Department Of Aerospace Engineering,
Indian Institute Of Technology Madras

AS2101 : Introduction to Aerospace Engineering

Report 7 : QR Decomposition

Pranit Zope
AE20B046

December 2, 2021

Contents

List of Figures

1 Aim

To decompose a square matrix A of order n into an orthogonal matrix Q and upper triangle matrix R and solve linear equations.

2 Theory

QR Decomposition is a method of decomposing a matrix A into two matrices Q and R, Q being an orthogonal matrix and R being an upper triangular matrix. We eill use **Eigenvalue Algorithm** to solve this problem.
This method only works for invertible matrices and is very useful in solveing linear equations of the form AX=B.3

2.1 Procedure for finding Q and R

Let us consider a matrix A given to us.

$$\mathbf{A} = [\mathbf{a}_1\mathbf{a}_2\mathbf{a}_3\mathbf{a}_4 \dots \mathbf{a_n}]$$

Now, we need to find Q which is orthogonal. So, we will use the Grahm-Schmidt Algorithm to find Q.
Thus,

$$\mathrm{q}_1 = \frac{\mathrm{a}_1}{\|\mathrm{a}_1\|}$$

To find q2 we need to take a2 and remove the component along q1 to make it perpendicular to q1, i.e orthogonal and then normalise.

$$\mathrm{q}_2 = \frac{\mathrm{a}_2 - < \mathrm{a}_2, \mathrm{q}_1 > \mathrm{q}_1}{\|\mathrm{a}_2 - < \mathrm{a}_2, \mathrm{q}_1 > \mathrm{q}_1\|}$$

Similarly, for q3;

$$\mathrm{q}_3 = \frac{\mathrm{a}_3 - < \mathrm{a}_3, \mathrm{q}_2 > \mathrm{q}_2 - < \mathrm{a}_3, \mathrm{q}_1 > \mathrm{q}_1}{\|\mathrm{a}_3 - < \mathrm{a}_3, \mathrm{q}_2 > \mathrm{q}_2 - < \mathrm{a}_3, \mathrm{q}_1 > \mathrm{q}_1\|}$$

and doing the same for q_n ;

$$\mathrm{q_n} = \frac{\mathrm{a_n} - \sum_{\mathrm{i}=1}^{\mathrm{n}-1} < \mathrm{a}_3, \mathrm{q_i} > \mathrm{q_i}}{\left\|\mathrm{a_n} - \sum_{\mathrm{i}=1}^{\mathrm{n}-1} < \mathrm{a}_3, \mathrm{q_i} > \mathrm{q_i}\right\|}$$

Now that we have Q

$$\mathbf{Q} = [\mathbf{q}_1\mathbf{q}_2\mathbf{q}_3\mathbf{q}_4 \dots \mathbf{q_n}]$$

Now we will write a_i in terms of q_i;

$$a_i = \sum_{j=1}^{j=i} < a_i, q_j > q_j$$

For factorisation, the coefficients of qi must form column vectors in the matrix R.
Hence we can say that R will be:

$$\begin{bmatrix} < a_1, q_1 > & < a_2, q_1 > & \dots & \dots & < a_n, q_1 > \\ < a_1, q_2 > & < a_2, q_2 > & \dots & \dots & < a_n, q_2 > \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ < a_n, q_1 > & < a_n, q_2 > & \dots & \dots & < a_n, q_n > \end{bmatrix}$$

Since there is no component of a_i about q_j, $j > i$, R takes the form of upper triangular matrix as others take the value 0.

$$R = \begin{bmatrix} < a_1, q_1 > & < a_2, q_1 > & \dots & \dots & < a_n, q_1 > \\ 0 & < a_2, q_2 > & \dots & \dots & < a_n, q_2 > \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \dots & < a_n, q_n > \end{bmatrix}$$

2.2 Solving Linear Equations using this

Lets say we have $AX = B$ linear equaiton. Using the QR Decompostion, we have $A = QR$ Thus,

$$QRX = B$$

$$RX = Q^{-1}B$$

Since Q is orthogonal, $Q^{-1} = Q^T$, Thus we have

$$RX = Q^T B$$

Now, since R is a upper triangular matrix, we can solve this by back substitution -

$$\begin{bmatrix} \sum_{i=1}^{n} r_{1,i}x_i \\ .. \\ .. \\ r_{(n-1)n}x_n + r_{n-1,n-1}x_{n-1} \\ r_{nn}x_n \end{bmatrix} = \begin{bmatrix} k_1 \\ .. \\ .. \\ k_{n-1} \\ k_n \end{bmatrix}$$

From here, we obtain $x_1, x_2, x_3, ..., x_n$ without any complicated solving.

3 Verification of the Method

Previously in this course, we had linearly regressed a data to obtain a best fitting line. We will use the same dataset to validate the QR- Decomposition method.

For "data2.txt", using linear regression, we got the slope as

```
m = 2.0056244629115776
c = 1.3809824773861692
```

Now, using the QR method, we get the following output :

```
x1 =

   2.0056
   1.3810
```

The former line being the slope and later the intercept.
Thus, we can say that the result we got in both the methods is same.

4 Efficiency comparision of several Methods

We will now compare several method of soving the linear equations using a graph

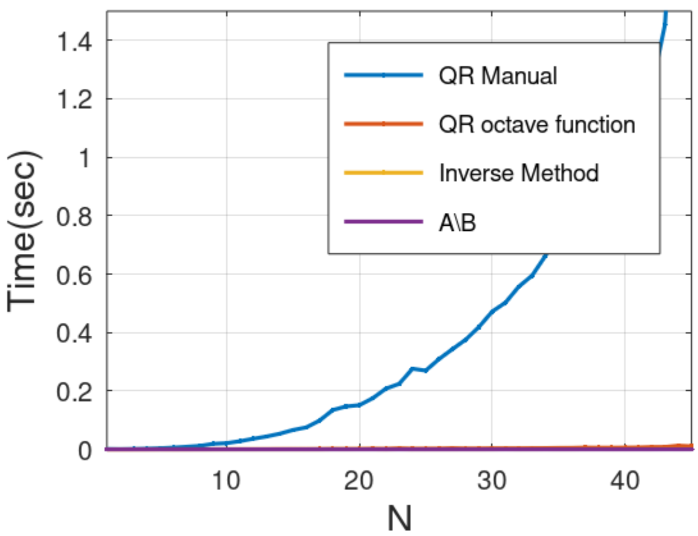

Figure 1: Comparision of time taken in different methods

Also, we can compare the time taken at each steap and the total time :

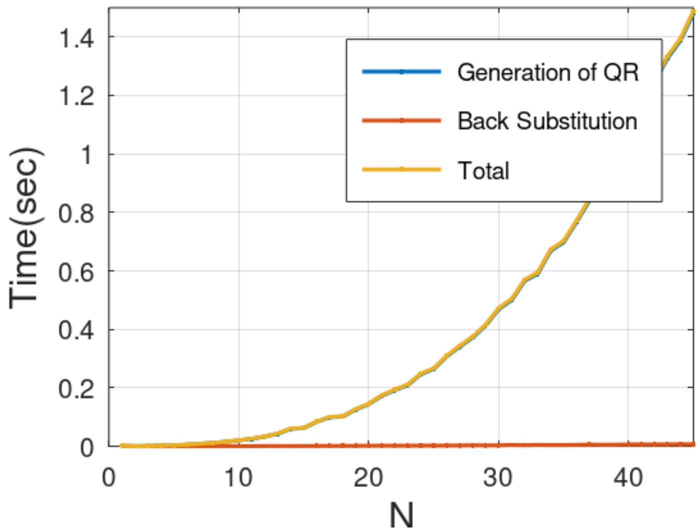

Figure 2: Time taken at each step and the total time

5 Result and Conclusion

We can see that the time taken in the Generation of QR is a lot. Hence, this method is simpler but takes a lot of time for the computer.
A possible reason for this is that the algorithm is very bulky as it involves calculating multiple $n \times n$ matrices.
Thus we can conclude that the method is correct, works fine. Also, it is not sufficient for a computer program since the time of computation is very high.