

7. Übung zur Vorlesung

NUMERIK I

SoSe 2017

http://numerik.mi.fu-berlin.de/wiki/SS_2017/NumerikI.php

Abgabe: Fr., 16. Juni 2017, 12:00 Uhr

ALLGEMEINE HINWEISE

Die Punkte unterteilen sich in Theoripunkte (TP) und Programmierpunkte (PP). Bitte beachten Sie die auf der Vorlesungshomepage angegebenen Hinweise zur Bearbeitung und Abgabe der Übungszettel, insbesondere der Programmieraufgaben.

1. Aufgabe (12 PP)

- a) Programmieren Sie in MATLAB eine Funktion `newton_coeff = ddiff(f_info, x)`, die die Koeffizienten `newton_coeff` eines Hermite-Interpolations-Problems bezüglich der zugehörigen Newton-Basis berechnet. Dabei bezeichne `x` einen Vektor nicht notwendigerweise paarweise verschiedener Stützstellen und `f_info` ein Cell-Array, das Funktionen-Handles auf die zu interpolierende Funktion `f` und alle gegebenenfalls benötigten Ableitungen von `f` enthält.

Hinweis: Sie dürfen beim Implementieren von `newton_coeff` davon ausgehen, dass die Stützstellen `x` wie für die Hermite-Interpolation üblich in einer „geeigneten“ Reihenfolge gegeben sind.

- b) Schreiben Sie auch eine Funktion `horner(newton_coeff, x, t)`, die das Interpolationspolynom an einer beliebigen Stelle `t` oder ggf. auch an einem Vektor von Stellen `t` entsprechend dem Horner-Schema auswertet.
- c) Implementieren Sie nun eine Funktion `p = hermite_interpolation(f_info, x)`, die die Hermite-Interpolation zu den gegebenen Daten `f_info` und `x` wie aus a) berechnet. Hierbei soll `p` ein Funktionen-Handle auf das Interpolationspolynom sein, sodass mit `p(t)` das Interpolationspolynom an einem Punkt (oder einem Vektor von Punkten) `t` ausgewertet werden kann.
- d) Wir interessieren uns nun für die stückweise Hermite-Interpolation mit Grad $n \in \mathbb{N}$ auf $m \in \mathbb{N}$ Teilintervallen. Dazu sei ein Referenzgitter $0 \leq x_0 \leq \dots \leq x_n \leq 1$ auf $[0, 1]$ mit möglicherweise mehrfach vorliegenden Stützstellen gegeben. Ferner sei ein Gitter $a = y_0 < \dots < y_m = b$ mit paarweise verschiedenen Stützstellen auf $[a, b]$ mit $y_i := a + ih$ und $h := \frac{b-a}{m}$ gegeben. Gesucht ist nun diejenige Funktion $q: [a, b] \rightarrow \mathbb{R}$, die in jedem Teilintervall $[y_{i-1}, y_i]$ mit $i = 1, \dots, m$ die Hermite-Interpolationsaufgabe

$$q^{(j)}(y_{i-1} + hx_k) = f^{(j)}(y_{i-1} + hx_k) \quad \forall k \in \{0, \dots, n\}, x_k = \dots = x_{k+j} \quad (1)$$

erfüllt.

Programmieren Sie eine Funktion

```
q = piecewise_hermite_interpolation( f_info, x, a, b, m),
```

die die Lösung q des Problems (1) in Form eines Funktionen–Handles zurückgibt, sodass das Interpolations–Polynom q sowohl punktweise als auch vektorwertig ausgewertet werden kann.

Hinweis: Sie dürfen Ihre Implementation auf die bereits aus den vorherigen Aufgaben eingeführten Funktionen stützen.

- e) Verwenden Sie im Folgenden die Beispielfunktion f , die inklusive ihrer Ableitung in den Dateien `f.m` und `df.m` gegeben ist, sowie die Funktion $g(x) = |x|^5$. Sei das Referenzgitter

$$\Delta_0 := (0, 0, 1, 1)$$

gegeben und für alle $n \in \mathbb{N}$ sei das Gitter

$$\Delta(n) := \left(-1, -1, \frac{1-n}{n}, \frac{1-n}{n}, \dots, -\frac{1}{n}, -\frac{1}{n}, 0, 0, \frac{1}{n}, \frac{1}{n}, \dots, \frac{n-1}{n}, \frac{n-1}{n}, 1, 1 \right)$$

definiert. Berechnen Sie für $n \in \{1, 2, \dots, 20\}$ jeweils das zu f und zum Gitter $\Delta(n)$ gehörige Hermite–Interpolationspolynom p und das zum Referenzgitter Δ_0 und zu $a = -1$, $b = 1$, $m = n$ gehörige stückweise Hermite–Interpolationspolynom q . Geben Sie zu jedem n jeweils die Werte von $\|f - p\|_{\infty, 10^3}$ und $\|f - q\|_{\infty, 10^3}$ aus, wobei $\|\cdot\|_{\infty, 10^3}$ die Supremumsnorm $\|\cdot\|_{\infty}$ für $\varphi \in C^0([0, 1])$ durch

$$\|\varphi\|_{\infty, 10^3} := \max_{i=0, \dots, 10^3} \left| \varphi \left(\frac{i}{10^3} \right) \right|$$

annähert. Wiederholen sie diesen Test anschließend, wobei Sie nun f durch g ersetzen. Welche Konvergenzverhalten beobachten Sie? Wie sind die Ergebnisse zu erklären?

Hinweis: Da die Auswertung von `f` und `df` relativ viel Zeit in Anspruch nimmt, empfiehlt es sich während der Implementation, zunächst nur mit g zu testen. Alternativ können Sie auch versuchen, die gegebene Implementation von f zu verbessern.

2. Aufgabe (4 TP)

Es sei eine Folge von Gittern $a \leq x_0^\nu \leq \dots \leq x_n^\nu \leq b$ gegeben, welche gegen das Gitter $a \leq x_0 \leq \dots \leq x_n \leq b$ konvergiert, d. h. $\lim_{\nu \rightarrow \infty} x_k^\nu = x_k$ für $k = 0, \dots, n$. Es sei nun k_0 so gewählt, dass für jedes ν die maximale Anzahl zusammenfallender Stützstellen kleiner oder gleich $k_0 + 1$ ist, d. h.,

$$k_0 = \max_{\nu \in \mathbb{N}_0} \max_{k \in \{0, \dots, n\}} \{j \in \{0, \dots, n-k\} : x_k^\nu = \dots = x_{k+j}^\nu\}$$

Beweisen Sie für $f \in C^{k_0}([a, b])$, dass

$$\lim_{\nu \rightarrow \infty} f[x_0^\nu, \dots, x_n^\nu] = f[x_0, \dots, x_n].$$

3. Bonusaufgabe (4 TP)

Seien $n, k \in \mathbb{N}$ und für alle $i \in \{0, \dots, n\}$ sei $x_i \in \mathbb{R}$ und $\nu_i \in \{0, \dots, k\}$. Weiter sei $N := n + \sum_{i=0}^n \nu_i$. Zeigen Sie, dass für $i \in \{0, \dots, n\}$ und $j \in \{0, \dots, \nu_i\}$ Hermite–Lagrange–Funktionen $L_{ij} \in \mathcal{P}_N$ existieren, sodass für alle $f \in C^k(\mathbb{R})$ die Lösung $p \in \mathcal{P}_N$ der Hermite–Interpolationsaufgabe

$$\forall i \in \{0, \dots, n\} \forall j \in \{0, \dots, \nu_i\}: p^{(j)}(x_i) = f^{(j)}(x_i)$$

gegeben ist durch

$$p = \sum_{i=0}^n \sum_{j=0}^{\nu_i} f^{(j)}(x_i) L_{ij}.$$