ECOR 2606 Lab #1

To get marks for completing the lab you must get each part checked by a TA (after you complete it) <u>during</u> the lab session. Of course, you may ask questions if you have difficulties. If you are not able to complete the entire lab during your lab section, you may complete the lab during another lab section **for that week's lab.** You can learn when these are scheduled by reviewing the course outline. Even if you do not complete the entire lab, **submit "*lab1.m*" at the end of the lab, to prove your attendance.**

Create a script file called "*lab1.m*". Put your name and student number at the top preceded by "%" (which indicates a comment). **You must always include your name and student number in your Matlab code for full marks!** After you get a line of code working in the command window, copy and paste it into "*lab1.m*", with a comment indicating which Part (e.g. Part 1) and subpart (e.g. a) for which the code is. **Note that all Matlab code in tests (written tests and lab quizzes) must have comments for full marks.**

Before you start the lab you should double **check that your calculator is on the list of approved calculators** ("*2606calculators.pdf*") in "*G:/2606*". It is also recommended that you **open the Matlab reference file** ("*matlabreference.pdf*") in "*G:/2606*". Only approved calculators will be allowed for use during tests and exams. The reference sheet will be the only external aid you will be allowed during exams.

Part 1

An experiment involves the growth of e-coli in a petri dish. It has been determined that the growth rate of the bacteria is governed by the following equation:

$$B(t) = \frac{250}{1 + 56.75e^{-0.17t}}$$

where $B(t)$ is the number of bacteria (in thousands)
t is the time (in hours)

Note that in Matlab we use function *exp(x)* to calculate e^x.

a) Use Matlab in "calculator" mode to calculate the number of bacteria at t = 1 hour. In other words, type the right hand side of the above formula into the Matlab command window substituting 1 in the place of t.

b) Now use Matlab in "calculator" mode to calculate the number of bacteria at t = 4 hours. Keep in mind that in Matlab previous commands can be recalled using the up arrow. To evaluate the expression for t = 4 hours you need only recall and edit the command that you used to evaluate the expression for t = 1 hours.

c) Implement $B(t)$ as an "anonymous" function.

d) Test your function with $t = 1$ hour. In other words, type $B(1)$. You should, of course, get the same answer as you did in part a).

e) Test your function with $t = 4$ hours. Again, you should get the same answer as in part b).

f) Now try your function with a vector containing 1 and 4 as shown here: B([1 4]). If you get an error, you will need to fix part c). Remember that we want element by element division, not matrix division. Hint: A "." is needed.

g) Use *fplot* to create a graph (figure 1) showing $B(t)$ for from 0 to 40 hours. Add a grid, a title, and appropriate x and y labels. **Note that all graphs required in tests (written tests or lab quizzes) must have a figure number, title, *x* and *y* axis labels, and a grid for full marks.**

<u>**Note:**</u> When we want to use root solving, the first step is to rearrange our formula so that we have a "0" on the right hand side. This rearranged formula is called the "root finding formula", or we may say that we have put things in "root finding form". Once we have the "root finding formula", we take the left hand side of the formula and create our "root finding function."

h) Suppose we would like to know exactly when there will be 150,000 bacteria in the dish. Remember that the formula above gives the number of bacteria in <u>thousands</u>, i.e. we want to know when $B(t) = 150$. This is a root solving problem. Write the root finding formula as a <u>comment</u> in your script file. <u>No</u> Matlab code is required in this step.

i) Now write your root finding function as an anonymous function. You can call your function anything you like except B (it doesn't have to be f). Note that function definitions can make use of previously defined functions. In this case you will find it very convenient to make use of $B(t)$. In other words, do <u>not</u> write out the above equation again. (We will find a root of this function in later steps.)

j) Use *linspace* to generate a vector of 50 equally spaced time values between $t = 0$ hours and $t = 40$ hours. Then use these x values and your $f(x)$ to create a vector containing the corresponding y values. Use *plot* to create another graph (figure 2) by plotting the x values against the y values. Again add a title, x and y axis labels, and a grid. Compare your two graphs and make sure that the y axis values are different.

k) By looking at figure 2 you can see the approximate root (i.e. the point where your root finding function is equal to 0). Use *fzero* to precisely locate the required root, passing in a small range determined by examining your graph. Output the root nicely (i.e. in a sentence, e.g. 'The number of bacteria is 150,000 when t=…') using *fprintf*. **Note that all values calculated using Matlab in tests (written tests or lab quizzes) must be output using *fprintf* for full marks.**

Verify that executing the script file, "*lab1.m*" gives exactly the same results as entering these commands in the command window.

Get Part 1 checked by a TA before starting Part 2 (on the next page).

Part 2

A hollow brass sphere has a wall thickness of 0.01m and a mass of 100 kg. What is the outside **diameter** of the sphere?

Useful information: Density of brass = 8900 kg/m^3
Volume of a sphere = $(4/3)\pi R^3$

The above information and some mathematics will give you the following root finding formula, where "R" is the radius of the sphere:

$$1118.406985R^2 - 11.18407R - 99.96272 = 0$$

Find the root twice, once using *fzero* and once using *roots*. To use *fzero*, you will need a graph to help you find the range to pass to the function. To use *roots*, you need a vector containing the coefficients of your root finding polynomial. Hint: Don't forget that you are asked to find the diameter (not radius).

Your solution to part 2 must include figure 3 (labelled with a grid, as per usual), the graph you used to help you find the range for *fzero*, and two *fprintf* statements, displaying the diameter you found, one *fprintf* statement each for *fzero* and *roots*. An example of expected output is:

"The diameter found using fzero is …… meters"

Of course, the correct answer will replace the ……!

Put your code for part 2 at the bottom of "*lab1.m*", including appropriate comments.

Get Part 2 checked by a TA before starting Part 3.

Part 3 (Bonus)

Derive the equation used in Part 2. Hint: The volume of brass is equal to the volume of the sphere minus its interior volume.

Include your calculations as comments at the bottom of "*lab1.m*".

Get Part 3 checked by a TA.