Name: Andrew Brown	**Lab Time: T 12:00**
People Worked With: Cailin Moore	**Websites Used:**
Time spent on zyBooks (hrs): 1	**Time spent on lab (hrs):3**
Submission Instructions	
Turn all work in to Lab 6 on Gradescope (PDF) and Canvas (.zip file), even if it is not complete yet. If you are not finished, complete the assignment outside of lab and re-submit to Lab 6 on Gradescope and Canvas. All labs are typically due at the same time on Monday every week, but check Canvas if in doubt.	

Learning objectives:

- Creating, and evaluating, a polynomial
- Fitting a polynomial to data
- Fitting a Gaussian to data
- Interpolating data
- Difference between interpolation and fitting
- Analytic versus data integration

New MATLAB commands

These are highlighted in **bold** in the instructions below.

- `polyval(polynomial, t) – Evaluate the polynomial at the t values`
- `polynomial = polyfit(xs, ys) – Fit a polynomial to the ys values`
- `yOut = interp1(xs, ys, xIn) – Interpolate the given xs,ys values at the given xIn values`
- `fit( x,y, fittype ) – Fit a function to the x,y data. fittype is which kind. We'll use 'smoothingspline' for this lab`
- `trapz(x,y) – calculate the integral of the data`
- `integral(f, xmin, xmax) – calculate the integral of the function f from xmin to xmax`

Lab Problems

Files to download to your Lab6 Folder

- *Lab6Data.txt*
-

Getting Started

Anything for experimentation goes here with bullet points

- Read in Lab6Data.txt (dlmread or csvread)
    - Determine what the dimensions of the data are

Problem 1

Read in the file Lab6Data.txt, plot it, then fit a 2nd degree (quadratic) polynomial to the data. Use fplot to plot the fitted polynomial. Extra credit: Try polynomial degrees 2,3, and 4

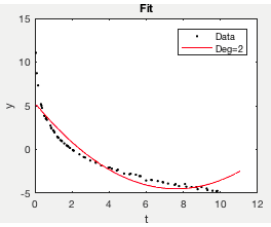
Deliverables:

1. Script that fits and plots
2. Plot with data and fitted function

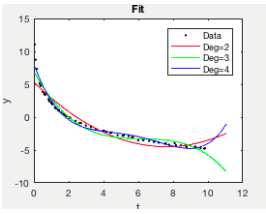
Step by Step Instructions:

- Read in the data and make sure you can get the x values (first column) and the y values (second column).
  - Plot them as dots
- Call polyfit to create a polynomial (which is just an array of polynomial coefficients). Coefficients are shown in the self-check
- Create an anonymous function from polyval and the polynomial you got back from polyfit. This function should take in an x value and return a y value
- Use the anonymous function to plot the fitted polynomial
  - You will need to set the min and max x values to print between – what are the minimum and maximum values in the original data?
- Extra credit: Wrap a for loop around your code. To do the legend, you will need to declare a cell array of strings.

Self-check: `Coefficients for polynomial: 0.1XX8 -2.5XX8 5.3XX0`



Extra credit

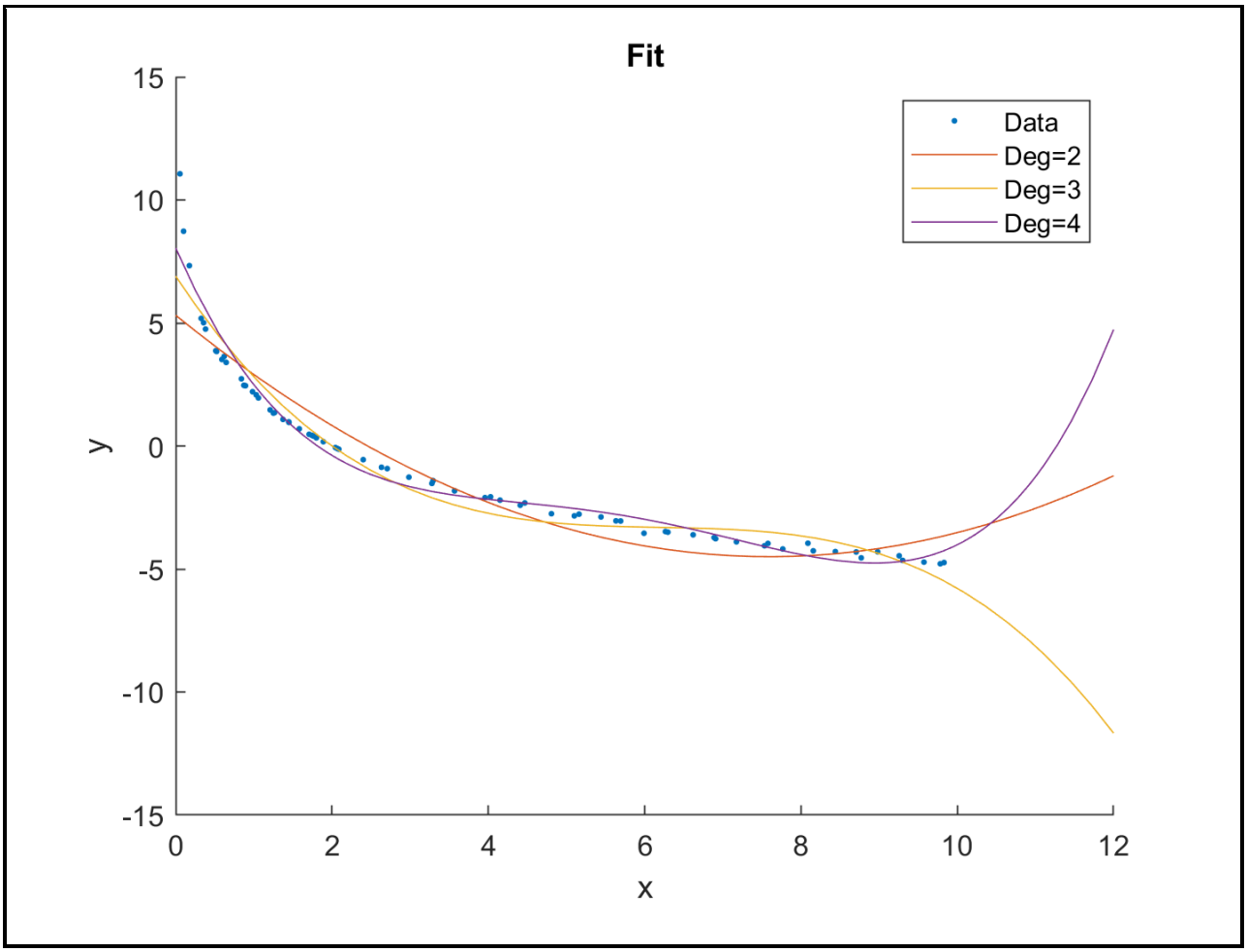


Grading Criteria:
[10 pts] [Read data]
[10 pts] [Plot data]
[20 pts] [Poly fit]
[30 pts] [Anonymous function to plot]
[20 pts] [fplot]
[10 pts] [title, labels and legend]
[+20 pts] [EC: Do for all 3 polynomial degrees]
Answer script here:

```
%Andrew Brown Lab 6 Problem 1
```

```
clc
clear
close all
hold on

%Practice with text files, and fit plotting

Data=dlmread('Lab6Data.txt'); %read the text file
x=Data(1,:); %define x values
y=Data(2,:); %define y values
plot(x,y, '.') %plot the data
for i=2:4
P=polyfit(x,y,i); %Create a polynomial of 2nd order to fit the data to a
function
polyFunc= @(x) polyval(P,x); %anonymous function so I can call polyval
fplot(polyFunc,[0,12]) %plot the fit polynomial
xlabel('x') %label x axis
ylabel('y') %label y axis
title('Fit') %title the plot
end
legend('Data','Deg=2','Deg=3','Deg=4') %give a legend
```

Plot here:


Fit
Data
Deg=2
Deg=3
Deg=4
15
10
5
0
-5
-10
-15
y
0
2
4
6
8
10
12
x

Problem 2

Evenly sample (in x) the data set from problem 1 with 10 samples. Plot the samples on top of the data. Extra credit: Do three plots, with 5, 10, and 15 samples each.

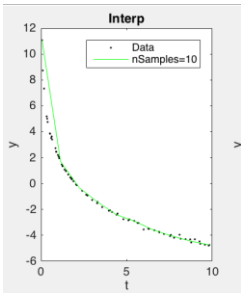
Deliverables:

3. Script for reading in data, doing the interpolation, and plotting
4. The plot
5. Extra credit: Plots: Should use a for loop

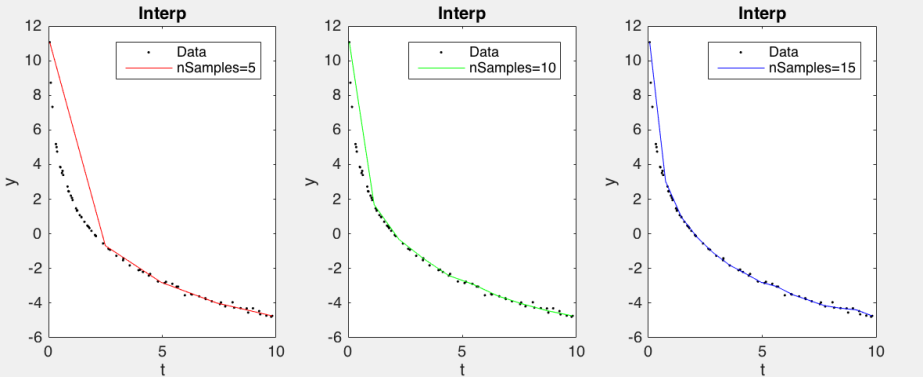
Step by Step Instructions:

- *Start with your script from problem one.*
- *Declare the x values you want to re-sample at*
    - *What x values should you start and stop with? Should be the same as fplot from problem 1*
- *Replace polyfit with interp1, using x values you just created*
- *plot using plot with your x and interpolated values*
- *Extra credit: Again, use a for loop. This time, make 3 subplots instead of plotting in the same figure; it's too hard to see what's happening.*

Self-check: `First few interpolated y values, 10 samples: 11.0630 1.6852 -0.3205`



`Extra credit:`

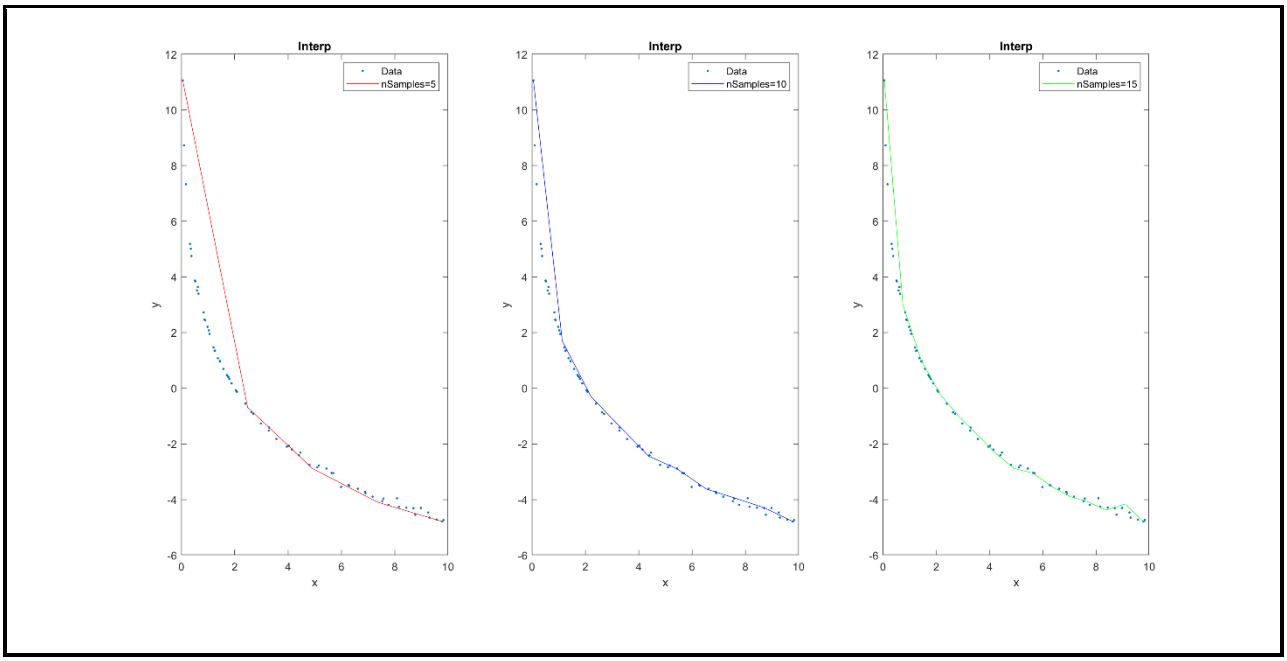


Grading Criteria:
[10 pts] [Read data]
[20 pts] [Find min/max x values]
[10 pts] [Make 10 evenly sampled values]
[30 pts] [Interpolate]
[20 pts] [plot]
[10 pts] [Labels]
[+20 pts] [Extra credit for loop]

Answer script here:

```
%Andrew Brown Lab 6 Problem 2

clc
clear
close all

%Practice with text files,fit plotting, and interpolation

%Read the data from the text file
Data=dlmread('Lab6Data.txt'); %read the text file
x=Data(1,:); %define x values
y=Data(2,:); %define y values

%Plot the 3 different interpolatd graphs against the data
i=[5,10,15]; %different number of samples
color={'r','b','g'}; %different color for each plot
tag={'nSamples=5','nSamples=10','nSamples=15'}; %loop of legend tags
for a=1:3
    subplot(1,3,a) %create 3 different plots
    plot(x,y, '.') %plot the data
    hold on
    newX=linspace(x(1),x(end),i(a)); %interpolated values of x on the same
interval
    P=polyfit(x,y,2); %Create a polynomial of 2nd order to fit the data to a
function
    polyFunc=interp1(x,y,newX,'spline'); %anonymous function so I can call
polyval
    plot(newX,polyFunc,color{a}) %plot the fit polynomial
    xlabel('x') %label x axis
    ylabel('y') %label y axis
    axis([0,10,-6,12])
    title('Interp') %title the plot
    legend('Data',tag{a}) %add legend
end
```

Plot here:


Interp
Data
nSamples=5
Interp
Data
nSamples=10
Interp
Data
nSamples=15
x
y

Problem 3

Find the area under the curve for the original data set. Compare it to the area under the curve for your fitted polynomial from problem 1. Plot the sorted data to show it is sorted. Extra credit: Compare to the fitted polynomial at increasing degree (2,3,4). Are you converging (getting closer) to the answer from the original data set as you increase the degree?

Deliverables:

6. Script for calculating area under the curve
    a. Must use trapz and integral
7. Command window output of answers
8. Plot of data, sorted, with line rendering style
9. [Extra credit:] Answer to question, degree 3 & 4 polynomials

---

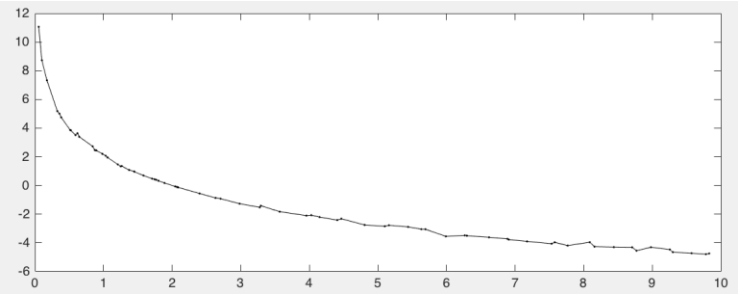
Step by Step Instructions:

- Start with your answer to problem 1.
- Calculate the area under the polynomial by using integral and the anonymous function you made for the fplot. The bounds for integral (xmin, xmax) should be the same as for fplot
- If you plot the data points with a line, you'll notice that they "double back" on themselves. If you just call trapz on the raw data (go ahead and do it) you'll get a number that's wrong. To fix this, you need to sort the data by the x values first
    - The tricky part of this is you need to sort the y values by the x values – if you call sort(data) you'll get every **column** of data sorted (try it and see). If you do sort(data, 1) to sort by rows, you'll sort x and y independently. What you really want is what happens in Excel when you select a bunch of data and sort ALL of it by a specific column
- Sort by x values
    - Sort the first row of data. Just like min, sort will return both the sorted numbers AND the order they ended up in
        - [sortedValues, sortedIndices] = sort(data(1,:))
    - You can use the sortedIndices variable to sort the y values
        - sortedYs = data(2, sortedIndices)
- Once the data is sorted, you can call trapz. Plot the data with a line to verify that you have it sorted correctly
- Extra credit: Same as before – just add a for loop

Self-check: Note: Degree 3 and 4 are extra credit

```
Area original data -1X.X33905
Area poly deg 2 fitted -1X.X23052
```

```
Area poly deg 3 fitted -1X.X57785
Area poly deg 4 fitted -1X.X00998
```



Grading Criteria:
[20 pts] [Call sort to sort x values]
[30 pts] [Use sorted indices to sort y values]
[20 pts] [Call trapz on sorted data]
[20 pts] [Call integrate with fitted polynomial function]
[10 pts] [Correct answer]
[+15 pts] [Extra credit: for loop]
[+10 pts] [Answer question]

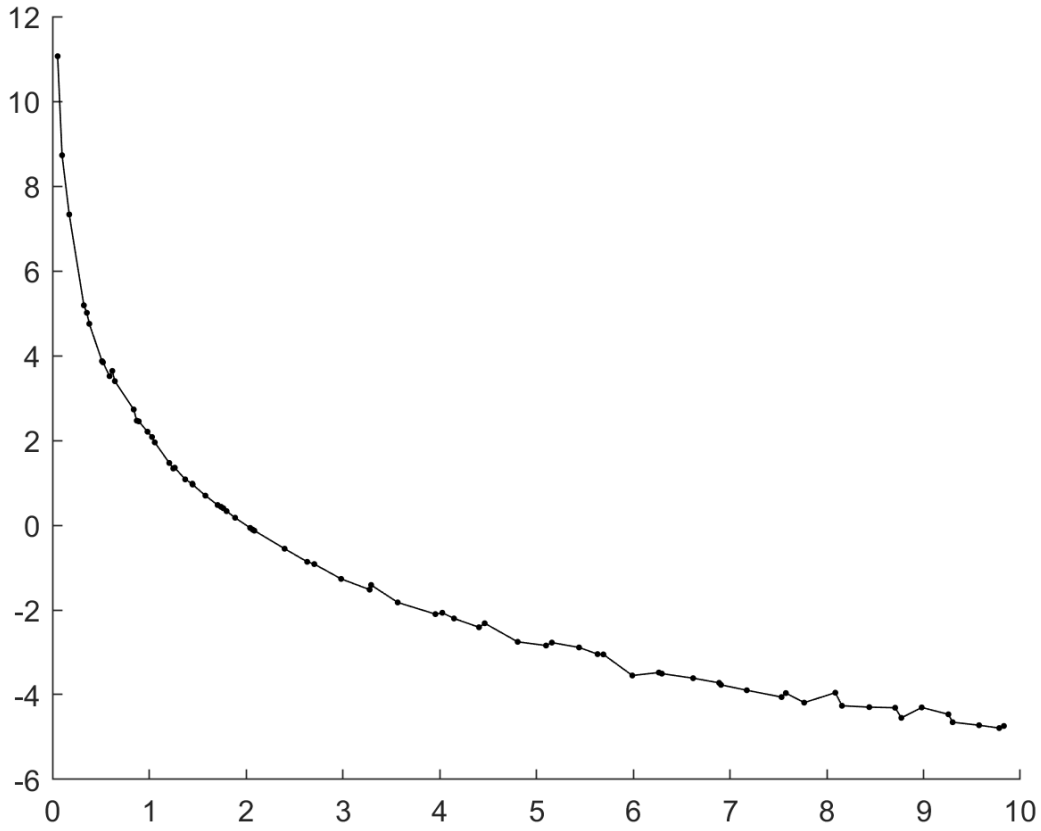
Answer script here:

```
%Andrew Brown Lab 6 Problem 3

clc
clear
close all
hold on

%Practice with text files,interpolation, fit plotting, and trapz

Data=dlmread('Lab6Data.txt'); %read the text file
[x, sortedIndices]=sort(Data(1,:)); %define x values
y=Data(2,sortedIndices); %define y values
plot(x,y, '.k') %plot the data
dataArea=trapz(x,y); %calculate the area under the original data
plot(x,y,'k') %plot the data line
fprintf('Area Original Data: %0.6f\n',dataArea) %print out the original data
integral

%plot the fit lines and calculate the area under the fit lines
for i=2:4
    P=polyfit(x,y,i); %Create a polynomial of 2nd order to fit the data to a
function
    polyFunc= @(x) polyval(P,x); %anonymous function so I can call polyval
    polyArea=integral(polyFunc,x(1),x(end)); %calculate the area under the
caluclated fit lines
    fprintf('Area poly deg %0.0f fitted: %0.6f\n',i,polyArea) %print out the
degrees and areas
end
```

Extra credit question answer here:

```
Question: Converging? Yes, this means the line of best fit is a higher degree
```

Plot here:



Command window output

```
Area Original Data: -18.333905
Area poly deg 2 fitted: -19.023052
Area poly deg 3 fitted: -18.657785
Area poly deg 4 fitted: -18.500998
>>
```

Extra Credit: Problem 1

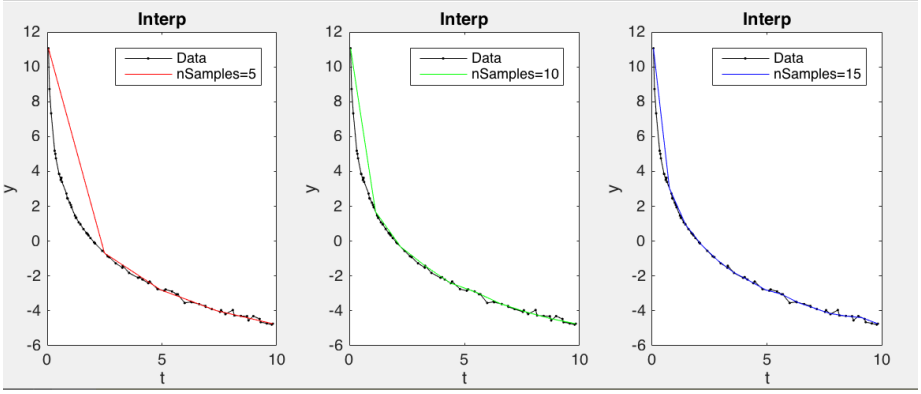
The original equation used to make the data was y(x) = 3 ln(2/x), evaluated from 0.1 to 10.0. Calculate the area under the original curve. Then use the interpolated values you found in problem 2 to calculate the area under the curve.

Step by Step Instructions:

- Start with your script from problem 2
- Copy over the data sort from problem 3 so your data is sorted
- Write the equation given above as an anonymous function
    - Use integral to calculate the area
- Now calculate the area using trapz from your samples instead of the original data.
- For full credit do this for 5, 10, and 15 samples (for loop)

Self-check:

```
Area original data -1X.XX3905, original function -1X.XX1857
nSamples 5 has area -1X.XX5273
nSamples 10 has area -1X.XX9311
nSamples 15 has area -1X.XX0330
```



Grading Criteria:
[15 pts] [Original anonymous function]
[15 pts] [Calculate area under the curve using integral]
[15 pts] [Calculate area under curve from samples]
[15 pts] [Do all 3 in a for loop]
Answer script here:
`% Copy and paste your script here. Make sure your formatting looks the same as MATLAB, with size 10 font.`
Command window output

zyBooks Challenge Exercises

Do the challenge activities for the following in Week 6

1. *Curve fitting*
    a. *Computing wind turbine output power with linear regression*
    b. *Calculating sound pressure level with non-linear regression*
2. *Interpolation*
    a. *Calculating wind turbine output power using interpolation*
3. *Row array resizing*
    a. *Array resizing: Removing elements*
    b. *Concatenating arrays*
    c. *Adding an element: Ticker tape*
4. *Numerical integration*
    a. *Left Riemann numerical integration*
5. *[Optional] Numerical differentiation*
    a. *Health app.*
    b. *Forward difference approximation: Car temperatures*