Math 495R Homework 19

In this lab you will use the python `numpy` and `numpy.linalg` modules to compute eigenvalues of matrices. To import these modules use the commands

```
>>> from numpy import *
>>> from numpy.linalg import *
```

To build a matrix in python, use the array constructor

```
>>> A=array([[1,2,3],[2,3,4],[5,4,3]])
```

Note, numpy has a "matrix" type, but this type does not work consistantly with other modules. You may use the built-in functions for matrix arithmetic if needed, but remember that to multiply two matrices `A` and `B` in Python, you should use the syntax `A.dot(B)` not `A*B`. The latter command will attempt to multiply the matrices entry-wise.

Throughout this lab, you may run into some numerical "noise" or precision issues in determining whether a value is in fact 0. For the purposes of this lab, you may assume that any value of absolute value $< 10^{-10}$ is 0.

(1) Write a function that finds the dimension and a basis for the null space of a given matrix. Recall that Nul A is the set of vectors $\mathbf{y}$ such that $A\mathbf{y} = \mathbf{0}$. You should use your function from lab 17 which row reduces a matrix to reduced echelon form. The dimension of the eigenspace is the number of non-pivot columns (i.e. the number of free variables).

(2) Write a function that takes an $n \times n$ matrix and prints out each eigenvalue along with the dimension and a basis for the corresponding eigenspace. You may use the `eig` command from the `numpy.linalg` module to calculate the eigenvalues of the matrix, but for this lab you should calculate the basis of the eigenvectors independently. The `eig` command returns a list of eigenvalues (repeated once per multiplicity), and a matrix P described below, which you should ignore for now. In order to assign just the list of eigenvalues of a square matrix A to a variable `EigVals` use the syntax

```
>>> EigVals,_=eig(A)
```

Recall that the eigenspace for a matrix A corresponding to a given eigenvalue λ is the null space of the matrix $A - \lambda I$.

(3) Write a function that tests whether a matrix is diagonalizable, and if so gives a diagonalization. The input of your function should be a square matrix A, and the output should be a list of the eigenvalues and a matrix P that satisfies $A = PDP^{-1}$, where D is a diagonal matrix containing the eigenvalues in the order given.

Recall that an $n \times n$ matrix A is diagonalizable if and only if the sum of dimensions of the eigenspaces of the matrix is n. If the matrix is diagonalizable, then the matrix P is an invertible matrix whose j-th column is an eigenvector for the j-th eigenvalue (as ordered in the given list). In order for P to be invertible, any columns corresponding to the same eigenvalues must be linearly independent, and therefore form a basis for the corresponding eigenspace.

In future labs you may use the `eig` command to diagonalize a matrix for you. If the input matrix is diagonalizable, the output of the `eig` command has essentially the same properties as the output of part 3 of this lab (though the matrices may not look the same). Unfortunately, the function does not test whether the matrix is diagonalizable. If the input matrix is not, then the output matrix P may not be invertible. One quick test would be to check the determinant of P. If it is sufficiently close to 0 you may assume that the original matrix is not diagonalizable.