# Eigenvalues

EE24BTECH11047 - Niketh Prakash Achanta

## 1 Introduction

### 1.1 Eigenvalues → What?

In linear algebra, an eigenvector or characteristic vector is a vector that has its direction unchanged (or reversed) by a given linear transformation. More precisely, an eigenvector, $\mathbf{v}$ is scaled by a constant factor, $\lambda$ when the linear transformation is applied to it:

$$A\mathbf{v} = \lambda\mathbf{v}$$

The corresponding eigenvalue, characteristic value, or characteristic root is the multiplying factor $\lambda$.

### 1.2 Choice of algorithm

To find eigenvalues of matrices, multiple algorithms have been designed with varying time complexities. A widely used option is the **QR** algorithm.

### 1.3 Working of algorithm

The **QR** algorithm involves the **QR** decomposition, where the given matrix is converted and denoted as the product of an orthogonal (**Q**) and an upper-triangular (**R**) matrix.

## 2 Comparison

The table below compares selected algorithms for computing eigenvalues, focusing on their descriptions and time complexities.

| **Algorithm** | **Description** | **Time Complexity** |
|---|---|---|
| Power Method | Iteratively finds the dominant eigenvalue and eigenvector. | $O(n^2)$ per iteration |
| QR Algorithm | Computes all eigenvalues using QR factorization. | $O(n^3)$ per iteration |
| Jacobi Method | Rotates the matrix to diagonal form, suitable for symmetric matrices. | $O(n^3)$ |

## 3 Implementation

### 3.1 QR decomposition

To apply QR decomposition, there are various methods. I have chosen Givens' rotations to decompose the initial matrix.

### *3.2 Givens' Rotations*

In the method of Givens Rotation, similar to Gram-Schmidt and Householder Transformation, we try to decompose each column vector in A to a set of linear combinations of orthogonal vectors in Q. We map the column vector to a set of orthogonal vectors by rotating it, instead of reflecting it.

### *3.3 Iterative QR decomposition*

Once the QR decomposition is applied once on the given matrix, we get the product QR. Then a matrix $A_1$ is defined as:

$$A_1 = RQ$$

The decomposition is then done again on this new matrix $A_1$, and this goes on till $A_k$ is found, where $k \to \infty$.

### *3.4 Convergence*

The diagonal entries of $A_k$ converge to the eigenvalues of matrix A.

### *3.5 Complexity*

The time complexity of the QR algorithm is calculated to be $O\left(n^3\right)$ where n is the order of the matrix.

### *3.6 Choice of programming language*

I have chosen "C" language to implement this algorithm.

## 4 Conclusion

In conclusion, QR algorithm works as an all-purpose algorithm, as:

- It works for all square matrices.
- It finds all eigenvalues (real and complex).
- When applied to symmetric matrices, the QR algorithm converges faster and directly computes real eigenvalues, leveraging the matrix's properties.