Heinrich Heine Universität Düsseldorf

Mathematisches Institut
Prof. Dr. Achim Schädle
Melinda Hagedorn
Dr. Marina Fischer
Dr. Christoph Matern

12. April 2023

4	5	6	Σ

Name: _____

Name: _____

Numerik I – 2. Übungsblatt

Aufgabe 4: (3+3 Punkte)

(a) Bestimmen Sie das Volumen V des Rotationskörpers R, der gegeben ist als

$$R = \{(x, y, z) | z \in [-\tfrac{h}{2}, \tfrac{h}{2}], \|(x, y)\| \leq f(z)\}$$

mit einer stetigen Funktion $f : [-\frac{h}{2}, \frac{h}{2}] \to \mathbb{R}_{\geq 0}$.

(b) Wir wollen nun das Volumen eines Fasses nach der Kepler'schen Fassregel bestimmen: Wir können das Fass von Innen gemäß der Skizze in Abbildung 1 ausmessen. Dabei können wir die Längen $l_i,\ i = 1, 2, 3$, der (gestrichelten) Messstäbe und die Neigungswinkel $\varphi_i,\ i = 1, 3$ messen. Stellen Sie eine allgemeine Formel auf, die das Volumen des Fasses mit Hilfe der Simpsonregel approximiert, indem Sie die Länge der Stäbe und die Winkel an den beiden Extrempunkten messen.

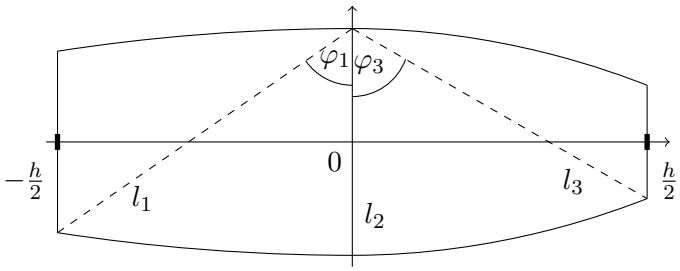
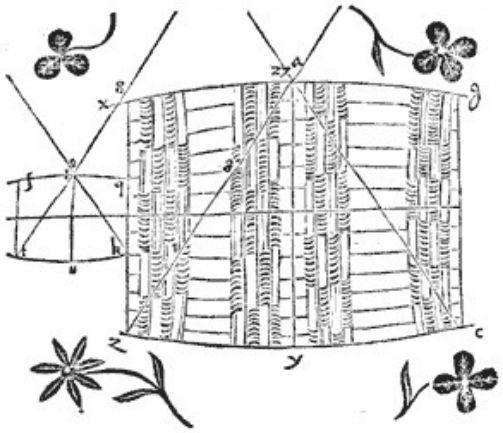


Abbildung 1: Veranschaulichung der Kepler'schen Messmethode für das Volumen eines Fasses



Quelle: `http://www.keplerraum.at/fass.html`

b.w.

Aufgabe 5: (6 Punkte)

Beweisen Sie, dass die Mittelpunktsregel optimal ist in dem Sinne, dass keine andere 1-stufige Quadraturformel Ordnung zwei oder mehr besitzt.

Aufgabe 6: (6 Punkte)

Beweisen Sie: Falls die Knoten einer Quadraturformel $(b_i, c_i)_{i=1}^s$ paarweise verschieden und symmetrisch im Sinne der Definition (2.5) i) sind, d.h.

$$c_i = 1 - c_{s+1-i},\ i = 1, \dots, s$$

erfüllen und ihre Ordnung $p \geq s$ ist, dann müssen auch die Gewichte symmetrisch sein, d.h.

$$b_i = b_{s+1-i},\ i = 1, \dots, s$$

erfüllen.

Programmieraufgabe 2:

Ihre Abgabe soll nur eine `.py` oder `.ipynb` Datei enthalten. Andere Dateien oder zip-Ordner sind weder nötig noch erwünscht!

(a) Implementieren Sie in PYTHON eine Methode `numint(f, gitter, b, c)`, welche für eine beliebige Quadraturformel $(b_i, c_i)_{i=1}^s$ auf einem Gitter

$$\texttt{gitter} = [x_0, x_1, x_2, \dots, x_N], \text{ mit } x_0 < x_1, \dots < x_N$$

und für eine gegebene Funktion $f : [x_0, x_N] \to \mathbb{R}$ die Approximation

$$\int_{x_0}^{x_N} f(x)\,\mathrm{d}x \approx \sum_{j=0}^{N-1} h_{j+1} \sum_{i=1}^{s} b_i f(x_j + h_{j+1} c_i) \tag{1}$$

berechnet (hierbei ist $h_{j+1} = x_{j+1} - x_j$) und als Rückgabewert vom Typ `float` zurückgibt. Vermeiden Sie `for`-Schleifen, d.h. implementieren Sie Ihre Funktion vektorisiert.

(b) Definieren Sie die Knoten und Gewichte der Rechtecksregel, Mittelpunktsregel und die beiden zweistufigen Verfahren $((b_1, b_2), (c_1, c_2)) = ((\frac{3}{4}, \frac{1}{4}), (\frac{1}{3}, 1))$ (Radau) und
$((b_1, b_2), (c_1, c_2)) = ((\frac{1}{2}, \frac{1}{2}), (\frac{1}{2} - \frac{\sqrt{3}}{6}, \frac{1}{2} + \frac{\sqrt{3}}{6}))$ (Gauß).

Laden Sie sich das PYTHON-Modul `datenp2.py` von der Vorlesungsseite herunter. Dieses definiert zwei Funktionen $\texttt{f} = x \mapsto f(x)$ und $\texttt{g} = x \mapsto g(x)$. Es beinhaltet auch eine Methode `erzeuge_gitter(N,typ)`, die für ein $\texttt{N} \in \mathbb{N}$ ein Gitter mit N Punkten in $[-1, 1]$ erzeugt (für `typ='aequi'` äquidistant und für `typ='cluster'` geclustert). Außerdem enthält es die Vergleichswerte $\texttt{fexakt} = \int_{-1}^{1} f(x)\,\mathrm{d}x$ und $\texttt{gexakt} = \int_{-1}^{1} g(x)\,\mathrm{d}x$.

(c) Testen Sie Ihr Verfahren wie folgt: Messen Sie den Approximationsfehler von (1) für die Verfahren aus (b) für $\texttt{N} = 5, 10, 20, 50, 100, 200, 500$ mit beiden Gittern und beiden Funktionen.

Tragen Sie für $f(x)$ in einem doppelt-logarithmischen Plot $\max_{1 \leq j \leq N}(h_j)$ gegen den Fehler für beide Gitter und alle Verfahren aus (b) auf. Denken Sie an Legenden und Beschriftungen und achten Sie darauf, dass die einzelnen Verfahren gut voneinander unterscheidbar sind.
Verfahren Sie (in einem neuen Plotfenster) analog mit $g(x)$.

Interpretieren Sie die Plots.

Abgabe der Übungsaufgaben am Mittwoch, 19. April bis 10:30 Uhr per ILIAS unter „Übungen zu Numerik I“.
Abgabe der Programmieraufgaben am Mittwoch, 19. April bis 10:30 Uhr per ILIAS unter „Programmierübungen zu Numerik I“.