

# **Anwesenheitsübung 1 zur Vorlesung 'Numerische Methoden der Physik' SS 2014**

Bastian Knippschild, Christian Jost und Mitarbeiter

**Bearbeitung in den Übungen am 15. – 17. 04. 2014**

## **Grundlagen C++**

Das Ziel dieser ersten Übung ist es Grundlagen zur Programmierung in C++ und zur Maschinengenauigkeit zu vermitteln. Im Folgenden sollen dazu mehrere, einfache Algorithmen implementiert werden.

Für C++ (und für C), lässt sich der Quellcode mit dem folgenden Compiler-Aufruf übersetzen:

```
g++ 'inputfilename' -o 'outputfilename'
```

Für Studenten mit wenig oder keiner Erfahrung in C++ empfiehlt es sich, die folgenden zwei Optionen anzugeben: `-Wall` und `-Werror`. Erstere Option sorgt dafür, dass der Compiler zusätzliche Warnungen ausgibt, letztere behandelt alle Warnungen als Fehler. Die Warnungen sind meistens selbsterklärend, ansonsten kann Google oder Ihr Tutor Ihnen weiterhelfen.

Der Compiler kann den Code selbstständig optimieren, so dass er effizienter wird. Dies geschieht durch die Optionen `-Ox` `-march=native` `-mtune=native`. Hier kann `x` die Werte 0 (keine Optimierung) bis 3 (sehr starke Optimierung) annehmen. Bei Verwendung der Option `-O3` kann es allerdings zu Problemen durch die Optimierung kommen, so dass das Program mit der Optimierung nochmals getestet werden sollte. Die Verwendung der Option `-O2` sollte jedoch zu keinerlei Problemen führen. Die Optionen `-march=native` `-mtune=native` sorgt dafür, dass das Program an den Computer angepasst wird, auf dem das Program kompiliert wird. Allerdings lässt sich dieses Program dann nur auf dem Computer ausführen, auf dem es kompiliert wurde. Ein typischer Compiler-Aufruf mit diesen Optionen sieht dann zum Beispiel wie folgt aus:

```
g++ -Wall -O2 -march=native -mtune=native 'inputfilename' -o 'outputfilename'
```

Für das Entwickeln von Programmen kann es hilfreich sein, die Option `-g` zu verwenden. Diese kann auch in Kombination mit den Optimierungsoptionen verwendet werden. Die Option sorgt dafür, dass der Compiler Debugging Informationen mitschreibt, so dass das Finden eines Fehlers einfacher wird.

Informationen über weitere Optionen können Sie sich mit der `--help` Option anzeigen lassen und auf der offiziellen Website der GNU Compiler Collection finden.

Falls mathematische Funktionen verwendet werden, muss noch die math-Bibliothek als Option hinzugefügt werden. Dies geschieht mit der Option `-lm`.

Eine weitere Einführung in C und C++ findet sich z.B. hier. Unter diesem Link finden Sie eine Referenz für alle C++ Funktionen, meistens auch mit kleinen Beispielen.

## Hallo Welt Program

Auf der eCampus-Seite finden Sie ein Beispielprogramm für die Ausgabe eines Strings. Schauen Sie sich den Code an und versuchen Sie ihn zu kompilieren und auszuführen.

Anschliessend kommentieren Sie Zeile 19 ein und versuchen Sie die Ausgabe des Compilers zu verstehen. Mit den Optionen `-Wall` `-Werror` sollte der Compiler eine Fehlermeldung ausgeben, dass eine Variable definiert wurde, aber nicht genutzt wurde.

Weisen Sie der Variable einen Wert zu und geben Sie ihn wie den String aus.

## Exponentialfunktion

Die Potenzreihendarstellung der Exponentialfunktion ist gegeben durch

$$\exp(x) = \sum_{n=0}^{\infty} \frac{x^n}{n!} \approx \sum_{n=0}^N \frac{x^n}{n!}. \quad (1)$$

Implementieren Sie die Potenzreihendarstellung in einem eigenen Program. Berechnen Sie  $\exp(-x)$  und  $\exp(x)$  für alle  $N \leq 10$  und für verschiedene  $x$  und lassen Sie sich die Ergebnisse in Tabellenform auf der Konsole ausgeben. Vergleichen Sie nun  $\exp(-x)$  und  $\frac{1}{\exp(x)}$  für alle  $N \leq 10$ . Was beobachten Sie und wie können Sie dies erklären?

Vergleichen Sie Ihre Ergebnisse auch mit der  $\exp$ -Funktion aus der C-Mathebibliothek. Diese können Sie mit `#include <cmath>` in Ihren C++ Code einbinden.