1 Adaptive numerische Integration

Literatur:
H.R. Schwarz, N. Köckler: *Numerische Mathematik*, Teubner Stuttgart, 5. überarbeitete Auflage, 2004.
W. Gander: *Computer Mathematik*, Birkhäuser Basel, 1985.

Rombergverfahren

Grosser Rechenaufwand durch *uniforme* Verteilung der Stützstellen. Der Aufwand ist direkt proportional zur Anzahl Funktionsauswertungen.

Problem der Effizienz

Mit möglichst wenig Funktionsauswertungen soll ein möglichst genauer numerischer Wert für

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

bestimmt werden.

Reduktion des Aufwandes

Wahl der Stützstellen wird dem individuellen Integranden angepasst. Dabei soll diese Wahl automatisch auf Grund von bestimmten Kriterien erfolgen.

Beispiel 1

Betrachten wir

$$I = \int_0^1 \sqrt{x}\,dx = \frac{2}{3}$$

Auf den ersten Blick ist bei diesem „harmlosen“ Beispiel nicht ersichtlich, was hier die Adaptivität zu suchen hat.
$f'(x) = \frac{1}{2\sqrt{x}}$, d.h. $|f'(0)| = \infty$ ($f'(x)$ hat in $x = 0$ einen Pol) und die zweite Ableitung $f''(x) = -\frac{1}{4}\,\frac{1}{\sqrt{x^3}}$ ist für $x = 0$ erst recht nicht definiert, m.a.W. $M = \infty$ für die Fehlerabschätzung der Trapezwertmethode.

Diese Tatsache schlägt sich bei der Integration mit der Trapezmethode wie folgt nieder (fortgesetzte Halbierung): sei z.B. die Toleranz $\varepsilon = 10^{-5}$

h	$T(h)$	h	$T(h)$
1	0.5	2^{-6}	0.666270811
2^{-1}	0.603553391	2^{-7}	0.666525657
2^{-2}	0.643283046	2^{-8}	0.666616549
2^{-3}	0.658130222	2^{-9}	0.666648882
2^{-4}	0.663581197	2^{-10}	0.666660362
2^{-5}	0.665558936		

Benötigt wird also $h = 2^{-10} \approx 10^{-3}$, um diese Genauigkeit zu erreichen.
Abschätzung des Integrationsfehlers für das $2-$te Teilintervall $[10^{-3}, 2\cdot 10^{-3}]$:

$$|I - T(h)| = \frac{b-a}{12}\,h^2\,|f''(\xi)| \le \frac{b-a}{12}\,h^2\,M = \frac{1}{48}\cdot 10^{-\frac{9}{2}} \approx 6.588\cdot 10^{-7} \quad \text{wobei} \quad M = \frac{1}{4}\,10^{\frac{9}{2}}$$

D.h. bereits für das $2-$te Teilintervall ist die Schrittweite viel zu *klein*!
Noch krasser wird es beim letzten Teilintervall $[1 - 10^{-3}, 1]$:

$$|I - T(h)| = \frac{h^3}{12}\,|f''(\xi)| \le \frac{10^{-9}}{12}\,M \approx 2.01\cdot 10^{-11} \quad \text{wobei} \quad M = 0.25037\ldots$$

Folgerung

Die Schrittweite h der Methode wird bei diesem Beispiel nur durch das Verhalten von f im *ersten* Teilintervall $[0, h]$ bestimmt.

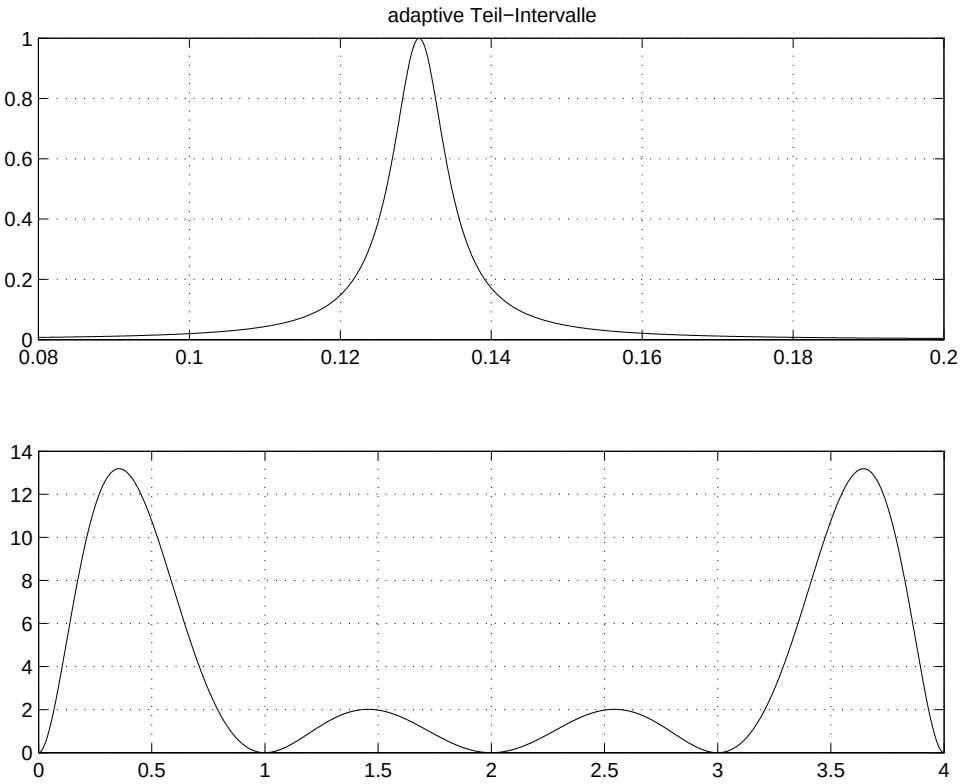

Abbildung 1: Funktionsverläufe, die eine adaptive Steuerung verlangen.

Bemerkung 1

- I. allg. kann ein Integral über ein kurzes Intervall genauer und schneller berechnet werden, als ein entsprechendes Integral über ein langes Intervall.
- Es ist von Vorteil, das gegebene Intervall $[a, b]$ vor der Anwendung einer Quadraturformel *geeignet* zu unterteilen.
- Anschliessend wird die Quadraturformel auf die einzelnen Teilintervalle angewendet (mit Hilfe einer linearen Transformation).
- Summation der einzelnen Teilintegrale.
- *adaptive* Verfahren unterteilen $[a, b]$ fortgesetzt so lange, bis in jedem Teilintervall mit der verwendeten Quadraturformel die verlangte Genauigkeit erreicht ist.
- Die Unterteilung wird dort feiner, wo $f(x)$ stark variiert und dort gröber, wo $f(x)$ schwach variiert.
- Die Entscheidung, ob ein Teilintervall weiter unterteilt werden soll, erfolgt auf Grund des Vergleichs zweier verschiedener Näherungswerte $\tilde{I}_1$ und $\tilde{I}_2$ für dasselbe Teilintervall.

Prinzip

Teilintervall $I_j = [a_j, b_j] \subset [a, b]$
Sei z.B. $\tilde{I}_1 =$ der Trapezwert für I_j und $\tilde{I}_2 =$ der Simpsonwert für I_j, d.h.

$$\tilde{I}_1 = \frac{1}{2}\, h_j \; (f(a_j) + f(b_j)) \qquad h_j = b_j - a_j$$

und

$$\tilde{I}_2 = \frac{1}{3}\left(\tilde{I}_1 + 2h_j f(m_j)\right) \qquad m_j = \frac{a_j + b_j}{2}$$

Die berechneten Funktionswerte lassen sich so ökonomisch weiter verwenden. Mit *einer* zusätzlichen Funktionsauswertung $f(m_j)$ bekommt man $\tilde{I}_2$.

Abbruchkriterium für die lokale Intervall-Halbierung

Die fortgesetzte Halbierung auf einem Teilintervall wird dann abgebrochen, falls gilt (cf. Gander):

$$\tilde{I}_1 + I_s = \tilde{I}_2 + I_s \quad \text{in Maschinenarithmatik}$$

Dabei ist $I_s \neq 0$ ein Schätzwert für das gesuchte Integral $I = \int_a^b f(x)\,dx$. I_s baucht *kein* guter Näherungswert für $I \neq 0$ zu sein, sondern muss nur mit der Grössenordnung von I übereinstimmen.

Bemerkung 2

- Dieses Kriterium bewirkt, dass betragsmässig kleine Beiträge zu I nicht mit zu kleinem relativem Fehler berechnet werden, cf. Beispiel 1.
- Für I wird ein Näherungswert mit praktisch voller Rechnergenauigkeit geliefert.
- Ist man für $I \neq 0$ mit der relativen Genauigkeit $\varepsilon > 0$ zufrieden, dann wird dies mit

  $$I_s \approx \varepsilon \cdot \frac{I}{\delta}$$

  erreicht. Dabei ist $\delta =$ kleinste positive Zahl des Rechners, d.h. $(1 + \delta \neq 1)$.
- Die kompakteste und eleganteste algorithmische Beschreibung kann mit *rekursiver* Programmierung realisiert werden, cf. Gander. Nachteile: Speicherbedarf, Geschwindigkeit

Algorithmus ohne Rekursion

- Speicherung der Teilpunkte a_j und der zugehörigen Funktionswerte $f_j := f(a_j)$ in entsprechenden Vektoren a bzw. f, um sie wieder verwenden zu können.
- Zudem wird die Information über diejenigen Teilintervalle, über welche die Integrale noch zu berechnen sind, benötigt. Dazu wird ein Indexvektor u verwendet, der die Indices p der laufend generierten Teilpunkte a_p enthält. Mit diesen Teilpunkten können die Integrationsintervalle erklärt werden.
- Die Zahl der Komponenten von u variiert im Laufe des folgenden Algorithmus, während jene der Vektoren a und f monoton zunimmt.

1.1 Adaptive Integration (iterativ)

```
Start: a_0 = a, a_1 = b, f_0 = f(a), f_1 = f(b), I = 0
       j = 0, k = 1, p = 1, l = 1, u_1 = 1

       Halb: h = a_k - a_j, m = (a_j + a_k)/2, f_m = f(m)
             I_1 = h(f_j + f_k)/2, I_2 = (I_1 + 2 h f_m)/3
             falls I_s + I_1 <> I_s + I_2
                   p = p + 1, a_p = m, f_p = f_m, k= p
                   l = l + 1, u_l = p
                   gehe nach Halb
             sonst
                   I = I + I_2, j = u_l, l = l - 1, k = u_l
             falls l > 0
                   gehe nach Halb
```

Bemerkung 3

- Es gibt keinen absolut sicheren, auf deterministische Weise arbeitenden automatischen Quadratur-Algorithmus.
- Falls ein solches hypothetisches Verfahren auf die Funktion $f(x) = 0$ für $[a, b]$ mit $a < b$ angewendet wird, so erhalten wir nach einer gewissen Anzahl N von Funktionsauswertungen $f(x_j)$ an den Stützstellen x_j das richtige Resultat $\tilde{I} = 0$.
- Für die Funktion
$$g(x) := \prod_{j=1}^{N} (x - x_j)^2$$
die $g(x_j) = 0$, $j = 1, 2, \ldots, N$, erfüllt, bildet der Algorithmus die genau gleichen Stützstellen x_j und somit erhalten wir ebenfalls $\tilde{I} = 0$, obwohl $I = \int_a^b g(x)\,dx > 0$ ist!
- Ebenso ist es nicht leicht, einen praktikablen, universellen Quadratur-Algorithmus anzugeben, der z.B.
$$f(x) = \begin{cases} 1000 & 0.334 < x < 0.335 \\ 0 & \text{sonst} \end{cases}$$
für das Intervall $[a, b] = [-1, 1]$ korrekt integriert.

Bemerkung 4 *Die adaptive Quadratur ist natürlich effizienter, falls einfache Quadraturformeln (cf. Gauss–Quadratur) mit höheren Fehlerordnungen kombiniert werden.*

Beispiel 2

Betrachten wir das folgende Beispiel
$$I = \int_0^1 x^{\frac{3}{2}}\,dx = 0.4$$
Scheinbar ist auch dieser Integrand harmlos.

Die Anwendung des Schemas von Romberg zeigt folgendes:

h_k	$R_{k,0}$	$R_{k,1}$	$R_{k,2}$	$R_{k,3}$	$R_{k,4}$
1	0.5000000000				
0.5	0.4267766953	0.4023689271			
0.25	0.4070181109	0.4004319161	0.4003027820		
0.125	0.4018124648	0.4000772494	0.4000536050	0.4000496498	
0.0625	0.4004634013	0.4000137135	0.4000094777	0.4000087773	0.4000086170

Aus dem Schema ist ersichtlich, dass nur die Kolonne $R_{k,1}$ gegenüber $R_{k,0}$ eine wesentliche Verbesserung bringt. Der Grund für dieses Verhalten ist die Singularität der zweiten Ableitung $f''(x) = \frac{3}{4}\,x^{-\frac{1}{2}}$ des Integranden $f(x) = x^{\frac{3}{2}}$ an der Stelle $x = 0$.

Offensichtlich ist auch dieses Beispiel *nicht* ganz harmlos, cf. im Vergleich dazu das Beispiel 1.

Wird nun mit dem Algorithmus 1.1 gerechnet, so erhalten wir bei $14-$stelliger Rechnung mit $\delta = 10^{-14}$ und dem Schätzwert $I_s = 0.5$ für I folgende Resultate:

ε	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
$\tilde{I}$	0.4000004636	0.4000000214	0.4000000029	0.4000000005	0.4000000001
N	43	85	207	387	905

Dabei ist N die Anzahl der Auswertungen des Integranden $f(x)$.