

ENGR 0012 – Engineering Problem Solving

Goals for this week:

- Review!

Please submit your HW!

Please submit through both new and old
submission systems

Written midterm is Tuesday, February 26, 7pm

- You will receive an email with the location
- If you have a class at that time, please let me know today!

Computer exam is Thursday, February 28

- You will receive an email telling you the time and computer you have been assigned

Reminders for the computer exam:

- Make sure you save your file(s)!!
- Make sure you know how to create a zipped folder
- Make sure you know how to drop files in the right place

Thank you for your feedback!

- Polyfit and polyval review
- Semilog and loglog

Polyfit/polyval example for a polynomial

- Use “ReviewingPolyfit.txt”

```

clear
clc

%Get data
filename=input('Which file?','s');
data=load(filename);

%Get x and y
x=data(1,:);
y=data(2,:);

%Find coefficients
Coeffs=polyfit(x,y,2)

% Polyfit gives you the equation (the coefficients) for the line
% y = -0.0082x2 + 0.7459x + 13.4722

% Polyval gives you the y value for the x that you provide (try x=3)
yPolyval=polyval(Coeffs,3)

% Polyval is the same thing as plugging in that x=3 value into the equation
% y = -0.0082(3)^2 + 0.7459(3) + 13.4722
yEquation=(-0.0082*(3^2)) + (0.7459*3) + 13.4722
%These coefficients are the result of using polyfit

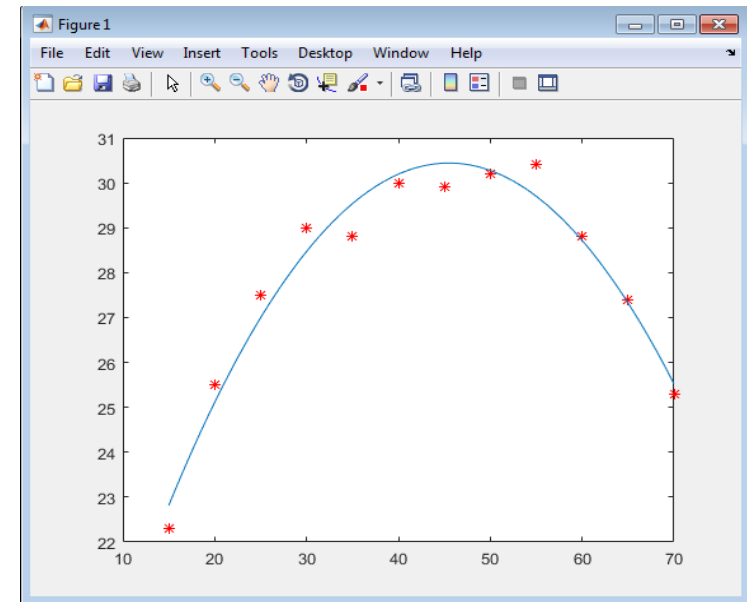
%To plot polynomials, we want to get a smooth line, so we create 200 x points
xnew=linspace(min(x), max(x), 200);

% We have 200 x points, now we need the 200 y points we get
% when we plug in those 200 x points into the equation
yfitpoints=polyval(Coeffs, xnew);

% Now we plot the 200 x points and the 200 y points
plot(x,y,'r*',xnew,yfitpoints);

%What if you want get the y value for an x provided by the user?
x_user=input('Please provide a value for x: ');
y_userPolyval=polyval(Coeffs, x_user)
%Which is the same thing as:
y_userEquation=(-0.0082*(5^2)) + (0.7459*5) + 13.4722

```



(assuming x_user=5)

Semilog example

```
SplineExample.m  PolySplineProblem.m  SemiLogLogPro
1 - clear
2 - clc
3
4 % define variables
5 - x=[1 2 3 4 5 6 7];
6 - y=[1.1 3.0 8.0 21.8 59.4 161.4 438.6];
7
8 %change y to ln(y) for fitting
9 - ln_y=log(y)
10
11 %plot ln_y versus x
12 - plot(x,ln_y,'x*')
13
14 %fit straight line through x and ln_y
15 - coeff=polyfit(x,ln_y,1)
```


Loglog example

```
SplineExample.m  PolySplineProblem.m  SemiLogLogProblem.m
1 - clear
2 - clc
3
4 % define variables
5 - x=[1 2 3 4 5 6 7];
6 - y=[2 9.2 22.4 42.2 69.0 103.0 144.6];
7
8 %change y to ln(y) and x to ln(x) for fitting
9 - ln_y=log(y);
10 - ln_x=log(x);
11
12 %plot ln_y versus ln_x
13 - plot(ln_x,ln_y,'r*')
14
15 %fit straight line through ln_x and ln_y
16 - coeff=polyfit(ln_x,ln_y,1)
```

Review Problems

Review 1

- Create a program that calls the following functions:
- Function **Header**: Displays your group number and student names
- Function **ReviewLoad**: Asks user for file name, check if it exists, and load the data. Should return data to the main
- From the main: Ask user to provide a number. It must be greater than 0 and should not exceed the number of rows in the data matrix
- Function **GetXandY**: Assigns the first row of the data to be x. Takes the user-entered number (from the main) to get y, where y will be row(# from user) from the data. The function should send x and y to the main
- Function **MyStats**: Calculates the mean of y, calculates the variance and standard deviation of y, and gets the sum of all the even numbers in y, and displays these values from the function. Then, it switches the first element in y with the last element in y, and displays the original and the revised arrays.
- From the main: Ask user to provide a number, then count and display how many times that number appears in the matrix
- Function **MyPlot**: Plots x vs. y, asks user to provide x and y labels and title. Use a menu and switch-case to allow the user to decide if they want to plot red squares or yellow circles
- Try with "FuncData.txt"

Submit zipped folder called "Mena_Time_ReviewTeam#" (Mena_10am_ReviewL01) to Classwork folder

Review 2

1. What is `x=linspace(2,20,5)`?
2. What is `y=[1:0.5:3]`?
3. Find the coefficients of the 2nd degree polynomial for `polynomialData.txt` (use MATLAB).
 - a. By hand, write out the equation.
 - b. Then, in MATLAB, evaluate the equation for `x=[2 4 5]`.
4. Find the coefficients of the line for `semilogData.txt` (use MATLAB).

Review 3

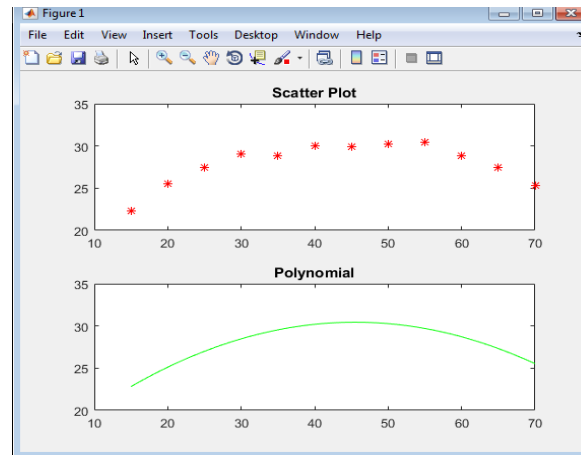
- The main program should provide information about the purpose of this program
- **Get_inputs** should not receive any variables but should return the user-entered values for P, N, I, and A
 - Make sure the user is able to enter the interest value (i) as a percent and have the program divide it by 100 for the calculation
- **Future_withP** should convert to a future value (F_1) given a present single payment (P) $\rightarrow F_1 = P [(1 + i)^n]$
- **Future_withA** should convert to a future value (F_2) given an annuity (A) per interest period $\rightarrow F_2 = A [((1 + i)^n - 1) / i]$
- The main program should display “The calculated value for F_1 is ## and for F_2 is ##”
- *Practice using different variable names within each function!*

Try P=3000, i=2, n=30, A=1200
you get
F_1=5434.08 and F_2=48681.70

Review 4

- Create a data file called PracticeData with a 4x10 data set (any numbers)
- Have the program call the following functions:
- **LoadData**: This function asks the user for the data file, loads the data, and sends the data to the main
- **SelectRows**: This function asks the user to pick two rows (make sure they exist) and from within the function displays “You selected row #RowNum” after each one is selected. The function then sends these variables to the main, and the specified rows will be called x and y.
 - Assume you do not know the data matrix dimensions
- **MinMax**: This function calculates and displays (from within the function) the min and max of x
- **Plotting**: This function will plot the x and y points, and ask the user to provide a title and x and y labels

- Main will display purpose of the program
- **f_header** will display name and group number
- **f_coeff** will ask the user to provide a data file and will find the line for a second order polynomial
- **f_plotting** will use `subplot` to plot the scatter plot and the polynomial as shown in the image (include titles as shown)



Use a data file of your choice (note that images may not match).

- Main will ask user to provide a value for x
- **f_calc** will find the corresponding value of y
- Main will display “For $x=xvalue$, $y=yvalue$ ”

Review 6

```
a=8;
n=13;
disp('Begin');
for k=1:10
    if k==2 || k==4
        a=k*9;
        disp('First condition was true')
    elseif k<6 && k>=4 || k==8
        a=k-7
        disp('Second condition was true')
    elseif k>=7 && k<10
        a=k/n;
        disp('Third condition was true')
    elseif k==5 || k<=2 || k==7
        a=k+3
        disp('Fourth condition was true')
    end
    n=n-1;
end
n
```


Review 7

First find the answers yourself (by hand), then
create a script and check your answers!

```
if a >= b
    answer = 100
elseif c < a & b ~ c
    answer = 200
elseif c < a | (a + b) < c
    answer = 300
elseif a == b | b == c
    answer = 400
elseif (a * b) == 6
    answer = 500
else
    answer = 1
end
```

a)	a = 2	b = 3	c = 4
	answer = _____		
b)	a = -5	b = 10	c = 4
	answer = _____		
c)	a = 5	b = 6	c = 6
	answer = _____		
d)	a = 1	b = 6	c = 7
	answer = _____		

Review 8

First find the answers yourself (by hand), then
create a script and check your answers!

```
1      %Start program
2
3      disp('Hello!');
4      r=1;
5      s=-2;
6
7      %While loop
8      while r<3
9          s=s+r;
10         t=s+1;
11         disp(t);
12         disp('I like loops');
13         r=r+1;
14     end
15
16     disp('Bye!')
17
```

Review 9

```
a=3;  
b=2;  
c=6;  
d=7;  
e=10;  
Fun=[' '];
```

```
for i=1:3  
    Num=findMinMax(a, b, c, d, e);  
  
    switch Num  
        case 18/2  
            NewText=mytext(i,Text);  
            c=11;  
        case 5  
            Text=mytext(i, Fun);  
        case 1  
            disp('Oh!')  
            Text=mytext(i, Fun);  
            c=8;  
            d=9;  
        otherwise  
            disp(NewText)  
    end  
end
```

```
function [Str1]=mytext(j, Fun)  
  
    T1='I like programming';  
    T2='reading is fun!';  
    Str1=Fun;  
  
    if j==1  
        disp(['how', T2(8), T1(1:2)]);  
        Str1=[Str1, T1(3), T2(2:3)];  
    else  
        disp('love');  
        disp(j);  
        Str1=[Str1, T1(j*6), T2(14:15)];  
    end
```

```
function MinMax=findMinMax(e, d, c, b, a)  
  
    x=[1 5 9 2 4 6];  
  
    if (e==2 || b>6) && b<9  
        MinMax=min(x);  
    elseif rem(c,2)==0  
        MinMax=max(x);  
    elseif d+a<15 && mod(e,3)==0  
        MinMax=2*min(x);  
        disp('...')  
    end
```