Fachbereich Mathematik & Informatik
Freie Universität Berlin
Prof. Dr. Carsten Gräser, Lasse Hinrichsen

1. Übung zur Vorlesung

Computerorientierte Mathematik II

SoSe 2020

http://numerik.mi.fu-berlin.de/wiki/SS_2020/CoMaII.php

Abgabe: Mo., 11. Mai 2020, 12:00 Uhr

1. Aufgabe (6 TP)
Seien $\varepsilon \in \mathbb{R}_{>0}$ und eine Funktion $f\colon \mathbb{R} \to \mathbb{R}$ gegeben. Von f sei darüber hinaus bekannt, dass es in den Punkten $x_1 = 0$, $x_2 = 1$, $x_3 = 1 + \varepsilon$ die Funktionswerte

$$f(x_1) = 0, \qquad f(x_2) = 0, \qquad f(x_3) = 1$$

annehme. Wir wollen f durch ein Interpolationspolynom vom Grad $d = 2$ an der Stelle $\overline{x} = \frac{1}{2}$ approximieren. Hierfür machen wir den Ansatz, dass wir die Koeffizienten $a = (a_0, a_1, a_2)^T \in \mathbb{R}^3$ desjenigen Polynoms

$$p = \sum_{k=0}^{d} a_k x^k$$

bestimmen möchten, das f in den Punkten x_i interpoliert. Zu diesem Zweck stellen wir wiederum das Gleichungssystem $Ma = b$ mit

$$M_{ij} = x_i^{j-1}, \qquad b_i = f(x_i) \qquad \text{für } i, j \in \{1, 2, 3\}$$

auf und lösen dieses anschließend nach a. Sobald a bekannt ist, können wir schließlich p an beliebigen Stellen $x \in \mathbb{R}$ auswerten, um den Funktionswert $f(x)$ (mehr oder minder gut) zu approximieren.

a) Stellen Sie in Abhängigkeit von ε die Matrix M und den Vektor b auf.

b) Berechnen Sie den Koeffizientenvektor a.

c) Werten Sie p an der Stelle $\overline{x}$ aus.

d) Bestimmen Sie die Kondition $\kappa_\infty(M)$ der Matrix M bezüglich der Maximumsnorm in Abhängigkeit von ε. Was geschieht für $\varepsilon \to 0$?

2. Aufgabe (4 TP)
Seien $n \in \mathbb{N}_{>0}$ paarweise verschiedene Punkte $x_i \in \mathbb{R}$ gegeben. Wir definieren $M \in \mathbb{R}^{n\times n}$ durch $M_{ij} = x_i^{j-1}$ für $i, j \in \{1, \dots, n\}$. Beweisen Sie, dass M regulär ist.

Bitte wenden!

3. Aufgabe (6 PP)
Ihnen liegt der folgende Python-Code vor:

```
def F(g, x, x0):
    y = 0
    n = len(x)

    for i in range(n):
        z = 1
        for j in range(n):
            if i != j:
                z *= (x0 - x[j]) / (x[i] - x[j])
        y += g(x[i]) * z

    return y
```

a) Beschreiben Sie kurz in Worten (höchstens ein Absatz) und mit Blick auf die Vorlesung, was dieser Code implementiert.

b) Bestimmen Sie den (etwaigen) Rechenaufwand des skizzierten Algorithmus, indem Sie die Gesamtzahl der arithmetischen Rechenoperationen bestimmen, die notwendig sind, um `F(g,x,x0)` auszuwerten. (Ignorieren Sie dabei die Aufrufe an `g`.)

c) Angenommen, wir interessieren uns für den Rückgabewert von `F` für viele verschiedene Werte von `x0`, aber mit immer denselben Parametern `g` und `x`. Warum ist die Verwendung der Funktion `F` für diese Situation ungeeignet und wie könnte das Problem besser gelöst werden?

Extrafrage (nicht bepunktet): Wie können Sie den obigen Code in Python so umschreiben, dass dort keine `for`-Schleifen mehr vorkommen?

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.