR-Zerlegung, Pivotsuche

1) Kondition einer Matrix

a) Bestimmung der Inversen A^{-1} unter Verwendung der Gauß-Jordan-Elimination:

$$\left(\begin{array}{cc|cc} 1 & a & 1 & 0 \\ -1 & 1 & 0 & 1 \end{array}\right) \rightsquigarrow \left(\begin{array}{cc|cc} 1 & a & 1 & 0 \\ 0 & 1+a & 1 & 1 \end{array}\right) \rightsquigarrow$$
$$\left(\begin{array}{cc|cc} 1 & a & 1 & 0 \\ 0 & 1 & \frac{1}{1+a} & \frac{1}{1+a} \end{array}\right) \rightsquigarrow \left(\begin{array}{cc|cc} 1 & 0 & 1-\frac{a}{1+a} & \frac{-a}{1+a} \\ 0 & 1 & \frac{1}{1+a} & \frac{1}{1+a} \end{array}\right) \rightsquigarrow \left(\begin{array}{cc|cc} 1 & 0 & \frac{1}{1+a} & \frac{-a}{1+a} \\ 0 & 1 & \frac{1}{1+a} & \frac{1}{1+a} \end{array}\right)$$

Damit erhalten wir:

$$A^{-1} = \frac{1}{1+a}\begin{pmatrix} 1 & -a \\ 1 & 1 \end{pmatrix}$$

b) Ermittlung der Konditionszahl κ

Es lässt sich zeigen, dass die Zeilensummennorm einer Matrix M äquivalent zur Maximumsnorm einer Matrix aus der Vorlesung ist. Die Zeilensummennorm ist wie folgt definiert:

$$||M||_\infty = \max_i \sum_{j=1}^{n} |m_{ij}|$$

Daraus folgt in unserem Fall, daß

$$\kappa = ||A^{-1}|| \cdot ||A|| = \max\left(\frac{2}{|1+a|}, \frac{1}{|1+a|} + \left|\frac{a}{1+a}\right|\right) \cdot \max(2, 1+|a|).$$

Da $a > 2$:

$$\kappa = \left(\frac{1}{|1+a|} + \left|\frac{a}{1+a}\right|\right) \cdot (1+|a|) = 1+a$$

2) LR-Zerlegung

Zunächst folgt eine kleine Einführung zur LR-Zerlegung. Eine mögliche Umsetzung der drei Abschnitte in Pseudocode sieht folgendermaßen aus:

```
for i=1:n % Fuer jede Zeile i
    for k=1:i-1 % Berechne Elemente L[i,k]
        L[i,k] := A[i,k];
        for j=1:k-1
            L[i,k] := L[i,k]-L[i,j]*R[j,k];
        end
        L[i,k] := L[i,k]/R[k,k];
    end

    for k=i:n % Berechne Elemente R[i,k]
        R[i,k] := A[i,k];
        for j=1:i-1
            R[i,k] := R[i,k]-L[i,j]*R[j,k];
        end
    end
end
```

2. Vorwärtssubstitution: $Ly = b$

```
for i=1:n
    y[i] := b[i];
    for j=1:i-1
        y[i] := y[i]-L[i,j]*y[j];
    end
end
```

3. Rückwärtssubstitution: $Rx = y$

```
for i=n:-1:1
    x[i] := y[i];
    for j=i+1:n
        x[i] := x[i]-R[i,j]*x[j];
    end
    x[i] := x[i]/R[i,i];
end
```

Die Grundidee ist für die Algorithmen Gaußelimination und LR-Zerlegung natürlich die gleiche: Bringe A durch Zeilenumformungen auf Zeilenstufenform. Der Gauß-Algorithmus eliminiert sukzessive Einträge in einer Spalte i (äußere Schleife), geht also spaltenweise vor. Dagegen berechnet unser LR-Algorithmus für jedes i der äußeren Schleife die Einträge von L und R in der Zeile i; er arbeitet also zeilenweise. Die zweite Schleife über k läuft dementsprechend über die nötigen Spaltenindizes von L und R.

Um zu verstehen, warum der LR-Algorithmus die angegebene Form hat, betrachten wir die Zerlegung genauer. $A = L \cdot R$ bedeutet in Indexnotation

$$A_{ik} = \sum_{j=1}^{n} L_{ij} \cdot R_{jk}, \qquad i,k = 1,\ldots,n$$

wobei n die Dimension der Matrix A ist. Die Einträge A_{ik} können in solche unterhalb bzw. oberhalb der Diagonalen aufgesplittet und getrennt betrachtet werden.

- $i > k$: Einträge unterhalb der Diagonalen: "Zeile mal Spalte" im Produkt $L \cdot R$ ergibt aufgrund der Nullen eine Summe, die nur bis k läuft (kleinerer der beiden Indizes):

$$\begin{aligned} A_{ik} &= \sum_{j=1}^{n} L_{ij} \cdot R_{jk} = \sum_{j=1}^{k} L_{ij} \cdot R_{jk} = \sum_{j=1}^{k-1} L_{ij} \cdot R_{jk} + L_{ik} \cdot R_{kk} \\ \Rightarrow \quad L_{ik} &= \left(A_{ik} - \sum_{j=1}^{k-1} L_{ij} \cdot R_{jk} \right) / R_{kk}. \end{aligned}$$

- $i \leq k$: Einträge oberhalb der Diagonalen: Die Summe läuft nur bis i (kleinerer der beiden Indizes):

$$\begin{aligned} A_{ik} &= \sum_{j=1}^{n} L_{ij} \cdot R_{jk} = \sum_{j=1}^{i} L_{ij} \cdot R_{jk} = \sum_{j=1}^{i-1} L_{ij} \cdot R_{jk} + L_{ii} \cdot R_{ik} \\ \Rightarrow \quad R_{ik} &= \left(A_{ik} - \sum_{j=1}^{i-1} L_{ij} \cdot R_{jk} \right) / \underbrace{L_{ii}}_{=1}. \end{aligned}$$

Damit erhalten wir exakt die Formeln der LR-Zerlegung aus dem angegebenen Algorithmus. Man macht sich leicht klar, dass aufgrund des zeilenweisen Durchlaufs der äußeren Schleife (über i) stets alle nötigen Werte schon vorhanden sind, um die neuen Einträge L_{ik} und R_{ik} zu berechnen.

a) Lösungsschritte bei Anwendung von Gauß-Elimination auf das Problem $Ax = b$

1. $\left(\begin{array}{ccc|c} 4 & 2 & 3 & 5 \\ 2 & 2 & 1 & -3 \\ 2 & 2 & 2 & 0 \end{array}\right)$

2. $\left(\begin{array}{ccc|c} 4 & 2 & 3 & 5 \\ 0 & 1 & -1/2 & -11/2 \\ 0 & 1 & 1/2 & -5/2 \end{array}\right)$

3. $\left(\begin{array}{ccc|c} 4 & 2 & 3 & 5 \\ 0 & 1 & -1/2 & -11/2 \\ 0 & 0 & 1 & 3 \end{array}\right)$

4. $\left(\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & -4 \\ 0 & 0 & 1 & 3 \end{array}\right)$

b) Berechnung der Zerlegung (1.) von $A = \begin{pmatrix} 4 & 2 & 3 \\ 2 & 2 & 1 \\ 2 & 2 & 2 \end{pmatrix}$: Die Matrizen L und R denken wir uns mit lauter Nulleinträgen vorbelegt. Neu hinzukommende Einträge werden mit einer Box gekennzeichnet.

- i=1:

$$
\begin{aligned}
L: \quad & for\ k = 1:0 \ \Rightarrow \text{nichts zu tun (empty)} \\
R: \quad & for\ k = 1:3 \\
& \qquad R[1,k] = A[1,k]; \\
& for\ j = 1:0 \ \Rightarrow \text{empty} \quad \Rightarrow\ R = \begin{pmatrix} \boxed{4} & \boxed{2} & \boxed{3} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}
\end{aligned}
$$

- i=2:

$$
\begin{aligned}
L: \quad & for\ k = 1:1 \\
& \qquad L[2,1] = A[2,1] = 2; \\
& \qquad for\ j = 1:0 \ \Rightarrow \text{empty} \\
& \qquad L[2,1] = L[2,1]/R[1,1] = 2/4; \quad \Rightarrow\ L = \begin{pmatrix} 0 & 0 & 0 \\ \boxed{1/2} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\
R: \quad & for\ k = 2:3 \\
& \qquad R[2,k] = A[2,k]; \\
& \qquad for\ j = 1:1 \\
& \qquad\qquad R[2,k] = R[2,k] - L[2,1] \cdot R[1,k]; \\
& \qquad\qquad \rightarrow k = 2: \ R[2,2] = R[2,2] - L[2,1] \cdot R[1,2] = 2 - 1/2 \cdot 2 = 1; \\
& \qquad\qquad \rightarrow k = 3: \ R[2,3] = R[2,3] - L[2,1] \cdot R[1,3] = 1 - 1/2 \cdot 3 = -1/2; \\
& \Rightarrow\ R = \begin{pmatrix} 4 & 2 & 3 \\ 0 & \boxed{1} & \boxed{\text{-1/2}} \\ 0 & 0 & 0 \end{pmatrix}
\end{aligned}
$$

- i=3:

$$
\begin{aligned}
L: \quad & for\ k = 1:2 \\
& \quad L[3,k] = A[3,k]; \\
& \quad for\ j = 1:k-1 \\
& \quad\quad \rightarrow k=1: \text{empty} \\
& \quad\quad \rightarrow k=2: L[3,2] = L[3,2] - L[3,1] * R[1,2] = 1; \\
& \quad L[3,k] = L[3,k]/R[k,k]; \quad \Rightarrow\ L = \begin{pmatrix} 0 & 0 & 0 \\ 1/2 & 0 & 0 \\ \boxed{1/2} & \boxed{1} & 0 \end{pmatrix} \\
R: \quad & for\ k = 3:3 \\
& \quad R[3,3] = A[3,3]; \\
& \quad for\ j = 1:2 \\
& \quad\quad R[3,3] = R[3,3] - L[3,j] \cdot R[j,3]; \\
& \quad\quad \rightarrow j=1: R[3,3] = 2 - 1/2 \cdot 3 = 1/2; \\
& \quad\quad \rightarrow j=2: R[3,3] = 1/2 - 1 \cdot (-1/2); \\
& \Rightarrow\ R = \begin{pmatrix} 4 & 2 & 3 \\ 0 & 1 & -1/2 \\ 0 & 0 & \boxed{1} \end{pmatrix}
\end{aligned}
$$

Damit haben wir unsere gesuchte Zerlegung:

$$
A \ = \ L \cdot R \ = \ \begin{pmatrix} 1 & 0 & 0 \\ 1/2 & 1 & 0 \\ 1/2 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 4 & 2 & 3 \\ 0 & 1 & -1/2 \\ 0 & 0 & 1 \end{pmatrix}.
$$

c) Substitution:

- Vorwärtssubstitution $Ly = b$:

$$
\begin{pmatrix} 1 & 0 & 0 \\ 1/2 & 1 & 0 \\ 1/2 & 1 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} \ = \ \begin{pmatrix} 5 \\ -3 \\ 0 \end{pmatrix}
$$

$$
\Rightarrow \ y = (5,\ -11/2,\ 3)^T.
$$

- Rückwärtssubstitution $Rx = y$:

$$
\begin{pmatrix} 4 & 2 & 3 \\ 0 & 1 & -1/2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \ = \ \begin{pmatrix} 5 \\ -11/2 \\ 3 \end{pmatrix}
$$

$$
\Rightarrow \ x = (1,\ -4,\ 3)^T.
$$

d) Die Zerlegung ist bereits bekannt, daher müssen zur Lösung von $Ax = c$ lediglich Vorwärts- und Rückwärtssubstitution durchgeführt werden. Der Aufwand hierfür ist

$O(n^2)$ Operationen (vergleiche $O(n^3)$ Operationen für die Berechnung der Zerlegung bei der Lösung von $Ax = b$).

Substitution:

- Vorwärtssubstitution $L\tilde{y} = c$:

$$\begin{pmatrix} 1 & 0 & 0 \\ 1/2 & 1 & 0 \\ 1/2 & 1 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \\ 2 \end{pmatrix}$$
$$\Rightarrow \quad \tilde{y} = (2,\ 0,\ 1)^T.$$

- Rückwärtssubstitution $Rx = \tilde{y}$:

$$\begin{pmatrix} 4 & 2 & 3 \\ 0 & 1 & -1/2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \\ 1 \end{pmatrix}$$
$$\Rightarrow \quad x = (-1/2,\ 1/2,\ 1)^T.$$

3) Gauß-Elimination und Pivotsuche

a) Gauß-Elimination ohne Pivotsuche und mit exakter Arithmetik:

$$\left(\begin{array}{cc|c} -\frac{1}{1000} & 1 & 1 \\ 2 & 1 & 0 \end{array}\right) \Rightarrow \left(\begin{array}{cc|c} -\frac{1}{1000} & 1 & 1 \\ 0 & 2001 & 2000 \end{array}\right)$$
$$\Rightarrow \quad x_2 = \frac{2000}{2001} = 0.9995\ldots, \qquad x_1 = (x_2 - 1) \cdot 1000 = -0.4997\ldots$$
$$\Rightarrow \quad x = \begin{pmatrix} -0.4997\ldots \\ 0.9995\ldots \end{pmatrix}.$$

b) Gauß-Elimination ohne Pivotsuche und mit Rundungsfehlern (3 Stellen):

$$\left(\begin{array}{cc|c} -\frac{1}{1000} & 1 & 1 \\ 2 & 1 & 0 \end{array}\right) \Rightarrow \left(\begin{array}{cc|c} -1.000 \cdot 10^{-3} & 1 & 1 \\ 0 & 2.001 \cdot 10^3 & 2.000 \cdot 10^3 \end{array}\right)$$
$$\overset{\text{Runden}}{\Rightarrow} \left(\begin{array}{cc|c} -1.00 \cdot 10^{-3} & 1 & 1 \\ 0 & 2.00 \cdot 10^3 & 2.00 \cdot 10^3 \end{array}\right)$$
$$\Rightarrow \quad x_2 = \frac{2.00}{2.00} = 1, \qquad x_1 = (x_2 - 1) \cdot 10^3 = 0$$
$$\Rightarrow \quad x = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Der zweite Eintrag x_2 ist in Ordnung, aber x_1 ist komplett falsch!

c) Gauß-Elimination mit Pivotsuche und mit Rundungsfehlern (3 Stellen):

$$\begin{pmatrix} -\frac{1}{1000} & 1 & \vert & 1 \\ 2 & 1 & \vert & 0 \end{pmatrix} \quad \Rightarrow \quad \begin{pmatrix} 2 & 1 & \vert & 0 \\ -\frac{1}{1000} & 1 & \vert & 1 \end{pmatrix} \Rightarrow \begin{pmatrix} 2 & 1 & \vert & 0 \\ 0 & \frac{2001}{2000} & \vert & 1 \end{pmatrix}$$

$$\overset{\text{Runden}}{\Rightarrow} \quad \begin{pmatrix} 2 & 1 & \vert & 0 \\ 0 & 1 & \vert & 1 \end{pmatrix}$$

$$\Rightarrow \quad x_2 = 1, \quad x_1 = -x_2/2 = -0.5$$

$$\Rightarrow \quad x = \begin{pmatrix} -0.5 \\ 1 \end{pmatrix}.$$

Diesmal ist x für beide Einträge in der passenden Größenordnung (vgl. a)).