Technische Universität München
Institut für Informatik
Prof. Dr. Hans-Joachim Bungartz
Hayden Liu Weng
Sebastian Wolf
Michael Obersteiner

WiSe 2021/22

Numerisches Programmieren, Übungen

13. Übungsblatt: Klausurvorbereitung

1) Gleitkomma-Arithmetik (Klausuraufgabe aus dem WiSe 2009/10)

a) Bei der Lösung mathematischer Probleme im Computer lassen sich Diskretisierungsfehler nicht vermeiden. Dies beginnt schon beim Übergang in ein diskretes Zahlensystem wie das der Gleitkommazahlen.

In dieser Aufgabe gehen wir von einer Darstellung durch Gleitkommazahlen $\mathbb{F}_{B,t}$ der folgenden Form aus:

- Sei $B = 10$ die Basis der Darstellung
- Sei M die Mantisse zur Basis B mit $t = 4$ Stellen
- Sei E der Exponent (in beliebiger Darstellung)

Die Umrechnung eines Tupels (M, E) in die reellen Zahlen erfolgt dann über die Formel

$$(M, E)_{(\mathbb{F}_{10,4})} = (0, m_0 m_1 m_2 m_3 \cdot 10^E)_{(10)},$$

wobei die Eindeutigkeit der Darstellung durch die implizite 0 vor der Mantisse sowie die Forderung $m_0 \neq 0$ sichergestellt wird.

Stellen Sie die Hexadezimalzahl $(40a)_{(16)}$ im $\mathbb{F}_{10,4}$-System dar. Ist diese Darstellung exakt?

b) Rechnen Sie die reelle Zahl $\frac{43}{8}$ in eine Festkommazahl mit den folgenden Spezifikationen um!

- Die Darstellung erfolgt bezüglich der Basis 2.
- Das Vorzeichen wird explizit als "$+$" oder "$-$" angegeben.
- Die Darstellung besitzt genau eine Nachkommastelle.
- Es wird möglichst genau gerundet. Im uneindeutigen Fall wird abgerundet.

2) Interpolation (Klausuraufgabe aus dem WiSe 2010/11)

a) Gegeben sind die folgenden Stützpunkte:

$$P_0(6{,}2) \quad P_1(4{,}2) \quad P_2(2{,}3)$$

Ermitteln Sie den Funktionswert des durch die Stützpunkte (P_0, P_1, P_2) verlaufenden Polynoms p_0 an der Stelle $\tilde{x}_0 = 5$. Wählen Sie ein Verfahren, welches für die Erhöhung der Anzahl von Stützstellen sowie der Auswertung der Funktion an verschiedenen Stellen besonders geeignet ist.

Hinweis: Beachten Sie bei der Wahl Ihres Verfahrens, dass Sie in b) einen Stützpunkt P_3 hinzufügen werden und das resultierende Polynom p_1 an einer **anderen** Stelle $\tilde{x}_1$ auswerten müssen.

b) Erweitern Sie die in a) aufgestellte Interpolationsfunktion derart, dass Sie durch den Stützpunkt $P_3 = (3, 2)$ verläuft. Ermitteln Sie den Wert der resultierenden Interpolationsfunktion p_1 an der Stelle $\tilde{x}_1 = 2$.

c) Von einer zu interpolierenden Funktion seien an 20 Stützstellen die Funktionswerte gegeben. Zu diesen 20 Stützpunkten soll nun eine interpolierende Funktion ermittelt werden, die außerdem global mindestens zweimal stetig differenzierbar ist.
Nennen Sie zwei verschiedene interpolierende Funktionen, die die Anforderungen erfüllen. Welches dieser Verfahren ist Ihrer Meinung das bessere? Begründen Sie **kurz** Ihre Antwort.

3) Numerische Quadratur (Klausuraufgabe aus dem SS 2010)

$$\int_{-1}^{1} 7x^6 - x^2 \, dx. \tag{1}$$

Abildung 1 zeigt die Fehlerkurve für den Fehler von vier verschiedenen Quadraturmethoden, mit denen das Integral (1) numerisch berechnet wurde, abhängig von der Anzahl an Stützstellen n bzw. Teilintervallen $n - 1$.

Weisen Sie jeder der drei unten angegebenen Quadraturmethoden die zugehörige Fehlerkurve **A**, **B**, **C** oder **D** zu. Eine Fehlerkurve gehört zu keiner der aufgeführten Quadraturmethoden.

(1) Gaußquadratur ☐

(2) Trapezsumme ☐

(3) Simpsonsumme ☐

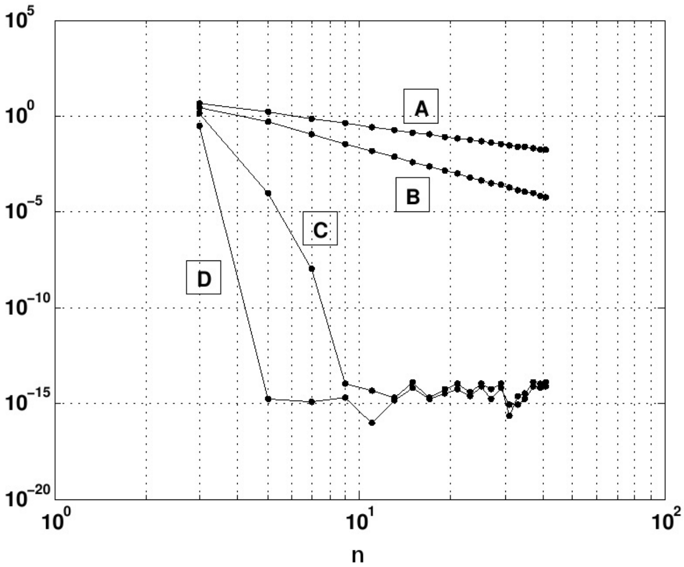

Abbildung 1: Fehler der verschiedenen Quadraturmethoden abhängig von der Anzahl an Stützstellen n (bzw. im Fall von Summenformeln abhängig von der Anzahl an Teilintervallen $n-1$)

4) Programmieraufgabe (Klausuraufgabe aus dem SoSe 2017)

In dieser Aufgabe wird Polynominterpolation mit Aitken-Neville betrachtet.

Bezeichne $p_{i,k}$ das eindeutig bestimmte Interpolationspolynom k-ten Grades zu den k+1 Stützpunkten $(x_i, y_i),\ \ldots,\ (x_j, y_j)$, wobei $k = j - i$.

$$\begin{aligned} p_{i,0}(x) &= y_i \\ p_{i,k}(x) &= p_{i,k-1} + \frac{x - x_i}{x_{i+k} - x_i} \cdot (p_{i+1,k-1} - p_{i,k-1}) \end{aligned}$$

Die sukzessive Berechnung der $p_{i,k}$ kann mit einem Dreiecksschema veranschaulicht werden:

$$\begin{array}{r|c|ccccccc}
x_i & i \setminus k & 0 & & 1 & & 2 & & \cdots \\
\hline
x_0 & 0 & p_{0,0} = y_0 & \rightarrow & p_{0,1} & \rightarrow & p_{0,2} & \rightarrow & \cdots \\
 & & & \nearrow & & \nearrow & & & \\
x_1 & 1 & p_{1,0} = y_1 & \rightarrow & p_{1,1} & \rightarrow & \vdots & & \\
 & & & \nearrow & & & & & \\
x_2 & 2 & p_{2,0} = y_2 & \rightarrow & \vdots & & & & \\
\vdots & \vdots & \vdots & & & & & &
\end{array}$$

Implementieren Sie nun die Methoden prepareAitkenNeville(..), welche die Initialisierung mit den allen übergebenen Stützstellen vornimmt, und runAitkenNeville(..), welche den Interpolationswert mit dem Schema von Aitken-Neville an der Stelle *interpolationsPunkt* berechnet und dabei alle übergebenen Stützstellen verwendet. Verwenden Sie dafür **Java**-ähnlichen Pseudocode.

Bitte auf nächster Seite bearbeiten

```java
// Berechnet Polynominterpolation mit Aitken-Neville Schema
public static double getInterpolation(double interpolationsPunkt,
    double[] stuetzstellen, double[] werteAnStuetzstellen) {

    int n = stuetzstellen.length;
    double[][] p = new double[n][n];

    //Implementieren Sie diese Methoden
    prepareAitkenNeville(werteAnStuetzstellen, n, p);
    return runAitkenNeville(interpolationsPunkt, stuetzstellen, n, p);
}

// Setzt Startwerte fuer k=0
private static void prepareAitkenNeville(double[] werteAnStuetzstellen,
    int n, double[][] p) {














}
// Berechnet mit Aitken Neville die Interpolation in interpolationsPunkt
private static double runAitkenNeville(double interpolationsPunkt,
    double[] stuetzstellen, int n, double[][] p) {
















}
```