Comparative study of the Bisection and the Newton Methods in solving for Zeros and Extremes of a single-variable Function

[*1]Iwetan C.N., [2]Fuwape I. A., [3]Olajide M.S., and [4]Adenodi R.A

[1,4]Physics Deparement, Adeyemi college of Education, Ondo
[2]Physics Department, Federal University of Technology Akure.
[3]Computer Science Department, Adeyemi College of Education, Ondo

**e-mail: carolineiwetan@yohoo.com 08036698068*

ABSTRACT

The Bisection and the Newton methods were employed to solve a single-variable problem using FORTRAN programming language for its implementation. The root obtained from the bisection algorithm after few iterations is compared with the root obtained from the Newton algorithm. The Newton method is found to be more efficient, having much smaller error, fast convergence rate (fewer number of iterations) and in agreement with [1] in the study of some methods of numerical evaluation (secant, bisection, and Newton) to achieve an adaptive scheme.

INTRODUCTION

In solving Physics problems, we often encounter situations in which we need to find possible values of x that ensures the equation $f(x) = 0$, where $f(x)$ can either be an explicit or an implicit function of x. If such value exists, it is called a root of the equation. An associated problem is to find the minima or maxima of a function $g(x)$. Relevant situation in Physics where such problems are needed to be solved include finding the equilibrium position of an object, potential surface of a field and the quantized energy levels of confined structure [1].

If $f(x)$ is continuous in the interval $[a,b]$ and $f(a)f(b) < 0$ then a root must exist in the interval (a,b) The last point about the interval is one of the most useful properties, numerical methods use to find the roots. Various methods converge to the root at different rates. That is, some methods are slow to converge and it takes a long time to arrive at the root, while other methods can lead us to the root faster. There is in general a compromise between ease of calculation and time. For a computer program however, it is generally better to look at methods which converge quickly. The rate of convergence could be linear or of some higher order. The higher the order, the faster the method converges. The Bisection method is one of the methods that are strongly based on the property of intervals [2].

Newton's method (also known as the Newton–Raphson method) is also an efficient algorithm for finding approximations to the zeros (or roots) of a real-valued function. This method is an example of a root-finding algorithm. Any zero-finding method (Bisection Method, False Position Method, Newton-Raphson, etc.) can also be used to find a minimum or maximum of such a function, by finding a zero in the function's first derivative [3].

In this study, we consider a single variable problem using two methods (Bisection and Newton method) by implementing two different programs, one for each method, and compare the efficiency of each method.

METHODS

Bisection Method

The bisection method is a root bracketing algorithms. Root bracketing solvers work by bracketing the zero in an interval. The target function must be positive on one end of the interval, and negative on the other end. If the target function is continuous, this guarantees that the function has a zero in the interval [4].

The Bisection method works by dividing the bracketing interval into two equal parts in each of the iteration. If there is a root in the region x ∈ [a,b] for $f(x) = 0$, the bisection method can be used to find this root within a required accuracy. Since there is a root in the region, $f(a)f(b) < 0$, we divide the region into two equal part with $x_1 = \frac{(a+b)}{2}$, so that $f(a)\, f(x_1) < 0$ or $f(x_1)\, f(b) < 0$. If $f(a)\, f(x_1) < 0$, the next stage is $x_2 = \frac{(a+x_1)}{2}$ otherwise $x_2 = \frac{(x_1+b)}{2}$. The procedure is repeated until the improvement on x_1 becomes very small.

Considering a single-variable problem: $f(x) = e^x \ln x - x^2$,………………………………..(1)

/ with solutions, $f(x) = -1$ at $x = 1$

And $f(x) = e^2 \ln 2 - 4 \approx 1$ at $x = 2$

This problem is to find a value $x_0 \in [1, 2]$ that would make $f(x_0) = 0$.

In the neighbourhood of x_0, we have $f(x_0 + \Delta) > 0$ and $f(x_0 - \Delta) < 0$.

A Fortran programme is written for the implementation of the bisection method for this problem

Newton-Raphson Method

Thomas Simpson described Newton's method is an iterative method for solving general nonlinear equations using fluxional calculus. In the some publications, Simpson gives the generalization to systems of two equations and notes that Newton's method can be used for solving optimization problems by setting the gradient to zero [5].

Newton's method can be used to find a minimum or maximum of a function. The derivative is zero at a minimum or maximum, so minima and maxima can be found by applying Newton's method to the derivative.

It has been shown that the Newton's method for finding the roots of a cubit with one double root is conjugate [6] and this method is base on linear approximation of smooth function round its root. The function *f(x) = 0* can be expanded in the neighbourhood of the root x_0 through the Tailor expansion:

$$f x_0) \approx f(x) + (x_0 - x)f'(x) + \ldots\ldots = 0 \text{..............................}(2)$$

Where x can be viewed as a trial value for the root at the nth step and the approximate value of the next step x_{n+1} can be derived from

$$f(x_{n+1}) = f(x_n) + (x_{n+1} - x_n)f'(x_n) \approx 0 \text{................................}(3)$$

that is $x_{n+1} = x_n - {f_n}/{f'_n}$ where $f_n = f(x_n)$ ....(4)

This iterative scheme known as the Newton-Raphson method is applied to a single-variable Problem; $f(x) = e^x \ln x - x^2$. Implementing a FORTRAN program [7], gives the following results

RESULTS AND DISCUSSION

Table 1: Iteration data for the Bisection Method

N	X_0	F(x)
1	1.50000000	0.50000000
2	1.75000000	0.25000000
3	1.62500000	0.12500000
4	1.68750000	0.06250000
5	1.71875000	0.03125000
6	1.70312500	0.01562500
7	1.69531250	0.00781250
8	1.69140625	0.00390625
9	1.69335938	0.00195313
10	1.69433594	0.00097656
11	1.69482422	0.00048828
12	1.69458008	0.00024414
13	1.69470215	0.00012207
14	1.69464111	0.00006104
15	1.69461060	0.00003052
16	1.69459534	0.00001526
17	1.69460297	0.00000763
18	1.69459915	0.00000381
19	1.69460106	0.00000191
20	1.69460011	0.00000095

The root obtained from the FORTRAN implementation of the bisection method

$x_0 = 1.69460011 \pm 0.00000095$ in twenty iterations.

Since the first guess is closed to the inflection point, the roots diverge from the first iteration up to the sixth iteration and then begin to converge.

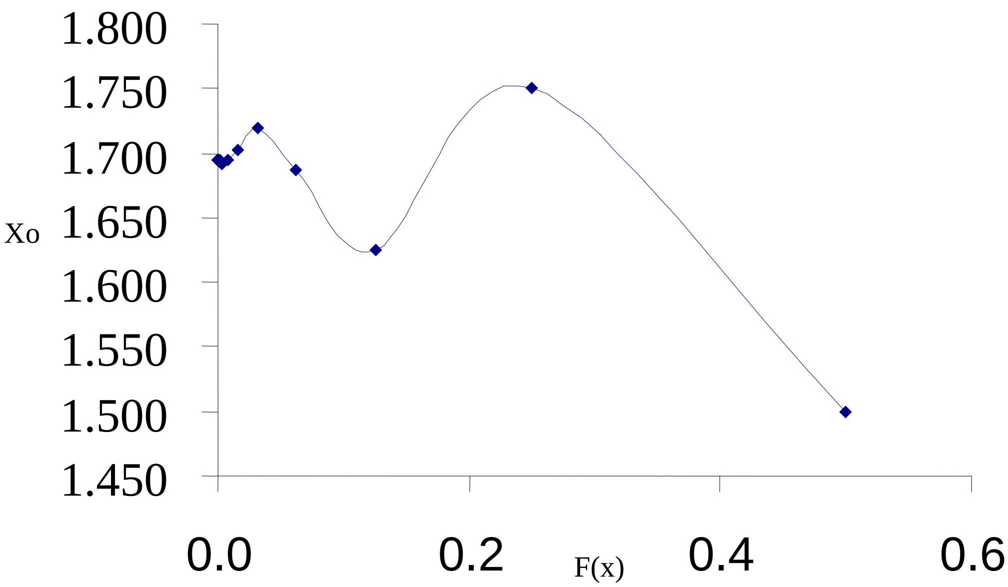

Fig1: 2-D graph of X_0 against F(x)

Figure 1 shows a wide range of divergence of F(x) before convergence towards the root of the equation after twenty iterations.

X_0 = 1.69460011 with error = 0.00000095

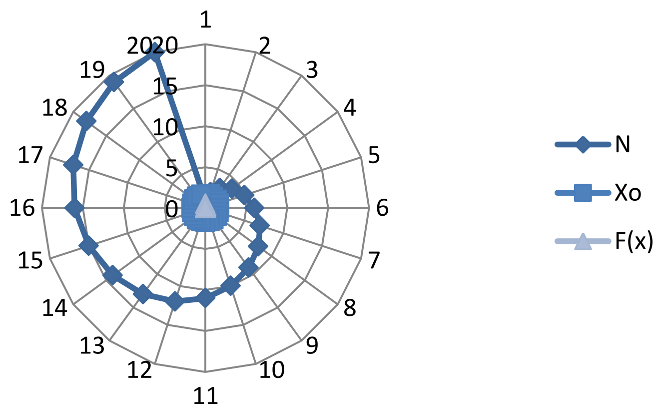

Fig. 2: Phase space diagram of bisection method

This graph shows the route of the iterations toward the root.

NEWTON-RAPHSON DATA

N	X0	F(x)
1	1.73980093	0.23980093
2	1.69656825	-0.04323268
3	1.69460475	-0.00196350
4	1.69460094	-0.00000381
5	1.69460094	0.00000000

The root obtained from the FORTRAN implementation for the Newton's method is x_0 = 1.69460094 after only five iterations. Error = 0.00000000

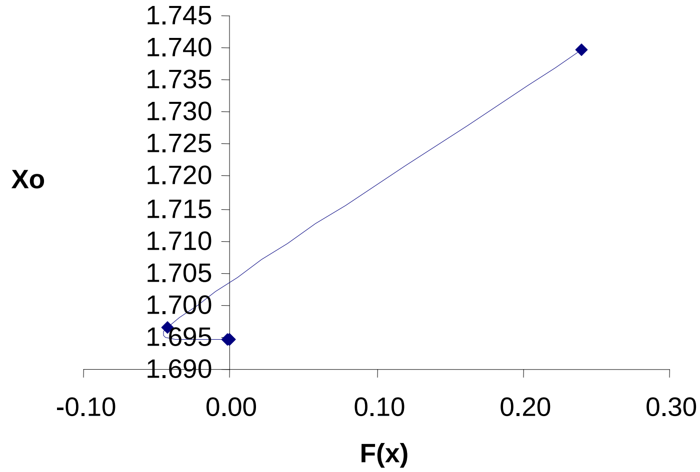

Fig. 3: 2D graph of X_0 against F(x) for the Newton-Raphson method

This graph shows a linear progression to the root of the equation

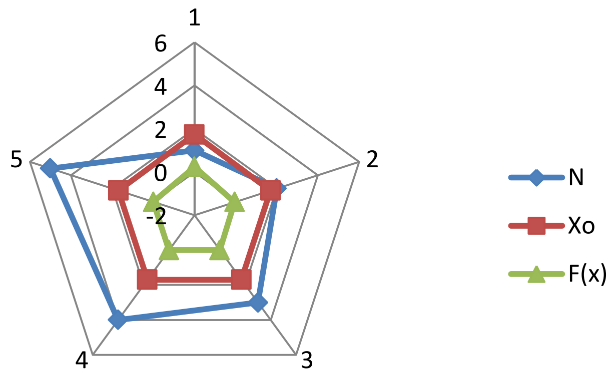

Fig. 4:
Phase space diagram for the Newton-Raphson Implementation

From the results above, there are five iterations in the Newton Raphson's method to arrive at the finite end point of the computation as implemented by the FORTRAN Compiler 95. Compared to the twenty iterations which hitherto was observed in the Bisection method using the same Fortran compiler 95, with finite solution still tends towards zero, the Newton-Raphson method is found to converge to the root faster.

The shorter the number of iterations, the more efficient the method could be as evidenced by the Newton-Raphson's method implementation using the FORTRAN programming language.

CONCLUSION

In this study, the bisection method (arriving at root 1.69460011 with error = 0.00000095) is found to be somewhat slower and hence less efficient compared with Newton-Raphson method which converges to the exact root after a few iterations. This is in agreement with [1] who studied some methods of numerical evaluation (secant, bisection, and newton) to achieve an adaptive scheme.

REFERENCE

1. Tao Pang (1996); An introduction to Computational Physics
2. Wikibooks ; Numerical Methods/Equation Solving

3. R. P. Brent, "An algorithm with guaranteed convergence for finding a zero of a function", Computer Journal, 14 (1971) 422-425
4. J. C. P. Bus and T. J. Dekker, "Two Efficient Algorithms with Guaranteed Convergence for Finding a Zero of a Function", *ACM Transactions of Mathematical Software*, Vol. 1 No. 4 (1975) 330-345
5. Wikipedia, the free encyclopedia
6. William J. Gilbert (1991);The complex dynamics of Newton’s method for a double root, Computer Math application; Vol. 22 No 10, Pp115-119
7. I.M. Smith (2001); Programming in Fortran 90 (A first Course for Engineers and Scientists).