Technische Universität München
Institut für Informatik
Prof. Dr. Hans-Joachim Bungartz
Michael Lieb, M. Sc.
Dipl.-Inf. Gerrit Buse

WS 2009/2010

Numerisches Programmieren

3. Programmieraufgabe: Direkte Lösung linearer Gleichungssysteme

Direkte Lösung linearer Gleichungssysteme

Gauß-Elimination

Ziel dieser Programmieraufgabe ist die Implementierung von Verfahren zur Lösung von Systemen linearer Gleichungen. Man will also zu gegebenem

$$\begin{aligned} A &= (a_{i,j})_{0\le i,j<n} \in \mathbb{R}^{n,n} \\ b &= (b_i)_{0\le i<n} \in \mathbb{R}^n \end{aligned}$$

ein $x \in \mathbb{R}^n$ finden mit

$$A \cdot x = b.$$

Zunächst soll das Gauß-Eliminationsverfahren mit Spaltenpivotsuche implementiert werden. Im ersten Schritt des Gauß-Eliminationsverfahrens müssen alle Elemente der ersten Spalte unterhalb der Hauptdiagonalen zu Null werden. Dies wird dadurch erreicht, dass von jeder Zeile i die erste Zeile multipliziert mit $\frac{a_{i,0}}{a_{0,0}}$ abgezogen wird. Die Pivotsuche setzt noch vor diesem Schritt an und sorgt dafür, dass kein Element unter dem Hauptdiagonalenelement $a_{0,0}$ betragsmäßig größer ist als $a_{0,0}$. Dazu wird $a_{0,0}$ mit sämtlichen Elementen darunter verglichen. Falls es ein betragsmäßig Größeres gibt werden die beiden zugehörigen Zeilen vertauscht. Die Pivotsuche wird in jeder Spalte durchgeführt bevor die Einträge unter dem Hauptdiagonalenelement eliminiert werden. Im folgenden Beispiel ist die erste Spalte schon fertig bearbeitet. D. h., es muss nun auf der zweiten Spalte die Pivotsuche durchgeführt werden. Von den zu untersuchenden Matrixelementen ist das in der dritten Zeile das betragsmäßig Größte, die zweite Zeile wird daher mit der dritten Zeile vertauscht:

$$\left(\begin{array}{cccc|c} 1 & 4 & 8 & 3 & 7 \\ 0 & 2 & 2 & 4 & 0 \\ 0 & -3 & -7 & 2 & 1 \\ 0 & 1 & 5 & 2 & 2 \end{array}\right) \Rightarrow \left(\begin{array}{cccc|c} 1 & 4 & 8 & 3 & 7 \\ 0 & -3 & -7 & 2 & 1 \\ 0 & 2 & 2 & 4 & 0 \\ 0 & 1 & 5 & 2 & 2 \end{array}\right)$$

Bandmatrizen

Für ein Gleichungssystem mit n Gleichungen benötigt die Gauß-Elimination einen Rechenaufwand von $O(n^3)$ Operationen. Für Bandmatrizen lässt sich der Rechenaufwand aber erheblich reduzieren. Bandmatrizen sind Matrizen, bei denen nur auf der Hauptdiagonalen und einigen Nebendiagonalen Elemente ungleich Null sind:

$$\begin{pmatrix}
a_{0,0} & \cdots & a_{1,p_A-1} & & & \\
\vdots & \ddots & \ddots & \ddots & & \\
a_{q_A-1,1} & \ddots & \ddots & \ddots & \ddots & \\
 & \ddots & \ddots & \ddots & \ddots & a_{n-p_A,n-1} \\
 & & \ddots & \ddots & \ddots & \vdots \\
 & & & a_{n-1,n-q_A} & \cdots & a_{n-1,n-1}
\end{pmatrix}$$

Die untere Bandbreite q_A ist die Anzahl der Nebendiagonalen unterhalb der Hauptdiagonalen inklusive der Hauptdiagonalen. Die obere Bandbreite p_A ist analog definiert. Die Gesamtbandbreite des Bandes ist somit $p_A + q_A - 1$. Ein Gleichungssystem einer solchen Struktur lässt sich mit $O(n \cdot p_A \cdot q_A)$ Operationen lösen.

LR-Zerlegung (= LU-Zerlegung)

Ein Nachteil des Gauß-Eliminationsverfahrens ist, dass man das komplette Gleichungssystem erneut lösen muss, selbst wenn sich nur der Vektor b ändert, die Matrix A aber gleich bleibt. Hier bietet die LR-Zerlegung einen Vorteil. Zunächst wird die Matrix A in die beiden Matrizen L und R zerlegt. Dazu wird nur die Matrix A verwendet, b muss noch nicht festgelegt sein. Danach kann man für verschiedene b durch Anwenden der Vorwärts- und Rückwärts-Substitution die Lösung berechnen ohne die Matrix A erneut zerlegen zu müssen.

In der Vorlesung wurde die LR-Zerlegung vollbesetzter Matrizen vorgestellt. Im Rahmen dieser Programmieraufgabe soll die LR-Zerlegung von Bandmatrizen implementiert werden. Dazu soll die Klasse *BandMatrix* verwendet werden. Diese benötigt zur Speicherung der Bandmatrix nur $O(n \cdot (p + q - 1))$ Speicherplatz, auf die Matrixelemente kann aber über die übliche Matrixindizierung zugegriffen werden. Es darf aber natürlich nur auf solche Matrixelemente zugegriffen werden, die auch wirklich im Bereich des Bandes liegen.

Wird die LR-Zerlegung auf einer solchen Matrix durchgeführt, so sind die entstehenden Matrizen L und R auch Bandmatrizen. L ist hierbei eine Bandmatrix mit $q_L = q_A$ und $p_L = 1$ und R ist eine Bandmatrix mit $q_R = 1$ und $p_R = p_A$.

Führt man z.B. die LR-Zerlegung der Matrix

$$A = \begin{pmatrix}
* & * & & & & \\
* & * & * & & & \\
 & * & * & \ddots & & \\
 & & * & \ddots & * & \\
 & & & \ddots & * & * \\
 & & & & * & *
\end{pmatrix}, \; p_A = q_A = 2$$

durch, so haben die Matrizen L und R folgende Struktur:

$$L = \begin{pmatrix} 1 & & & & & \\ * & 1 & & & & \\ & * & 1 & & & \\ & & * & \ddots & & \\ & & & \ddots & 1 & \\ & & & & * & 1 \end{pmatrix}, \qquad R = \begin{pmatrix} * & * & & & & \\ & * & * & & & \\ & & * & \ddots & & \\ & & & \ddots & * & \\ & & & & * & * \\ & & & & & * \end{pmatrix}.$$

Splines

Als Anwendungsbeispiel des Gleichungssystemlösers für Bandmatrizen soll eine Funktion mit Splines interpoliert werden. Zunächst ein paar Worte zum Thema Interpolation. Im Verlauf der Vorlesung wird die Interpolation noch ausführlich besprochen werden. Die Grundidee ist, eine Funktion, von der die Funktionswerte nur an bestimmten Stellen (den sogenannten Stützstellen) bekannt sind, durch eine Interpolationsfunktion darzustellen. Die Interpolationsfunktion geht dabei auf jeden Fall durch die gegebenen Stützpunkte. Der übrige Verlauf der Interpolationsfunktion hängt vom verwendeten Interpolationsverfahren ab. In diesem Beispiel ist dieses Verfahren eben die Spline-Interpolation. Dabei werden zusätzlich zu den Stützpunkten noch die Ableitungen an den beiden Intervallgrenzen vorgegeben. Ziel ist die Berechnung der Ableitung an sämtlichen Stützstellen. Wie das Verfahren genau funktioniert, ist an dieser Stelle nicht relevant, es wird in der Vorlesung und den Tutorien ausführlich besprochen. Die Bestimmung der Ableitungen entspricht der Lösung des folgenden linearen Gleichungssystems:

$$\begin{pmatrix} 4 & 1 & & \\ 1 & 4 & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & 4 \end{pmatrix} \begin{pmatrix} y_1' \\ y_2' \\ \vdots \\ y_{n-2}' \\ y_{n-1}' \end{pmatrix} = \frac{3}{h} \begin{pmatrix} y_2 - y_0 - \frac{h}{3} y_0' \\ y_3 - y_1 \\ \vdots \\ y_{n-1} - y_{n-3} \\ y_n - y_{n-2} - \frac{h}{3} y_n' \end{pmatrix}.$$

Die Matrix A dieses LGS ist eine Bandmatrix mit $q_A = p_A = 2$. Durch Vorgabe eines Arrays mit $n+1$ Stützwerten, den Intervallgrenzen x_0 und x_n sowie den Ableitungen an den Intervallgrenzen y_0' und y_n' wird das LGS eindeutig lösbar. In der Klasse *Spline* sollen Methoden implementiert werden, die zu den eben genannten Daten einen Vektor mit den Ableitungen y_1' bis y_{n-1}' berechnen.

Konkrete Aufgaben

Im Folgenden werden die zu implementierenden Methoden aufgelistet. Details zur Implementierung finden Sie jeweils in den Kommentaren zu den einzelnen Methoden.

- Klasse *Gauss*, Methode *loese*: Gauß-Elimination mit Spaltenpivotsuche.
- Klasse *LRZerlegung*, Methode *zerlege*: LR-Zerlegung der Matrix A.
- Klasse *LRZerlegung*, Methode *substitution*: Vorwärts- und Rückwärts-Substitution.
- Klasse *Spline*, Konstruktor: Bandmatrix und zugehörige LR-Zerlegung erzeugen.
- Klasse *Spline*, Methode *loese*: Baut aus den Stützwerten und Randbedingungen den Vektor b auf und berechnet die Ableitungen.

Formalien

- Das Programmgerüst erhalten Sie auf den Webseiten zur Vorlesung.
- Ergänzen Sie das Programmgerüst bitte **nur an den dafür vorgegebenen Stellen!** Falls Sie die Struktur der Programme eigenmächtig verändern, können wir sie evtl. nicht mehr testen.
- Beseitigen Sie vor Abgabe Ihres Programms alle Ausgaben an die Konsole!
- Bitte reichen Sie Ihre Abgabe bis zum **18. Januar 2009, 12:00 Uhr** über das Web-Portal ein. Die Auswertung steht dann ab spätestens 20. Januar, 12:00 Uhr zum Abruf bereit.