

## A1: Nullstellen eines quadratischen Polynoms

Gegeben sei ein quadratisches Polynom

$$f(x) = Ax^2 + Bx + C$$

mit  $A, B, C \in \mathbb{R}$  und  $A \neq 0$ . Aus der Schule wissen wir, dass die Lösungen der Gleichung

$$Ax^2 + Bx + C = 0$$

durch die Formel

$$x = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \quad (1)$$

gegeben sind (evtl. kennen Sie nur die  $p$ - $q$ -Formel, das ist im Prinzip dieselbe Formel, nur mit  $p = \frac{B}{A}$ ,  $q = \frac{C}{A}$ ).

### Aufgaben:

- 1) Schreiben Sie ein Programm, das  $A$ ,  $B$  und  $C$  einliest,  $A \neq 0$  prüft, und in diesem Fall mit Hilfe von (1) die Nullstellen von  $f$  ausgibt. Testen Sie das Programm mit folgenden Daten:
  - $A = 2.25$ ,  $B = -4.5$ ,  $C = -6.75$  (Nullstellen: 3 und -1)
  - $A = 3$ ,  $B = 12$ ,  $C = -63$  (Nullstellen: -7 und 3)
  - $A = 1$ ,  $B = -2$ ,  $C = 1$  (Nullstellen: ???)
  - $A = 1$ ,  $B = -2$ ,  $C = 2$  (Nullstellen: ???)<sup>1</sup>
- 2) Erweitern Sie Ihr Programm so, dass es die Nullstellen nur dann ausgibt, wenn es reelle Zahlen sind. Fügen Sie anschließend den Code hinzu, um auch komplexe Nullstellen (z.B.  $3 + 2i$ ,  $3 - 2i$ ) ausgeben zu können, siehe Abschnitt „Hinweise“.
- 3) Ihr Programm soll nun zusätzlich die Extremalstelle von  $f$  samt Funktionswert ausgeben. Außerdem soll ausgegeben werden, ob es sich dabei um ein Minimum oder ein Maximum handelt.

### Hinweise

Die Wurzelfunktion lautet `sqrt` (*square root*). Um sie benutzen zu können, benötigen Sie am Anfang des Programmes

```
#include <cmath>
using std::sqrt;
```

um die Funktion einzubinden (`#include`) und im Folgenden `sqrt` statt `std::sqrt` schreiben zu können.

Standardmäßig rechnet C++ in reeller Arithmetik. Daher führt z.B. der Aufruf `sqrt(-1)` entweder zu `nan`<sup>1</sup> oder sogar zu einem Abbruch des Programms. Sie sollten daher immer die Diskriminante, aus der Sie dann die Wurzel ziehen wollen, erst in einer eigenen Variable ausrechnen und prüfen, ob diese nicht-negativ ist. Ist diese negativ, so ziehen wir die Wurzel aus der negativen Diskriminante, die dann ja positiv ist, und berücksichtigen bei der Ausgabe, dass dies der Imaginärteil ist.

---

<sup>1</sup>Hinweis: `nan` steht für **not-a-number**.

## Erweiterungen

Für die besonders schnellen Köpfe unter Euch: Schreiben Sie ein Programm, das

- 1\*)  $A$ ,  $B$  und  $C$  einliest und auch für  $A = 0$  und dann ggf.  $B = 0$  und  $C = 0$  die Nullstellen richtig ausgibt.
- 2\*) die zwei Nullstellen einliest und dann ein quadratisches Polynom ausgibt, welches diese Nullstellen besitzt. Wie bekommt man ein Polynom mit  $A \neq 1$ ?
- 3\*) Erstellen Sie ein neues Projekt und fügen Sie die Zeilen

```
#include <complex>
typedef std::complex<double> Complex;
```

vor der Funktion `main` ein. Deklarieren Sie Ihre Variablen vom Typ `Complex` statt vom Typ `double`. Eine reelle Zahl  $R$  (vom Typ `double`) kann man mittels `Complex(R)` in eine komplexe Zahl mit Realteil  $R$  und Imaginärteil  $0$  konvertieren. Eine komplexe Zahl  $z = (Re, Im)$  kann mithilfe von `Complex z( Re, Im)`; angelegt werden. C++ verwendet automatisch eine andere Version von `sqrt`, falls der Radikand komplex ist. Die Berechnung der Nullstellen sollte nun deutlich einfacher möglich sein, da Fallunterscheidungen wegfallen.

## Lernziele

- **Programm**  
Was ist ein Programm? Aus welchen Teilen besteht ein C++-Programm? Was ist ein Kommentar?
- **Variablen**  
Wie wird eine Variable deklariert? Was ist ein Datentyp? Welche Namen können Variablen gegeben werden? Was ist der Gültigkeitsbereich einer Variablen? Welchen Unterschied gibt es zwischen einer mathematischen Variablen und einer Variablen in C++?
- **Ein- und Ausgabe**  
Wie gebe ich eine Variable aus? Wie gebe ich Text aus (z.B. eine Eingabeaufforderung)? Wie erzwinge ich einen Zeilenumbruch? Wie lese ich Daten über die Tastatur ein?
- **Ausdrücke und Operatoren**  
Was ist ein Operator? Was ist ein Ausdruck? In welcher Reihenfolge werden die Operatoren eines Ausdrucks ausgewertet?  
Unterschied zwischen  $1/2$  und  $1.0/2$  bzw.  $1/2.0$  bzw.  $1.0/2.0$  kennen!
- **Anweisungen**  
Was ist eine Anweisung? Wie wird eine Anweisung beendet? Wo können Anweisungen im Programm stehen? Was ist ein Anweisungsblock?
- **Bedingte Ausführung von Anweisungen** (`if` bzw. `if ... else`)  
Was ist ein logischer Operator? Was ist ein logischer Ausdruck? Wie wird eine bedingte Anweisung geschrieben?  
Unterschied zwischen `==` (Vergleich) und `=` (Zuweisung) beachten!