ESO-208A
Computational Methods in Engineering
Assignment 1

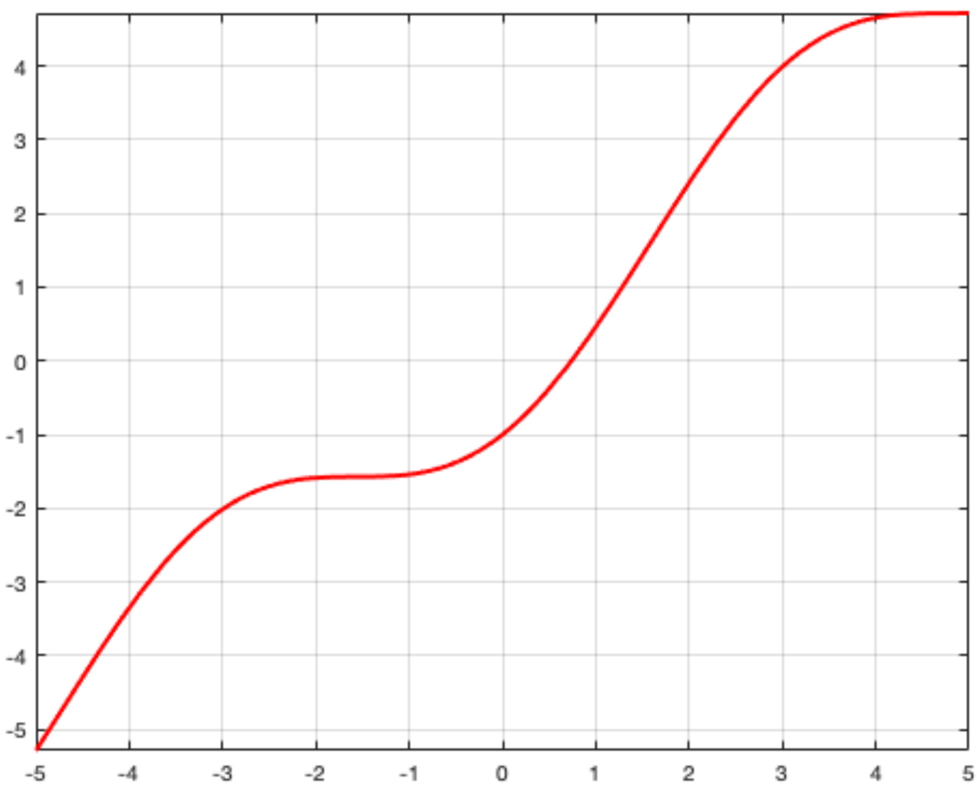
NAVNEET SINGH
200626

Q1.

Test functions:

(1) $f(x) = x - \cos x$

Output plot of this function

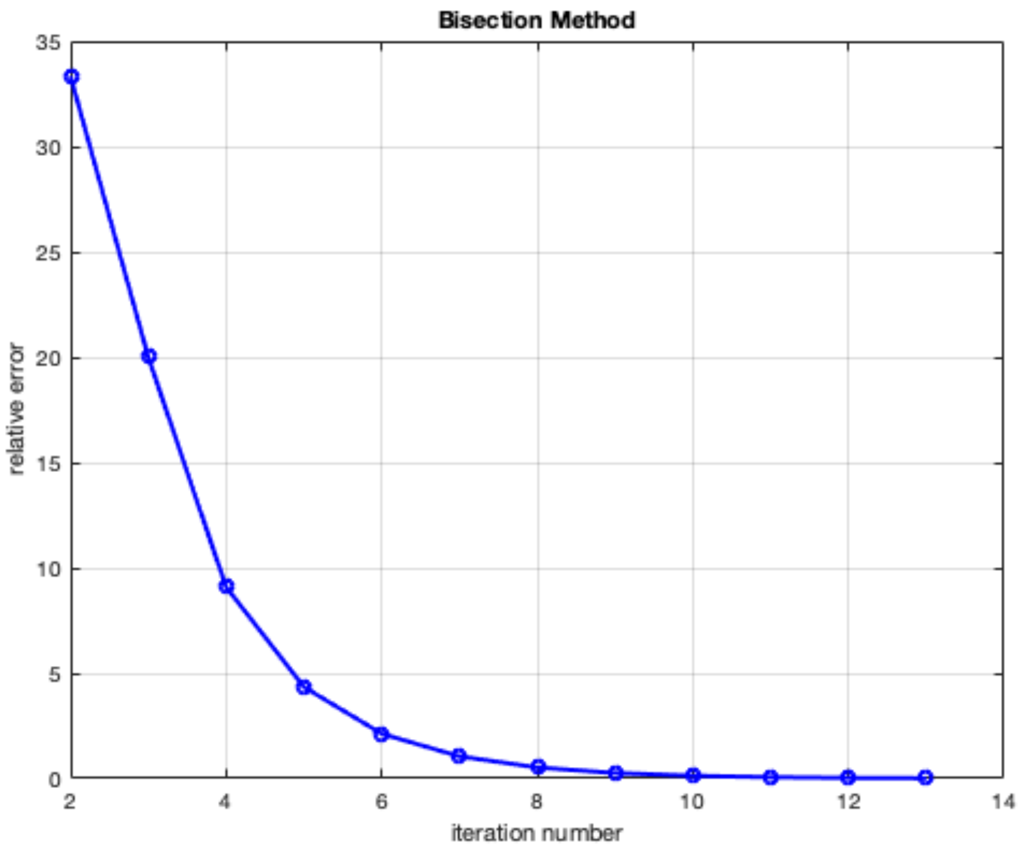


a) Bisection Method

```
>> eso208_200626_q1
List of methods:
1. Bisection Method
2. Regula Falsi
3. Fixed Point
4. Newton Raphson
5. Secant
Enter method number to use : 1
Enter your function : x-cos(x)
Enter the Value of x_low :  0
```

Enter the value of x_up : 1
Enter the max no. of iterations : 50
Enter the max relative error in(%) : 0.01

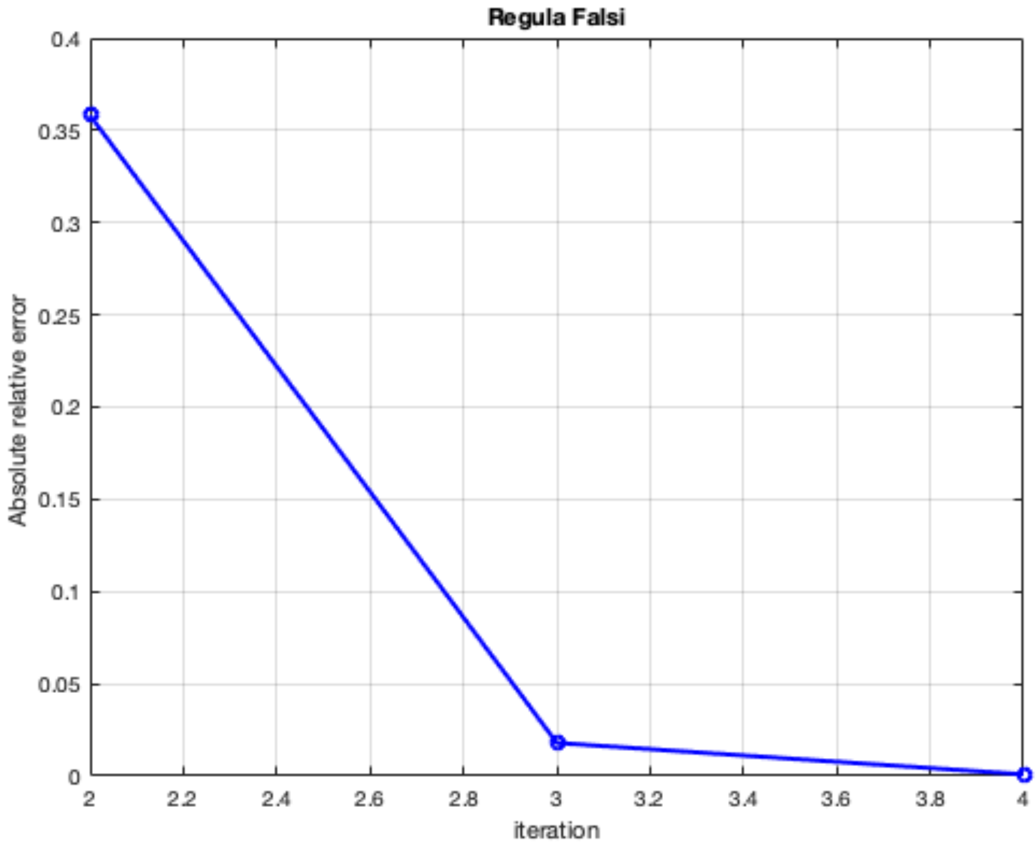
Output :
error condition reached
Root is :
    0.7391



b**) Regula Falsi**
>> eso208_200626_q1
List of methods:
1. Bisection Method
2. Regula Falsi
3. Fixed Point
4. Newton Raphson
5. Secant
Enter method number to use : 2
Enter your function : x-cos(x)
Enter the first guess : 0
Enter the second guess : 1
Enter the max no. of iterations : 50
Enter the max error(%) : 0.01

Output :

Error condition reached
Root is :
0.7391

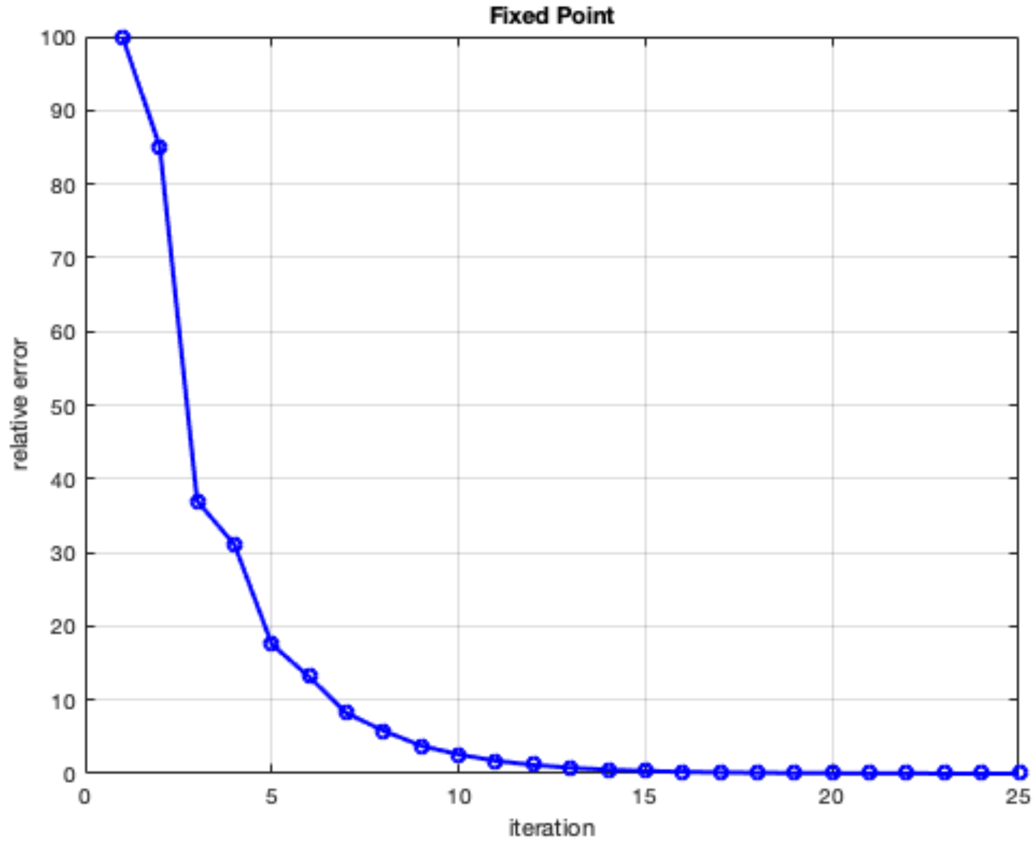


c**) Fixed – point**
>> eso208_200626_q1
List of methods:
1. Bisection Method
2. Regula Falsi
3. Fixed Point
4. Newton Raphson
5. Secant

Enter method number to use : 3
Enter your function : cos(x)
Enter the Inital Guess : 0
Enter the maximum number of iterations : 50
Enter the max error in (%) : 0.01

Output :

error condition reached
Root is :
0.7391

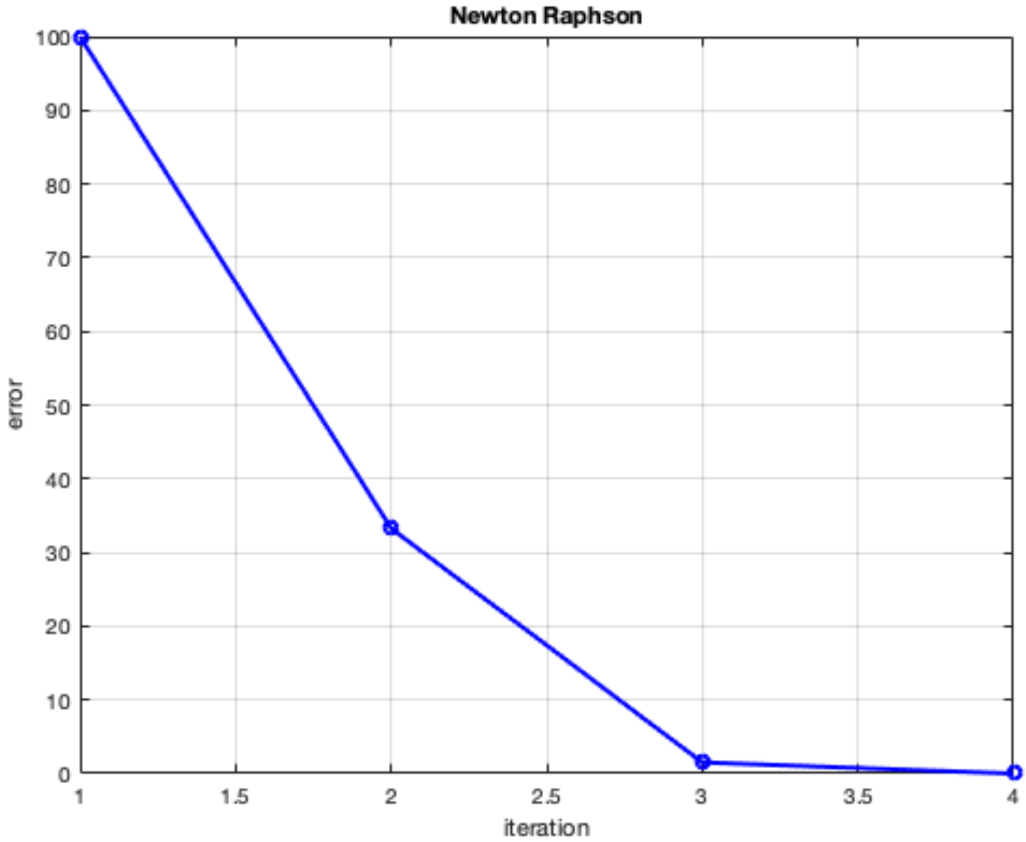


d) **Newton – Raphson**

```
>> eso208_200626_q1
List of methods:
1. Bisection Method
2. Regula Falsi
3. Fixed Point
4. Newton Raphson
5. Secant
Enter method number to use : 4
Enter your function : x-cos(x)
Enter the derivative function :  1+sin(x)
Enter the initial point :  0
Enter the max no. of iterations :  50
Enter the max error in (%) :  0.01
```

Output :

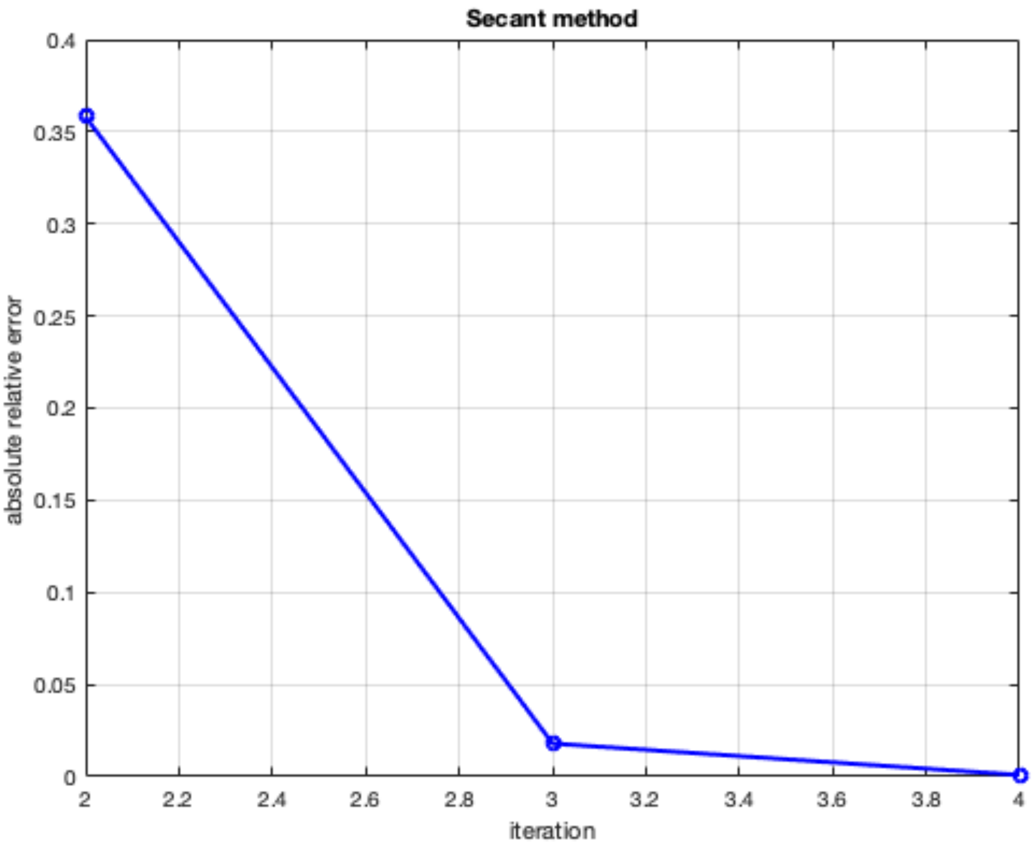
```
error condition reached
Root is :
   0.7391
```



e) **Secant method**
>> eso208_200626_q1
List of methods:
1. Bisection Method
2. Regula Falsi
3. Fixed Point
4. Newton Raphson
5. Secant
Enter method number to use : 5
Enter your function : x-cos(x)
Enter the first guess : 0
Enter the second guess : 1
Enter the max no. of iterations : 50
Enter the max relative error in (%) : 0.01

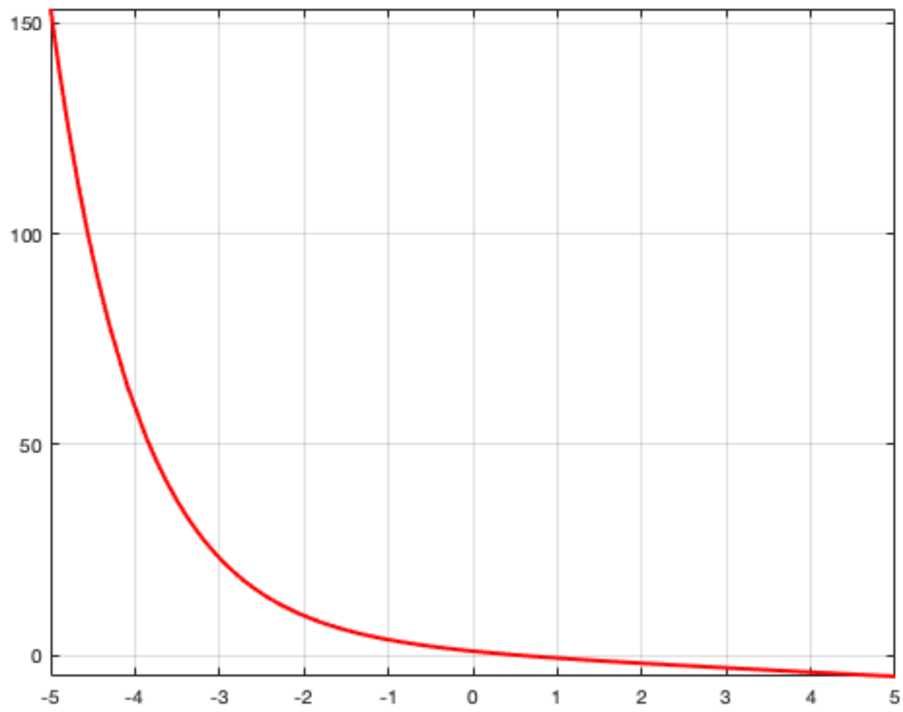
Output:

error condition reached
Root is :
   0.7391


Secant method
absolute relative error
iteration
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
2
2.2
2.4
2.6
2.8
3
3.2
3.4
3.6
3.8
4

(2) $f(x) = exp(-x) - x = 0$

Output plot of this function

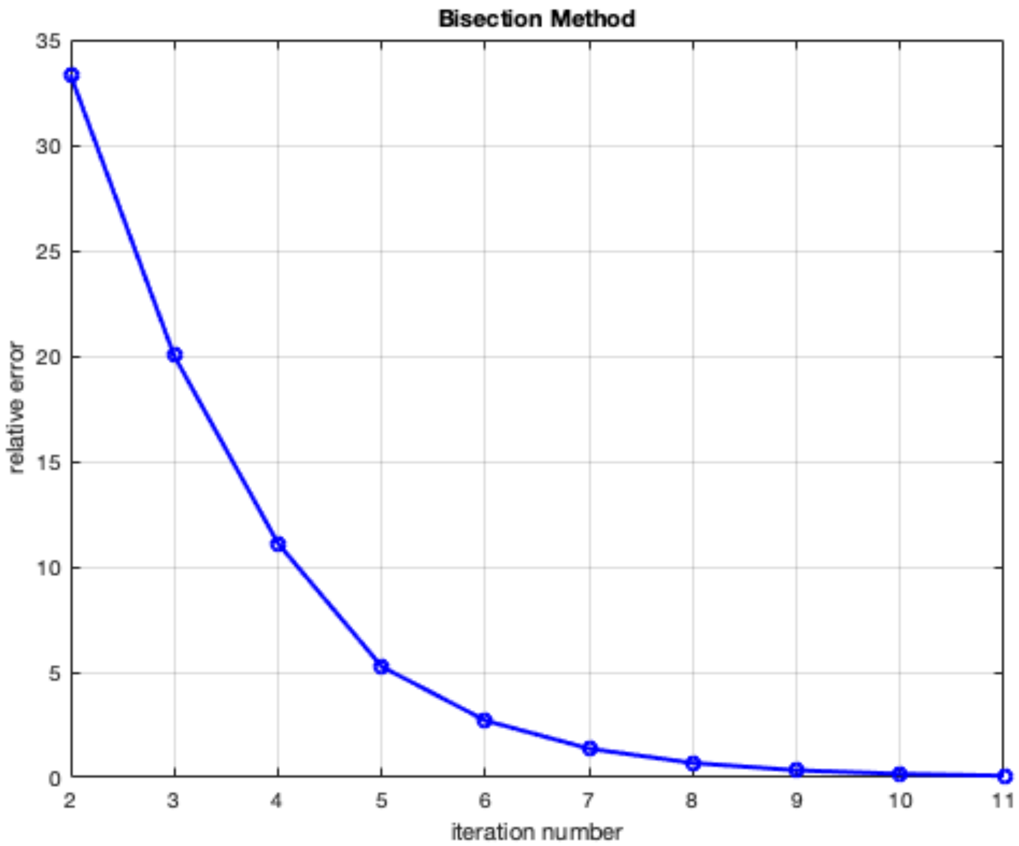


a) Bisection method

>> eso208_200626_q1
List of methods:
1. Bisection Method
2. Regula Falsi
3. Fixed Point
4. Newton Raphson
5. Secant
Enter method number to use : 1
Enter your function : exp(-x)-x
Enter the Value of x_low : 0
Enter the value of x_up : 1
Enter the max no. of iterations : 50
Enter the max relative error in(%) : 0.05

Output :

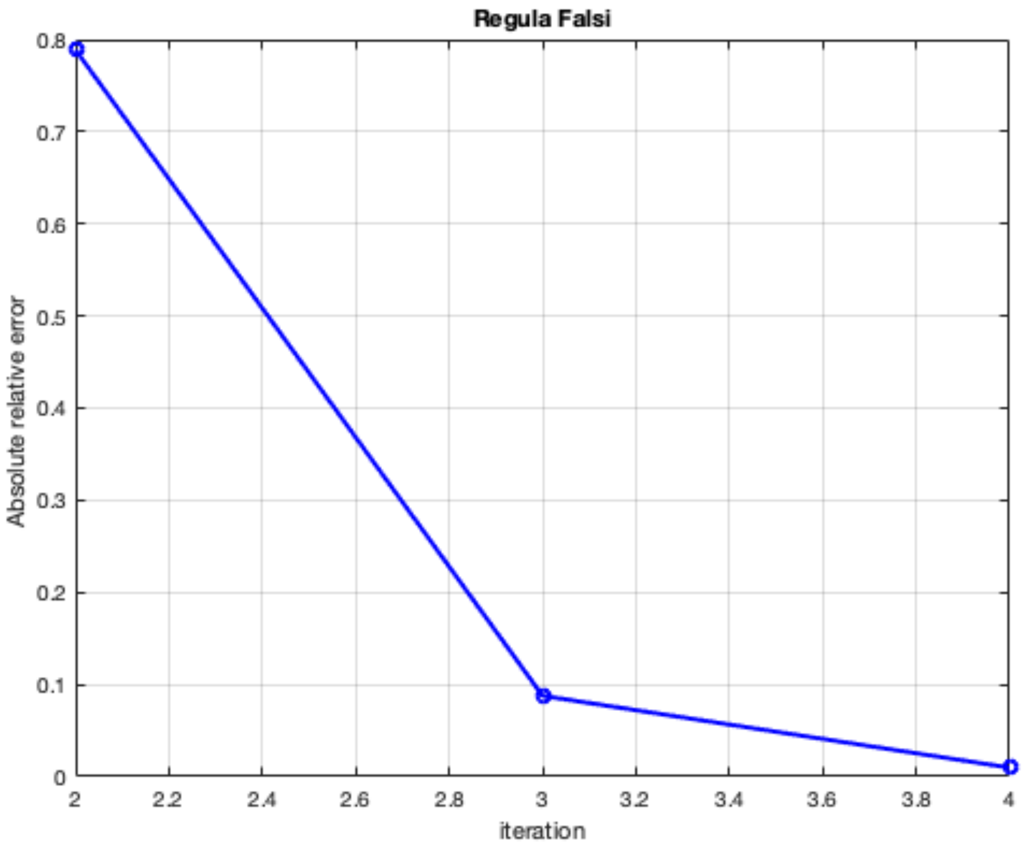
error condition reached
Root is :
0.5672



b**) Regular Falsi method**

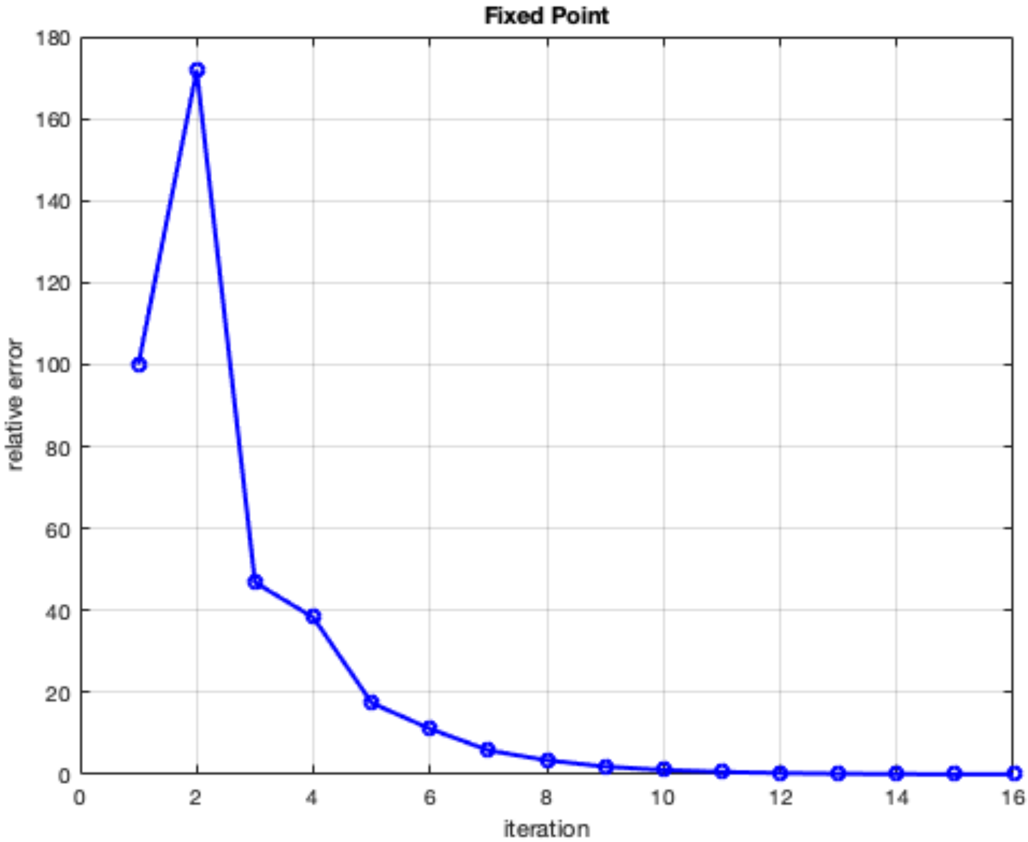
>> eso208_200626_q1
List of methods:
1. Bisection Method
2. Regula Falsi
3. Fixed Point
4. Newton Raphson
5. Secant
Enter method number to use : 2
Enter your function : exp(-x)-x
Enter the first guess : 0
Enter the second guess : 1
Enter the max no. of iterations : 50
Enter the max error(%) : 0.05

Output :
Error condition reached
Root is :
    0.5672



c) Fixed point method
>> eso208_200626_q1
List of methods:
1. Bisection Method
2. Regula Falsi
3. Fixed Point
4. Newton Raphson
5. Secant
Enter method number to use : 3
Enter your function : exp(-x)
Enter the Inital Guess : 0
Enter the maximum number of iterations : 50
Enter the max error in (%) : 0.05

Output :
error condition reached
Root is :
    0.5671

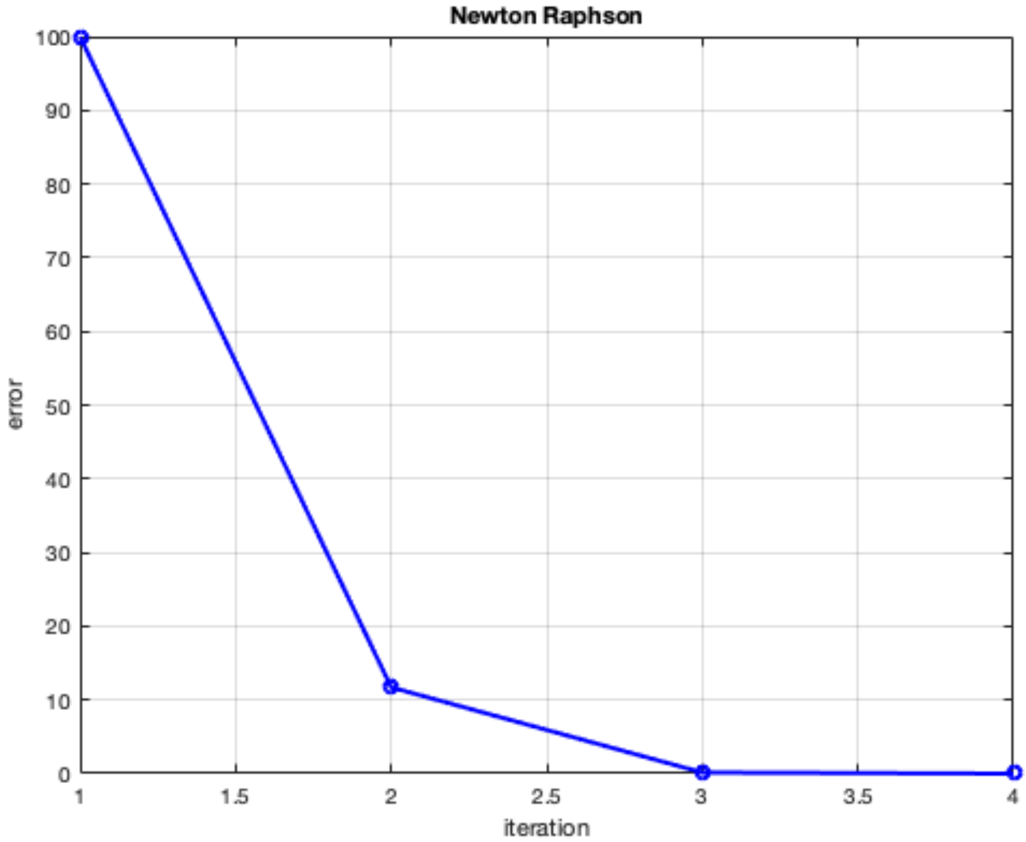


d) Newton – Raphson Method

```
>> eso208_200626_q1
List of methods:
1. Bisection Method
2. Regula Falsi
3. Fixed Point
4. Newton Raphson
5. Secant
Enter method number to use : 4
Enter your function : exp(-x)-x
Enter the derivative function :  -exp(-x)-1
Enter the initial point :  0
Enter the max no. of iterations :  50
Enter the max error in (%) :  0.05
```

Output :

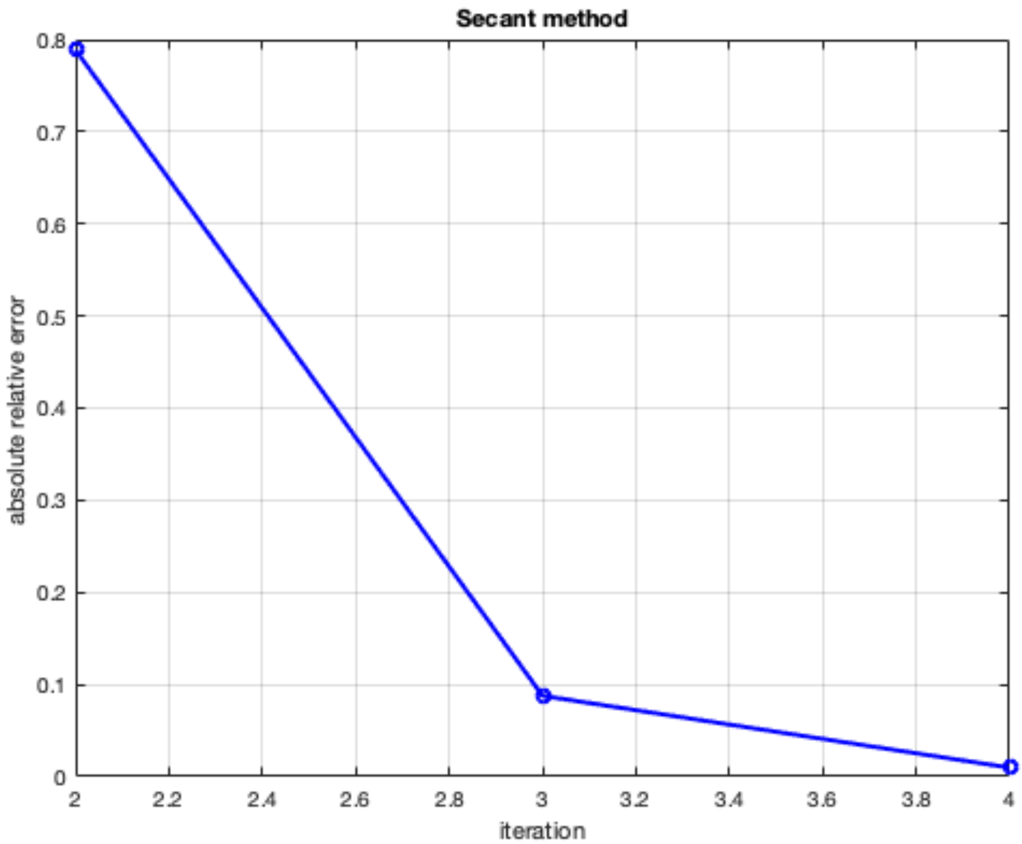
```
error condition reached
Root is :
   0.5671
```



e **) Secant method**
>> eso208_200626_q1
List of methods:
1. Bisection Method
2. Regula Falsi
3. Fixed Point
4. Newton Raphson
5. Secant
Enter method number to use : 5
Enter your function : exp(-x)-x
Enter the first guess : 0
Enter the second guess : 1
Enter the max no. of iterations : 50
Enter the max relative error in (%) : 0.05

Output :

error condition reached
Root is :
   0.5672



Comment on the convergence and stability of different methods.

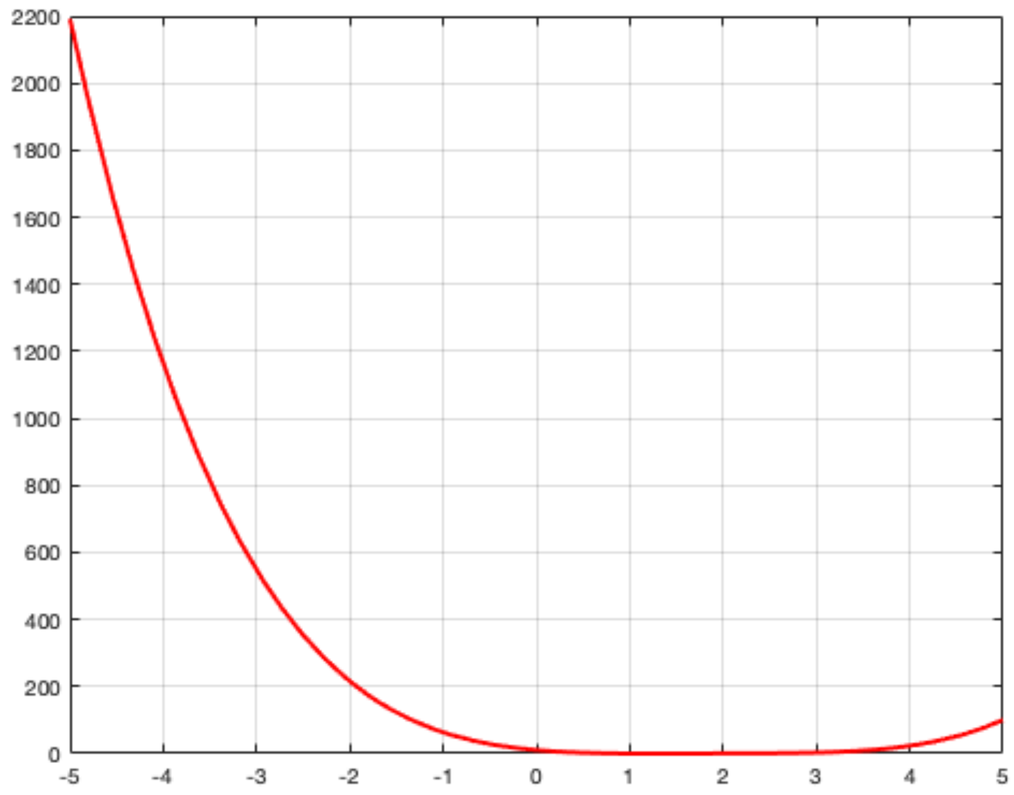
Bracketing methods (Bisection & Regular – Falsi) has slow convergence
Open Methods (Fixed pt. , Newton – Raphson & Secant) has fast convergence but convergence not always occurs.
Secant method has convergence of 0.618 which is better than bisection & false position but not as good as newton.

Q2.

Test polynomial:

$f(x) = x^4 - 7.4x^3 + 20.44x^2 - 24.184x + 9.6448 = 0$

Output plot of this function



a) i) **Muller method for** $(-1,0,1)$

>> eso208_200626_q2

List of methods:

1. Mullers method

2. Bairstows method

Enter method number to use : 1

Enter your function : x.^4-7.4*x.^3+20.44*x.^2-24.184*x+9.6448

Enter the first guess : -1

Enter the second guess : 0

Enter the Third guess : 1

Enter the maximum number of iteration : 50

Enter the max relative error (%): 0.01

Output :
error condition reached
Roots is
0.8000

ii) Muller method for (0,1,2)

>> eso208_200626_q2
List of methods:
1. Mullers method
2. Bairstows method

Enter method number to use : 1
Enter your function : x.^4-7.4*x.^3+20.44*x.^2-24.184*x+9.6448
Enter the first guess : 0
Enter the second guess : 1
Enter the Third guess : 2
Enter the maximum number of iteration : 50
Enter the max relative error (%): 0.01

Output :
error condition reached
Roots is :
2.2000

b) i) **Bairstow method for ($\alpha_0 = -5$, $\alpha_1 = 4$)**

>> eso208_200626_q2
List of methods:
1. Mullers method
2. Bairstows method

Enter method number to use : 2
Enter your function : x.^4-7.4*x.^3+20.44*x.^2-24.184*x+9.6448
Again enter the Test function this time to use power use only " ^ " : x^4-7.4*x^3+20.44*x^2-24.184*x+9.6448
Enter the value of r : -5
Enter the valuse of s : 4
Enter the Number of maximum iterations : 50
Enter the max relative error limit : 0.01

Output :
max iterations reached
The Roots are :
Root 1 is :
2.2000 + 0.8000i

Root 2 is :
2.2000- 0.8000i

ii) **Bairstow method for ($\alpha_0 = -2$, $\alpha_1 = 2$)**

```
>> eso208_200626_q2
List of methods:
1. Mullers method
2. Bairstows method
Enter method number to use : 2
Enter your function : x.^4-7.4*x.^3+20.44*x.^2-24.184*x+9.6448
Again enter the Test function this time to use power use only " ^ " :
x^4-7.4*x^3+20.44*x^2-24.184*x+9.6448
Enter the value of r :  -2
Enter the valuse of s :  2
Enter the Number of maximum iterations :  50
Enter the max relative error limit :  0.01
max iterations reached
```

Output :

```
The Roots are :
Root 1 is  :
   2.2000

Root 2 is :
   0.8000
```

Comment on the convergence and stability of different methods.

Muller method :

Order of convergence for Muller is 1.839 which is better than secant. It is an extension of the secant method. The method does not require derivatives Muller‟s method can converge to a complex root from an initial real number.

Bairstow method :

The major advantage is that it has the capabilities of returning of both real and complex roots, and it may take a long time, but it doesn't fail. An advantage of the method is that it uses real arithmetic only. Since it is a 2nd order method, convergence is relatively fast. It has disadvantage that the farther the starting values from the roots, the longer it takes to converge.