ME 318M Homework #6

Name: Jingsi Zhou

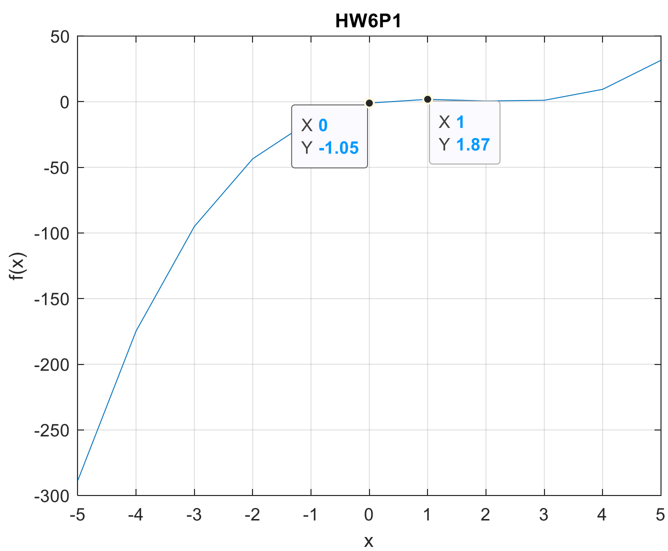
UT EID: jz24729

Section Number: 17460

Problem 1:

```
function value = Hw6_P1(x)
value = x.^3 - 5.1*x.^2 + 7.02*x - 1.05;
```

a) The researcher most likely graphed the function and estimated the range of the root from inspection. Because the researcher can use root-finding methods after obtaining the range, all that is needed are endpoints where the function f(x) changes sign.



b)

```
function value = HW6_P1(x)
value = x.^3 - 5.1*x.^2 + 7.02*x - 1.05;

------

function [Root1, NumIterations] = FalsePostion(xL, xR);

LeftSol = HW6_P1(xL);
RightSol = HW6_P1(xR);

if LeftSol*RightSol < 0
    LeftX = xL;
    RightX = xR;
    count = 0;
    while count < 5
        XApp = LeftX - ((RightX-LeftX)/(RightSol-LeftSol))*LeftSol
        XApp_Sol = HW6_P1(XApp);
        if LeftSol*XApp_Sol < 0
```

```
            RightX = XApp;
        else
            LeftX = XApp;
        end
        count = count + 1;
        disp(['iteration ', num2str(count), ': ', num2str(LeftX), ', ',
num2str(RightX)])
    end
else
    disp(['There is no root between ', num2str(a), ' and ', num2str(b)])
end

Root1 = (['The root of the equation is located between ',
num2str(LeftX), ...
    ' and ', num2str(RightX)]);
NumIterations = count;
```

Output:

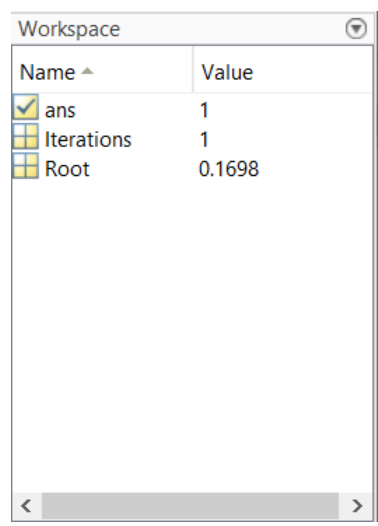
```
>> FalsePostion(0,1)

XApp =

    0.3596

iteration 1: 0, 0.35959

XApp =

    0.1293

iteration 2: 0.1293, 0.35959

XApp =

    0.2121

iteration 3: 0.1293, 0.21211

XApp =

    0.1591

iteration 4: 0.15908, 0.21211

XApp =

    0.1782

iteration 5: 0.15908, 0.17815

ans =

    'The root of the equation is located between 0.15908 and 0.17815'
```

c) I picked the Newton-Raphson's method for root finding because the function was given. Thus, the derivative of the function can be calculated as well, making this method the most efficient.

```
function value = f_derivative(x);
value = 3*x.^2 - 10.2*x + 7.02;

------

function [Root, NumIterations] = NewtonRhapson(x);

x_n = x;
Value_xn = HW6_P1(x_n);
count = 0;

while Value_xn >= (10^(-5))
    Value_xn = HW6_P1(x_n);
    Deriv_xn = f_derivative(x_n);
    x_np1 = x_n - (Value_xn/Deriv_xn);
    x_n = x_np1;
    count = count + 1;
    disp(['iteration ', num2str(count), ': ', num2str(x_n)])
    Value_xn = HW6_P1(x_n);
end

Root = x_n;
NumIterations = count;
```

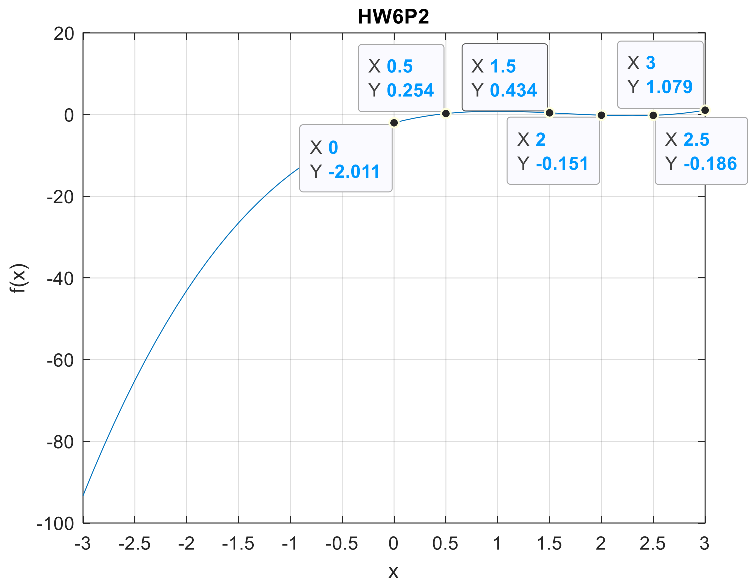
Outputs:

```
    >> HW6_P1(0.17815)

ans =

    0.0444

>> HW6_P1(0.15908)

ans =

   -0.0583

>> [Root, Iterations] = NewtonRhapson(0.17815)
iteration 1: 0.16977

Root =

    0.1698


Iterations =

     1

>> HW6_P1(0.1698)
```



```
ans =

  -1.5172e-04

>> -1.5172e-04 < 10^(-5)

ans =

  logical

   1
```

Problem 2:



a)
```
>> f_derivative(0)

ans =

    6.7300
>> f_derivative(.5)

ans =

    2.5800

>> f_derivative(1.5)

ans =

   -1.2200

>> f_derivative(2)

ans =

   -0.8700

>> f_derivative(3)

ans =

    4.3300

>> f_derivative(2.5)

ans =

    0.9800
```

By plotting the function and through inspection, we can estimate the range for the roots of the function. The function evaluated at the indicated x values on the graph show that there are three roots. This is true because the value of the function at the ranges multiply to a negative number, which means the function crosses the y=0 axis three times. By finding the derivative at specified x-values, we can see that the function increases, decreases, and then increases again. This solidifies the proposal that there are three real roots for this function.

b) Bisection method code:

```
function [Root1, NumIterations] = BisectRootHW6(xL, xR);

Error = input('What do you need your maximum absolute error to be? ');
LeftSol = f(xL);
RightSol = f(xR);

if LeftSol*RightSol < 0
    LeftX = xL;
    RightX = xR;
    count = 0;
    while count < 5
        MidX = (LeftX + RightX)/2;
        Middle = f(MidX);
        if Middle*RightSol < 0
            LeftX = MidX;
        else
            RightX = MidX;
        end
        count = count + 1;
        disp(['iteration ', num2str(count), ': ', num2str(LeftX), ', ',
num2str(RightX)])
    end
else
    disp(['There is no root between ', num2str(xL), ' and ', num2str(xR)])
end

Root1 = (['The root of the equation is located between ', num2str(LeftX), ...
    ' and ', num2str(RightX)]);
NumIterations = count;
```

Newton-Raphson's Code:

```
function [Root, NumIterations] = NewtonRhapson(x);

x_n = x;
Value_xn = f(x_n);
count = 0;

while Value_xn >= (10^(-5))
    Value_xn = f(x_n);
    Deriv_xn = f_derivative(x_n);
    x_np1 = x_n - (Value_xn/Deriv_xn);
    x_n = x_np1;
    count = count + 1;
    disp(['iteration ', num2str(count), ': ', num2str(x_n)])
    Value_xn = f(x_n);
end

Root = x_n;
NumIterations = count;
```

Function f(x) code:

```
function value = f(x)
value = x.^3 - 4.9*x.^2 + 6.73*x - 2.011;
```

Derivative of function f(x) code:

```
function value = f_derivative(x);
value = 3*x.^2 - 9.8*x + 6.73;
```

INTERVAL [0, 0.5] Outputs: **ROOT = 4.120**

```
>> BisectRootHW6(0, 0.5)
iteration 1: 0.25, 0.5
iteration 2: 0.375, 0.5
iteration 3: 0.375, 0.4375
iteration 4: 0.40625, 0.4375
iteration 5: 0.40625, 0.42188

ans =

'The root of the equation is located between 0.40625 and 0.42188'

>> f(0.40625)

ans =

   -0.0186

>> f(0.42188)

ans =

    0.0312

>> [Root, NumIt] = NewtonRhapson(0.40625)1

Root =

    0.4063

NumIt =

     0

>> BisectRootHW6(0.40625, 0.42188)
iteration 1: 0.40625, 0.41407
iteration 2: 0.41016, 0.41407
iteration 3: 0.41016, 0.41211
iteration 4: 0.41113, 0.41211
iteration 5: 0.41162, 0.41211
```

```
ans =

    'The root of the equation is located between 0.41162 and 0.41211'

>> f(0.41162)

ans =

   -0.0013

>> f(0.41211)

ans =

   3.0106e-04

>> [Root, Iterations] = NewtonRhapson(0.41211)
iteration 1: 0.41202

Root =

    0.4120


Iterations =

     1
```

INTERVAL [1.5, 2] Outputs: **ROOT = 1.8508**

```
>> BisectRootHW6(1.5,2)
iteration 1: 1.75, 2
iteration 2: 1.75, 1.875
iteration 3: 1.8125, 1.875
iteration 4: 1.8438, 1.875
iteration 5: 1.8438, 1.8594

ans =

    'The root of the equation is located between 1.8438 and 1.8594'

>> f(1.8438)

ans =

    0.0079

>> f(1.8594)

ans =
```

```
   -0.0097

>> [Root, NumIts] = NewtonRhapson(1.8438)
iteration 1: 1.8507
iteration 2: 1.8508

Root =

    1.8508


NumIts =

     2
```

INTERVAL [2.5, 3] Outputs: **ROOT = 2.6372**

```
>> BisectRootHW6(2.5, 3)
iteration 1: 2.5, 2.75
iteration 2: 2.625, 2.75
iteration 3: 2.625, 2.6875
iteration 4: 2.625, 2.6563
iteration 5: 2.625, 2.6406

ans =

    'The root of the equation is located between 2.625 and 2.6406'

>> f(2.625)

ans =

   -0.0209

>> f(2.6406)

ans =

    0.0060

>> [Root, NumIts] = NewtonRhapson(2.6406)
iteration 1: 2.6372
iteration 2: 2.6372

Root =

    2.6372


NumIts =
```