Math 313/513, Homework 8 (due Thurs. Mar. 22)

Name:___________________________________ 313 or 513 (circle)

Reading

- Read sections 6.2, 6.3, and the first four pages of 4.2

Problems

- Math 313:
  - Section 6.2, problem 1 (show your work)
  - Section 6.2, problem 4 (explain your answers)
  - Section 6.2, problem 7
  - Section 6.2, problem 11
  - Section 6.2, problem 15
  - Section 6.3, problem 8 (ignore the comment on $-w^2$)
  - Find a 3×3 matrix whose eigenvalues are $3, 1, 0$ and whose corresponding eigenvectors are

$$\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \quad \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix}, \quad \begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix}.$$

1. Suppose A is 3×3 with eigenvalues $0, 1, 2$. Must A be diagonalizable? Why?
2. Suppose A is 3×3 with eigenvalues 0 and 2. Must A be diagonalizable? Could A be diagonalizable possibly? Why?

- Section 4.2, problem 1 (note Strang's notation differs from that given in class)
- Section 4.2, problem 2 (note Strang's notation differs from that given in class)
- Math 513: all of the above, plus:
  - Section 6.2, problem 36
  - Prove that for any matrix A, e^A is always invertible. (Hint: what should its inverse be?)

(OVER)

MATLAB assignment

How does a computer find the eigenvalues of a square matrix A? Rather than trying to find roots of a polynomial, many systems use the "power method," described below, at least for the problem of finding the largest real eigenvalue. First, choose a tolerance value, such as 10^{-6}. Start with x_0, a random vector of unit length with n entries. Start looping, where you set $y_k = Ax_{k-1}$. x_k will be the result of normalizing y_k to have unit length. Keep looping until $x_k - x_{k-1}$ has length less than the tolerance. (Note: by "length" I mean Euclidean length, not the number of entries; by "unit" I mean length 1; to "normalize" a vector, you divide it by its length.) Now the length of y_k is your estimate for the largest real eigenvalue, and x_k is your estimate for the eigenvector.

Write a function: `function [lambda, V]=lastname_eig(A)` stored in `lastname_eig.m` that takes in a square matrix A and returns your best guess for the largest real eigenvalue `lamdba` and eigenvector `V`.

Check your code on some matrices of your choosing to make sure it works correctly. In your comments, answer the following questions:

a. If you try your code on the matrix `[0 -1; 1 0]` what goes wrong? Answer in two ways: algebraically, in terms of eigenvalues, and geometrically, in terms of what the power method is doing. Also, please fix your code to avoid infinite loops!

b. Why does a 3×3 matrix always have a real eigenvalue?

Include your comments, and submit your code to Blackboard. Please remember to name your file as requested.