

# 2. Aufgabe: Exponentialfunktion

## 1 Einleitung

Vor ein paar Jahrzehnten waren die Ergebnisse der Exponentialfunktion in langen Tabellen in Büchern hinterlegt. Da nicht für alle  $x$ -Werte Einträge in der Tabelle vorhanden waren, musste zwischen benachbarten Zeilen interpoliert werden.

In heutigen Computern wird die Exponentialfunktion über eine Potenzreihe nachgebildet:

$$\exp(x) = \sum_{k=0}^{\infty} \frac{x^k}{k!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots$$

Nach ähnlichen Mustern werden weitere Funktionen wie Sinus und Kosinus implementiert.

## 2 Aufgaben

### 2.1 Exponentialfunktion

Erstellen Sie ein Programm, das eine Zahl vom Typ *double* vom Benutzer abfragt und damit den Exponentialwert bestimmt. Brechen Sie die Iteration nach 10 Summanden ab.

### 2.2 Testen der Exponentialfunktion

Binden Sie in Ihr Programm die Header-Datei *math.h* ein. Bestimmen Sie die Abweichung Ihres Ergebnisses der Exponentialfunktion von dem der *exp(x)*-Funktion aus der Header-Datei *math.h* in Prozent.

Bestimmen Sie die Abweichungen für verschiedene  $x$ -Werte und analysieren Sie das Problem der Berechnung. (Hinweis: Eine andere Quelle für genaue Werte der Exponentialfunktion: [www.wolframalpha.com](http://www.wolframalpha.com))

### 2.3 Schleife zur mehrfachen Abfrage von Werten

Implementieren Sie um Ihre Benutzerabfrage und Berechnung eine Schleife, die solange wiederholt wird, bis der Benutzer den Wert null eingibt.

### 2.4 Höhere Genauigkeit durch mehr Summanden

Ändern Sie die Schleife zur Berechnung der Exponentialwerte: Fügen Sie weitere Summanden hinzu, bis ein Summand die Summe nicht weiter verändert. Geben Sie mit dem Ergebnis die Anzahl der Summanden aus.

Testen Sie Ihr Programm. Wie ändert sich die Anzahl der benötigten Summanden mit dem  $x$ -Wert? Bis zu welchem Exponenten liefert die Funktion brauchbare Werte? Ihre Funktion soll bis zu einem Exponenten von 709 Ergebnisse auf 14 Stellen genau berechnen, d.h. mit einer Abweichung weniger als  $\pm 10^{-12}$  % von der Funktion *exp()* aus der *math.h*.

### 3 Optionale Zusatzaufgaben

#### 3.1 Anzahl der Iterationen reduzieren

Überlegen Sie sich, wie Sie die Anzahl der Iterationen reduzieren können. Z.B. können Sie einige Ergebnisse in Ihrem Programm hinterlegen und damit den Definitionsbereich für die Iteration auf das Intervall  $[0, 1)$  einschränken.

#### 3.2 Erstellen einer Funktion

Lagern Sie die Berechnung in eine Funktion aus: *double Exp(double x);*

#### 3.3 Funktionen für Sinus und Kosinus

Erstellen Sie Funktionen für Sinus und Kosinus:

$$\sin(x) = \sum_{k=0}^{\infty} \frac{(-1)^k x^{2k+1}}{(2k+1)!} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \dots$$
$$\cos(x) = \sum_{k=0}^{\infty} \frac{(-1)^k x^{2k}}{(2k)!} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \dots$$

*Viel Spaß beim Programmieren!*