

Übungen zur Vorlesung Wissenschaftliches Rechnen

C++ Mini-Projekt Nummer 1

Schreiben Sie die Programme, beantworten Sie die Fragen, und schicken Sie beides an meine Emailadresse, sehen unten. Im Betreff soll bitte das Wort “Miniprojekt” vorkommen. Es reicht mir der C++-Quellcode; die Fragen können Sie im Kommentar im Quellcode beantworten, oder in der Programmausgabe. Es reichen ja jeweils ein bis drei Sätze. Bitte beide Lösungen in einem einzigen C++-Programm zusammenfassen, und dieses bitte mit Ihrem techfakaccount benennen (also z.B. `dfrettloeh.cpp`).

Aufgabe 1:

Schreiben Sie ein C++-Programm, das die Werte der harmonischen Reihe $\sum_{k=1}^n \frac{1}{k}$ berechnet, jeweils für $n = 10^8$ und $n = 10^9$, und zwar einmal durch Berechnen von $\frac{1}{1} + \frac{1}{2} + \dots + \frac{1}{n-1} + \frac{1}{n}$ (vorwärts), und einmal durch Berechnen von $\frac{1}{n} + \frac{1}{n-1} + \dots + \frac{1}{2} + \frac{1}{1}$ (rückwärts). Benutzen Sie dabei für die Summe jeweils die Datentypen `float` und `double`.

Wichtig: `cout` gibt standardmäßig nur sechs Dezimalstellen aus. Sie können die Anzahl der Stellen in der Ausgabe z.B. auf 15 Dezimalstellen setzen mit dem Befehl `cout.precision(15)`.

Es ergeben sich also die folgenden acht Fälle:

		10 ⁸	10 ⁹			10 ⁸	10 ⁹
float:	vorwärts			double:	vorwärts		
	rückwärts				rückwärts		

Was sind die jeweiligen Ergebnisse? Was fällt auf? Wie erklärt sich das? Was ist Ihre beste Schätzung für den wahren Wert?

Aufgabe 2:

Schreiben Sie ein C++-Programm, das die Werte von π näherungsweise auf verschiedene Weisen berechnet, durch die folgenden Methoden. Benutzen Sie für die Summen den Typ `double`. Das heißt, für die Genauigkeit sind die ersten 15 Dezimalstellen relevant.

1. Erzeugen Sie n Zufallspunkte in dem Quadrat $[0, 1] \times [0, 1]$. Zählen Sie die Punkte, deren Abstand von $(0, 0)$ kleiner als 1 ist. Teilen Sie diese Zahl durch $n/4$.
2. $\pi \approx 4 \sum_{k=0}^n \frac{(-1)^k}{2k+1}$
3. $\pi \approx \frac{16}{1 \cdot 5^1} - \frac{4}{1 \cdot 239^1} - \frac{16}{3 \cdot 5^3} + \frac{4}{3 \cdot 239^3} + \frac{16}{5 \cdot 5^5} - \frac{4}{5 \cdot 239^5} - \frac{16}{7 \cdot 5^7} + \frac{4}{7 \cdot 239^7} + \dots \mp \frac{16}{(2n+1) \cdot 5^{2n+1}} \pm \frac{4}{(2n+1) \cdot 239^{2n+1}}$

Wie groß muss n sein, damit Sie bei 1. drei korrekte Nachkommastellen erhalten? Wie groß muss n sein, damit Sie bei 2. vier korrekte Nachkommastellen erhalten? Wie groß muss n sein, damit Sie bei 3. elf korrekte Nachkommastellen erhalten? Mir reicht die Größenordnung, in der Art ‘ $2 \cdot 10^5$ ’ oder ‘etwa 210 000’.

Wichtig: Die Funktion `rand()` produziert Zufalls-Integer zwischen 0 und `RAND_MAX`. Sie brauchen für 1. aber `double` zwischen 0 und 1. Dazu müssen Sie etwas tricksen.