ME 2016 - Computing Techniques
FALL 2018
Section A

Homework 0
Due Tuesday, September 4 at 2:55 pm
(5 minutes before class time)

MATLAB coding and file uploading to Canvas

The three main purposes of this homework are (1) to familiarize yourself with the guidelines provided in the Resources folder of Canvas, and to use them to (2) write a simple MATLAB code and (3) write a brief report. You will also get acquainted with the process of uploading your files to Canvas for grading. This assignment should be considered a practice run for future Matlab homework and projects.

You will determine the velocity of the falling parachutist using the method described in class; in order to do that, you will code the two *functions* `para` and `Euler` as discussed in class.

First, code `para` in exactly the same manner as the class version.

Then, modify `Euler` to have the following syntax:

```
[TOUT,YOUT] = Euler(problem,yi,ti,tf,dt)
```

where all the input arguments (except the first one) are the same as those defined in class, TOUT is a vector containing the incremental time values from ti to tf and YOUT is the solution vector corresponding to the times in TOUT (i.e.: each element in YOUT is the computed Euler approximation at the corresponding element (time) in TOUT; the first element in TOUT is ti and the first element in YOUT is yi).

The first input argument, `problem`, is called a *function handle* and will allow you to make Euler a completely generic function, (i.e. without any reference to the specific parachutist problem). `problem` is the generic name of the file that contains the slope that is needed by `Euler`. To use the function handle in `Euler`, simply replace the call to `para` (in the class version) by a call to `problem`.

Instead of the `while` loop presented in class, use a `for` loop. If you are having trouble with this, you can use a `while` loop, but with a 15-point penalty.

In either case, make sure that the last element in TOUT is always tf, regardless of the values of ti, tf and dt (i.e. even in the case when (tf – ti) is not evenly divisible by dt) and make sure that the last value of YOUT is always the Euler approximation calculated at the correct tf.

Finally, write a *script* called HW0 that does the following:

- plots the exact solution (Equation 5 in my notes) for the velocity from t = 0 to t = 25 s, using a *smooth (i.e., use at least 100 points to plot), continuous line* (use the values given in class for *c* and *m* and use *g* = 9.81 m.s^{-2}).
- calls Euler and plots the approximate solution for step sizes of 2s, 1s and 0.5s, all on the same plot as for the exact solution, using *discrete points* (such as circles) of different colors to differentiate the different step sizes (use tf = 25s). Include a legend. Your script needs to call Euler 3 times, one for each value of the step size. To call Euler to solve the parachutist problem, use the following syntax: `Euler(@para,yi,ti,tf,dt)`
- for the 0.5s step size case, display the value of the approximate velocity at tf = 25 s in the Matlab command window in a *nicely formatted way*, which means not just a number (with a reasonable number of significant figures), but a sentence with enough information (including units), so that a user who is not familiar with your code can understand what this returned result is. Look-up the `fprintf` command to do this.

In a PDF-formatted report, include a copy of your plot and a brief discussion of your results. Please make sure that your script and functions are **formatted according to the instructions posted on Canvas** (including *headers* and plenty of *useful comments*), and that your plot has **descriptive titles and appropriate labels**. Also, please look at the guidelines for report writing posted on Canvas, as well as information on how to use the Matlab Help and Documentation (in the Files/Resources folder)

Include your 3 m-files and your pdf report in a **zipped folder** named **HW0_Lastname_Firstname.zip** and upload to Canvas by the due date and time above. Please use the **.zip** format only to avoid problems with opening your folder zipped in other formats.

Please note the following, valid for this and all subsequent MATLAB assignments:

1. **Refrain from using the commands `clear`, `clc` , `clf` and `close` (or any of their variations) in any of your m-files, as they interfere with the grading process.**

2. **When your code is run, it should produce only what is asked for in the assignment (for HW0, only one plot and one result displayed in the command window; nothing else). Please make sure to suppress any unnecessary intermediate result in any form: no other figure, curve or value printed in the command window should appear.**

You will lose points if you ignore these instructions.

Please note that Canvas will only accept a file with a ZIP extension: all of your M-files and the PDF report must be in that zipped folder. If we cannot open your files because you failed to follow these instructions, you will get a zero for this assignment.

You can upload your folder as often as you like, in case you find an error in your work. Older files will be kept in Canvas, but **only the last submitted folder will be graded**. It is your responsibility for this (and all other) electronic assignment to make sure we are grading the correct folder.

You can upload your work up to 24 hours after the due date, but with a 50% penalty, as described in the syllabus. For this particular assignment, no submission will be accepted after 2:55 pm on Wednesday, September 5.

Finally, please keep in mind the policies about late assignments and collaboration outlined in the syllabus, and copied here for your convenience:

- Late assignments: homework and projects will be accepted up to 24 hours after the deadline with a 50% penalty. This policy will be strictly enforced and no submission will be accepted after 24 hours. Hard copies of homework will be due at the end of class. ***It is your responsibility to make sure that you have successfully uploaded and submitted all the required files in the required format to Canvas on time***: please double-check to avoid having to re-submit your work after the due date. The only acceptable proof that you have submitted your work on time is the time stamp of your Canvas submission. ***Files will not be accepted by email even if they show a "last modified date" that is before the due date.***

- Collaboration: students may *discuss* their assignments with each other, but homework and projects must be *completed individually* by each student. ***You must turn in your own work***. Copying someone else's work and submitting it as your own will not be tolerated. In particular, ***for Matlab assignments***, this policy means that students can discuss aspects such as the general approach to solve a problem or the syntax of a specific command, but ***they should not look at each other's codes***. If it is suspected that this has occurred, you will be reported to the Dean of Students for an honor code violation.