QUIZ 4 – MATLAB

Directions:

1. This is a limited open book quiz, meaning that the ***only*** sources of information allowed are *lecture notes, tutorial problems, textbook and your handwritten notes* reiterating/reinforcing the material presented in class. This information can be accessed in soft or hard copy form.
2. Please note that the access of files, documents, or information ***other than*** that explicitly listed at Point 1 will be treated as cheating and will be dealt with accordingly.
3. The access of social media, text messages, email or any other form of electronic communication will be treated as cheating and will be dealt with accordingly.
4. "Teamwork" and/or discussions are not permitted during the quiz.
5. The use of cell phones during the quiz is prohibited.
6. Only your owl submission will be marked.
7. **Quiz duration: 90 minutes (9:30 am – 11:00 am) + 10 minutes upload time**
8. **Total Duration: 100 minutes (9:30 am – 11:10 am)**

Submission Directions:

9. Code all your solutions in MATLAB. Save each problem as a separate '.m' file.
10. Ensure your upload to OWL is successful and that all files are functional. If there is a problem with your upload you must let the professor or TA know **BEFORE** leaving the zoom room. After leaving the quiz missing or corrupt files will not be accommodated for.

Note: While performing problems you can use the help function built into MATLAB to help you with syntax, and the proper use of commands.

Problem 1 [10 marks]

Tutorial 4 asked you to write a function called `SFPI` that implemented the simple fixed-point iteration method of solving roots. This is also described in the Chapter 2 lecture notes.

a) Use the `SFPI` function provided to you on OWL to calculate the root of the following function between $-50 \leq x \leq 20$. You will need to make an educated guess regarding what to use as an initial guess (x_o), and desired error (e_s). **[3 marks]**

$$f(x) = \cos(\sqrt{x})$$

Note: A poor initial guess or an e_s which is too low will cause SFPI to diverge. You can plot the function to allow you to make a good initial guess.

b) Now, calculate the root of the function in this range using the built in MATLAB `fzero` function. **[3 marks]**

c) Format your output to include i) the root as solved by `SFPI`, ii) the root as solved by `fzero`, and iii) the error of your `SFPI` calculated root compared to the `fzero` result in percent format. You must have the three desired results output to the command window in a clear and concise manner after the script is executed. **[2 marks]**

d) Plot the given function over the range $-50 \leq x \leq 20$ as well as the two calculated roots using the `plot` command, both in Figure 1. The function should be plotted as a green dotted line, the `SFPI` root should be plotted as a black diamond, and the `fzero` root should be plotted as a magenta plus sign. A table of plot specifiers has been included below, typing `help plot` input into the command window will show you how to use them. **[2 marks]**

Hint: if the command '`figure(1)`' placed before several plot commands, all the plotting will be output in Figure 1.

Hint: a list of all plot identifies can be viewed by looking in the output of the help text for the plot command, which can be accessed using the command `help plot` in the command window.

Upload your appropriately named '.m' file(s) to OWL when completed.

Problem 2 [10 marks]

The downward velocity of a bungee jumper starting from rest can be described by the following analytical solution.

$$v(t) = \sqrt{\frac{gm}{c_d}} tanh\left(\sqrt{\frac{gc_d}{m}}\, t\right)$$

Assume that $g = 9.81 m/s^2$ and $m = 70\ kg$ but the value of c_d is not precisely known for this particular jumper. The average drag coefficient c_d for a human is normally distributed about the mean of 0.25 kg/m with a standard deviation of 0.05 kg/m.

a) Use the `randn` function to generate 1000 random values for c_d. These should be normally distributed and have the desired mean and standard deviation as outlined above. Store these values in an appropriately named array. **[2 marks]**

   Hint: recall that we can use `randn` to generate a normally distributed range of values with a specific mean and standard deviation. For example, as stated in the `help rand` documentation, `r = 1 + 2.*randn(1000,1)` will populate r with 1000 values having a mean value of 1 and a standard deviation of 2.

b) Calculate the corresponding jumper velocity at t = 4 s for each random value of c_d produced in part (a). Store these values in an appropriately named array. **[1 mark]**

c) Calculate the mean and standard deviation of the resulting values of c_d and v, and store them in appropriately named variables. **[1 mark]**

d) Construct two histograms from the resulting values of c_d and v with proper titles and axis labels. You can use the `hist` function and the default MATLAB histogram settings. Please output these in one figure using the `subplot` function. If you need clarification on how to use this function, `help subplot` will show you examples. **[3 marks]**

e) Use the provided `linregr.m` function, fit a linear line to c_d with respect to v. The resulting plot should be output after the two histograms output in part (d). **[3 marks]**

Upload your appropriately named '.m' file(s) to OWL when completed.

Problem 3 [10 marks]

An engineer involved in construction requires a final mixture containing 4800 m3 of sand, 5800 m3 of fine gravel, and 5700 m3 of coarse gravel for a building project. There are three pits of various compositions that are available to draw from shown below.

	Sand %	Fine Gravel %	Coarse Gravel %
Pit1	55	30	15
Pit2	25	45	30
Pit3	25	20	55

We wish to compute how many cubic meters must be hauled from each pit to receive the desired amount of sand, fine gravel, and coarse gravel in the final mixture.

a) Arrange this problem into the form [A]{x}={b} such that it can be solved using MATLAB, and define the [A] and {b} matrices in your script. **[2 marks]**

   Note: In this case, the [A] matrix will be defined from the table above, the {x} vector will contain the volumes that must be drawn from each pit (x1, x2, x3), and the {b} vector will be populated with the volumes of each constituent desired in the final mixture (4800, 5800, 5700).

   Hint: Make sure your [A] matrix is oriented correctly! Feel free to write out the equations to make sure you have everything in the correct position.

b) Use the provided Gauss naïve elimination `GaussNaive.m` function to solve this problem. **[4 marks]**

c) Use the 'left hand division' technique to solve the problem. **[4 marks]**

Upload your appropriately named '.m' file(s) to OWL when completed.