ASEN2012 - Coding Challenge 3 (25 pts) - Due Wednesday 10/18 by 11:59 pm
EXPERIMENTAL AND COMPUTATIONAL METHODS IN AEROSPACE ENGINEERING SCIENCES

Key Deliverables:

- 5 "Check your answer" questions in Canvas (**10 pts**)
  - **You only get 3 attempts** (correct answers carry over to next attempt) at the quiz.
  - **There is a 5 minute "cool-down" between attempts.** Use this time to debug and reevaluate your code. Once you compute a new answer, make sure to critically think about it before submitting again.
  - **No additional attempts will be provided for any reason.**
- 1 .zip file containing the following (naming convention lastname_firstname_CC4.zip) (**15pts**):
  - Functioning Matlab code (.m file(s) - if using multiple files, please name driver code main.m)
  - Published code (.pdf file)
  - Flowchart sketch (.pdf file)
- All numeric quiz answers must be **computed** in your code!

Background

Numerical integration comes in many different flavors, which often have varying levels of fidelity. Today we will explore Euler integration and 4th order Runge-Kutta integration.

Explicit Euler integration was published in 1768 and derived in class from Taylor Series Expansion of $f(x_{i+1})$ about some point x_i, where $\Delta x = x_{i+1} - x_i$. When higher order terms are neglected, as $\Delta x \ll 1$, an equation for explicit Euler integration can be found:

$$y_{i+1} \approx y_i + g(x_i, y_i) \cdot \Delta x \text{ where, } g(x,y) = \frac{dy}{dx}$$

4th order Runge-Kutta integration was first proposed sometime between 1895 and 1901 and shown in class as a weighted average of 4 different representations for the slope of some function. As before, $g(x,y) = \frac{dy}{dx}$ and we'll write dy/dx as $\dot{y}$. 4th order Runge-Kutta is then defined by the following:

$$y_{i+1} \approx y_i + \frac{\Delta x}{6}\left(\dot{y}_1 + 2\dot{y}_2 + 2\dot{y}_3 + \dot{y}_4\right) \text{ where,}$$

$$\dot{y}_1 = g(x_i, y_i)$$
$$\dot{y}_2 = g(x_i + \frac{\Delta x}{2}, y_i + \dot{y}_1\frac{\Delta x}{2})$$
$$\dot{y}_3 = g(x_i + \frac{\Delta x}{2}, y_i + \dot{y}_2\frac{\Delta x}{2})$$
$$\dot{y}_4 = g(x_i + \Delta x, y_i + \dot{y}_3\Delta x)$$

In Matlab, anonymous function handles are often easy ways to evaluate equations - the syntax for writing $g(x,y) = 1 + x^2 - \exp\left(y^4\right)$ would be `g = @(x,y) 1 + x.^2 - exp(y.^4);` with suppressed output.

Problem 0: Write functions for Euler and RK4 integration

Write two separate Matlab functions to perform Euler and RK4 integration on *anonymous* function handles. Other inputs for your functions should be the initial conditions, as a vector - $[x_0, y_0]$, the final x value - x_f, and step size - Δx.

Problem 1: Euler Integration

For this problem, we will be integrating the following differential equation:

$$\frac{dy}{dx} = e^{-x^2}$$

The initial condition is represented by $y_0 = y(x = 0) = 1$. Integrate this equation forward with a Δx of 0.1 until you reach $x_f = 2$ and compare your values stored in the downloaded .mat file, AccurateData_CC4.mat (produced by `ode45`) - the x and y coordinates of the accurate data are provided in variables `x_accurate` and `y_accurate`, respectively.

In Canvas quiz, enter the *final* value, y_f, with 4 significant digits.

Compute the root mean square error (RMSE) of the Euler integration solution, and that of the accurate data. RMSE is defined by the equation below:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \hat{x}_i)^2}{N}}$$

Here x_i is the predicted value (Euler integration), $\hat{x}_i$ is the actual value (from the provided data set), N is the number of points. You will note that the two solutions are of different sizes! You will need to use `interp1` (`https://www.mathworks.com/help/matlab/ref/interp1.html`) to "down-sample" or interpolate the accurate data to corresponding steps as your Euler integration using `newy_accurate = interp1(x_accurate,y_accurate,x,'linear','extrap');` explore what happens if you don't include extrap option.

In Canvas quiz, enter the calculated RMSE value, with 4 significant digits.

Problem 2: Euler and RK4 Integration

We will now be working with the following equation:

$$\frac{dy}{dx} = y\sin^2(x)$$

For this, the initial condition is defined as $y_0 = y(x = 0) = \pi$. We are interested in integrating this differential equations until a final value of $x_f = 3\pi$. Compare the Euler explicit integration and RK4 for 4 different Δx values of π, $\pi/2$, $\pi/4$, $\pi/8$.

The exact, analytical solution to this differential equation is:

$$y_{\text{exact}}(x) = \pi \cdot \exp\left(\frac{2x - \sin(2x)}{4}\right)$$

Create plots showing a comparison of Euler integration, RK4, and the exact solution for each different step size (4 plots).

Compute a vector of RMSE (1×4) of Euler integration compared to the exact solutions and for RK4 integration.

In the Canvas quiz, you will be asked for three values from the eight values in the RMSE matrices from Euler and RK4 integration. (3 separate questions)

Reflection Question

In Problem 1, why might you "down-sample" the exact/accurate data instead of "up-sampling" the Euler integration to match the accurate data?

Why does "extrap" need to be included in the `interp1` call?

Please write out the answers to these questions in the comments of your Matlab script. This should be about 1 paragraph in length.

Coding Rubric

	Excellent (100%)	Above Average (80%)	Average (70%)	Below Average (50%)
Requirements and Delivery (2pts)	• Completed 90-100% of the requirements • Delivered on time, and in correct format.	• Completed 80-90% of the requirements • Delivered on time, and in correct format.	• Completed 70-80% of the requirements • Delivered on time, and in correct format.	• Completed <70% of the requirements • Delivered on time, but not in correct format.
Coding Standards (2pts)	• Includes full header* • Excellent use of variables (no global or unambiguous variable naming)	• Includes full header* • Good use of variables (1-2 global or ambiguous variable naming)	• Includes incomp. header* • Fine use of variables (3-5 global or ambiguous variable naming)	• No header • Poor use of variables (many global or ambiguous variable naming)
Documentation (2pts) Comment your code	• Clearly documented • Specific purpose noted for each function and/or section	• Well documented • Specific purpose noted for each function and/or section	• Some documentation • Purpose noted for each function and/or section	• Limited to no documentation
Runtime (1pt)	• Executes quickly, without errors	• Executes without errors, over 1 min runtime	• Executes with warnings/errors	• Does not execute
Efficiency (2pts)	• Easy to understand, and maintain	• Logical, without sacrificing readability and understanding	• Difficult to follow	• Difficult to follow, huge and appears patched together
Figure Quality (2pts)	• Easy to understand, labels and legend present	• Easy to understand, lacks labels and legend	• Difficult to understand, labels and legend present	• Difficult to understand, lacks labels and legend
Reflection Questions (4pts)	• Easy to understand, fully thought out	• Easy to understand, mostly thought out	• Hard to understand, effort made	• Hard to understand, lacking effort

* Header includes author name(s), assignment title, purpose, creation date, revisions (applicable to **group** projects only)