Dirk Praetorius
Christoph Überhuber Sommersemester 07
23. Mai 2007

Übungen zur Vorlesung
Einführung in das Programmieren für TM

Serie 9

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 `serie08` *Ihres Home-Verzeichnisses. Überprüfen Sie bitte vor der Übung, ob Ihre Source-Codes mit* `matlab` *auf der* `cad.zserv.tuwien.ac.at` *interpretiert werden können. In den folgenden Aufgaben sollen im wesentlichen* ***Bedingungsschleifen*** *geübt werden.*

Aufgabe 81*. Der Funktionspointer (sog. *function handle*) einer Funktion `fct` ist in Matlab `@fct`. Er kann mit Hilfe des Befehls `feval` ausgewertet werden: Enthält die Variable `pointer` beispielsweise den Funktionspointer der Funktion `sin`, so sind

`y = feval(pointer,x)` und `y = sin(x)`

äquivalent. Man schreibe eine nicht-rekursive Funktion `bisektion(f,a,b,tau)`, die das Bisektionsverfahren aus Aufgabe 17 realisiert und eine Nullstelle z der Funktion $f : [a,b] \to \mathbb{R}$ approximiert, d.h. die Approximation $\widetilde{z}$ erfülle $|z - \widetilde{z}| \le \tau$. Im Fall $f(a) \cdot f(b) > 0$ werde mit Fehlermeldung abgebrochen, siehe `help error`. — Neben $\widetilde{z}$ soll die Funktion die Folge der Iterierten (z.B. die Folge der linken Intervallgrenzen) sowie die Folge der dazugehörigen Funktionswerte zurückgeben.

Aufgabe 82*. Alternativ zum Bisektionsverfahren 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.

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

Verifizieren Sie diese Folge, und schreiben Sie eine Funktion `sekante(f,x0,x1,tau)` die die Folge der Iterierten berechnet, bis entweder

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

oder

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

gilt. In zweiten Fall werde $\widetilde{z} := x_n$ als Approximation der Nullstelle x_0 von f zurückgegeben. Im ersten Fall werde mit Fehlermeldung abgebrochen. — Neben x_n sollen zusätzlich die Folgen $(x_0, \dots, x_n)$ der Nullstellen und der dazugehörigen Funktionswerte zurückgegeben werden.

Aufgabe 83*. 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(f,fprime,x0,tau)`, wobei die Iteration abgebrochen wird, falls entweder

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

oder

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

gilt. In zweiten Fall werde $\widetilde{z} := x_n$ als Approximation der gesuchten Nullstelle zurückgegeben. Im ersten Fall werde mit Fehlermeldung abgebrochen. — Neben x_n sollen zusätzlich die Folgen $(x_0, \dots, x_n)$ der Nullstellen und der dazugehörigen Funktionswerte zurückgegeben werden.

Aufgabe 84*. Schreiben Sie eine Matlab-Prozedur `testsqrt`, die für gegebenes $x > 0$ und $\tau > 0$ die Wurzel $\sqrt{x}$ mittels der Funktionen aus Aufgabe 81–83 approximiert, d.h. die positive Nullstelle z der Funktion $f(t) := t^2 - x$ numerisch berechnet. Die Prozedur gebe für alle drei Verfahren den relativen Fehler $|\sqrt{x} - z|/\sqrt{x}$ sowie die Anzahl der benötigten Iterationsschritte aus.

Aufgabe 85. 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 83?

Aufgabe 86. 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 $|f'(x) - \Phi(h)| = \mathcal{O}(h)$. Beweisen Sie diese Aussage mathematisch! Schreiben Sie eine Funktion `diff(f,x,h0,tau)`, die für $h_n := 2^{-n} h_0$ die Folge der $\Phi(h_n)$ berechnet, bis gilt

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

Die Funktion liefere in diesem Fall $y = \Phi(h_n)$ zurück. — Ferner soll die vollständige Folge $(\Phi(h_0), \dots, \Phi(h_n))$ der Iterierten zurückgegeben werden.

Aufgabe 87. Das Newton-Verfahren aus Aufgabe 83 benötigt neben der Funktion `f` auch eine Funktion `fprime`, die die Ableitung f' der Funktion f auswertet. Alternativ kann man $f'(x_k)$ durch den Differenzenquotienten $\Phi_h(x_k)$ ersetzen. Man realisiere dieses Vorgehen, z.B. für $h = \tau$. Man erweitere Aufgabe 84 um dieses Verfahren und vergleiche mit den anderen.

Aufgabe 88. 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 89. Man schreibe eine 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.

Aufgabe 90. Für einen stetigen Integranden $f : [a, b] \to \mathbb{R}$ berechnet man das Integral $I := \int_a^b f\,dx$ numerisch über geeignete Summen. Bei der *summierten Trapezregel* berechnet man für gegebenes $n \in \mathbb{N}$ und $h := (b-a)/n$ zum Beispiel

$$I_n := \frac{h}{2}\Big(f(a) + 2\sum_{j=1}^{n-1} f(a+jh) + f(b)\Big). \tag{1}$$

Dies ist gerade das Integral über die stetige und stückweise affine Funktion p mit $p(a+jh) = f(a+jh)$. Man schreibe eine Funktion `trapezregel(f,a,b,tau)`, die die Folge der Approximationen I_n berechnet, bis gilt

$$|I_n - I_{n-1}| \leq \begin{cases} \tau & \text{für } |I_n| \leq \tau, \\ \tau\,|I_n| & \text{anderenfalls.} \end{cases}$$

Der Wert I_n werde dann als Approximation von $\int_a^b f\,dx$ zurückgegeben. Ferner gebe man die vollständige Folge $(I_1, \dots, I_n)$ der Approximationen zurück. Man teste die numerische Integration am Beispiel $f(x) = \exp(x)$ auf dem Intervall $[0, 10]$ und gebe abhängig von n den Fehler $|I - I_n|$ tabellarisch aus.