

---

**Bisektionsmethode (10 Punkte)****[hw01]**

Verfügbar ab:	16.10.2018
Abgabe bis spätestens:	29.10.2018, 10:00

Gesucht ist die Nullstelle einer im Intervall  $[a, b]$  streng monoton steigenden oder fallenden stetigen Funktion  $f$ . Diese soll mit einer Genauigkeit  $\varepsilon$  angegeben werden, es wird also ein Teilintervall von  $[a, b]$  gesucht, das die Nullstelle enthält und höchstens die Länge  $\varepsilon$  hat.

**Es gilt:** Eine streng monotone stetige Funktion  $f$  hat in einem offenen Intervall  $]l, r[$  genau dann eine Nullstelle, wenn  $f(l) < 0$  und  $f(r) > 0$  oder  $f(l) > 0$  und  $f(r) < 0$  ist. Für das geschlossene Intervall  $[l, r]$  muss zusätzlich in Betracht gezogen werden, dass die Nullstelle auf einer Intervallgrenze liegen kann ( $f(l) = 0$  oder  $f(r) = 0$ ). Der Intervall wird dann im ersten Schritt halbiert und das Vorzeichen der neuen Intervallgrenze überprüft.

Übersetzen sie diese *Bisektionsmethode* in einen Algorithmus und schreiben Sie eine Funktion `bisection` mit folgender Signatur:

```
1 void bisection(std::vector<double>& interval, double eps);
```

Der in `interval` übergebene `vector` soll die Länge 2 besitzen und enthält die Intervallgrenzen. Die Funktion führt so lange das Bisektionsverfahren durch, bis die Länge des Intervalls kleiner als `eps` ist. Dieses Intervall wird bei der Rückkehr aus der Funktion in der Referenz `interval` zurückgegeben. Sie dürfen für dieses Beispiel annehmen, daß im ursprünglich übergebenen Intervall eine Nullstelle enthalten ist.

Bitte implementieren Sie in ihrem Starterpaket im git repository die Funktion `bisection` an der durch einen Kommentar markierten Stelle in `Bisection.cpp`. Die `main`-Routine in `main.cpp` können Sie benutzen, um Ihre Funktion zu testen. Hier können Sie auch die Funktion implementieren, deren Nullstelle Sie finden wollen – testen Sie Ihre Funktion am besten mit mehreren verschiedenen Funktionen (die Testsuite macht das auch).

**Zusatz (2 Punkte): Graphische Darstellung**

Es wird im fachlichen Alltag immer wieder nötig sein Daten oder eine Funktion graphisch darzustellen (Excel ist damit nicht gemeint!). Überlegen Sie daher wie Sie eine Funktion  $f(x)$  aufzeichnen könnten um zum Beispiel den Intervall näher einzugrenzen! Dazu könnte man  $x$  und  $f(x)$  durch das Programm in einer Schleife berechnen und dann den Inhalt in einer eigenen Datei abspeichern. Unter Windows kann man zum Beispiel mit CodeBlocks oder VisualStudio mit einem Rechtsklick auf die obere Fensterleiste des Terminalfensters und unter Bearbeiten dessen Inhalt kopieren. Unter Linux könnten Sie in der Kommandozeile den Output mit `>` in eine Datei umlenken.

Wenn Sie eine Datei erstellt haben, plotten Sie diese mit einem Tool ihrer Wahl und speichern das Ergebnis als JPEG Datei. Stellen Sie auf diese Weise zwei verschiedene Funktionen dar, die in dem entsprechenden Intervall eine oder mehrere Nullstellen besitzen. Es bietet sich hier an open-source Software zu verwenden, die Ihnen beim Arbeiten immer wieder begegnen wird. Benutzen Sie beispielsweise gnuplot, welches auch für Windows heruntergeladen werden kann.