

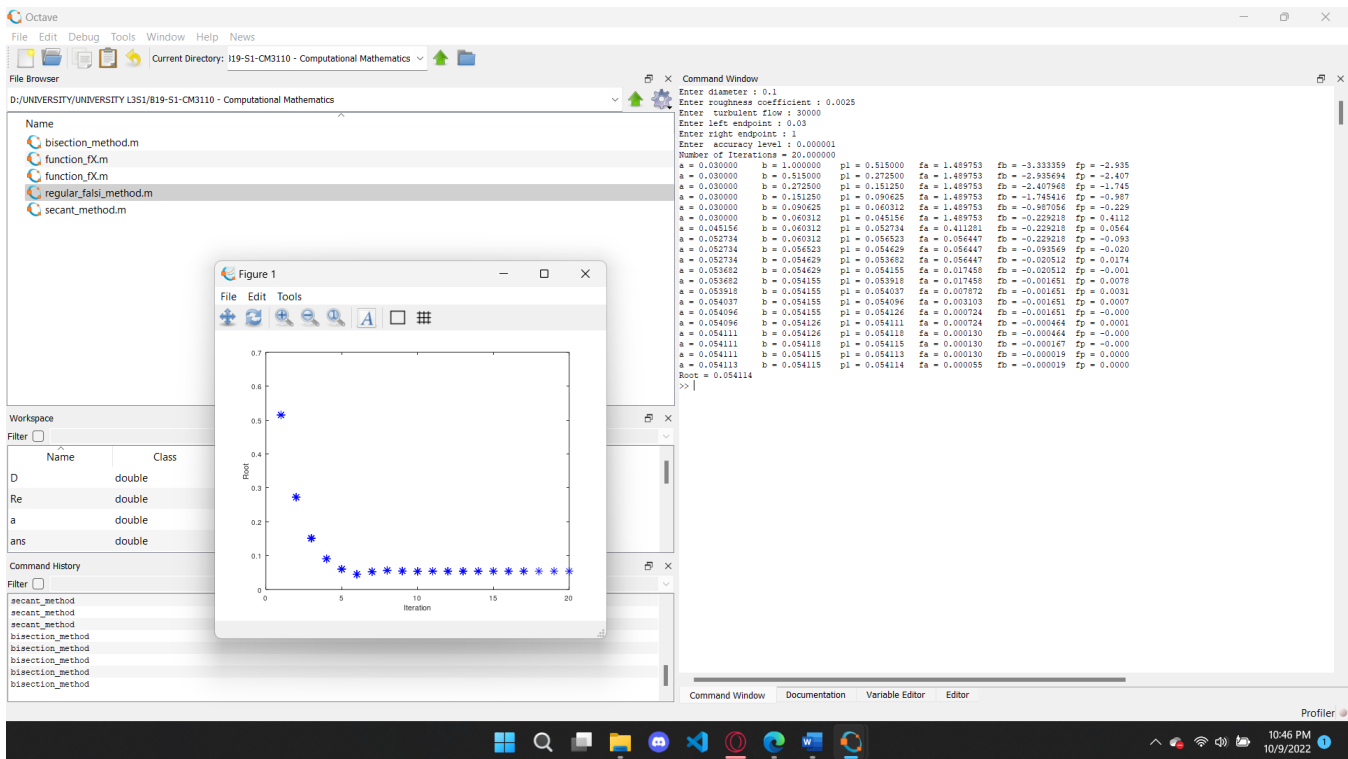
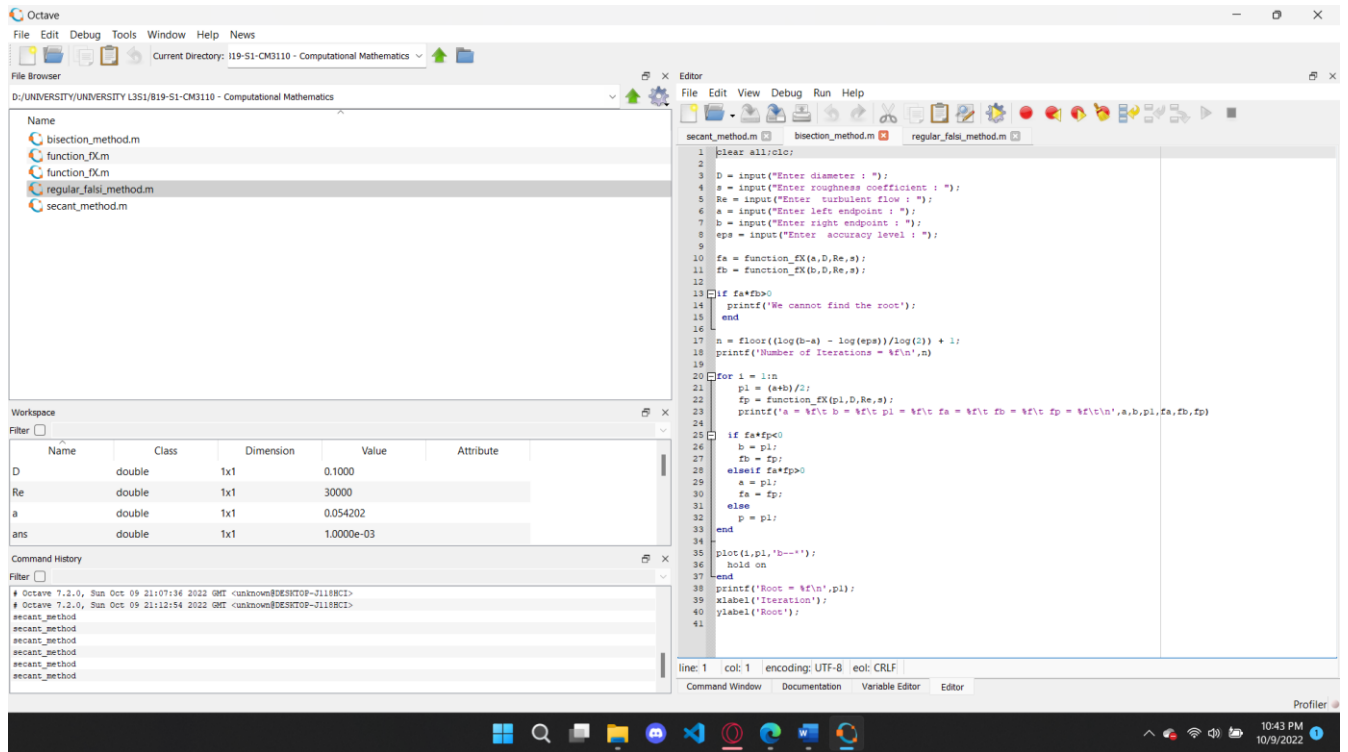
CM 3110 – Computational Mathematics

Assignment 1
Introduction to numerical Methods

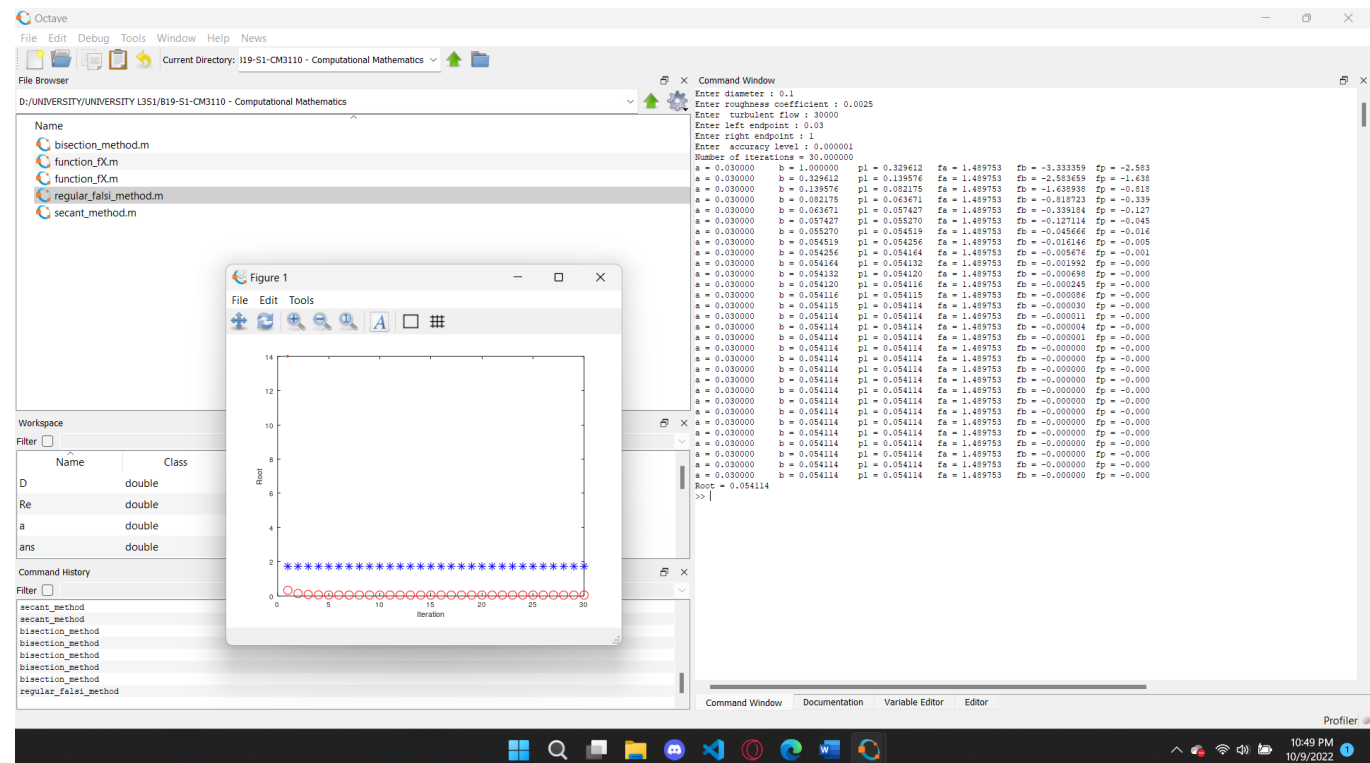
Student name : Herath H.M.J.T

Student reference : 194055L

- Screenshots of program running
 - a) Bisection Method



The image shows the Octave IDE interface. At the top, the menu bar includes File, Edit, Debug, Tools, Window, Help, and News. Below the menu is a toolbar with icons for file operations and execution. The main window is divided into three panes. The left pane, titled 'Workspace', shows a list of variables: D (double, 1x1, value 0.1000), Re (double, 1x1, value 30000), a (double, 1x1, value 0.054114), and ans (double, 1x1, value 1.0000e-06). Below this is the 'Command History' pane, which lists several calls to the 'bisection_method' function. The right pane, titled 'Editor', shows the source code of the 'regular_falsi_method.m' file. The code is a MATLAB script that implements the regularized false position method. It takes inputs for diameter (D), roughness coefficient (a), turbulent flow (Re), left endpoint (a), right endpoint (b), and accuracy level (eps). It calculates the function values at the endpoints and iteratively refines the root estimate using the false position method, with a safeguard against oscillations. The script includes a plot of the function and the root estimate. The status bar at the bottom indicates the current line is 43, column 1, with UTF-8 encoding and CRLF line endings. The system tray at the very bottom shows the date and time as 10:47 PM on 10/9/2022.



c) Secant method

The screenshot shows the Octave IDE with the `secant_method.m` script open in the Editor. The script implements the secant method for finding roots of a function `f(x)` over a given interval `[a, b]` with a specified accuracy level `eps`. The script uses a loop to iteratively refine the root estimate until the desired accuracy is achieved.

The Workspace panel displays the following variables:

Name	Class	Dimension	Value	Attribute
D	double	1x1	0.1000	
Re	double	1x1	30000	
a	double	1x1	0.030000	
ans	double	1x1	1.0000e-06	

The Command History panel shows the execution of the `secant_method` function.

The screenshot shows the Octave IDE with the `secant_method.m` script open in the Editor. The script implements the secant method for finding roots of a function `f(x)` over a given interval `[a, b]` with a specified accuracy level `eps`. The script uses a loop to iteratively refine the root estimate until the desired accuracy is achieved.

The Workspace panel displays the following variables:

Name	Class	Dimension	Value	Attribute
D	double	1x1	0.1000	
Re	double	1x1	30000	
a	double	1x1	0.030000	
ans	double	1x1	1.0000e-06	

The Command History panel shows the execution of the `secant_method` function.

A Figure window titled "Figure 1" displays a plot of the function `f(x)` over the interval `[a, b]`. The x-axis ranges from 0 to 6, and the y-axis ranges from 0 to 6. The plot shows the function `f(x)` and its root, which is approximately 0.03.

- **Program file**

- a) Bisection Method**

```
clear all;clc;
```

```
D = input("Enter diameter : ");
```

```
s = input("Enter roughness coefficient : ");
```

```
Re = input("Enter turbulent flow : ");
```

```
a = input("Enter left endpoint : ");
```

```
b = input("Enter right endpoint : ");
```

```
eps = input("Enter accuracy level : ");
```

```
fa = function_fX(a,D,Re,s);
```

```
fb = function_fX(b,D,Re,s);
```

```
if fa*fb>0
```

```
    printf('We cannot find the root');
```

```
end
```

```
n = floor((log(b-a) - log(eps))/log(2)) + 1;
```

```
printf('Number of Iterations = %f\n',n)
```

```
for i = 1:n
```

```
    p1 = (a+b)/2;
```

```
    fp = function_fX(p1,D,Re,s);
```

```
    printf('a = %f\t b = %f\t p1 = %f\t fa = %f\t fb = %f\t fp = %f\t\n',a,b,p1,fa,fb,fp)
```

```
    if fa*fp<0
```

```
        b = p1;
```

```
        fb = fp;
```

```
    elseif fa*fp>0
```

```

    a = p1;

    fa = fp;

else

    p = p1;

end

plot(i,p1,'b--*');

    hold on

end

printf('Root = %f\n',p1);

xlabel('Iteration');

ylabel('Root');

```

b) Regular Falsi method

```

clear all;clc;

D = input("Enter diameter : ");

s = input("Enter roughness coefficient : ");

Re = input("Enter turbulent flow : ");

a = input("Enter left endpoint : ");

b = input("Enter right endpoint : ");

eps = input("Enter accuracy level : ");

fa = function_fX(a,D,Re,s);

fb = function_fX(b,D,Re,s);

if fa*fb>0

    printf('We cannot find the root');

end

```

```

n = 30; % Number of iterations

fprintf('Number of iterations = %f\n',n);

for i=1:n

    p1 = (a*fb - b*fa)/(fb - fa);

    fp = function_fX(p1,D,Re,s);

    printf('a = %f\t b = %f\t p1 = %f\t fa = %f\t fb = %f\t fp = %f\t\n',a,b,p1,fa,fb,fp)

    plot(i, sqrt(3),'b--*',i, p1,'ro--', 14);

    hold on

    if(abs(a-p1)<eps)

        break

    end

    if fa*fp<0

        b=p1;

        fb=fp;

    else if fa*fp>0

        a=p1;

        fa=fp;

    else

        p=p1;

    end

end

end

end

printf('Root = %f\n',p1);

xlabel('Iteration');

ylabel('Root');

```

c) Secant method

```
clear all;clc;
```

```

D = input("Enter diameter : ");

s = input("Enter roughness coefficient : ");

Re = input("Enter turbulent flow : ");

a = input("Enter left endpoint : ");

b = input("Enter right endpoint : ");

eps = input("Enter accuracy level : ");


f0 = function_fX(a,D,Re,s);

f1 = function_fX(b,D,Re,s);


n = 30; % Number of iterations

for i=1:n

    x = (a*f1 - b*f0)/(f1 - f0);

    fx = function_fX(x,D,Re,s);

    printf('a = %f\t b = %f\t x = %f\t f0 = %f\t f1 = %f\t fx = %f\t\n',a,b,x,f0,f1,fx);

    a = b;

    b = x;

    f0 = f1;

    f1 = fx;

    plot(i, 'ro--',i, x, 'b--*');

    hold on

    if(abs(b - a)<eps)

        root = b;

        fprintf('Root=%f\n',root);

        fprintf('Number of iterations=%d\n',i);

        break

    end

end

```


- **Output file**

a) Bisection Method

```

Enter diameter : 0.1
Enter roughness coefficient : 0.0025
Enter turbulent flow : 30000
Enter left endpoint : 0.03
Enter right endpoint : 1
Enter accuracy level : 0.000001
Number of Iterations = 20.000000
a = 0.030000    b = 1.000000    p1 = 0.515000    fa = 1.489753    fb = -3.333359    fp = -2.935694
a = 0.030000    b = 0.515000    p1 = 0.272500    fa = 1.489753    fb = -2.935694    fp = -2.407968
a = 0.030000    b = 0.272500    p1 = 0.151250    fa = 1.489753    fb = -2.407968    fp = -1.745416
a = 0.030000    b = 0.151250    p1 = 0.090625    fa = 1.489753    fb = -1.745416    fp = -0.987056
a = 0.030000    b = 0.090625    p1 = 0.060312    fa = 1.489753    fb = -0.987056    fp = -0.229218
a = 0.030000    b = 0.060312    p1 = 0.045156    fa = 1.489753    fb = -0.229218    fp = 0.411281
a = 0.045156    b = 0.060312    p1 = 0.052734    fa = 0.411281    fb = -0.229218    fp = 0.056447
a = 0.052734    b = 0.060312    p1 = 0.056523    fa = 0.056447    fb = -0.229218    fp = -0.093569
a = 0.052734    b = 0.056523    p1 = 0.054629    fa = 0.056447    fb = -0.093569    fp = -0.020512
a = 0.052734    b = 0.054629    p1 = 0.053682    fa = 0.056447    fb = -0.020512    fp = 0.017458
a = 0.053682    b = 0.054629    p1 = 0.054155    fa = 0.017458    fb = -0.020512    fp = -0.001651
a = 0.053682    b = 0.054155    p1 = 0.053918    fa = 0.017458    fb = -0.001651    fp = 0.007872
a = 0.053918    b = 0.054155    p1 = 0.054037    fa = 0.007872    fb = -0.001651    fp = 0.003103
a = 0.054037    b = 0.054155    p1 = 0.054096    fa = 0.003103    fb = -0.001651    fp = 0.000724
a = 0.054096    b = 0.054155    p1 = 0.054126    fa = 0.000724    fb = -0.001651    fp = -0.000464
a = 0.054096    b = 0.054126    p1 = 0.054111    fa = 0.000724    fb = -0.000464    fp = 0.000130
a = 0.054111    b = 0.054126    p1 = 0.054118    fa = 0.000130    fb = -0.000464    fp = -0.000167
a = 0.054111    b = 0.054118    p1 = 0.054115    fa = 0.000130    fb = -0.000167    fp = -0.000019
a = 0.054111    b = 0.054115    p1 = 0.054113    fa = 0.000130    fb = -0.000019    fp = 0.000055
a = 0.054113    b = 0.054115    p1 = 0.054114    fa = 0.000055    fb = -0.000019    fp = 0.000018
Root = 0.054114
>>

```

b) Regular Falsi method

```

Enter diameter : 0.1
Enter roughness coefficient : 0.0025
Enter turbulent flow : 30000
Enter left endpoint : 0.03
Enter right endpoint : 1
Enter accuracy level : 0.000001
Number of iterations = 30.000000
a = 0.030000    b = 1.000000    p1 = 0.329612    fa = 1.489753    fb = -3.333359    fp = -2.583659
a = 0.030000    b = 0.329612    p1 = 0.139576    fa = 1.489753    fb = -2.583659    fp = -1.638938
a = 0.030000    b = 0.139576    p1 = 0.082175    fa = 1.489753    fb = -1.638938    fp = -0.818723
a = 0.030000    b = 0.082175    p1 = 0.063671    fa = 1.489753    fb = -0.818723    fp = -0.339184
a = 0.030000    b = 0.063671    p1 = 0.057427    fa = 1.489753    fb = -0.339184    fp = -0.127114
a = 0.030000    b = 0.057427    p1 = 0.055270    fa = 1.489753    fb = -0.127114    fp = -0.045666
a = 0.030000    b = 0.055270    p1 = 0.054519    fa = 1.489753    fb = -0.045666    fp = -0.016146
a = 0.030000    b = 0.054519    p1 = 0.054256    fa = 1.489753    fb = -0.016146    fp = -0.005676
a = 0.030000    b = 0.054256    p1 = 0.054164    fa = 1.489753    fb = -0.005676    fp = -0.001992
a = 0.030000    b = 0.054164    p1 = 0.054132    fa = 1.489753    fb = -0.001992    fp = -0.000698
a = 0.030000    b = 0.054132    p1 = 0.054120    fa = 1.489753    fb = -0.000698    fp = -0.000245
a = 0.030000    b = 0.054120    p1 = 0.054116    fa = 1.489753    fb = -0.000245    fp = -0.000086
a = 0.030000    b = 0.054116    p1 = 0.054115    fa = 1.489753    fb = -0.000086    fp = -0.000030
a = 0.030000    b = 0.054115    p1 = 0.054114    fa = 1.489753    fb = -0.000030    fp = -0.000011
a = 0.030000    b = 0.054114    p1 = 0.054114    fa = 1.489753    fb = -0.000011    fp = -0.000004
a = 0.030000    b = 0.054114    p1 = 0.054114    fa = 1.489753    fb = -0.000004    fp = -0.000001

```

```

a = 0.030000    b = 0.054114    p1 = 0.054114    fa = 1.489753    fb = -0.000001    fp = -0.000000
a = 0.030000    b = 0.054114    p1 = 0.054114    fa = 1.489753    fb = -0.000000    fp = -0.000000
a = 0.030000    b = 0.054114    p1 = 0.054114    fa = 1.489753    fb = -0.000000    fp = -0.000000
a = 0.030000    b = 0.054114    p1 = 0.054114    fa = 1.489753    fb = -0.000000    fp = -0.000000
a = 0.030000    b = 0.054114    p1 = 0.054114    fa = 1.489753    fb = -0.000000    fp = -0.000000
a = 0.030000    b = 0.054114    p1 = 0.054114    fa = 1.489753    fb = -0.000000    fp = -0.000000
a = 0.030000    b = 0.054114    p1 = 0.054114    fa = 1.489753    fb = -0.000000    fp = -0.000000
a = 0.030000    b = 0.054114    p1 = 0.054114    fa = 1.489753    fb = -0.000000    fp = -0.000000
a = 0.030000    b = 0.054114    p1 = 0.054114    fa = 1.489753    fb = -0.000000    fp = -0.000000
a = 0.030000    b = 0.054114    p1 = 0.054114    fa = 1.489753    fb = -0.000000    fp = -0.000000
a = 0.030000    b = 0.054114    p1 = 0.054114    fa = 1.489753    fb = -0.000000    fp = -0.000000
a = 0.030000    b = 0.054114    p1 = 0.054114    fa = 1.489753    fb = -0.000000    fp = -0.000000
a = 0.030000    b = 0.054114    p1 = 0.054114    fa = 1.489753    fb = -0.000000    fp = -0.000000
a = 0.030000    b = 0.054114    p1 = 0.054114    fa = 1.489753    fb = -0.000000    fp = -0.000000
Root = 0.054114

```

>>

c) Secant method

Enter diameter : 0.1

Enter roughness coefficient : 0.0025

Enter turbulent flow : 30000

Enter left endpoint : 0.03

Enter right endpoint : 0.07

Enter accuracy level : 0.000001

```

a = 0.030000    b = 0.070000    x = 0.059584    f0 = 1.489753    f1 = -0.524483    fx = -0.204162
a = 0.070000    b = 0.059584    x = 0.052946    f0 = -0.524483    f1 = -0.204162    fx = 0.047647
a = 0.059584    b = 0.052946    x = 0.054202    f0 = -0.204162    f1 = 0.047647    fx = -0.003527
a = 0.052946    b = 0.054202    x = 0.054116    f0 = 0.047647    f1 = -0.003527    fx = -0.000057
a = 0.054202    b = 0.054116    x = 0.054114    f0 = -0.003527    f1 = -0.000057    fx = 0.000000
a = 0.054116    b = 0.054114    x = 0.054114    f0 = -0.000057    f1 = 0.000000    fx = -0.000000

```

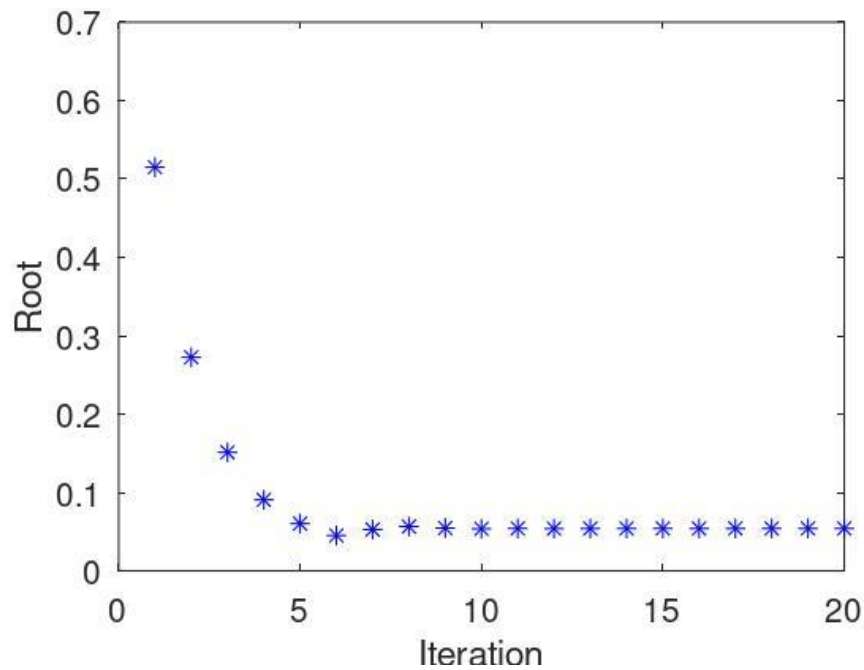
Root=0.054114

Number of iterations=6

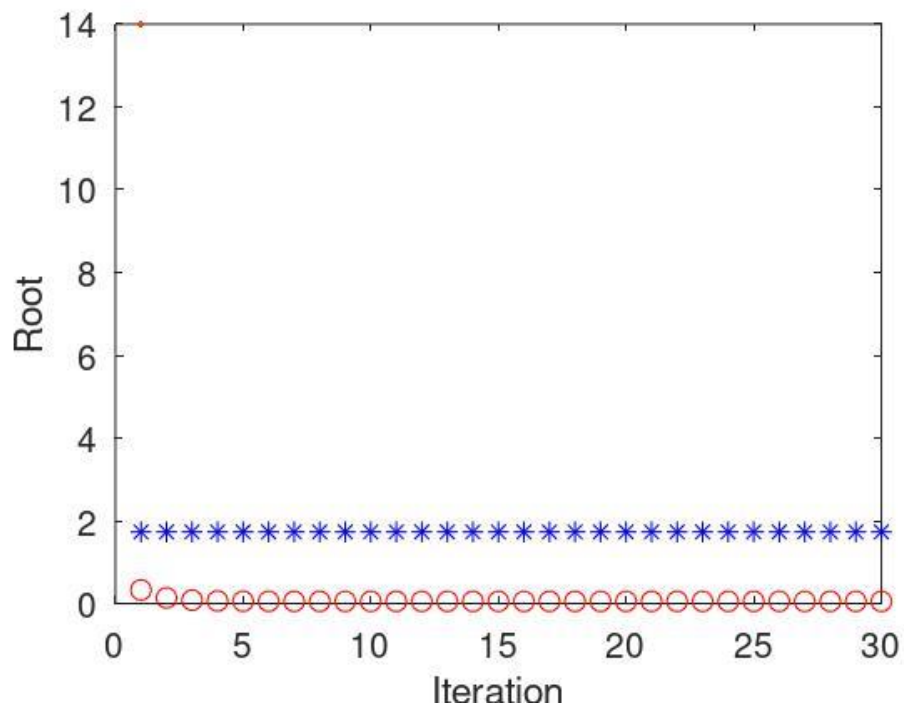
>>

- Plot of approximate solution vs. iteration in each techniques

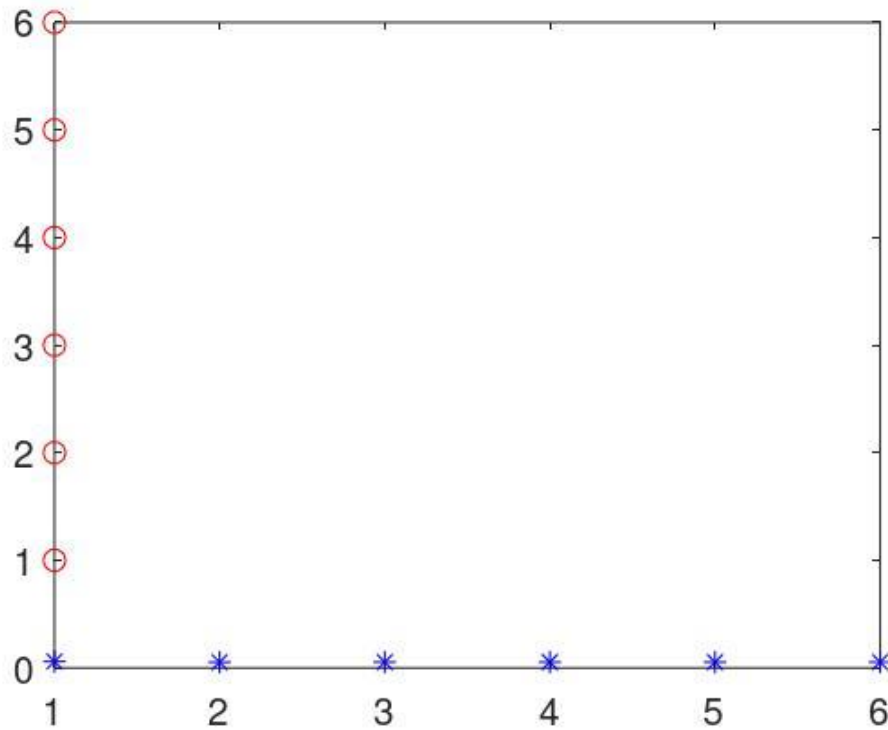
1. Bisection Method



2. Regular Falsi method



3. Secant method



- **Comparison and analysis of different techniques**

- The estimated solution of the plot grows increasingly accurate as the number of iterations increases.
- Comparison and analysis of different techniques
 1. The bisection Method is the simplest numerical scheme to find the roots. This method is very slow but the accuracy is high.
 2. The method of False position which is also known as the Regular Falsi Method is one of the bracketing methods for finding the roots of the equation. This was developed since the convergence of the Bisection method is very slow. Since this is a bracketing method, the root is located within an interval prescribed by a lower and an upper bound.
 3. The Secant method requires the selection of 2 initial approximates x_0 and x_1 which may or may not bracket the desired root, but which are chosen reasonably close to the exact root.
- When comparing the outputs of these 3 methods every method gives 0.054114.