****

**STEM LAB AS.no.2**

**Student Name: Adeeb Shaban.**

**Student ID: 18110018**

**Assignment Author:**

**Eng.Laith Abu Omar.**

**Eng.Ahmad Azzam.**

**Eng.Tahani Ghobon.**

**Academic Year: 2018/2019**

**Task (I):**

1-

%Ploting the the following function:

%(1/y)x3-(1/y) = x3+x2+x

%firstly we start by putting x and y.

%We take the biggest common factor(1/y):

%(1/y)x3-1 = x3+x2+x

%we devide x3-1.

%Then, it will be:

%(1/y) = x3+x2+x/x3-1

%Then, we flip the equation and it will

%look like this:

%y = x3-1/ x3+x2+x

%Then we write it by the MatLab equation.

%In the MatLab equation its:

%y = (x.^3-1)./(x.^3+x.^2+x);

%And we have the x period from

%-10 to 10 (from the question)

%We separate them along and plot them

solution:

x = -10: 1/100:10; %We put the sign ‘;’ to

%store the value of x.

y = (x.^3-1)./(x.^3+x.^2+x);

%Y according to the

%equations before.

plot(x,y) %plot: is the command %that plots the stored variables

grid on %We can use grid on to %underline the plotted graph, it is optional %to beautify the equation.

%Now we have the plotted equation.

adeeb_task1%20part%201.pdf2-a) and b)

%To draw the required graph

%we start by giving the graph the period from %start to end

%And we multiply the sin and cos with 7 and 5 %because the shape is a big and a small circle %when we multiply them by 7 and 5 it will be %a big and a small oval.

per=-pi:1/10: pi; %period from to

a =7\*sin(per); %to give the big oval a start %and aa end

b =7\*cos(per); %to give the big oval a start %and a end

plot(a, b,'bo'); %plotting the big oval

%b is for blue color

%o to make the shape ‘o’ %instead of line

hold on; %hold will let the graph plots %together

xlabel('X') %to put a label for the x axis

ylabel('Y') %to put a label for the y axis

c = 5\*sin(per);

d = 5\*cos(per);

plot(c,d,'y'); %plotting the small oval

poi1 = [0 2 -2]; %to make the red shaped %intersects these 3 points and give them a %limit from -2 to 2

f= [0 8 8]; %to make oval maintains its shape %and the graph stays between -8 and 20

%polyval and polyfit is mentioned in task 3

p1 = polyfit (poi1, f, 2)

f = polyval(p1, per);

plot(per, f,'\*r'); %plotting the red shaped

g = 5\*sin(per);

plot(per, g,'g'); %plotting the green curve

y5 = 5\*cos(per);

plot(per, y5,'m'); %plotting the pink curve

grid minor %grid minor is to line a lot of lines more than grid on

title('Graph 2') %to put a title for the %second figure

legend ('1','2','3','4') %legend is to put a %label for the symbols such as: \*,etc…



**Task (II):**

%In this task, we will start by defining the %matrixes in a separated matrix

%such as:

%first in a matrix, second in a matrix…etc

first = [10; 14; 5; 7; 4; 10; 20; 13; 15; 14; 12; 13]

second = [12; 4; 7; 8; 20; 18; 14; 15; 10; 10; 3; 6]

third = [13; 14; 12; 10; 11; 11; 11; 9; 9; 8; 5; 5]

total = [25; 32; 24; 25; 35; 39; 45; 37; 34; 32; 20; 24]

%Then we define the max,min,etc…

%in a different variables

1-a = max(first) %max: finds the maximum value of the first column

2-b = min(first) %min: finds the minimum value of the first column

3-c = mode(first) %mode: finds the mode value of the first column

4-d = median(first) %median: finds the median value of the first column

5-e = mean(first) %mean: finds the mean value of the first column

6-f = std(first) %std: finds the standard deviation value of the first column

7-g = var(first) %var: finds the variance value of the first column

1-a1 = max(second) %max: finds the maximum value of the column

2-b1 = min(second) %min: finds the minimum value of the second column

3-c1 = mode(second) %mode: finds the mode value of the second column

4-d1 = median(second) %median: finds the median value of the second column

5-e1 = mean(second) %mean: finds the mean value of the second column

6-f1 = std(second) %std: finds the standard deviation value of the second column

7-g1 = var(second) %var: finds the variance value of the second column

1-a2 = max(third) %max: finds the maximum value of the third column

2-b2 = min(third) %min: finds the minimum value of the third column

3-c2 = mode(third) %mode: finds the mode value of the third column

4-d2 = median(third) %median: finds the median value of the third column

5-e2 = mean(third) %mean: finds the mean value of the third column

6-f2 = std(third) %std: finds the standard deviation value of the third column

7-g2 = var(third) %var: finds the variance value of the third column

1-a3 = max(total) %max: finds the maximum value of the total column

2-b3 = min(total) %min: finds the minimum value of the total column

3-c3 = mode(total) %mode: finds the mode value of the total column

4-d3 = median(total) %median: finds the median value of the total column

5-e3 = mean(total) %mean: finds the mean value of the total column

6-f3 = std(total) %std: finds the standard deviation value of the total column

7-g3 = var(total) %var: finds the variance value of the total column

8-

%in this part, we should define every variable %we did before in a completely different %variables to display them in a single matrix

first1 = [a; b; c; d; e; f; g]

second1 = [a1; b1; c1; d1; e1; f1; g1]

third1 = [a2; b2; c2; d2; e2; f2; g2]

total1 = [a3; b3; c3; d3; e3; f3; g3]

display1 = [first1 second1 third1 total1]

**Task (III):**

1-

%x(v) = A(v-v0)3+B(v-v0)2+C(v-0)+D

%When v is the velocity

%In the polymal expression that reproduces the %same shape of the path in this question,

%We will have the x axis and y axis of a %position are by three degree polynomials.

%We will focas on x(v)and y(v) the %coefficients.

%In this task we start by defining two %matrixes to draw the graph, one for the x %axis and one for the y axis.

%Then we put the each point(x,y) in these %matrixes according to x axis and y axis

%But after defining the points, we put ‘;’ in %the last to store them only.

x =[-3.2000 -3.0000 -2.8000 -2.6000 -2.4000 -2.2000 -2.0000 -1.8000 -1.6000 -1.4000 -1.2000 -1.0000 -0.8000 -0.6000 -0.4000 -0.2000 0.0000 0.2000 0.4000 0.6000 0.8000 1.0000 1.2000 1.4000 1.6000 1.8000 2.0000 2.2000 2.4000 2.6000 2.8000 3.0000 3.2000];

y =[4.0000 0.0000 -2.0000 -2.0000 -1.6000 -0.8000 0.0000 0.4000 0.5000 0.4000 0.2000 0.0000 -0.2000 -0.3000 -0.2000 -0.1000 0.0000 0.1000 0.2000 0.3000 0.2000 0.0000 -0.2000 -0.4000 -0.5000 -0.4000 0.0000 0.6000 1.4000 2.0000 1.8000 0.0000 -4.0000];

%Now, we use polyfit and polyval

%polyfit: is a matlab polynomial function %that fits the polynomial to the data we %stored before.

%polyval: is a matlab polynomial function that %makes the polynomial (evaluation)

poly1 = polyfit(x,y,7) %7 is to find the %coefficients for a 7th degree line in the %curve.

per1 = -3.2:1/100:3.2; %to make a period to %the graph that starts by -3.2 and ends by 3.2 %for the x axis.

poly2 = polyval(poly1,per1);

plot(per1,poly2) %now we plot after putting %the period and the polyval for the curve.

grid on %We can use grid on to %underline the plotted graph, it is optional %to beautify the equation.

%The graph is in page 11.

untitled.pdf

2-

%To verify the answer in the previous part

%we declare the x and y in part a.

%Then we use the polyval and polyfit to fit %this polymial to the data we have and make %the evaluation for ths polynomial

%we declare new variables to put polyval and %polyfit in them

%But we should declare x and y from the part a %first or we can declare them by storing the x %and y values first from part a

Sp=polyfit(x,y,7)

Sp1=-3.2:1/100:3.2;

Sp2=polyval (Sp,Sp1);

plot (Sp1,Sp2) %plotting the stored in %variables

grid on %to line the graph

%The graph will show as its shown in part a.

**Task (IV):**

oct=(2/32:2/16:2)\*2\*pi; %to make the shape %octagonal with 8 faces and to have all the %sides have meters from the centre

%2/32 is 1/16 and 2/16 is 1/8 so it will have %the 8 faces

%then we use the sin and cos to have the small %8 octagonals

%I tried these number to add them to the small %8 octagonals because 3 of them will have the %positive x axis and the other three will have %the negative x axis

%but one of these will have the positive y %axis and one in the (x,0) and one will have %the negative y axis

%I tried to add and subtract these numbers to %have the shape as requested as possible.

i1=cos(oct);

ii1=sin(oct)+2.749;

i2=cos(oct);

ii2=sin(oct)-2.749;

i3=cos(oct)-1.89;

ii3=sin(oct)+2;

i4=cos(oct)-2.749;

ii4=sin(oct);

i5=cos(oct)+1.899;

ii5=sin(oct)-2;

i6=cos(oct)+1.899;

ii6=sin(oct)+2;

i7=cos(oct)+2.749;

ii7=sin(oct);

i8=cos(oct)-1.899;

ii8=sin(oct)-2;

if1=4\*cos(oct);

if2=4\*sin(oct);

%Now, we plot these variables to have the %shape of the octagonal and the small %ogtagonals in it.

plot(if1,if2,i1,ii1,i2,ii2,i3,ii3,i4,ii4,i5,ii5,i6,ii6,i7,ii7,i8,ii8)

%now we use define to draw the final squared %shape in the center and the command %‘position’ will give the squared shape a %limitation to be a rectangle and the -1 -1 2 %2 is for -1 for x axis and -1 for y axis and %2 for x axis and 2 for y axis.

centre = rectangle('Position',[-1 -1 2 2]')

plot (centre) %Now, we plot the centre rectangletask%204%202.pdf

**Task (V):**

%Matrix A is only one column and one row

%so we should have 8x8 matrix as requested in %the question so we start by defining 4 %matrixes each one of them have 4x4 matrix

%but we can use for loop to make the matrix %from 1x1 to 4x4, after that, we will have 4 %matrixes each one of them is 4x4 we add them %together to have the 8x8 matrix.

%for loop will start reading from 1 to 4

%we used 2 for loops for each matrixes of %these 4 matrixes to read one for the rows and %one for the columns

%We start now to put expressions in the for %loop that counts the columns

%I used different expressions to match the %numbers that is requested in the question

%we use end when the for loop has done the 4th %loop then, it will exit the loop

Matx1=1;

for row = 1:4;

for column = 1:4

Matx1 (row,column)= 2^(column+4\*(row-1))

end

end

Matx2=1;

for row = 1:4;

for column = 1:4

Matx2(row,column)= sin((column\*pi/8)+(row-1)\*pi/2)

end

end

Matx3=1;

for row = 1:4;

for column = 1:4

Matx3(row,column)= 2\*sin((column-(column+1)/2)\*pi-pi/2)+4\*row

end

end

Matx4=1;

for row = 1:4;

for column = 1:4

Matx4(row,column)= (-3\*(column-2)-12\*(row-1))

end

end

format short g %Its for formatting the float %value with 5 degits to display them with %formatted specific value with 4 decimal %places

EX= [Matx1 Matx2; Matx3 Matx4] %displaying the 4 matrixes %with a single matrix 8x8

EX(2,8) = 0 %replacing the second column and %the 8th row with 0

EX(4,8) = 0 %replacing the 4th columns and the %8th row with 0

**Task (VI):**

%We start the command input to scan what the %player will choose between these three %choices

%We use if statement to give conditions the %answer is true or faulse

%We use strcmpi to compare numbers to strings

%string is more than one character

%’s’: string

%the == is a question its like “does this %value equals this value?’

%else if is for when the if condition is false %it will check the else if expression, if its %false also

%we can use how much else if(conditions) as %much as we want

%if we want the condition to apply in case it %didn’t apply the if and else if we use else %its used for if none of the conditions are %true apply this expression

%The expressions could be to print something %or to store some values or to use another %loop, etc...

%disp is to print a word or sentence on the %output screen

%ceil is to round the element to the nearest %integer greater than of equal is element.

%Rad is a random choice for the string or %integer.

student=input('Rock?Paper?Scissors?','s');

if strcmpi(student,'rock')

student\_score=1;

elseif strcmpi(student,'paper')

student\_score=2;

elseif strcmpi(student,'scissors')

student\_score=3;

else

student\_score=0;

end

robot\_score=ceil(rand\*3);

if robot\_score==1

comupter='rock';

elseif robot\_score==2;

computer='paper';

else

computer='scissor';

end

if student\_score>robot\_score;

disp('student wins');

elseif robot\_score>student\_score;

disp('computer wins');

else

disp('draw');

student\_score = 0;

robot\_score = 0;

end