Global Scientific JOURNALS

GSJ: Volume 11, Issue 9, September 2023, Online: ISSN 2320-9186
www.globalscientificjournal.com

CONTRAST BETWEEN BISECTION, NEWTON-RAPHSON AND SECANT METHODS FOR DETERMING THE ROOT OF THE NON-LINEAR EQUATION USING PYTHON PROGRAMS

[1]Luka, Joshua, [2]Abdullahi M.A and [3]Danladi Biong'ahu

[1'2]Federal Polytechnic, Bauchi State Nigeria
[3]Nuhu Bamali Polytechnic Zaria, Kaduna State Nigeria.
chieflukajayn@gmail.com
07035863238.

ABSTRACT

In this paper, we present the mathematical background of the three most common numerical methods of solving non-linear equations. The Bisection, Newton-Raphson and Secants method are indicated to show the numerical approximation of the non-linear equation $f(x) = x^3 -4x +2 = 0$ on a closed interval [a,b]. the paper wants to display the comparison of the implementation and the rate of convergence among the numerical methods to detect the root of the non-linear equation using PYTHON PROGRAMS. The Newton- Raphson, and Secant methods are more absolutely accurate and speedy to converge with a few steps of iterations while the Bisection method takes too much iteration to converge. It was observed that the Bisection method converges at the 13^{th} iteration while Newton-Raphson and Secant method converges to get the exact root of 0.5392 at the 3^{rd} and 5^{th} iterations respectively. It was then concluded that of the three methods considered the Secant method is the most effective to use although the Newton-Raphson method converges faster but it requires difficulty in taking a derivation, this is in line with the results on the table of iterations.

Keywords: *Roots, Convergence, Algorithms, Iterations, Bisection method, Newton-Raphson method, Secant method. python programming*

INTRODUCTION

In the field of mathematics, chemistry, physics, bioscience, engineering, etc. diverse types of equations become available. Actually, the discovery of any unknown appearing completely in the engineering or scientific formula brings about the root-finding problem. Root-finding problem is a problem of obtaining a root of equation f(X)=0, where F(X) is a function of a single variable x. Root-finding problem is of the most basic problems of numerical estimation, this involves finding a root (or zero, or solution), of the equation of the form f(x)=0 for a given function f. Generally, it will be impossible to solve such root-finding problems analytically. Truly algorithms for solving problems numerically can be divided into mean types: direct methods and iterative methods. Direct methods can be completed in a calculated finite number of steps while iterative methods are the method that converges to a solution gradually over a long period. The most common root-finding methods of the nonlinear equation include; the bisection method, Newton-Raphson method, secant method, false position method, fixed point iteration method, steffensen's method, etc. The different method converges to the root at different rates, that is some method are faster in converging to the root than the other, and this is also based on the order because the higher the order the faster the method of converges.

Nonlinear equation f(x)=0, where f(x) may be algebraic, trigonometric or transcendental functions. Its more unlike to see the form of a nonlinear equation in the case of cubic and quadratic polynomial situations because if the value of $n \geq 5$ then finding the solution to these polynomial functions is radical (suli & mayers, 2003). This study is at finding the root of the nonlinear equation by using some selected numerical methods (bisection, newton Raphson, and secant) ad also comparing the rate of performance (convergence) of bisection, newton Raphson, and secant methods of root-findings. Actually, all numerical methods have their own limitation that is related to the properties of the function developed in the equation besides, we must know the interval in which the root of a given equation is. On the contrary, we may use many different methods in the process of finding the approximation root of a nonlinear equation. The most important methods in finding the root of a nonlinear equation will be used in these studies which are bisection, Newton-Raphson, and secant methods, the computation that is made using these methods is very effortful because of many loops appearing. for that reason, we should use a computer program which is the python programing language. python programs are one of the most effective and common programming languages that allow the writing of short and simple applications that solve mathematical problems. In addition, python programs have no limitation on introduced numbers.

REVIEW OF RELATED STUDIES

In the case of earlier research, the study that is related to our research topic is the consideration of determining the root of the equation and illustration of the rate of convergence between three used numerical methods and demonstrating its application under the implementation of Mathematics programs by Ehiwario and Aghamie (C&O, 2014)
In another study by Abdulaziz G. Ahmad (Ahmad, 2015) on the interpretation of numerical solutions to determine the root-finding problem of the nonlinear equations and considered the rate of convergence of each formula.

In another study by Ali Wahid Nwry, Hemin Muheddin Kareem , Ribwar Bakhtyar Ibrahim and sundws Mustafa Mohammed on the comparison between bisection, Newton-Raphson and secant methods for determining the root of the nonlinear equation using MATLAB is also related to our study.

METHODS

1. BISECTION METHOD

Is the first and the simplest method in the field of numerical analyses, the method is based on repeated bisection of an interval containing the root. The bisection method is a technique to discover a root of the considered nonlinear equation in the form of f(X)=0 with its algorithm which frequently bisects a given interval and chooses a midpoint between a lower and upper of the closed interval to proceed of root finding lying until new subintervals are created. The bisection method Is based on the intermediate value theorem

Theorem 1: let f(X) be a continuous function defined on the interval [a,b]. such that f(a) and f(b) is the opposite sign. if f(a). f(b) < 0, then there exist in an interval (a,b) at least one of the root x=c of the equation f(x)=0.

Let f(x) be a continuous function defined on the interval [a,b] such that f(x) are opposite signs and let f(a) . f(a)<0 which means the sign of the value of the function changes at the interval [a,b]. then we will find the root of the equation f(x)=0 (or zero of f). From the idea of bisection, the analyzed interval will be divided into two identical parts using the midpoint of the interval then we will choose the part of the interval, where the value of the function changes the sign. This is repeated until we get the exact solution or until the required iteration is reached. This method of bisection is easy and the root always converges, but very slow in converging.

Algorithm 1.1: Bisection method

Inputs:(i) f(x) - the given equation

(ii) a_0 , b_0 - the two numbers such that f(a).f(b)<0

Outputs: An approximation of the root of f(x)= 0 in $[a_0,b_0]$ for n= 0,1,2,3,... do until satisfied

Compute $C_n = \frac{a_n+b_n}{2}$

Test if C_n is the desired root, if so, stop

If C_n is not the desired root,

test if $f(c_n)\, f(a_n) < 0$. if so,

set $b_n+1=c_n$ and $a_n+1 = a_n$

otherwise, set $a_{n+1} = c_n$, $b_{n+1} = b_n$

End.

2. NEWTON – RAPHSON METHOD

Newton – Raphson method which is also called the successive substitution method or tangent method it's an iterative process that can approximate a solution to a nonlinear equation with incredible accuracy. And it's a method that approximates numerical solutions i.e (zeros or roots) to nonlinear equations that are too complex to solve by hand. Newton-Raphson's method finds the slope of the function at the current point and uses the zero of the tangent line as the next reference point. The process is repeated until the root is found. Newton – Raphson method is one of the best and one of the most popular methods for solving nonlinear equations as its convergence is very quick, but it is sometimes choked if the wrong initial guesses are used. Newton – Raphson method is more effective and more accurate than the bisection method, although it requires the computation of the derivative of a nonlinear function as a reference point which is difficult, or either the derivative does not exist at all or it cannot be expressed in terms of elementary function. The function f in the equation must satisfy the following conditions

Theorem 2:
if f(X) is continuous in the interval [a,b] and:
◦ The signs of the function are different on the endpoints of the interval [a,b] i.e f(a). f(b) < 0,

◦ There exists only one root of the function f in the analyzed interval,

◦ The first and second derivatives of the function f do not change their signs in the interval [a,b],

◦ Then f(X) = 0 has only one solution x_2E [a,b].

The function f(x) can be expanded in the neighborhood of the root x_0 through the Taylors expansion $f(x_0) = f(x) + (x_0-x)\, f^{l}(x) + \frac{(x_0-x)^2}{2!} f^{ll}(\mathcal{E}(x_0)) = 0$

Where x can be the trial value for the root at the n^{th} step and the approximate value of the next step x_{n+1} can be derived from:

$f(x_{n+1}) = f(x_n)+ f(x_{n+1}- x_n)\, f^{l}(x_n)$

$x_{n+1} = x_n - \frac{f(x_n)}{f^{l}(x_n)}$, $n = 0,1,2,...$ is called the Newton-Raphson method

Algorithm 1.2 Newton-Raphson method

Inputs: f(x) – the given function

x_0 – the initial approximation

N – the maximum number of iteration

Ɛ – the error of tolerance

Outputs: An approximation to the root

x = Ɛ or a message of failure

Assumption: x = Ɛ is a simple root of f(x) = 0

Compute f(x) and $f^l(x)$

Compute $x_{n+1} = x_n - \frac{f(x_n)}{f^l(x_n)}$, $n = 0,1,2,\ldots$ do until it convergence or fail.

Test for convergence or failure: if

$$|f(x_n)| < \varepsilon \quad \text{or} \quad \frac{|x_{n+1} - x_n|}{x_n} < \varepsilon \quad \text{or } n > N, \text{ stop}$$

End.

3. SECANTS METHOD

In numerical analysis, the secant method is a root-find algorithm for nonlinear equations that uses a succession of roots of secant lines to approximate the root of a function f(x)=0. The secant method is similar to the Newton-Raphson method in that a straight line is used to determine the approximation to the root. In difference to the Newton-Raphson method, the secant method uses two initial guesses for the root x_0 and x_1 and a straight line is fitted between the evaluation of f(x). The line is called the secant line and an approximation of the root x_2, is given by the intercept of the secant line with the x-axis. The secant method does not require $f^l(x)$ which is an advantage over the Newtown-Raphson method.

As we have noticed. The major disadvantage of the Newton-Raphson method is the requirement of finding the value of the derivative of f(x) at each approximation. There are functions that are either entirely different (if not possible) or time-consuming. The way out is to approximate the derivatives by knowing the values of the function at that and the previous approximation. Knowing, $f(x_{n-1})$, we can then approximate $f^l(x_n)$ as :

$$f^l(x_n) = \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}$$

Then the newton iterations, in this case, are:

$$x_{n+1} = x_n - \frac{f(x_n)}{f^l(x_n)}$$

$$= x_n - \frac{f(x_n)(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})} \quad \text{is referred to as secant method}$$

Algorithm 1.3 The secant method

Inputs: f(x) = The given function

x_o, x_1 = The two initial approximations of the root

Ɛ = The error of tolerance

N = The maximum number of iteration

Outputs: An Approximation of the exact solution Ɛ or a massage of failure for n=1, 2,… do until convergence or otherwise

Compute $f(x_n)$ and $f(x_{n-1})$

Compute the next approximation $x_{n+1} = x_n - \frac{f(x_n)(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})}$

Test for convergence or maximum number iteration: if $|x_{n+1} - x_n| < Ɛ$ or if n >N, stop.

End:

Results

We have applied each of the Bisection, Newton-Raphson, and Secant methods using python programs and in the theoretical form to find an approximate solution to the non-linear equation $f(x) = x^3 - 4x + 2 = 0$ which is on a closed interval [ab] the result is considered below.

BISECTION METHOD – PROGRAM IN PYTHON LANGUAGE

```
def f(x) :
   return x**3-4*x+2
a=int(input("first initial value="))
b=int(input("second initial value="))
if f(a)*f(b)>0:
   print("Bisection method fails")
else:
   n=0
   while n<=10:
      c=(a+b)/2
      if f(a)*f(c)<0:
          b=c
      else:
         a=c
         n=n+1
      print("Root of given equation=",c)
```

```
first initial value=1
second initial value=0
Root of given equation= 0.5
Root of given equation= 0.75
Root of given equation= 0.625
Root of given equation= 0.5625
Root of given equation= 0.53125
Root of given equation= 0.546875
Root of given equation= 0.5390625
Root of given equation= 0.54296875
```

Root of given equation= 0.541015625
Root of given equation= 0.5400390625
Root of given equation= 0.53955078125
Root of given equation= 0.539306640625
Root of given equation= 0.5391845703125
Root of given equation= 0.53924560546875
Root of given equation= 0.539215087890625

NEWTON-RAPHSON METHOD– PROGRAM IN PYTHON LANGUAGE

```
def f(x):
   return x**3-4*x+2
def df(x):

   return 3**x-4
a=float(input("initial guess ="))
n=int(input("number of iterations"))
m=1
while (m<=n):
   r=a-f(a)/df(a)
   a=r
   m=m+1
   print("root=",r,"at iteration,",m)
```

first initial value=1
second initial value=0
the root is 0.53846153846
the root is 0.53923347773
the root is 0.53920000000

SECANT METHOD – PROGRAM IN PYTHON LANGUAGE

```
def secantmethod(func,x0,x1,n):
   def f(x):
      f= eval(func)
      return f
   for intercept in range (1,n):
      fx0 = f(x0)
      fx1 = f(x1)
      xi = x0 - (fx0/((fx0 - fx1)/(x0 -x1)))

      x0 = x1
      x1 = xi
      n="number of iterations"
```

```
    print(f"the root is {xi}")
secantmethod("x**3 -4*x +2",0,1,7)
the root is 0.6666666666666666
the root is 0.47058823529411764
the root is 0.5440427698574338
the root is 0.5393492275452046
the root is 0.5391884679860588
the root is 0.5391888728444665
```

THEORETICAL PART OF THE BISECTION METHOD

Bisection method for x^3- 4x +2

1st iteration

Here f(0) = 2 > 0 and f(1) = -1 < 0 $\therefore$
$Now, the\ root\ lies\ between\ 0\ and\ 1$

$X_0 = \frac{0+1}{2} = 0.5$

$f(x_0) = f(0.5) = 0.5^3-4(0.5) +2 = 0.125 > 0$

2nd iteration

Here f(0.5) = 0.125> 0 and f(1) = -1 < 0 $\therefore$
$Now, the\ root\ lies\ between\ 0.5\ and\ 1$

$X_1 = \frac{0.5+1}{2} = 0.75$

$f(x_1) = f(0.75) = 0.75^3-4(0.75) +2 = -0.5781< 0$

3rd iteration

Here f(0.5) = 0.125> 0 and f(0.75) = -0.5781 < 0 $\therefore$
$Now, the\ root\ lies\ between\ 0.5\ and\ 0.75$

$X_2 = \frac{0.5+0.75}{2} = 0.625$

$f(x_2) = f(0.625) = 0.625^3-4(0.625) +2 = -0.2559 < 0$

4th iteration

Here f(0.5) = 0.125> 0 and f(0.625) = -0.2559 < 0 $\therefore$
$Now, the\ root\ lies\ between\ 0.5\ and\ 0.625$

$X_3 = \frac{0.5+0.625}{2} = 0.5625$

$f(x_3) = f(0.5625) = 0.5625^3-4(0.5625) +2 = -0.072< 0$

5th iteration

Here f(0.5) = 0.125> 0 and f(0.5625) = -0.072 < 0 $\therefore$
$Now, the\ root\ lies\ between\ 0.5\ and\ 0.5625$

$X_4 = \frac{0.5+0.5625}{2} = 0.5312$

$f(x_4) = f(0.5312) = 0.5312^3-4(0.5312) +2 = 0.0249 > 0$

6th iteration

Here f(0.5312) = 0.0249> 0 and f(0.5625) = -0.072 < 0 ∴
$Now, the\ root\ lies\ between\ 0.5312\ and\ 0.5625$

$X_5 = \frac{0.5312+0.5625}{2} = 0.5469$

$f(x_5) = f(0.5469) = 0.5469^3 - 4(0.5469) + 2 = -0.0239 < 0$

7th iteration

Here f(0.5312) = 0.0249> 0 and f(0.5469) = -0.0239 < 0 ∴
$Now, the\ root\ lies\ between\ 0.5312\ and\ 0.5469$

$X_6 = \frac{0.5312+0.5469}{2} = 0.5391$

$f(x_6) = f(0.5391) = 0.5391^3 - 4(0.5391) + 2 = -0.0004 > 0$

8th iteration

Here f(0.5391) = 0.0004 > 0 and f(0.5469) = -0.0239 < 0 ∴
$Now, the\ root\ lies\ between\ 0.5391\ and\ 0.5469$

$X_7 = \frac{0.5391+0.5469}{2} = 0.543$

$f(x_7) = f(0.543) = 0.543^3 - 4(0.543) + 2 = -0.0118 < 0$

9th iteration

Here f(0.5391) = 0.0004 > 0 and f(0.543) = -0.0118 < 0 ∴
$Now, the\ root\ lies\ between\ 0.5391\ and\ 0.543$

$X_8 = \frac{0.5391+0.543}{2} = 0.541$

$f(x_8) = f(0.541) = 0.541^3 - 4(0.541) + 2 = -0.0057 < 0$

10th iteration

Here f(0.5391) = 0.0004 > 0 and f(0.541) = -0.0057 < 0 ∴
$Now, the\ root\ lies\ between\ 0.5391\ and\ 0.541$

$X_9 = \frac{0.5391+0.541}{2} = 0.54$

$f(x_9) = f(0.54) = 0.54^3 - 4(0.54) + 2 = -0.0027 < 0$

11th iteration

Here f(0.5391) = 0.0004 > 0 and f(0.54) = -0.0027 < 0 ∴
$Now, the\ root\ lies\ between\ 0.5391\ and\ 0.54$

$X_{10} = \frac{0.5391+0.54}{2} = 0.5396$

$f(x_{10}) = f(0.5396) = 0.5396^3 - 4(0.5396) + 2 = -0.0011 < 0$

12th iteration

Here f(0.5391) = 0.0004 > 0 and f(0.5396) = -0.0011 < 0 ∴
$Now, the\ root\ lies\ between\ 0.5391\ and\ 0.5396$

$X_{11} = \frac{0.5391+0.5396}{2} = 0.5393$

$f(x_{11}) = f(0.5393) = 0.5393^3 - 4(0.5393) + 2 = -0.0004 < 0$

13th iteration

Here f(0.5391) = 0.0004 > 0 and f(0.5393) = -0.0004 < 0 ∴
$Now, the\ root\ lies\ between\ 0.5391\ and\ 0.5393$

$X_{12} = \frac{0.5391+0.5393}{2} = 0.5392$

$f(x_{12}) = f(0.5392) = 0.5392^3 - 4(0.5392) + 2 = 0$

The approximate root of the equation $x^3-4x+2=0$ using the Bisection method after 13 iterations is 0.5392

Table 1: illustration of the Bisection method

n	*A*	*f(a)*	*B*	*f(b)*	*C* $= \frac{a+b}{2}$	*f(c)*	*Update*
1	*0*	*2*	*1*	*-1*	*0.5*	*0.125*	*a = c*
2	*0.5*	*0.125*	*1*	*-1*	*0.75*	*-0.5781*	*b = c*
3	*0.5*	*0.125*	*0.75*	*-0.5781*	*0.625*	*-0.2559*	*c b =*
4	*0.5*	*0.125*	*0.625*	*-0.2559*	*0.5625*	*-0.072*	*b = c*
5	*0.5*	*0.125*	*0.5625*	*-0.072*	*0.5312*	*0.0249*	*a = c*
6	*0.5312*	*0.0249*	*0.5625*	*-0.072*	*0.5469*	*-0.239*	*b = c*
7	*0.5312*	*0.0249*	*0.5469*	*-0.0239*	*0.5391*	*0.0004*	*a = c*
8	*0.5391*	*0.0004*	*0.5469*	*-0.0239*	*0.543*	*-0.0118*	*b = c*
9	*0.5391*	*0.0004*	*0.543*	*-0.0118*	*0.541*	*-0.0057*	*b = c*
10	*0.5391*	*0.0004*	*0.541*	*-0.0057*	*0.54*	*-0.0027*	*b = c*
11	*0.5391*	*0.0004*	*0.54*	*-0.0027*	*0.5396*	*-0.0011*	*b = c*
12	*0.5391*	*0.0004*	*0.5396*	*-0.0011*	*0.5393*	*-0.0004*	*b = c*
13	*0.5391*	*0.0004*	*0.5393*	*-0.0004*	*0.5392*	0	*a = c*

THEORETICAL PART OF THE NEWTON-RAPHSON METHOD
Newton-Raphson method for x^3- 4x + 2
Let $f(x) = x^3-4x+2$

$\frac{d}{dx}(x^3-4x+2) = 3x^2-4$

$\therefore f(x) = 3x^2-4$

Here f(0) = 2 >0 and f(1) = -1 <0

The root lies between 0 and 1

$x_0 = \frac{0+1}{2} = 0.5$

$x_0 = 0.5$

1^{st} iteration

$f(x_0) = f(0.5) = 0.5^3 - 4(0.5) + 2 = 0.125$

$f^l(x_0) = f^l(0.5) = 3(0.5)^2 - 4 = -3.25$

$x_1 = x_0 - \frac{f(x_0)}{f^l(x_0)}$

$x_1 = 0.5 - \frac{0.125}{-3.25} = 0.5385$

$x_1 = 0.5385$

2^{nd} iteration

$f(x_1) = f(0.5385) = 0.5385^3 - 4(0.5385) + 2 = 0.0023$

$f^l(x_1) = f^l(0.5385) = 3(0.5385)^2 - 4 = -3.1302$

$x_2 = x_1 - \frac{f(x_1)}{f^l(x_1)}$

$x_2 = 0.5385 - \frac{0.0023}{-3.1302} = 0.5392$

$x_2 = 0.5392$

3^{rd} iteration

$f(x_2) = f(0.5392) = 0.5392^3 - 4(0.5392) + 2 = 0$

$f^l(x_2) = f^l(0.5392) = 3(0.5392)^2 - 4 = -3.1278$

$x_3 = x_2 - \frac{f(x_2)}{f^l(x_2)}$

$x_3 = 0.5392 - \frac{0}{-3.1278} = 0.5392$

$x_3 = 0.5392$

The approximate root of the equation x^3-4x+2=0 using the New-Raphson method after 3 iterations is 0.5392

Table 2: illustration of the Newton-Raphson method

N	x_0	$f(x_0)$	$f^l(x_0)$	x_1	Update
1	*0.5*	*0.125*	*-3.25*	*0.5385*	$x_0 = x_1$
2	*0.5385*	*0.0023*	*-3.1302*	*0.5392*	$x_0 = x_1$
3	*0.5392*	*0*	*-3.1278*	*0.5392*	$x_0 = x_1$

THEORETICAL PART OF THE SECANT METHOD

Secant method for x^3- 4x + 2

Let $f(x) = x^3-4x+2$

Here $f(0) = 2$

$f(1) = -1$

1st iteration

$x_0 = 0$ and $x_1 = 1$

$f(x_0) = f(0) = 2$ and $f(x_1) = f(1) = -1$

$\therefore x_2 = x_0 - f(x_0).\frac{x_1 - x_0}{f(x_1) - f(x_0)}$

$x_2 = 0 - 2 .\frac{1-0}{-1-2}$

$x_2 = 0.6667$

$f(x_2) = f(0.6667) = 0.6667^3-4(0.6667) +2 = -0.3704$

2nd iteration

$x_1 = 1$ and $x_2 = 0.6667$

$f(x_1) = f(1) = -1$ and $f(x_2) = f(0.6667) = -0.3704$

$\therefore x_3 = x_1 - f(x_1).\frac{x_2 - x_1}{f(x_2) - f(x_1)}$

$x_3 = 1 -(-1) .\frac{0.6667-1}{-0.3704-(-1)}$

$x_3 = 0.4706$

$f(x_3) = f(0.4706) = 0.4706^3-4(0.4706) +2 = 0.2219$

3rd iteration

$x_2 = 0.6667$ and $x_3 = 0.4706$

$f(x_2) = f(0.6667) = -0.3704$ and $f(x_3) = f(0.4706) = 0.2219$

$\therefore x_4 = x_2 - f(x_2).\frac{x_3 - x_2}{f(x_3) - f(x_2)}$

$x_4 = 0.6667 -(-0.3704) .\frac{0.4706-0.6667}{0.2219-(-0.3704)}$

$x_4 = 0.544$

$f(x_4) = f(0.544) = 0.544^3-4(0.544) +2 = 0.0151$

4th iteration

$x_3 = 0.4706$ and $x_4 = 0.544$

$f(x_3) = f(0.4706) =0.2219$ and $f(x_4) = f(0.544) = -0.0151$

$\therefore x_5 = x_3 - f(x_3).\frac{x_4 - x_3}{f(x_4) - f(x_3)}$

$x_5 = 0.4706 -0.2219 .\frac{0.544-0.4706}{-0.0151-0,2219}$

$x_5 = 0.5393$

$f(x_5) = f(0.5393) = 0.5393^3-4(0.5393) +2 = - 0.0005$

5th iteration

$x_4 = 0.544$ and $x_5 = 0.5393$

$f(x_4) = f(0.544) = -0.0151$ and $f(x_5) = f(0.5393) = -0.0005$

$\therefore x_6 = x_4 - f(x_4).\frac{x_5 - x_4}{f(x_5) - f(x_4)}$

$x_6 = 0.544 - (-0.0151)\,.\frac{0.5393 - 0.544}{-0.0005 - (-0.0151)}$

$x_6 = 0.5392$

$f(x_5) = f(0.5392) = 0.5392^3 - 4(0.5392) + 2 = 0$

The approximate root of the equation $x^3-4x+2=0$ using the Secant method after 5 iterations is 0.5392

Table 3: illustration of the Secant method

N	x_0	$f(x_0)$	x_1	$f(x_1)$	x_2	$f(x_2)$	Update
1	0	2	1	-1	0.6667	-0.3704	$x_0 = x_1$ $x_1 = x_2$
2	1	-1	0.6667	-0.3704	0.4706	0.2219	$x_0 = x_1$ $x_1 = x_2$
3	0.6667	-0.3704	0.4706	0.2219	0.544	-0.0151	$x_0 = x_1$ $x_1 = x_2$
4	0.4706	0.2219	0.544	-0.0151	0.5393	-0.0005	$x_0 = x_1$ $x_1 = x_2$
5	0.544	-0.0151	0.5393	-0.0005	0.5392	0	$x_0 = x_1$ $x_1 = x_2$

Discussion

The Bisection, Newton-Raphson and Secant methods were applied to a non-linear equation $f(x) = x^3 - 4x + 2 = 0$ on a closed interval [a,b] using python programing language and the theoretical part. The results are presented in Tables 1 to 3

By focusing on the earlier result, we have proceeded with the interpretation of our used numerical methods developed by the iterations technique for the approximation solutions of non-linear equations (Burden & Faires, 2011). One problem was solved by using Bisection, Newton, and Secant Methods. The Bisection Method accumulates a proper digit value in each step of iteration and the aspect of convergence is linearly ordered demonstrated. The wonderful and fastest Newton Raphson Method is quadratically converged as considered the digital values were doubled within the iteration steps (Kiusalaas, 2009). Also, Newton's Method required calculation of the function and its derivative in each iteration step while the Secant Method needed evaluation of the function only.

Conclusion

Considering the result that we have done before, the Newton- Raphson, and Secant methods are more absolutely accurate and speedy to converge with a few steps of iterations while the Bisection method takes too much iteration to converge. It was observed that the Bisection method converges at the 13th iteration while Newton-Raphson and Secant methods converge to get an exact root of 0.5392 at the 3rd and 5th iterations respectively. It was then concluded that of the three methods considered the Secant method is the most effective to use although the Newton-Raphson method converges faster but it requires difficulty in taking a derivation.

References

Ahmad, A. G. (2015). "Comparative Study of Bisection and Newton-Raphson Methods of Root-Finding Problems". *International Journal of Mathematics Trends and Technology*, 19(2), 121-129.

Ali, W. N., Hemin M. K., Ribwar B. I and Sundws M. M.(2021). *Turkish Journal of Computer and Mathematics Education* Vol.12 No.14 (2021),1115 – 1122.

Biswa, N., D. (2012), Lecture Notes on Numerical Solution of root Finding Problems. www.math.niu.edu/dattab.

Burden, R. L., & Faires, J. D. (2011). "Numerical Analysis". Boston: Brooks/Cole, Cengage Learning. C, E. J., & O, A. S. (2014). Comparative Study of Bisection, Newton-Raphson and Secant Methods of Root- Finding Problems. *IOSR Journal of Engineering* (IOSRJEN), 04(04), 01 - 07. Cheney.

Dawson M\ . Python dla kazdego. Podstawy programowania, Helion, Warszawa 2014.

Ehiwario, J.C., Aghamie, S.O.(2014). "Comparative Study of Bisection, Newton-Raphson and Secant Methods of Root- Finding Problems ". Department of Mathematics, College of Education, Agbor, Delta State. *IOSR Journal of Engineering* (IOSRJEN) www.iosrjen.org ISSN (e): 2250-3021, ISSN (p): 22788719 Vol. 04, Issue 04 (April. 2014), ||V1|| PP 01-07.

Kiusalaas, J. (2009). "Numerical Methods in Engineering With MATLAB®". New York: Cambridge University Press.

McDonough, J.M (2001), Lecture in Computational Numerical Analysis. Dept. of Mechanical Engineering, University of Kentucky.

Suli, E., & Mayers, D. F. (2003). An Introduction to Numerical Analysis. New York: Cambridge University \Press.