Übungen zur Vorlesung Einführung in die Numerische Mathematik

Übungsblatt 12 , Abgabe: 25.01.2002 , 13.00 Uhr, Übungskasten F17, F25, F29

Die Programmieraufgabe 48 darf eine Woche später abgegeben werden.

Aufgabe 45: (4 Punkte)
Seien $x_0, \ldots, x_n$ paarweise verschiedene Stützstellen und $y_0, \ldots, y_n$ zugehörige Stützwerte.

Zeigen Sie: Ist $\sigma_0, \ldots, \sigma_n$ irgend eine Permutation der Zahlen $0, \ldots, n$, so ist

$$[y_0, \ldots, y_n] = [y_{\sigma_0}, \ldots, y_{\sigma_n}] \ .$$

Aufgabe 46: (4 Punkte)
Berechnen Sie sämtliche Eigenwerte und Eigenvektoren der Matrix W aus Satz 1 aus §3, Teil V.

Aufgabe 47: (4 Punkte)
Seien $f = (f_0, \ldots, f_{n-1})$ und $g = (g_0, \ldots, g_{n-1})$ Vektoren aus $I\!R^n$, welche wir periodisch fortsetzen durch

$$f_{m \cdot n + r} = f_r \qquad \forall\, m \in \mathbb{Z} \quad 0 \le r < n \ ,$$

ebenso für g. Wir nennen Vektoren mit dieser Eigenschaft n-periodisch. Für n-periodische Vektoren e, h definieren wir

$$\begin{array}{llcll} e \cdot h & \text{durch} \quad (e \cdot h)_j & = & e_j h_j \ , & \\ e * h & \text{durch} \quad (e * h)_j & = & \frac{1}{n} \sum\limits_{\ell=0}^{n-1} e_{j-\ell} h_\ell \ , & \text{die Faltung} \ , \end{array}$$

und $\hat{e}$ als n-periodische Fortsetzung der Fourier - Transformierten der Länge n von $(e_0, \ldots, e_{n-1})$.

(a) Zeigen Sie, daß für die n-periodischen Vektoren gilt

$$\widehat{f * g} = \hat{f} \cdot \hat{g} \ .$$

(b) Sei W die Matrix

$$W = \begin{pmatrix} w_0 & w_1 & \cdots & w_{n-1} \\ w_{n-1} & w_0 & \cdots & w_{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ w_1 & w_2 & \cdots & w_0 \end{pmatrix} \ , \quad w_i \in \mathbb{C} \ ,$$

und $y \in \mathbb{C}^n$. Zeigen Sie, daß Wy und $W^{-1}y$ (falls W^{-1} existiert) in $\mathrm{O}\,(n \log_2 n)$ Rechenoperationen berechnet werden kann ($n = 2^p$).

Aufgabe 48: (Programmieraufgabe, 4 Punkte) Schreiben Sie ein rekursives Programm $fft(y, n, \text{sign})$ zur schnellen Fourier - Transformation der Länge $n = 2^k$. Für sign $= -1$ soll die direkte, für sign $= +1$ die inverse Transformation durchgeführt werden. Prüfen Sie nach, ob Ihr Programm für einige Folgen komplexer Zahlen $y_0, \ldots, y_{n-1}$ nach den Aufrufen $fft(y, n, -1)$, $fft(y, n, +1)$, eventuell bis auf einen Faktor n, wieder die ursprüngliche Folge liefert.