ECE 111 - Homework #2

Math 103 - Algebra, Functions & Solving f(x) = 0. Make-Up Homework Set for Fall 2024

Newton's Method

1) Let x and y be related by:

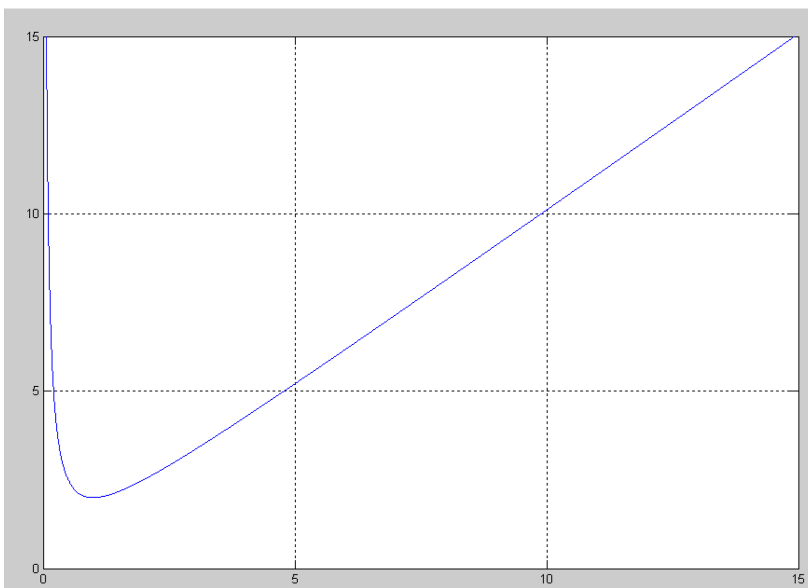
$$y = x + \frac{1}{x}$$

Use Newton's method to solve for x when

- y = 5
- y = 10

It helps to know the answer when trying to find the answer. Start by plotting y(x). In Matlab

```
>> x = [0.01:0.001:15]';
>> y = x + 1./ x;
>> plot(x,y);
>> ylim([0,15])
>> grid
>>
```



From the graph, you can see there are two solutions for y=5 and y=10.

Solution: y = 5

First set up a function where

- You guess x, and
- It returns the error in y

```
function [e] = Prob1(x)
   y = x  + 1/x;
   e = y - 5;
end
```

Next, use Newton's method to find the solution to f(x) = 0. Start with a guess of x=2:

```
x3 = 2;
for i=1:10
    x1 = x3;
    y1 = Prob1(x1);

    x2 = x1 + 0.001;
    y2 = Prob1(x2);

    x3 = x1 - (x2-x1)/(y2-y1) * y1;
    disp([i, x1, y1]);
    pause(0.1);
end
```

Result: x = 4.7913

i	x	error
1	2.000000000000000	-2.500000000000000
2	5.332778147902003	0.520297668171830
3	4.793521858293856	0.002136740954666
4	4.791287916073244	0.000000065607260
5	4.791287847478571	0.000000000000623
6	4.791287847477920	0
7	4.791287847477920	0
8	4.791287847477920	0
9	4.791287847477920	0
10	4.791287847477920	0

Start with a guess of x = 0.1

i	x	error
1	0.100000000000000	5.100000000000000
2	0.152035559147389	1.729444197021115
3	0.193232802281003	0.368337588816065
4	0.207596400118358	0.024635613851674
5	0.208711505713025	0.000014201669741
6	0.208712155760884	-0.000000071112597
7	0.208712152505852	0.000000000356318
8	0.208712152522161	-0.000000000001785
9	0.208712152522080	0.000000000000009
10	0.208712152522080	-0.000000000000001

The second answer is x = 0.02871215

Now find the solution for y = 10. Just change the cost function

```
function [e] = Prob1(x)
   y = x  + 1/x;
   e = y - 10;
end
```

Starting at x=2:

i	x	error
1	2.000000000000000	-7.500000000000000
2	11.998334443711927	2.081679345013964
3	9.902095052391751	0.003083782020079
4	9.898979498911700	0.000000013209153
5	9.898979485566370	0.000000000000014
6	9.898979485566356	0
7	9.898979485566356	0
8	9.898979485566356	0
9	9.898979485566356	0
10	9.898979485566356	0

The solution is x = 9.8989794855

Starting at x = 0.1:

i	x	error
1	0.100000000000000	0.100000000000000
2	0.101020305081321	0.000020305081321
3	0.101020516527160	-0.000000203049733
4	0.101020514412704	0.000000002030919
5	0.101020514433853	-0.000000000020314
6	0.101020514433642	0.000000000000204
7	0.101020514433644	-0.000000000000002
8	0.101020514433644	-0.000000000000002
9	0.101020514433644	0
10	0.101020514433644	0

Another soluion is x = 0.1010205144

2) Let x and y be related by

$$y = \sin(2x)$$

$$y = 4 - x^2$$

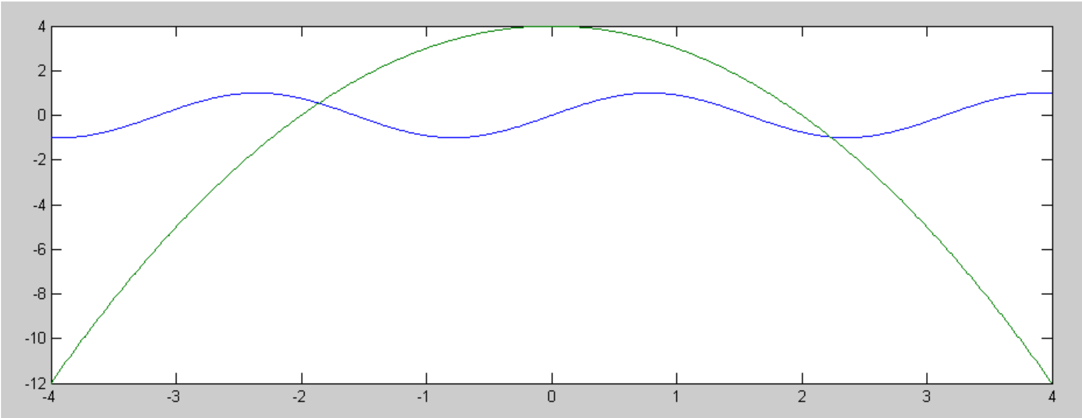
Find all solutions in the range of (-4 < x < 4) using graphical methods. (Plot both functions on the same graph. The solution is when the two functions intersect.)

In Matlab

```
>> x = [-4:0.01:4]';
>> y1 = sin(2*x);
>> y2 = 4 - x.^2;
>> plot(x,y1,x,y2)
>>
```

The solutions are where the two curves intersect

- x = -1.859, y = 0.545
- x = 2.23, y = -0.97



3) Find the solutions to problem #2 using Newton's method.

Let

$$y_1 = \sin(2x)$$

$$y_2 = 4 - x^2$$

$$e = y_1 - y_2$$

Find the solutions for f(x) = 0 using Netwon's method.

First, create a funciton where you guess x and it returns the error:

```
function [e] = Prob3(x)
   y1 = sin(2*x);
   y2 = 4 - x^2;
   e = y1 - y2;
end
```

Use Newton's method to solve.

Make the initial guess -3:

```
x3 = -3;
for i=1:10
    x1 = x3;
    y1 = Prob3(x1);

    x2 = x1 + 0.001;
    y2 = Prob3(x2);

    x3 = x1 - (x2-x1)/(y2-y1) * y1;
    disp([i, x1, y1]);
    pause(0.1);
end
```

i	x	error
1	-3.000000000000000	5.279415498198926
2	-1.705778041765037	-0.823625080880989
3	-1.860053965345311	0.006582171961893
4	-1.858833858658778	-0.000000024460679
5	-1.858833863192756	-0.000000000000401
6	-1.858833863192831	0.000000000000000
7	-1.858833863192831	0.000000000000000
8	-1.858833863192831	0.000000000000000
9	-1.858833863192831	0.000000000000000
10	-1.858833863192831	0.000000000000000

The answer is x = -1.858833863192831

Make the intial guess +3

```
x3 = +3;
for i=1:10
    x1 = x3;
    y1 = Prob3(x1);

    x2 = x1 + 0.001;
    y2 = Prob3(x2);

    x3 = x1 - (x2-x1)/(y2-y1) * y1;
    disp([i, x1, y1]);
    pause(0.1);
end
```

This converges to the other solution

i	x	error
1	3.000000000000000	4.720584501801074
2	2.404109415700676	0.784330249987927
3	2.247324142004158	0.074077794055570
4	2.229103300775707	0.001032312121802
5	2.228842499615354	0.000000965406150
6	2.228842255622775	0.000000000716491
7	2.228842255441692	0.000000000000531
8	2.228842255441558	0.000000000000000
9	2.228842255441558	0.000000000000000
10	2.228842255441558	0.000000000000000

The other solution is x = 2.228842255441558

Newton's Method with a CdS Light Sensor

Assume the light - resistance relationship of a CdS light sensor:

$$R = 5000 \cdot (lux)^{-0.6}\Omega$$

$$e = R - R_0$$

4) Write a Matlab funciton which

- Is passes the light level in lux, and
- Returns e (the difference between R and R0)

```
function [e] = Prob4(lux)
   R0 = 900;
   R = 5000 * (lux) ^ (-0.6);
   e = R - R0;
end
```

5) Use Newton's method to find the temperature when

- R0 = 900 Ohms
- R0 = 600 Ohms

Start with a guess of 10 Lux

```
x3 = 10;
for i=1:10
    x1 = x3;
    y1 = Prob4(x1);

    x2 = x1 + 0.001;
    y2 = Prob4(x2);

    x3 = x1 - (x2-x1)/(y2-y1) * y1;
    disp([i, x1, y1]);
    pause(0.1);
end
```

R0 = 900 Ohms:

i	Lux	error
1	10.000000000000000	355.9432157547901
2	14.723829423668382	95.766560491249152
3	17.084033059771691	10.784775524031375
4	17.421207648714404	0.166921764968379
5	17.426592043162437	0.000033598225173
6	17.426593127476178	-0.000000001540798
7	17.426593127426450	0.000000000000227
8	17.426593127426457	0
9	17.426593127426457	0
10	17.426593127426457	0

When R = 900 Ohms, the light level is 17.426593127426457 lux

R0 = 600 Ohms

i	Lux	error
1	10.0000000000000	655.9432157547901
2	18.7052196128181	262.5659642410335
3	28.195438872607234	74.314795667713838
4	33.374517802663576	9.426036324822235
5	34.234881869814203	0.189929454942444
6	34.252938277399338	0.000075666165003
7	34.252945476979022	-0.000000001754415
8	34.252945476812087	0
9	34.252945476812087	0
10	34.252945476812087	0

When R = 600 Ohms, the light level is 34.252945476812087 lux

Newton's Method and a Voltage Divider

Assume

$$V = \left(\frac{R}{R+500}\right) \cdot 5V$$

$$e = V - V_0$$

6) Write a Matlab function which

- Is passed the light level in lux, and
- Returns the error, e.

```
function [e] = Prob6(lux)
   V0 = 3.20;
   R = 5000 * (lux) ^ (-0.6);
   V = R / (500+R) * 5;
   e = V - V0;
end
```

7) Use Netwon's method to determine the light level (lux) when

- V0 = 3.20V
- V0 = 2.20V

```
x3 = 10;
for i=1:10
    x1 = x3;
    y1 = Prob6(x1);

    x2 = x1 + 0.001;
    y2 = Prob6(x2);

    x3 = x1 - (x2-x1)/(y2-y1) * y1;
    disp([i, x1, y1]);
    pause(0.1);
end
```

When V0 = 3.20V

i	lux	error
1	10.000000000000000	0.376263755245993
2	16.158403808113093	0.065983930426075
3	17.727318027052636	0.002483932601756
4	17.791063895322207	0.000003647643175
5	17.791157785697976	-0.000000000077282
6	17.791157783708734	0.000000000000002
7	17.791157783708780	0
8	17.791157783708780	0
9	17.791157783708780	0
10	17.791157783708780	0

When V0 = 3.20V, the light level is 17.791157783708780 lux

When V0 = 2.20V

1	10.000000000000000	1.376263755245993
2	32.525656093911579	0.565702209379918
3	57.339410087567217	0.141702301921733
4	68.216620072577257	0.012492837418121
5	69.368150921781577	0.000111428556644
6	69.378607589505606	0.000000008141160
7	69.378608353610971	-0.000000000000063
8	69.378608353605060	0.000000000000000
9	69.378608353605102	-0.000000000000000
10	69.378608353605060	0.000000000000000

When V0 = 2.20V, the light level is 69.378608353605060 lux

Comments: Matlab & Newton's method allows you to solve pretty much any fuction using numerical methods as long as you can set it up as f(x) = 0. It's a pretty useful tool.