

Angewandte Numerik 2

Abgabetermin: Freitag, 20.11.2015, vor der Übung

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

Für dieses Übungsblatt gibt es 14 Theorie- und 25 Matlab-Punkte, sowie 16 Theorie- und 16 Matlab-Zusatzpunkte. Punkte, die mit einem * gekennzeichnet sind, sind Zusatzpunkte.

Die 50-Prozent-Grenzen liegen aktuell (inklusive Blatt 5) bei 46 Theoriepunkten und 47 Matlabpunkten.

Aufgabe 19 (*Konvergenzrate des Gradienten-Verfahrens*)

(1T+8T+1T+2T+2T* Punkte)

Betrachten Sie das lineare Gleichungssystem

$$Ax = b \quad \text{mit} \quad A = \begin{pmatrix} 1 & 0 \\ 0 & a \end{pmatrix} \quad \text{und} \quad b = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

wobei $a \gg 1$. Auf dieses Gleichungssystem wollen wir das Gradienten-Verfahren mit dem Startwert $x_0 = (a, 1)^T$ anwenden.

- Wie lautet die exakte Lösung x^* dieses Gleichungssystems?
- Zeigen Sie, dass für die Iterierten $x^{(k)} = (x_1^{(k)}, x_2^{(k)})^T$ des Gradienten-Verfahrens

$$x^{(k+1)} = \begin{pmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \end{pmatrix} = \rho \cdot \begin{pmatrix} x_1^{(k)} \\ -x_2^{(k)} \end{pmatrix} \quad \text{mit} \quad \rho = \frac{a-1}{a+1}$$

gilt.

Hinweis: Sie können diese Behauptung mit vollständiger Induktion zeigen.

- Geben Sie die Konvergenzrate $c^{(k)}$ im k -ten Schritt an, die definiert ist als $c^{(k)} = \frac{\|x^{(k)} - x^*\|}{\|x^{(k-1)} - x^*\|}$.
- Wie lauten die Konvergenzraten $c^{(k)}$ für $a = 1$ und für $a = 1000$? Was bedeuten diese für die Iterationen des Gradienten-Verfahrens?
- Zeichnen Sie für $a = 20$ die ersten Iterierten x_0, \dots, x_5 in ein zweidimensionales Koordinatensystem. Achten Sie auf eine geeignete Skalierung der Achsen.

Aufgabe 20 (*cg-Verfahren*)

(8T* Punkte)

Betrachten Sie die quadratische Funktion $f(x) = \frac{1}{2}x^T A x - b^T x$, $x \in \mathbb{R}^2$, mit

$$A = \begin{pmatrix} 2 & 0 \\ 0 & 8 \end{pmatrix} \quad \text{und} \quad b = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Bestimmen Sie das Minimum von f unter Verwendung des cg-Verfahrens. Wählen Sie als Startwert $x_0 = \begin{pmatrix} 4 \\ 1 \end{pmatrix}$ und rechnen Sie mit Brüchen. Erklären Sie in jedem Schritt, welche anschauliche Bedeutung dieser Schritt hat. Sie müssen nicht den Algorithmus 2.4.1 aus dem Skript anwenden, sondern können sich an der Herleitung des cg-Verfahrens in der Vorlesung orientieren. Wie viele Iterationsschritte haben Sie benötigt, um das exakte Minimum zu berechnen? Woran liegt das?

Aufgabe 21 (*Programmieraufgabe: cg-Verfahren*)

(6M+3M+3M* Punkte)

- Schreiben Sie eine Matlabfunktion `xk = cgVerfahren(A, b, x0, maxIt, tol)`, die mit dem cg-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 steuern. Ihre Matlabfunktion soll im Vektor `xk` alle Iterationswerte $x^{(k)}$ zurück geben.
- Schreiben Sie ein Matlabskript `testCgVerfahren`, das Ihre Matlabfunktion `cgVerfahren` am Beispiel aus Aufgabe 16 vom letzten Übungsblatt testet. Zeichnen Sie den Verlauf der Iteration in ein Schaubild. Zeichnen Sie in das gleiche Schaubild auch den Verlauf der Iteration des Gradienten-Verfahrens. Erläutern Sie das Schaubild.
- Erweitern Sie Ihr Matlabskript `testCgVerfahren`. Zeichnen Sie in ein neues Schaubild den Verlauf der Iteration des Gradienten-Verfahrens und den Verlauf der Iteration des cg-Verfahrens für das Beispiel aus Aufgabe 20. Erklären Sie auch dieses Schaubild.

Aufgabe 22 (*Programmieraufgabe: cg-Verfahren*)

(4M*+2T* Punkte)

- Erweitern Sie Ihr Matlabskript aus Aufgabe 17 vom letzten Übungsblatt: Zeichnen Sie in das Schaubild auch den Iterationsverlauf des cg-Verfahrens ein. Zur Berechnung der Iterierten können Sie Ihre Matlabfunktion `xk = cgVerfahren(A, b, x0, maxIt, tol)` aus der obigen Aufgabe 21 verwenden.
- Wie verläuft die Iteration des cg-Verfahrens hinsichtlich der Höhenlinien?

Aufgabe 23 (*Programmieraufgabe: Konvergenzraten Gradienten- und cg-Verfahren*)

(10M+2T+3M*+2T* Pkte)

- Schreiben Sie ein Matlabskript `konvergenzRaten`, mit dem Sie die Konvergenzraten des Gradienten-Verfahrens und des cg-Verfahrens vergleichen können.

Die Konvergenzrate eines Verfahrens im k -ten Schritt sei dabei wie in Aufgabe 19 definiert als $c^{(k)} = \frac{\|x^{(k)} - x^*\|}{\|x^{(k-1)} - x^*\|}$. x^* ist die exakte Lösung, die Sie mit dem Matlaboperator `\` berechnen können. Als Norm können Sie die Euklidische Norm oder die Energie-Norm verwenden.

Wählen Sie als Testbeispiel eine Tridiagonalmatrix A der Dimension n , die auf der Diagonalen jeweils den Wert 2 und auf den beiden Nebendiagonalen jeweils den Wert -1 hat. Mit den Matlabbefehlen `e = ones(n,1)`; `A = spdiags([-e 2*e -e], -1:1, n,n)` können Sie eine solche Matrix erzeugen. Wählen

Sie als rechte Seite $\mathbf{b} = \mathbf{ones}(n, 1)$ und als Startwert $\mathbf{x}_0 = \mathbf{zeros}(n, 1)$. Testen Sie für die Dimension $n = 100$, aber auch mit anderen Werten für n .

Zeichnen Sie in ein Schaubild für jeden Iterationsschritt die Konvergenzrate des Gradienten-Verfahrens und in ein weiteres Schaubild die Konvergenzrate des cg-Verfahrens ein.

- b) Erläutern Sie die Schaubilder. Erhalten Sie für unterschiedliche Werte von n qualitativ unterschiedliche Schaubilder?
- c) Vergleichen Sie die Konvergenzraten der beiden Verfahren auch am Beispiel aus Aufgabe 19.
- d) Erläutern Sie auch diese Schaubilder. Was passiert, wenn Sie a klein wählen, was, wenn a groß ist? Woran liegt das?

Aufgabe 24 (Programmieraufgabe: Vorkonditioniertes cg-Verfahren) (6M+3M+3M*+2T* Punkte)

a) Schreiben Sie eine Matlabfunktion $\mathbf{xk} = \text{pcgMatrix}(\text{invP}, \mathbf{A}, \mathbf{b}, \mathbf{x}_0, \text{maxIt}, \text{tol})$, die mit dem pcg-Verfahren das lineare Gleichungssystem $Ax = b$ iterativ löst. invP ist dabei die Vorkonditionierungsmatrix P^{-1} , \mathbf{x}_0 der Startwert und maxIt eine obere Schranke für die Anzahl der durchgeführten Iterationen. tol soll die Genauigkeit der Lösung steuern. Ihre Matlabfunktion soll im Vektor \mathbf{xk} alle Iterationswerte $x^{(k)}$ zurück geben.

b) Schreiben Sie ein Matlabskript `konvergenzRatenA24`, mit dem Sie die Konvergenzraten des cg-Verfahrens und des vorkonditionierten cg-Verfahrens vergleichen können. Sie dürfen auch Ihr Matlabskript `konvergenzRaten` aus der vorigen Aufgabe anpassen.

Verwenden Sie zum Vergleich der beiden Verfahren die auf der Homepage in der Datei `A.txt` bereitgestellte Matrix. Diese können Sie mit dem Matlabbefehl $\mathbf{A} = \text{load}(\text{'A.txt'})$ laden. Wählen Sie für die rechte Seite $\mathbf{b} = \mathbf{A} \cdot (1, \dots, 1)^T$, für den Startwert $\mathbf{x}^{(0)} = (0, \dots, 0)^T$ und für die Toleranz `tol` den Wert 10^{-10} .

Die Vorkonditionierung soll mittels Diagonalskalierung erreicht werden.

- c) Zeichnen Sie in ein Schaubild für jeden Iterationsschritt die Konvergenzrate des cg-Verfahrens und in ein weiteres Schaubild die Konvergenzrate des vorkonditionierten cg-Verfahrens ein. Dazu dürfen Sie Ihre Matlabfunktion `cgVerfahren(A, b, x0, maxIt, tol)` aus Aufgabe 21 verwenden.
- d) Was beobachten Sie? Berücksichtigen Sie bei Ihren Erläuterungen auch die Laufzeiten der beiden Verfahren (Matlab-Befehle `tic` und `toc`).

Hinweise:

Die Programmieraufgaben sind in Matlab zu erstellen. Senden Sie alle Matlab-Files in einer E-mail mit dem Betreff **Loesung-Blatt05** 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.