Technische Universität München
Institut für Informatik
Prof. Dr. Thomas Huckle
Michael Lieb, M. Sc.
Dipl.-Tech. Math. Stefanie Schraufstetter

SS 2010

Numerisches Programmieren

3. Programmieraufgabe: Quadratur

Einfache Quadraturformeln

Bei den einfachen Quadratur-Regeln wird das Integrationsgebiet nicht zerlegt, die Regeln werden einmal auf das gesamte Integrationsgebiet angewandt. Im Rahmen dieser Programmieraufgabe sollen folgende einfache Regeln zur Integration der Funktion f im Integrationsgebiet $[a, b]$ implementiert werden:

- *Trapezregel*: Bei der Trapezregel wird der Integrand mit der linearen Funktion durch $(a, f(a))$ und $(b, f(b))$ interpoliert. Die berechnete Fläche entspricht der Fläche unter dem Interpolanten, also dem Trapezes mit der Breite $H = b - a$ und den Höhen $f(a)$ (linke Seite des Trapezes) bzw. $f(b)$ (rechte Seite des Trapezes).
- *Keplersche Fassregel*: Die Keplersche Fassregel berechnet die Fläche unter dem quadratischen Interpolanten,wobei die Stützstellen die Ränder und die Mitte des Integrationsgebiets sind.

Quantile und Konfidenzintervalle

Beim Messen physikalischer Größen oder beim Prüfen von Stichproben sind Fehler nie zu vermeiden. An Stelle eines exakten Wertes fordert man also als Meßergebnis ein Konfidenzintervall $[a, b]$, das so gewählt sein soll, dass es bei mindestens $\alpha * 100\%$ aller Messungen den wahren Wert enthält. Diese Forderung lässt sich meist zu $\alpha \leq P(X \in [a, b]) = \int_a^b f(t)\, dt$ umformen, wobei X eine Zufallsvariable mit einer Dichte f ist. Ziel dieser Aufgabe ist es also, $[a, b]$ so zu bestimmen, dass $\alpha = \int_a^b f(t)\, dt$.

Wir nehmen an, dass die Dichte f symmetrisch und somit ein Konfidenzintervall der Form $[-a, a]$ gesucht ist. Gesucht ist dann ein a mit $\int_{-a}^{a} f(t)\, dt = \alpha$ bzw. $\int_0^a f(t)\, dt = \alpha/2$ (vgl. Abb 1).

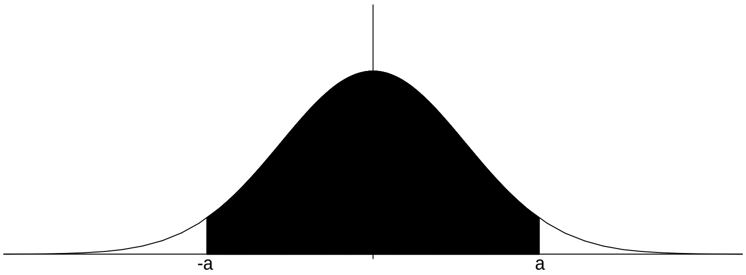

Abb. 1: Der Inhalt der gefärbten Fläche soll α sein

Quadratur mit fester Schrittweite

Ein möglicher Ansatz lehnt sich an die Idee der Summenformel an. Das Gebiet $[0, \infty]$ wird dazu in kleine Teilintervalle der Länge h aufgeteilt. Von 0 ausgehend wird die Funktion dann mit Hilfe einer einfachen Quadraturformel - in dieser Programmieraufgabe verwenden wir die Keplersche Fassregel - solange auf Teilintervallen integriert, bis das Gesamtintegral $\alpha/2$ übersteigt (Oder gleich $\alpha/2$ ist). Falls dazu n Teilintervalle nötig waren, ist $a \approx nh$ eine Näherung für das gesuchte a.

Achten Sie bei der Implementierung darauf, dass die Stützstellen an den Rändern der Teilintervalle (ausser denen am Rand des gesamten Integrationsgebiets) zu zwei Teilintervallen gehören. Bei einer effizienten Implementierung darf die Funktion aber an jeder Stützstelle nur ein Mal ausgewertet werden (d.h. bei der Fassregel $2n + 1$ Auswertungen für n Teilintervalle)

Adaptive Quadratur

Bei der Quadratur mit fester Schrittweite h hängt die Qualität der Näherung natürlich von h ab. Es ist aber nicht genau bekannt, wie klein h gewählt werden muss, um eine geforderte Toleranz TOL einzuhalten. Das Vorantasten in kleine Schritten führt zudem zu einem hohen Rechenaufwand. Es bietet sich deshalb an, eine adaptive Quadratur durchzuführen.

In dieser Programmieraufgabe soll die adaptive Quadratur mit Trapezsumme verwendet werden. Dabei wird das zu integrierende Intervall in zwei Teilintervalle aufgeteilt und die Trapezsumme wird sowohl auf das Gesamtintervall als auch auf die beiden Teilintervalle angewendet. Die Differenz zwischen dem Wert für das Gesamtintervall und der Summe der Werte für die Teilintervalle dient als Schätzung für den Fehler (s. Abb 2). Falls eine Verfeinerung nötig ist, wird das Verfahren rekursiv auf beiden Teilintervallen fortgesetzt, andernfalls wird die Summe der Integrale der Teilintervalle als Ergebnis zurückgegeben.

Hier soll für einen vorgegebenen Parameter $l > 0$ zunächst das Intervall $[0, l]$ integriert werden, wobei folgende zwei Fälle zu einer Verfeinerung führen:

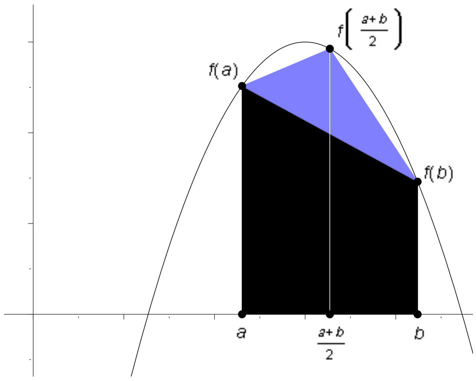

Abb. 2: Adaptive Quadratur: Das Dreieck schätzt den Fehler

- Der geschätzte relative Fehler $\left|\frac{Dreieck}{Trapez+Dreieck}\right|$ übersteigt $TOL/4$. [1]
- Das Integral bis einschließlich des aktuellen Teilintervalls übersteigt $\alpha/2 + TOL/4$.

Außerdem soll die adaptive Quadratur abbrechen, sobald das Integral über die bisher ausgewerteten Teilintervalle im Bereich $[\alpha/2 - TOL/4, \alpha/2 + TOL/4]$ liegt (dazu ist es natürlich nötig die Teilintervalle strikt von links nach rechts abzuarbeiten, wir verwenden also Tiefensuche). Die Rückgabe besteht dann aus dem Integralwert und dem Punkt, bis zu dem integriert wurde.

Falls der zurückgegebene Integralwert $\alpha/2 - TOL/4$ unterschreitet, wird auf $[l, 2l]$ weiterintegriert, danach ggf. auf $[2l, 3l]$ usw.

Auch in dieser Aufgabe darf die Funktion an keiner Stelle mehrmals ausgewertet werden.

Aufgaben

Alle zu implementierenden Methoden befinden sich in der Klasse `Quadrature`.

- Implementieren Sie für die einfachen Quadraturregeln die Methoden `trapezoid` und `kepler`.
- Implementieren Sie in `quantileFixedStep` die Quantilsuche mit fester Schrittweite.
- Implementieren Sie in `quantileAdapt` die adaptive Quantilsuche. Sie dürfen dazu eine Hilfsfunktion einführen.
- Testen Sie die Verfahren z.B. mit der Normalverteilung (Klasse `NormalDist`).

[1] der zulässige Gesamtfehler TOL wird auf $[-\infty, 0]$ und $[0, \infty]$, sowie auf die Ungenauigkeit beim Berechnen des Integrals und das Finden des richtigen Integrationsbereiches aufgeteilt

Formalien und Hinweise

- Das Programmgerüst erhalten Sie auf den Webseiten zur Vorlesung.
- Ergänzen Sie das Programmgerüst bitte **nur an den dafür vorgegebenen Stellen!** Falls Sie die Struktur der Programme an anderer Stelle verändern, können wir sie evtl. nicht mehr testen.
- Beseitigen Sie vor Abgabe Ihres Programms alle Ausgaben an die Konsole und reichen sie bis zum **21. Juni 2010, 12:00 Uhr** über das Web-Portal (siehe Homepage) ein.
- Wir empfehlen, die Programme unter Linux (Rechnerhalle) zu testen.