



Bestätigung der Verhaltensregeln

Hiermit versichere ich, dass ich diese Klausur ausschließlich unter Verwendung der unten aufgeführten Hilfsmittel selbst löse und unter meinem Namen abgebe.

Unterschrift oder vollständiger Name, falls keine Stifteingabe verfügbar

Numerisches Programmieren

Klausur: IN0019 / Endterm

Datum: Mittwoch, 2. März 2022

Prüfer: Prof. Dr. Hans-Joachim Bungartz

Uhrzeit: 14:15 – 15:45

Bearbeitungshinweise

- Diese Klausur umfasst **14 Seiten** mit insgesamt **5 Aufgaben**.
Bitte kontrollieren Sie jetzt, dass Sie eine vollständige Angabe erhalten haben.
- Die Gesamtpunktzahl in dieser Klausur beträgt 55 Punkte.
- Das Heraustrennen von Seiten aus der Prüfung ist untersagt.
- Als Hilfsmittel sind zugelassen:
 - alle Hilfsmittel, insbesondere Bücher, persönliche Notizen, Internetsuchmaschinen, selbst erstellte Skripte und Programmcodes.
 - nicht erlaubt sind Hilfestellungen von Dritten oder Kommunikation mit Dritten.
 - nicht erlaubt sind Plagiate jeder Art.
- Mit * gekennzeichnete Teilaufgaben sind ohne Kenntnis der Ergebnisse vorheriger Teilaufgaben lösbar.
- **Es werden nur solche Ergebnisse gewertet, bei denen der Lösungsweg erkennbar ist.** Auch Textaufgaben sind **grundsätzlich zu begründen**, sofern es in der jeweiligen Teilaufgabe nicht ausdrücklich anders vermerkt ist.
- Schreiben Sie weder mit roter / grüner Farbe noch mit Bleistift.
- Beachten Sie die TUMExam Empfehlungen zur Abgabe von PDFs.

Aufgabe 1 Gleitkommazahlen und Kondition (10 Punkte)

Wir betrachten Gleitkommazahlen mit geringer Präzision. Eine Gleitkommazahl wird mit 16 Bits dargestellt, wobei ein Bit das Vorzeichen (S), acht Bit den Exponenten (E) und sieben Bits die Mantisse kodieren:

S	E	E	E	E	E	E	E	E	M	M	M	M	M	M	M
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Zahldarstellung und Rundung wird wie für IEEE754 double und single precision Gleitkommazahlen definiert. Der Bias für den Exponenten ist 127.

a) Stellen Sie die Zahl $\frac{19}{6}$ als Zahl im Binärsystem dar.

$$\left(\frac{19}{6}\right)_{10} = 10011_2 : 110_2$$

$$\begin{array}{r} 10011 : 110 = 11,00101... \\ \underline{110} \\ 111 \\ \underline{110} \\ 1000 \\ \underline{110} \\ 1000 \\ \underline{110} \\ 1000 \\ \underline{110} \\ 000 \end{array}$$

$$\Rightarrow \left(\frac{19}{6}\right)_{10} = 11.00\overline{1}_2.$$

b) Runden Sie die Zahl die Zahl $\frac{19}{6}$ korrekt in unser Gleitkommazahlssystem und geben Sie die Bitdarstellung an.

$$\frac{19}{6}_{10} = 1.100\overline{1}_2 \cdot 2^1$$

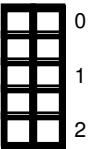
Vorzeichen = +: $S \Rightarrow 0$

Exponent: 1, der Exponent wird mit Bias 127 gespeichert: $E = 128_{10}$

Mantisse: 1.1001010101..., wir runden auf: $M = 1001011$

Die Bitdarstellung von $\left(\frac{19}{6}\right)_{10}$ ist 0|1000000|1001011.

c)* Finden Sie die kleinste positive Ganzzahl, die nicht exakt von unserem Gleitkommazahlssystem dargestellt werden kann.



$$255_{10} = 1.1111111|_2 \cdot 2^7$$

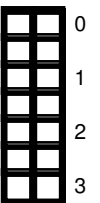
$$256_{10} = 1.0000000|_2 \cdot 2^8$$

$$257_{10} = 1.0000000|_2 \cdot 2^8$$

$$258_{10} = 1.0000001|_2 \cdot 2^8$$

257_{10} lässt sich also nicht mehr exakt in unserem Gleitkommazahlssystem darstellen. Alle kleineren Zahlen lassen sich exakt darstellen.

d)* Gegeben sei die Funktion $f(x) = \frac{1-x}{1+x}$, $x > 0$. Bestimmen Sie die relative Konditionszahl $\text{cond}(f, x)$ der Funktion f .



$$\text{cond}(f, x) := \left| \frac{x \cdot f'(x)}{f(x)} \right|.$$

Zunächst bestimmen wir $f'(x)$ mit der Quotientenregel:

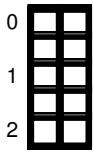
$$f'(x) = -\frac{2}{(1+x)^2}.$$

Damit berechnen wir die Kondition:

$$\text{cond}(f, x) = \left| \frac{x \cdot f'(x)}{f(x)} \right| = \left| \frac{\frac{-2x}{(1+x)^2}}{\frac{1-x}{1+x}} \right| = \left| \frac{-2x}{(1+x)^2} \cdot \frac{1+x}{1-x} \right| = \left| \frac{-2x}{(1+x)(1-x)} \right|$$

Aufgabe 2 Interpolation (13 Punkte)

Wir möchten eine Funktion mit kubischen Splines approximieren. Gegeben sind die Stützstellen $x_0 = 0, x_1 = 1, x_2 = 2$ und die Funktionswerte $y_0 = 4, y_1 = 5, y_2 = 8$. Außerdem sind die Randbedingungen $s'_0 = -1$ und $s'_2 = 1$ gegeben.



a) Berechnen Sie die fehlende Ableitung s'_1 .

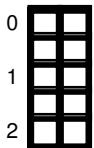
In Übung 4, Aufgabe 2 ist erklärt, wie man die kubischen Splines löst: Zunächst berechnen wir die Ableitung s'_1 . Wir müssen das lineare Gleichungssystem

$$\begin{pmatrix} 4 & 1 & & \\ 1 & 4 & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & 4 \end{pmatrix} \begin{pmatrix} s'_1 \\ s'_2 \\ \vdots \\ s'_{n-2} \\ s'_{n-1} \end{pmatrix} = \frac{3}{h} \begin{pmatrix} y_2 - y_0 - \frac{h}{3}s'_0 \\ y_3 - y_1 \\ \vdots \\ y_{n-1} - y_{n-3} \\ y_n - y_{n-1} - \frac{h}{3}s'_n \end{pmatrix}$$

lösen. In unserem Fall gibt es nur eine Gleichung und eine Unbekannte:

$$4 \cdot s'_1 = \frac{3}{h} \left(y_2 - y_0 - \frac{h}{3}s'_0 - \frac{h}{3}s'_2 \right)$$

Für $y_0 = 4, y_1 = 5, y_2 = 8$ ergibt sich als Lösung: $s'_1 = 3.0$.

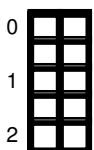


b)* Beschreiben Sie allgemein, wie man die Spline Funktion an einer Stelle $x_0 < \hat{x} < x_2$ auswertet.

Die Funktion s ist stückweise definiert. Zunächst müssen wir das **Intervall** finden, in dem sich \hat{x} befindet. Ist $x_i < \hat{x} < x_{i+1}$, dann ist

$$s(\hat{x}) = y_i H_0(t_i(\hat{x})) + y_{i+1} H_1(t_i(\hat{x})) + s'_i H_2(t_i(\hat{x})) + s'_{i+1} H_3(t_i(\hat{x})), \quad (2.1)$$

mit $t_i(\hat{x}) = (\hat{x} - x_i)/(x_{i+1} - x_i)$. Die Funktionen H_j sind die Hermite Polynome aus Aufgabe 1 auf Übungsblatt 4.



c)* Werten Sie den Spline Funktion mit den oben gegebenen Stützstellen an der Stelle $\hat{x} = 1.5$ aus. Falls Sie die Aufgabe a) nicht gelöst haben, verwenden Sie $s'_1 = 2$.

\hat{x} liegt zwischen x_1 und x_2 : $t_1(\hat{x}) = \hat{x} - 1 = 0.5$.

Wir werten also aus:

$$\begin{aligned} s(\hat{x}) &= y_1 \cdot H_0(0.5) + y_2 \cdot H_1(0.5) + s'_1 \cdot H_2(0.5) + s'_2 \cdot H_3(0.5) \\ &= y_1 \cdot \frac{1}{2} + y_2 \cdot \frac{1}{2} + s'_1 \cdot \frac{1}{8} - s'_2 \cdot \frac{1}{8}. \end{aligned}$$

Für $y_0 = 4, y_1 = 5, y_2 = 8$ ergibt sich als Lösung: $s(1.5) = 6.75$.

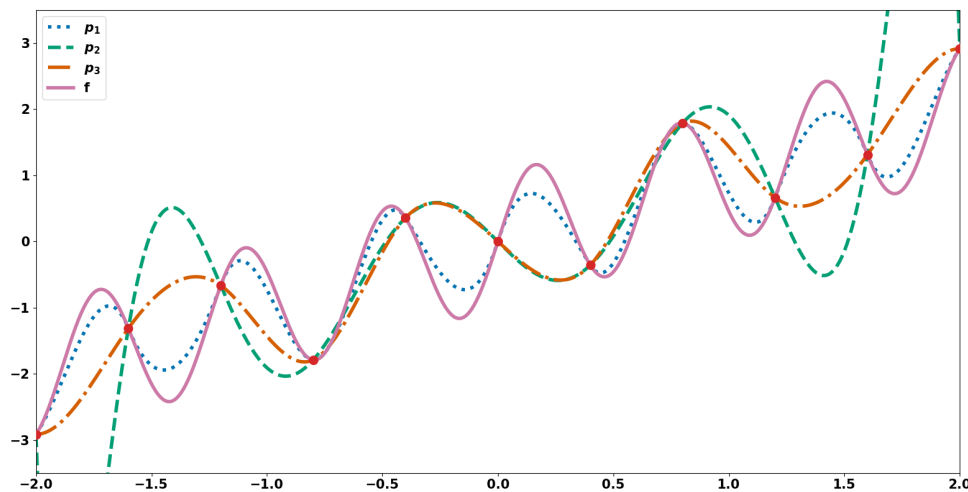


Abbildung 2.1: Die Funktion f und drei verschiedene Interpolationsfunktionen, sowie die Interpolationspunkte

d)* Gegeben Sei die Funktion $f(x) = x + \sin(10x)$. Gegeben sind außerdem die Stützstellen $x_i = -2, -1.6, -1.2, \dots, 1.6, 2.0$. Wir interpolieren diese Funktion mit drei verschiedenen Methoden:

- Polynominterpolation auf den Stützstellen x_i
- Kubische Splines mit $s'(x_0) = s'(x_n) = 0$
- Stückweise Hermite Interpolation

	0
	1
	2
	3

In Abbildung 2.1 sind die Funktion f , sowie die drei Interpolationsfunktionen p_1, p_2 und p_3 abgebildet. Geben Sie an, welcher Plot zu welcher Methode gehört und begründen Sie Ihre Wahl kurz.

- p_1 : Stückweise Hermite Interpolation
- p_2 : Polynominterpolation
- p_3 : Kubische Splines

Begründung:

p_2 oszilliert am Rand. Dieses Phänomen (Runge Effekt) tritt bei Polynominterpolation auf äquidistanten Stützstellen auf.

Wir erkennen, dass die Steigung von p_1 an den Interpolationspunkten mit der Steigung von f übereinstimmt. Die anderen beiden Interpolationsfunktionen tun dies nicht. Also muss p_1 die Hermite Interpolation sein. Bei p_3 erkennen wir, dass die Ableitung am Rand ($x = -2, x = 2$) gleich 0 ist. Also muss es sich bei p_3 um die kubische Spline Interpolation mit den angegebenen Randbedingungen handeln.

0			
1			
2			
3			
4			

e)* Wir wollen nun den Rechenaufwand für die Interpolation mit kubischen Splines und für die stückweise Hermite Interpolation analysieren. Nehmen Sie an, es sind n paarweise verschiedene Stützstellen x_0, \dots, x_{n-1} gegeben, sowie die Funktionswerte y_i und die Werte der Ableitungen y'_i . Analysieren Sie wie viele Gleitkommaoperationen ($+$, $-$, \cdot , \div) Sie benötigen um die Interpolationsfunktionen aufzustellen. Geben Sie darüber hinaus an, wie viele Gleitkommaoperationen Sie benötigen um die Interpolationsfunktion an einer Stelle auszuwerten. Sie müssen keine exakte Anzahl angeben, eine Abschätzung mit \mathcal{O} genügt. Begründen Sie Ihre Antwort kurz. Sie können davon ausgehen, dass die Stützstellen aufsteigend geordnet sind.

Kubische Splines:

Um die Werte der Ableitung des Splines an den Stützstellen zu berechnen, müssen wir ein Lineares Gleichungssystem der Größe $(n-2) \times (n-2)$ lösen: $\mathcal{O}(n^3)$. Achtung: Die Ableitung des Splines an einer Stützstelle entspricht nicht unbedingt der Ableitung der zu interpolierenden Funktion f !

Um den Spline an einer Stelle \hat{x} auszuwerten muss zunächst das Intervall $x_i < \hat{x} < x_{i+1}$ gefunden werden. Mit binärer Suche: $\mathcal{O}(\log(n))$. Dann ist der Aufwand nur noch konstant um die Hermite Polynome H_0, \dots, H_3 auszuwerten.

Hermite Interpolation:

Kein Setup notwendig: $\mathcal{O}(1)$, man kann die Funktion direkt auswerten, da die y'_i gegeben sind.

Aufwand für eine Auswertungen: $\mathcal{O}(\log(n))$, wie bei den Splines

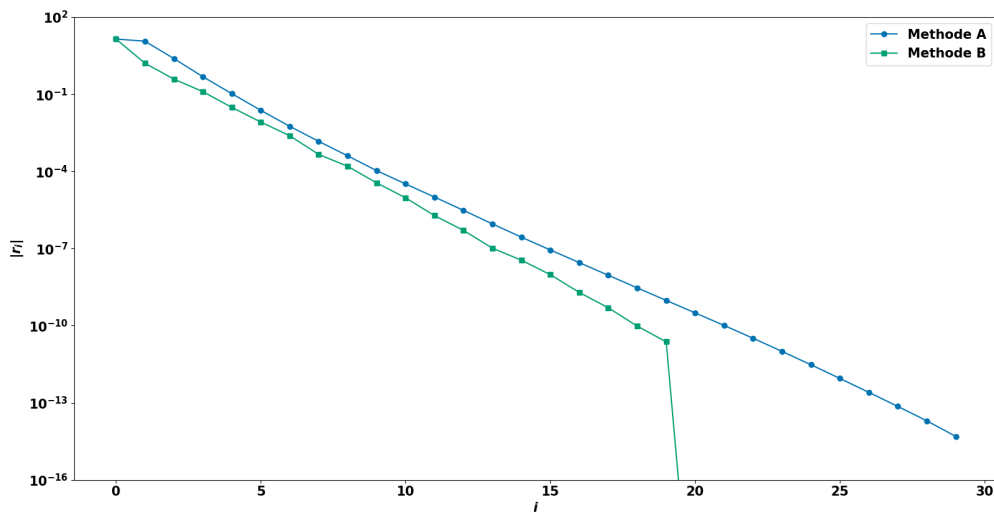


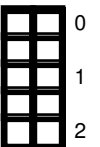
Abbildung 3.1: **Residuen** der verschiedenen Methoden

Aufgabe 3 Iterative Lösung linearer Gleichungssysteme (7 Punkte)

Wir betrachten das lineare Gleichungssystem $Ax = b$ mit $A \in \mathbb{R}^{20 \times 20}$ und $b \in \mathbb{R}^{20}$:

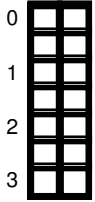
$$A = \begin{pmatrix} 4 & 1 & 0 & 0 & \cdots & 0 \\ 1 & 4 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 4 & 1 & \cdots & 0 \\ 0 & 0 & 1 & 4 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 4 \end{pmatrix}$$

a) Wir lösen das Gleichungssystem $Ax = b$ mit dem CG Verfahren und dem Gauss-Seidel Verfahren. In Abbildung 3.1 sind die **Residuen** der zwei Methoden nach der i ten Iteration angegeben. Ordnen Sie die Verfahren dem jeweiligen Plot zu und begründen Sie ihre Wahl kurz.



- Methode A: Gauss-Seidel
- Methode B: CG Verfahren

Begründung: Methode B gibt nach 20 Schritten die exakte Lösung (im Rahmen der Rechengenauigkeit) \Rightarrow CG Verfahren
Methode A ist also das Gauss-Seidel Verfahren.



b)* Finden Sie eine obere Schranke für die Kondition $\kappa_2(A)$

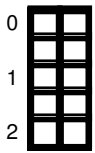
Der Satz von Gerschgorin erlaubt es uns die Eigenwerte von A abzuschätzen. Sie liegen innerhalb der Gerschgorinkreise. Wegen der Struktur von A finden wir nur zwei Kreise:

$$|\lambda - 4| \leq 1, \text{ und}$$

$$|\lambda - 4| \leq 2.$$

Dann liegen alle Eigenwerte im Intervall $[2, 6]$.

$$\therefore \kappa_2(A) = \frac{\lambda_{\max}}{\lambda_{\min}} \leq \frac{6}{2} = 3.$$



c)* Die Methoden in Teil a) sind alle (semi-)iterative Verfahren. Wäre es sinnvoll, ein direktes Verfahren anzuwenden? Begründen Sie Ihre Wahl kurz mit einem Vorteil (bzw. einen Nachteil).

Je nach Begründung können beide Antworten akzeptiert werden:

Vorteile: Die LR-Zerlegung erlaubt es, das Lineare Gleichungssystem für mehrere verschiedene rechte Seiten nur mit Vorwärts- und Rückwärtssubstitution zu berechnen (mit einem $\mathcal{O}(N^2)$ Aufwand). Der Thomas-Algorithmus kann es sogar in $\mathcal{O}(N)$ Laufzeit berechnen.

Nachteile: ein direktes Verfahren (z.B. die LR-Zerlegung) braucht in der Regel viel mehr Speicher und kann dadurch schnell mit steigendem N nicht mehr durchgeführt werden.

Aufgabe 4 Vektoriteration und Nullstellen von Polynomen (16 Punkte)

Für jedes Polynom $p(x) = x^n + a_{n-1}x^{n-1} + a_1x + a_0$ existiert die Begleitmatrix $A_p \in \mathbb{R}^{n \times n}$ mit

$$A_p = \begin{pmatrix} 0 & 0 & \dots & 0 & -a_0 \\ 1 & 0 & \dots & 0 & -a_1 \\ 0 & 1 & \dots & 0 & -a_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & -a_{n-1} \end{pmatrix}. \quad (4.1)$$

Die Eigenwerte dieser Matrix sind genau die Nullstellen von p . Um die betragsmäßig größte Nullstelle zu bestimmen, verwenden wir die Vektoriteration:

```
double groessteNullstelle = vektorIteration(begleitmatrix(polynomKoeffizienten), x_0);
```

Hinweis: Die Vektoriteration funktioniert auch für nicht-symmetrische Matrizen. Dies können Sie als gegeben annehmen und müssen es nicht weiter begründen.

Matrizen speichern wir als Arrays `double[][]`, wobei A_{ij} als `a[i][j]` gespeichert wird. Vektoren sind als Arrays `double[]` realisiert. Sie können die Funktionen

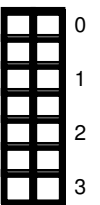
- `public static double[] matrixVektorMultiplikation(double[][] matrix, double[] vektor)`
- `public static double skalarprodukt(double[] x, double[] y)`
- `public static double norm(double[] x)`

verwenden ohne diese selbst implementieren zu müssen.

a) Füllen Sie das untenstehende Codegerüst aus, um die Matrix A_p zu bestimmen.

```
public static double[][] begleitmatrix(double[] polynomKoeffizienten) {
```

```
    int n = polynomKoeffizienten.length;
    double[][] matrix = new double[n][n];
    matrix[0][n-1] = -polynomKoeffizienten[0];
    for (int i = 1; i < n; i++) {
        matrix[i][i-1] = 1;
        matrix[i][n-1] = -polynomKoeffizienten[i];
    }
    return matrix;
}
```



```
}
```

0

 b)* Als nächstes implementieren wir das Abbruchkriterium $\|w^{(k)} - \lambda^{(k)}x^{(k)}\| \leq \epsilon \|w^{(k)}\|$. Füllen Sie das untenstehende Codegerüst aus, um zu überprüfen, ob $w^{(k)}, x^{(k)}, \lambda^{(k)}$ das Abbruchkriterium erfüllen. Geben Sie `true` zurück, wenn die Approximation des Eigenwertes gut genug ist. Verwenden Sie $\epsilon = 10^{-8}$.

1

 public static boolean abbruchKriterium(double[] w, double[] x, double lambda) {

2


```

    int n = w.length;
    assert x.length == n;
    double eps = 1e-8;
    double[] residuum = new double[n];
    for (int j = 0; j < n; j++) {
        residuum[j] = w[j] - lambda * x[j];
    }
    return (norm(residuum) <= eps * norm(w));

```

3

 }

0

 c)* Füllen Sie das untenstehende Codegerüst aus, um eine Vektoriteration zur Bestimmung des betragsmäßig größten Eigenwerts einer Matrix durchzuführen. Es sollen dabei maximal 10.000 Iterationen ausgeführt werden. Verwenden Sie das Abbruchkriterium aus Aufgabe b).

1

 public static double vektorIteration(double[][] matrix, double[] x_0) {

2


```

    int n = x.length;
    assert matrix.length == n;
    assert matrix[0].length == n;
    int maxIteration = 10000;
    double[] x = x_0.clone();

    double lambda = 0;
    for (int i = 0; i < maxIteration; i++) {
        double[] w = matrixVektorMultiplikation(matrix, x);
        lambda = skalarprodukt(x, w);
        if (abbruchkriterium(w, x, lambda) {
            break;
        }
        double wNorm = norm(w);
        for (int j = 0; j < n; j++) {
            x[j] = w[j] / wNorm;
        }
    }
    return lambda;

```

3

 }

d)* Für die Vektoriteration benötigen wir keinen expliziten Zugriff auf die Elemente der Matrix A_p . Anstatt die Matrix A_p aufzustellen und dann in `vektorIteration` die generische Matrix-Vektor-Multiplikation zu verwenden, können wir eine Funktion verwenden, die direkt die Multiplikation $A_p v$ auswertet: Füllen Sie das untenstehende Codegerüst aus, um das Ergebnis der Matrix-Vektor-Multiplikation $A_p \cdot v$ zu bestimmen. Stellen Sie die Matrix A_p nicht explizit auf.

```
public static double[] multipliziereMitBegleitmatrix(double[] polynomKoeffizienten, double[] vektor) {
```

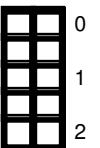
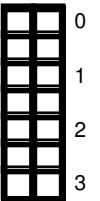
```
    int n = polynomKoeffizienten.length;
    assert n == vektor.length;
    double[] result = new double[n];

    result[0] = -polynomKoeffizienten[0] * vektor[n-1];
    for (int i = 1; i < n; i++) {
        result[i] = vektor[i-1] - polynomKoeffizienten[i] * vektor[n-1];
    }
    return result;
}
```

```
}
```

e)* Analysieren Sie die Anzahl an Gleitkommaoperationen (+, −, ·, ÷), die für `multipliziereMitBegleitmatrix` nötig sind und vergleichen Sie Ihr Ergebnis mit der generischen Matrix-Vektor-Multiplikation. Eine Abschätzung mit \mathcal{O} genügt.

Aufwand für `multipliziereMitBegleitmatrix`: $1 + 2 \cdot (n - 1) = \mathcal{O}(n)$
Aufwand für `matrixVektorMultiplikation`: $\mathcal{O}(n^2)$.



Aufgabe 5 Quadratur in der Wahrscheinlichkeitstheorie (9 Punkte)

Ein großer Anwendungsschwerpunkt der Quadratur ist die Wahrscheinlichkeitstheorie. Hier ist die Berechnung des Erwartungswertes eines der zentralen Elemente. Häufig wird betrachtet wie sich das Ergebnis einer Funktion verhält, wenn die Eingabewerte x einer Wahrscheinlichkeitsdichte $p(x)$ folgen. Der Erwartungswert $E[f]$ kann dann wie folgt berechnet werden:

$$E[f] = \int_D f(x) \cdot p(x) dx. \quad (5.1)$$

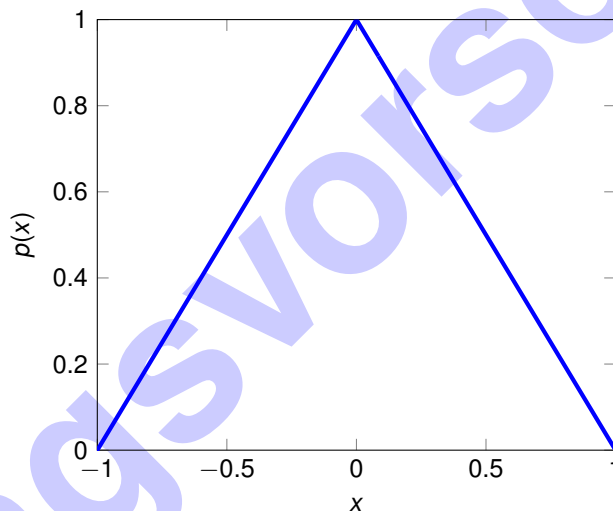
Folglich ist die Berechnung des Erwartungswertes nur eine spezielle Art der Quadratur der Funktion $f(x)$ bei der die Wahrscheinlichkeit der jeweiligen Werte berücksichtigt wird. Numerisch approximieren wir dieses Integral wieder mithilfe von Gewichten und Stützpunkten, d.h.

$$E[f] = \int_D f(x) \cdot p(x) dx \approx \sum_{i=0}^{n-1} w_i f(x_i). \quad (5.2)$$

In dieser Aufgabe betrachten wir Integrale dieser Art. Als Wahrscheinlichkeitsdichte verwenden wir die Dreiecksdichte

$$p(x) = -|x| + 1, \quad (5.3)$$

welche auch hier zu sehen ist:



Im Folgenden betrachten wir verschiedene Verfahren, welche für die Berechnung des Integrals verwendet werden könnten. Als Integrationsgebiet verwenden wir $D = [a, b] = [-1, 1]$.

a) Erläutern Sie warum die Verwendung der Trapezregel (also nur mit einem Trapez!) für die Berechnung des Integrals nicht geeignet ist.

Die Trapezregel kann nicht verwendet werden, da die Dichte an den Rändern 0 ist und somit der Erwartungswert immer 0 wäre.

b)* In der Praxis werden für spezielle Wahrscheinlichkeitsdichten eigene Quadraturregeln berechnet, welche das Integral effizienter berechnen. Hierfür kann, wie auf dem Übungsblatt, die Methode der unbestimmten Koeffizienten verwendet werden. Hierfür muss lediglich die rechte Seite angepasst werden:

$$f_k(x) := x^k$$

$$\sum_{i=0}^{n-1} f_k(x_i) \cdot w_i \stackrel{!}{=} \int_a^b f_k(x) p(x) dx = M_k.$$

Die Terme M_k werden auch als Momente der Wahrscheinlichkeitsdichte $p(x)$ bezeichnet.

Wir wollen nun eine Gaussquadratur mit 2 Punkten speziell für die Dreiecksdichte berechnen. Verwenden Sie hierfür $M_0 = 1$, $M_1 = 0$, $M_2 = 1/6$ und $M_3 = 0$. Zusätzlich kann zur Vereinfachung der Berechnung die Beziehung $x_0 = -x_1$ verwendet werden. Stellen Sie alle nötigen Gleichungen auf und berechnen Sie die Gewichte w_0 und w_1 und die Punkte x_0 und x_1 . Geben Sie auch den Rechenweg an!

$$\begin{aligned} w_0 + w_1 &\stackrel{!}{=} \int_a^b 1 p(x) dx = 1 \\ w_0 x_0 + w_1 x_1 &\stackrel{!}{=} \int_a^b x p(x) dx = 0 \\ w_0 x_0^2 + w_1 x_1^2 &\stackrel{!}{=} \int_a^b x^2 p(x) dx = 1/6 \\ w_0 x_0^3 + w_1 x_1^3 &\stackrel{!}{=} \int_a^b x^3 p(x) dx = 0 \end{aligned}$$

Lösung nahezu identisch zum Übungsblatt.

$$w_0 = w_1 = 0.5$$

$$x_0 = -\sqrt{1/6}, x_1 = +\sqrt{1/6}$$

c)* Berechnen Sie nun für die Funktion $f(x) = x^2 + 5$ und die Dreiecksdichte $p(x)$ den Erwartungswert $E[f]$ mithilfe der von Ihnen hergeleiteten Gaussquadratur. Falls Sie die Aufgabe b) nicht gelöst haben, verwenden Sie $x_0 = -\frac{1}{3}$, $x_1 = \frac{1}{3}$ und $w_0 = \frac{1}{3}$, $w_1 = \frac{2}{3}$.

$$Q_G = 0.5 * (1/6 + 5) + 0.5 * (1/6 + 5) = 31/6$$

d)* Ist der berechnete Erwartungswert aus der vorherigen Teilaufgabe exakt? Begründen Sie Ihre Antwort! Berechnen Sie dabei den Erwartungswert $E[f]$ nicht analytisch.

Ja, da die Gaussformel bis zum Grad 3 (bezogen auf $f(x)$) exakt ist.