Linear Algebra, Spring 2016
Homework 5, Due Tuesday, March 8

The file Homework5.zip contains the following MatLab files:

- `hasSolution.m`
- `hasUniqueSolution.m`
- `isInSpan.m`
- `isLinInd.m`
- `spanningSetToBasis.m`
- `linIndSetToBasis.m`
- `compareSubspaces.m`

Unzip the Homework5.zip file and put all the files into a single directory. Open MatLab and change to this directory. The files `hasSolution.m` and `hasUniqueSolution.m` are complete files. They contain the code for a function that checks if a linear system has a solution, and if a homogeneous linear system has a unique solution. To test that they work, enter the following commands into MatLab:

> hasSolution([1 2 4;3 5 6])

> hasUniqueSolution([1 2;2 4])

The first command should return 1 (true) because the system

$$\begin{cases} x + 2y = 4 \\ 3x + 5y = 6 \end{cases}$$

has a solution. The second command should return 0 (false) because the homogeneous system

$$\begin{cases} x + 2y = 0 \\ 2x + 4y = 0 \end{cases}$$

does not have a unique solution (in other words, it has infinitely many solutions).

The remaining files are incomplete and your goal in this homework assignment is to complete them. The comments at the beginning of each of the files explains what they should do. I recommend completing them in the order that they are listed, because the earlier functions may be useful to call in the later functions. All of the functions can be completed with only a few lines of code (I needed less than 10 for each one when I did it). You are not allowed to use any built-in MatLab functions other than rref, basic matrix manipulation, and control statements (like if-then, for loops, etc).

When you hand in your work, hand in the following things in the order listed below:

1. (4 points) Explain in words, using complete sentences, how the algorithms in the functions hasSolution and hasUniqueSolution work. You don't need to write a lot, just a couple of sentences for each one that explains the most important aspects.

2. (this is considered showing your work for the next question) A printout of your programs `isInSpan.m, isLinInd.m, spanningSetToBasis.m, linIndSetToBasis.m, compareSubspaces.m`

3. (.5 points each) Run the following commands in MatLab and turn in a printout of the results, including the commands (Remark: `eye(n)` is the command for the nxn identity matrix. Its columns are the standard basis of $\mathbb{R}^n$.):

   > `isInSpan([-1 -2;2 6],[-8;22])`

   > `isInSpan([-2 5 3;3 7 10;4 -5 -1],[0;0;1])`

   > `isLinInd(eye(7))`

   > `isLinInd([eye(4) [1;0;0;0]])`

   > `spanningSetToBasis([0 1 2;0 -1 -2])`

   > `spanningSetToBasis([[1;2;3;0] [0;1;0;0] eye(4)])`

   > `linIndSetToBasis([[1;2;0;-1] [0;1;1;0]],eye(4))`

   > `linIndSetToBasis([[-2;1;3;4] [2;6;7;-10]],[0 -4 3;7 -5 -12;10 -4 8;-6 14 11])`

   > `compareSubspaces(eye(3),[1 3;4 6;2 -11])`

   > `compareSubspaces([1 2 3;4 5 6;7 8 9],[3 1;6 1;9 1])`

   > `compareSubspaces([3;6;9],[1 2 3;4 5 6;7 8 9])`

4. (.5 points each) For each of the above commands, write a sentence that interprets the result. Use symbols and notations we usually use in class instead of MatLab notation. For instance, for the command and output

   > `isInSpan([-1 -2;-2 -4],[-8 ;12])`

   > `ans=0`

   write something like: The vector $\begin{bmatrix}-8\\12\end{bmatrix}$ is not in span$\left\{\begin{bmatrix}-1\\-2\end{bmatrix}, \begin{bmatrix}-2\\-4\end{bmatrix}\right\}$.