A COMBINATORIAL APPROACH TO DECIDE INITIAL ROOT VALUE FOR THE SOLUTION OF NON-LINEAR SYSTEM OF EQUATIONS

Gananathan Ramaiyan¹, Sulaiman.S.M² and Irene Hepzibah.R³

¹School of Engineering, Asia Pacific University of Technology and Innovation, Bukit Jalil, Kuala Lumpur, Malaysia; E-mail: gananathan@staffemail.apu.edu.my
²School of Computing, Asia Pacific University of Technology and Innovation, Bukit Jalil, Kuala Lumpur, Malaysia; E-mail: sulaiman.s.m@apu.edu.my
³Department of Mathematics, T. B. M. L. College, Porayar, Tamil Nadu State, India

E-mail: ireneraj74@gmail.com

Abstract. This study aims at developing the real root of an algebraic or a transcendental equation based on non-linear model and a combinatorial numerical approach for solving nonlinear equations. The introduction of Bisection, False-Position and Newton–Raphson methods to identify the real positive root of an equation which is provided here. The rate of convergence is fastest in Newton–Raphson method due to the quadratic convergence compared with the other two iterative methods. We have applied here the three iterative methods in solving nonlinear equations in MATLAB environment and compared the outcomes of results as well as geometrically also. Finally, an illustrative algebraic example is discussed, and the MATLAB outputs are provided to check the validity of the new approach.

Keywords: Bisection method, False-Position method, Newton–Raphson method, Non-Linear equations and Rate of Convergence.

1. Introduction

We can find the roots of an algebraic or transcendental equation by using 3 iterative methods in Numerical methods namely Bisection, False position and Newton–Raphson method. Applications in engineering is to determining the root value of \( t \) in Fourier analysis of the instantaneous value of a waveform, the value of \( t \) in damped oscillation of a system and determining the value of \( t \) in motion of a particle in an electrostatic field which is represented as \( y = f(t) \). This area of research has been studied by many researchers. Claudio Gutierrez et al studied about Complexity of the bisection method in 2007. Saba Akram et al [7] wrote a research paper about Newton Raphson Method which is all-inclusive to solve the non-square and non-linear problems. The study also compared the rate of performance, rate of convergence of Bisection method, root findings of the Newton method and False Position method. Later a Comparative Study of Bisection, Newton-Raphson and False Position Methods of Root Finding Problems were done by Ehiwario, J.C [5]. In a more recent review, Ali Ebrahimnejad et al [2] proposed a revisit of numerical approach for solving linear fractional programming problem in a fuzzy environment. Extensive studies have been done and many methods of solving nonlinear equations are also available in the literature [1,3,4,6,7]. In this paper we consider nonlinear equations and operational methodologies for solving the problem are provided. We have applied here three iterative methods in solving nonlinear equations and compared the outcomes with the help of results attained and their MATLAB outputs.
The paper is organized as follows. In section 2, the preliminaries are briefly introduced and the perception of nonlinear equations with essential definitions and notions with iterative methods are explained. An application of these results and their MATLAB outputs are specified by numerical illustration in section 3 and some concluding remarks are given in section 4.

2. Preliminaries

2.1. Algebraic Equation

An equation which involves only function of \( x \) is called an algebraic equation.

2.2 Transcendental Equation

A transcendental equation is an equation containing a transcendental function of the variable(s) being solved for. It contains the exponential functions, the logarithmic functions and the trigonometric functions.

2.3 Root

A real number \( \square \) which satisfies the real root of the equation \( f(x) = 0 \) if and only if \( f(\square) = 0 \).

2.4 Intermediate value theorem

Let \( f(x) \) be a continuous function in \([a, b]\). Let \( f(a) \square 0 \) and \( f(b) \square 0 \) then \( f(x) = 0 \) will have at least one root lies between \( a \) and \( b \). This is depicted in figure1 shown below:

3. Iterative methods

There are three iterative methods (repetitive methods) are going to discuss and the corresponding figure are explained as below: They are:
3.1. Bisection Method or Bolzano Iterative method
3.2. False Position Method or Regula-Falsi method
3.3. Newton-Raphson Method or Method of Tangents
3.1. The Bisection method (or) The Bolzano method (or) Interval Halving method

Consider the equation \( f(x) = 0 \). Assume the value \( a \) and \( b \) so that \( f(a) \leq 0 \) and \( f(b) \geq 0 \) or vice versa. There is a change of sign between \( a \) and \( b \) so that the root lies between \( a \) and \( b \). Let \( a < b \) then

\[
\frac{a+b}{2} \text{ find } x_1 = \text{ and } f(x_1) = 0. \text{ Therefore } x_1 \text{ is the root of } f(x) = 0 \text{ otherwise: 2}
\]

(i) \( f(x) \leq 0 \), the root lies between \( x_1 \) and \( b \). \( x_2 = \frac{x_1 + b}{2} \) and find \( f(x_2) = 0 \) and so on (or)

(ii) \( f(x) \geq 0 \), the root lies between \( a \) and \( x_1 \). \( x_2 = \frac{a + x_1}{2} \) and find \( f(x_2) = 0 \) and so on. Proceeding in 2 this way until the last two consecutive roots are correct up to the desired accuracy equal, stop the iterations. This method is known as the Bisection or Bolzano method.

For example, identify the real positive root of an algebraic equation \( f(x) = -x^3 + x^2 - 1 = 0 \). The MATLAB output results are depicted in table 3.1.1 and corresponding graph is shown in figure 3.1.2.

| Iteration | a(ve) | b (+ve) | c=(a+b)/2 | f(c) | Sign of \( f(c) \) |
|-----------|-------|---------|-----------|------|-----------------|
| 1         | 0     | 1       | 0.5       | 0.625| negative        |
| 2         | 0.5   | 1       | 0.75      | 0.016| negative        |
| 3         | 0.75  | 0.875   | 0.813     | 0.436| positive        |
| 4         | 0.75  | 0.875   | 0.813     | 0.197| positive        |
| 5         | 0.75  | 0.813   | 0.781     | 0.087| positive        |
| 6         | 0.75  | 0.781   | 0.766     | 0.035| positive        |
| 7         | 0.75  | 0.766   | 0.758     | 0.009| positive        |
| 8         | 0.75  | 0.758   | 0.754     | 0.003| negative        |
| 9         | 0.754 | 0.758   | 0.756     | 0.003| positive        |
| 10        | 0.754 | 0.756   | 0.755     | 0.001| positive        |
3.2. The Regula-Falsi method (or) The method of False-Position:

Consider the equation \( f(x) = 0 \). Assume the value \( a \) and \( b \) so that \( f(a) \neq 0 \) and \( f(b) \neq 0 \) or vice versa. There is a change of sign between \( a \) and \( b \) so that the root lies between \( a \) and \( b \). The graph of \( y = f(x) \) crosses x-axis and \( y = 0 \). The equation of the tangent line joining between two points \( A \) and \( B \) is:

\[
y - f(a) = \frac{f(b) - f(a)}{b-a}(x-a)
\]

The first approximation to the root is \( x = x_1 \) and in x-axis \( y = 0 \)

\[
\begin{align*}
\text{B is: } y &= y_{0}m \ (x = x_{0}), \text{ where } m \text{ is the slope of tangent line. } m &= \frac{dy}{dx} |_{(x_0, y_0)} \\
y - f(a) &= \frac{f(b) - f(a)}{b-a}(x-a)
\end{align*}
\]

The first approximation to the root is \( x = x_1 \) and in x-axis \( y = 0 \)

\[
\begin{align*}
\text{Box 1: } f(a) &= \frac{f(b) - f(a)}{b-a} \\
\text{Box 2: } a &= \frac{b-a}{f(b) - f(a)} \\
\text{Box 3: } x &= a - \frac{f(a)}{b-a} \\
\text{Box 4: } x_{n+1} &= \frac{af(b) - bf(a) + af(a)}{b-a}
\end{align*}
\]
If \( f(x) \leq 0 \), then \( x \) is \( a \) and \( f(x) \) is \( f(a) \).

If \( f(x) \geq 0 \), then \( x \) is \( b \) and \( f(x) \) is \( f(b) \). Proceeding like this we will get the root \( x_2, x_3 \) and so on. Finally we will see that the two successive roots are equal to up to the desired accuracy then stop the iterations. This method is called the method of false position. The procedure is geometrically depicted in figure

![Figure 3.2.1.](image)

For example, identify the real positive root of an algebraic equation \( f(x) = x^3 - x^2 - 10 \). The MATLAB output results are depicted in table 3.2.1. and corresponding graph are as shown in figure 3.2.1.

| Iteration | a(ve) | b(+ve) | f(a) | f(b) | c | f(c) |
|-----------|-------|--------|------|------|---|------|
| 1         | 0     | 1      | -1   | 1    | 0.5| 0.625|
| 2         | 0.5   | 1      | 0.625| 1    | 0.692| 0.189|
| 3         | 0.692 | 1      | 0.189| 1    | 0.741| 0.043|
| 4         | 0.741 | 1      | 0.043| 1    | 0.752| 0.009|
| 5         | 0.752 | 1      | 0.009| 1    | 0.754| 0.002|
| 6         | 0.754 | 1      | 0.002| 1    | 0.755| 0 |
3.3 The Newton-Raphson method (or) The method of Tangents:

Consider the equation \( f(x) = 0 \). Draw the tangent to the curve and it meets at point \( P \) to the curve and meets the x-axis is \( (x_{n+1}, 0) \).

The equation of tangent at a point \( P(x_n, f(x_n)) \) is\( y - f(x_n) = \frac{dy}{dx}(x_n) (x - x_n) \), where \( \frac{dy}{dx} \) is the slope of tangent line.

\[
m = \frac{dy}{dx}(x_n) = f'(x_n) = \frac{f(x_n) - f(x_{n+1})}{x_{n+1} - x_n}
\]

Since \( (x_{n+1}, 0) \) lies in equation (1), we get

\[
-0f(x_n) = f'(x_n)(x_n - x_{n+1}) \Rightarrow f(x_{n+1}) = -x_nf'(x_n)
\]

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

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

\[
x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}\]
This is known as the Newton-Raphson method. Taking the initial root value choose from the root lies between a and b in \([a, b]\). Substituting the root value in the formula consecutively we will get the root which is equal to the desired accuracy.

The rate of convergence is faster in Newton-Raphson method compare with the other two methods. The condition for convergence of Newton-Raphson method is similar to the iteration method:
Newton Raphson method has quadratic convergence.

Suppose we decide the initial root value as

\[ x_0 = 0.500 \]

\[ f(x) \]

\[ f'(x) \]

\[ f'(x)^2 \]

We are deciding the initial root value from

Regula-Falsi method as \( x_0 = 0.500 \) and check

with condition that satisfy the relation

\[ f(x)f'(x) \]

\[ f'(x)^2 \]

\[ f(0.5)f'(0.5) \]

\[ f(0.5)^2 \]

Table 3.3.1

| Newton-Raphson Method |
|-----------------------|
| Iteration | x | x(n+1) |
|-----------|---|--------|
| 1         | 0.5 | 0.857  |
| 2         | 0.857 | 0.764  |
| 3         | 0.764 | 0.755  |
| 4         | 0.755 | 0.755  |

For example, identify the real positive root of an algebraic equation \( f(x) = x^3 - x^2 + 1 \). The MATLAB output results are depicted in table 3.3.1 and the corresponding graph are shown in figure 3.3.2.

\[ f(x) = x^3 - x^2 + 1 \]

\[ f'(x) = 3x^2 - 2x \]

\[ f''(x) = 6x \]

The rate of
convergence is: \[ |f(x) - f'(x)|^2 \]

We can decide any value from [0, 1] must satisfy the relation.

Suppose we decide the initial root value as \( x_0 = 0.5 \) and \( f(0.5) f'(0.5) \) are given by the equation \( f(x) = f'(x) = x^2 + x - 1 \).

\[
\begin{align*}
|0.625(5) - 1.75| & = 0.625 \\
|-3.125| & = 3.125
\end{align*}
\]

We are deciding the initial root value from Regula-Falsi method as \( x_0 = 0.5 \).

3.4 Numerical Example:

The critical speed of oscillation of a loaded beam is denoted by \( x \) are given by the equation \( f(x) = x^3 + x^2 - 1 \).

Identify the real root of critical speed of oscillation of a loaded beam given in an algebraic equation \( f(x) = x^3 + x^2 - 1 \), correct to 3 decimal places writing MATLAB code:

(a) the Bisection method,
(b) the Regula-Falsi method, and
(c) the Newton-Raphson method, taking the initial root value from Regula-Falsi method,
(d) Identify with reason which method converges fastest. Compare the results with other two methods in table and project these values in graph.

### 3.5 MATLAB Programming code:

Finding the root of the following function:

- \( f(x) = x^3 + x^2 - 1 \)
- \( f(x) = 4 \times x^2 - 6 \times x - 7 \)

Finding the initial guess for:

- \( i = -5:5 \)
- \( low = i; \)
- \( high = i+1; \)

if \( f(low) \times f(high) < 0 \)

break;  end  

% Bisection Method

```matlab
fprintf('Bisection Method');
i = 1;  a = low;  b = high;  c = (a + b)/2;
while abs(c - prevc) >= 1e-3    if ( i ~= 1)
    prevc =c;    end    c = (a + b)/2;    x(i) = c;    sgn = '-ve';
    if f(c) > 0        sgn = '+ve';
    end   fprintf('%d %t%.3f %t%.3f %t%.3f %t%.3f %t%s%n',i,a,b,c,f(c),sgn);
    if f(a)*f(c)<0         b = c; else        a = c;              end    i = i + 1;

end
```

% Plot the results

```matlab
figure('Color','White')
plot(1:i,x(1:i-1),'*-b') hold on
title('Root of: $f(x) = x^3 + x^2 - 1$','FontSize', 20,'Interpreter','latex')
xlabel('Iteration','FontSize',16)
ylabel('$x$','FontSize',16,'Interpreter','latex') bs =x;
```

% Regula-Falsi Method

```matlab
fprintf('Regula-Falsi Method');
i = 1;  a = low;  b = high;
c = (a * f(b) - b * f(a)) / (f(b) - f(a));
x(i) = c;  prevc = c + 10; while abs(c - prevc) > 1e-3    if ( i ~= 1)
    prevc =c;    end
```
\[ c = \frac{a \cdot f(b) - b \cdot f(a)}{f(b) - f(a)}; \]
x(i) = c;
fprintf('\n\t\t\t\t%.3f\n',i,a,b,f(a),f(b),c,f(c));
if f(a)*f(c)<0         b = c;    else        a = c;              end    i = i + 1; end
% Plot the results
plot(1:i,x(1:i),'*-g')
hold on rf = x;
Newton-Raphson Method fprintf("\nNewton-Raphson Method\n\nFirst-Order derivative of f fp = @(x) 3*x^2 + 2*x; x0 = 1; % Initial guess
N = 10; % Maximum number of iterations tol = 1e-3; % Convergence tolerance
x = zeros(N + 1,1); % Pre allocate solution vector where row => iteration
x(1) = x0; % Set initial guess n = 2;
nfinal = N + 1; % Store final iteration if tol is reached before N iterations
while (n <= N + 1)     fe = f(x(n - 1));     fpe = fp(x(n - 1));     x(n) = x(n - 1) - fe/fpe;
        fprintf('\n\t\t\t\t%.3f\n',n-1,x(n-1),x(n));
if (abs(fe) <= tol)
    nfinal = n; % Store final iteration
break;    end    n = n + 1; end
% Plot the results plot(1:nfinal,x(1:nfinal),'*-r')

4. Results and Discussion

Comparison the results with three iterative methods:

| Iterations | N-R method | R-F method | Bisection method |
|------------|------------|------------|------------------|
| 1          | 0.857      | 0.5        | 0.5              |
| 2          | 0.764      | 0.692      | 0.75             |
| 3          | 0.755      | 0.741      | 0.875            |
| 4          | 0.755      | 0.752      | 0.813            |
| 5          | 0.754      | 0.781      |                  |
| 6          | 0.735      | 0.766      |                  |
| 7          | 0.755      | 0.758      |                  |
The following MATLAB Figure shows the convergence of the three methods and the number of iterations to be taken to obtain the root. It can be seen that the error of the Bisection method are frustrated along the iteration. While the errors of the Regula-Falsi and Newton Raphson method decrease when the number of iteration increases. The result shows that the Bisection method presents the biggest number of iterations and time in all cases. However, the Regula Falsi method have fairly similar results whereas the Newton Raphson method have the smallest number of iterations and have the smallest time in most cases.

![Figure 4.2.1](image)

From the figure 4.2.1, we see that the Newton-Raphson method is the fastest in the rate of convergence. Comparing Newton Raphson with Bisection method 2.5 times faster and with RegulaFalsi method 1.5 times faster.

**Conclusion**

In this work, we have presented the three iterative methods in solving non-linear equations in MATLAB environment. The MATLAB outputs of the optimal solutions are also provided to check the effectiveness of the proposed method and compared the outcomes. The solution methodology offered here will be useful for many real life problems in engineering. This research work can be extended to solve non-linear equations in fuzzy environment in the future.
References

[1] Abdulaziz G. Ahmad, Comparative Study of Bisection and Newton-Rhapson Methods of RootFinding Problems, International Journal of Mathematics Trends and Technology, 19 (2015) 121-129.

[2] Ali Ebrahimnejad et al., A revisit of numerical approach for solving linear fractional programming problem in a fuzzy environment, Applied Mathematical Modelling, 57 (2018) 459–47.

[3] Ching-Long Su et al., A two-stage algorithm integrating genetic algorithm and modified Newton method for neural network training in engineering systems, Expert Systems with Applications, 38 (2011) 12189–12194.

[4] Claudio Gutierrez et al., Complexity of the bisection method, Theoretical Computer Science, 382 (2007) 131–138.

[5] Ehiwario, J.C., Aghamie, S.O, Comparative Study of Bisection, Newton-Raphson and Secant Methods of Root-Finding Problems, IOSR Journal of Engineering, 04 (2014) 01-07.

[6] Khalid Ali Hussein et al., Parallel Hybrid Algorithm of Bisection and Newton-Raphson Methods to Find Non-Linear Equations Roots, IOSR Journal of Mathematics, 11(2015) 32-36.

[7] Saba Akram et al., Newton Raphson Method, International Journal of Scientific & Engineering Research, 6(2015) 1748-1752.