Samuel Ferraz-Leite, Dirk Praetorius
Christoph Überhuber

Sommersemester 2008
16. April 2008

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. 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 `serie05` *Ihres Home-Verzeichnisses. In den folgenden Aufgaben sollen* ***Bedingungsschleifen*** *und* ***dynamische Matrizen*** *geübt werden.*

Aufgabe 31*. Eine 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

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

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. Im ersten Fall gebe man zusätzlich eine Warnung aus, dass das numerische Ergebnis vermutlich falsch ist. Als Testfunktion verwende man $f(x) = x^2 + \exp(x) - 2$ auf $[0, \infty)$, die man als eigene Funktion realisiere. Schreiben Sie ein aufrufendes Hauptprogramm, in dem x_0, τ eingelesen und x_n ausgegeben werden.

Aufgabe 32*. Für $x > 0$ konvergiert die Folge

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

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

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

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 31?

Aufgabe 33*. Man schreibe eine Funktion `transpose`, die zu einer dynamisch gespeicherten Matrix $A \in \mathbb{R}^{m \times n}$ die transponierte Matrix $A^T \in \mathbb{R}^{n \times m}$ berechnet. Dabei sind die Einträge von A^T gerade durch $(A^T)_{jk} = A_{kj}$ definiert. Im Hauptprogramm sollen die Dimensionen $m, n \in \mathbb{N}$ sowie die Einträge der Matrix A eingelesen und A^T ausgegeben werden.

Aufgabe 34*. Man schreibe eine Funktion `matrixvectorU`, die das Matrix-Vektor-Produkt $y = Ux$ mit einer oberen Dreiecksmatrix $U \in \mathbb{R}^{n\times n}$ mittels geeigneter Schleifen berechnet. Dabei bezeichnet man eine Matrix

$$U = \begin{pmatrix} u_{11} & u_{12} & u_{13} & \dots & u_{1n} \\ & u_{22} & u_{23} & \dots & u_{2n} \\ & & u_{33} & \dots & u_{3n} \\ & & & \ddots & \vdots \\ \mathbf{0} & & & & u_{nn} \end{pmatrix}$$

als obere Dreiecksmatrix. Mathematisch formuliert gilt also $U_{jk} = 0$ für $j > k$. Bei der Berechnung soll auf die offensichtlichen Nulleinträge von U nicht zugegriffen werden, um den Rechenaufwand gering zu halten. Schreiben Sie ferner ein Hauptprogramm in dem die Dimension $n \in \mathbb{N}$ sowie die Einträge der Matrix U und die Koeffizienten des Vektors x eingelesen und Ux ausgegeben werden.

Aufgabe 35. 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.

Aufgabe 36. 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 $|C_n(x) - \cos(x)|$ sowie der relative Fehler $|C_n(x) - \cos(x)|/|\cos(x)|$ im Fall $\cos(x) \neq 0$ ausgegeben werden. 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.

Aufgabe 37. 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 38. Schreiben Sie Funktionen `allocLMatrix`, `freeLMatrix` und `reallocLMatrix`, die untere Dreiecksmatrizen, d.h. $L \in \mathbb{R}^{n\times n}$ mit $L_{ij} = 0$ für $j > i$, möglichst speicherarm anlegen und initialisieren, reallokieren bzw. freigeben. Testen Sie Ihre Funktionen mithilfe eines geeigneten Hauptprogrammes.

Aufgabe 39. Schreiben Sie eine Funktion, die überprüft, ob eine gegeben Matrix $L \in \mathbb{R}^{n\times n}$ untere Dreiecksgestalt hat, d.h. $L_{ij} = 0$ für $j > i$. Ist dies der Fall, so soll die Matrix geeignet reallokiert werden, sodass keine triavialen Nulleinträge gespeichert werden.

Aufgabe 40. Man schreibe eine Funktion `matrixvectorL`, die das Matrix-Vektor-Produkt $y = Lx$ mit einer unteren Dreiecksmatrix $L \in \mathbb{R}^{n\times n}$ mittels geeigneter Schleifen berechnet. Vergleichen Sie den Rechenaufwand einer naiven Implmenetierung, bei der die Struktur von L keine Berücksichtigung findet, mit dem Aufwand einer effizienten Implementierung. Testen Sie den effizienten Fall mithilfe der Funktionen zur Speicherverwaltung aus Aufgabe 38 Schreiben Sie ferner ein Hauptprogramm in dem die Dimension $n \in \mathbb{N}$ sowie die Einträge der Matrix L und die Koeffizienten des Vektors x eingelesen und Lx ausgegeben werden.