Dirk Praetorius
Christoph Überhuber

Wintersemester 2006/07
24. November 2006

Übungen zur Vorlesung
Einführung in das Programmieren für TM

Serie 6

Die Aufgaben mit Stern (*) sind bis zur nächsten Übung vorzubereiten und werden dort abgeprüft. Die übrigen Aufgaben dienen nur Ihrer Übung und mir als zusätzliche Grundlage für den Prüfungsstoff in den schriftlichen Tests. Kopieren Sie bitte den Source-Code in ein Unterverzeichnis `serie06` *Ihres Home-Verzeichnisses. Überprüfen Sie bitte vor der Übung, ob Ihre Source-Codes mit dem* `gcc` *compiliert werden können. In den folgenden Aufgaben sollen im wesentlichen* ***Bedingungs- und Zählschleifen*** *geübt werden.*

Aufgabe 49*. 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. Speichern Sie den Source-Code unter `wurzelschranke.c` ins Verzeichnis `serie06`.

Aufgabe 50*. Die Funktion $f(x) = x^2 + \exp(x) - 2$ hat eine Nullstelle bei $x \approx 1/2$. Die Existenz dieser Nullstelle folgt aus dem Zwischenwertsatz, denn $f(0) = -1$ und $f(1) = e-1 > 0$. Diese Nullstelle soll numerisch berechnet werden mittels *Newton-Verfahren*: Ausgehend von einem Startwert $x_0 \in \mathbb{R}$ 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`, wobei die Iteration abgebrochen wird, falls entweder $|f(x)| \leq \varepsilon_1$ oder $|x_n - x_{n+1}| \leq \varepsilon_2$ gilt. In diesem Fall werde $x = x_{n+1}$ als Approximation der gesuchten Nullstelle zurückgegeben. Dabei sind $\varepsilon_1, \varepsilon_2 > 0$ gegebene Parameter. Die Daten für das Newton-Verfahren sollen in einer Struktur `function` abgelegt werden, die neben den Funktionspointern für f und f' den Definitionsbereich von f enthält. Speichern Sie den Source-Code unter `newton.c` ins Verzeichnis `serie06`.

Aufgabe 51. 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 $\varepsilon > 0$ als Ergebnis das erste Folgenglied $y = x_n$ zurückgibt, für das gilt

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

Welcher Zusammenhang besteht mit dem Newton-Verfahren aus Aufgabe 50?

Aufgabe 52. 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.

Aufgabe 53. Für eine differenzierbare Funktion $f : [a,b] \to \mathbb{R}$ kann man die Ableitung $f'(x)$ in einem festen Punkt $x \in \mathbb{R}$ durch den einseitigen Differenzenquotienten

$$\Phi(h) := \frac{f(x+h) - f(x)}{h} \quad \text{für } h > 0$$

approximieren. Nach Taylor-Formel gilt für $f \in C^2(\mathbb{R})$ die Fehlerabschätzung $|f'(x) - \Phi(h)| = \mathcal{O}(h)$. Man schreibe eine Funktion `diff`, die für $h_n := 2^{-n}h_0$ die Folge der $\Phi(h_n)$ berechnet, bis gilt

$$|\Phi(h_n) - \Phi(h_{n+1})| \leq \begin{cases} \varepsilon & \text{für } |\Phi(h_n)| \leq \varepsilon, \\ \varepsilon\,|\Phi(h_n)| & \text{anderenfalls.} \end{cases}$$

Die Funktion liefere in diesem Fall $y = \Phi(h_n)$ zurück. Die Funktion f ist mittels einer geeigneten Struktur zu realisieren. Ferner werden der Funktion `diff` die Schrittweite $h_0 > 0$, der Auswertungspunkt x und die Toleranz $\varepsilon > 0$ übergeben.

Aufgabe 54. Die Sinus-Funktion hat die Reihendarstellung

$$\sin(x) = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k+1}}{(2k+1)!}.$$

Wir betrachten die Partialsummen

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

Man schreibe eine C-Funktion `sin_`, die für gegebene $x \in \mathbb{R}$ und $\varepsilon > 0$ den Wert $S_n(x)$ zurückliefert, sobald

$$|S_n(x) - S_{n-1}(x)|/|S_n(x)| \leq \varepsilon \quad \text{oder} \quad |S_n(x)| \leq \varepsilon$$

gilt. Man schreibe ferner ein aufrufendes Hauptprogramm, in dem $x \in \mathbb{R}$ und $\varepsilon > 0$ eingelesen werden. Neben dem berechneten Wert $S_n(x)$ sollen auch der korrekte Wert $\sin(x)$ und der absolute Fehler $|S_n(x) - \sin(x)|$ ausgegeben werden sowie der relative Fehler $|S_n(x) - \sin(x)|/|\sin(x)|$ im Fall $\sin(x) \neq 0$.

Aufgabe 55. Man schreibe die Funktion in Aufgabe 54 so, dass diese mit einer Schleife auskommt und dass x^{2k+1} und $(2k+1)!$ möglichst kostensparend realisiert werden. Man verwende also insbesondere keine (vor- oder selbst implementierten) Funktionen zur Berechnung der Potenz oder der Faktoriellen.

Aufgabe 56. Man schreibe eine C-Funktion `kalender(jahr1)`, die das nächste Jahr `jahr2` > `jahr1` berechnet, an dem man einen Kalender von Jahr `jahr1` vollständig wiederverwenden kann, d.h. die Wochentage der Jahre `jahr1` und `jahr2` stimmen vollständig überein.