



Praktische Mathematik II

Sommersemester 2006

Prof. Dr. M. Griebel

Aufgabenblatt 2

Abgabe der Lösungen: 27.04.2006 in der Vorlesung,
Abgabe der Programmieraufgabe: 2.5.-5.5. im CIP-Pool

Aufgabe 1 (Hermite-Interpolation):

- a) Zeigen Sie, daß das Hermite-Interpolationspolynom $P(x)$ für die $n + 1$ Stützstellen (x_i, f_i) und Ableitungen (x_i, f'_i) , $0 \leq i \leq n$, die Lagrange'sche Darstellung

$$P(x) = \sum_{i=0}^n (f_i M_i(x) + f'_i N_i(x))$$

hat mit

$$\begin{aligned} M_i(x) &= (1 - 2(x - x_i)L'_i(x_i))(L_i(x))^2 \quad \text{und} \\ N_i(x) &= (x - x_i)(L_i(x))^2 \end{aligned}$$

wobei $L_i(x)$ die bekannten Lagrange-Interpolationspolynome sind.

- b) Wir definieren nun eine neue Folge von Stützstellen z_i , $0 \leq i \leq 2n - 1$, durch $z_{2i} = z_{2i+1} := x_i$. In dem Teilabschnitt $z_i \leq z_{i+1} \leq \dots \leq z_{i+k}$, $0 \leq i \leq i+k \leq 2n - 1$ komme x_j insgesamt m_j -mal vor. Definiere das Polynom $P_{i,i+1,\dots,i+k}$ als die Lösung des hermiteschen Interpolationsproblems

$$P_{i,i+1,\dots,i+k}(z_j) = f_j^{(r)} \quad \text{für } 0 \leq r \leq m_j - 1.$$

Man zeige, dass

$$P_{i,\dots,i+k}(x) = \sum_{r=0}^{m_j-1} \frac{f_j^{(r)}}{r!} (x - x_j)^r \quad \text{falls } z_i = z_{i+k} = x_j$$

und

$$P_{i,\dots,i+k}(x) = \frac{(x - z_i)P_{i+1,\dots,i+k} - (x - z_{i+k})P_{i,\dots,i+k-1}}{z_{i+k} - z_i} \quad \text{falls } z_i < z_{i+k}.$$

Es sei $f_{i,\dots,i+k}$ der Koeffizient von x^k des Polynoms $P_{i,\dots,i+k}$. Man zeige, dass für die Newtonsche Darstellung

$$P_{0,\dots,n}(x) = a_0 + a_1(x - t_0) + a_2(x - t_1)(x - t_0) + \dots + a_n(x - t_{n-1}) \dots (x - t_0)$$

die Relation $a_k = f_{0,\dots,k}$ gilt.

5 Punkte

Aufgabe 2 (Interpolationsfehler): Betrachten Sie den Fehler der Polynom-Interpolation einer Funktion f mit $|f^{(n+1)}(x)| \leq c$ auf dem Intervall $[-1, 1]$ für $n + 1$ äquidistante Stützstellen $x_i = -1 + \frac{2i}{n}$. Vergleichen Sie den Fehler $P(x) - f(x)$ in den Intervallen $[0, \frac{2}{n}]$ und $[1 - \frac{2}{n}, 1]$ für wachsendes n .

3 Punkte

Aufgabe 3 (Tschebyscheff-Interpolation)

- (a) Die normierten Tschebyscheff-Polynome T_n sind definiert als

$$\begin{aligned} T_0(x) &:= 1 \\ T_n(x) &:= 2^{1-n} \cos(n \cdot \arccos x) \text{ für } n \geq 1. \end{aligned}$$

Bestimmen Sie $T_1(x)$ bis $T_3(x)$ direkt aus dieser Definition.

- (b) Zeigen Sie, daß die normierten Tschebyscheff-Polynome für $n > 2$ der Rekursionsformel

$$T_{n+1}(x) = xT_n(x) - \frac{1}{4}T_{n-1}(x)$$

genügen und berechnen Sie daraus $T_4(x)$ und $T_5(x)$.

- (c) Zeigen Sie: die Nullstellen x_k , $1 \leq k \leq n$, des Tschebyscheff-Polynoms T_n sind für $n \geq 1$ gegeben als

$$x_k = \cos\left(\pi \frac{2k-1}{2n}\right).$$

- (d) Beweisen Sie die Minimaleigenschaft der normierten Tschebyscheff-Polynome:

$$\frac{1}{2^{n-1}} = \max_{x \in [-1,1]} |T_n(x)| \leq \max_{x \in [-1,1]} |Q_n(x)|$$

für alle normierten (d.h. mit führenden Koeffizienten 1) Polynome Q_n vom Grad n .

- (e) Beweisen Sie die Bestapproximationseigenschaft der Tschebyscheff-Polynome: $P(x)$ sei das Interpolationspolynom für die Stützstellen x_k , welche die Nullstellen von T_{n+1} sind. Dann gilt für jedes $f \in C^{n+1}$

$$\max_{x \in [-1,1]} |f(x) - P(x)| \leq \frac{1}{2^n(n+1)!} \max_{x \in [-1,1]} |f^{(n+1)}(x)|.$$

6 Punkte

Programmieraufgabe

Schreiben Sie folgende Programme:

- Den Neville-Algorithmus zur Auswertung eines Interpolationspolynoms an einer beliebigen Stelle x für gegebene (x_i, f_i) , $0 \leq i \leq n$.
- Den Newton-Algorithmus zur Bestimmung der Polynomkoeffizienten in Newton-Darstellung.
- Ein Verfahren zur direkten Auswertung des Interpolationspolynoms in Newton-Darstellung an einer beliebigen Stelle x .
- Das dividierte Differenzenverfahren zur Bestimmung der Polynomkoeffizienten in Newton-Darstellung.
- Das Horner-Schema zur Auswertung des Interpolationspolynoms in Newton-Darstellung an einer beliebigen Stelle x .

Betrachten Sie als Beispiel die Gauß'sche Hutfunktion

$$f(x) = \frac{1}{\sqrt{2\pi}} \cdot e^{-\frac{x^2}{2}}$$

im Intervall $[-3, 3]$ mit $n = 1, 3, 5, 9, 17$ und 33 Stützstellen.

- Werten Sie zur Überprüfung Ihrer Programme das Interpolationspolynom für $f(x)$ an den Stellen $x=0.02$, $x = 0.2$ und $x = 2$ einmal mit dem Neville-Algorithmus, einmal mit dem Newton-Algorithmus und dem direkten Verfahren und einmal mit dividierten Differenzen und dem Horner-Schema aus.
- Plotten Sie $f(x_k) - P(x_k)$ gegen n (Aufwand zu Genauigkeit) in einen gemeinsamen Graphen, wobei Sie Punkte für gleiches x_k miteinander verbinden. Verwenden Sie für beide Axen eine logarithmische Skala. Bestimmen Sie jeweils die Konvergenzrate.
- Vergleichen Sie den Rechenaufwand der Programme (a), (b)+(c) und (d)+(e) für den Fall, daß die Zahl der Stützstellen anwächst und für den Fall, daß die Zahl der Auswertungen anwächst.

15 Punkte