Dr. Steffen Basting
Dipl.-Technomath. Christopher Basting
Numerische Mathematik für Physiker und Ingenieure

Sommersemester 2017
Übungsblatt 6

Abgabe der **Theorieaufgaben bis Freitag, 16.06.2017, 10:00 Uhr** (einzeln oder zu zweit) **in den Briefkasten Ihrer Übungsgruppe**. Bitte vermerken Sie auf Ihrer Abgabe Ihre Übungsgruppe sowie Name und Matrikelnummer. Heften Sie mehrere Blätter zusammen.
Abgabe der **Programmieraufgaben bis Freitag, 16.06.2017, 10:00 Uhr digital im Moodle-Arbeitsraum der Veranstaltung**. Bei Abgabe zu zweit bitte nur einmal einreichen und im Kommentar den Namen und E-Mail Adresse des/der Koautors/Koautorin nennen. Die Aufgaben sollen mit MATLAB bzw. OCTAVE gelöst werden.

Aufgabe 6.1 (Zusammengesetzte Quadraturformeln | 5+3 Punkte)

a) Berechnen Sie eine Näherung für

$$\ln 2 = \int_0^1 \frac{1}{1+x}\, dx$$

durch numerische Quadratur mit Hilfe der summierten Simpsonregel in einer Genauigkeit von $4 \cdot 10^{-5}$.

Hinweis: Bestimmen Sie zunächst die nötige Anzahl an Teilintervallen m, indem Sie die Fehlerabschätzung aus dem Satz zur Konvergenz zusammengesetzter Quadraturformeln verwenden.

b) Wie viele Teilintervalle m sind dagegen bei der summierten Trapezregel erforderlich, um die gleiche Genauigkeit garantieren zu können?

Aufgabe 6.2 (Gauss-Quadratur | 2+2+2 Punkte)

Bestimmen Sie folgende Integrale näherungsweise mit Hilfe der Gaußschen Quadraturformel mit 1 und 2 Punkten ($I_G^{(n)}$ mit $n = 0$ und $n = 1$). Bestimmen Sie jeweils den Fehler $\left|I_G^{(n)}(f) - I(f)\right|$.

a) $\displaystyle\int_0^1 \cosh x \,\mathrm{d}x\,, \qquad \left(\cosh x := \frac{e^x + e^{-x}}{2}\right),$

b) $\displaystyle\int_2^3 \frac{1}{x} \,\mathrm{d}x\,,$

c) $\displaystyle\int_0^1 4\sqrt{1-x^2}\, dx \qquad (= \pi).$

Programmieraufgabe 6.1 (Zusammengestzte Quadratur | 4 + 4 Punkte)

a) Implementieren Sie eine Funktion `[v]=myQuadraturSum1D(f,w,p,a,b,N)`, die eine zusammengestzte (summierte) Quadraturformel umsetzt. Dabei soll für die Eingabeargumente gelten:

- `f` ein `function_handle` der zu integrierenden Funktion;
- `w` ein Vektor der Dimension R welcher die Quadraturgewichte enthält;

- `p` ein Vektor der Dimension R welcher die Stützstellen der Quadraturformel auf dem Einheitsintervall enthält;
- `a` die untere Integrationsgrenze;
- `b` die obere Integrationsgrenze;
- `N` die Anzahl der Teilintervalle I_i, $i = 1, \ldots, N$ mit $h = \frac{b-a}{N}$ und

$$I_i := [a + (i-1) \cdot h, a + i \cdot h].$$

Auf jedem Teilintervall I_i soll dann die durch `w` und `p` spezifizierte Quadraturformel umgesetzt und somit eine Näherung für

$$\int_a^b f(x)\mathrm{d}x = \sum_{i=1}^{N} \int_{I_i} f(x)\mathrm{d}x$$

berechnet werden.

b) Testen Sie Ihre Implementierung für folgende Quadraturformeln:

- Trapezregel
- Simpson-Regel
- Newton's $\frac{3}{8}$-Regel

und für die numerische Integration der Runge-Funktion

$$f(x) := \frac{1}{1 + 25x^2}$$

aus Programmieraufgabe 2.1 auf dem Interval $[-1, 1]$. Erstellen Sie dazu einen aussagekräftigen Plot, welcher die drei Quadraturverfahren hinsichtlich des Fehlers gegenüber der exakten Lösung für $n = 1, 2, \ldots, 1000$ vergleicht. Tragen Sie dabei den Fehler logarithmisch auf (mit dem Matlab-Befehl `loglog`).
Speichern Sie Ihren Test in einem Skript `myQuadraturSum1DTest.m` und zusätzlich den durch das Skript erstellten Plot als `myQuadraturSum1DPlot.fig`.