Dirk Praetorius,
Michele Ruggeri

Sommersemester 2017
02.05.2017

Übungen zur Vorlesung
Einführung in das Programmieren für TM

Serie 7

Aufgabe 7.1. 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. Schreiben Sie eine Funktion `double* diff(double x, double h0, double tau, int* n)`, die für $h_n := 2^{-n} h_0$ die Folge der $\Phi(h_n)$ berechnet, bis gilt

$$|\Phi(h_n) - \Phi(h_{n+1})| \le \begin{cases} \tau & \text{falls } |\Phi(h_n)| \le \tau, \textit{oder} \\ \tau\,|\Phi(h_n)| & \text{anderenfalls.} \end{cases}$$

Die Funktion liefere in diesem Fall die vollständige Folge $(\Phi(h_0), \dots, \Phi(h_n))$ der Iterierten zurück. Beachten Sie, dass auch die Länge des Vektors „zurückgegeben“ werden muss. Die Funktion soll mit einer beliebigen reellwertigen Funktion `double f(double x)` arbeiten. Schreiben Sie ein Main-Programm, in dem Sie Ihre Funktion eingehend testen. Speichern Sie den Source-Code unter `diff.c` in das Verzeichnis `serie07`.

Aufgabe 7.2. Für einen stetigen Integranden $f : [a, b] \to \mathbb{R}$ berechnet man das Integral $I := \int_a^b f\,dx$ numerisch über geeignete Summen (Quadraturformeln). 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 $p(a+jh) = f(a+jh)$. Schreiben Sie eine Funktion `double* trapezregel(double a, double b, double tau, int* n)`, die die Folge der Approximationen I_n berechnet, bis gilt

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

Die Funktion gebe die vollständige Folge $(I_1, \dots, I_n)$ der Approximationen zurück. Beachten Sie, dass auch die Länge des Vektors „zurückgegeben“ werden muss. Die Funktion soll mit einer beliebigen reellwertigen Funktion `double f(double x)` arbeiten. Man teste die numerische Integration am Beispiel $f(x) = \exp(x)$ auf dem Intervall $[0, 10]$ und gebe abhängig von n neben dem Fehler $|I - I_n|$ auch die experimentelle Konvergenzordnung tabellarisch aus. Speichern Sie den Source-Code unter `trapezregel.c` in das Verzeichnis `serie07`.

Aufgabe 7.3. Schreiben Sie eine Funktion `int palindrome(char* str)`, die überprüft, ob ein Wort ein Palindrom ist. Ein Palindrom ist ein Wort, welches von vorne und hinten gelesen gleich lautet, z.B.: Anna, Otto, Reliefpfeiler. Wenn der Input-String ein Palindrom ist, gibt die Funktion den Wert 1 zurück, sonst den Wert 0. Schreiben Sie ferner ein aufrufendes Hauptprogramm, welches ein Wort einliest und überprüft, ob das Wort ein Palindrom ist. Testen Sie Ihr Programm entsprechend! Speichern Sie den Source-Code unter `palindrom.c` in das Verzeichnis `serie07`.

Aufgabe 7.4. Das römische Zahlensystem ist eine additive Zahlschrift, in der die verwendeten Zeichen einen festen Wert haben: $I = 1$, $V = 5$, $X = 10$, $L = 50$, $C = 100$, $L = 500$, $M = 1000$. Bei der Darstellung soll nur die Subtraktionsregel angewandt werden, d.h. $IV = 4$, $IX = 9$, $XL = 40$, $XC = 90$, $CD = 400$, $CM = 900$. Abgesehen von der Subtraktionsregel ist der Wert unabhängig von der Position. Zum Beispiel: $LXXII = 72$, $CDXII = 412$, $MCMXCIV = 1994$ $MMXVII = 2017$. Schreiben Sie:

- eine Funktion `char* int2roman(int n)`, die die Darstellung als römische Zahl eines Integers $1 \leq n \leq 3999$ berechnet und zurückgibt und
- eine Funktion `int roman2int(char* str)`, die die Dezimaldarstellung einer römischen Zahl zwischen I und $MMMCMXCIX$ berechnet und zurückgibt.

Schreiben Sie ferner ein aufrufendes Hauptprogramm, in dem sie Ihre Funktionen eingehend testen. Speichern Sie den Source-Code unter `roman.c` in das Verzeichnis `serie07`.

Aufgabe 7.5. Schreiben Sie eine Funktion `float2dec`, die für eine gegebene Mantissenlänge $M \in \mathbb{N}$, Ziffern $a_1, \dots, a_M \in \{0,1\}$ und einen Exponenten $e \in \mathbb{Z}$ den Dezimalwert $x = \left(\sum_{k=1}^{M} a_k 2^{-k}\right) 2^e$ berechnet und zurückgibt. Schreiben Sie ein aufrufendes Hauptprogramm, in dem M, a_j und e eingelesen und der Wert x ausgegeben wird. Realisieren die Potenzen 2^{-k} möglichst rechenökonomisch. Speichern Sie den Source-Code unter `float2dec.c` in das Verzeichnis `serie07`.

Aufgabe 7.6. Schreiben Sie eine Funktion `double* dec2bin(int N, int* n)`, die zu einer natürlichen Zahl $0 \leq N < 65535$ die Binärdarstellung berechnet und zurückgibt. Es sollen die Koeffizienten $a_i \in \{0,1\}$ für $i = 0, \dots, n$ ermittelt werden, sodass $N = \sum_{i=0}^{n-1} a_i 2^i$ gilt ($n \leq 16$). Die Funktion liefere die Binärdarstellung von N ohne führende Nullen. Beachten Sie, dass auch die Länge des Vektors „zurückgegeben" werden muss. Beispielsweise gebe die Funktion für $N = 77$ den Vektor `1 0 0 1 1 0 1` zurück. Schreiben Sie ferner ein aufrufendes Hauptprogramm, in dem N eingelesen und die entsprechende Binärdarstellung ausgegeben werden. Speichern Sie den Source-Code unter `dec2bin.c` in das Verzeichnis `serie07`.

Aufgabe 7.7. Die Funktion `squareVec` soll alle Einträge eines Vektors $x \in \mathbb{R}^n$ quadrieren, d.h. aus $(-1, 2, 0)$ soll $(1, 4, 0)$ werden. Der Vektor soll dabei als Pointer übergeben werden.

```
#include <stdio.h>

int squareVec(double vec, int n) {
  int j=0;
  for(j=1, j<dim; --j) {
    *vec[j] = &vec[j] * &vec[j];
  }
  return vec;
}

main() {
  double vec[3] = {-1.0,2.0,0.0};
  int j=0;

  squareVec(vec,3);
  for(j=0; j<3; ++j) {
    printf("vec[%d] = %f ",j,vec[j]);
  }
  printf("\n");
}
```

Ändern Sie nur die Funktion `squareVec`, so dass die `main`-Funktion das richtige Ergebnis ausgibt. Wie viele Fehler finden Sie? Welchen Aufwand hat Ihre korrigierte Funktion `squareVec`?

Aufgabe 7.8. Was ist ein Gleitkommazahlsystem? Aus welchen Bestandteilen setzt sich eine Gleitkommazahl zusammen? Wie bestimmt man daraus ihren Wert? Was verbirgt sich hinter den Symbolen `Inf`, `-Inf` und `NaN`? Was ist eine normalisierte Gleitkommazahl? Was ist ein implizites erstes Bit? Welchen Wert haben die größte und die kleinste positive normalisierte Gleitkommazahl im `double`-Gleitkommazahlsystem $\mathbb{F}(2, 53, -1021, 1024)$?