

# Comparative Study of Bisection and Newton-Raphson Methods of Root-Finding Problems

Mandar Molawade<sup>1</sup>, Tejas patil<sup>2</sup>, Vishal V. Mehtre<sup>3</sup>

Student Of B. Tech, Department of Electrical Engineering, Bharti Vidypeeth (Deemed to Be University) College of Engineering, Pune, Maharashtra, India<sup>1,2</sup>

Assistant Professor, Department of Electrical Engineering, Bharati Vidyapeeth (Deemed to Be University) College of Engineering, Pune, Maharashtra, India<sup>3</sup>

**Abstract:** In this paper, we have given basic information on 'Bisection and Newton-raphson method' which includes derivation and their programs on matlab. newton raphson method can be also be applied to the solution of power flow problem when the bus voltage is expressed in polar form & bisection method is can be used to achieve the exact solution when all the parameters are reconstructed.

**Keywords:**Convergence, Roots, Algorithm, Iterations, Bisection method, Newton-Raphson method, and function, Matlab code

## INTRODUCTION

An expression of the form  $f(x) = a_0x^n + a_1x^{n-1} + \dots + a_{n-1} + a_n$ , where  $a^r$ 's are constants ( $a_0 \neq 0$ ) and  $n$  is a positive integer, is called a polynomial of degree  $n$  in  $x$ . if  $f(x)$  contains some other functions as trigonometric, logarithmic, exponential etc. then,  $f(x) = 0$  is called a transcendental equation. The value of  $\alpha$  of  $x$  which satisfies  $f(x) = 0$  is called a root of  $f(x) = 0$ , and a process to find out a root is known as root finding. Geometrically, a root is that value of  $(x)$  where the graph of  $y = f(x)$  crosses the  $x$ -axis. The root finding problem is one of the most relevant computational problems. It arises in a wide variety of practical applications in Sciences and Engineering. As a matter of fact, the determination of any unknown appearing implicitly in scientific or engineering formulas, gives rise to root finding problem[1]. Relevant situations in Physics where such problems are needed to be solved include finding the equilibrium position of an object, potential surface of a field and quantized energy level of confined structure[2]. The common root-finding methods include: Bisection and Newton-Raphson methods etc. Different methods converge to the root at different rates. That is, some methods are faster in converging to the root than others. The rate of convergence could be linear, quadratic, or otherwise. The higher the order, the faster the method converges [3]. The study is at comparing the rate of performance (convergence) of Bisection and Newton-Raphson as methods of root-finding. Obviously, Newton-Raphson method may converge faster than any other method but when we compare performance, it is needful to consider both cost and speed of convergence. An algorithm that converges quickly but takes a few seconds per iteration may take more time overall than an algorithm that converges more slowly, but takes only a few milliseconds per iteration[5]. In comparing the rate of convergence of Bisection and Newton's Raphson methods [8] used MATLAB programming language to calculate the cube roots of numbers from 1 to 25, using the three methods. They observed that the rate of convergence is in the following order: Bisection method < Newton's Raphson method. They concluded that Newton method is 7.678622465 times better than the Bisection method.

## 1)BISECTION METHOD

The Bisection Method is the most primitive method for finding real roots of function  $f(x) = \bar{0}$  where  $f$  is a continuous function. This method is also known as Binary-Search Method and Bolzano Method. Two initial guess is required to start the procedure. This method is based on the Intermediate value theorem: Let function  $f(x) = 0$  be continuous between  $a$  and  $b$ . For definiteness, let  $f(a)$  and  $f(b)$  have opposite signs and less than zero. Then the first approximations to the root is  $x_1 = 1/2 (a + b)$ . if  $f(x_1) = 0$  otherwise, the root lies between  $a$  and  $x_1$  Or  $x_1$  and  $b$  according as  $f(x_1)$  is positive or negative. Then we Bisect the interval

as before and continue the process until the root is found to the desired accuracy.

**Algorithm of Bisection Method for root-finding:-**

Input:

- 1)  $f(x)$  is the given function
- 2)  $a, b$  the two numbers such that  $f(a)f(b) < 0$

Output:

An approximation of the root of  $f(x) = 0$  in  $[a, b]$ , for  $k = 0, 1, 2, 3, \dots$  do until satisfied.

- 1)  $c_k = \frac{a_k + b_k}{2}$
- 2) Test if  $c_k$  is desired root. if so, stop.
- 3) If  $c_k$  is not the desired root, test if  $f(c_k)f(a_k) < 0$ . if so, set  $b_{k+1} = c_k$  and  $a_{k+1} = a_k$ . otherwise  $c_k = b_k + 1 = b_k$

**Stopping Criteria for Bisection Method:-**

The stopping criteria as suggested by [5]: that let  $e$  be the error tolerance, that is we would like to obtain the root with an error of at most of  $e$ . Then, accept  $x = c_k$  as root of  $f(x) = 0$ . if any of the following criteria satisfied

- 1)  $|f(c_k)| \leq e$  (i.e., the functional value is less than or equal to the tolerance).
- 2)  $\left| \frac{c_k - 1 - c_k}{b - a} \right| \leq e$  (i.e., the relative change is less than or equal to the tolerance).
- 3)  $\frac{b - a}{2^k} \leq e$  (i.e., the length of the interval after  $k$  iterations is less than or equal to tolerance).
- 4) The number of iterations  $k$  is greater than or equal to a predetermined number, say  $N$ .

**Theorem 1:** The number of iterations,  $N$  needed in the Bisection method to obtain an accuracy of is given by:  
 $N \geq \frac{\log(b-a) - \log(e)}{\log 2}$

**Proof:** Let the interval length after  $N$  iteration be  $(b-a)/2^N$ . So to obtain an accuracy of  $(b-a)/2^N \leq e$

$$\text{That is } \rightarrow -N \log 2 \leq \log(e/b-a) \quad N \geq \frac{-\log(e/b-a)}{\log 2}$$

$$\text{Therefore } N \geq \frac{\log(b-a) - \log(e)}{\log 2}$$

Note: Since the number of iterations,  $N$  needed to achieve a certain accuracy depends upon the initial length of the interval containing the root, it is desirable to choose the initial interval  $[a, b]$  as small as possible.

**MATLAB code of Bisection Method:-**

```
function root = bisection(fname,a,b,delta,display)
% The bisection method.
%Input: fname is a string that names the function f(x)
% a and b define an interval [a,b]
% Delta is the tolerance
% display = 1 if step-by-step display is desired,
% = 0 otherwise
%Output: root is the computed root of f(x)=0
Fa= feval(fname); fb= feval(fname,b);
if sign(fa)*sign(fb) > 0
```

```
disp('function has the same sign at a and b') return
end
if fa =0,
root = a; return
end
if fb =0
root = b; return
end
c = (a+b)/2;
fc = feval(fname); e_bound = abs(ba)/2;
if display,
disp(' ');
disp(' a b c f(c) error bound'); disp(' ');
disp([a b c fc e_bound])
end
while e_bound > delta if fc =0,
    root = c;
    return
end
if %sign(fa)*sign(f% c<0 a root %exists in [a,c];
b = c;
fb=fc;
else % a root %exists in [c,b].
    a = c;
    fa = fc;
end
c = (a+b)/2;
fc = feval(fname);
e_bound = e_bound/2;
if display, disp([a b c fc e_bound]), end
end
root = c;
return
end
if sign (fa)*sign(fc) <0% a root exists in [a,c]
    b=c;
    fb=fc;
else % a root exists in [c,b]
    a=c;
    fa=fc;
end
c= (a+b)/2;
fc=feval(fname);
e_bound=e_bound/2;
if display , disp [a b c fc e bound], end
end
root=c;
```

**Example:** I solve  $f(x) = \cos x - x \exp(x) = 0$  at  $[0, 1]$  using Bisection method with aid of the MATLAB Code.

**Solution:** Input from the MATLAB command window.

```
>>fname = @(x)(cos(x)-x*exp(x));
>> a=0;
>> b=1;
>> display=1;
>> delta=10^-4;
>> root = bisection23(fname,a,b,delta,display)
```

Output value:-

N	A	B	C	F(c)	Error bound
1.	0	1.0000	0.5000	0.0532	0.5000
2.	0.5000	1.0000	0.7500	-0.8561	0.2500
3.	0.5000	0.7500	0.6250	-0.3567	0.1250
4.	0.5000	0.6250	0.5625	-0.1413	0.0625
5.	0.5000	0.5625	0.5313	-0.0415	0.0313
6.	0.5000	0.5313	0.5156	0.0065	0.0156
7.	0.5156	0.5313	0.5234	-0.0174	0.0078
8.	0.5156	0.5195	0.5176	0.0005	0.0020
9.	0.5176	0.5195	0.5186	-0.0024	0.0010
10.	0.5176	0.5186	0.5181	-0.0009	0.0005
11.	0.5176	0.5181	0.5178	-0.0002	0.0002
12.	0.5176	0.5178	0.5177	0.0002	0.0001
13.	0.5177	0.5178	0.5178	-0.0000	0.0001

Root= 0.5178

## 2) NEWTON'S RHAPSON METHOD

Given a point  $x_0$ . Taylor series expansion about  $x_0$

$$f(x) = f(x_0) + f'(x_0)(x-x_0) + \frac{f''(x_0)}{2} (x-x_0)^2 \quad (1)$$

we can use  $l(x) = f(x_0) + f'(x_0)(x-x_0)$  as an approximation to  $f(x)$ . in order to solve  $f(x) = 0$ , we solve  $l(x) = 0$ , which gives the solution-

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \quad (2)$$

So  $x_1$  can be regarded as an approximate root of  $f(x)$ . If this  $x_1$  is not a good approximate root, we continue this iteration. In general, we have the Newton iteration:

$$X_{n+1} = X_n - \frac{f(X_n)}{f'(X_n)}, n=0,1,\dots$$

Here we assume  $f(x)$  is differentiable and  $f'(X_n) \neq 0$ .

### Newton's Method Algorithm-

Impute: Given  $f, f', x$  (initial point),  $xtol, f tol, n_{max}$  (the maximum number of iterations)

FOR  $n = 1: n_{max}$

$$d = \frac{f(X)}{f'(X)} \quad (4)$$

$X = X - d$

If  $|d| < xtol$  or  $|f(x)| < f tol$ , then

root=  $x$

STOP

END

END

root= x

**Stopping criteria**

1.  $|x_{n+1} - x_n| < \text{xtol}$
2.  $|f(x_{n+1})| < \text{ftol}$
3. the maximum number of criteria reached

**Convergence Analysis of Newton’s Method Def.** Quadratic convergence (QC). A sequence  $x_n$  is said to have QC to a limit  $x$  if

$$\lim_{n \rightarrow \infty} \frac{|x_{n+1} - x|}{|x_n - x|^2} = C,$$

where  $c$  is some finite non-zero constant.

Newton iteration:

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

Suppose  $x_n$  converges to a root  $r$  of  $f(x) = 0$ . Taylor series expansion about  $x_n$  is

$$f(r) = f(x_n) + f'(x_n)(r - x_n) + \frac{f''(z_n)}{2} (r - x_n)^2$$

where  $z_n$  lies between  $x_n$  and  $r$ .

But  $f(r) = 0$ , and dividing by  $f'(x_n)$

$$0 = \frac{f(x_n)}{f'(x_n)} + (r - x_n) + \frac{f''(z_n)}{2} (r - x_n)^2$$

The first term on the RHS is  $x_n - x_{n+1}$ , so

$$r - x_{n+1} = C_n (r - x_n)^2, \quad C_n = \frac{f''(z_n)}{2f'(x_n)}$$

Writing the error  $e_n = r - x_n$ , we see

$$e_{n+1} = C_n (e_n)^2$$

Since  $x_n$  approached  $r$ , and  $z_n$  is between  $x_n$  and  $r$ , we see  $z_n$  approached  $r$  and so

$$\lim_{n \rightarrow \infty} \frac{|e_{n+1}|}{|e_n|^2} = \lim_{n \rightarrow \infty} |C_n| = \frac{f''(r)}{2f'(r)}$$

assuming that  $f'(r) \neq 0$ . So, the convergence rate of Newton method is usually quadratic. At the  $(n + 1)^{\text{th}}$  step, the new error is equal to  $c_n$  times the square of the old error. Also, we can show  $f(x_n)$  usually converges to 0 quadratically.

**MATLAB code of Newton’s Raphson:-**

```
function root=newton(fname,fdname,x,xtol,ftol,n_max,display)
% Newton’s method.
%Input:
% fname string that names the function f(x).
% fdname string that names the derivative f'(x).
% x the initial point
% xtol and ftol termination tolerances
% nmax the maximum number of iterations
% display = 1 if step-by-step display is desired,
% = 0 otherwise
%Output: root is the computed root of f(x)=0
```

```
n = 0;
fx = feval(fname,x);
if display,
    disp(' n x f(x)')
    disp('-----')
    disp(sprintf('%4d %23.4e %23.4e', n, x, fx))
end
if abs(fx) <= xtol
    root = x;
    return
end
for n = 1:n_max
    fdx = feval(fname,x);
    d = fx/fdx;
    x = x - d;
    fx = feval(fname,x);
    if display,
        disp(sprintf('%4d %23.4e %23.4e',n,x,fx))
    end
    if abs(d) <= xtol | abs(fx) <= f tol
        root = x;
        return
    end
end
end
```

Example: I solve  $f(x) = x - \cos x = 0$  at  $[0, 1]$  using Newton Raphson method with aid of the MATLAB Code.

Solution: Input from the MATLAB command window.

```
>> fname = @(x)(cos(x)-x*exp(x));
>> fdname = @(x)(-sin(x)-(x*exp(x)+exp(x)));
>> x=0.5;
>> xtol=10^-4;
>> f tol=10^-4;
>> nmax=10;
>> display=1;
>> root=newton24(fname,fdname,x,xtol,ftol,n_max,display)
```

Output values-

n	X	F(x)
0	5.000e-001	5.3222e-002
1	5.1803e-001	-8.1744e-004
2	5.1776e-001	-1.8378e-007

Root = 5.178

## RESULT AND DISCUSSION

It was observed that the bisection method converges at the 14<sup>th</sup> iteration while newton methods converge to exact root of 0.5718

With error 0.0000 at the 2<sup>nd</sup> iteration respectively. So, these two methods considered, newton method is the most effective scheme & sometime Newton's raphson method does not always work well. It may not converge.

- If  $f'(x_n) = 0$  the method is not defined.
- If  $f'(x_n)$  approximate to 0 then there may be difficulties. The new estimate  $x_n + 1$  may be a much worse approximation to the solution than  $x_n$  is.

### CONCLUSION

Based on our results from the two methods, I now conclude that the Newton's method is formally the most effective of the methods compared with Bisection method in term of it order of convergence. In Bisection method the root is bracketed within the bound of interval, so the method is guaranteed to converged but is very slow. This is sequel to the fact that it has a converging rate close to that of Newton-Raphson method, but requires only a single function evaluation per iteration. I also concluded that though the convergence of Bisection is certain, its rate of convergence is too slow and as such it is quite difficult to extend to use for systems of equations compared to Newton's Raphson method.

### REFERENCES

- [1]. Burden, R.L., Fairs, J.D. 2011 Numerical Analysis 9th Edition, Brook/Cole, a part of Cengage Learning.
- [2]. Rudra Pratap, 2013 Getting Started With MATLAB: A Quick Introduction for Science and Engineers, Oxford University Press.
- [3]. Chadha, N.M, 2014 Lecture Note On MATLAB, Sharda University.
- [4]. Cleve B. Moler, Numerical Computing With MATLAB, PHI Learning, Rimjhim house Delhi, 2013.
- [5]. Noreen, Jamil, A Comparison of Iterative Methods for the Solution of Non-Linear Systems of equations, Int. J. Emerg. Sci., June 2013
- [6]. Biswa Nath Datta (2012), Lecture Notes on Numerical solution of root finding Problems.
- [7]. Srivastava, R.B and Srivastava, S(2011), Comparison of Numerical Rate of Convergence of bisection & Newton-Raphson methods.
- [8]. Allahviranloo T., Asari s., 'Numerical Solution of Newton-Raphson Method' Journal of applied mathematics Islamic Azad Univercity of Lahijan,, vol7
- [9]. Ehiwario, J.C., Aghamie ,S.O. " Comparative study of Bisection , Newton-Raphson methods root finding problems", IOSR journal of engineering (IOSRJEN), vol.04
- [10]. Cheney, W., & Kincaid, D. (2013). "Numerical Mathematics and Computing". Boston: Thomson Brooks/Cole.
- [11]. Gautschi, W. (2012). "Numerical Analysis". New York: Springer science+Business Media LLC
- [12]. Stoer, J., & Bulirsch, R. (2013). "Introduction to Numerical Analysis" (Vol. 12). New York: Springer Science & Business Media.
- [13]. Ahmad, A. G. (2015). "Comparative Study of Bisection and Newton-Rhapson Methods of Root-Finding Problems". International Journal of Mathematics Trends and Technology.
- [14]. Iwetan, C.N. Fuwape.LA, Olajide, M.S and Adenodi, R.A(2012) Comparative Study of the Bisection and Newton Methods in solving for Zero and Extremes of a Single-Variable Function, J. of NAMP vol.21.
- [15]. Steven C. Chapra, "Applied Numerical Methods with MATLAB or Engineers and Scientists," 3rd Edition, McGraw-Hill Companies, Inc., (2012)