AE1106: Programming I
# Graded Assignment #2
December 8, 2010

Please read these instructions first:
As an individual assignment, the code and solutions you submit should represent your own work. Discussion with others is not allowed, and neither is the use of online resources other than the standard Matlab help pages (i.e., no Wikipedia, Google searches, chat windows, etc.). During this assignment, the student assistants and the lecturer will only answer questions related to what the question is asking you to do, and will not assist you with how to actually complete the assignment. When finished, you will submit your Matlab source code file(s) to BlackBoard for grading (see below for submission instructions). The deadline to complete the assignment is 12:00 for morning sessions, and 17:00 for afternoon sessions. These deadlines are managed automatically by BlackBoard, so it is important that you submit your documents before the deadline in order to receive credit for the assignment (i.e., manage your time so that you start the submission process at least 15 minutes before the deadline). Those students who have been approved by the student counselors to receive extra time on their exams will receive an extra 30 minutes to complete the assignment.

Questions:

1. (3 points)
A text file **DU300.txt** is provided with the lift coefficient C_l (2nd column) and the drag coefficient C_d (3rd column) as function of angle of attack α (1st column; in degrees) of a typical wind turbine airfoil. Notice that the angles of attack α are given in increasing order.
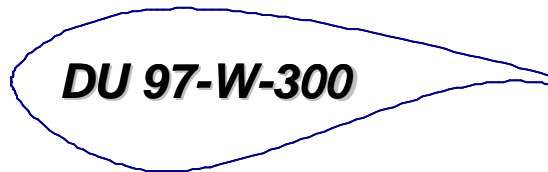
Note: do not use the built-in functions *max* or *find*; also dot arithmetic is not allowed since this week's topic is loops.

Write a script file named **profile.m** which performs the following tasks:
- reads the text file
- determines the angle of attack α_0 for which the lift to drag ratio is maximal. Display this maximum lift to drag ratio as well as the angle of attack α_0 to the screen.

- determines the smallest angle of attack α_1 for which the lift coefficent is larger than 1.008 **and** the drag coefficient less than or equal to 0.204. Display this angle of attack α_1 to the screen. Use a **for-loop** (and no while-loop)
- determines the largest angle of attack α_2 for which the lift to drag ratio is larger than or equal to 100 **or** the drag coefficient less than 0.108. Display this angle of attack α_2 to the screen. Use a **while-loop** (and no for-loop)
  Hint: you can use one of the following identities in order to arrive at a correct condition:

  $\sim(L1 \ \& \ L2) \equiv \sim L1 \ | \sim L2$

  $\sim(L1 \ | \ L2) \equiv \sim L1 \ \& \ \sim L2$

2. (4 points)
Write a function file named **ExpSeries.m** that computes the value of e^x at a given x using the series expansion of the exponential function (up to the term with x^n):

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + ... = \sum_{k=0}^{n} \frac{x^k}{k!}$$

Let x and n be the input to your function and y and err be the output. y is the series expansion and err is the percentage error in y when compared to (Matlab built-in routine) exp(x).

Take care of the following points:
- **ExpSeries.m** is a function file; so do not prompt a user for input (*input* command) and do not display (intermediate) results to screen
- Header (H1 line) and comments
- The function should accept a vector x. **Note: for this assignment you are NOT allowed to use dot arithmetic**
- Include a check on the input n. The function should proceed only if $n>0$ is an integer
- A default value of $n=8$ should be taken if the user does not specify n
- If you did everything right you will see that the variable 'err' becomes smaller for larger n. Try it yourself

3. (1 point)
Write a short script file named **aero.m** which contains an example of a for-loop which is aerospace related (i.e. not about money or beer). Make sure (through variable names and comments) that your example can be understood by a fellow student and is recognizable as an aerospace topic.

4. (2 points)
Write a function file named ***squares.m*** with three inputs (integers i1, i2 and i3) and two outputs (x and y). The function should perform two tasks (read it a couple of times in order to figure out what is exactly asked):

- the output x is a vector of exactly i2 elements: the squares of integers starting from i1 (inclusive)
- the output y is a scalar: the first square of an integer (starting from 0) that is larger than i3.

Do not forget the header (H1 line) and comments.

You have to upload 4 files in total:
- ***profile.m***
- ***ExpSeries.m***
- ***aero.m***
- ***squares.m***