ECOR 2606 Lab #6

To get marks for completing the lab you must get each part checked by a TA (after you complete it) during the lab session. Of course, you may ask questions if you get stuck. If you are not able to complete the entire lab during your lab section, you may complete the lab during another lab section. Even if you do not complete the entire lab, **submit "*lab6.m*" at the end of the lab, to prove your attendance.**

Part 1

Analysis of an engineering problem has produced the following equations:

$$\begin{aligned} 2a - 14b + 5c &= 52 \\ 5a + 6b - 2c &= 2 \\ a + 5b + 3c &= 17 \end{aligned}$$

(i) What are a, b, and c? Obtain the solution **three** times, twice using the inverse of A (calculated two different ways), and once using left division. For each solution, use *fprintf* to output a message containing the values of a, b, and c, and use *fprintf* to output only the value of b (i.e. output a message of the form "The value of b is ...."). This is a total of **6** *fprintf* statements (2 for each of the three solutions). Place all of the necessary commands in a script file called *lab6.m*.

(ii) Now solve the system of equations by hand using Gaussian elimination with partial pivoting exactly as taught in the lectures. Note that if you deviate from the process taught in the lectures, you will get only part marks on tests!

Get Part 1 checked by a TA.

Part 2

Analysis of another engineering problem has produced the system of equations shown below. Y is a parameter that can be varied.

$$\begin{aligned} (5-Y)x_1 - 4x_2 + 4x_3 &= 1 \\ -5x_1 + (4+Y)x_2 + 3x_3 &= 4 \\ 5x_1 + 3x_2 + (1+Y)x_3 &= -2 \end{aligned}$$

(i) Write a function m-file called *findAllX* that given Y will calculate and return a vector (x) containing *x1*, *x2*, and *x3* based on the above equations.

(ii) In your script file (*lab6.m*) calculate and output (using *fprintf*) *x1*, *x2*, and *x3* if Y is 0. Hint: Be sure to use your function from part (i).

(iii) Now write a function m-file called *findX1* that given Y will use *findAllX* to calculate the vector *x*, and return just *x1*.

(iv) In your script, create a graph (figure 1) using *fplot* of *x1* vs. *Y* for *Y* from 0 to 12. Hints: Use your function from part (iii), and remember what the "function handle" is for a function file.

(v) In your script, create a graph (figure 2) using *plot* of *x1* vs. *Y* for *Y* from 0 to 12. Hint: *findX1* is not vector friendly, so you need a loop.

(vi) Also in your script calculate the *Y* value in this interval (0 to 12) that minimizes *x1*, and the minimum *x1* value. As usual, output these values nicely using *fprintf*.

(vii) Now change your script to a "no input / no output function" and include your functions (*findX1* and *findAllX*) as subfunctions, as follows:

```
function [] = lab6 ()
% the commands required to produce the required output  for parts 1 and 2 go here
% this is your "script"
end

function [x1] = findX1 (Y)
% solves system and returns value of x1 only
% should make use of "findAllX"
end

function [x] = findAllX (Y)
%  solves system of equations and returns a vector of x values
end
```

Get Part 2 checked by a TA.

Part 3 (Bonus)

In the 1976 Gordon Lightfoot song "The Wreck of the Edmund Fitzgerald," when the ship left Superior, Wisconsin "there was a load of iron ore 26,000 tons more than the Edmund Fitzgerald weighed empty." Suppose that the original plan was to offload 14,000 tons of ore at Sault Ste. Marie, Ontario at which point the ship and load would have had a mass of 112,000 tons, then proceed to Cleveland, Ohio to offload another amount such that the ship and cargo had a mass half as much as it did when it took on the original cargo in Superior.

(i) Set up the equations to let you establish the mass of the ship, the mass of the original cargo, and the mass to have been offloaded in Cleveland.

(ii) Write the Matlab code to solve the set of equations you created in part (i), and write the results using *fprintf*.

Get Part 3 checked by a TA.