MA202: Lab 6

Name: Hritik Kumar

Roll no.: 202051088

MATLAB Code and Output in Line

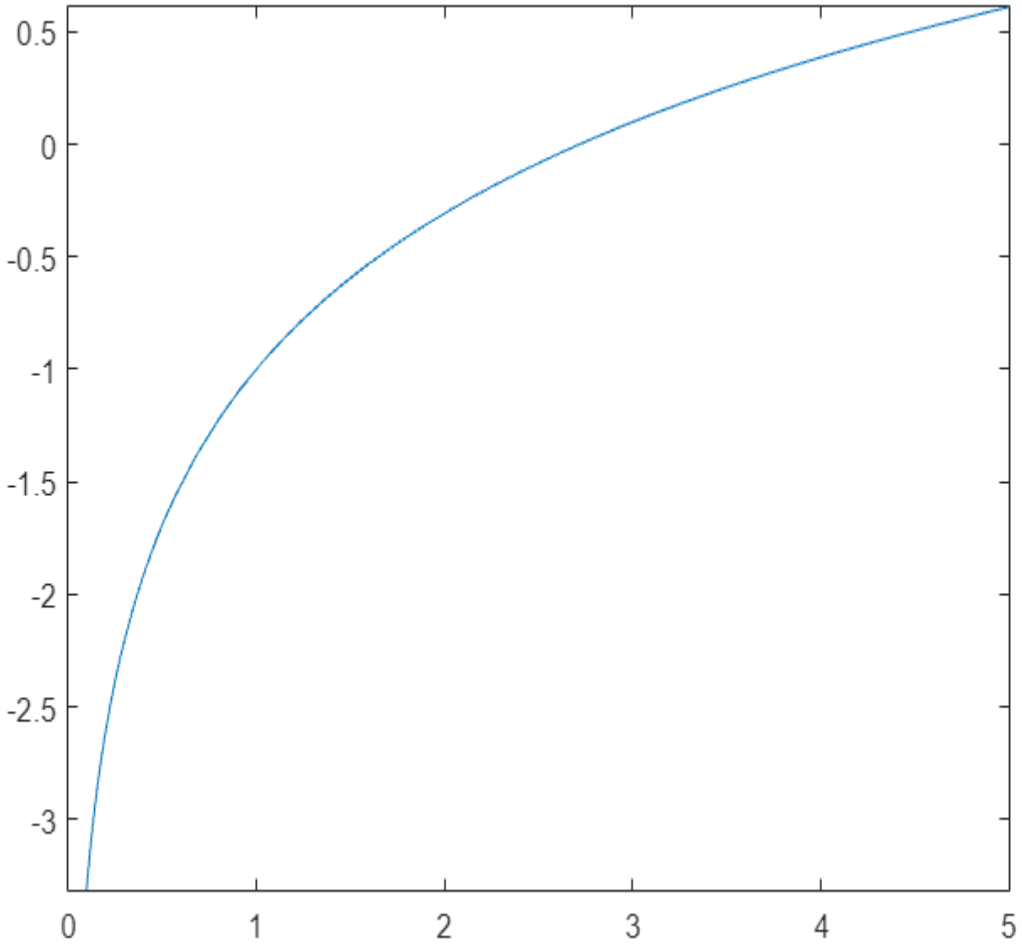
Assignment 6

Exercise 1

```
f = @(x) log(x) - 1
```

```
f = function_handle with value:
    @(x)log(x)-1
```

```
fplot(f)
```



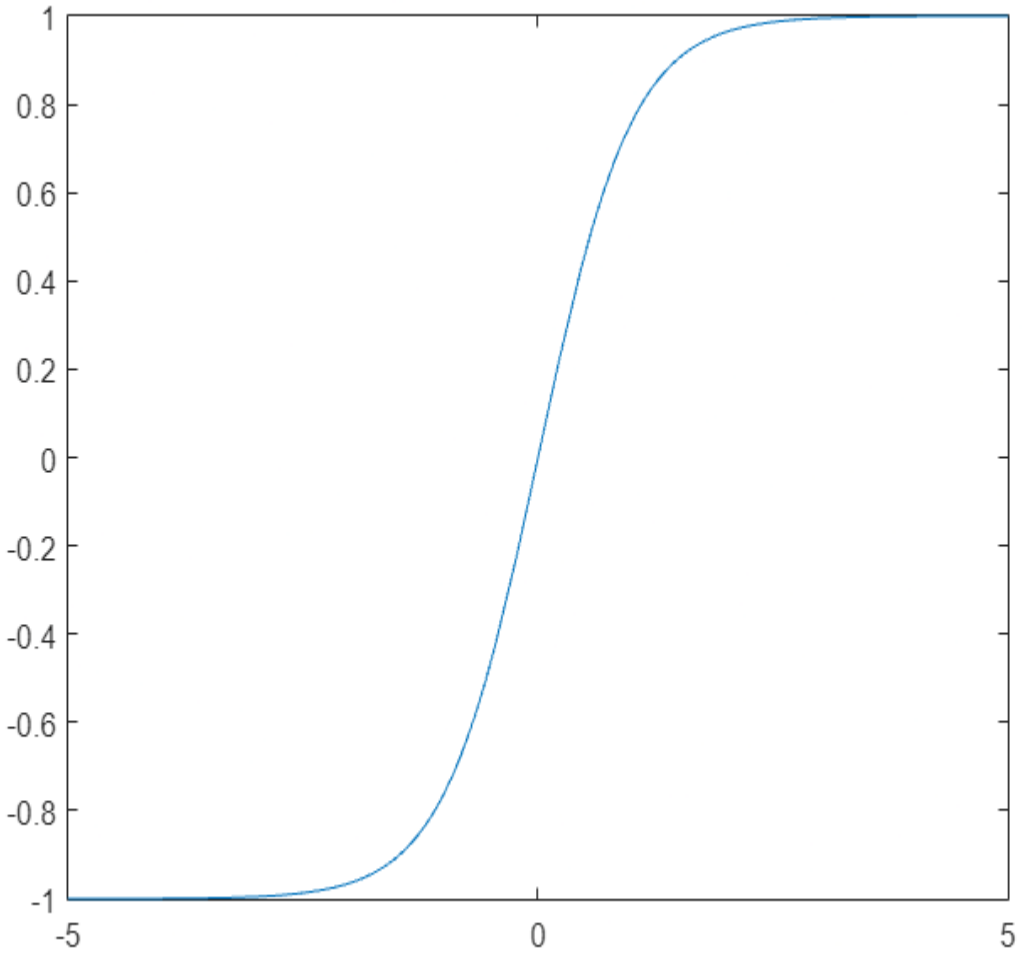

```
callMethod(f,3,2)
Bijection Method :
iteration = 11
root = 2.7183
uncertainity = 7.3980e-06
False Position Method :
iteration = 4
root = 2.7184
ans = 8
uncertainity = 3.7840e-05
Secant Method :
x0 = 2.7568
x0 = 2.7236
```

```
x0 = 2.7182
iteration = 3
root = 2.7182
uncertainity = 1.3868e-05
```

```
newtonRaphsonsMethod(f,3)
Newton Raphsons Method :
x = 2.7042
x = 2.7182
iteration = 2
root = 2.7182
uncertainity = 1.3512e-05
ans = 2.7182
```

```
g = @(x) tanh(x)
g = function_handle with value:
    @(x)tanh(x)
```

```
fplot(g)
```




```
callMethod(g,10,-6);
Bijection Method :
iteration = 3
root = 0
uncertainity = 0
False Position Method :
iteration = 5
root = 2.1105e-07
ans = 8
uncertainity = 2.1105e-07
Secant Method :
x0 = 2.0000
x0 = -1.9268
x0 = 0.0309
x0 = -0.0302
x0 = 2.1105e-07
iteration = 5
```

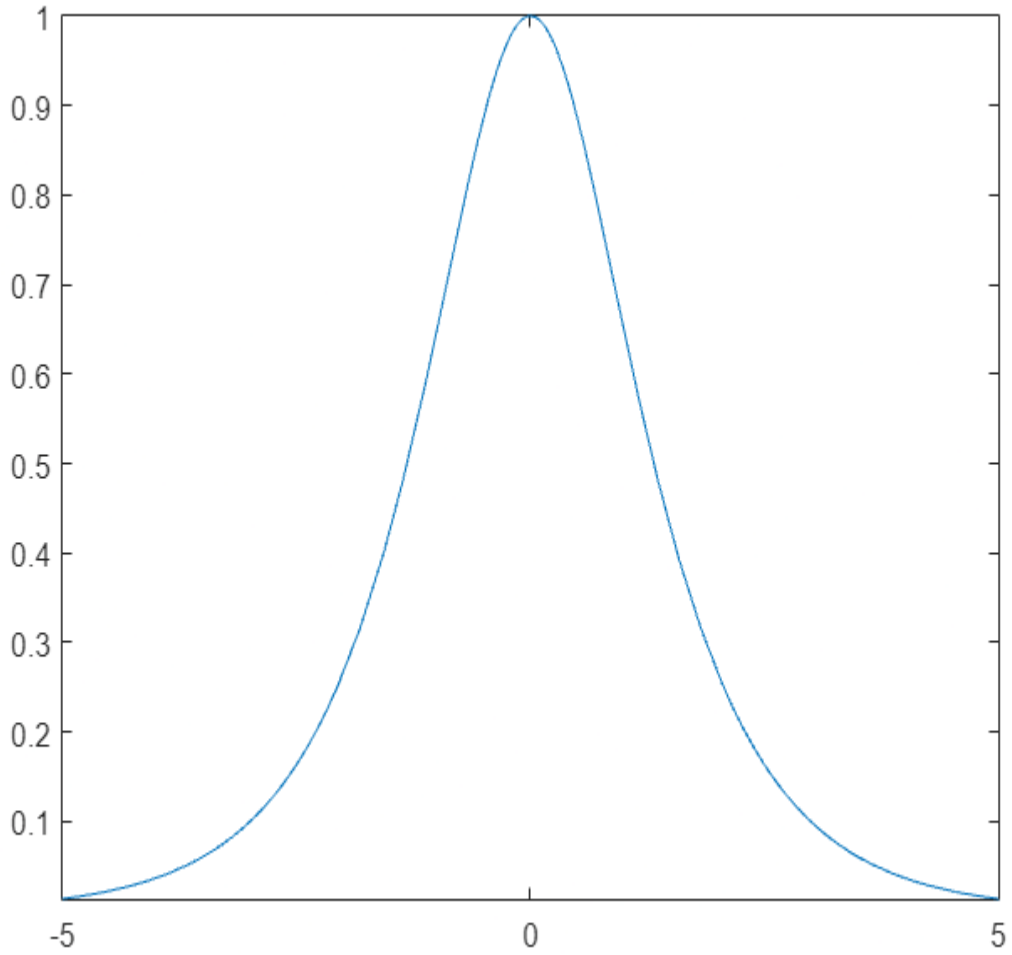
```
root = 2.1105e-07
uncertainity = 2.1105e-07
```

```
newtonRaphsonsMethod(g,10)
Newton Raphsons Method :
x = -1.2129e+08
x = Inf
x = NaN
iteration = 3
root = NaN
uncertainity = NaN
ans = NaN
```

Exercise 2

```
clear;
f = @(x) sech(x)
f = function_handle with value:
    @(x)sech(x)

fplot(f)
```

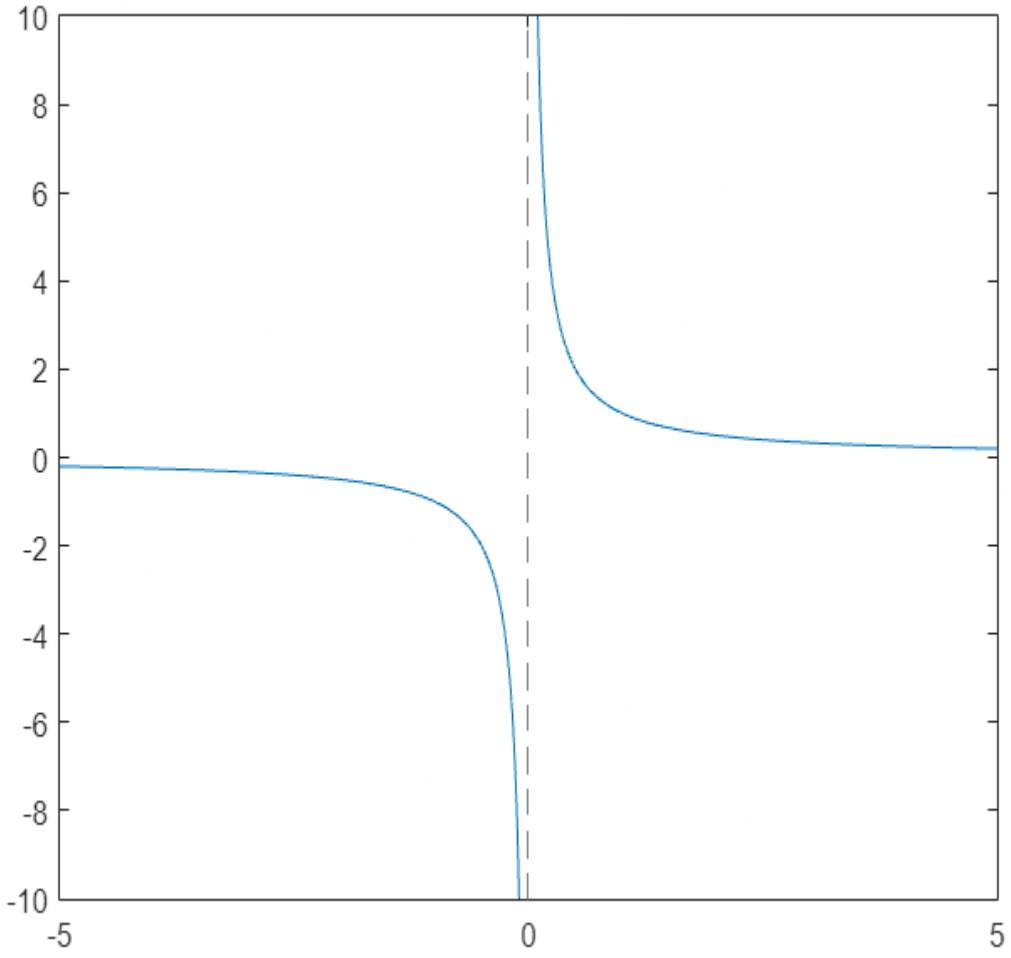



```
newtonRaphsonsMethod(f,10)
Newton Raphsons Method :
x = 11.0000
iteration = 1
root = 11.0000
uncertainity = 3.3403e-05
ans = 11.0000
```

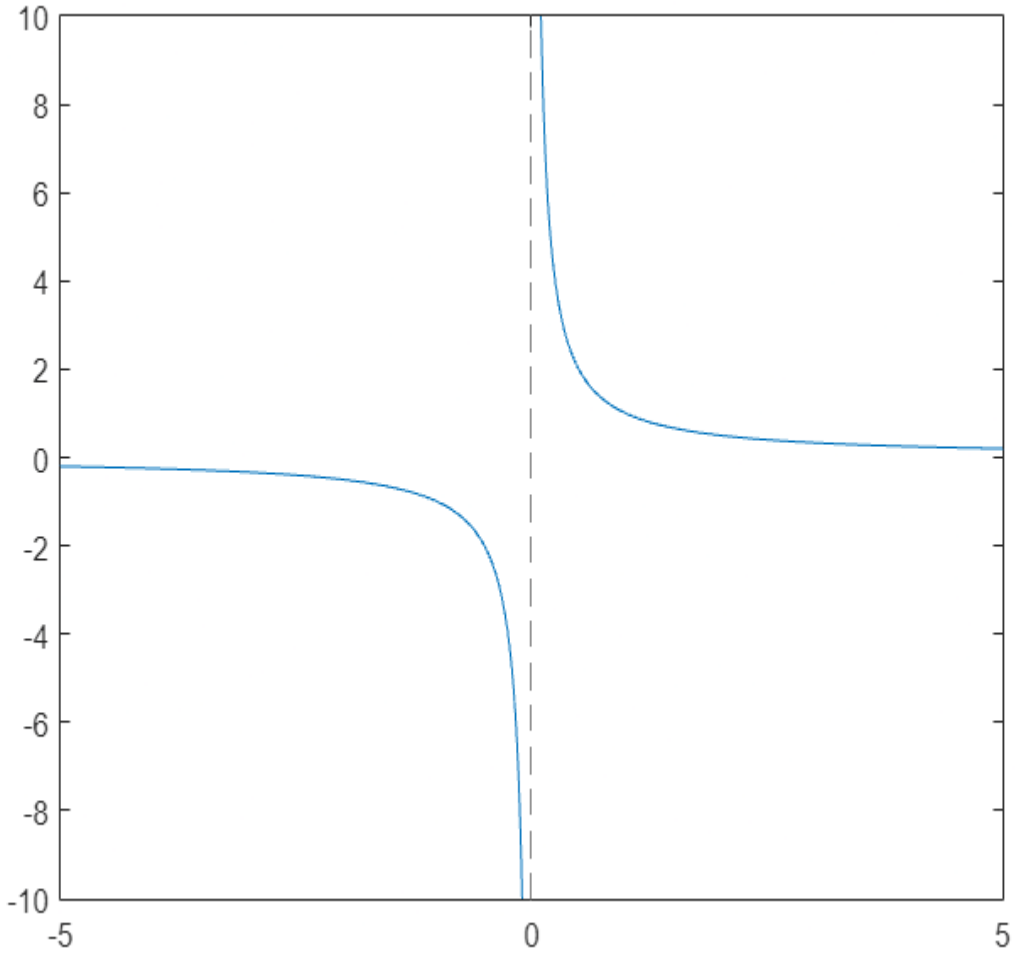
Exercise 3

```
clear;
```

```
f = @(x) 1/x;
fplot(f)
```
```
Warning: Function behaves unexpectedly on array inputs. To improve performance, properly
vectorize your function to return an output with the same size and shape as the input
arguments.
```




```
clear;
f = @(x) 1/x;
fplot(f)
```
```
Warning: Function behaves unexpectedly on array inputs. To improve performance, properly
vectorize your function to return an output with the same size and shape as the input
arguments.
```




```
callMethod(f,-8,10)
```

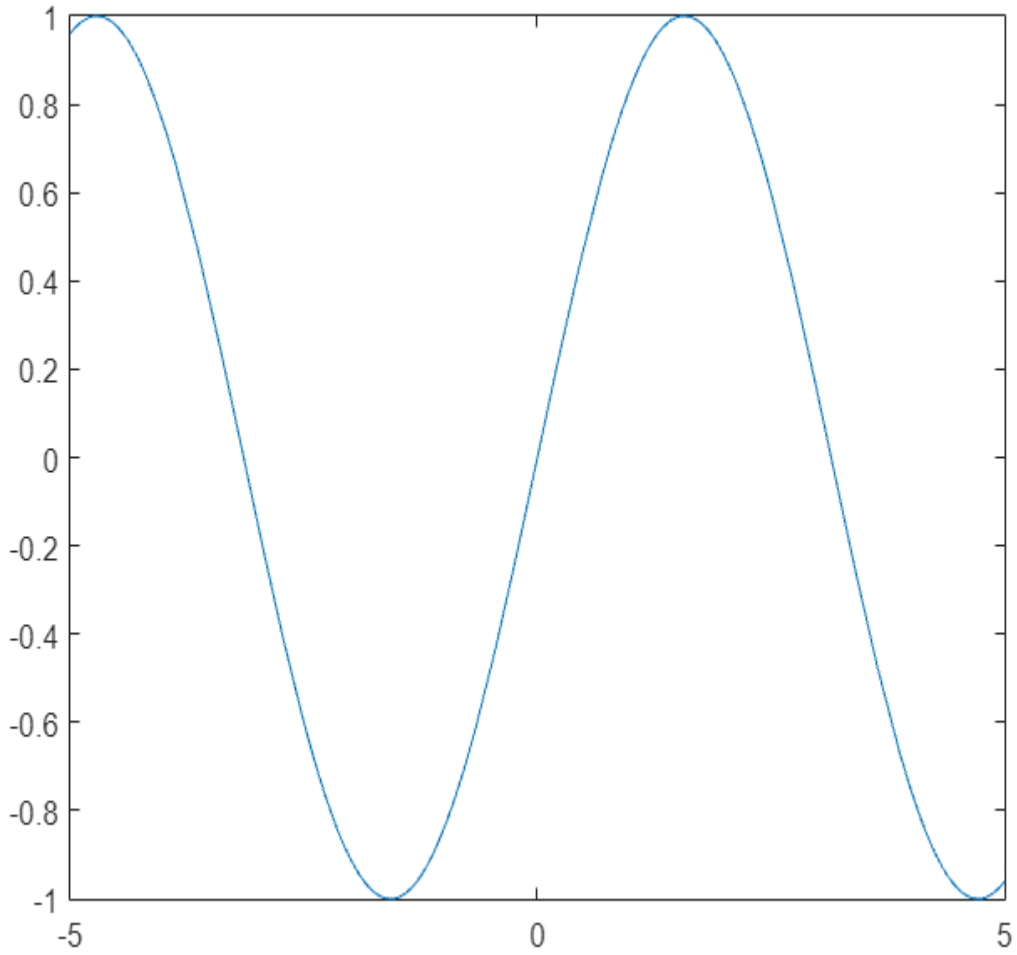
```
Bijection Method : Root could not be found even after 1000 iteration
False Position Method : Root could not be found even after 1000 iteration
Secant Method :
x0 = 2.0000
x0 = 12.0000
x0 = 14.0000
x0 = 26.0000
x0 = 40
x0 = 66.0000
x0 = 106
x0 = 172
x0 = 278
```

```
x0 = 450.0000
x0 = 728
x0 = 1.1780e+03
x0 = 1906
x0 = 3.0840e+03
x0 = 4990
x0 = 8074
x0 = 1.3064e+04
iteration = 17
root = 1.3064e+04
uncertainity = 7.6546e-05
```

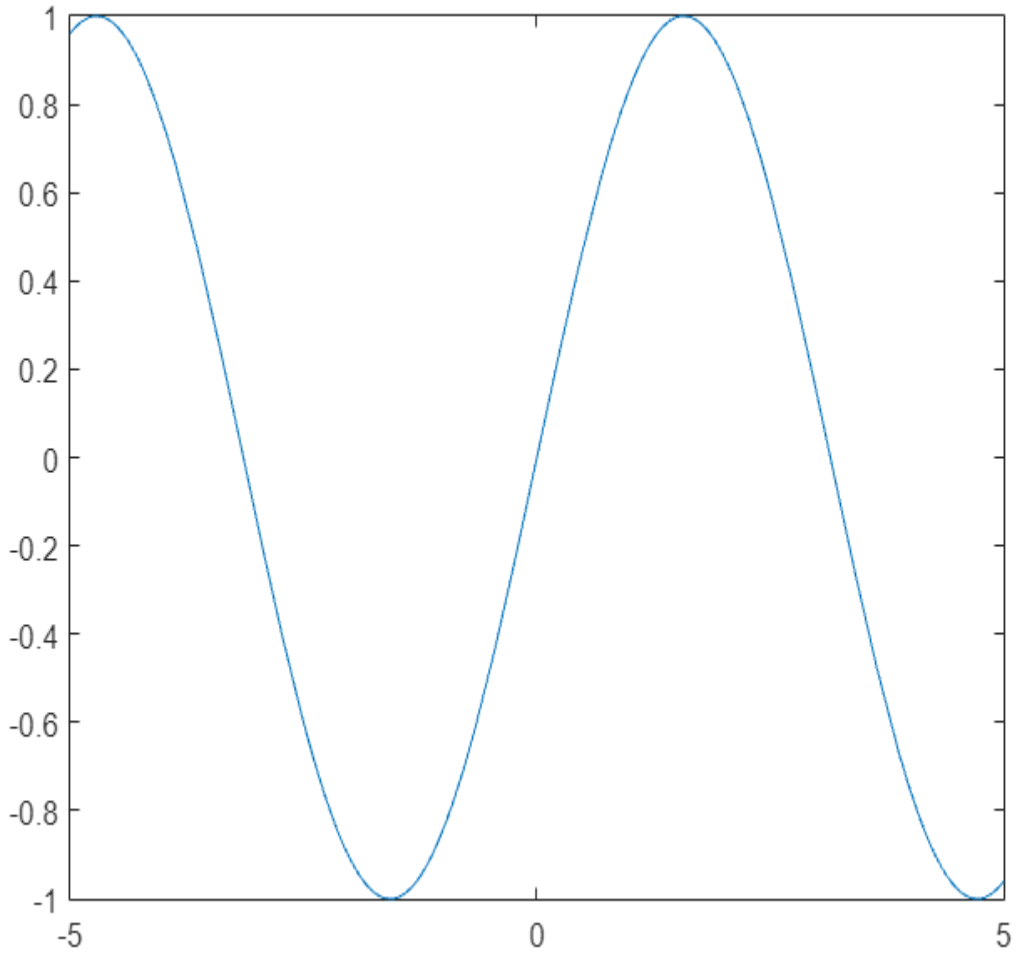
```
newtonRaphsonsMethod(f,10)
Newton Raphsons Method :
x = 20
x = 40
x = 80
x = 160
x = 320
x = 640
x = 1280
x = 2560
x = 5120
x = 10240
iteration = 10
root = 10240
uncertainity = 9.7656e-05
ans = 10240
```

Exercise 4

```
clear;
f = @(x) sin (x);
fplot(f)
```




```
clear;
f = @(x) sin (x);
fplot(f)
```




```
callMethod(f,4,1)
Bijection Method :
iteration = 10
root = 3.1416
uncertainity = 8.9089e-06
False Position Method :
iteration = 4
root = 3.1416
ans = 8
uncertainity = 1.2697e-07
Secant Method :
x0 = 2.5795
x0 = 5.3084
```

```
x0 = 3.6485
x0 = 1.2926
x0 = 2.8581
x0 = 3.5004
x0 = 3.1429
x0 = 3.1416
iteration = 8
root = 3.1416
uncertainity = 2.8112e-05
```

```
newtonRaphsonsMethod(f,2.5)
Newton Raphsons Method :
x = 3.2470
x = 3.1412
x = 3.1416
iteration = 3
root = 3.1416
uncertainity = 2.0137e-11
ans = 3.1416
```

```
function callMethod(f,a,b)
fprintf("Bijection Method : ")
bijectionMethod(f,a,b);
fprintf("False Position Method : ")
falsePositionMethod(f,a,b);
fprintf("Secant Method : ")
secantMethod(f,a,b);
end
function [root,uncertainity] = bijectionMethod(f,a,b)
root = nan;
error = Inf;
epsilon = 10e-5;
iteration = 0;
while error>=epsilon
    % x calculation
    x0 = (a+b)/2;
    if (f(a)*f(x0)>0)
        a = x0;
    else b = x0;
    end
    error = abs(f(x0));
    iteration = iteration +1;
```

```
    if(iteration > 1000)
        disp("Root could not be found even after 1000 iteration")
        return;
    end
end
iteration
root = x0
uncertainity = error
end
function [root,uncertainity] = falsePositionMethod(f,a,b)
root = nan;
error = Inf;
epsilon = 10e-5;
iteration = 0;
while error>=epsilon
    % x calculation
    x0 = (b*f(a) - a*f(b))/(f(a)-f(b));
    if (f(a)*f(x0)>0)
        a = x0;
    else b = x0;
    end
    error = abs(f(x0));
    iteration = iteration +1;
    if(iteration > 1000)
        disp("Root could not be found even after 1000 iteration")
        return;
    end
end
iteration
root = x0
8
uncertainity = error
end
function [root,uncertainity] = secantMethod(f,a,b)
root = nan;
error = Inf;
epsilon = 10e-5;
iteration = 0;
while error>=epsilon
    % x calculation
    x0 = (b*f(a) - a*f(b))/(f(a)-f(b))
    a = b;
    b = x0;
    error = abs(f(x0));
    iteration = iteration +1;
end
iteration
root = x0
uncertainity = error
end
function [root,uncertainity] = newtonRaphsonsMethod(f,x)
```

```
fprintf("Newton Raphsons Method : ")
root = nan;
error = Inf;
epsilon = 10e-5;
iteration = 0;
while error>=epsilon
    % x calculation
    x = x - (f(x))/double(subs(diff(sym(f)),x))
    error = abs(f(x));
    iteration = iteration +1;
end
iteration
root = x
uncertainity = error
end
```