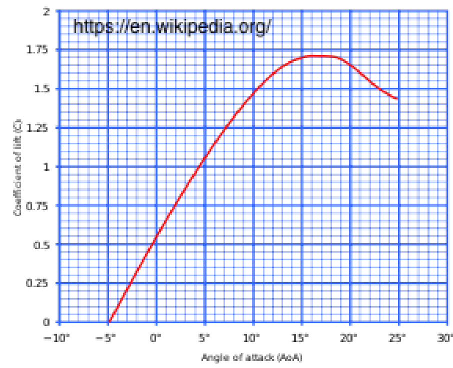
ASEN 1320 Fall 2020
Homework Assignment - Angle of Attack and Lift
Due: 11:59pm on November 6 (Friday)

Problem:



As a wing moves through the air, the wing is inclined to the flight direction at some angle. The angle between a reference line on an air plain and the flight direction is called the angle of attack and has a large effect on the lift generated by a wing. The lift coefficient of a fixed-wing aircraft varies with angle of attack. Increasing angle of attack is associated with increasing lift coefficient up to the maximum lift coefficient, after which lift coefficient decreases as shown in the Figure. Suppose that an airfoil model was placed in a wind tunnel to measure the coefficient of lift, C_l, data for various angles of attack, α. In analyzing experimental data, we often wish to fit a curve or a line to data which may contain experimental error. The method of least-squares allows us to find the "best-fit" line $y = mx + b$ which approximates the data. Given a set of N data values (x_i, y_i), the method of least squares yields the values of m and b which minimize the total error

$$Error = \sum_{i=1}^{N} (y_i - (mx_i + b))^2.$$

The corresponding equations for m (slope) and b (intercept) are given by

$$m = \frac{AB - NC}{A^2 - ND} \qquad \text{and} \qquad b = \frac{AC - BD}{A^2 - ND}$$

where

$$A = \sum_{i=1}^{N} x_i\,, \quad B = \sum_{i=1}^{N} y_i\,, \quad C = \sum_{i=1}^{N} x_i y_i\,, \quad D = \sum_{i=1}^{N} x_i^2\,.$$

Tasks: Write a MATLAB program that consists of a main script and three functions, which calculates the best-fit line ($y = mx + b$) for a given set of experimental data in a piecewise fashion. Specific tasks for each function are described below.

1. **GenerateData.m**
   Write a function named **GenerateData** that for a given input vector of angles of attack, α generates synthetic wind-tunnel experimental data of the coefficient of lift, C_l.

   - Input: A vector of data points.
   - Tasks: Evaluate the polynomial curve by using the MATLAB built-in function `polyval` at data points given in the input vector. The polynomial coefficients are provided in the ASCII file named **Pcoef.dat**. Add random noises that range from -0.5 to 0.5 to this smooth polynomial curve by using the MATLAB built-in function `rand`. Set the random number seed with the current time by using `rng(uint64(now*1000))`.

- Outputs: A row vector of the same length as the input vector filled with noisy experimental data generated from the above task.

2. **LeastSquares.m**
   Write a function named **LeastSquares.m** that computes the slope m and intercept b of the least-square best-fit line for given a set of N data values (x_i, y_i)

   - Inputs: A vector of x data points and a vector of y data points (in this order). These two vectors should be the same length.
   - Tasks: Compute the slope m and intercept b of the least-square best-fit line by using the formula given in the page 1 of this document.
   - Outputs: The computed slope and intercept values (in this order)

3. **PiecewiseLeastFit.m**
   Write a function named **PiecewiseLeastFit** that evaluates a piecewise linear function as defined as

$$y = \begin{cases} m_1 x + b_1 & x < 10 \\ m_2 x + b_2 & 10 \leq x < 15 \\ m_3 x + b_3 & 15 \leq x < 20 \\ m_4 x + b_4 & 20 \leq x \end{cases}$$

   - Inputs: a vector of slope values $[m_1\ m_2\ m_3\ m_4]$, a vector of intercept values $[b_1\ b_2\ b_3\ b_4]$, and a vector of x data points (in this order)
   - Tasks: Evaluate a piecewise linear function using input slope and intercept values.
   - Output: A row vector of the same length as the input x data point vector, filled with fitted piecewise linear function values.

4. Write a main script that calls the **GenerateData**, **LeastSquares**, and **PiecewiseLeastFit** functions.

   - Set up a vector of length 100 with linearly spaced points ranging from -5 to 25, and pass this vector as input to the **GenerateData** function. Assign the output to a row vector named **yData**.
   - Call the **LeastSquares** for 4 times to compute the slopes values $[m_1\ m_2\ m_3\ m_4]$ and intercepts $[b_1\ b_2\ b_3\ b_4]$ that can be used in the **PiecewiseLeastFit** function. Assign the output values of $[m_1\ m_2\ m_3\ m_4]$ to a row vector named **M**, assign the output values $[b_1\ b_2\ b_3\ b_4]$ to a row vector named **B**.
   - Set up a vector of length 1000 with linearly spaced points ranging from -5 to 25, and pass this vector as input along with **M** and **B** to the **PiecewiseLeastFit** function. Assign the output to a row vector named **yFit**.