Dirk Praetorius
Christoph Überhuber

Wintersemester 2007/08
31. Oktober 2007

Übungen zur Vorlesung
Einführung in das Programmieren für TM

Serie 4

Die Aufgaben mit Stern (*) sind bis zur nächsten Übung vorzubereiten und werden dort abgeprüft. Kopieren Sie bitte den Source-Code in ein Unterverzeichnis `serie04` Ihres Home-Verzeichnisses. Überprüfen Sie bitte vor der Übung, ob Ihre Source-Codes mit dem `gcc` compiliert werden können. In den folgenden Aufgaben üben Sie ***Bedingungsschleifen****.*

Aufgabe 31*. Schreiben Sie eine Funktion `scanfpositive`, die vom Benutzer die Eingabe einer positiven Zahl $\tau > 0$ verlangt und diese dann zurückgibt. Die Eingabe soll solange wiederholt werden, bis die eingegebene Zahl $\tau \in \mathbb{R}$ strikt positiv ist, d.h. bei Eingabe einer Zahl $\tau \leq 0$ wird der Benutzer zu erneuter Eingabe aufgefordert. Speichern Sie den Source-Code unter `scanfpositive.c` ins Verzeichnis `serie04`.

Aufgabe 32*. Die Cosinus-Funktion hat die Darstellung $\cos(x) = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k}}{(2k)!}$. Wir betrachten die Partialsummen

$$C_n(x) = \sum_{k=0}^{n} (-1)^k \frac{x^{2k}}{(2k)!}.$$

Man schreibe eine Funktion `cos_`, die für gegebene $x \in \mathbb{R}$ und $\tau > 0$ den Wert $C_n(x)$ zurückliefert, sobald

$$|C_n(x) - C_{n-1}(x)|/|C_n(x)| \leq \tau \quad \text{oder} \quad |C_n(x)| \leq \tau$$

gilt. Man schreibe ferner ein aufrufendes Hauptprogramm, in dem $x \in \mathbb{R}$ und $\tau > 0$ eingelesen werden. Neben dem berechneten Wert $C_n(x)$ sollen auch der korrekte Wert $\cos(x)$ und der absolute Fehler $|C_n(x) - \cos(x)|$ ausgegeben werden sowie der relative Fehler $|C_n(x) - \cos(x)|/|\cos(x)|$ im Fall $\cos(x) \neq 0$. Man schreibe die Funktion möglichst so, dass diese mit einer Schleife auskommt und dass x^{2k} und $(2k)!$ möglichst kostensparend realisiert werden. Man vermeide also insbesondere (vor- oder selbst implementierte) Funktionen zur Berechnung der Potenz oder der Faktoriellen. Speichern Sie den Source-Code unter `cos.c` ins Verzeichnis `serie04`.

Aufgabe 33*. Gegeben sei eine stetige Funktion $f : [a, b] \to \mathbb{R}$, d.h. eine Funktion ohne Sprünge. Es gelte

$$f(a) \cdot f(b) \leq 0.$$

Dann hat f eine Nullstelle z_0, die im Folgenden mittels Bisektion (= Intervallhalbierung) approximiert werden soll: Der Bisektionsalgorithmus arbeitet wie folgt: In jedem Bisektionsschritt definiert man $c := (a + b)/2$ als Intervallmittelpunkt. Aufgrund der Voraussetzung gilt

$$f(a) \cdot f(c) \leq 0 \quad \text{oder} \quad f(c) \cdot f(b) \leq 0.$$

Im Fall $f(a) \cdot f(c) \leq 0$ liegt eine Nullstelle im Intervall $[a, c]$, und man ersetzt daher b durch c. Im Fall $f(c) \cdot f(b) \leq 0$ liegt eine Nullstelle im Intervall $[c, b]$, und man ersetzt daher a durch c. In beiden Fällen geht man also vom Intervall $[a, b]$ zu einem Teilintervall halber Länge über. — Machen Sie sich das Verfahren zunächst anhand des Beispiels $f : [0, 3] \to \mathbb{R}$, $f(x) = x - 1$ klar. — Schreiben Sie eine Funktion `bisection`, die als Parameter a, b und eine Toleranz $\tau > 0$ übernimmt und den Bisektionsschritt wiederholt, bis man vom Startintervall $[a, b]$ zu einem Intervall $[a, b]$ mit Länge $|b - a| \leq \tau$ übergegangen ist. In diesem Fall gebe man a zurück. Es gilt dann $|a - z_0| \leq |a - b| \leq \tau$, d.h. a ist eine Approximation einer Nullstelle z_0 bis auf eine Genauigkeit von τ. Als Testfunktion verwende man $f(x) = x^2 + \exp(x) - 2$ auf $[0, \infty)$, die man als eigene Funktion realisiere. Man schreibe ferner ein Hauptprogramm, das $b, \tau > 0$ einliest und die Approximation von z_0 ausgibt. Speichern Sie den Source-Code unter `bisection.c` ins Verzeichnis `serie04`.

Aufgabe 34. Man schreibe eine Funktion `power`, die für gegebene reelle Zahlen $x > 1$ und $C > 0$ die kleinste Zahl $n \in \mathbb{N}$ berechnet mit $x^n > C$. Schreiben Sie ferner ein aufrufendes Hauptprogramm, in dem x und C eingelesen werden und n ausgegeben wird.

Aufgabe 35. Man schreibe eine Funktion `wurzelschranke`, die zu einer gegebenen Zahl $x \geq 0$ die natürliche Zahl $k \in \mathbb{N}_0$ mit $k \leq \sqrt{x} < k + 1$ zurückgibt. Dabei dürfen weder die Wurzel-Funktion `sqrt`, noch Rundungsoperationen (z.B. `floor` oder `ceil` etc.) verwendet werden. Schreiben Sie ein aufrufendes Hauptprogramm, in dem x eingelesen und k ausgeben wird.

Aufgabe 36. Für $x > 0$ konvergiert die Folge

$$x_1 := \frac{1}{2}(1+x), \quad x_{n+1} := \frac{1}{2}\Big(x_n + \frac{x}{x_n}\Big) \quad \text{für } n \geq 1$$

gegen $\sqrt{x}$. Man schreibe eine Funktion `sqrt_`, die für gegebene $x > 0$ und $\tau > 0$ als Ergebnis das erste Folgenglied $y = x_n$ zurückgibt, für das gilt

$$\frac{|x_n - x_{n+1}|}{|x_n|} \leq \tau \quad \text{oder} \quad |x_n| \leq \tau.$$

Schreiben Sie ferner ein aufrufendes Hauptprogramm, in dem x eingelesen und neben der Approximation x_n von $\sqrt{x}$ auch der exakte Wert sowie der absolute Fehler $|x_n - \sqrt{x}|$ ausgegeben werden. Welcher Zusammenhang besteht mit dem Newton-Verfahren aus Aufgabe 39?

Aufgabe 37. Schreiben Sie eine Funktion `sqrt2_`, die für gegebene $x > 0$ und $\tau > 0$ die Wurzel $\sqrt{x}$ mit Hilfe des Bisektionsverfahrens aus Aufgabe 33 approximiert, sodass der approximative Wert a die Fehlerschranke $|a - \sqrt{x}| \leq \tau$ erfüllt: Welche Funktion $f : [0, \infty) \to \mathbb{R}$ wählt man? Wie wählt man das Startintervall $[a, b]$? Vergleichen Sie die Anzahl der Iterationen mit der Anzahl der Iterationen in Aufgabe 36.

Aufgabe 38. Alternativ zum Bisektionsverfahren aus Aufgabe 33 kann eine Nullstelle von $f : [a, b] \to \mathbb{R}$ auch mit dem *Sekantenverfahren* berechnet werden. Dabei sind x_0 und x_1 gegebene Startwerte und man definiert induktiv x_{n+1} als Nullstelle der Geraden durch $(x_{n-1}, f(x_{n-1}))$ und $(x_n, f(x_n))$, d.h.

$$x_{n+1} := x_n - f(x_n)\,\frac{x_{n-1} - x_n}{f(x_{n-1}) - f(x_n)}$$

Schreiben Sie eine Funktion `sekante(x0,x1,tau)` die die Folge der Iterierten berechnet, bis entweder

$$|f(x_n) - f(x_{n-1})| \leq \tau$$

oder

$$|f(x_n)| \leq \tau \quad \text{und} \quad |x_n - x_{n-1}| \leq \begin{cases} \tau & \text{für } |x_n| \leq \tau, \\ \tau|x_n| & \text{sonst} \end{cases}$$

gilt. Es werde dann x_n als Approximation einer Nullstelle z_0 von f zurückgegeben. Im ersten Fall gebe man zusätzlich eine Warnung aus, dass das numerische Ergebnis vermutlich falsch ist. Verwenden Sie das Beispiel aus Aufgabe 33 zum Test. Schreiben Sie ein aufrufendes Hauptprogramm, in dem x_0 und x_1 eingelesen werden und x_n ausgegeben wird.

Aufgabe 39. Eine weitere Variante zur Berechnung einer Nullstelle einer Funktion $f : [a, b] \to \mathbb{R}$ ist das *Newton-Verfahren*. Ausgehend von einem Startwert x_0 definiert man induktiv eine Folge (x_n) wie folgt: Zu gegebenem x_k sei x_{k+1} die Nullstelle der Tangente an den Graphen von f im Punkt $(x_k, f(x_k))$, d.h. $x = x_{k+1}$ erfüllt $0 = f(x_k) + f'(x_k)(x - x_k)$. Auflösen nach x zeigt

$$x_{k+1} = x_k - f(x_k)/f'(x_k).$$

Man realisiere das Newton-Verfahren in einer Funktion `newton(x0,tau)`, wobei die Iteration abgebrochen wird, falls entweder

$$|f'(x_n)| \leq \tau$$

oder

$$|f(x_n)| \leq \tau \quad \text{und} \quad |x_n - x_{n-1}| \leq \begin{cases} \tau & \text{für } |x_n| \leq \tau, \\ \tau|x_n| & \text{sonst} \end{cases}$$

gilt. Im ersten Fall gebe man zusätzlich eine Warnung aus, dass das numerische Ergebnis vermutlich falsch ist. Verwenden Sie das Beispiel aus Aufgabe 33 zum Test. Schreiben Sie ein aufrufendes Hauptprogramm, in dem x_0 eingelesen und x_n ausgegeben wird.

Aufgabe 40. Die Quotientenfolge $(a_{n+1}/a_n)_{n\in\mathbb{N}}$ zur Fibonacci-Folge $(a_n)_{n\in\mathbb{N}}$,

$$a_0 := 1, \quad a_1 := 1, \quad a_n := a_{n-1} + a_{n-2} \quad \text{für } n \geq 2,$$

konvergiert gegen den goldenen Schnitt $(1 + \sqrt{5})/2$. Insbesondere konvergiert die Differenz

$$b_n := \frac{a_{n+1}}{a_n} - \frac{a_n}{a_{n-1}}$$

gegen Null. Man schreibe eine Funktion `cauchy`, die zu gegebenem $k \in \mathbb{N}$ die kleinste Zahl $n \in \mathbb{N}$ mit $|b_n| \leq 1/k$ zurückgibt.