UNIVERSITÄT
DUISBURG
ESSEN

Offen im Denken

Prof. Dr. Johannes Kraus
Dr. Maria Lymbery

Programmierung Numerischer Algorithmen in C++

Übungsblatt 1

SS 2019

Übungsaufgabe 6

Die Eulersche Konstante sei definiert als Grenzwert $e = \lim_{n\to\infty} e_n$ mit

$$e_n = \sum_{i=0}^{n} \frac{1}{i!}.$$

Mit dieser Formel lässt sich der Wert von e im Prinzip mit beliebiger Genauigkeit annähern. Man nehme an, dass e mit der Genauigkeit $\varepsilon = 10^{-k}$ (für $k = 1, \ldots, 10$) berechnet werden soll. Es muss also nur en für steigende n berechnet werden, bis $e_n - e_{n-1} \leq \varepsilon$ gilt.

Es soll jetzt in C++ ein Programm geschrieben werden, das e als `double` mit der entsprechenden Genauigkeit berechnet und auf dem Bildschirm ausgibt.

Um das Ergebnis mit hinreichend vielen Nachkommastellen auszugeben, verwende man den Befehl `std::cout.precision(m)`, wobei `m` die Anzahl der auszugebenden Stellen ist.

Man verwende zunächst die Funktion `int factorial(int n)` zur Berechnung der Fakultät aus der Vorlesung und vergleiche jeweils $e_n - e_{n-1}$ mit `1.0/factorial(n)`. Anschließend wiederhole man dann alles mit einer neuen Funktion `double factorial(double n)`. Was fällt auf?

Übungsaufgabe 7

Man schreibe ein Programm, das Partialsummen der harmonischen Reihe berechnet:

$$s = \frac{1}{1} + \frac{1}{2} + \ldots + \frac{1}{n}.$$

Das Programm soll einmal die Summe „vorwärts" berechnen, d. h.

$$s_v = \left(\left(\left(\frac{1}{1} + \frac{1}{2}\right) + \frac{1}{3}\right) + \ldots + \frac{1}{n-1}\right) + \frac{1}{n}.$$

Zusätzlich soll die Summe auch „rückwärts" berechnet werden, also

$$s_r = \frac{1}{1} + \left(\frac{1}{2} + \ldots + \left(\frac{1}{n-2} + \left(\frac{1}{n-1} + \frac{1}{n}\right)\right)\right).$$

Es sollen s_v, s_r und $s_v - s_r$ mit 15 Nachkommastellen für $n = 10^k (k = 1, \ldots, 6)$ auf dem Bildschirm ausgeben werden. Man interpretiere die Ergebnisse. Hierzu deklariere man s_v und s_r als Variablen des Typs `float`.

Übungsaufgabe 8

Mit dem Bisektionsverfahren kann man eine Nullstelle einer stetigen Funktion f im Intervall $[a, b]$ bestimmen: wenn $f(a)$ und $f(b)$ unterschiedliche Vorzeichen besitzen, muss dazwischen eine Nullstelle der Funktion zu finden sein. Teilt man das Intervall $[a, b]$ durch Hinzufügen der Stelle $c = \frac{a+b}{2}$ in zwei halb so

große Intervalle, so muss, wenn nicht $f(c) = 0$ gilt, eine Nullstelle in einem der beiden Intervalle $[a, c]$ (falls sign$(f(a)) \neq$ sign$(f(c))$) bzw. in $[c, b]$ (falls sign$(f(b)) \neq$ sign$(f(c))$) liegen. Die Intervalllänge hat sich nach diesem Bisektionsschritt halbiert, und es kann wieder ein neuer Bisektionsschritt mit dem entsprechenden Teilintervall durchgeführt werden.

Das Verfahren soll abbrechen, wenn entweder eine maximale Anzahl Iterationen `max_iter` überschritten wird (z. B. `max_iter = 50`) oder eine gewisse Grenze `tol_f` für den Wert $|f(c)|$ in der Mitte des aktuellen Intervalls unterschritten wird (z. B. `tol_f = 1.0e-6`).

Man implementiere das Bisektionsverfahren unter Berücksichtigung dieser Eigenschaften und bestimme eine Nullstelle der Funktion $f(x) = x^3 - 2x + 5$ auf dem Intervall $[-10, 10]$. Hierbei implementiere man die Auswertung von f im Punkt x als separate Funktion double f(double x). Man benutze `std::abs` aus `<cmath>` für $|f|$.

Übungsaufgabe 9

Man beseitige alle Fehler und Warnungen in dem Programm `Prog_C++2019_9-wrongsyntax.cpp`. Was ist der Grund für diese Meldungen? Falls nötig, verbessere man den Code.

Übungsaufgabe 10

Man schreibe ein Programm das die Fläche eines Dreiecks mit von dem Nutzer gegebenen 2D Koordinaten berechnet. In dieser Implementierung erstelle man `struct Dreieck` und verwende Zeiger.