

Numerical Methods

Math 3338 – Spring 2022

Worksheet 16

Eigenvalues and Eigenvectors

1 Reading

CP 6.2
NMEP Chapter 9

Table 1: Sections Covered

2 Overview

Recall, for an $n \times n$ matrix A we say λ is an eigenvalue and \vec{v} is an eigenvector if,

$$A\vec{v} = \lambda\vec{v}.$$

Think about what an algorithm to find eigenvalues of a matrix would need to do. It must find and solve the characteristic polynomial, or $\det(A - \lambda I)$. This is an n -degree polynomial in λ , which is hard to solve. One may think that Newton's method is the ideal solution to this problem, but you still need to find the characteristic polynomial which requires a determinant and a variable. It's a whole thing.

There are algorithms to compute eigenvalues and eigenvectors and they tend to be complicated. To work around this we'll be using numpy's linear algebra functions. [Link to all Numpy linear algebra functions.](#) In particular you should familiarize yourself with `numpy.linalg.eig` and `numpy.linalg.eigvals`. If your matrix happens to be symmetric, you should use the functions that specialize in symmetry (these end in "h" for Hermitian). They will be far more efficient.

3 Symmetric Matrices

In most applications in Physics and Engineering the matrix A is symmetric (or Hermitian if it's complex). In other words $A = A^T$ (or $A = \bar{A}^T$ that bar is a complex conjugate this is Hermitian). What this means is that the methods in the book only work for symmetric matrices. However, not all matrices are symmetric. We'll explore both symmetric and non-symmetric matrices.

Theorem 1 (Spectral Theorem). *A real symmetric $n \times n$ matrix has real eigenvalues and n orthonormal eigenvectors.*

Numerical Methods

Math 3338 – Spring 2022

Homework 16 (Due: Thursday, March 10)

Problem 1 (1 pt) Verify the Spectral Theorem by generating at least 10 random symmetric matrices, finding their eigenvalues and eigenvectors, checking that the eigenvalues are real and the eigenvectors are orthogonal (you don't need normal since that's just a scale). You should structure your programs in pieces, so like this

1. Make a function `random_symmetric(n)` to generate a random $n \times n$ matrix. I recommend looking into the transpose operation and thinking about why that might be helpful.
2. Make a function `is_orthogonal(L)` to test if the vectors in a list L are orthogonal. You'll want to check the output of `numpy.linalg.eig` and make L match that type.
3. Make a function `test_spectral_theorem(M)` that takes a matrix M and returns True if the matrix satisfies the spectral theorem and False if it does not. Here are a couple of suggestions off the top of my head that may help you,
 - Use the python function `all` to test if all the eigenvalues are real. Numpy has a function to test if a number is real.
 - I thought I had more than one. Wait, don't do the following,


```
if a and b:
    return True
else:
    return False
```

Just return `a and b`. It's shorter and more clear.

4. Finally, make a function `random_symmetric_spectral(N)` to test if N symmetric matrices all satisfy the spectral theorem. Randomize the size of each matrix to between 10 and 50.

Problem 2 (1 pt) Using the framework you set up in the previous problem, try testing a few non-symmetric matrices. Can you come up with a non-symmetric matrix that has real eigenvalues and orthogonal eigenvectors?