

A Study about QR Algorithm Using Givens Rotations

EE24BTECH11053 - S A Aravind Eswar

November 18, 2024

CONTENTS

1	Abstract	2
2	Introduction	2
3	Given's Rotations	2
3.1	Performing QR decomposition with Givens Rotation	3
3.2	Analysis of the Givens algorithm	4
3.3	Algorithm	4
4	QR iteration algorithm	4
4.1	Analysis of the QR algorithm	5
4.2	Hessenberg reduction	5
4.3	Shifts	5
4.3.1	Raleigh Shift	5
4.3.2	Wilkinson Shift	5
4.4	Algorithm	5
5	Extracting eigenvalues after performings the QR algorithm	6
6	Conclusion	6

1 ABSTRACT

Matrices are an important tool for performing calculations in a variety of applications. And the eigenvalues of the matrices have a wide variety of applications from AI and ML to Molecular Physics. There are multiple algorithms to compute the eigenvalues of a given matrix. We shall look into the QR Algorithm using Givens Rotations and discuss about its pros and cons and implement the algorithms.

2 INTRODUCTION

Matrices are an important tool for performing calculations in a variety of applications. And the eigenvalues of the matrices have a wide variety of applications from AI and ML to Molecular Physics. There are multiple algorithms to compute the eigenvalues of a given matrix. The most widely used algorithm for computing the eigenvalues is the QR algorithm or QR iteration algorithm. Other widely used algorithms include the Power iteration algorithm, Inverse iteration algorithm and so on. To perform the said QR decomposition for the QR algorithm, we are going to look into the method of Given's rotation. There are other algorithms that can perform QR decomposition such as the Gram-Schmidt algorithm and Householder transformations and so on. The reasons of choosing the said algorithms will be discussed in the later sections.

3 GIVEN'S ROTATIONS

Givens rotation is a rotation in a plane spanned by 2 coordinate axes in a \mathbb{R}^n space. Let a vector \mathbf{x} in \mathbb{R}^n to be rotated in the plane (i, j) by an angle of θ . Then,

$$\mathbf{x}_r = G(i, j, \theta)\mathbf{x} \quad (3.1)$$

is the vector produced after rotation and the matrix $G(i, j, \theta)$ is given by,

$$G(i, j, \theta) = \begin{pmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & & \vdots & & \vdots & & \vdots \\ 0 & \cdots & c & \cdots & s & \cdots & 0 \\ \vdots & & \vdots & & \vdots & & \vdots \\ 0 & \cdots & -s & \cdots & c & \cdots & 0 \\ \vdots & & \vdots & & \vdots & & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{pmatrix} \quad (3.2)$$

$$G_{kk} = 1 \text{ for } k \neq i, j \quad (3.3)$$

$$G_{kk} = c \text{ for } k = i, j \quad (3.4)$$

$$G_{ij} = -G_{ji} = s \quad (3.5)$$

$$\text{or } 0 \text{ otherwise} \quad (3.6)$$

where, The rotation can be scaled to \mathbb{C}^n by replacing G_{jj} with \bar{c} and G_{ji} with $-\bar{s}$

3.1 Performing QR decomposition with Givens Rotation

Let \mathbf{A} be a $n \times n$ matrix in \mathbb{C} . We can use Givens rotation to make exactly one of the coordinates of \mathbf{A} as 0.

Then,

$$\mathbf{A} = (A_{ij}), A_{ij} \in \mathbb{C} \quad (3.7)$$

$$(3.8)$$

Let,

$$\mathbf{G}(p, q) = \begin{pmatrix} 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & & \vdots \\ 0 & \cdots & c & s & \cdots & 0 \\ 0 & \cdots & -s & c & \cdots & 0 \\ \vdots & & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 1 \end{pmatrix} \quad (3.9)$$

$$G_{kk} = 1 \text{ for } k \neq i, j \quad (3.10)$$

$$G_{p,p} = \bar{G}_{p-1,p-1} = c \quad (3.11)$$

$$\bar{G}_{p-1,p} = -G_{p,p-1} = s \quad (3.12)$$

$$\text{or } 0 \text{ otherwise} \quad (3.13)$$

where,

$$r = \sqrt{\mathbf{A}_{p,q}^2 + \mathbf{A}_{p-1,q}^2} \quad (3.14)$$

$$c = \mathbf{A}_{p,q}/r \quad (3.15)$$

$$s = \mathbf{A}_{p-1,q}/r \quad (3.16)$$

Then,

$$\mathbf{A}_1 = \mathbf{G}(p, q)\mathbf{A} \quad (3.17)$$

will make $A_{p,q}$ zero.

Performing another rotation on $\mathbf{A}_{1,1}$ and so on,

$$\mathbf{G}(1, 2)\mathbf{G}(1, 3) \cdots \mathbf{G}(n, n-1)\mathbf{A} = \mathbf{R} \quad (3.18)$$

where \mathbf{R} is an Upper Triangular Matrix. Then let,

$$\mathbf{Q}^\top = \mathbf{G}(1, 2)\mathbf{G}(1, 3) \cdots \mathbf{G}(n, n-1) \quad (3.19)$$

$$\mathbf{Q} = (\mathbf{G}(1, 2)\mathbf{G}(1, 3) \cdots \mathbf{G}(n, n-1))^\top \quad (3.20)$$

$$\mathbf{Q} = \mathbf{G}^\top(n, n-1)\mathbf{G}^\top(n, n-1) \cdots \mathbf{G}^\top(1, 2) \quad (3.21)$$

Then,

$$\mathbf{A} = \mathbf{QR} \quad (3.22)$$

3.2 Analysis of the Givens algorithm

Givens rotation of a matrix has a property where it only modifies only 2 of the columns of a matrix. This makes it more numerically stable, unlike Gram-Schmidt which is not very stable numerically. This also comes with a perk of being more parallelizable than other algorithms like Householder transformations.

Givens Algorithm is not too efficient for a dense matrix as it doesn't have many zero entries to skip over. But it is more efficient when it comes to sparse matrix when compared to Householder, which is more efficient when it comes to dense matrix.

The time complexity of the Givens algorithm is $O(n^3)$, which is similar to other algorithms for QR decomposition.

3.3 Algorithm

```

1: Let  $\mathbf{A}_{m \times m}$ 
2:  $\mathbf{Q} \leftarrow \mathbf{I}_{n \times n}$ 
3:  $\mathbf{R} \leftarrow \mathbf{A}$ 
4: for  $i = 0, 1, \dots, n-1$  do
5:   for  $j = n-1, n-2, \dots, i+1$  do
6:     if  $|\mathbf{R}(j, i)| \neq 0$  then
7:       Generate  $\mathbf{G}(j, i)$ 
8:        $\mathbf{Q} \leftarrow \mathbf{Q}\mathbf{G}^T(j, i)$ 
9:        $\mathbf{R} \leftarrow \mathbf{G}(j, i)\mathbf{R}$ 
return  $\mathbf{Q}, \mathbf{R}$ 

```

4 QR ITERATION ALGORITHM

QR Iteration algorithm as discussed is a widely used algorithm. It works on the principle of converging a matrix to its Schur form and extracting eigenvalues from it.

$$\mathbf{A}_{k-1} = \mathbf{QR} \quad (4.1)$$

$$\mathbf{A}_k = \mathbf{RQ} \quad (4.2)$$

$$\mathbf{A}_k = \mathbf{Q}^T \mathbf{RQ} \quad (4.3)$$

Eventually \mathbf{A}_k will converge to the Schur form of \mathbf{A} , where is a $n \times n$ matrix in

4.1 Analysis of the QR algorithm

QR rotation theoretically is an infinitely iterating algorithm. But for it to converge to machine level precision, it takes it about $O(n^3)$ time to converge to the Schur form. The convergence rate can be accelerated but using techniques like Hessenberg reduction and Shifts.

4.2 Hessenberg reduction

Hessenberg matrix is where all the elements below the lower sub-diagonal is all zero.

We can write,

$$\mathbf{H}_0 = \mathbf{U}_0^\top \mathbf{A} \mathbf{U} \quad (4.4)$$

where \mathbf{H}_0 is Hessenberg matrix. We can show that both \mathbf{H}_0 and \mathbf{A} are identical and thus have the same eigenvalues.

Performing QR decomposition for the Hessenberg matrix is much faster ($O(n^2)$) as the number of calculations is much less. But Hessenberg transformations can only be used when $A_{i,j} \in \mathbb{R}$

4.3 Shifts

Let,

$$\mu \rightarrow \mathbb{R} \quad (4.5)$$

$$\mathbf{Q} \mathbf{R} = \mathbf{A}_{k-1} - \mu \mathbf{I} \quad (4.6)$$

$$\mathbf{A}_k = \mathbf{R} \mathbf{Q} + \mu \mathbf{I} \quad (4.7)$$

This is called shifting. There are 2 ways of selecting the shift value μ

4.3.1 Raleigh Shift: If \mathbf{A}_k is a Hessenberg matrix of size $n \times n$, then $\mu = \mathbf{A}_k(n, n)$

4.3.2 Wilkinson Shift: If \mathbf{A}_k is a Hessenberg matrix of size $n \times n$ then μ is determined based on the eigenvalues of bottom right most 2×2 matrix of \mathbf{A}_k (let a_k and b_k). If both a_k and b_k are real, then the smallest of $|a_k - A_k(n, n)|$ and $|b_k - A_k(n, n)|$ is taken. Else if both are conjugate pairs of each other, then $\mu = \mathbb{R}(a_k)$.

Shifting accelerates the process of convergence of the eigenvalues.

4.4 Algorithm

```

Hk ← U⊤ A U
for  $i = 1, 2, \dots$  do
    Compute  $\mu$ 
    Q R ← Hk −  $\mu \mathbf{I}$ 
    Hk ← R Q +  $\mu \mathbf{I}$ 
return Hk

```

5 EXTRACTING EIGENVALUES AFTER PERFORMING THE QR ALGORITHM

If \mathbf{A} has Complex eigenvalues, QR algorithm will not converge to a completely upper triangular matrix. But rather "blocks" of size 2×2 present along the diagonal. The eigenvalues of the blocks are also eigenvalues of \mathbf{A} .

We can compute it using the following algorithm

Let A_k

Let $eigs$ be array of eigenvalues

while $i < n$ **do**

if $i = \text{nor } (|A_k(i+1, i)| = 0)$ **then**

$eigs[i] \leftarrow A_k(i, i)$

else

 Find eigenvalues of the 2×2 spanning from $A_k(i, i)$ to $A_k(i+1, i+1)$ and let them be $e1$ and $e2$

$eigs[i] \leftarrow e1$

$eigs[i+1] \leftarrow e2$

$i \leftarrow i+1$

$i \leftarrow i+1$
return $eigs$

6 CONCLUSION

We discussed about how to find eigenvalues of a Real $n \times n$ matrix using the QR algorithm