Dr. Frank Wübbeling, Dipl.-Math. Marzena Franek WS 10/11

Übung zur Vorlesung Einführung in die Numerische Lineare Algebra

Übungsblatt 1, Abgabe: Montag, 18.10.2010, 12.00 Uhr

Übungstermine:

Gruppe 1:	Mi.	08.00 - 10.00 Uhr	N1
Gruppe 2:	Mi.	12.00 - 14.00 Uhr	N3
Gruppe 3:	Mi.	16.00 - 18.00 Uhr	N1

Aufgabe 1: (4 Punkte)

(a) Sei $f : \mathbb{R} \to \mathbb{R}$ eine Funktion und $x \in \mathbb{R} \neq 0$. Zeigen Sie: In jeder Maschinenarithmetik gilt

$$\lim_{h \to 0} \frac{f(x \oplus h) \ominus f(x)}{h} = 0.$$

(b) Erläutern Sie warum die Fehlerfunktion

$$\epsilon(h) := \left| \frac{f(x \oplus h) \ominus f(x)}{h} \ominus f'(x) \right|$$

für $h \to 0$ zunächst abnimmt und dann größer wird. Was muss man tun, um den Unterschied zwischen der Ableitung und dem numerischen Differenzenquotienten möglichst klein zu halten.

(c) Schreiben Sie einen Matlab-Algorithmus mit dem man die Maschinengenauigkeit ermitteln kann. Vergleichen Sie ihr Resultat mit dem in Matlab vorgegebenen Wert für *eps*.

Aufgabe 2: (4 Punkte)

(a) Berechnen sie alle Eigenvektoren und Eigenwerte der $n \times n$ - Matrix

$$A = \begin{pmatrix} 2 & -1 & 0 & \dots & \dots & 0 \\ -1 & 2 & -1 & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ \vdots & & & \ddots & \ddots & -1 \\ 0 & \dots & & 0 & -1 & 2 \end{pmatrix}.$$

Machen Sie für die Eigenvektoren den Ansatz $(x_k)_j = \sin(kjh)$ mit $h = \frac{\pi}{n+1}$ und nutzen Sie die Additionstheoreme für den Sinus und Cosinus. Begründen Sie die Wahl von h.

(b) Zeigen Sie, dass $u = \sin(kx)$ Eigenvektor des Differentialoperators $Lu = -u''$ ist, also dass $Lu = \lambda u$ für ein λ gilt. Begründen Sie zudem, dass für $h \to 0$ die Eigenwerte der Matrix $\frac{1}{h^2}A$ gegen die des Differentialoperators gehen.

Aufgabe 3: (4 Punkte)
Berechnen Sie mit Matlab das Ergebnis von $f(x) = \cos(1e^{-8}) - 1$ und den relativen Fehler. Geben Sie eine alternative Auswertungsvorschrift an und berechnen Sie wieder den relativen Fehler.

Aufgabe 4: (4 Punkte)
Die Fast-Fourier Transformation ist ein schneller Algorithmus um die Berechnung der diskreten Fourier Transformation durchzuführen. Man bezeichnet damit die Transformation eines Signals abhängig von der Zeit in den Frequenzbereich. Die umgekehrte Transformation wird Inverse Fourier Transformation genannt. Matlab stellt einige vordefinierte Matlab-Funktionen zur Verfügung, mit denen man leicht Fourier-Transformationen durchführen kann. Machen Sie sich mit
» lookfor fourier
» help fft
» help ifft
mit den Matlab-Funktionen vertraut und lösen Sie damit die folgende Aufgabe:
Gegeben sei das Signal

$$x(t) = 3e^{-\frac{t}{a}} \sin(2\pi \frac{f_0}{f_a} t) \quad \text{mit } t \geq 0,$$

mit der Grundfrequenz $f_0 = 1/\pi$ und Dämpfungskonstante $\alpha = 2s$. Das Signal wird folgendermaßen abgetastet:

(a) Abtastintervall $T_a = 0.2s$
Abtastfrequenz $f_a = 5Hz$
Abtastungen $N = 50$
Gesamte Messzeit $N * T_a = 10s$

(b) Abtastintervall $T_a = 0.1s$
Abtastfrequenz $f_a = 10Hz$
Abtastungen $N = 100$
Gesamte Messzeit $N * T_a = 10s$

Schreiben Sie ein Matlab-Programm, welches die beiden Signale in dem Bereich $0 \leq t \leq 10s$ darstellt. Berechnen Sie die Fast-Fourier Transformation beider Signale mit Hilfe der Matlab-Funktion *fft*, ermitteln Sie den Betrag und stellen Sie die Ergebnisse graphisch dar.
Wenden Sie auf Ihre Ergebnisse die Inverse Fourier Transformation (Befehl *ifft*) an.