ECE 111 - Homework #1

Week #1: Matlab Introduction.

Bison Academy: Homework Sets & Solutions

1) What are the solutions to

$$y = \left(\frac{\sin(x)}{x^2+1}\right)$$

$$y = \cos(x)$$

hint: See homework #2, problem #4 solutions for Spring 2023

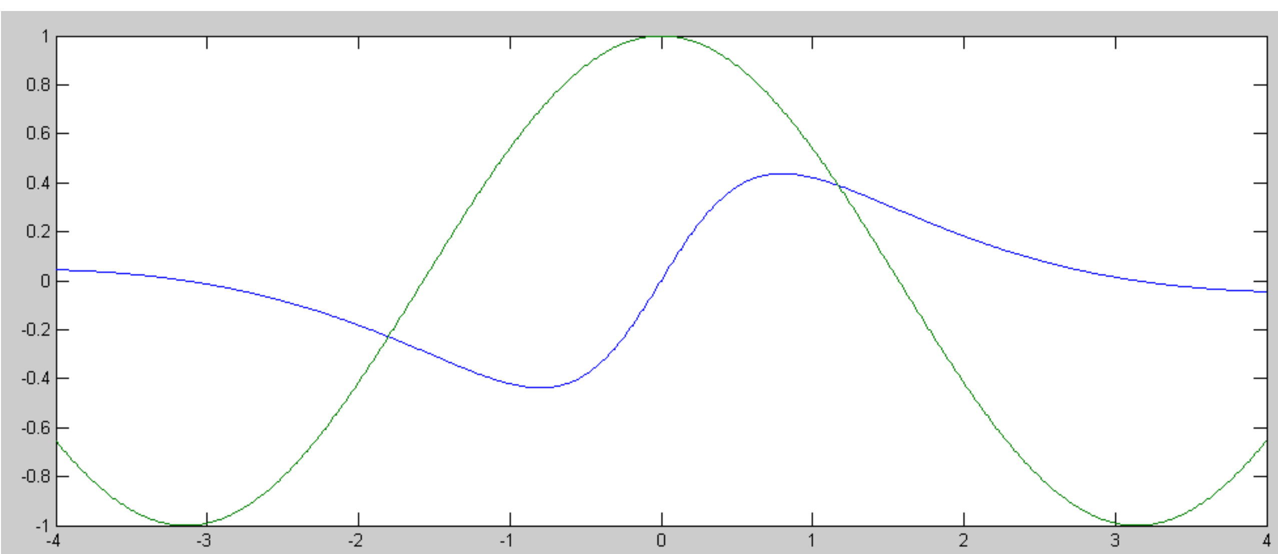
Solution #1:

```
x = -1.802028387303862
y = -0.229176966114031
```

Solution #2:

```
x = 1.172093617285669
y = 0.388223131286656
```

Comment: Each semester, different homework probems are assigned. The approach and the Matlab commands used to solve these problems are similar, however. If you get stuck and need help with a problem, looking at the solutions to previous semester's problems usually helps.



Roots to a Polynomial

2) Use the *roots()* command to find the roots to

a) $y = x^3 - 13x^2 + 35x + 49$

```
>> roots([1, -13, 35, 49])

   7.0000 + 0.0000i
   7.0000 - 0.0000i
  -1.0000
```

b) $y = x^4 - 26x^3 + 125x^2 + 572x + 420$

```
>> roots([1, -26, 125, 572, 420])

   15.0000
   14.0000
   -2.0000
   -1.0000
```

c) $y = x^5 + 15x^4 + 58x^3 - 120x^2 - 1184x - 1920$

```
>> roots([1, 15, 58, -120, 1184, -1920])

  -9.2060 + 4.6750i
  -9.2060 - 4.6750i
   0.9105 + 3.2389i
   0.9105 - 3.2389i
   1.5911
```

Note: The roots to a polynomial are the zero crossings

- Real roots means y(x) passes through the x-axis at the root
- Complex roots means y(x) misses the x axis (stays above or below)

Matlab as a Graphing Calculator: (Thermistor equations)

Assume a thermistor (temperature sensor) and voltage divider have the following relationship:

$$R = 2000 \cdot \exp\left(\frac{4200}{T+273} - \frac{4200}{298}\right)\Omega$$

$$V = \left(\frac{R}{R+4000}\right) \cdot 10V$$

3) Determine the resistance and voltage if

- T = -20 degrees C
- T = +20 degrees C

Solution in Matlab: T = -20C

```
>> T = -20;
>> R = 2000 * exp(4200 / (T+273) - 4200/298)

R =   2.4532e+004

>> V = R / (R + 4000) * 10

V =    8.5981
```

Solution: T = +20C

```
>> T = 20;
>> R = 2000 * exp(4200 / (T+273) - 4200/298)

R =  2.5438e+003

>> V = R / (R + 4000) * 10

V =    3.8873
```

Note: Matlab can be used like a regular calculator

4) Plot the resistance vs. temperature for -20C < T < +20C. From the graph, determine

- The temperature if R = 5,000 Ohms
- The temperature if R = 10,000 Ohms

For this problem, use a bunch of temperatures from -20C to +20C. In this example, let T cover this range with steps of 0.01C (4001 different temperatures in T).

Use Matlab to calculate the resistance at each of the 4001 points in T then plot the results

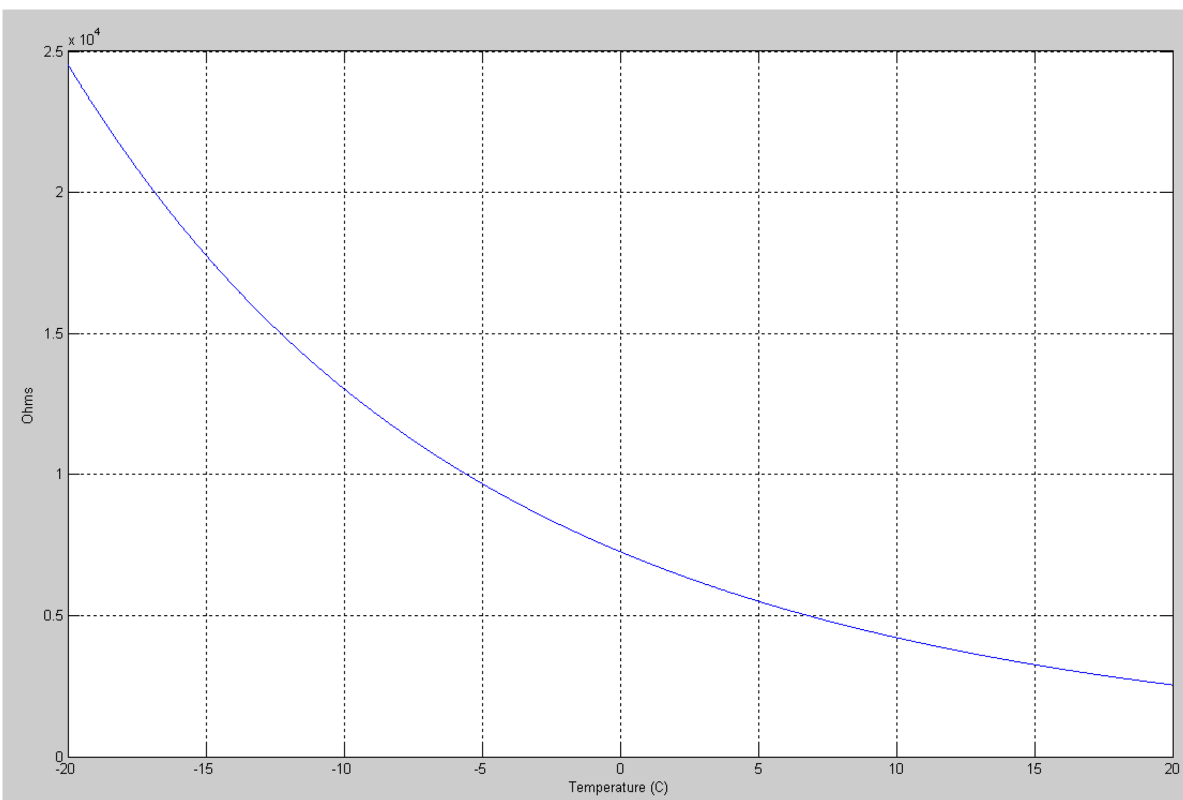
- note: Use the dot-notation telling Matlab to do an element-by-element division (treat each number like a different calculation)

```
>> T = [-20:0.01:20]';
>> R = 2000 * exp(4200 ./ (T+273) - 4200/298);
>> plot(T,R)
>> xlabel('Temperature (C)');
>> ylabel('Ohms');
>> grid
```

From the graph

- R = 5000 Ohms, T = 6 degrees (approx)
- R = 10,000 Ohms, T = -6 degrees (approx)

There are ways to get more accurate answers (next week)



5) Plot the votlage vs. temperature for -20C < T < +20C. From the graph, determine

- The temperature if V = 8.00 Volts
- The temeprature if V = 6.00 Volts

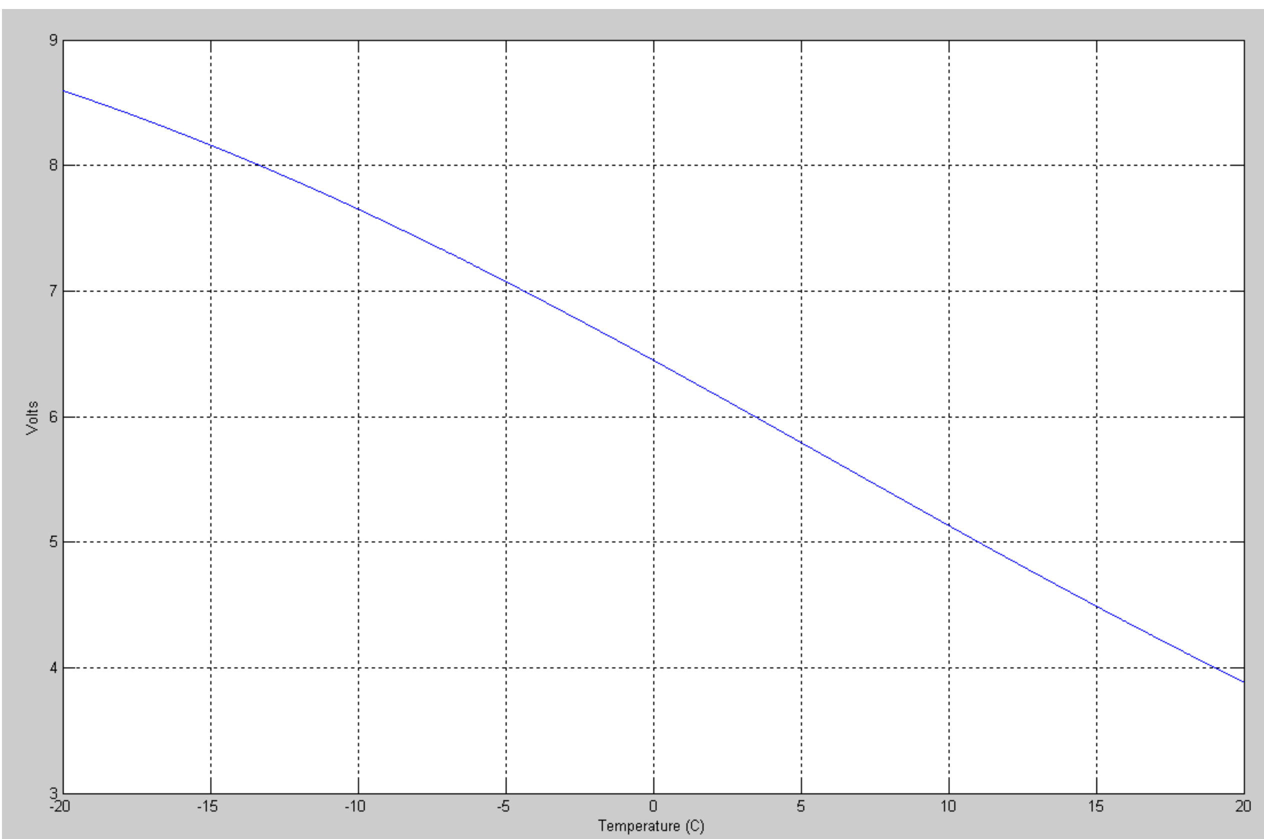
In Matlab:

```
>> T = [-20:0.01:20]';
>> R = 2000 * exp(4200 ./ (T+273) - 4200/298);
>> V = R ./ (R + 4000) * 10;
>> plot(T,V);
>> xlabel('Temperature (C)');
>> ylabel('Volts');
>> grid
```

From the graph

- V = 8.00V, T = -13C
- V = 6.00V, T = 3.5C

(again, there are ways to get more accurate results - these are covered next week)



For-Loops

6) A and B are playing a game

- A rolls five 10-sided dice and takes the sum (A = 5d10)
- B rolls four 12-sided dice and takes the sum (B = 4d12).

Whoever has the higher total wins. Determine the odds that A wins / ties / loses using a Monte-Carlo simulation with 100,000 games.

In Matlab

- Start with playing a single game
- When you get this to work, play 10 matches, displaying the results after each match
- If this works, repeat 100,000 times (add a for-loop).

Start with 10 matches

```
W = 0;
T = 0;
L = 0;
tic
for i=1:1e1
    d10 = ceil(10*rand(5,1));
    d12 = ceil(12*rand(4,1));
    A = sum(d10);
    B = sum(d12);
    if(A > B) W = W + 1; end
    if(A == B) T = T + 1; end
    if(A < B) L = L + 1; end
    disp([A, B, W, T, L])
end
toc
```

The results are:

A	B	W	T	L
18	34	0	0	1
27	20	1	0	1
29	23	2	0	1
33	11	3	0	1
19	37	3	0	2
23	37	3	0	3
16	31	3	0	4
35	34	4	0	4
22	10	5	0	4
18	27	5	0	5

Checking:

- When B > A, losses increment by one
- When A > B, wins increments by one

This looks OK, so now run 100,000 matches

- remove the disp command

New Matlab Code

- play 100,000 matches
- comment out the disp command for each match

```
W = 0;
T = 0;
L = 0;
tic
for i=1:1e5
    d10 = ceil(10*rand(5,1));
    d12 = ceil(12*rand(4,1));
    A = sum(d10);
    B = sum(d12);
    if(A > B) W = W + 1; end
    if(A == B) T = T + 1; end
    if(A < B) L = L + 1; end
%    disp([A, B, W, T, L])
end
toc
disp([W, T, L] / 1e5);
```

The result is:

```
Elapsed time is 1.498729 seconds.

      W          T          L
    0.5429    0.0415    0.4156
```

meaning team A has a

- 54.29% chance of winning
- 4.15% chance of a tie, and
- 41.56% chance of a loss

Note: This is one of the reasons Matlab is so powerful. This isn't an easy problem to solve. With a Monte-Carlo simulation, however, you can determine the odds pretty easily:

- If you can run the experiment one time,
- You can run the experiment 100,000 times

7) A and B are playing a match. For any given game,

- A has a 55% chance of winning (+1 point for A), and
- A has a 45% chance of losing (+1 point for B).

If the match consists of seven games, determine the odds that A wins the match

- A has 4 or more points

Similar to probem #6

- Write the Matlab code to play one match of seven games.
- Start with 10 matches, displaying the results
- Once you get that to work, play 100,000 matches

Starting with 10 matches

```
W = 0;
T = 0;
L = 0;
tic
for i=1:1e1
    A = 0;
    B = 0;
    for j=1:7
        if(rand < 0.55)
            A = A + 1;
        else
            B = B + 1;
        end
    end
    if(A > B) W = W + 1; end
    if(A == B) T = T + 1; end
    if(A < B) L = L + 1; end
    disp([A, B, W, T, L])
end
toc
```

A	B	W	T	L
3	4	0	0	1
4	3	1	0	1
3	4	1	0	2
1	6	1	0	3
4	3	2	0	3
4	3	3	0	3
5	2	4	0	3
3	4	4	0	4
3	4	4	0	5
4	3	5	0	5

Check:

- When A > B, wins increment (check)
- When A < B, losses increments (check)

Looks like this code works

Now that the code works,

- Run 100,000 times, and
- Commend out the display for each match

```
W = 0;
T = 0;
L = 0;
tic
for i=1:1e5
    A = 0;
    B = 0;
    for j=1:7
        if(rand < 0.55)
            A = A + 1;
        else
            B = B + 1;
        end
    end
    if(A > B) W = W + 1; end
    if(A == B) T = T + 1; end
    if(A < B) L = L + 1; end
%    disp([A, B, W, T, L])
end
toc
disp([W, T, L] / 1e5)
```

The result is

```
Elapsed time is 0.106048 seconds.

       W            T       L
    0.6088          0    0.3912
```

Player A has a 60.88% chance of winning any given match

Comment: Again, this isn't an easy problem to solve. With Matlab and Monte-Carlo simulations, you can determine the odds (roughly) by playing 100,000 matches and seeing how many times A wins.

While-Loops

8) A and B are playing a match. For any given game,

- A has a 55% chance of winning (+1 point for A), and
- A has a 45% chance of losing (+1 point for B).

If the match continues until one player is up by 3 or more games, determine

- The odds that A wins (A has 3 or more points than B)
- Using a Monte-Carlo simulation with 100,000 matches

Comment: This is a while-loop. Keep playing until abs(A - B) >= 3

- Change the for-loop to a while-loop
- Play just 10 matches to check the code
- Display everything after each match

```
W = 0;
T = 0;
L = 0;
tic
for i=1:1e1
    A = 0;
    B = 0;
    while( abs(A - B) < 3)
        if(rand < 0.55)
            A = A + 1;
        else
            B = B + 1;
        end
    end
    if(A > B) W = W + 1; end
    if(A == B) T = T + 1; end
    if(A < B) L = L + 1; end
    disp([A,B,W,T,L])
end
toc
```

The 10 matches result in

A	B	W	T	L
8	5	1	0	0
3	0	2	0	0
0	3	2	0	1
9	6	3	0	1
4	1	4	0	1
2	5	4	0	2
4	1	5	0	2
6	9	5	0	3
3	0	6	0	3
4	1	7	0	3

Checking:

- Each match ends when one player is up 3 games (check)
- If A is up three games, the wins increment by one (check)
- If B is up three games, the losses increments by one (check)

Looks like the program is OK. Now run 100,000 times

```
W = 0;
T = 0;
L = 0;
tic
for i=1:1e5
    A = 0;
    B = 0;
    while( abs(A - B) < 3)
        if(rand < 0.55)
            A = A + 1;
        else
            B = B + 1;
        end
    end
    if(A > B) W = W + 1; end
    if(A == B) T = T + 1; end
    if(A < B) L = L + 1; end
end
toc
disp([W,T,L] / 1e5)
```

```
Elapsed time is 0.076789 seconds.

      W             T      L
    0.6481         0    0.3519
```

A has a 64.81% chance of winning any given match with this format

9) A and B are playing a match. For any given game,

- A has a 55% chance of winning (+1 point for A), and
- A has a 45% chance of losing (+1 point for B).

If the match continues until one player

- Wins at least 4 games, and
- Is up by 2 games

Determine the odds that player A wins the match using a Monte-Carlo simulation with 100,000 matches

This is a bit trickier. One solution is to

- Play until someone wins 4 games (while loop).
- Once that happens, keep playing until someone is up by three games

Checking: Play 10 matches and display everything

```
W = 0;
T = 0;
L = 0;
tic
for i=1:1e1
    A = 0;
    B = 0;
    while( max(A, B) < 4)
        if(rand < 0.55)
            A = A + 1;
        else
            B = B + 1;
        end
    end

    while( abs(A - B) < 3)
        if(rand < 0.55)
            A = A + 1;
        else
            B = B + 1;
        end
    end
    if(A > B) W = W + 1; end
    if(A == B) T = T + 1; end
    if(A < B) L = L + 1; end
    disp([A,B,W,T,L])
end
toc
```

The results of these 10 matches are

- A or B have at least 4 wins (check)
- The winning margin is 3 games (check)

A	B	W	T	L
1	4	0	0	1
7	4	1	0	1
5	2	2	0	1
8	5	3	0	1
11	8	4	0	1
3	6	4	0	2
4	1	5	0	2

```
      1     4     5     0     3
      0     4     5     0     4
      4     1     6     0     4
```

Now that I'm convinced this program works, play 100,000 matches

```
W = 0;
T = 0;
L = 0;
tic
for i=1:1e5
    A = 0;
    B = 0;
    while( max(A, B) < 4)
        if(rand < 0.55)
            A = A + 1;
        else
            B = B + 1;
        end
    end

    while( abs(A - B) < 3)
        if(rand < 0.55)
            A = A + 1;
        else
            B = B + 1;
        end
    end
    if(A > B) W = W + 1; end
    if(A == B) T = T + 1; end
    if(A < B) L = L + 1; end
 %   disp([A,B,W,T,L])
end
toc
disp([W,T,L] / 1e5)
```

```
Elapsed time is 0.090778 seconds.

      W             T       L
    0.6515          0    0.3485
```

With this format, player A has a 65.15% chance of winning any given match

Comments:

- Again, this is a really difficult problem to calculate by hand.
- With Matlab, if you can play one match, you can play 100,000 matches
- The odds are then the number of times A wins divided by the number of matches played
- (approximately)