

Angewandte Numerik 2

Abgabetermin: Freitag, 14.11.2014, vor der Übung

Für dieses Übungsblatt gibt es 15 Theorie- und 22 Matlab-Punkte, sowie 7 Theorie- und 7 Matlab-Zusatzpunkte. Punkte, die mit einem * gekennzeichnet sind, sind Zusatzpunkte. Die 50-Prozent-Grenzen liegen aktuell (inklusive Blatt 4) bei 40 Theoriepunkten und 37 Matlabpunkten.

Aufgabe 14 (*reduzible und irreduzible Matrizen*)

(4T+4T Punkte)

Gegeben seien die beiden Matrizen

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \quad \text{und} \quad B = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 2 & 2 & 0 & 0 & 1 \end{pmatrix}.$$

Zeigen Sie, dass A irreduzibel und B reduzibel ist.

Aufgabe 15 (*Gradienten-Verfahren*)

(7T Punkte)

Gegeben sei das lineare Gleichungssystem

$$\begin{pmatrix} 5 & 5 \\ 5 & 10 \end{pmatrix} x = \begin{pmatrix} 4 \\ 7 \end{pmatrix}.$$

Führen Sie zwei Schritte des Gradienten-Verfahrens mit dem Startwert $x^{(0)} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ durch.

Aufgabe 16 (*Programmieraufgabe: Gradienten-Verfahren für lineare Gleichungssysteme*)(6M+3M Punkte)

- Schreiben Sie eine Matlabfunktion `xk = gradVerfahrenLinear(A, b, x0, maxIt, tol)`, die mit dem Gradienten-Verfahren das lineare Gleichungssystem $Ax = b$ iterativ löst. `x0` soll dabei der Startwert und `maxIt` eine obere Schranke für die Anzahl der durchgeführten Iterationen sein. `tol` soll die Genauigkeit der Lösung über ein geeignetes Abbruchkriterium steuern. Ihre Matlabfunktion soll im Vektor `xk` alle Iterationswerte $x^{(k)}$ zurück geben.
- Testen Sie Ihre Matlabfunktion am Beispiel der vorigen Aufgabe. Zeichnen Sie den Verlauf der Iteration in ein Schaubild.

Aufgabe 17 (Programmieraufg.: Gradienten-Verfahren für allg. Funktionen) (10M+3M+3M*+4M* Pkte)

- a) Schreiben Sie, ausgehend von Ihrer Matlabfunktion `xk = gradVerfahrenLinear(A, b, x0, maxIt, tol)` aus der letzten Aufgabe, eine Matlabfunktion `xk = gradVerfahrenAllgemein(f, g, x0, maxIt, tol, beta, gamma)`, die das Gradientenverfahren mit Schrittweitenregel von Armijo zur Minimierung einer stetig differenzierbaren Funktion $f : \mathbb{R}^n \rightarrow \mathbb{R}$ implementiert.

f und g sollen dabei Matlabfunktionen sein, die den Funktionswert $f(x)$ der Funktion f an der Stelle x bzw. den Wert $g(x) = \nabla f(x)$ des Gradienten g von f an der Stelle x zurück geben. x_0 ist wieder der Startwert, `maxIt` eine obere Schranke für die Anzahl der durchgeführten Iterationen. `tol` ist die Toleranz für die Abbruchbedingung $\|g(x^{(k)})\| \leq \text{tol} * \min(1, \|g(x^{(0)})\|)$. `beta` und `gamma` sind die Parameter β und γ der Schrittweitenregel von Armijo.

Rückgabewert soll wieder ein Vektor `xk` mit allen Iterationswerten $x^{(k)}$ sein.

Realisieren Sie die Schrittweitenregel von Armijo in einer eigenen Matlabfunktion `armijo`. Überlegen Sie sich, welche Parameter Sie dieser Funktion übergeben müssen und welche Werte die Funktion zurück geben muss.

- b) Wenden Sie das von Ihnen implementierte Gradienten-Verfahren auf die Rosenbrock-Funktion

$$f : \mathbb{R}^2 \rightarrow \mathbb{R}, \quad f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

mit Startwert $x^{(0)} = \begin{pmatrix} -1.2 \\ 1 \end{pmatrix}$ an. Verwenden Sie hierbei für die Schrittweitenregel von Armijo die Parameter $\beta = 0.5$ und $\gamma = 10^{-4}$, für die Toleranz `tol` den Wert 10^{-2} .

- c) Zeichnen Sie mit dem Matlab-Befehl `contour` die Höhenlinien der Rosenbrock-Funktion. Zeichnen Sie in dieses Schaubild auch den Iterationsverlauf ein.
- d) Wie viele Iterationsschritte benötigt das Gradientenverfahren mit den vorgegebenen Parametern? Können Sie sich erklären, warum?

Aufgabe 18 (Wahr oder falsch?)

(7T* Punkte)

Geben Sie an, ob die folgenden Aussagen wahr oder falsch sind. Begründen Sie Ihre Antwort.

- a) Die Matrix $\begin{pmatrix} 2 & 0 & 1 \\ 1 & 2 & 0 \\ 0 & 1 & 2 \end{pmatrix}$ ist strikt diagonaldominant.
- b) Falls die Matrix A strikt diagonaldominant ist, konvergiert das Jacobi-Verfahren für jeden Startvektor.
- c) Falls das Gauß-Seidel-Verfahren für eine Matrix A konvergiert, so konvergiert auch das Jacobi-Verfahren für diese Matrix.
- d) Falls das Gauß-Seidel-Verfahren für eine Matrix A konvergiert, so konvergiert auch das Gradienten-Verfahren für diese Matrix.
- e) Das Gauß-Seidel-Verfahren ist genau dann für eine Matrix A konvergent, wenn A irreduzibel ist und das schwache Zeilensummenkriterium erfüllt.
- f) Durch Einführung eines Relaxationsparameters kann das Gauß-Seidel-Verfahren beschleunigt werden.
- g) Heute wird das Gradienten-Verfahren hauptsächlich als Glätter für das Gauß-Seidel-Verfahren eingesetzt.

Hinweise:

Die Programmieraufgaben sind in Matlab zu erstellen. Senden Sie alle Files in einer E-mail mit dem Betreff **Loesung-Blatt04** an angewandte.numerik@uni-ulm.de (Abgabetermin jeweils wie beim Theorieteil). Drucken Sie zusätzlich allen Programmcode sowie die Ergebnisse aus und geben Sie diese vor der Übung ab. Der Source Code sollte strukturiert und, wenn nötig, dokumentiert sein.