Comparison Between Bisection, Newton and Secant Methods for determining the root of the Non-Linear equation using MATLAB.

Ali Wahid Nwry [1], **Hemin Muheddin Kareem** [2], **Ribwar Bakhtyar Ibrahim**[3] , **Sundws Mustafa Mohammed** [4].

[1,4]Assistant Lecturer at Sulaimani Polytechnic University, Technical College of Health, Anaesthesia.
[2]Assistant Lecturer at Sulaimani Polytechnic University, Technical College of Health, Nursing.
[3]Lecturer at Sulaimani Polytechnic University, Technical College of Informatics, Database Technology.

Abstract: The Bisection, Newton, and Secant Methods are indicated for demonstrating the numerical approximation of the non-linear equation problem $\mathrm{f(x) = e^{x} + 2^{-x} + 2\cos(x) - 6 = 0}$. the paper wants to display the comparison of implementation, the convergence rate among methods of detect the root of the non-linear equation. The MATLAB R2015b is used to determine the root-finding problem within the desired closed interval $[1,2]$ and illustrate the numerical solution compared. The Newton and Secant are speedy to converge with very small error part and requiring a few steps of iterations while the bisection method is converged with taking too much computingof iterations. Consequently, the numerical approximation solution of the methods on the sample problem interprets that the Newton and Secant are more absolutely accurate and efficient than the results achieved from the Bisection method.

Keywords: Bisection Method, Newton Raphson Method, Secant Method, algorithm of MATLAB R2015b, root of the equation, rate of convergence, relative of error.

1. Introduction

A Verity types of equations appeared in a space of mathematics, physics, chemistry and engineering is dedicated to their study. Some of these equations are simple and more of scientists and researchers have seen the linear equation $ax + b = 0$ whose solution is $x = \frac{-b}{a}$ where a and b are real numbers and $a \neq 0$. Indeed, it's more unlike to see the form of non-linear 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 formulae is radicals(Suli & Mayers, 2003).This paper dedicated to finding the root of the nonlinear equation by using some numerical methods. In our considered problem equation contained each of the exponential and trigonometric functions together within the devoted equation. Assume that the given function is continuous and bounded within the closed interval $[a, b]$ in the set of real numbers $f: R \rightarrow R$. Since we have taken throughout the paper $a < b$ then it is implying that the desired closed interval is not empty. Now we want to find the real number $\xi \in [a, b]$ provided that$f(\xi) = 0$. Consequently, if the value of ξ is exists between the interval, then it is an approximate solution to the considered equation $f(x) = 0$. In order to implement the procedure of iterations techniques we introduce each of the Bisection, Newton and Secant methods to determine an approximate solution and implanting its algorithms by using MATLAB program(Karris, 2007; Stoer & Bulirsch, 2013).The calculation of convergence is very important and absolute of error should be computed in order to guessing the speed of each method as much as presented iteration closes to its limit, more ever calculated by the number of evaluations taken in achieving an offered accuracy. The amount of convergence is calculated where the relative error decreases among sequential approximation solution.

2. Review Of Related Studies

A study that we have seen throughout our seeking to know which case of earlier researches have a relativity with our investigation is theinterpretationof numerical solution to determine the root-finding problem of the non-linear equations and consider the rate of convergence of each method,the study conductedthecomparisonbetween the used methods and considered the convergence proportion for both utilized methods by Abdulaziz G. Ahmad (Ahmad, 2015). Another paper that we have recognized is the consideration of determining the root of the equation and illustration of the convergence rate between the executed known three used methods and demonstrated its application under the implementation of Mathematica program by Ehiwario and Aghamie(C & O, 2014). Bay the way, we receivedsome of proportionality between theseinvestigationsabout root-finding problem and other researchers who want to determine that which method can give a better approximate solution for the given non-linear equation and applying it using different programs.

3. Methods

1.1. Bisection method

In the field of Numerical Analysis, the bisection method is a way to detect a root of the considered equation in the form of $f(x) = 0$ with its algorithm which is frequently bisects a given interval and choose a midpoint between a lower and upper of the closed interval so as to the process of root finding lying till to create a new subinterval. This method is easy but it is also strictly slow. Normally, give an interval $[a, b]$ such that each of the $f(a)$ and $f(b)$ should be opposite signs, after that we are going to determine a midpoint between the interval $[a, b]$. Accordingly, the process will be continuing until we get the interval that becomes to be completely small.

Consider a function $f(x) = 0$ and assume that f is continuous on the interval $[a, b]$, if $f(a) * f(b) < 0$ then the function has at least one root on the given interval $[a, b]$ .The appropriate of bisection method for implementing by MATLAB program permits to identify the root-finding problem, depended to the following intermediate value theorem:If $f(x) = 0$ is a continuous real valued-function and $x$is a mid-point value belong to the interval $[a, b]$ and if $f(a) * f(b) < 0$, then there exists a root of f within the closed interval [a, b].The process of determine the root of the equation consists of three situations provided that p is a midpoint belong to $[a, b]$;

- If $f\left(\frac{a+b}{2}\right) = 0$ or$f(p) = 0$, then the first possibility has been destinated.
- If $f(p) \neq 0$, then $f(p)$ is the same sign with $f(a)$ or $f(b)$:
- If $f(p) < 0$, then the root is between $p = \frac{a+b}{2}$ and b.
- If $f(p) > 0$, then the root is between a and $p = \frac{a+b}{2}$.

The procedure will be continuing over the new interval $[a_1, b_1]$ to find p_1 as a new midpoint (Burden & Faires, 2011)⍰.

1.1.1.Algorithm of Bisection method using MATLAB

The bisection method is the technique uses to compute the root of $f(x) = 0$ that is should be continuous function on the given interval $[a, b]$. The number of iterations n is required to decrease the interval by creating a new subinterval till to tolerance is calculated within the relative error form in Eq. (3.1).In the beginning, we will Include each of the function, interval endpoints, tolerance, and available number of iterations as follows;

```
% INPUT:
% function = Input the given function which is $f(x)$;
% $a$      = Enter the lower of the given interval;
% $b$      = Enter the upper of the given interval;
% $i$      = First step of the iteration (singular filter);
% $n$      = Iterations Number;
% $tol$    = Tolerance of error;
% OUTPUT: finding approximate solution.
Step 1  while $(i < No)$
Step 2  $p = (a + b)/2;$   (Compute $p_i$)
Step 3  if $(abs(f(p)) < TOL)$ then
          break;
     end
Step 4  if $(f(a) * f(p) > 0)$ then
      set $a = p;$
     else
       set $b = p;$
     end
Step 5 $i = i + 1;$
end
```

```
fprintf('TheRootofagivenequationis %f',p);
```

1.1.2. Rate of convergence of Bisection Method

Relative Error $= \frac{|p_n - p_{n-1}|}{|p_n|} < \epsilon$, provided that $p_n \neq 0$. (3.1)

The relative error formulae (3.1) is the perfect stopping criteria technique to compute because it becomes mostly super form to determine a predictable relative of error(Burden & Faires, 2011).

Rate of Convergence is $\mathrm{M} = \frac{e_{n+1}}{e_n^{\alpha}}$.Order of Convergence is $\alpha = \frac{\ln\left(\frac{p_{n+1}}{p_n}\right)}{\ln\left(\frac{p_n}{p_{n-1}}\right)}$, if the number of α is equal to 1 then the convergence will be linear and if the number of α is equal to 2 the convergence will be quadratic.

1.2. Newton's Method

Newton's method is one of the strong and famous methods that is used to identify the root of the equation. Assume that $f$is a continuous real valued function belong to the closed interval $[a, b]$ andsuppose that $p_0 \in [a, b]$ is some suitable initial approximation to the root x such that $f\,'(\,p_0) \neq 0$ and the absolute value of $|\,\mathrm{p} - p_0|$ is small then by given the first Taylor polynomial for $f(p)$ developed about p_0 and computed at p.

$$f(x) = f(x_0) + (x - x_0)f'(x_0) + \frac{(x-x_0)^2}{2!} f''(\xi(x)), \tag{3.2}$$

Where $\xi(x)$ lies between x and x_0. And since $f(x) = 0$, the Taylor polynomial gives us

$$0 = f(x_0) + (x - x_0)f'(x_0) + \frac{(x-x_0)^2}{2!} f''(\xi(x)), \tag{3.3}$$

And because the absolute value of $(\mathrm{x} - x_0)$ is small then the part of $(x - x_0)^2$ also too small, so

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}, \text{ for } n = 1,2,3 \ldots. \tag{3.4}$$

achieved by cutting the Taylor expansion of the function f at x_0 after that we can obtain the result of the linear equation which is called the approximate solutionx_n as interpreted in (3.4). Consequently, Newton's Method can extremely emphasize to nonlinear equations types, not only to singular equations but also used to apply over the system of non-linear equations(Gautschi, 2012; Kiusalaas, 2009).

1.2.1.Algorithm of Newton' Method

The following some steps is the algorithm of Newton Raphson Method that the root could be calculated within the given desired closed interval [a, b] in the problem. The midpoint of the interval must be computed and it is used as the initial approximate solution of the root of the given differential function. Bay the way, the function and its derivative will be considered by the user.

```
% Newton Raphson Method
% Determine the roots of the given equation.
% INPUT
% p            = Input the value of the Initial condition;
% Tol          = Input the amount of desired tolerance;
% i            = Input the value of the first iteration;
% dp           = Input any random value so as to the while loo will be run;
% f            = Input the value of f(p);
% fprintf      = ('Step    pdpf(p)\n')
% fprintf('------ ---------- ---------    ----------\n')
%fprintf('%3i %12.8f %12.8f %12.8f\n',i,p,dp,f)
OUTPUT: finding approximate solution.
Step 1   while (dp > Tol || abs(f) > Tol)   % Remember that each of fand dpmust be defined in order to perform this statement.
Step 2   i = i + 1;
Step 3  $f$prime = Input the considered function $f(p)$;
Step 4  $pnew = p - (f/f$prime); % Calculate the new value of $p$.
Step 5  $dp = abs(p - p$new);   % Execute how much $p$has to be performed till the last step
Step 6  $p = p$new;
Step 7  input the given function $f(p)$;   % Find the new value of $f(p)$.
% $fprintf$('%3i %12.8f %12.8f %12.8f\n',$i, p, dp, f$)
End
```

1.2.2.Convergence of Newton's Method

In this part of convergence, we want to identify the convergence of the fastest method which is Newton Raphson Method. Now, we need to describe the error ε_n and ε_{n+1} which is the difference between the root denoted by p_r and n^{th} and $(n+1)^{th}$ iterations, as; Error part $\varepsilon_n = |p_r - p_n|$, then by using Taylor Series expansion we can get;

$$0 = f(p_r) = f(p_n + \varepsilon_n) = f(p_n) + f'(p_n)(p_r - p_n) + \frac{f''(p_n)(p_r-p_n)^2}{2!} + \frac{f'''(\xi)(p_r-p_n)^3}{3!} \quad (3.5)$$

For some ξ is located between p_n and p_r. So, it is time to use the Newton's Method in terms of n^{th} iterations scheme for the error, then we can subtract both sides of Newton's Method from the root p_r and we will get;

$$p_r - p_{n+1} = p_r - p_n + \frac{f(p_n)}{f'(p_n)}$$

and because then $p_r - p_n$ is too small then the above formula becomes;

$$f(p_n) = f'(p_n)(p_r - p_{n+1}) \quad (3.6)$$

$$0 = f'(p_n)(p_r - p_{n+1}) + \frac{f''(p_n)(p_r - p_n)^2}{2}$$

Since we have $\varepsilon_n = |p_r - p_n|$ then we can consider $\varepsilon_n = (p_r - p_n)$ and $\varepsilon_n = (p_r - p_{n+1})$.

$$\varepsilon_{n+1} = -\frac{f''(p)}{2f'(p)}\varepsilon_n^2$$

this convergence form is for order two which is known as (quadratic formula).

1.3. Secant Method

While the Newton Raphson Method is rapid, it has involved some difficulty like seeking to find the derivative of some implicit function f in order to use it within the formula of Newton's Method. Consequently, maybe determine the derivate for some function is not easy and workable in the real life. So, to solve this situation we need to depend to the two tangent line to the graph of $y = f(x)$ with the two tangent points $(p_0, f(p_0))$ and $(p_1, f(p_1))$ then the process of Secant method formula begin with the two initial approximate solution p_0 and p_1 then p2 is the x-axis intersection of the tangent line of a function $f$combining at both $(p_0, f(p_0))$ and $(p_1, f(p_1))$ and the third initial approximate solution p_3 is determined based on using the tangent line to the graph $f$combining at $(p_1, f(p_1))$ and $(p_2, f(p_2))$ and so on. By the way, we can avoid to taking a derivative to the function f because it is known that sometimes this operation is terrible and maybe it is seldom for some complicated functions(Cheney & Kincaid, 2013).In the technique of Secant Method and depended to the interpolation in the procedure of approximating tangent line. Suppose we have two initial approximate solutions like p_0 and p_1,then we should to consider a linear function formula, as;

$$\text{when} f(x) = p_0 + p_1 x \,, (p_0, f(p_0)) \text{ and } (p_1, f(p_1))$$

When the Secant line intersect the function f in both two points $(p_0, f(p_0)$ and $(p_1, f(p_1))$ or more then by using point-slope equation for a line, we can get;

$$f(x) = \frac{f(p_1) - f(p_0)}{p_1 - p_0}(x - p_0) + f(p_0)$$

Since the Secant line intersect the x-axis in a point, then left hand of the above equation is equal to zero, as

$$0 = \frac{f(p_1) - f(p_0)}{p_1 - p_0}(x - p_1) + f(p_1)$$

We solve the equation above with respect to x, as

$$x = p_0 - f(p_1)\frac{p_1 - p_0}{f(p_1) - f(p_0)}$$

Now, we use p_2 instead of x in this equation, yields;

$$p_2 = p_0 - f(p_1)\frac{p_1 - p_0}{f(p_1) - f(p_0)}$$

$$p_n = p_{n-1} - f(p_{n-1})\frac{p_{n-1} - p_{n-2}}{f(p_{n-1}) - f(p_{n-2})} \quad \text{where } n = 2,3,4 \ldots \qquad (3.7)$$

The equation of (3.7) is the desired formula that used to apply the Secant method in the MATLAB program. Fortunately, the rate of convergence in the Secant method is mostly quick compare to the bisection method and also within this method do not require to take the derivative of the function as needed and desired in Newton Raphson Method. Note that, the Secant method will be diverged if $f(p_n) = f(p_{n-1})$ and also if the $x - axis$ coordinate is the tangent to the given Non-linear function, then the method will not be converged to the solution(Conte & Boor, 1980). In this part of our investigation, we use the Secant Method formula and illustrate its algorithm.

1.3.1.Algorithm of Secant Method

In order to determine the SecantMethod approximate solution to $f(x) = 0$ and provided that we have two initial conditionsp_0 and $p_1$which was taken within the given interval then the procedure starts with entering the value of the both initial approximations, desired tolerance, and the required number of iterations.

```
% Newton Raphson Method.
% Determine the roots of the given equation.
% INPUT
% $p0$              = Input the value of the first initial condition;
% $p1$              = Input the value of the second initial condition;
% Tol               = Input the amount of desired tolerance;
% $i$               = Input the value of the first iteration;
% $f$               = Input the value of $f(p)$;
% $fprintf$         = ('Step      $pdpf(p)\n'$)
% $fprintf$('------ ----------- ---------    ----------$\n'$)
%$fprintf$('%3i %12.8f %12.8f %12.8f\n',$i, p, dp, f$)
OUTPUT: finding approximate solution.
Step 1   put $i = 0$;
Step 2   while ($i <= n$)
Step 3   put $p_2 = p_1 - (p_1 - p_0) * f(p_1)/(f(p_1) - f(p_0))$; % Compute the new value of $p$.
Step 4   $if\ (abs(p_2 - p_1) < tol)$ %Execute how much $p$ has to be implied till to the last step.
         break;
         end
Step 5   $i = i + 1$;
Step 6   put $p_0 = p_1$;
           $p_1 = p_2$;
Step 7 $fprintf('\%3i\ \%12.8f\ \%12.8f\backslash n', i, p2, f(p2))$
         end
```

1.3.2.Convergence of Secant Method

The approximate solution p_n of the Secant method will be converged to the desired function $f(x)$ provided that the given both initial conditions are appropriately close to the predictable root of the of the equation. By the way, the order of convergence of secant method consists of α since $\alpha = \frac{1+\sqrt{5}}{2} \approx 1.618$. in specific situation, this type of convergence is not absolutely superliner convergence. Consequently, the result becomes to be happened if the considered function could be continuously differentiable for twice times and the root of the problem should be simple. In general, if the initial condition of the desired function is not quite close to the root, then the Secant method will not be converged significantly(C & O, 2014).

4. Results

We have been applied each of the Bisection, Newton and Secant Methods to find an approximate solution to $f(x) = e^x + 2^{-x} + 2\cos(x) - 6 = 0$ which is continuous within $1 \leq x \leq 2$. Using the MATLAB software and the result considered in the table 1 to table 3.

Table 1: Illustration of Bisection Method Iterations.

Iterations	p_n	$f(p_n)$	Relative Error (ϵ)	α: Order of p_n	M:Rateof Convergence.
0	1.50000000	-1.02328314	-	-	-
1	1.75000000	-0.30458766	0.14285714	-	-
2	1.87500000	0.19437904	0.06666667	0.86499276	-
3	1.81250000	-0.06827443	0.03448276	1.07735506	0.3588496
4	1.84375000	0.05963743	0.01694915	0.96394871	0.6377785
5	1.82812500	-0.00515671	0.00854701	1.01865967	0.4353361
6	1.83593750	0.02702888	0.00425532	0.99083135	0.5441388
7	1.83203125	0.01088346	0.00213220	1.00153498	0.4766016
8	1.83007812	0.00285022	0.00106724	1.00078111	0.5052829
9	1.82910156	-0.00115652	0.00053390	1.00114862	0.5029454
10	1.82958984	0.00084603	0.00026688	0.99942262	0.5042118
11	1.82934570	-0.00015545	0.00013346	1.00028885	0.4976965
12	1.82946777	0.00034524	0.00006672	0.99997380	0.5012567
13	1.82940674	0.00009490	0.00003336	0.99966952	0.4998498
14	1.82937622	-0.00003028	0.00001668	1.00027255	0.4983895
15	1.82939148	0.00003231	0.00000834	0.99998195	0.5014972
16	1.82938385	0.00000102	0.00000417	1.00189508	0.4998965
17	1.82938004	-0.00001461	0.00000208	1.00190043	0.5112066
18	1.82938194	-0.00000682	0.00000104	0.98117220	0.5112405
19	1.82938290	-0.00000288	0.00000052	1.04617772	-
20	1.82938337	-0.00000095	0.00000026	-	-

Table 2: Illustration of Newton's Method Iterations.

Iterations	p_n	Relative Error	$f(p_n)$
1	1.92990232	0.42990232	-1.02328314
2	1.83908356	0.09081876	0.44840947
3	1.82970829	0.00937528	0.04010912
4	1.82939219	0.00031610	0.00133205
5	1.82938383	0.00000836	0.00003521
6	1.82938361	0.00000022	0.00000092
7	1.82938360	0.00000001	0.00000002
8	1.82938360	0.00000000	0.00000000

Table 3: Illustration of Secant Method Iterations.

Iterations	p_n	Relative Error	$f(p_n)$
1	2.87738838	-	9.97329650
2	1.42182266	1.02373226	-1.18524516
3	1.57643087	0.09807484	-0.83829989
4	1.95000003	0.19157393	0.54714228
5	1.80246916	0.08184932	-0.10792497
6	1.82677547	0.01330558	-0.01067389
7	1.82944323	0.00145824	0.00024459
8	1.82938347	0.00003267	-0.00000054
9	1.82938360	0.00000007	-0.00000000

5. Discussion

By focusing on the earlier result, we have proceeded on the interpretation of our used numerical methods developed by the iterations technique for the approximation solutions of non-linear equation(Burden & Faires, 2011). One problem solved by using Bisection, Newton and Secant Methods. The Bisection Method accumulate 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 was doubled within the iteration steps(Kiusalaas, 2009). Also, the Newton's Method required to calculate the function and its derivative in each iteration step while the Secant Method needed evaluation of the function only.

Finally, the absolute error illustrated in Tables 1, 2 and 3,clearly considered that the Bisection Method demanded the 20th iterations, the Newton Raphson Method needed 8th iterations, and The Secant Method required 9th iterations which is indicated that the Secant Method much more acerated than the other two methods.

6. Conclusion

Depended to the result that we have done before, we conclude that the Secant method is much more and strictly converged method compare to the other two methods. Although Newton's Method required difficulty in taking a derivation but also it is extremely closed to engage approximate solution with desiring a few steps of iterations. Finally, Bisection method is too slow and needed more iterations to destinate its approximate solution.

7. References

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

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, W., & Kincaid, D. (2013). *"Numerical Mathematics And Computing".* Boston: Thomson Brooks/Cole.

Conte, S. D., & Boor, C. D. (1980). *"Elementary Numerical Analysis: An Algorithmic Approach".* New York: McGraw-Hill.

Gautschi, W. (2012). *"Numerical Analysis".* New York: Springer Science+Business Media, LLC.

Karris, S. T. (2007). *"Numerical Analysis Using MATLAB and Excel".* Berkeley, California: Orchard Publications.

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

Stoer, J., & Bulirsch, R. (2013). *"Introduction to Numerical Analysis"* (Vol. 12). New York: Springer Science & Business Media.

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