

Praktikum:

Einführung in die Programmierung zur Numerik mit C++

Montag, 14.03.2011

Aufgabe 1 (Potenzieren)

Schreiben Sie ein Programm `potenz.cc`, mit dem man eine ganze Zahl ($x \in \mathbb{Z}$) beliebig oft Potenzieren kann (mit $N \in \mathbb{N}$). Beachtet werden soll:

- Legen Sie ein neues Verzeichnis in Ihrem Home- oder Praktikums-Directory an (`mkdir`) und erstellen Sie dort das Programm.
- Das Programm soll 'potenzieren' heißen.
- Kompilieren Sie das Programm per Hand in der Konsole mit Debugging-Option.

Aufgabe 2 (Reihenentwicklung von π)

Schreiben Sie ein Programm `pi.cc` das π durch die Reihendarstellung

$$\pi = 4 \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1}$$

berechnet. In folgenden Schritten soll vorgegangen werden:

- Das Programm soll eine natürliche Zahl N einlesen
- Das Programm soll die Reihe bis zum N 'ten Glied berechnen, d.h.

$$4 \sum_{n=0}^N \frac{(-1)^n}{2n+1}$$

- Das Programm soll alle Zwischenergebnisse in eine Datei schreiben, in der Form:

0 4.0

1 2.66667

...

$$N \quad 4 \sum_{n=0}^N \frac{(-1)^n}{2n+1}$$

(Hierbei soll die Reihe natürlich durch die berechnete Zahl ersetzt werden.)

- Visualisieren Sie das Ergebnis mit `gnuplot` und schauen Sie sich die Konvergenzgeschwindigkeit an. Vergleich: $\pi = 3.141592653589793238462643383\dots$

Hinweise zu Aufgabe 2:

Um bei `cout` mehr Nachkommastellen anzeigen zu lassen, kann man den folgenden Code verwenden:

```
cout.setf( ios::fixed , ios::floatfield );  
cout.precision(10);
```

Es werden nun 10 Nachkommastellen angezeigt.

Die geschätzte Konvergenzgeschwindigkeit (EOC) lässt sich berechnen durch:

$$EOC = \frac{\log \left(\frac{|\pi - F(N)|}{|\pi - F(2N)|} \right)}{\log(2)}$$

wobei

$$F(N) := 4 \sum_{n=0}^N \frac{(-1)^n}{2n+1}.$$

$EOC = 1$ heißt lineare Konvergenz, $EOC = 2$ heißt quadratische Konvergenz, usw.
Wählen Sie beispielweise $N = 5000$ als Referenzwert.