In this paper we will try to make an approximation of the square root of 2($\sqrt{2} \in \mathbb{R}\setminus\mathbb{Q}$) using the Python programming language. The code we use is as follows:

```
import time
import mpmath as mpm # library for arbitrary precision calculations
mpm .mp.dps = 60 # define decimal precision for calculations
a, x0 , err , n , h = mpm .mpf (2) , mpm .mpf (1) , mpm .mpf (10**( -50) ),
1 , 0.0001
start = time .time ()
rec = lambda x: (x **2+h*x+2) /(2*x+h) #N-R recursive formula
x1 = rec (x0)
while mpm .fabs(x1 -x0) >= err :
# print (n, x1 , mpm . fabs(x1 -x0))
 x0 = x1
 x1 = rec (x0)
 n = n + 1
 end = time.time ()
print (" Approximation of the square root of %s using the N -R method
:"%(a))
print ("root = %s, repetitions = %s"%(x1 , n))
print ("absolute error = %s"%( mpm . fabs(x1 -x0)))
print ("Time elapsed (seconds ): %s"%( end - start ))
```

Output(h=0.0001):

```
Approximation of the square root of 2.0 using the N -R method :

root = 1.41421356237309504880168872420969807856967187537694807342063,
repetitions = 14

absolute error =
6.89984455754972925194089692086837433606111019158518338811 68e-51

Time elapsed (seconds ): 0.000999927520751953 1
```

In order to better understand the code we will change the values of the variable h and create the following table:

h = 0.0001	Approximation of the square root of 2.0 using the N -R method : root = 1.41421356237309504880168872420969807856967187537694807342063, repetitions = 14 absolute error = 6.89984455754972925194089692086837433606111019158518338811 68e-51 Time elapsed (seconds): 0.0010001659393310547

h = 0.001	Approximation of the square root of 2.0 using the N -R method : root = 1.41421356237309504880168872420969807856967187537694807461276, repetitions = 17 absolute error = 4.06183439683255416806755083496493133384870347253830479470572e-51 Time elapsed (seconds): 0.0009999275207519531
h = 0.01	Approximation of the square root of 2.0 using the N -R method : root = 1.41421356237309504880168872420969807856967187537694807417125, repetitions = 23 absolute error = 2.81306580154091075395639153365261613843043089590752429640812e-52 Time elapsed (seconds): 0.0010001659393310547
h = 0.1	Approximation of the square root of 2.0 using the N -R method : root = 1.414213562373095048801688724209698078569671875376948119418 79repetitions = 36 absolute error = 1.30792452064894934307065289776775335629416807625729874395801e-51 Time elapsed (seconds): 0.0009982585906982422
h = 1	Approximation of the square root of 2.0 using the N -R method : root = 1.41421356237309504880168872420969807856967187537694594756577repetitions = 86 absolute error = 6.01213554126478528973840752046577595911402753784427981283535e-51 Time elapsed (seconds): 0.0020003318786621094

h = 10	Approximation of the square root of 2.0 using the N -R method : root = 1.41421356237309504880168872420969807856967187537692003345421, repetitions = 455 absolute error = 7.93083115993860423694762064897130223461322920124943361314916e-51 Time elapsed (seconds): 0.012000799179077148
h = 100	Approximation of the square root of 2.0 using the N -R method : root = 1.41421356237309504880168872420969807856967187537659633283562, repetitions = 3974 absolute error = 9.94871921547672218101650145766630591894799186605516962474549e-51 Time elapsed (seconds): 0.10200953483581543
h = 1000	Approximation of the square root of 2.0 using the N -R method : root = 1.41421356237309504880168872420969807856967187537342120236784, repetitions = 38429 absolute error = 9.97549706133315197522157478015209481804596658316148163078482e-51 Time elapsed (seconds): 1.0560789108276367
h = 10000 (extra)	Approximation of the square root of 2.0 using the N -R method : root = 1.41421356237309504880168872420969807856967187534159720004488 repetitions = 375653 absolute error = 9.99873684483533478997310239288364651254161218752560371147281e-51 Time elapsed (seconds): 10.202839374542236

By observing the above tables we conclude that as the variable h increases, the number of repetitions, the absolute error and the time elapsed (time elapsed) of the algorithm increases. In this way, however, we manage to get a better approximation of the root.

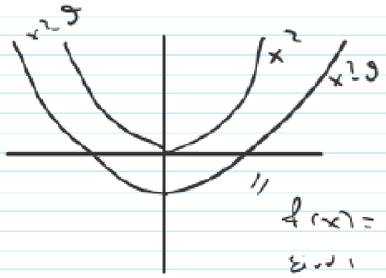
Below we can see how the new type of function was derived:

Εργασία μαθηματικά

Δευτέρα, 24 Ιανουαρίου 2022 11:30 μμ

Έστω $f(x_n) = x_n^2 - \theta$

$f(x_n+h) = (x_n+h)^2 - \theta$



$f(x) = x^2 - \theta$ είναι συνεχής για $x>0$ έχει μοναδική λύση

$$\boxed{x_{n+1} = x_n - \frac{f(x_n)h}{f(x_n+h) - f(x_n)}}$$

Άρα $x_{n+1} = x_n - \frac{(x_n^2 - \theta)\cdot h}{(x_n+h)^2 - \cancel{\theta} - x_n^2 + \cancel{\theta}} = x_n - \frac{h(x_n^2-\theta)}{\cancel{x_n^2} + 2x_nh + h^2 - \cancel{x_n^2}}$

$= x_n - \frac{\cancel{h}(x_n^2-\theta)}{\cancel{h}(+2x_n+h)} = \frac{x_n(+2x_n+h) - (x_n^2-\theta)}{+2\cdot x_n + h} =$

$\frac{+2\cdot x_n^2 + h\cdot x_n - x_n^2 + \theta}{+2x_n+h} = \frac{x_n^2 + h\cdot x_n + \theta}{2x_n+h}$

για $\theta = 2$: $\boxed{x_{n+1} = \frac{x_n^2 + h x_n + 2}{2x_n + h}}$