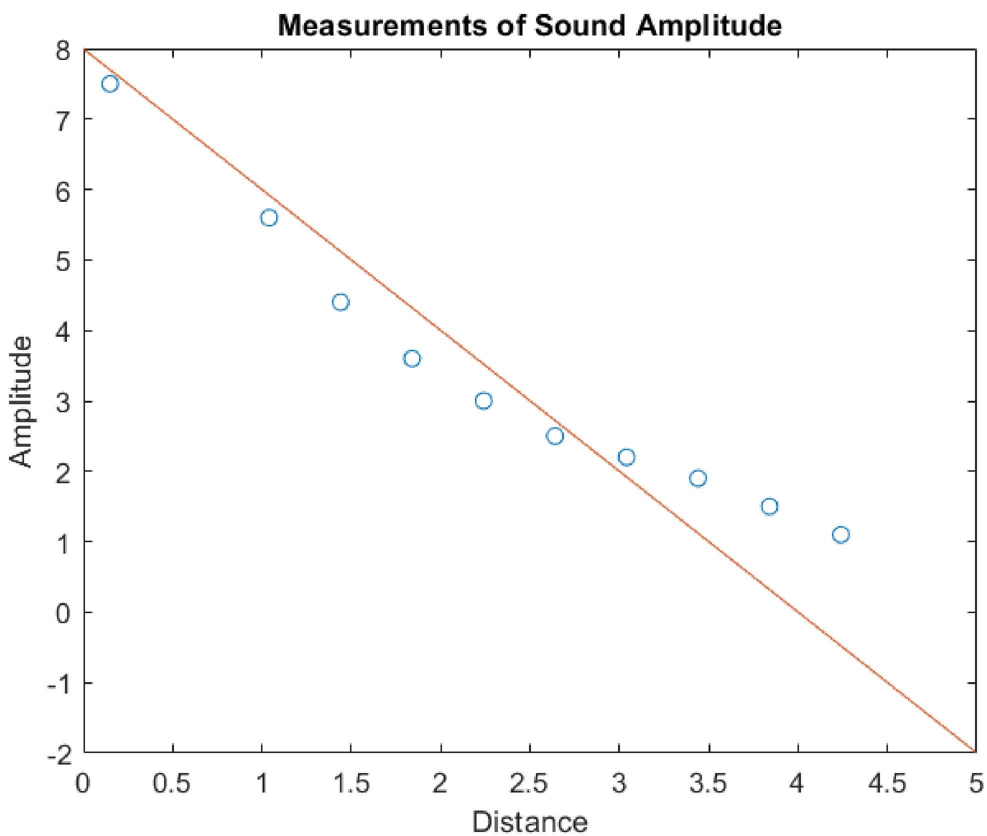
Name: Jingsi Zhou
Date: 11/22/2020
Section: 17460
Assignment: Lab 9

Lab 9

Problem 9.1 On the same set of axes, plot the data in Table 9.1 and the straight line given by Eq. (9.1). ***The table data should be plotted as blue circles.*** Equation (9.1) must be plotted as a solid line. Include your figure and the MATLAB code that generate the figure in your solution sheet.




```
figure(1)
x = [0.15, 1.04, 1.44, 1.84, 2.24, 2.64, 3.04, 3.44, 3.84, 4.24];
y = [7.5, 5.6, 4.4, 3.6, 3.0, 2.5, 2.2, 1.9, 1.5, 1.1];
plot(x,y, 'bo')
hold on
x2 = 0:5;
y2 = -2.*x2 + 8;
plot(x2, y2)
hold off
xlabel('Distance')
ylabel('Amplitude')
title('Measurements of Sound Amplitude')
```

Problem 9.2 In MATLAB, calculate e for the data set and the straight line curve fit given by Eq. (9.1). (You do not have to use a FOR loop. It is much easier to perform vector subtraction, square the result, and then use the SUM command.) Include the value obtained for e and the lines of code used in the calculation in your solution sheet.

```
>> x = [0.15, 1.04, 1.44, 1.84, 2.24, 2.64,...,
3.04, 3.44, 3.84, 4.24];
>> y_approx = -2.*x + 8;
>> y_actual = [7.5, 5.6, 4.4, 3.6, 3.0,...,
2.5, 2.2, 1.9, 1.5, 1.1];
>> y_approx - y_actual

ans =

  Columns 1 through 4

    0.2000    0.3200    0.7200    0.7200

  Columns 5 through 8

    0.5200    0.2200   -0.2800   -0.7800

  Columns 9 through 10

   -1.1800   -1.5800

>> difference = y_approx - y_actual;
>> diffsquared = difference.^2

diffsquared =

  Columns 1 through 4

    0.0400    0.1024    0.5184    0.5184

  Columns 5 through 8

    0.2704    0.0484    0.0784    0.6084

  Columns 9 through 10

    1.3924    2.4964

>> e = sum(diffsquared)
```

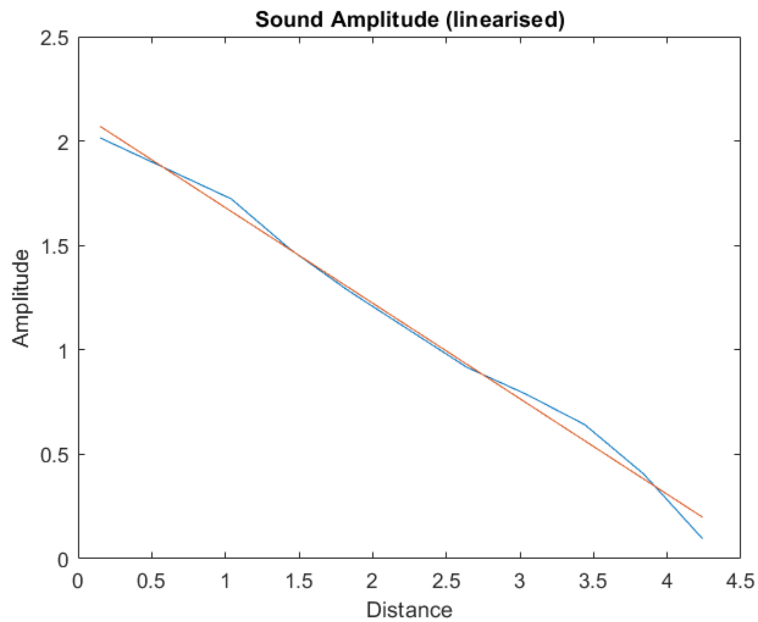
e =

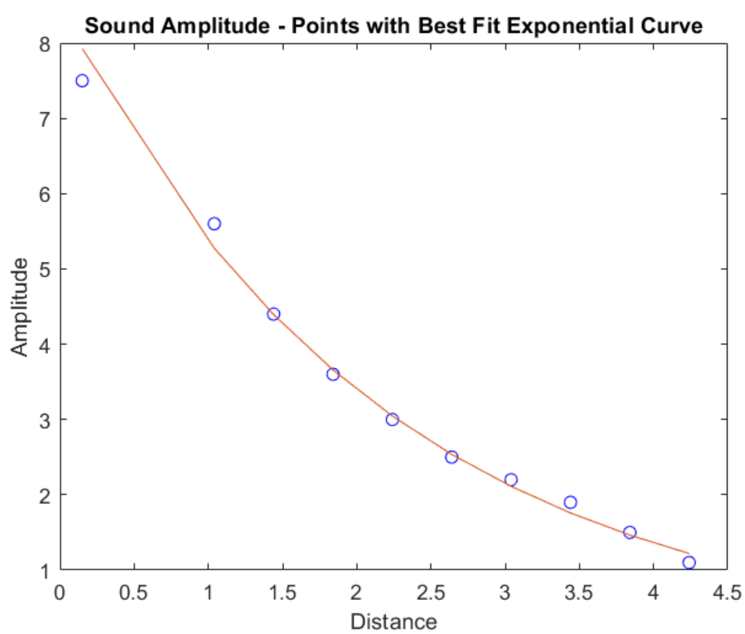
6.0736

Problem 9.3 Use the *polyfit* command with the data in Table 9.1 to find the best fit values of a and b for a straight line. As you did in the previous problem, calculate the total squared error for your best fit line. Include the values you find for a, b, and e and your code in the solution sheet.

```
>> p = polyfit(x, y_actual, 1)

p =

   -1.4982 -> a    6.9123 -> b

>> y_polyfit = polyval(p, x)

y_polyfit =

  Columns 1 through 4

    6.6875    5.3541    4.7548    4.1555

  Columns 5 through 8

    3.5562    2.9569    2.3576    1.7584

  Columns 9 through 10

    1.1591    0.5598

>> polyfitdiff = y_polyfit - y_actual;
>> polysquare = polyfitdiff.^2

polysquare =

  Columns 1 through 4

    0.6601    0.0605    0.1259    0.3086

  Columns 5 through 8

    0.3094    0.2088    0.0249    0.0201

  Columns 9 through 10

    0.1162    0.2918

>> e = sum(polysquare)

e =

    2.1263
```

Problem 9.4 In a single script, perform the following:
(a) Define a variable called yprime that is equal to the natural logarithm of the y data in the table above. Plot the variable yprime as a function of distance x.
(b) Using the polyfit command, find the best fit straight line to the transformed data set (x,yprime). Plot the fitted line on the figure created in (a). Turn in a copy of your plot and the coefficients a' and b' that you obtained from polyfit.
(c) Based on the best fit values of a' and b' determined in the previous problem, calculate the constants a and b that appear in Eq. (9.2).
(d) On a single figure, overlay plots of the original data points (amplitude vs. position) and the best fit exponential function given by Eq. (9.2). Label the axes of your plot and turn it in along with the a and b you found.
Make sure that you include your script in your solution.





```
V =

   -0.4576    2.1388


b =

    0.4576


a =

    8.4890
```

Script:

```
x = [0.15, 1.04, 1.44, 1.84, 2.24, 2.64, 3.04, 3.44, 3.84, 4.24];
y = [7.5, 5.6, 4.4, 3.6, 3.0, 2.5, 2.2, 1.9, 1.5, 1.1];
figure(1)
yprime = log(y);
plot(x, yprime);
hold on
V = polyfit(x, yprime, 1)  %polyfit output vector for linearised
function
y_newfit = polyval(V, x);  %THIS IS ALSO A VECTOR
plot(x, y_newfit);
aprime = V(1);
b = aprime*-1
bprime = V(2);
a = exp(bprime)
xlabel('Distance');
ylabel('Amplitude');
title('Sound Amplitude (linearised)');
hold off

figure(2)
plot(x, y, 'bo');
hold on
yexp = a.*exp(-b.*x);
plot(x, yexp);
xlabel('Distance');
ylabel('Amplitude');
title('Sound Amplitude - Points with Best Fit Exponential Curve');
hold off
```

Problem 9.5 Predict the sound amplitude at x = 4.

```
>> y_4 = a*exp(-b*4)

y_4 =

    1.3611
```

Problem 9.6 Explain at least 3 issues that you can think of for using curvefit. ***Hint:*** Think in terms of how rich your data is, how closely you want to fit and the toll that a very complex fit would take on the computer.

Using curve fit will run into complications as the data becomes increasingly related by more variables. That is, a function that depends on multiple variables will require more computational power from the computer. Another issue that comes with using curve fit is human sampling errors, such as lack of good data. If there are not enough samples that can allow a relationship to be computed, then the resulting model function/equation will not match the behavior of the observed system. Another issue that the usage of curve fit brings is that if there are too many samples, there could be an infinite number of ways to construct a model equation.