Numerisches Programmieren
Nachtrag zur Tutorübung vom 19.12.2011
Julius Adorf

1 Romberg-Quadratur

In der Tutorstunde am 19.12.2011 kamen Fragen zu der Romberg-Quadratur und zu dem zugehörigen Algorithmus auf. Dieser Nachtrag ist als kleine Hilfestellung gedacht, um den Zusammenhang zwischen der mathematischen und der ausprogrammierten Formulierung des Algorithmus zur Romberg-Quadratur besser herauszustellen.

1.1 Ziel der Romberg-Quadratur

Die Romberg-Quadratur gehört zu den Extrapolationsverfahren. Wie in der Aufgabe 7.1. beschrieben, beginnen wir mit der Berechnung von Trapezsummen $T(h_1), T(h_2), ..., T(h_n)$ zu verschiedenen Schrittweiten $h_1, h_2, ..., h_n$ (absteigend nach Größe geordnet). Dann definieren wir Stützpunkte

$$(h_1^2, T(h_1)), (h_2^2, T(h_2)), ..., (h_n^2, T(h_n)$$

Wir legen dann ein Interpolationspolynom wie aus Aufgabenblatt 6 gewohnt durch diese Stützpunkte. Schließlich extrapolieren wir den Wert an der Stelle $h = 0$, er gibt uns eine Näherung für die Fläche und ist das Ergebnis der Romberg-Quadratur (der Begriff der Extrapolation deutet an, dass wir das Interpolationspolynom außerhalb des Bereichs der Stützstellen auswerten). Der intuitive Gedanke hinter dieser Methode ist, dass wir bei der Ermittlung der exakten Fläche unter dem Integranden mit einer Trapezsumme umso näherkommen, je kleiner die Schrittweite wird. Da wir auf dem Computer aber nicht beliebig kleine Schrittweiten wählen können, versuchen wir den Wert bei einer infitesimal kleinen Schrittweite durch Extrapolation zu schätzen. Die Herleitung über die Euler-MacLaurin-Formel sagt uns wie und warum das auch funktioniert, und dass man dabei die Stützstellen als Quadrate der Schrittweite wählen muss.

1.2 Formel zur Romberg-Quadratur

Die Formel der Romberg-Quadratur erhalten wir über die MacLaurin-Formel. Oder aber, sofern wir wissen, wie die Extrapolation vonstatten geht und wie wir die Stützpunkte dafür wählen müssen, erhalten wir die Formel durch

lineare Interpolation. So kommt man dann auch auf die in der Musterlösung angegebene Formel:

$$I(f) \approx T(h_2) + \frac{T(h_2) - T(h_1)}{(h_1^2/h_2^2) - 1} \tag{1}$$

die besagt, wie man aus zwei Trapezsummen die Fläche durch Extrapolation an der Stelle $h = 0$ annähert. Warum hat das was mit der Formel für linearen Interpolation zu tun? Schreibt man (1) um

$$I(f) \approx T(h_2) + \frac{T(h_2) - T(h_1)}{(h_1^2/h_2^2) - 1} = T(h_2) - \frac{T(h_1) - T(h_2)}{h_1^2 - h_2^2} h_2^2 \tag{2}$$

dann steht hier die Formel für lineare Interpolation, für die Auswertung einer Interpolationsgeraden (Extrapolation) zu den beiden Stützpunkten $(h_1^2, T(h_1)), (h_2^2, T(h_2^2))$ (wobei, wie gesagt, $h_2 < h_1$) an der Stelle $h = 0$.

1.3 Algorithmus zur Romberg-Quadratur

Haben wir mehr als zwei Schrittweiten gewählt, so können wir das Gesamtergebnis nach dem Neville-Schema (siehe Aufgabenblatt 3) berechnen. Hierzu sollten wir uns erinnern, wie der Neville-Algorithmus arbeitet: er verküpft die Teilergebnisse von jeweils zwei Interpolationsgeraden zu einem neuen Wert an der Stelle, an der ausgewertet werden soll. Ähnlich funktioniert die Romberg-Quadratur. Wir wollen hier ein Dreiecksschema aufstellen und veranschaulichen, wie der Algorithmus das Schema befüllt.

Aus der Abbildung und dem Algorithmus erkennen wir, dass zu einem gegebenen Index k der Wert $T[k]$ sich aus dem alten $T[k]$ und dem alten $T[k+1]$ berechnet. Ein Umschreiben der Formel aus dem Algorithmus in mathematisch anmutendere Notation ergibt:

$$T[k] \longleftarrow T[k+1] + \frac{T[k+1] - T[k]}{\frac{(h[k])^2}{(h[i])^2} - 1} \tag{3}$$

Wir vergleichen nun mit (2). Im ersten Schritt ist klar, was passiert: wir extrapolieren. In der Abbildung sehen wir folgendes: Im Schritt $i = 3, k = 1$ werden mit Vorschrift (3) zwei Näherungen $T[1], T[2]$ verknüpft, welche ursprünglich aus je zwei Trapezsummen unterschiedlicher Schrittweite berechnet wurden. Hier sei wieder auf das Aufgabenblatt 3 verwiesen, wo der gleiche Mechanismus im Neville-Algorithmus greift.

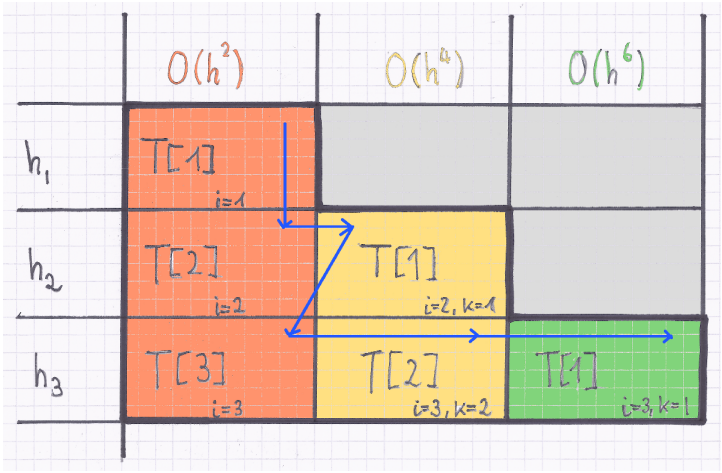

Abbildung 1: Dreiecksschema für die Romberg-Quadratur. Die einzelnen Zellen (6 insgesamt) korrespondieren zu den Einträgen eines Arrays T der Länge 3. Die rot hinterlegten Zellen enthalten Näherungen $T[k]$, die aus einer Trapezsumme gewonnen werden; die gelb und grün hinterlegten Zellen entsprechend aus zwei bzw. drei Trapezsummen. Der blaue Pfad beschreibt die Reihenfolge, in der der Algorithmus die Zellen beschreibt.

```
1    for i=1:n
2        waehle n[i];
3        h[i]:=(b-a)/n[i];
4        T[i]:=Trapezsumme zur Schrittweite h[i]
5        for k:=i-1:-1:1
6            T[k] := T[k+1] + 1/(h[k]^2/h[i]^2-1)*(T[k+1]-T[k])
7        end
8    end
```