

Abgabe der **Theorieaufgaben bis Donnerstag, 12.05.2016, 16:15 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 Donnerstag, 12.05.2016, 16:15 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 Koauthors/Koautorin nennen. Sie können die Aufgaben in MATLAB bzw. OCTAVE schreiben. Alternativ kann auch in C oder C++ abgegeben werden, allerdings ohne mögliche Hilfestellung. Es kann nur lauffähiger Code bewertet werden.

Bei Fragen zu Vorlesung und Übungen können Sie neben der Übungsgruppe auch das **Tutorium** nutzen, jeweils **Mittwochs von 13:15 bis 14:45** Uhr im CIP-Pool der Mathematik (M946).

Aufgabe 3.1 (Quadratur | 2 + 2 + 2 Punkte)

Berechnen Sie Näherungen und Fehlerschranken für das Integral

$$\int_{0.5}^1 \frac{1}{x^2} dx$$

mit der (einfachen)

- a) Trapezregel,
- b) Simpsonregel,
- c) $\frac{3}{8}$ -Regel.

Aufgabe 3.2 (Quadratur | 1 + 6 Punkte)

Zur numerischen Berechnung von

$$I(f) := \int_{-1}^1 f(x) dx$$

für eine stetige Funktion $f : [-1, 1] \rightarrow \mathbb{R}$ wird die folgende Formel vorgeschlagen:

$$I_h(f) := \frac{1}{9} \left(f(-1) + 8f\left(-\frac{1}{2}\right) + 8f\left(\frac{1}{2}\right) + f(1) \right).$$

- a) Finden Sie ein Polynom $p \in \Pi_4$ welches durch $I_h(p)$ nicht exakt integriert wird.
- b) Überprüfen Sie, dass alle Polynome dritten Grades exakt integriert werden.

Programmieraufgabe 3.1 (Quadratur | 3 + 3 Punkte)

- a) Implementieren Sie eine Funktion `[c]=myQuadratur1D(f,w,p)`, welche eine beliebige abgeschlossene Quadraturformel vom Grad R mit Quadraturgewichten $w \in \mathbb{R}^d$ und entsprechenden Quadraturpunkten $p \in \mathbb{R}^d$ umsetzt. Dabei soll die Funktion f , repräsentiert durch ein `function_handle` `f`, numerisch auf dem Intervall $[a, b] \subset \mathbb{R}$ integriert werden, wobei a dem ersten Quadraturpunkt und b dem letzten Quadraturpunkt entspricht.
- b) Schreiben Sie eine Routine `myQuadratur1DTest()`, in welcher Sie Ihre Implementierung aus Teil a) für die Funktion und Quadraturformeln aus Aufgabe 3.1 testen.

