D-ITET
Dr. R. Käppeli
L. Rueff

Serie 4

1. *Qualitatives Kennenlernen adaptiver Quadratur*

In dieser Aufgabe wollen wir uns mit adaptiver Quadratur vertraut machen. Benutzen Sie die MATLAB Funktion `adaptsim.m` um folgende Funktionen Integrale zu berechnen:

(i) $\int_{3/2}^{4} f_1(x)dx$ mit $f_1(x) = \frac{1}{2x^3 - x^2}\left(5\sin\left(\frac{20}{x}\right)\right)^2$

(ii) $\int_{3/2}^{4} f_2(x)dx$ mit $f_2(x) = \min\left(f_1(x), \frac{1}{2}\right)$

(iii) $\int_{-5}^{5} f_3(x)dx$ mit $f_3(x) = \frac{1}{1+x^2}$

(iv) $\int_{0}^{1} f_4(x)dx$ mit $f_4(x) = \sqrt{x}$

(v) $\int_{0}^{1} f_5(x)dx$ mit $f_5(x) = \sin(4\pi x)e^{-2x}$

(vi) $\int_{0}^{0.6} f_5(x)dx + \int_{0.6}^{1} f_5(x)dx$

(vii) $\int_{0}^{+\infty} \exp(-x^4)\, dx$
Hinweis: Verwenden Sie eine Substitution aus Aufgabe 3 der Serie 2.

Für (i)-(vii) plotten Sie die Funktion und das adaptive Quadratur Gitter. Verwenden Sie als Toleranz `tol=1e-4` und für die maximal Anzahl Verfeinerungen `maxlevel=12`.

Was geht schief bei (v) und warum klappt es bei (vi)?
Hinweis: Zu Verfügung steht im File `adaptive_quadrature_example.m` ein Beispiel für die Verwendung der Funktion `adaptsim`.

2. *Adaptive Quadratur mit der Trapezregel*

In dieser Aufgabe wollen wir eine adaptive Quadratur Methode zur Berechnung des bestimmten Integrals

$$I[f] = \int_a^b f(x)dx$$

entwickeln und implementieren basierend auf der Trapezregel. Wie in der Vorlesung diskutiert, benötigt man dafür einen Fehler-Schätzer. Hierzu vergleichen wir das Resultat der Trapezregel

$$Q_1[f] = \frac{b-a}{2}\,(f(a) + f(b))$$

mit dem Resultat der zusammengesetzten Trapezregel

$$Q_1^2[f] = \frac{b-a}{4}\left(f(a) + 2f\left(\frac{a+b}{2}\right) + f(b)\right)$$

(mit zwei Teil-Intervallen).

a) Bestimmen Sie den Fehler-Schätzer für $E^2[f] = |Q_1^2[f] - I[f]|$.
Hinweis : Beispiel (14) und (15) in der Vorlesung.

b) Implementieren Sie die adaptive Quadratur Methode in der MATLAB Funktion `adapttrapez_simple_Template.m`.
Hinweis : Verwenden Sie den in der Vorlesung gezeigten Pseudo-MATLAB Code.

c) Der in der Vorlesung gezeigte Pseudo-MATLAB Code ist sehr simple und besitzt einige Schwächen. Geben Sie zwei offensichtliche Schwächen an und versuchen Sie diese zu beheben.
Hinweis : Die `adaptsimp.m` Funktion von Aufgabe 2 könnte hilfreich sein.

3. *Homogen geladenes Quadrat in kartesischen Koordinaten*

Betrachten Sie ein quadratisches Gebiet in der x-y-Ebene welches eine konstante elektrische Ladungsdichte ϱ_0 aufweist

$$\varrho(x,y) = \begin{cases} \varrho_0, & (x,y) \in [-1,1]^2 \\ 0, & \text{sonst.} \end{cases}$$

Das elektrostatische Potential ϕ an einem Punkt (x_p, y_p) ausserhalb des geladenen Quadrats ist dann durch Integration Über die geladene Region gegeben

$$\phi(x_p, y_p) = \frac{\varrho_0}{4\pi\epsilon_0} \int_{-1}^{+1} \int_{-1}^{+1} \frac{1}{\sqrt{(x-x_p)^2 + (y-y_p)^2}}\, dx\, dy\,.$$

Der Einfachheit halber setzen wir $\frac{\varrho_0}{4\pi\epsilon_0} = 1$.

Implementieren Sie die zusammengesetzte Trapezregel in zwei Dimensionen und berechnen Sie dann $\phi(x_p, y_p)$ für $x_p = y_p = 2, 10, 20$. Verwenden Sie $N = 128$ Teil-Intervalle für beide Dimensionen und vergleichen Sie Ihre Werte mit den exakten:

$$\begin{aligned} \phi(2,2) &= 1.4493948762686699 \\ \phi(10,10) &= 0.2830800703857426 \\ \phi(20,20) &= 0.1414508706242226. \end{aligned}$$

Siehe nächstes Blatt!

Hinweis: Verwenden Sie die Templates `potential_Template.m` und `trapez2D_Template.m`.

4. *Richardson-Extrapolation*

Die sog. Richardson-Extrapolation nutzt die asymptotische Fehler-Entwicklung einer numerischen Methode um den dominierenden Fehler-Term zu eliminieren. Damit kann man numerische Verfahren höherer Ordnung aus Verfahren niedriger Ordnung generieren.

Es bezeichne $A(h)$ eine numerische Approximation einer Grösse A^* mit einer Fehler-Entwicklung der Form

$$A(h) = A^* + a_1 h^{k_1} + \mathcal{O}(h^{k_2}). \tag{1}$$

Hier ist h ein Diskretisierungs-Parameter (z.B. die Intervallsbreite oder die Distanz zwischen Gitterpunkten). Nun berechnet man die numerische Approximation mit halbiertem Diskretisierungs-Parameter

$$A(h/2) = A^* + a_1 \left(\frac{h}{2}\right)^{k_1} + \mathcal{O}(h^{k_2}). \tag{2}$$

Multiplizieren wir die Gl. (2) mit 2^{k_1} und ziehen das Resultat von Gl. (1) ab, erhalten wir

$$A(h) - 2^{k_1} A(h/2) = (1 - 2^{k_1})A^* + \mathcal{O}(h^{k_2}).$$

D.h. der dominierende Fehler-Term wurde eliminiert! Teilen wir nun noch durch $(1 - 2^{k_1})$ erhalten wir

$$\tilde{A}(h) = \frac{A(h) - 2^{k_1} A(h/2)}{1 - 2^{k_1}} = A^* + \mathcal{O}(h^{k_2}), \tag{3}$$

wobei wir die Bezeichnung $\tilde{A}(h)$ für die neue Approximation von A^* eingeführt haben. Aus Gl. (3) erkennen wir, dass wir durch geschickte Kombination der numerischen Approximationen $A(h)$ und $A(h/2)$ der Ordnung k_1 die Approximation $\tilde{A}(h)$ der Ordnung k_2 erhalten haben. Dies ist das Prinzip der Richardson-Extrapolation.

Nun soll dieses Prinzip auf numerische Differentiation und Quadratur angewendet werden und die resultierende höhere Ordnung experimentell bestimmt werden:

a) Wenden Sie die Richardson-Extrapolation auf die zentrierten finiten Differenzen an,

$$A(h) = \frac{f(x+h) - f(x-h)}{2h} \approx A^* = \frac{\mathrm{d}f}{\mathrm{d}x},$$

um den dominierenden Fehler-Term zu eliminieren.
Hinweis: Sie haben die Fehler-Entwicklung von zentrierten finiten Differenzen in Aufgabe 4 der Serie 2 untersucht.

Bitte wenden!

b) Wenden Sie die Richardson-Extrapolation auf die summierte Trapez-Regel an,

$$A(h) = Q_1^N[f] = \frac{h}{2}\left(f(a) + 2\sum_{j=1}^{N-1} f(a+jh) + f(b)\right) \approx A^* = \int_a^b f(x)\mathrm{d}x,$$

um den dominierenden Fehler-Term zu eliminieren. Hier ist $h = (b-a)/N$ und N die Anzahl Teilintervalle.

c) Bestimmen Sie in beiden Fällen experimentell die höhere Ordnung von $\tilde{A}(h)$. *Hinweis*: Arbeiten Sie im Template `richardson.m`.

Abgabe: Online bis **Freitag** den 24.03.2023 unter `sam-up.math.ethz.ch`.