
Übung zur Vorlesung
Numerische Analysis
SoSe 2017 — Blatt 1

Abgabe: Do. 4. Mai 2017, vor der Vorlesung

Aufgabe 1 (Polynominterpolation)

(4 Punkte)

Sie haben in der Vorlesung die Polynominterpolation durch Lösung eines linearen Gleichungssystems diskutiert. Dabei haben Sie die Wahl der Polynombasis. Berechnen Sie das Interpolationspolynom für die folgenden Messwerte

x_i	1	4	9
f_i	1	2	3

mit Hilfe der

- (a) Normalform
- (b) Lagrange-Form
- (c) Newton-Form

Geben Sie die Matrix sowie das Polynom in der jeweiligen Darstellung an.

Aufgabe 2 (Čebyšev-Polynome)

(4 Punkte)

$T_n(t) = \cos(n \arccos(t))$, $t \in [-1, 1]$, heißt Čebyšev-Polynom der Ordnung n . Zeigen Sie:

- (a) Für $n > 0$ ist $\widehat{T}_n(t) = 2^{-n+1}T_n(t)$ ein Polynom n -ten Grades und der Koeffizient von x^n ist 1.
- (b) $\{T_i\}_{i=0}^n$ bilden eine Orthogonalbasis des Polynomraums P_n bezüglich eines gewichteten reellen L^2 -Innenprodukts, d.h.

$$\langle T_i, T_j \rangle := \int_{-1}^1 T_i(t) T_j(t) \frac{1}{\sqrt{1-t^2}} dt = \begin{cases} 0, & i \neq j \\ \pi, & i = j = 0 \\ \pi/2, & i = j \neq 0 \end{cases}$$

Aufgabe 3

(4 Punkte)

Seien $(x_j, y_j) \in \mathbb{R}^2, j = 0, 1, 2$ mit $y_1 < y_0, y_1 < y_2, x_1 - x_0 = x_2 - x_1 = h$.

- (a) Zeigen Sie, dass das zugehörige Interpolationspolynom $p \in \mathbb{P}^2$ ein eindeutig bestimmtes Minimum im Punkt x_* hat. Geben Sie eine Formel für x_* an.
- (b) Zeigen Sie, dass gilt $|x_1 - x_*| < \frac{h}{2}$.

Wiederholung aus der Vorlesung: Mit den dividierten Differenzen lassen sich die Koeffizienten der Newton-Polynome zu gegebenen Stützstellen und -Werten berechnen. Es ist durch die rekursive Formel

$$f[x_{i_0}, \dots, x_{i_k}] = \frac{f[x_{i_1}, \dots, x_{i_k}] - f[x_{i_0}, \dots, x_{i_{k-1}}]}{x_{i_k} - x_{i_0}} \quad f[x_{i_l}] = f_{i_l}$$

definiert. Dabei ist $f[x_{i_0}, \dots, x_{i_k}]$ der Koeffizient von $N_k(x)$ im Interpolationspolynom zu den Stützstellen $x_{i_0}, \dots, x_{i_k}$.

Aufgabe 4 (Programmieraufgabe)

(4 Punkte)

Auf der Homepage zur Vorlesung wird ein python-Skript bereit gestellt. Vervollständigen Sie es, indem Sie die folgenden Funktionen füllen:

- (a) Die Funktion `eval_nodes` soll eine gegebene Funktion an gegebenen Stützstellen auswerten und die Werte zurückgeben.
- (b) Die Funktion `div_diff` soll zu gegebenen Stützstellen und -Werten die Polynomkoeffizienten für die Newton-Polynome mit Hilfe von dividierten Differenzen berechnen und zurückgeben.
Hinweis: Zur Berechnung können Sie ein 2-dimensionales numpy-array verwenden, indem Sie die dividierten Differenzen speichern. Das array erstellen Sie z.B. mit `c = np.zeros((k,k))` und greifen mit `c[i,j]` auf die Einträge zu.
- (c) Die Funktion `horner` soll zu gegebenen Polynomkoeffizienten der Newton-polynome mit Hilfe des Horner-Schemas das Polynom an einer gegebenen Stelle auswerten und den Wert zurückgeben.

Testen Sie Ihren Code ausgiebig, indem Sie z.B. die Stützstellen oder die zu interpolierende Funktion ändern. Fügen Sie auch Stützstellen hinzu oder entfernen Sie welche.

Die Abgabe der Programmieraufgabe erfolgt über eine E-mail an `thomasbuddenkotte@freenet.de`. Möchten Sie eine schriftlich Korrektur Ihrer Programmieraufgabe, werfen Sie den ausgedruckten Quellcode bitte in Briefkasten 102. Die Abgabe der Theorieaufgaben werfen Sie bitte wie üblich in den Briefkasten Ihres Übungsgruppenleiters.

Achtung: Achten Sie darauf, Ihre Programme ordentlich zu formatieren und gut zu kommentieren. Die Form wird mit in die Bewertung eingehen.