OBJECTIVES

The main purpose of this in-class exercise is to use MATLAB to fit cubic splines, fit linear functions to exponential and power functions, and filtering data. By the end of this exercise you should be able to obtain the piecewise polynomial form of a cubic spline and evaluate it, find coefficients of a linear function by fitting it to exponential/power data that has been properly converted, and process data by filtering it.

NOTES

MATLAB COMMANDS

Throughout this document, the MATLAB commands that you must use in your scripts will appear in **`bold`** face.

COMMENTING

Comment your codes extensively. You can use the percent symbol `%` to enter comments in your scripts. Comments allow the user to understand and follow code easily; it is therefore highly recommended to develop a habit to extensively provide commentary in your codes.

A nice feature you may want to use in your scripts is code sectioning. Code sections allow you to organize, add comments, and execute portions of your code. Code sections begin with double percent signs (`%%`), e.g.,

```
%% Vector Operations
% You can perform a number of binary operations on vectors.
%%
A = 1:3;
B = 4:6;

%% Dot Product
% A dot product of two vectors yields a scalar.
% MATLAB has a simple command for dot products.
s = dot(A,B);

%% Cross Product
% A cross product of two vectors yields a third
% vector perpendicular to both original vectors.
% Again, MATLAB has a simple command for cross products.
v = cross(A,B);
```

CODE COPY-AND-PASTE

If you decide to copy and paste example command(s) presented in lecture slides and in-class exercises in MATLAB, be wary of single quotation marks—you may need to delete and re-enter single quotation marks after pasting the command(s) in MATLAB.

EXERCISES

EXERCISE 9: CURVE FITTING - EXPONENTIAL FUNCTION

The purpose of this exercise is to write a script to fit data pairs that are derived from an exponential function of the form: $y = be^{ax}$. Given the dataset (**expData.dat**), containing independent (first row) and dependent (second row) samples, find the coefficients (a_1, i.e., slope, and a_0, i.e., intercept) of a polynomial $f(x) = a_1x + a_0$ (degree=1) that best fits the data.

Create a script and name it `matlabExercise9.m`. In your script,

- Load **expData.dat**
- Extract independent and dependent values in variables `x` and `y`, respectively
- Plot `y` versus `x` using a black square (**`'sk'`**) marker. Remember to provide appropriate figure labels
- Transform `y` to `log(y)` using MATLAB's **`log`** command
- Plot (on a new figure) `log(y)` versus `x` using a black square (**`'sk'`**) marker. Remember to provide appropriate figure labels
- Find coefficients (a_1, the slope, and a_0, the intercept) by calling MATLAB's **`polyfit`** command (remember that the order, or polynomial degree, is 1)
- Evaluate $f(x)$, i.e., the line equation, at the same `x` values by calling MATLAB's **`polyval`** command (save the output of **`polyval`** in a variable called `yHat`)
- Plot the best fit line, i.e., `yHat`, on the same graph where you plotted `log(y)` versus `x`. Use a red color to depict the line. Label the figure appropriately
- Using MATLAB's **`gtext`** command, add the equation of the line to the graph. In addition, add the equation of the derived exponential function

EXERCISE 10: CURVE FITTING - POWER FUNCTION

The purpose of this exercise is to write a script to fit data pairs that are derived from a power function of the form: $y = bx^a$. Given the dataset (**powData.dat**), containing independent (first row) and dependent (second row) samples, find the coefficients (a_1, i.e., slope, and a_0, i.e., intercept) of a polynomial $f(x) = a_1x + a_0$ (degree=1) that best fits the data.

Create a script and name it `matlabExercise10.m`. In your script,

- Load **powData.dat**
- Extract independent and dependent values in variables `x` and `y`, respectively
- Plot `y` versus `x` using a black square (**`'sk'`**) marker. Remember to provide appropriate figure labels
- Transform `x` to `log(x)` and `y` to `log(y)` using MATLAB's **`log`** command

- Plot (on a new figure) `log(y)` versus `log(x)` using a black square (**`'sk'`**) marker. Remember to provide appropriate figure labels
- Find coefficients (a_1, the slope, and a_0, the intercept) by calling MATLAB's **`polyfit`** command (remember that the order, or polynomial degree, is 1)
- Evaluate $f(x)$, i.e., the line equation, at the same `log(x)` values by calling MATLAB's **`polyval`** command (save the output of **`polyval`** in a variable called `yHat`)
- Plot the best fit line, i.e., `yHat`, on the same graph where you plotted `log(y)` versus `log(x)`. Use a red color to depict the line. Label the figure appropriately
- Using MATLAB's **`gtext`** command, add the equation of the line to the graph. In addition, add the equation of the derived power function

EXERCISE 11: CURVE FITTING - CUBIC SPLINE

The purpose of this exercise is to write a script to fit data pairs that do not fall on a straight line. Given the dataset (**splineData.dat**), containing independent (first row) and dependent (second row) samples, fit a cubic spline to the data.

Create a script and name it `matlabExercise11.m`. In your script,

- Load **splineData.dat**
- Extract independent and dependent values in variables `x` and `y`, respectively
- Plot `y` versus `x` using a black square (**`'sk'`**) marker. Remember to provide appropriate figure labels
- Find the piecewise polynomial form of the cubic spline by calling MATLAB's **`spline`** command
- Evaluate $f(x)$, i.e., the polynomial, at new `x` values, spanning between 0.5 and 8.5 with increments of 0.01, by calling MATLAB's **`ppval`** command
- Plot the polynomial fit, i.e., `yHat`, on the same graph where you plotted `y` versus `x`. Use a red color to depict the curve. Label the figure appropriately

EXERCISE 12: FILTERING

The purpose of this exercise is to write a script to filter data. Given the dataset (**filterData.dat**), containing independent (first row) and dependent (second row) samples, remove negative and zero values from the dependent samples.

Create a script and name it `matlabExercise12.m`. In your script,

- Load **filterData.dat**
- Extract independent and dependent values in variables `x` and `y`, respectively
- Write a **`for`**-loop that checks each element in `y` to see if it is greater than zero. If the value is indeed greater than zero, then store that element, as well as its corresponding `x` value, in new variables called `yFilt` and `xFilt`, respectively

EXTRA EXERCISES

Additional exercises are included for extra practice purposes. I encourage everyone to work on them, but first complete the regular exercises—the ones without an X next to their number—then work on the extra exercises.

EXERCISE 8X: CURVE FITTING - CUBIC SPLINE

Modify the script developed in EXERCISE 11 to fit the data (**splineData.dat**) without calculating the piecewise polynomial form of the cubic spline. (*Hint: Additional parameter* `xNew`*; see* ***`spline`*** *command's help documentation for more information*)

EXERCISE 9X: CURVE FITTING - EXPONENTIAL FUNCTION

The data stored in **filterData.dat** comes from an exponential function. Use EXERCISE 12 script and fit a linear line to this data à la EXERCISE 9.