Familienname:

Vorname:

Matrikelnummer:

Studienkennzahl:

Email:

Aufgabe 1 (8 Punkte):
Aufgabe 2 (14 Punkte):
Aufgabe 3 (8 Punkte):

Gesamtpunktzahl:

Note:

Matlab - Nachtest (90 Minuten)
VU Einführung ins Programmieren für TM

3. März 2008

Aufgabenstellung. Gegeben seien eine reguläre Matrix $A \in \mathbb{R}^{n\times n}$ sowie eine rechte Seite $b \in \mathbb{R}^n$. Eine Möglichkeit zur Lösung des linearen Gleichungssystems $Ax = b$ ist das Gaußsche Eliminationsverfahren, das im Folgenden mit und ohne Pivot-Suche realisiert werden soll.

Aufgabe 1 (8 Punkte). Es sei $A \in \mathbb{R}^{n\times n}$ eine reguläre untere Dreiecksmatrix, d.h.

$$A = \begin{pmatrix} a_{11} & & & & \mathbf{0} \\ a_{21} & a_{22} & & & \\ a_{31} & a_{32} & a_{33} & & \\ \vdots & \vdots & \vdots & \ddots & \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{pmatrix}$$

bzw. $a_{jk} = 0$ für $k > j$. Dann gilt $a_{jj} \neq 0$ für alle $j = 1, \dots, n$, und zu gegebenem $b \in \mathbb{R}^n$ existiert ein eindeutiges $x \in \mathbb{R}^n$ mit $Ax = b$. Man schreibe mittels geeigneter Schleifen eine Funktion `solveL`, die x berechnet. — Um den Algorithmus herzuleiten, schreibe man das Matrix-Vektor-Produkt $b = Ax$ komponentenweise für b_j mit $j = 1, \dots, n$ als Summe hin. Man überlege, wie die spezielle Gestalt von A die Laufindizes der Summe vereinfacht und löse diese Gleichung nach x_j auf.

Herleitung der Formel für x_j.

Lösung zu Aufgabe 1.

Aufgabe 2 (14 Punkte). Falls die reguläre Matrix $A \in \mathbb{R}^{n\times n}$ keine untere Dreiecksmatrix ist, kann man A durch geeignete Eliminationsschritte auf untere Dreiecksgestalt bringen. Dies entspricht dem Vorgehen, wenn man ein lineares Gleichungssystem händisch löst:

- Im ersten Eliminationsschritt zieht man geeignete Vielfache der letzten Zeile von den übrigen Zeilen ab und erhält dadurch eine Matrix der Form

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1,n-1} & 0 \\ a_{21} & a_{22} & \dots & a_{2,n-1} & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{n,n-1} & a_{nn} \end{pmatrix},$$

- Im zweiten Eliminationsschritt zieht man nun geeignete Vielfache der vorletzten Zeile von den übrigen Zeilen ab und erhält eine Matrix der Form

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & 0 & 0 \\ a_{21} & a_{22} & \dots & 0 & 0 \\ a_{31} & a_{32} & \dots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ a_{n-1,1} & a_{n-1,2} & \dots & a_{n-1,n-1} & 0 \\ a_{n1} & a_{n2} & \dots & a_{n,n-1} & a_{nn} \end{pmatrix},$$

- Nach $n-1$ Eliminationsschritten erhält man also eine untere Dreiecksmatrix A.

Man beachte, dass auch die rechte Seite $b \in \mathbb{R}^n$ geeignet modifiziert werden muss. Das resultierende Gleichungssystem $Ax = b$ mit regulärer unterer Dreiecksmatrix kann dann mittels `solveL` aus Aufgabe 1 gelöst werden. Schreiben Sie eine Funktion `gauss`, die diesen Algorithmus realisiert und die Lösung $x \in \mathbb{R}^n$ eines Gleichungssystems $Ax = b$ berechnet. — Machen Sie sich das Vorgehen zunächst an einem Beispiel mit $A \in \mathbb{R}^{2\times 2}$ sowie $A \in \mathbb{R}^{3\times 3}$ klar.

Betrachtung des Beispiels $A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$.

Betrachtung des Beispiels $\boldsymbol{A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}}$**.**

Lösung von Aufgabe 2.

Aufgabe 3 (8 Punkte). Das Gaußsche Eliminationsverfahren scheitert, falls im k-ten Schritt $a_{kk} = 0$ gilt, auch wenn das Gleichungssystem $Ax = b$ eine eindeutige Lösung x besitzt, so beispielsweise in den Fällen

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & 0 \end{pmatrix} \quad \text{oder} \quad A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & 1 & 1 \\ a_{31} & 1 & 1 \end{pmatrix}.$$

Deshalb kann man das Verfahren um eine sogenannte *Pivot-Suche* erweitern:

- Im k-ten Schritt wählt man aus $a_{1k}, \dots, a_{kk}$ das betragsgrößte Element a_{pk}.
- Dann vertauscht man die k-te und die p-te Zeile von A (und b).
- Schließlich führt man den Eliminationsschritt aus wie zuvor.

Schreiben Sie eine Funktion `gausspivot`, die dieses Vorgehen realisiert. Realisieren Sie die Funktion möglichst rechenökonomisch. Sofern möglich, vermeiden Sie insbesondere das unnötige Kopieren von Speicher.

Lösung von Aufgabe 3.