

## Aerodynamics Computational Assignment Guidelines

### MATLAB Code Layout Details and Best Practices:

**Main Script:** Your computational assignment should be organized such that it can be run from *one* “Driver” or “Main” script. You should not expect the graders to input any values or interact with the code (e.g. un-commenting lines) to allow it to run properly. Your code, with all subroutines, should run “out-of-the-box” from the same directory and print all of the desired results.

**Header:** Each of your scripts, subroutines, and functions associated with your computational assignment code should have a clear and detailed header including the following fields and structure:

#### *Main.m*

---

```
%% ASEN 3111 - Computational Assignment XX - Main
% Provide a brief summary of the problem statement and code so that
% you or someone else can later understand what you attempted to do
% it doesn't have to be that long.
%
% Author: {primary author, should be you}
% Collaborators: J. Doe, J. Smith {acknowledge whomever you worked with}
% Date: {should include the date last revised}

% Then start your code, i.e.
clc
clear all
close all
...
```

---

#### *NewFunction.m*

---

```
function [outputArg1,outputArg2] = NewFunction(inputArg1,inputArg2)
%NewFunction Summary of this function goes here
% Detailed explanation goes here
%
% Author: {primary author, should be you}
% Collaborators: J. Doe, J. Smith {acknowledge whomever you worked with}
% Date: {should include the date last revised}
outputArg1 = inputArg1;
outputArg2 = inputArg2;
end
```

---

**Subroutines and Functions:** Try not to write one single “comprehensive” code. Instead break-up tasks and operations into subroutines and functions that can be executed on repeat and may also be useful for you in later assignments, projects, and your professional life.

**Comments:** Use comments regularly throughout your code to help better explain what you are doing to yourself and others.

**Printed Comments:** It is often value to print status comments to the command window during different stages of your code operation. This helps you better debug issues in your code

operation and will help Graders determine how long a code is running or if it is hanging up. You can also time how long it takes to run the code and have it output to the command window to help you improve efficiency.

---

```
for i = 1:5
fprintf('I am on interation: %i \n', i);
end
```

---

*Prints to the command window:*

```
I am on interation: 1
I am on interation: 2
I am on interation: 3
I am on interation: 4
I am on interation: 5
```

**Variable/Function Names:** Appropriately name variables and functions to provide clarity to what they represent or do.

**Data Files:** If the data files are relatively small in size and given to you by the course instructor include them with your code submission so that your code is fully functional. If you have to load, parse, and potentially modify the data files then you should perform two operations:

1. Build a MATLAB script that “Post-Processes” the data storing only the essential variables needed for your analysis and plotting and then save out of this script a MATLAB data file, i.e. .mat extension file, that includes all the required variables. This will be more efficient and easy for you to use.
2. Build a “Main” analysis script that loads in the post-processed MATLAB data file, analyzes the results, and plots/prints the requested figures/findings.

**Figure/Plot Details:**

- All of your figures and plots should contain axis labels with units, if applicable.
- All figures and plots should include legends to help understand the data that is plotted. Please ensure that the legend does not interfere with seeing or understanding the figure.
- For all symbols in the plots and figures, the symbol should be used rather than text. Use  $2\pi$ , not  $2*pi$ .
- Whenever possible, utilize a single figure to illustrate multiple data sets. For instance, if you are plotting sectional coefficient of lift versus angle of attack for four different airfoils, include one figure with four graphs rather than four figures with one graph each.

**Printed responses to the Command Window:**

- When responding to questions on the assignment with a print-out to the command window please include a caption or comment line defining what the printed value or variable means. In other words provide context to what ever is printed to the command window.

**Calculation Time:**

- For each Computational Assignment your final submitted computational code must run to completion in roughly **1 min** or less. This requirement is imposed to reduce the required time that the instruction team must spend grading (or rather waiting to grade) each submission.
- Whenever possible remove or comment out unnecessary steps in your code prior to submission.
- If you are unable to accomplish all the necessary processes in the allotted time consider using a “*switch*” and storing variables that you have previously computed, but detail in comments how those variables are computed and provide the code needed to compute them.

**Submission Details:**

- Copies of all MATLAB files, including processing functions, scripts, and a consolidated data file (i.e. .mat format), should be uploaded to the appropriate folder on Canvas by 11:59pm on the due date. Please combine all MATLAB files into a single ZIP folder prior to uploading. Failure to adhere to this requirement will result in points being deducted.
- MATLAB files will **NOT** be accepted after the given due date. If there are extreme extenuating circumstances, you must let the Professor know **BEFORE** the assignment is due for accommodations to be considered.
- MATLAB files submitted to Canvas should adhere to the following naming convention or points will be deducted:

**ZIP File:** {CA1,... or CA4}\_{LastName}\_{First Name}.zip  
**Driver Script:** Main\_{CA1,... or CA4}\_{LastName}\_{First Name}.m  
**Data File:** Data\_{CA1,... or CA4}\_{Last Name}\_{First Name}.mat  
**MATLAB Sub Routines and Functions:** Name Appropriately

**Grading Details:** The following table summarizes the grading rubric for the computational assignments:

Component	Grading Category	Grading Criteria
MATLAB Code (100%)	Functionality (30%)	Does the code run “out-of-the-box”?
	Completeness (30%)	Are all the requested results/figures included?
	Accuracy (20%)	Are the included results/figures correct?
	Professionalism (20%)	Are the results clearly presented/defined?

**Standardized Point Deductions:****1. Improper MATLAB file Submission**

Each student is required to submit all associated MATLAB files they created. If a student does not submit all the files separately with the requested naming convention 5 points will be deducted from the overall score.