



Numerik 1

Matlab-Blatt 7

Abgabe wie auf dem ersten Matlab-Übungsblatt beschrieben
bis Dienstag, 12.02.2013, 24:00 Uhr per Email (s.u.)

- Alle Aufgaben auf diesem Blatt sind **Zusatzaufgaben** (deswegen auch mit * gekennzeichnet).
- Die Korrektur Ihrer eingereichten Lösungen erfolgt durch Ihren Tutor. **Nicht lauffähige** und / oder **schlecht kommentierte** Programme führen zu Punktabzügen.

Aufgabe 12 (*Rang-Approximation und Bildkompression*)

(7*+5* Punkte)

- a) Wir wollen zunächst die theoretischen Ergebnisse von Theorie-Übungsblatt 7, Aufgabe 27 numerisch validieren. Schreiben Sie dazu ein Programm `rankapproximation[A,maxNumOfRanks]`, das für eine gegebene Matrix $A \in \mathbb{R}^{m \times n}$ und eine maximale Anzahl an Rängen `maxNumOfRanks` eine Rang-Approximation wie in Aufgabe 27 beschrieben berechnet. Vergleichen Sie den resultierenden Approximationsfehler mit dem Resultat aus 27 a).
- b) Die Rang-Approximation kann man zur Datenkompression verwenden. Ein Schwarzweißbild z.B. lässt sich als Matrix speichern, bei der jeder Eintrag dem Grauwert des Pixels an der entsprechenden Position entspricht.
Laden Sie mit den Befehlen `load gatin`, `image(X)` und `colormap(map)` ein Schwarzweißbild¹. Nutzen Sie die Singulärwertzerlegung zusammen mit der Rangapproximation aus Aufgabe 27 um Kompressionen X_k der Matrix X (und damit des Bildes) zu erzeugen.
Plotten Sie Kompressionen des Bildes, z.B. für $k = 40$. Schreiben Sie in die Kommentare Ihres Programms den Speicherbedarf des Originalbildes als auch den des komprimierten Bildes für allgemeines k . Geben Sie dazu die jeweilige Anzahl an Grauwerten an, die gespeichert werden müssen.

Hinweis: Verwenden Sie die Funktion `[U,S,V] = svd(A)` um die Singulärwertzerlegung einer Matrix A zu berechnen. Programmrümpfe finden Sie auf der Homepage.

Aufgabe 13 (*Householder-Transformation und Ausgleichsprobleme*)

(3*+2*+2*+1* Punkte)

Wir betrachten die Gamma-Funktion

$$\Gamma(x) := \int_0^{\infty} t^{x-1} e^{-t} dt, \quad x > 0.$$

Für gegebene Werte $\{x_1, \dots, x_m\} \subset [0, 1]$, $m > 6$, wollen wir $\Gamma(1+x)$ durch ein Polynom fünften Grades, $p_5(x) := a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5$, möglichst gut approximieren. Dazu soll das lineare Ausgleichsproblem

$$\sum_{i=1}^m (\Gamma(1+x_i) - p_5(x_i))^2 \rightarrow \min$$

mittels QR -Zerlegung gelöst werden, wobei die QR -Zerlegung über eine Householder-Transformation berechnet werden soll.

¹Dieses Bild zeigt die sechs Numeriker J.H. Wilkinson, W. Givens, G. Forsythe, A. Householder, G. Henrici und F.L. Bauer auf der sog. Gatinburg-Tagung.

- a) Stellen Sie die Matrix $A \in \mathbb{R}^{m \times 6}$ und den Vektor $b \in \mathbb{R}^m$ zu obigem linearen Ausgleichsproblem in Matlab auf und lösen Sie das Problem mithilfe des Backslash-Operators. Nutzen Sie die Matlab-Funktion `gamma` zur Auswertung der Gamma-Funktion. Plotten Sie die Punkte $((x_1, \Gamma(1 + x_1)), \dots, (x_m, \Gamma(1 + x_m)))$ zusammen mit dem berechneten Polynom auf einem ausreichend feinen Gitter. Kommentieren Sie ihr Vorgehen.
- b) Lösen Sie das Ausgleichsproblem über die QR -Zerlegung mit Householder-Transformationen. Verwenden Sie dazu die Methode `[Q,R] = Householder(A)` aus dem Skript. Vergleichen Sie das Ergebnis mit dem Ergebnis aus a).
- c) Lösen Sie das Ausgleichsproblem über die QR -Zerlegung mit Householder-Transformationen. Verwenden Sie dazu die Methode `A = Householder(A)`, die Q und R kompakt in der Matrix A abspeichert und die Methoden aus `HouseholderMult.m` aus dem Skript. Vergleichen Sie das Ergebnis mit dem Ergebnis aus a).
- d) Plotten Sie den Absolutfehler für ein ausreichend feines Gitter.

Hinweis: Programmrümpfe und Routinen aus dem Skript finden Sie auf der Homepage.