

Die Bisektionsmethode

bisektion.py

Satz von Bolzano: Sei f eine reelle Funktion, stetig in $]a,b[$, rechtsstetig in a und linksstetig in b . Wenn $f(a)$ und $f(b)$ verschiedene Vorzeichen haben, gibt es eine Nullstelle in $[a,b]$.

Durch iterative Anwendung dieses Satzes können wir die Nullstelle näherungsweise berechnen. Angenommen, $f(a)$ und $f(b)$ haben verschiedene Vorzeichen, dann liegt zwischen a und b eine Nullstelle. Berechne nun den Mittelwert $c = \frac{a+b}{2}$.

- So lange $f(c)$ und $f(a)$ verschiedene Vorzeichen haben, ersetzen wir b durch c .
- So lange $f(c)$ und $f(b)$ verschiedene Vorzeichen haben, ersetzen wir a durch c .
- Die Funktion hat nun eine Nullstelle im neuen (kleineren) Intervall $[a,b]$.

Es wird wieder die Intervallmitte $c = \frac{a+b}{2}$ berechnet und die Vorgangsweise wird wiederholt.

So erhalten wir immer kleiner werdende Intervalle, die die Nullstelle immer besser annähern. Dieser Algorithmus wird *Bisektionsmethode* genannt.



```

2.4 3.1 3.2 *PyKurz RAD X
Bisektion.py 11/11
from math import *
def f(x):
    return x**2-2
a=float(input("untere Grenze: "))
b=float(input("obere Grenze: "))
n=int(input("Iterationen: "))
for i in range(n):
    c=(a+b)/2
    if f(a)*f(c)<0:b=c
    else:a=c
print("Nullstelle in [",a,",",b,"]")

```



```

2.4 3.1 3.2 *PyKurz RAD X
Python-Shell 8/8
>>>#Running Bisektion.py
>>>from Bisektion import *
untere Grenze: 0
obere Grenze: 2
Iterationen: 25
Nullstelle in [ 1.414213538169861 , 1.414213597
774506 ].
>>>

```

Mit diesen Schritten erfolgt die Implementation in Python:

- Das Modul `math` wird importiert und die Funktion f definiert. Dann werden die Grenzen und die Anzahl der vorgesehenen Iterationsschritte abgefragt.
- Die `for`-Schleife lässt uns die Iteration n mal ausführen, in der jedes Mal der Wert $c = \frac{a+b}{2}$ berechnet wird.
- Mit `if f(a)*f(b)` wird untersucht, ob die beiden Funktionswerte unterschiedliche Vorzeichen haben. Dann wird die Grenze b verändert, anderenfalls a .
- Nach dem Verlassen der Schleife wird das letzte Intervall $[a,b]$ ausgegeben.

Dieses Programm kann in mehrfacher Weise verbessert werden. (Damit wird es aber den Rahmen des Schirms sicher sprengen!)

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