M20 - Introduction to Computer Programming with MATLAB
Instructor: Prof. Enrique López Droguett, Ph.D.
Teacher Assistants: M. Fidansoy, G. San Martín, M. Pishahang, V. Vela.
Fall 2023 – UCLA
Student: *Alex Lie*
UCLA ID: *905901892*

HOMEWORK 7

Task 1: General Linear Least Squares

Introduction

I am using MATLAB to create a linear model to explain the amount of diseases-carrying organisms over time. Using the data and the equation given to me, I will create a general linear least-squares model by using matrices. I will find the values of X, Y, and Z in the equation that will make it fit the data the best. I will then report the original equation and the linearized equation using the coefficients I found.

Model and Theory

1) $P(t) = Xe^{(-1.5t)} + Ye^{(-0.3t)} + Ze^{(-0.05t)}$
2) $[Z]^T[Z][A] = [Z]^T[Y]$

Methodology

I first cleared the workspace and command window. I then inputted the time and organism amount data in separate column vectors. After, I defined the basis functions matrix by evaluating the different basis functions at each time in t, and then inputting those values into a matrix. I found the basis functions from equation 1. The length of the time column vector is the amount of rows the basis functions matrix has, and the amount of basis functions there are are the amount of columns the basis functions matrix has. I then solved for the coefficient matrix A by solving the normal equation (2). I then outputted each value in the coefficient matrix A. Using these coefficients, I outputted the original equation using fprintf(). Lastly, I added comments to make my code easier to understand for other users.

Calculations and results

3) 4)

```
X =

    4.1375


Y =

    2.8959


Z =

    1.5349
```

```
Equation:
P(t) = 4.1375 e^(-1.5t) + 2.8959 e^(-0.3t) + 1.5349 e^(-0.05t)
```

X, Y, and Z are the coefficients in from equation 1. The original and linearized equations are shown above as well.

Discussions and Conclusions

The coefficients and the equation are displayed in figures 3 and 4. As time increases, the amount of organisms seems to decrease. When the time equals zero, we can estimate that there were initially 8.5683 organisms.

Task 2: Polynomial Regression

Introduction

I am using MATLAB to find the coefficients of a third-order regression model with zero intercept that best fits the data. The data explains how the force changes when the velocity of the water flow from a pump changes. I will find the coefficients that provide the least squares. I will then plot my linear least squares regression model with the data points to see the graph of the third-order polynomial regression.

Model and Theory

1) $y = a * x + b * x^2 + c * x^3 + d$
2) $[Z]^T[Z][A] = [Z]^T[Y]$

Methodology

I first cleared the workspace and command window. I then inputted the velocity and force data in separate column vectors. Afterwards, I defined the basis functions matrix by evaluating the different basis functions at each velocity in v, and then inputting those values into a matrix. I found the basis functions from equation 1. The length of the velocity column vector is the amount of rows the basis functions matrix has, and the amount of basis functions there are are the amount of columns the basis functions matrix has. There will be only three basis functions (x, x-squared, and x-cubed) because the intercept of the regression must be zero. I then solved for the coefficient matrix A by solving the normal equation (2). I then outputted each value in the coefficient matrix A. I then created a range of velocity values and force values using the coefficients and equation 1. I then plotted the data points and the regression line. I also added labels, a title, and axis limits. Lastly, I added comments to make my code easier to understand for other users.

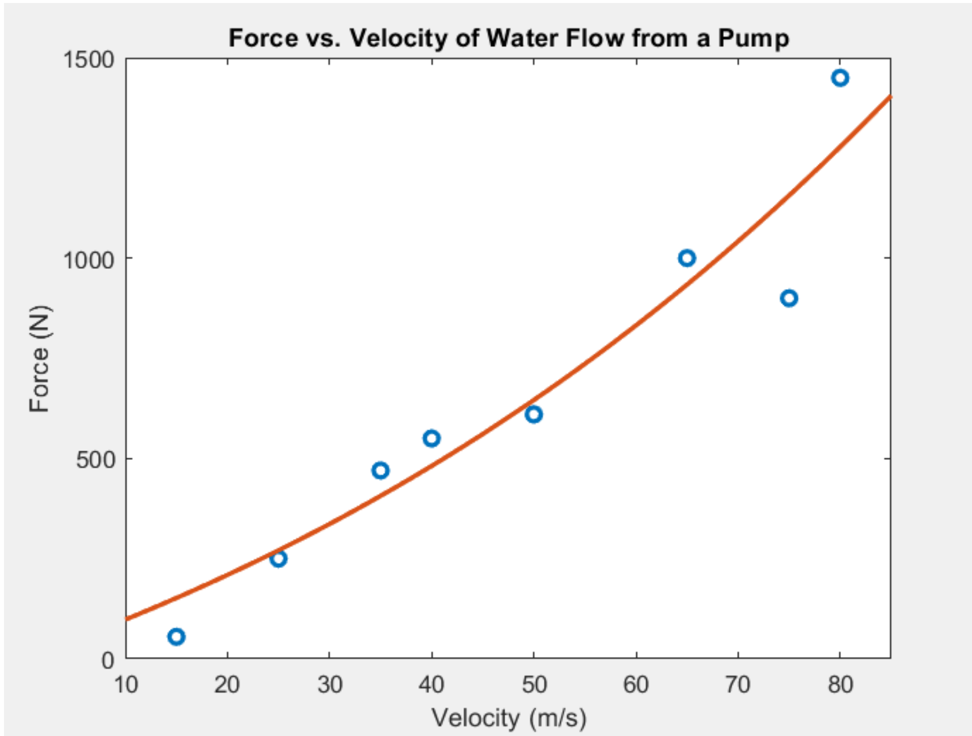
Calculations and results

See next page.

3)

```
a =

    9.1874


b =

    0.0577


c =

   3.3921e-04


d =

     0
```

4)



a, b, c, and d are the coefficients from equation 1. The graph shows the data points plotted and the equation graphed.

<u>Discussions and Conclusions</u>

The general linear regression seems to be a good fit because there are 4 points above the regression line and 4 points below the regression line (4). The value of d is zero because the problem statement asked us to find the polynomial regression model with zero intercept (3). The equation will have a zero intercept only when d equals zero. We might find an even better equation to fit the data points if we are able to make the y-intercept any point we want. To find the ideal y-intercept and find a regression line with the least squares, I would add another basis function of 1 to the basis functions matrix and resolve the normal equation. As the velocity of the water flow increases, the flow increases.

<u>Task 3: Linearization</u>

<u>Introduction</u>

I am using MATLAB to create a function that will solve linear regression problems. I will then use my function to estimate the coefficients in the nonlinear equation given to me using the data points. I will then linearize my model.

M20 - Introduction to Computer Programming with MATLAB
Instructor: Prof. Enrique López Droguett, Ph.D.
Teacher Assistants: M. Fidansoy, G. San Martín, M. Pishahang, V. Vela.
Fall 2023 – UCLA
Student: *Alex Lie*
UCLA ID: *905901892*

<u>Model and Theory</u>

1) $y = (a/(20 * b)) * e^{b*x}$
2) $[Z]^T[Z][A] = [Z]^T[Y]$

<u>Methodology</u>

I first cleared the workspace and command window. I put the x and y data in separate column vectors. I then linearized the equation (1) by hand by taking the natural log on both sides of the equation. I then transformed the data by taking the natural log of the y-values column vector in order to linearize the data. I then declared the basis functions row vector with the functions 1 and x (from the linearized version) using function handles.

I then wrote the generalLinearRegression() function. This function takes in a column vector of the independent variable, a column vector of the dependent variable, and the row vector of the basis function using function handles. I set up the basis functions matrix Z with a width of the size of the number of basis functions, and a height of the size of the number of dependent variable values. I used a nested for loop to evaluate the different basis functions at each x value and set it to the corresponding index in the matrix. Then the coefficient matrix A is solved by using the normal equation (2), and then it is outputted.

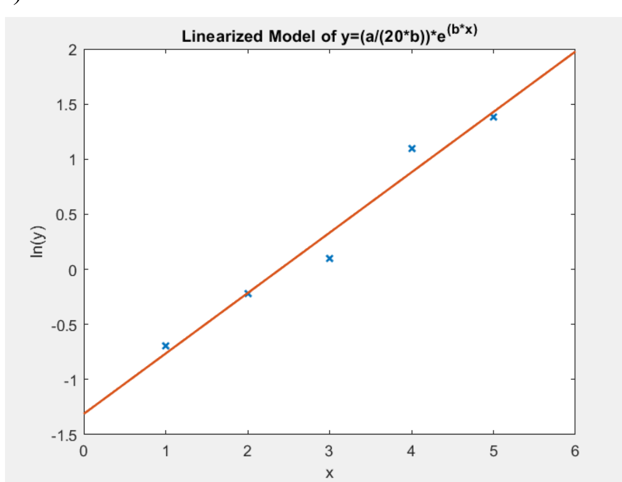
I ran the generalLinearRegression() function, using the x-values, the natural log of the y-values, and the basis functions row vector I declared as parameters. The coefficients in matrix A are not the values of a and b in the original equation because I used the linearized data to find coefficients. In order to find the value for a and b, we have to solve for a and b by hand using the linearized equation. I solved for a and b, and then outputted the values. I then outputted the linearized equation using fprintf(). I then plotted the linearized data points (the original x-values and the natural log of the y-values) and the linearized equation. I also added labels and a title. Lastly, I added comments to make my code easier to understand for other users.

<u>Calculations and results</u>

3)

```
b =

    0.5481


a =

    2.9534


Linear Model:
ln(y) = -1.3114 + 0.5481x
```

4)



The values of a and b are shown above. The linearized equation, plotted data, and graph is also shown.

Student: *Alex Lie*
UCLA ID: *905901892*

Discussions and Conclusions

I estimated a and b by hand, as shown in figure 3. To linearize the model, you have to take the natural log of the y data points. By looking at the graph in figure 4, as x increases, the natural log of y increases. The data points also seem to be following a linear trend, confirming that our linearization technique is correct.