Prof. Dr. Karsten Urban
Dipl.-Math. oec. Klaus Stolle
Institut für Numerische Mathematik
Universität Ulm

WiSe 2015/2016
Blatt 4
06.11.2015

Angewandte Numerik 2

Abgabetermin: Freitag, 20.11.2015, **vor der Übung**

Dieses Übungsblatt hat eine Bearbeitungszeit von zwei Wochen.
Am Freitag, 13.11.2015 fallen die Übungen aufgrund der 10-Jahres-Feier des Ulmer Zentrums für Wissenschaftliches Rechnen aus. Übungsblatt 5 wird am Freitag, 13.11.2015 ebenfalls mit Abgabetermin 20.11.2015 auf der Homepage veröffentlicht.

Am Mittwoch, 18.11.2015 kann die Vorlesung aufgrund des Studientags nur bis 9:30 Uhr statt finden.

Für dieses Übungsblatt gibt es 7 Theorie- und 14 Matlab-Punkte, sowie 10 Theorie- und 12 Matlab-Zusatzpunkte. Punkte, die mit einem * gekennzeichnet sind, sind Zusatzpunkte.
Die 50-Prozent-Grenzen liegen aktuell (inklusive Blatt 4) bei 39 Theoriepunkten und 34,5 Matlabpunkten.

Aufgabe 14 *(Programmieraufgabe: Richardson-Verfahren)* (5M+6M*+2M* Punkte)

In dieser Aufgabe sollen Sie den optimalen Wert für den Parameter $\omega = \frac{1}{\gamma}$ des Richardson-Verfahrens

$$x^{(k+1)} = (I - \omega A)x^{(k)} + \omega b = x^{(k)} - \omega(Ax^{(k)} - b), \qquad k = 0, 1, 2, \dots$$

mit Hilfe eines numerischen Experiments bestimmen.
Dabei können Sie folgendermaßen vorgehen:

a) Schreiben Sie eine Matlabfunktion `x = richardson(A, b, x0, omega, N)`, die `N` Schritte des Richardson-Verfahrens zur Lösung des linearen Gleichungssystems $Ax = b$ durchführt. `x0` soll dabei als Startwert verwendet werden, und `omega` ist der Parameter für das Richardson-Verfahren. Ihre Funktion soll den Näherungswert für die Lösung des linearen Gleichungssystem nach `N` Schritten des Richardson-Verfahrens zurück geben.

b) Schreiben Sie ein Matlab-Skript `runRichardson`, in dem Sie

(1) die Matrix A, die rechte Seite b und den Startwert x_0 initialisieren. Verwenden Sie

$$A = \begin{pmatrix} 2 & 1 & 1 \\ -2 & 2 & -2 \\ -1 & 1 & 2 \end{pmatrix}, \quad b = \begin{pmatrix} 2 \\ 0 \\ 2 \end{pmatrix} \quad \text{und} \quad x_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

(2) die exakte Lösung x^* des linearen Gleichungssystems $Ax = b$ unter Verwendung des \-Operators berechnen,

(3) `omega` als einen Vektor mit 200 Werten aus dem Intervall $[0.15, 0.3]$ initialisieren,

(4) für jeden Wert von `omega` die Näherungslösung $x^{(20)}$ des Richardson-Verfahrens nach `N = 20` Iterationsschritten sowie den Fehler $||x^{(20)} - x^*||_2$ dieser Näherungslösung berechnen (für die Norm können Sie die Matlabfunktion `norm` verwenden) und

(5) die Fehler über `omega` plotten.

c) Welches ist der optimale Wert für den Parameter `omega`?

Aufgabe 15 *(Konvergenz des Jacobi-Verfahrens)* (7T Punkte)

Prüfen Sie zunächst mit Hilfe des Satzes 2.2.11 (Schwaches Zeilensummenkriterium), ob das Jacobi-Verfahren für die Matrizen

$$A_1 = \begin{pmatrix} 4 & 0 & 0 & -4 \\ 3 & -3 & 0 & 0 \\ 0 & 3 & 3 & 0 \\ 0 & 0 & -3 & 4 \end{pmatrix} \quad \text{und} \quad A_2 = \begin{pmatrix} 4 & 0 & 0 & -4 \\ 1 & 4 & 1 & -1 \\ -1 & 1 & 4 & 1 \\ -4 & 0 & 0 & 4 \end{pmatrix}$$

zur Lösung des Gleichungssystems $A_1 x = b$ bzw. $A_2 x = b$ geeignet ist. Kann Satz 2.2.11 nicht weiter helfen, so versuchen Sie die Konvergenz auf andere Art und Weise zu bestätigen oder zu widerlegen. Zur Berechnung von Eigenwerten dürfen Sie geeignete Hilfsmittel wie beispielsweise Matlab verwenden.

Aufgabe 16 *(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 17 *(Programmieraufgabe: Gradienten-Verfahren für lineare Gleichungssysteme)* (6M+3M+3M*+2T* Punkte)

a) Schreiben Sie eine Matlabfunktion `xk = gradVerfahrenLinear(A, b, x0, maxIt, tol)`, die mit dem Gradienten-Verfahren das lineare Gleichungssystems $Ax = b$ 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.

b) Berechnen Sie mit Ihrer Matlabfunktion `gradVerfahrenLinear` näherungsweise das Minimum der quadratischen Funktion $f : \mathbb{R}^2 \to \mathbb{R},\ f(x) = \frac{1}{2} x^T A x - b^T x$ mit

$$A = \begin{pmatrix} 2 & 1 \\ 1 & 5 \end{pmatrix} \quad \text{und} \quad b = \begin{pmatrix} 1 \\ 5 \end{pmatrix}.$$

Wählen Sie als Startwert $x_0 = \begin{pmatrix} 2 \\ 2 \end{pmatrix}$ und zeichnen Sie den Verlauf der Iteration in ein Schaubild.

c) Ergänzen Sie Ihr Schaubild durch einen zweidimensionalen Plot der Funktion f und zeichnen Sie Höhenlinien in Ihr Schaubild ein. Achten Sie auf sinnvolle Achsenabschnitte und wählen Sie geeignete Höhenlinien. Sie können beispielsweise die Matlabfunktionen `meshgrid`, `pcolor`, `colorbar`, `shading` und `contour` verwenden.

d) Wie verläuft die Iteration des Gradientenverfahrens hinsichtlich der Höhenlinien?

Aufgabe 18 *(10-Jahres-Feier UZWR)* (1T*+1M* Punkte)

Nehmen Sie an geeigneten Programmpunkten der 10-Jahres-Feier des Ulmer Zentrums für Wissenschaftliches Rechnen (UZWR) am Freitag, 13.11.2015, ab 11:00 Uhr im Senatssaal der Universität Ulm teil.

Hinweise:

Die Programmieraufgaben sind in Matlab zu erstellen. Senden Sie alle Matlab-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.