MATHEMATISCHES INSTITUT
PROF. DR. CHRISTIANE HELZEL
DR. GEORG JANSING
PAWEL BUCHMÜLLER
MAXIMILIAN SCHNEIDERS
15. JUNI 2016

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

28	29	30	31	Σ

NAME: ____________
MAT-NR.: ____________
NAME: ____________
MAT-NR.: ____________
NAME: ____________
MAT-NR.: ____________
GRUPPE: ____________

Numerik I – 10. Übungsblatt

Aufgabe 28: (6 Punkte)

Bestimmen Sie für das gewichtete Integral

$$I(f) = \int_{-\infty}^{\infty} f(x) e^{-x^2} dx$$

die Gauß'sche Quadraturformel mit zwei Knoten.

Aufgabe 29: (3+3 Punkte)

Wir betrachten die Legendre-Polynome $P_n(x)$ aus der Vorlesung, d.h. die Polynome, die bzgl. des Skalarproduktes

$$\langle f, g \rangle = \int_{-1}^{1} f(x) g(x) \, \mathrm{d}x.$$

orthogonal stehen mit $P_0(x) = 1$, $P_1(x) = x$ und mit der Skalierung $P_n(1) = 1$.

(a) Zeigen Sie die Rodrigues-Formel

$$P_n(x) = \frac{1}{2^n \, n!} \cdot \frac{\mathrm{d}^n}{\mathrm{d}x^n} \left[(x^2 - 1)^n \right].$$

(b) Zeigen Sie für $n \geq 1$ die Rekusionsformel

$$(n+1) P_{n+1}(x) = (2n+1) x P_n(x) - n P_{n-1}(x).$$

Hinweis: Die Polynome p_n der Vorlesung, deren Nullstellen die Stützstellen der Gauß-Quadraturformeln sind, sind vielfache der Legendre-Polynome. Genauer gilt $p_n = 2^{n+1} n!/(2n)! \cdot P_n$.

Aufgabe 30: (1+2+2+1 Punkte)

(a) Zeigen Sie: Für $x \in \mathbb{R}^s$ und $1 \leq p < \infty$ gilt

$$\|x\|_\infty \leq \|x\|_p \leq s^{1/p} \|x\|_\infty.$$

(b) Für $A \in \mathbb{R}^{m \times n}$ gilt

$$\|A\|_\infty = \max_{i=1,\dots,m} \sum_{j=1}^{n} |a_{ij}|.$$

(c) Für $A \in \mathbb{R}^{m \times n}$ gilt

$$\|A\|_2 = \max_{\lambda \in \lambda(A^T A)} \sqrt{\lambda}, \qquad \lambda(B) := \{ \lambda \in \mathbb{C} \mid \exists \, y \in \mathbb{R}^s : By = \lambda y \} \text{ für } B \in \mathbb{R}^{s \times s}.$$

(d) Bestimmen Sie $\|A\|_1$, $\|A\|_2$ und $\|A\|_\infty$ für

$$A = \begin{bmatrix} 1 & 0 \\ -1 & 4 \\ 4 & 1 \end{bmatrix}.$$

Aufgabe 31: (6 Punkte, Zusatzaufgabe)

Leiten Sie mithilfe der 1D-Quadraturformel

$$\int_0^1 f(x)\,\mathrm{d}x \approx \sum_{i=0}^{n} a_i f(x_i)$$

die Approximation

$$\int_R g(x,y)\,\mathrm{d}(x,y) \approx |R| \sum_{i,j=0}^{n} a_i a_j g(\phi_1(x_i), \phi_2(x_j))$$

her, wobei

$$R = [b_1, c_1] \times [b_2, c_2], \qquad \phi_k = x \mapsto b_k + (c_k - b_k)x, \quad k = 1, 2$$

und $|R|$ den Flächeninhalt von R bezeichnet.

Programmieraufgabe 10:

(a) Schreiben Sie eine Funktion `[ A, X, Y ] = quad2d(At, Xt, b1, c1, b2, c2)`, die aus der 1D-Quadraturformel mit Stützstellen $\mathtt{Xt}(i)$ und Gewichten $\mathtt{At}(i)$ auf dem Grundintervall $[0,1]$ eine 2D-Quadraturformel auf dem Rechteck $R = [\mathtt{b1}, \mathtt{c1}] \times [\mathtt{b2}, \mathtt{c2}]$ wie in Aufgabe 31 aufstellt.

(b) Verwenden Sie die Funtkion `gauss.m` der Homepage (oder Ihre eigene aus Programmieraufgabe 9), die Funktion `[ A, X, Y ] = gauss2d(n, b1, c1, b2, c2)` zu schreiben, die ein Gauß-Quadraturformel mit $(\mathtt{n}+1)^2$ Stützstellen auf dem Rechteck R, wie oben, aufstellt.

Hinweis: Achten Sie darauf, dass die Funktion `gauss` die Quadraturformel auf dem Intervall $[-1, 1]$ aufstellt.

(c) Adaptieren Sie die Funktion `quadApply.m` von der Homepage(oder Ihre eigene Version aus Programmieraufgabe 9) so, dass sie 2D-Quadraturformeln auswerten kann. Die Adaptation soll die Signatur `v = quadApply2D(f, A, X, Y)` besitzen.

(d) Schreiben Sie ein Skript `p10`, welches Ihre Routinen testet: Verwenden Sie $f = (x,y) \mapsto \exp(x) \cdot \cos(y)$ und das Rechteck $R = [0, \frac{1}{2}] \times [-\frac{1}{2}, \frac{1}{2}]$.

Erstellen Sie logarithmische Plots des Fehlers für $n = 0, 1, 2, 3, 4, 5$ gegen den Parameter n. Plotten Sie dieselben Fehler in einem zweiten Bild auch gegen die Anzahl der Funktionsauswertungen.

Hinweis: Die Dateien zu dieser Programmieraufgabe werden ab dem kommenden Samstag auf die Numerik I-Homepage gestellt.

Abgabe der Übungsaufgaben am Mittwoch, 22. Juni zu Beginn der Vorlesung.
Abgabe der Programmierübungen per E-Mail bis Freitag, 24. Juni, 23:59 Uhr an `num1@hhu.de` **mit Betreff** `PA# Gr#`**, wobei # für die Nummer der Programmieraufgabe bzw. der Programmierübungsgruppe steht.**