Fachbereich Mathematik & Informatik
Freie Universität Berlin
Prof. Dr. Ralf Kornhuber, Max Kahnt

5. Übung zur Vorlesung

Numerik I

SoSe 2019

http://numerik.mi.fu-berlin.de/wiki/SS_2019/NumerikI.php

Abgabe: Fr., 24. Mai 2019, 16:00 Uhr

1. Aufgabe (4 TP)

Sei $a = x_0 < x_2 < \cdots < x_{n-1} < x_n = b$ eine Knotenverteilung im Intervall $I = [a, b]$. Für eine stetige Funktion $g \in C(I)$ ist der *interpolierende Linienzug* $\mathcal{I}g \in C(I)$ definiert durch

$$\begin{aligned} \mathcal{I}g(x_i) &= g(x_i), && \text{für } i = 0, \dots, n, \\ \mathcal{I}g\big|_{[x_i, x_{i+1}]} &\in \mathcal{P}_1 && \text{für } i = 0, \dots, n-1. \end{aligned}$$

a) Zeigen Sie: Für $g \in C^2(I)$ und *Gitterweite* $h = \max\limits_{0 \le i \le n-1} (x_{i+1} - x_i)$ gilt

$$\|g - \mathcal{I}g\|_\infty \le \frac{h^2}{8} \|g''\|_\infty.$$

b) Berechnen Sie die absolute Kondition des Linienzuginterpolationsoperators $\mathcal{I}$.

2. Aufgabe (4 TP)

a) Bestimmen Sie das Hermite-Interpolationspolynom $p \in P_4$, welches die Bedingungen

$$p(0) = 0, \quad p(1) = 1, \quad p'(1) = 4, \quad p''(1) = 12, \quad p'''(1) = 24$$

erfüllt.

b) Sei $f = cos$. Bestimmen Sie das Hermite-Interpolationspolynom $p \in P_5$ mit

$$\begin{aligned} p(-\pi) &= f(-\pi), & p'(-\pi) &= f'(-\pi), & p''(-\pi) &= f''(-\pi) \\ p(\pi) &= f(\pi), & p'(\pi) &= f'(\pi), & p''(\pi) &= f''(\pi). \end{aligned}$$

Skizzieren Sie anschließend die Graphen von f und p.

3. Aufgabe (4 TP)
Sei $n \in \mathbb{N}_{\geq 0}$ und $c_i \in \mathbb{R}$ für $i \in \{0, \dots, n\}$. Zeigen Sie, dass das Hermite–Interpolationsproblem auf $[-1, 1]$

$$\forall i \in \{0, \dots, n\}: \qquad p^{(i)}(-1) = c_i, \quad p^{(i)}(1) = (-1)^i c_i$$

eine eindeutige Lösung in $U := \operatorname{span}_{i \in \{0,\dots,n\}}\{x^{2i}\}$ besitzt.

4. Bonusaufgabe (4 PP)
Gegeben ist eine stetig differenzierbare Funktion $N : X \times \mathbb{R}^d \to \mathbb{R}$ durch

$$N(x,t) = \sum_{i=1}^{n} w_{o,i}\, a\left(\sum_{j=1}^{d} w_{ij} t_j - v_i\right) - v_o, \quad w = (w_{ij}), v = (v_i), w_o = (w_{o,i})$$

mit $X = \mathbb{R}^{n \times d} \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R} \ni x = (w, v, w_o, v_o)$, sowie $a : \mathbb{R} \to \mathbb{R}$, $a(s) = \max(s^3, 0)$. [1]

a) Implementieren Sie eine `matlab`-Funktion zur numerischen Auswertung von N.

Sei $F : \mathbb{R}^d \to \mathbb{R}$ gegeben durch

$$F(t) = N(x^*, t)$$

mit $x^* = (w^*, v^*, w_o^*, v_o^*) \in Z$ und $T = \{(t^{(i)}, F(t^{(i)}))\}_{i=1,\dots,I}, I \in \mathbb{N}$. [2]

b) Verwenden Sie die im `matlab`-Skript *parameter.m* gegebenen Werte für x^* und erzeugen Sie T aus den im selben Skript gegebenen Stellen $\{t^{(i)}\}_{i=1,\dots,I}$ für $n = 3, d = 2, I = 100$.

Wir betrachten das nichtlineare Gleichungssystem

$$N(z, x_i) = F(x_i), \quad i = 1, \dots, I. \tag{1}$$

c) Versuchen Sie das Gleichungssystem (1) mittels der `matlab`-Routine *fsolve* für die im Skript gegebenen Startwerte zu lösen. Interpretieren Sie die Ergebnisse.

Allgemeine Hinweise

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

[1] Die Funktion N kann als einfaches *künstliches neuronales Netz* aufgefasst werden, genauer als einschichtiges feedforward-Netz mit n künstlichen Neuronen, Gewichten w, Schwellenwerten v und Aktivierungsfunktion a. Die Komponenten des Vektors w_o heißen Ausgabegewichte und v_o nennt man Verschiebung.

[2] Im NN-Sprech auch *Trainingsmenge* genannt.