MATH 284 Mathematical Programming
Final Project: Due Wednesday May 16 by 5:00pm

Project Problem Statement:

In class, we developed code that fit the data `speedData.csv` to a linear model of the form $s = at + b$. This was done by constructing an error function which depends on the parameters a and b which could then be minimized to determine the "best fit." By either writing your own code from scratch, or modifying the code introduced in class, use MATLAB to fit the data in `gData.csv` and `sData.csv`. Note that for the data `gData.csv` and `sData.csv` a linear model is not appropriate. Part of your task is to determine an appropriate functional form besides linear that will make a good model for the data. **Hint:** Start by trying some common function types like polynomials, exponentials, logarithms, or trigonometric functions. You will not be able to use the same functional form for both `gData.csv` and `sData.csv` since these data when plotted look very different from one another. Determine a small set of parameters (2-3) that can be adjusted to fit a function to the data by minimizing the resulting error in way that is analogous to what we did in class with the linear function.

You should have two sets of code, one for `gData.csv` and another for `sData.csv`.

Your code should do at least the following steps:

1. Import the appropriate data file as a matrix.
2. Plot the data such that the first column of the data matrix corresponds to the horizontal axis and the second column to the vertical.
3. Once you have chosen a particular form for your model function, overlay several example graphs onto the data plot. Make sure to indicate the values of the parameters you have chosen.
4. Use an error function to compute the error resulting from your guesses for the parameters.
5. Minimize your error function using `fminsearch` to find the "best fit" model that corresponds to the data.
6. Overlay a graph of the best fit curve on top of the data plot. Indicate the parameters that correspond to the best fit model.
7. Make sure all plots look nice and are appropriately labeled.
8. Make sure that your code is carefully commented so that another user can easily determine the function of each command.

The file `speedDataAnalysis.m` is a good template to follow. Note that it relies on the other functions `linearSpeed.m` and `speedError`. You must either have your data files saved in the same directory as your code of provide the full pathname where the data is saved.