MATLAB Programming NCTU Fall 2022 LAB#3 2022/10/4

For this lab, you will write a script that does function approximation and plot the results. Let the approximation function $y=f(x)$ be a polynomial of degree r, so it has $r+1$ coefficients. You need to find these coefficients using the matrix-division operator (`\`) given a set of sample points $\{(x_i,y_i)\}$.

- Generate the sample points from a polynomial function plus small random numbers. Example:

```
xi = -10:2:10;  yi = -.03*xi.^2 + .1*xi + 2 + 1*(rand(1,length(xi))-.5);
```

- Now our goal is to fit these points to the equation $y=f(x)=a_0+a_1x+...+a_rx^r$. This leads to the following over-specified set of linear equations (n = the number of sample points):

$$\begin{bmatrix} 1 & x_1 & \cdots & x_1^{\,r} \\ 1 & x_2 & \cdots & x_2^{\,r} \\ \cdots & & & \\ 1 & x_n & \cdots & x_n^{\,r} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_r \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

- Use the matrix-division operator (`\`) to solve for the coefficients in minimum-squared-error manner. (Check previous lecture slides for the explanation.) Note: You are NOT allowed to use the `poly*` functions in this lab.
- Plot the function $y=f(x)$ with the estimated coefficients by sampling the function to get the predicted y values for all the x_i.
- Plot the sampled (x_i,y_i) pairs together with the estimated function. You need to use `hold on`.
- In addition, add vertical line segments that connect the sample points and their predicted positions. (Note: You can use a single statement to plot all the vertical segments in one plot. Check the method#1 in the slide about "Multiple 2-D Plots in One Axes".)
- After doing these steps successfully, repeat them using polynomials of different degrees. At least do degree-1 to degree-3 polynomials. Plot them all in the same figure using `subplot`.
- Add titles to the subplots.
- Display the root-mean-square (rms) error within each subplot. Check the documentation to see how to specify the location and alignment when using function `text`.
- Use function `sprintf` to create the strings to display. The usage of `sprintf` is similar to that of `fprintf`, but it returns a vector of type `char`, which you can display as text.

Sample output: