Eigenvalues and Eigenvectors

In this lesson, we will discuss how to find eigenvalues and eigenvectors of non-hermitian and hermitian matrices.

WE'LL COVER THE FOLLOWING

Eigenvalues and **eigenvectors** play a prominent role in the study of ordinary differential equations and in the physical sciences. One example that always comes to mind is Quantum Mechanics, which depends heavily on eigenvectors and eigenvalues.

Eigenvalues and eigenvectors are especially helpful in the process of transforming a given matrix into a diagonal matrix, which is easy to work with.

The eigenvalue problem for a matrix can be defined as follows:

$$Av_n = \lambda_n v_n$$

$$\text{where } \lambda_n \text{ is } n^{th} \text{ eigenvalue}$$

$$\text{and } v_n \text{ is } n^{th} \text{ eigenvector}$$

Eigenvalues

To compute the eigenvalues, we use the `eigvals` method with the matrix as the input argument. It returns an `ndarray` with the eigenvalues of the input matrix.

```
import numpy as np

A = np.array([[1, 5, 8], [2, 2, 6], [2, 5, 9]])
```

```
A = np.array([[1, 3, 8], [2, 2, 0], [2, 3, 9]])

print(np.linalg.eigvals(A))
```

Eigenvectors

To compute both eigenvectors and eigenvalues, we use the `eig` method. This method takes and `ndarray` as an input and returns two `ndarray`s. One contains the eigenvalues and the other contains eigenvectors. The eigenvalues are returned sorted.

```
import numpy as np

A = np.array([[1, 0, 0], [0, 2, 0], [0, 0, 9]])

evals, evecs = np.linalg.eig(A)

print("Eigenvalues", evals)
print("Eigenvectors", evecs)
```

`eig` returns normalized eigenvectors.

The eigenvectors corresponding to the n^{th} eigenvalue, stored in `evals[n]`, is the n^{th} column in `evecs`, i.e., `evecs[:,n]`.

> 💡 Eigenvalues of a diagonal matrix are the values on the diagonal.

Hermitian matrices

For hermitian matrices, we use the `eigh()` method. The eigenvalues returned here are sorted as well. It also uses a faster underlying algorithm that takes advantage of the fact that the matrix is symmetric. If you know that your matrix is symmetric, use this function.

> **Note:** `eigh()` doesn't check if your matrix is symmetric. By default, it just takes the lower triangular part of the matrix and assumes that the

upper triangular part is defined by the symmetry of the matrix.

`eig()` works for general matrices and therefore uses a slower algorithm. Let’s see an implementation of this below:

```
import numpy as np

a = np.array([[-1, 1-2j, 0],[1+2j, 0, 0-1j],[0, 1j, 1]], dtype=complex) # Hermitian matrix
evals, evecs = np.linalg.eigh(a)

print(a)
print("Eigenvalues", evals)
print("-----")
print("Eigenvectors", evecs)
```

In the next lesson, we will discuss various matrix operations.