

JUNE 21, 2022

NUMERICAL COMPUTING

ASSIGNMENT#4



UMER AHMED

BCS203182

Section#4

The Applications of Numerical Approximation Methods upon Digital Images

Abstract:

Numerous problems arise in diverse areas of science and engineering, as well as from the physical, computer, biological, economic, and even social sciences. The need for choosing such an application is more clearly and concisely demonstrate how shall the numerical technique be applied in such real-life situations. Numerical techniques, explore the required theory to get numerical solutions to the mathematical problem, especially when computer deals with such huge and complicated calculations. By assumption that the root is exist in image's gray level interval, an attempt to explorer the application of some numerical algorithms is tested here by using some numerical analysis algorithms. An estimation to the error bounds for each approximation process had been calculated. Bisection, Newton Raphson, Secant and False Position methods are some of these methods which have been used here upon some digital images. Among the various used approximation methods and according to subjective and quantitative evaluation results, one can be noted that the Bisection method is the best approximation technique. Besides, image's histogram plays the main role in sequencing such methods of approximations.

Introduction:

One of the primary objectives of numerical analysis is to decide which algorithms can be implemented to solve mathematical problems on computer [1]. These algorithms accurately approximate the solution of problems which the latter can't be solved exactly and after that a typical technique for estimating error bounds for the approximation process can be handled [2]. Many researches focused their attention toward using such methods to solve their problems. C. Thinzar and N. Aye made use of the digital image correlation that is based on the sub-pixel accuracy with the aid of Newton Raphson method to detect storm direction for managing and to prevent the possible land's attacked region [3]. D. Biolè, etc. used Secant method in addition to other techniques to measure apparent contact angles of non-axisymmetric drops [4]. Md. G. Moazzam, etc. used a robust method for solving transcendental equations. The presented method by them gave better results rather than the well-known False Position method and the Bisection method for solving such equations [5]. W. Chu, etc. demonstrated a new registration algorithm based on Newton Raphson iteration process to align images with rigid body transformation [6]. So, an attempt to detect error propagation which can be occurred by applying such numerical methods to digital image had been utilized here. These methods are:

Bisection Method:

It is one of the most basic problems of numerical approximation. It is also known as the root-finding problem or Binary-search method. It is a process of dividing a set continually in half to search for the solution to a certain problem [2]. It states that if $f(x)$ is a continuous function defined on the interval $a < x < b$, with $f(a)$ and $f(b)$ of opposite sign, then there is at least one root in the

interval between a and b [5]. By setting $a_1=a$ and $b_1=b$ and take the midpoint of $[a,b]$ such that [7, 8].

$$p_1 = \frac{a_1 + b_1}{2}$$

So, the next steps must be taken in the considerations:

- If $f(p_1)=0$; a root was found at p_1 .
- If $f(a_1).f(b_1)<0$, then there is a root between p_1 and b_1 . And so on.
- If $f(a_1).f(b_1)>0$, then there is a root between a_1 and p_1 .

Pseudocode for Bisection Method

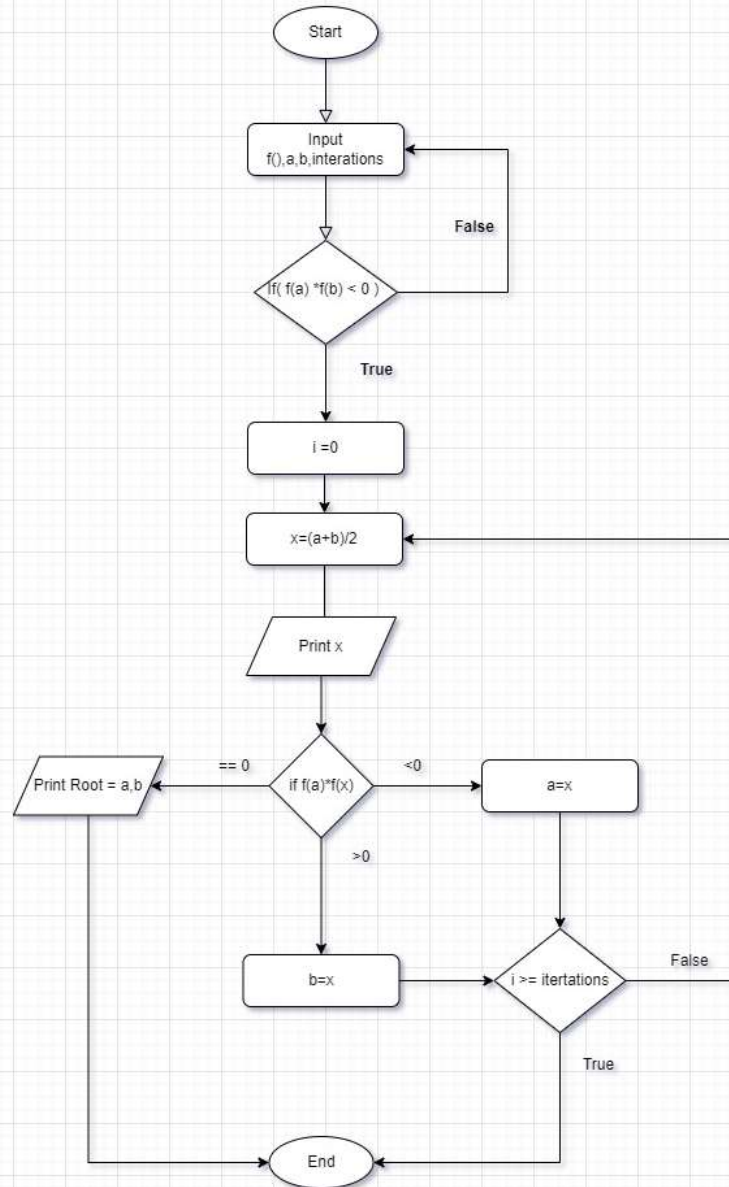
```

1. Start
2. Define function f(x)
3. Input
    a. Lower and Upper guesses x0 and x1
    b. tolerable error e
4. If f(x0)*f(x1) > 0
    print "Incorrect initial guesses"
    goto 3
End If
5. Do
    x2 = (x0+x1)/2
    If f(x0)*f(x2) < 0
        x1 = x2
    Else
        x0 = x2
    End If
    while abs(f(x2)) > e
6. Print root as x2
7. Stop
  
```

Flowchart for Bisection Method

Bisection

UMER AHMED (BCS203182)



Code for Bisection Method

```

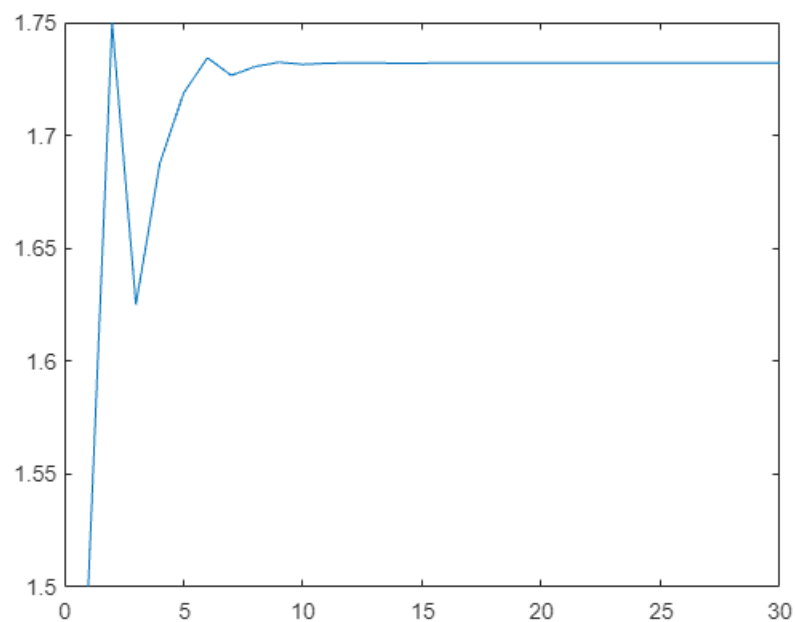
% Bisection Method      BCS203182
% Equation Input = @(x) x^2 -3 a=1 b=2
x = [];
functions= [];
iterations= [];

It = input('Enter Number of Iterations: ');
Eq = input('Enter the Equation: ');
A = input('Enter A: ');
B = input('Enter B: ');
fprintf('\n');
for i=1:It
    fprintf('Iteration No. %g\n', i);
    X = (A+B)/2;
    fprintf('X%g = %f\n\n', i, X);
    x = [x, X];
    functions = [functions, Eq(X)];
    iterations = [iterations, i];

    if Eq(X) < 0
        A = X;
    elseif Eq(X) > 0
        B = X;
    else
        break
    end
end
plot(iterations,x)
pause;

```

Iteration No. 30
X30 = 1.732051



Newton-Raphson Method:

It is one of the well-known numerical approaches for solving a root-finding problem [2]. This technique supposes that p_0 be an approximation to the root (i.e. iteration guess) and belongs to the interval $[a, b]$ in such that $f'(p_0) \neq 0$ and $|p - p_0|$ is so small and hence [7, 9].

$$p_n = p_{n-1} - \frac{f(p_{n-1})}{f'(p_{n-1})}, \text{ for } n \geq 1$$

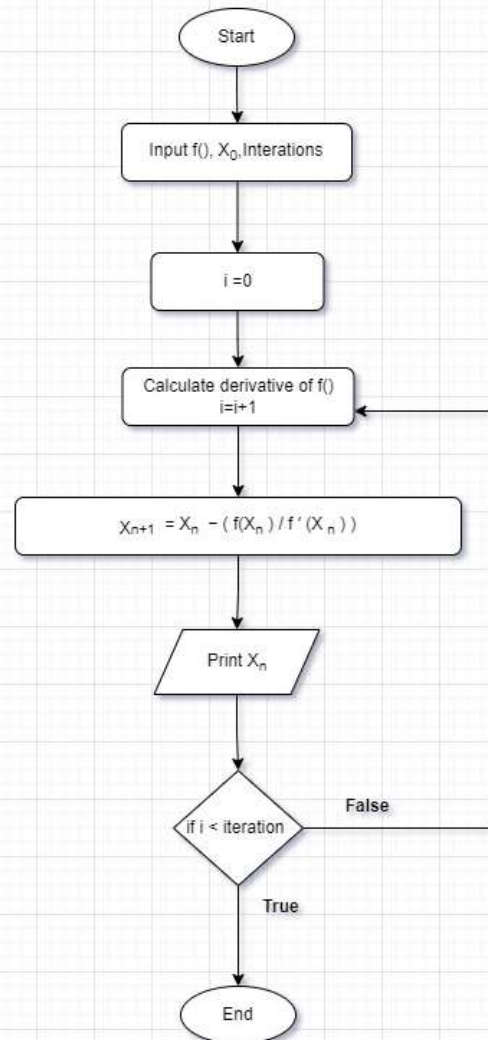
Pseudocode for Newton-Raphson Method

1. Start
2. Define function as $f(x)$
3. Define first derivative of $f(x)$ as $g(x)$
4. Input initial guess (x_0), tolerable error (e) and maximum iteration (N)
5. Initialize iteration counter $i = 1$
6. If $g(x_0) = 0$ then print "Mathematical Error" and goto (12) otherwise goto (7)
7. Calculate $x_1 = x_0 - f(x_0) / g(x_0)$
8. Increment iteration counter $i = i + 1$
9. If $i \geq N$ then print "Not Convergent" and goto (12) otherwise goto (10)
10. If $|f(x_1)| > e$ then set $x_0 = x_1$ and goto (6) otherwise goto (11)
11. Print root as x_1
12. Stop

Flowchart for Newton-Raphson Method

Newton Raphson Method

UMER AHMED (BCS203182)



Code for Newton-Raphson Method

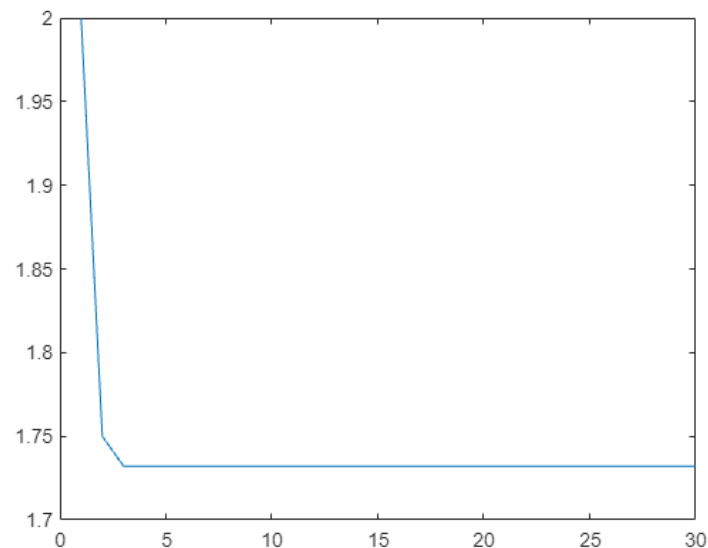
```
% Newton Raphson Method      BCS203182
Equation = @(x) x^2 -3 , X0=1
clc
clear
syms x
X = [];
I = [];
Iterations = input('Enter Iterations: ');
```

```

Equation = input('Enter Equation: ');
Derivative = eval(['@(x)' char(diff(Equation(x)))]);
pre = input('Enter Xo: ');
fprintf('\n');
for i=1:Iterations
    fprintf('Iteration No. %g\n', i);
    X = pre - (Equation(pre)/Derivative(pre));
    fprintf('X%g = %f\n\n', i, X);
    pre = X;
    X=[X,X];
    I=[I,i];
end
plot(I, X);

```

Iteration No. 30
X30 = 1.732051



False-Position Method (Regula-Falsi)

In this technique, one uses results that are known to be false to converge to the true root. This method chooses an initial approximations p_0 and p_1 such that $f(p_0) \cdot f(p_1) < 0$. The new approximation value will be then obtained in the way as described in the secant method mentioned above. After that and in order to decide which secant line to be used, the product of $f(p_2)$ and $f(p_1)$ should be taken and one of the following considerations shall be verified [7]:

- Choose p_3 as a line joining $(p_1, f(p_1))$ and $(p_2, f(p_2))$ if $f(p_1) \cdot f(p_2) < 0$
- Choose p_3 as a line joining $(p_0, f(p_0))$ and $(p_2, f(p_2))$ if $f(p_1) \cdot f(p_2) > 0$

Pseudocode for False-Position Method (Regula-Falsi)

```

1. Start

2. Define function f(x)

3. Input
    a. Lower and Upper guesses x0 and x1
    b. tolerable error e

4. If f(x0)*f(x1) > 0
    print "Incorrect initial guesses"
    goto 3
End If

5. Do

    
$$x2 = x0 - ((x0 - x1) * f(x0)) / (f(x0) - f(x1))$$


    If f(x0)*f(x2) < 0
        x1 = x2
    Else
        x0 = x2
    End If

    While abs(f(x2)) > e

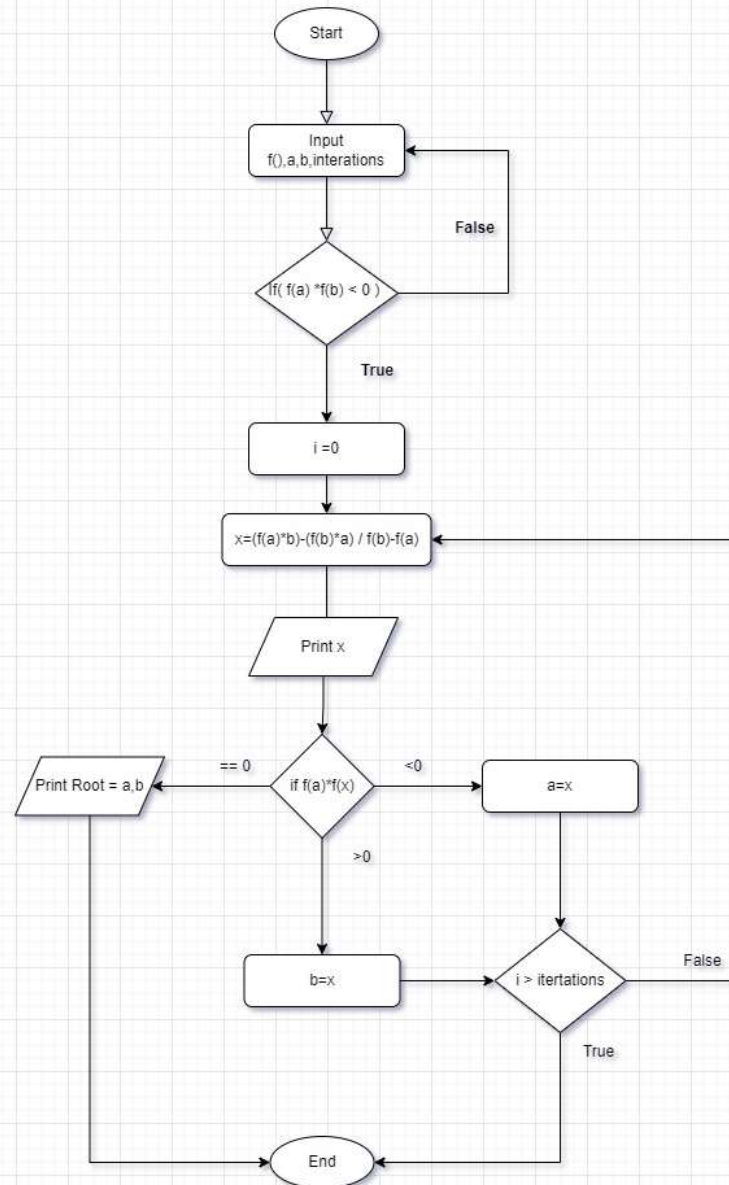
6. Print root as x2

7. Stop
  
```

Flowchart for False-Position Method (Regula-Falsi)

Regula-Falsi

UMER AHMED (BC S203182)



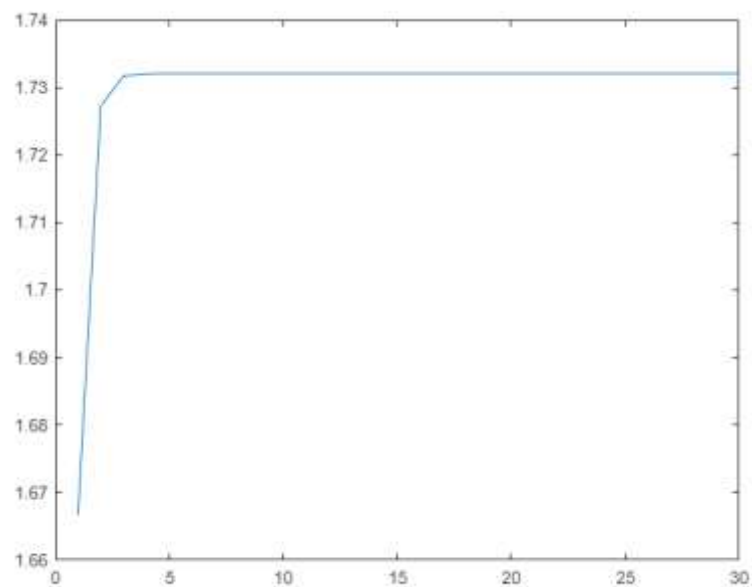
Code for False-Position Method (Regula-Falsi)

```

Equation = @(x) x^2 -3 , A=1 , B=2

clc
clear
X = [];
I = [];
It = input('Enter Iterations: ');
Eq = input('Enter Equation: ');
A = input('Enter A: ');
B = input('Enter B: ');
fprintf('\n');
for i=1:It
    fprintf('Iteration No. %g\n', i);
    Y = (A*Eq(B) - B*Eq(A)) / (Eq(B) - Eq(A));
    fprintf('X%g = %f\n\n', i, Y);
    X = [X, Y];
    I = [I, i];
    if Eq(Y) < 0
        A = Y;
    elseif Eq(Y) > 0
        B = Y;
    else
        break
    end
end
plot(I,X);

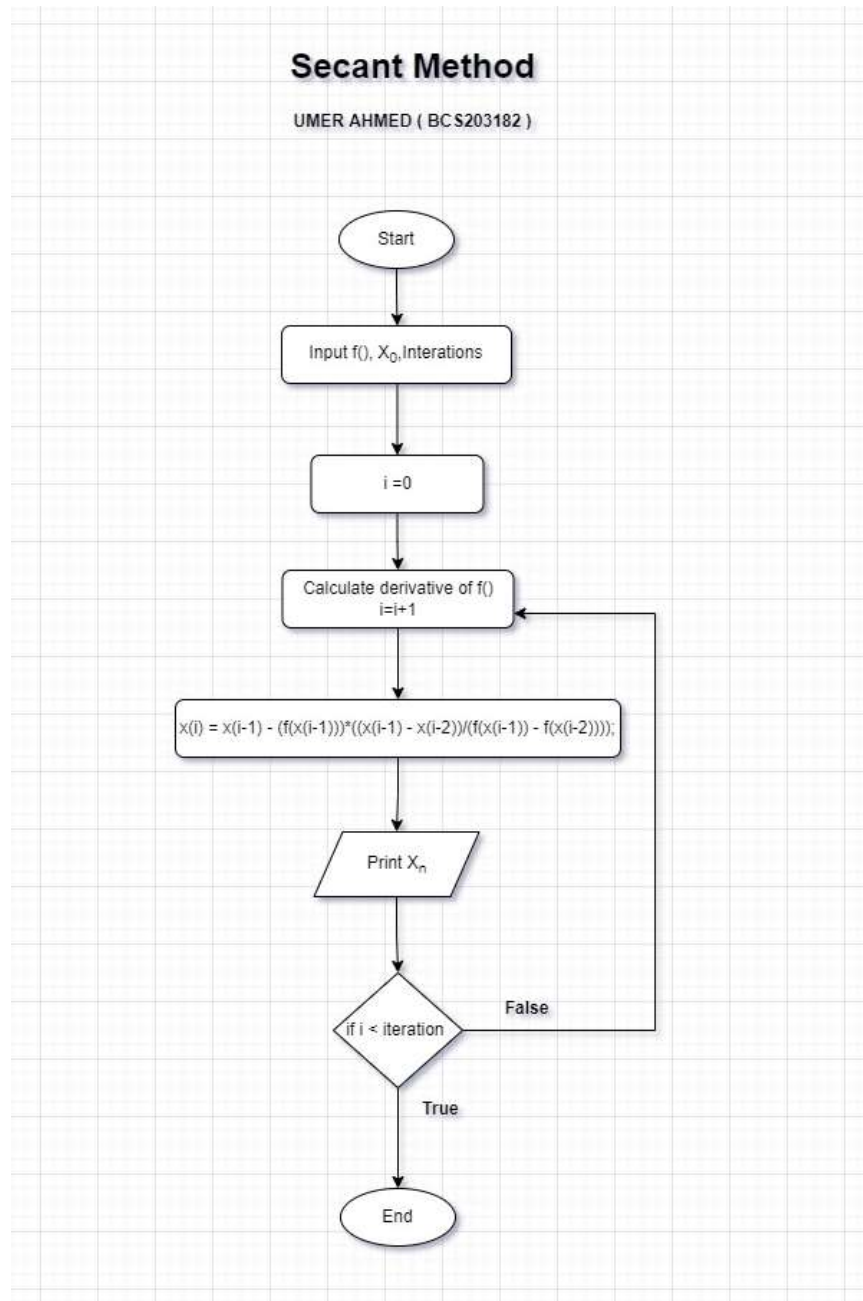
```



Secant Method:

This method uses a secant line joining two points that cut curve's function and can be presented in the following expression [2, 8]:

Flowchart for Secant Method



Pseudocode for Secant Method

1. Start

2. Define function as $f(x)$

3. Input:

- a. Initial guess x_0, x_1
- b. Tolerable Error e
- c. Maximum Iteration N

4. Initialize iteration counter $step = 1$

5. Do

```

If  $f(x_0) = f(x_1)$ 
    Print "Mathematical Error"
    Stop
End If

 $x_2 = x_1 - (x_1 - x_0) * f(x_1) / (f(x_1) - f(x_0))$ 
 $x_0 = x_1$ 
 $x_1 = x_2$ 
 $step = step + 1$ 
If  $step > N$ 
    Print "Not Convergent"
    Stop
End If
  
```

While $abs f(x_2) > e$

6. Print root as x_2

7. Stop

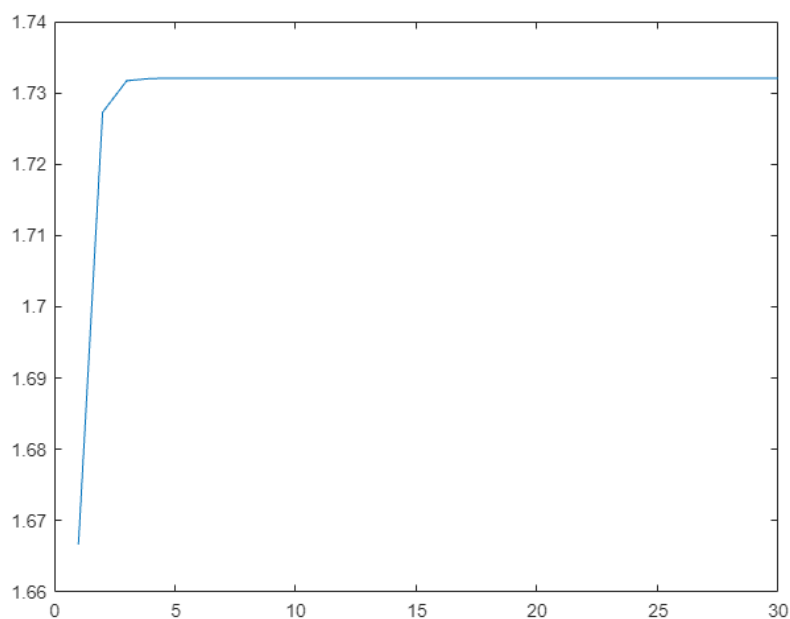
Code for Secant Method

```

% Secant Method      BCS203182
% Equation = @(x) x^2 -3 , A=1 , B=2

clc
clear
X = [];
I = [];
Iter = input('Enter Iterations: ');
Eq = input('Enter Equation: ');
A = input('Enter A: ');
B = input('Enter B: ');
fprintf('\n');
for i=1:Iter
    fprintf('Iteration No. %g\n', i);
    Y = (A*Eq(B) - B*Eq(A)) / (Eq(B) - Eq(A));
    fprintf('X%g = %f\n\n', i, Y);
    X = [X, Y];
    I = [I, i];
    if Eq(Y) < 0
        A = Y;
    elseif Eq(Y) > 0
        B = Y;
    else
        break
    end
end
plot(I,X);

```



MATLAB Code for grayscale level

```
clc ;
clear all;
close all;
a = imread('https://static.remove.bg/remove-bg-
web/eb1bb48845c5007c3ec8d72ce7972fc8b76733b1/assets/start-
1abfb4fe2980eabfbbaaa4365a0692539f7cd2725f324f904565a9a744f8e214.jpg') ;
b = rgb2gray ( a ) ;

ax1 = subplot ( 2,2,1 ) ;
imshow ( b ) ; title ( 'Image Input Grayscale');
ax2 = subplot ( 2,2,2 ) ;
imhist ( b ) ;
title ( ' Histogram Input Grayscale');
```

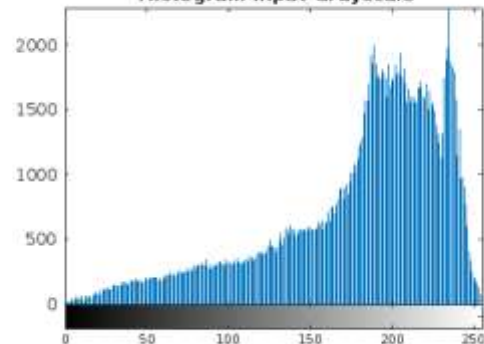
Original Image



Image Input Grayscale

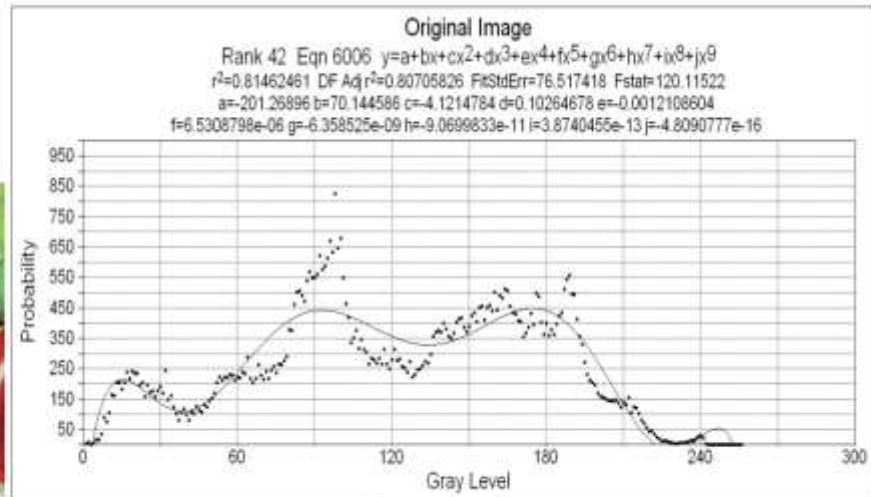


Histogram Input Grayscale





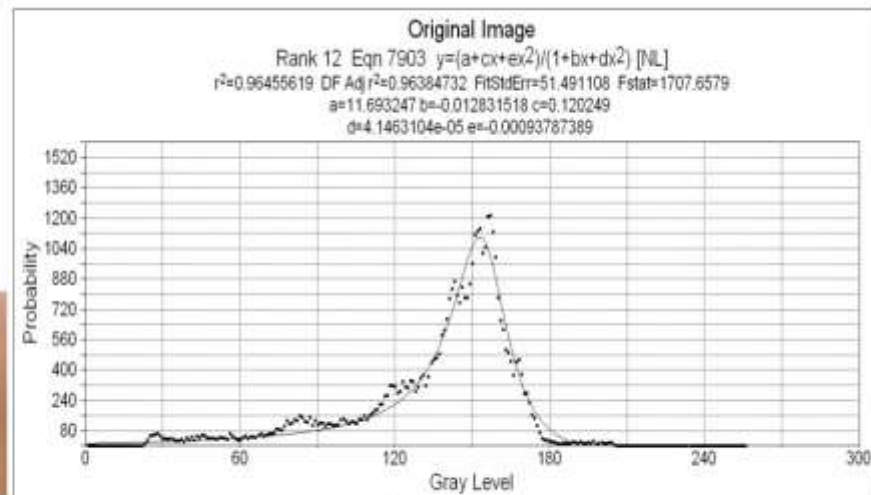
(a)



(b)



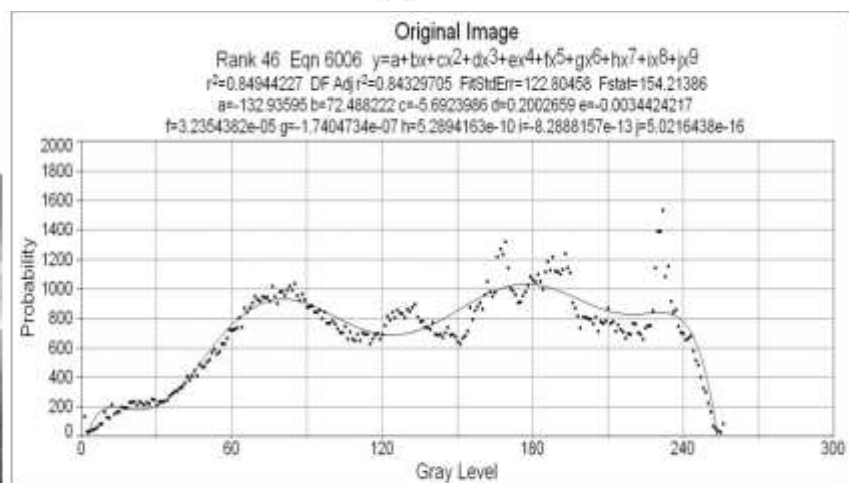
(c)



(d)



(e)



(f)

Result and Discussion:

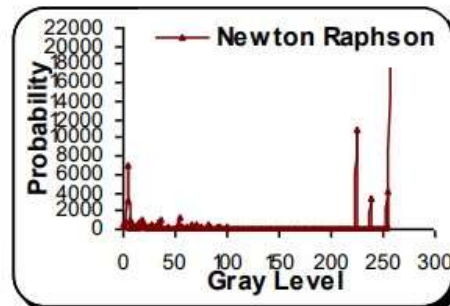
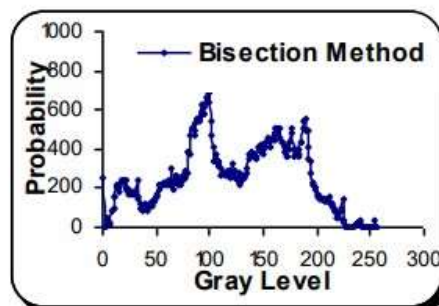
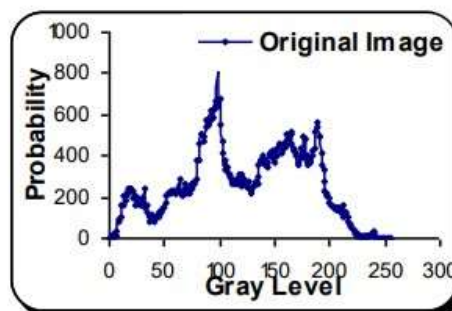
For simplicity, an assumption had been made here that the root in the interval (i.e. image's data) is unique and the stopping criteria shall be as follows:

$$\text{Newapproximationvalue} = |p_n|, \text{ if } |f(p_n)| \leq \varepsilon \text{ or } \text{iteration} > 5$$

where ε is the tolerance. Figure (1) shows the used images with their fitting histograms' curves. The fitting process had been executed by using Table Curve 2D V 5.01 program. In general, and as a result of application, numerical approximation methods distorted the images in alternatives ratios depending on image's histogram. An image with sharp chromatic gradient is better than other used images which have a wide range of color gradients, this can be seen through noticing Figs. (2-4). In addition to that, one can noticed that there is no clear sequence for algorithms' effects upon single image. In most cases, one can find that the best numerical methods' performance is the Bisection method with lowest errors presented in Table (1).

Table 1. Approximation Errors for all used images

Approximation Method	Vegetable Image		Eye Image		Balloon Image	
	Absolute Error	Relative Error	Absolute Error	Relative Error	Absolute Error	Relative Error
Bisection	748.023	9.948	250.236	1.846	791.473	12.133
Newton Raphson	9581.512	595.799	8011.957	141.260	9613.176	965.037
Secant	5633.692	279.477	5268.604	423.076	4726.446	565.178
False Position	4465.159	160.174	4117.540	173.656	3457.43	290.102



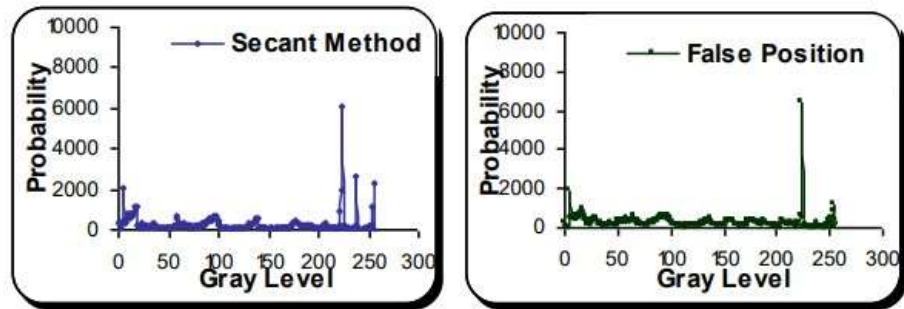


Figure 2. Histograms for vegetable image before and after numerical methods' applications

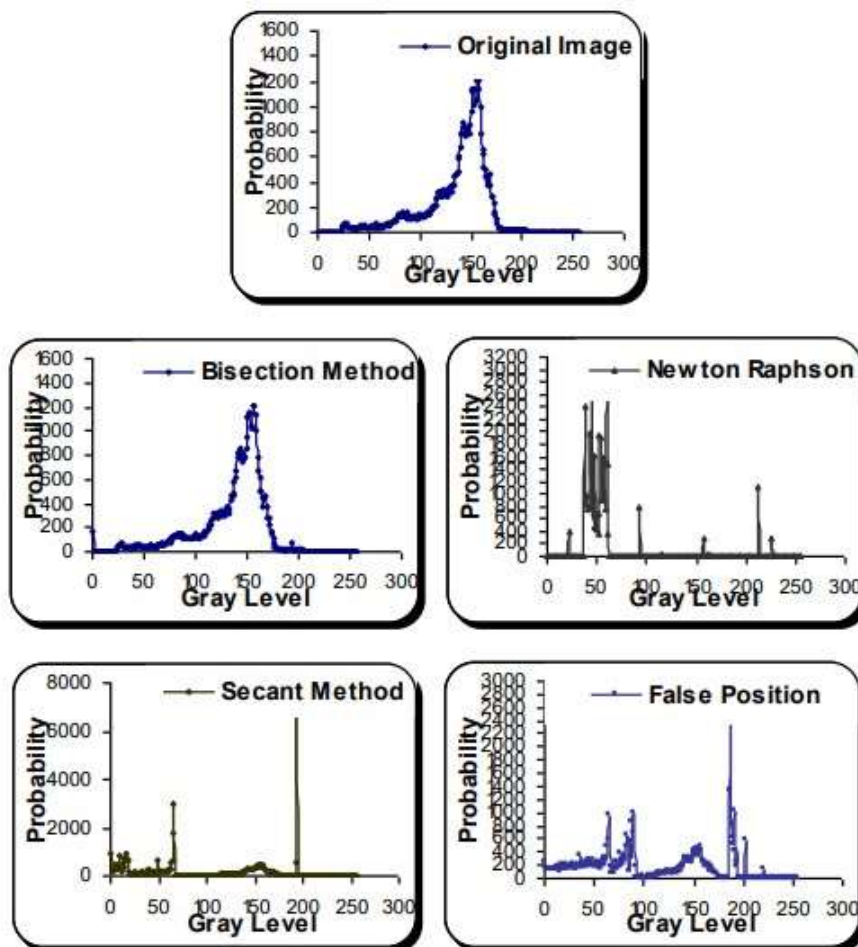


Figure 3. Histograms for eye image before and after numerical methods' applications

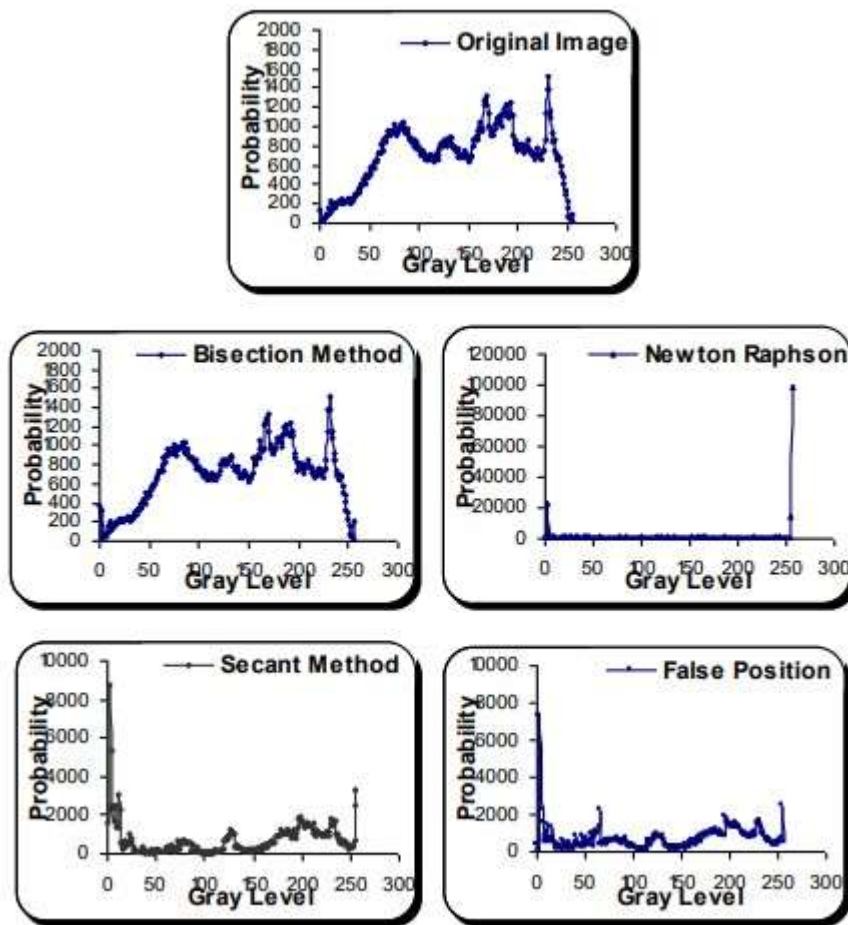


Figure 4. Histograms for balloon image before and after numerical methods' applications

A distorted case occurred in eye image in the application of the Newton Raphson method with a shift toward lower values of the image's gray level while it isn't the case for other used images; the distortion case resulted near the ends of images' gray levels for all used algorithms in general, this can be recognized through noticing approximation errors explained in the above table. Image's histogram restricts the sequence of algorithms in their applications; the symmetric behavior for both Secant and False Position methods resulted in the case of vegetable and balloon images in contraction to that occurred to the case in eye image which the latter witnessed a strong distortion state in the case of Secant method technique.

Conclusion

By assumption that the root is exist in image's gray level interval, an attempt to explorer the application of some numerical algorithms is tested here by using some numerical analysis algorithms. As a consequence, result, one can noticed that the used numerical algorithms distorted the image after all with different distortion ratios depend upon image's histogram. In addition to that, the sequence for techniques' effects is not clear upon such used images but in general one can be concluded that the Bisection method can be recognized as the best one with least absolute and relative errors.

References

- [1] Z. Xu. (2010) Numerical Analysis, ACMS 40390. [Online]. Available: [http://www.scribd.com/document/197468938/ac ms-40390f10-syllabus](http://www.scribd.com/document/197468938/ac-ms-40390f10-syllabus).
- [2] R. L. Burden and J. D. Faires, Numerical Analysis, Ninth Edition, Brooks/ Cole, Cengage Learning, 2011.
- [3] C. Thinzar, and N. Aye, "Detection the storm movement by sub pixel registration approach of Newton Raphson method" International Journal of e-Education, eBusiness, e-Management and e-Learning, Vol. 4, No. 1, 2014.
- [4] D. Biolè, M. Wang, and V. Bertola, "Assessment of direct image processing methods to measure the apparent contact angle of liquid drops" Experimental Thermal and Fluid Science, Vol. 76, 2016. Available: [https://doi.org/10.1016/j. expthermflusci.2016.04.006](https://doi.org/10.1016/j.expthermflusci.2016.04.006).
- [5] Md. G. Moazzam, A. Chakraborty, and Md. A. Bhuiyan, "A roubust method for solving transcendental equations" IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 6, No. 2, 2012.
- [6] W. Chu, L. Ma, J. Song, and T. Vorburger, "An iterative image registration algorithm by optimizing similarity measurement" Journal of Research of the National Institute of Standards and Technology, Vol. 115, No. 1, 2010.
- [7] J. Kiusalaas, Numerical Methods in Engineering with Matlab, Second Edition, Cambridge University Press, 2010.
- [8] S. C. Chapra, Applied Numerical Methods with Matlab ® for Engineers and Scientists, Third Edition, McGraw-Hill Companies, Inc., 2012.
- [9] S. T. Karris, Analysis Using Matlab ® and Excel ® , Third Edition, Orchard Publications, 2007