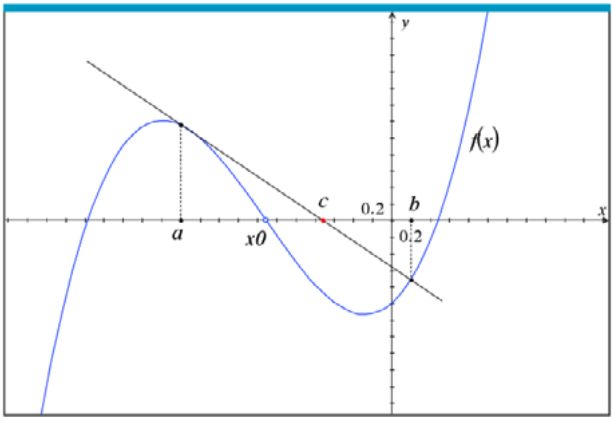
Regula Falsi

regula.py

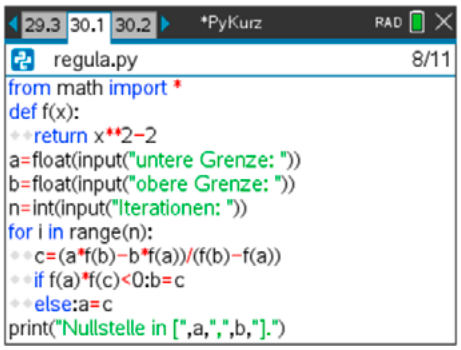
Wir betrachten eine auf $[a,b]$ stetige Funktion $f(x)$, wobei $f(a)$ und $f(b)$ verschiedene Vorzeichen haben. In der Bisektionsmethode wird der Mittelwert von a und b, $c = \frac{a+b}{2}$, verwendet, um das Intervall, das die Nullstelle enthält zu verkleinern, um damit einen besseren Näherungswert für die Nullstelle x_0 von $f(x)$ zu erhalten. Wir können durch die Punkte $(a, f(a))$ und $(b, f(b))$ eine Gerade ziehen und diese mit der x-Achse schneiden. Der Schnittpunkt $(c, 0)$ liefert einen neuen Randpunkt des Intervalls, das die Nullstelle enthält.

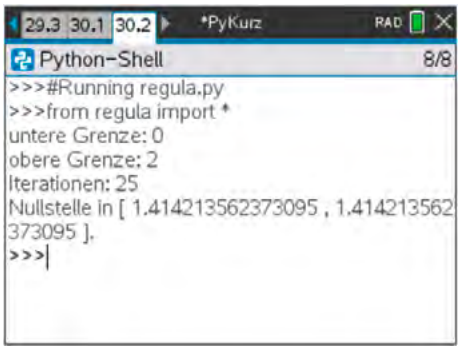


Die Gerade schneidet die x-Achse im Intervall $[a,b]$ in

$$c = \frac{a \cdot f(b) - b \cdot f(a)}{f(b) - f(a)}.$$

Wenn wir diesen Wert dazu verwenden, um das Intervall zu verkleinern, erhalten wir eine andere Methode zur näherungsweisen Bestimmung einer Nullstelle einer Funktion: die *Regula Falsi*. Hier ist das entsprechende Pythonprogramm:





Im Vergleich der beiden Methoden merkt man, dass die Regula Falsi für die gleiche Anzahl von Iterationen bessere Resultate liefert – sie konvergiert rascher.

Auch für dieses Programm stehen die gleichen Verbesserungsmöglichkeiten im Raum, die für Bisektionsmethode vorgeschlagen werden:

- Die Funktion kann über eine Abfrage eingeben werden (und nicht im Programm).
- Anstelle der Iterationsanzahl kann eine Genauigkeit vorgegeben werden.
- Das Ausgabeformat (Anzahl der Dezimalstellen) kann eingestellt werden.)