Martin-Luther-Universität Halle-Wittenberg
Institut für Numerische Mathematik
Prof. Dr. M. Arnold, B. Burgermeister
Wintersemester 2005/2006

Numerische Mathematik II

8. Übungsblatt, Abgabe am 07.12.2005

http://sim.mathematik.uni-halle.de/~arnold/courses/SoS05.num/

Aufgabe 1. (4 Punkte) *Polynom-Approximation*

Bestimmen Sie zu den gegebenen Stützpunkten

t_i	0	1	2	3	4
f_i	0	1	2	1	0

ein Polynom höchstens zweiten Grades $P(t)$, so dass der mittlere quadratische Fehler $\frac{1}{5}\sum_{i=0}^{4}(P(t_i)-f_i)^2$ minimal wird.

Skizzieren Sie die Funktion $P(t)$ und zeichnen Sie die Stützpunkte ein.

Aufgabe 2. (4 Punkte) *Trigonometrische Polynome*

a) Zeigen Sie, dass für $t_1, \ldots, t_{2n} \in \mathbb{R}$ die Funktion $f(t) = \prod_{k=1}^{2n} \sin \dfrac{t-t_k}{2}$ ein trigonometrisches Polynom der Form $\dfrac{1}{2}a_0 + \sum_{j=1}^{n}(a_j \cos jt + b_j \sin jt)$ mit $a_j, b_j \in \mathbb{R}$ ist.

   Hinweis: Verwenden Sie $\sin\varphi = \frac{1}{2i}(e^{i\varphi} - e^{-i\varphi})$.

b) Zeigen Sie mit Hilfe von a), dass das interpolierende trigonometrische Polynom zu den Stützstellen t_k mit $0 \le t_0 < t_1 < \cdots < t_{2n} < \pi$ und den Stützwerten $f_0, \ldots, f_{2n}$ identisch mit

$$\Phi_{2n+1}(t) = \sum_{j=0}^{2n} f_j T_j(t)$$

   ist, wobei $T_j(t) := \dfrac{\displaystyle\prod_{k=0,k\neq j}^{2n} \sin \frac{t-t_k}{2}}{\displaystyle\prod_{k=0,k\neq j}^{2n} \sin \frac{t_j-t_k}{2}}$.

Bitte wenden!

Aufgabe 3. (4 Punkte) *Orthogonalitätsrelationen*

a) Zeigen Sie für ganzes j und $t_k := 2\pi k/(2m+1), \ (k = 0, 1, \ldots, m)$

$$\sum_{k=0}^{2m} \cos jt_k = (2m+1)h(j), \quad \sum_{k=0}^{2m} \sin jt_k = 0.$$

Dabei ist

$$h(j) := \begin{cases} 1 & \text{für } j = 0 \bmod 2m+1, \\ 0 & \text{sonst.} \end{cases}$$

b) Leiten Sie aus a) für ganzzahlige j, k folgende Orthogonalitätsrelationen her:

$$\sum_{l=0}^{2m} \sin jt_l \sin kt_l = \frac{2m+1}{2}(h(j-k) - h(j+k))$$

$$\sum_{l=0}^{2m} \cos jt_l \cos kt_l = \frac{2m+1}{2}(h(j-k) + h(j+k))$$

$$\sum_{l=0}^{2m} \cos jt_l \sin kt_l = 0.$$

Aufgabe 4. (4 Punkte) *Programmieraufgabe: Schnelle Fouriertransformation*

Schreiben Sie eine Funktion `al=fastft(f,inverse)` zur Fouriertransformation (`inverse=0`) und inversen Fouriertransformation (`inverse=1`) eines Eingabevektors `f` mit `length(f)=2^p`. Zur Bitmanipulation stehen in Matlab die Befehle `bitget` und `bitset` zur Verfügung.

Auf der Website zur Übung liegen Messwerte einer Traktorspur vor (File `Hhwayre.dat`). Diese Daten wurden freundlicherweise von Herrn Prof. Pickel von der Landwirtschaftlichen Fakultät zur Verfügung gestellt. Kopieren Sie dieses File auf Ihren Rechner und laden Sie die Daten mit dem Befehl `load`.

Bestimmen Sie mit Ihrer Funktion `fastft` die diskrete Fouriertransformierte der gegebenen Daten. Blenden Sie die Koeffizienten zu höheren Frequenzen aus, indem Sie sie zu Null setzen so dass nur die Elemente zu den niedrigsten 1000, 100 bzw. 40 Frequenzen erhalten bleiben und berechnen Sie jeweils die inverse Fouriertransformierte. Stellen Sie die Ursprungsdaten und die rücktransformierte Funktion grafisch dar. Falls die Rechenzeit auf Ihrem Rechner zu lang ist, benutzen Sie nur einen Teil der Messpunkte und reduzieren Sie die Anzahl der nicht ausgeblendeten Frequenzen entsprechend.

Welche Wirkung hat das Ausblenden der hohen Koeffizienten?

Abgabe der Programmieraufgabe per eMail an `burgermeister@mathematik.uni-halle.de`.