

### Heron's Algorithmus/ Babylonisches Wurzelziehen

Berechnen Sie für eine beliebige nichtnegative Zahl  $a \geq 0$  die Wurzel  $\sqrt{a}$  bis auf einen (relativen) Fehler von  $10^{-10}$  nach der folgenden iterativen Methode:

$$x_{n+1} = \frac{1}{2} \left( \frac{x_n^2 + a}{x_n} \right).$$

1. Schreiben Sie dafür eine Python Funktion `heron(a, x0, tol)`.
2. Wählen Sie verschiedene Anfangswerte  $x_0$  und beobachten Sie das Konvergenzverhalten, indem Sie zum Beispiel die Fehler  $|x_n - \sqrt{a}|$  in einer Liste abspeichern.
3. Erweitern Sie die Funktionsschnittstelle um einen Parameter `maxiter` (`heron(a, x0, tol, maxiter)`). Brechen Sie (zusätzlich zum Fehlertoleranzkriterium) die Iteration ab, sobald die Anzahl der Schleifendurchläufe den Wert `maxiter` erreicht hat.

*Hinweis:* Benutzen Sie die built-in Funktion `abs()` und den Python-Wert `a ** 0.5` zur Fehlermessung.

### Solution:

```
1  #!/usr/bin/env python
2  # coding: utf-8
3  # <h1>Table of Contents<span class="tocSkip"></span></h1>
4  # <div class="toc"><ul class="toc-item"></ul></div>
5  def heron(a, x0=0.1, tol=10e-10, maxiter = 1000):
6      """
7      applies the iteration rule according to the so -called "Heron method"
8      """
9      counter = 0
10     # while the error is above our tolerance we do the iteration
11     while abs(x0 - a**0.5) > tol:
12         # print the current error
13         print("Error =", abs(x0 - a**0.5))
14         # perform one step of the heron method:
15         x0 = 0.5 * ((x0 ** 2 + a) / x0)
16         # update the counter
17         counter += 1
18         if counter > maxiter:
19             # stop the while loop if too many iterations
20             break
21         print("\n Result: sqrt(a) =", x0, "\n")
22     return x0
23 if __name__ == "__main__":
24     a = 2.
25     # Choose different initial values
26     x0 = [0.1, 1., 100000 , 1.3]
27     for x in x0:
28         print("\n x0 =", x, "\n-----\n")
29         heron(a, x0=x)
30     help(heron)
```