Übungsaufgaben zur Vorlesung

Computerphysik

apl. Prof. Dr. R. Bulla

SS 2018

Blatt 4: Abgabetermin: Montag, der 14.05.2018, 12:00

Aufgabe 1: Integration – Trapez-Regel

(10 Punkte)

a) Schreiben Sie eine Funktion `int_trapez(a,b,N)`, die das Integral

$$I = \int_0^\pi \sin(x)\,\mathrm{d}x$$

mit Hilfe der Trapez-Regel berechnet. Wie im Vorlesungsskript ist dabei N die Zahl der Stützstellen mit $a = x_1 = 0$ und $b = x_N = \pi$, die Zahl der Intervalle ist also $N - 1$. Berechnen Sie damit die Abweichung vom exakten Ergebnis

$$\Delta I = |I_{\mathrm{Trapez}} - I_{\mathrm{exakt}}|$$

für $N = 100$. (3 Punkte)

b) Wie in der Vorlesung gezeigt, gilt für den Fehler ΔI:

$$\Delta I \propto h^2 \ , \quad \text{mit} \ \ h = \frac{b-a}{N-1} \ ,$$

die Abhängigkeit $\Delta I(h)$ wird also durch ein Potenzgesetz beschrieben. Um generell zu prüfen, ob für eine Funktion $g(h)$ gilt: $g(h) \propto h^\alpha$, trägt man $\ln(g)$ gegen $\ln(h)$ auf; ein Potenzgesetz zeigt sich in dieser doppel-logarithmischen Auftragung als Gerade. Erstellen Sie ein Diagramm, in dem $\ln(\Delta I)$ gegen $\ln(h)$ aufgetragen ist (erweitern Sie dazu das Programm aus Teilaufgabe a) entsprechend). Wählen Sie die Zahl der Stützstellen im Bereich $N = 10$ bis $N = 2000$. (4 Punkte)

c) Unter der Annahme, dass ein Potenzgesetz vorliegt, lässt sich der Exponent direkt bestimmen, indem man die Ableitung

$$\frac{\mathrm{d}\ln g}{\mathrm{d}y} \ , \quad \text{mit} \ \ y = \ln h$$

bildet. Berechnen Sie damit den Exponenten α (verwenden Sie dazu den Vorwärtsdifferenzenquotienten). (3 Punkte)

Aufgabe 2: Integration – Simpson-Regel

(5 Punkte)

a) Schreiben Sie eine Funktion `int_simpson(a,b,N)`, die das Integral aus Aufgabe 1 mit Hilfe der Simpson-Regel berechnet. Hinweis: die Zahl der Stützstellen N muss ungerade sein. (3 Punkte)

b) Mit dieser Funktion und dem Programm aus Aufgabe 1c) lässt sich nun der Exponent α für den Fehler ($\Delta I(h) \propto h^{\alpha}$) bestimmen. Welcher Wert ergibt sich für α? (2 Punkte)

Aufgabe 3: Faktorisierung

(12 Punkte)

Mit dem folgenden Algorithmus wird eine ganze Zahl M in zwei Faktoren zerlegt, z.B. $15 \to 3 \cdot 5$; ist die Eingabe eine Primzahl, so ergibt sich z.B. $17 \to 1 \cdot 17$.

Zunächst wird berechnet

- $f_1 = \lfloor \sqrt{M} \rfloor$;
- $f_2 = f_1$.

Solange $f_1 > 0$ wird der folgende Block ausgeführt

- falls $(f_1 \cdot f_2) = M \longrightarrow$ Faktorisierung gefunden; die while-Schleife wird mit `break` beendet;
- falls $(f_1 \cdot f_2) > M \longrightarrow f_1$ wird um 1 verringert;
- falls $(f_1 \cdot f_2) < M \longrightarrow f_2$ wird um 1 erhöht.

Für $M = 731$ liefert der Algorithmus die folgende Sequenz an Werten für f_1 und f_2:

27 27; 27 28; 26 28; 26 29; 25 29; 25 30; 24 30; 24 31; 23 31; 23 32; ... 18 39; 18 40; 18 41; 17 41; 17 42; 17 43;

a) Schreiben Sie eine Funktion `M2f1f2(M)`, die die beiden Faktoren f_1 und f_2 als Tupel ausgibt (`return (f1,f2)`). Zerlegen Sie damit die in dem Feld `M_values = [1500320,87065078,87069473925]` gespeicherten Zahlen in Produkte aus je zwei ganzen Zahlen. (6 Punkte)

b) Dieser Algorithmus zerlegt die Zahl M in zwei ganze Zahlen, die nicht notwendigerweise Primzahlen sind. Schreiben Sie ein Programm, das mit Hilfe der Funktion `M2f1f2(M)` die Zahl $M = 870037764750$ in *alle* Primfaktoren zerlegt. Die Ausgabe soll folgende Form haben (hier für $M = 20$):
`M = 2*2*5`, d.h. die Primfaktoren sollen aufsteigend sortiert sein. (6 Punkte)

Aufgabe 4: Summen und Produkte II

(2 Punkte)

Berechnen Sie die Summen und Produkte aus Aufgabe 1 von Blatt 1 mit Hilfe der Funktionen `sum` und `prod`.