MATHEMATISCHES INSTITUT
DER UNIVERSITÄT ZU KÖLN

Prof. Dr. A. Klawonn
M. Kühn, M. Sc.

Sommersemester 2017

29. Juni 2017

10. Übung zur Numerischen Mathematik

Hinweis: Schreiben Sie bitte auf **jedes Blatt ihren Namen**. Auf die erste Seite Ihrer Übung schreiben Sie bitte zusätzlich **Ihre Matrikelnummer** und **Ihre Übungsgruppe**.

Aufgabe 1: (8 Punkte)
Im Intervall $[a, b]$ seien $n+1$ äquidistante Stützstellen $x_i = a+ih$ mit $h = \frac{b-a}{n}$ für $i = 0, \dots, n$ gegeben. Die n-te Newton-Cotes-Formel ist derart konstruiert, dass diese für Polynome vom Grad $\leq n$ den exakten Integralwert liefert. Nutzen Sie dies als Voraussetzung und zeigen Sie, dass für gerades n sogar Polynome vom Grad $n + 1$ exakt integriert werden können.
Hinweis: Definieren Sie $P_{n+1} \in \mathcal{P}_{n+1}$ unter Verwendung von $w(x)$. Betrachten Sie $I(P_{n+1}) - I_h(P_{n+1})$ und danach $I(w)$ und $I_h(w)$ gesondert. Beweisen und benutzen Sie, dass das entsprechende $w(x)$ punktsymmetrisch zu $x + \frac{a+b}{2}$ ist. Betrachten Sie die Faktoren $x - x_{\frac{n}{2}-i}$ und $x - x_{\frac{n}{2}+i}$ in $w(x)$ jeweils paarweise. Die Verwendung von Satz 4.3.4 aus der Vorlesung stellt keine Lösung dieser Aufgabe dar.

Aufgabe 2: (1+4+4 Punkte)
Betrachten Sie das Integral

$$\int_0^1 \frac{1}{\sqrt{x}} \mathrm{d}x.$$

i) Berechnen Sie das Integral exakt.

Bestimmen Sie Näherungswerte für den Integralwert. Verwenden Sie dazu

ii) die zusammengesetzte Mittelpunktsregel (offene Newton-Cotes-Formel) für $n = 3$ und $n = 4$ Teilintervalle.

iii) die Gauß-Legendre-Integration für $n = 3$ Stützstellen.

Aufgabe 3: (3+3 Punkte)
Geben Sie die Fehlerdarstellung

i) der 3/8-Regel

ii) der Mittelpunktsregel (offene Newton-Cotes-Formel mit $n = 0$)

explizit, d. h. soweit wie möglich berechnet bzw. reduziert, an.

Programmieraufgabe: (15 Punkte)
Die Idee der adaptive numerischen Integration einer Funktion f über einem Intervall $[a, b] \subset \mathbb{R}$ besteht in der Berechnung zweier Approximationen $I_n(f)$ und $I_m(f)$ verschiedener Ordnungen n und m mit $n < m$, sodass man

$$\varepsilon := |I_m(f) - I_n(f)|$$

als Approximation an den Integrationsfehler $|I(f) - I_n(f)|$ verstehen kann. Überschreitet ε eine vorgegebene Toleranz **tol**, so wird das Intervall $[a, b]$ in zwei gleichgroße Intervalle aufgeteilt und die Methode jeweils rekursiv auf die beiden Teilintervalle angewendet usw.

Verwendet man für die numerische Integration I_n und I_m jeweils die Gauß-Quadratur ergibt sich in der Durchfürhung die Schwierigkeit, dass in der Regel keine Stützstellen übereinstimmen (ausgenommen sind hier ungerade n, m, bei denen jeweils die Null gemein ist). Dies hieße für jeden Schritt in dem eine Fehlerabschätzung in obigem Sinne durchgeführt wird m zusätzliche Funktionsaufrufe ausführen zu müssen.

Abhilfe schafft hierbei die Gauß-Kronrod-Quadratur bei der zur Näherung G_n mit Exaktheitsgrad $2n-1$ die Kronrod-Näherung K_{2n-1} mit Exaktheitsgrad $3n+1$ berechnet. Letztere verwendet dabei die Funktionsauswertungen aus der G_n-Approximation, sodass nur $n-1$ zusätzliche Funktionsauswertungen durchzuführen sind.

Implementieren Sie eine adaptive Gauß-Kronrod-Quadratur mit 3 Gauß-Knoten für Funktionen $f : \mathbb{R} \to \mathbb{R}$.
Hinweis: Bei der Gauß-Kronrod-Quadratur werden die Legendre-Polynome verwendet und mit $w \equiv 1$ gilt

$$\int_a^b f(x)\,\mathrm{d}x \approx \sum_{i=1}^{N} \alpha_i f(a_i),$$

wobei α_i die Gewichte und a_i die Knoten bezüglich $[a, b]$ sind.

Gehen Sie dazu wie folgt vor:

- Laden Sie sich die Matlab-Dateien `adaptive_quadrature.m` und `test_quadrature.m` von unserer Veranstaltungs-Internetseite
  `http://www.numerik.uni-koeln.de/15701.html`
  herunter.
  - Setzen Sie sich mit dem objektorientierten Ansatz in `adaptive_quadrature.m` auseinander.
  - Verwenden Sie `test_quadrature.m` als Startdatei und erzeugen Sie mit entsprechenden Übergabeparametern ein Objekt *adaptQuadObj* der Klasse `adaptive_quadrature`.
- Verwenden Sie die obigen Vorgaben und implementieren Sie das Verfahren durch eine rekursive Methode `gausskronrod37(adaptQuadObj, a, b, aktRekTiefe)`.
- Verwenden Sie $\varphi(t) = \frac{b-a}{2}t + \frac{a+b}{2}$ um $[-1, 1]$ auf $[a, b]$ abzubilden.
- Berechnen Sie mithilfe der Gauß-Knoten in $[-1, 1]$
  - $g_1 = 0.77459666924148337703585307995648 0$
  - $g_2 = 0.00000000000000000000000000000000$
  - $g_3 = -0.77459666924148337703585307995648 0$

  und der zugehörigen Gewichte
  - $\alpha_{g_1} = 0.55555555555555555555555555555556$
  - $\alpha_{g_2} = 0.88888888888888888888888888888889$
  - $\alpha_{g_3} = 0.55555555555555555555555555555556$

  eine Gauß-Legendre-Approximation G_3 an den Integralwert.

- Berechnen Sie mit den gegebenen Gauß- und den dazu bestimmten Kronrod-Knoten in $[-1, 1]$
  - $k_1 = 0.96049126870802028342350709262908 0$
  - $k_3 = 0.43424374934680255800207150284462 8$
  - $k_5 = -0.43424374934680255800207150284462 8$
  - $k_7 = -0.96049126870802028342350709262908 0$

  mit $k_{2i} = g_i$ für $i = 1, 2, 3$ sowie den Kronrod-Gewichten
  - $\alpha_{k_1} = 0.10465622602646726519382385719207 3$
  - $\alpha_{k_2} = 0.26848808986833344072856928066671 0$
  - $\alpha_{k_3} = 0.40139741477596222290505181861843 2$
  - $\alpha_{k_4} = 0.45091653865847414234511008704557 1$
  - $\alpha_{k_5} = 0.40139741477596222290505181861843 2$
  - $\alpha_{k_6} = 0.26848808986833344072856928066671 0$
  - $\alpha_{k_7} = 0.10465622602646726519382385719207 3$

  die verbesserte Approximation K_7.
- Vergleichen Sie den *Fehler* $|K_7 - G_3|$ mit einer vorgegebenen Fehlertoleranz. Wird diese nicht eingehalten, teilen Sie das Intervall in zwei gleich große Teilintervalle auf und führen die beschriebene Vorgehensweise für jedes Teilintervall getrennt durch. Speichern Sie anderenfalls das Teilergebnis in *adaptQuadObj.integ_approx* und *adaptQuadObj.error*.
- Verwenden Sie eine maximale Rekursionstiefe von 10 und **tol** $= 1e - 12$. Betrachten Sie das Integral aus Aufgabe 2. Setzen Sie dies in Ihrer Startdatei um.
- Geben Sie die Integralapproximation und den geschätzten Fehler aus.

Abgabe des Programmierteils

- Den Code und das ausführbare Programm schicken Sie bitte an Ihren Übungsleiter / Ihre Übungsleiterin mit Betreff der Form: `Uebung1, Nachname, Vorname`. **Unkommentierter Programmcode wird nicht angenommen bzw. die Aufgabe wird als unbearbeitet gewertet!**
- Packen Sie Ihre Dateien in ein Archiv (Formate: .zip, oder .tar.gz) mit Dateinamen `ueb01_vorname_nachname.zip`
- Geben Sie bitte immer eine **ausgedruckte Version** Ihrer Programmcodes mit den schriftlichen Aufgaben ab ($\rightarrow$ Kasten), falls dies in der Aufgabenstellung nicht eindeutig anders vermerkt wurde.

Abgabedatum der Aufgaben 1-3: 06. Juli 2017 bis 14:00 Uhr im entsprechenden Kasten in Raum 3.01 des Mathematischen Instituts.
Abgabedatum der Programmieraufgabe: 10. Juli 2017 bis 10:00 Uhr im entsprechenden Kasten in Raum 3.01 des Mathematischen Instituts.