QR Decomposition

Reva Dhillon
AE21B108

October 15, 2022

1 Introduction

This report discusses the algorithm and efficiency of QR decomposition.
It compares the method with Octave's in-built QR code, the function $inv(A) * b$ and the function $A \backslash b$ with respect to L_∞ error and computational time.

2 QR Decomposition

In linear algebra, a QR decomposition, also known as a QR factorization or QU factorization, is a decomposition of a matrix A into a product $A = QR$ of an orthonormal matrix Q and an upper triangular matrix R.
Here, QR decomposition is used to solve the system of linear equations $Ax = b$ where we take A as invertible to obtain a unique solution.
The following steps are followed:

$$Ax = b$$

$$QRx = b$$

$$Q^{-1}QRx = Q^{-1}b$$

$$I_n Rx = Q^T b$$

$$Rx = c$$

where $c = Q^T b$ which is again a vector.
As Q is orthogonal, $Q^{-1} = Q^T$.
As R is an upper triangular matrix, the computations involved in obtaining x have been significantly simplified. We can find $x_n = c_n / R_{nn}$ and sequentially back-substitute values of x_i to obtain x.
Now the only task is obtaining Q and R.

2.1 Gram-Schmidt orthogonalization:

Our objective is to find Q which is an orthonormal matrix such that we can express A in terms of Q. Therefore, we construct Q using the columns of A by applying the Gram - Schmidt process.
General steps:

- $A = \begin{bmatrix} a_1 & a_2 & . & . & . & a_n \end{bmatrix}$.
- $Q = \begin{bmatrix} q_1 & q_2 & . & . & . & q_n \end{bmatrix}$.
- $\begin{matrix} \langle u & v \rangle \end{matrix} = u^T v$.
- Let $u_1 = a_1$.

- $$u_2 = a_2 - \frac{\langle u_1 \quad a_2 \rangle}{\langle u_1 \quad u_1 \rangle} u_1$$
  .

- $$u_k = a_k - \sum_{i=1}^{k-1} \frac{\langle u_i \quad a_k \rangle}{\langle u_i \quad u_i \rangle} u_i$$
  .

- Constructing u_k for $k = 1, 2, ..., n$ gives us an orthogonal system of vectors from the columns of A.

- As Q is orthonormal, we want an orthogonal system of vectors obtained from the columns of A such that each vector has unit magnitude. Therefore, $q_i = \frac{u_i}{||u_i||}$ for $i = 1, 2, ..., n$. As each q_i represents a column of Q, we can now construct Q.

2.2 Derivation of R:

Our objective is to find R. We begin by simplifying and rearranging the equation obtained for u_k.

$$u_k = a_k - \sum_{i=1}^{k-1} \frac{\langle u_i \quad a_k \rangle}{\langle u_i \quad u_i \rangle} u_i$$

We obtain the following results:

- $$q_1 = \frac{a_1}{||a_1||}$$

- Therefore,
  $$a_1 = q_1 ||a_1||$$

- Solving and simplifying,
  $$a_2 = \langle q_1 \quad a_2 \rangle q_1 + q_2 ||a_2||$$
  $$a_3 = \langle q_1 \quad a_3 \rangle q_1 + \langle q_2 \quad a_3 \rangle q_2 + q_3 ||a_3||$$
  $$a_k = (\sum_{i=1}^{k-1} \langle q_i \quad a_k \rangle q_i) + q_k ||a_k||$$

- Writing this in matrix form:

$$\begin{bmatrix} a_1 & a_2 & . & . & . & a_n \end{bmatrix} = \begin{bmatrix} q_1 & q_2 & . & . & . & q_n \end{bmatrix} \begin{bmatrix} ||a_1|| & \langle q_1 \quad a_2 \rangle & . & . & . & . & \langle q_1 \quad a_n \rangle \\ 0 & ||a_2|| & . & . & . & . & \langle q_2 \quad a_n \rangle \\ 0 & 0 & . & . & . & . & \langle q_3 \quad a_n \rangle \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ 0 & 0 & . & . & . & . & ||a_n|| \end{bmatrix}$$

- We know

$$Q = \begin{bmatrix} q_1 & q_2 & . & . & . & q_n \end{bmatrix}$$

.

- Therefore,

$$R = \begin{bmatrix} ||a_1|| & \langle q_1 \quad a_2 \rangle & . & . & . & . & \langle q_1 \quad a_n \rangle \\ 0 & ||a_2|| & . & . & . & . & \langle q_2 \quad a_n \rangle \\ 0 & 0 & . & . & . & . & \langle q_3 \quad a_n \rangle \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ 0 & 0 & . & . & . & . & ||a_n|| \end{bmatrix}$$

- Therefore, we have all the required parameters to solve the system of linear equations.

3 Comparative Analysis:

This section elucidates the following three comparative studies:

- The L_∞ error between the solutions from the QR code(self-written) and in-built $A\backslash b$ octave operation as we vary matrix size from 1 upto 100.
- Variation in computational time as we vary matrix size for the following 4 operations:
  1. Self-written QR decomposition.
  2. Octave's in-built QR code.
  3. $inv(A) * b$ operation in octave.
  4. $A\backslash b$ operation in octave.
- Computational time taken by the self-written QR code as we vary matrix size.

3.1 L_∞ error:

L_∞ norm gives the absolute value of that component of a vector that has the largest magnitude.
Therefore, L_∞ error between two vectors of order n, x and $\hat{x}$ is defined as:

$$L_\infty = max(|x_i - \hat{x}_i|) \quad \forall i = 1, 2, ..., n \tag{1}$$

L_∞ error plotted here is between the solutions from the QR code(self-written) and in-built $A\backslash b$ octave operation for matrix size upto 100.
This backslash operator in octave is used for left matrix division. Left matrix division takes the inverse of the first matrix(on the left of the backslash) and multiplies it to the second matrix(on the right of the backslash).
From the graph shown below we can conclude that the variation in error fluctuates, with L_∞ error increasing as n increases. However, L_∞ error has an order of magnitude less than 10^{-12}. That is, the error is very small. Hence, we can infer that the self-written QR code approximates the solution given by $A\backslash b$ to a high degree of accuracy.

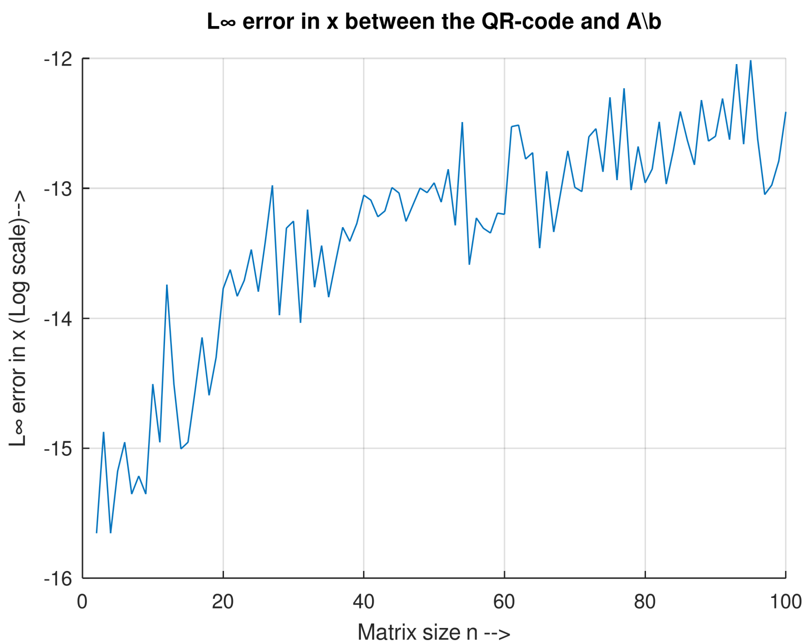

Figure 1: L_∞ error (Log scale) against Matrix size n.

3.2 Timing - I:

This subsection discusses the computational time taken by the following operations as we vary the matrix size:

1. Self-written QR decomposition.

2. Octave's in-built QR code.

3. $inv(A) * b$ operation in octave.

4. $A\backslash b$ operation in octave.

The following figure shows the computation time(in seconds) on the log scale.
We observe that for each method there is an overall increase in computational time as we increase the matrix size n, albeit erratic.
The $A\backslash b$ operation requires the least computational time overall while the self-written QR code takes longest.
The order of magnitude of the computational time remains the same with some variation as matrix size increases over 60 for Octave's in-built QR code, $inv(A) * b$ and $A\backslash b$ operations in octave.

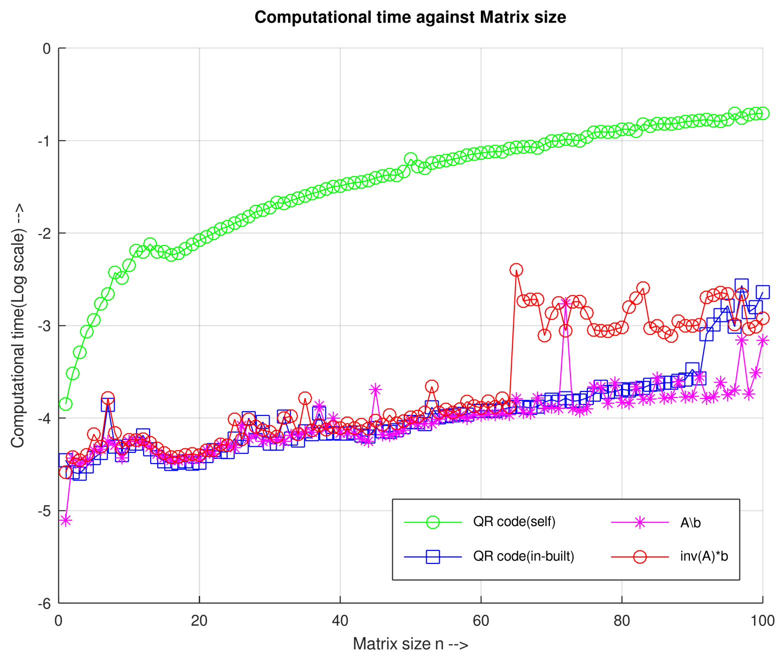

Figure 2: Computational time(Log scale) against Matrix size n(4 operations).

3.3 Timing - II:

This subsection discusses the computational time taken by the self-written QR code as we vary the matrix size.
The following plot graphs the time taken by the algorithm to perform the Q and R computations, the time taken to calculate x and the total time taken (in seconds).
The time taken to calculate x is significantly smaller than the time taken to compute Q and R.
The individual timings and the total time taken by the algorithm show a net increase as we increase the matrix size n.

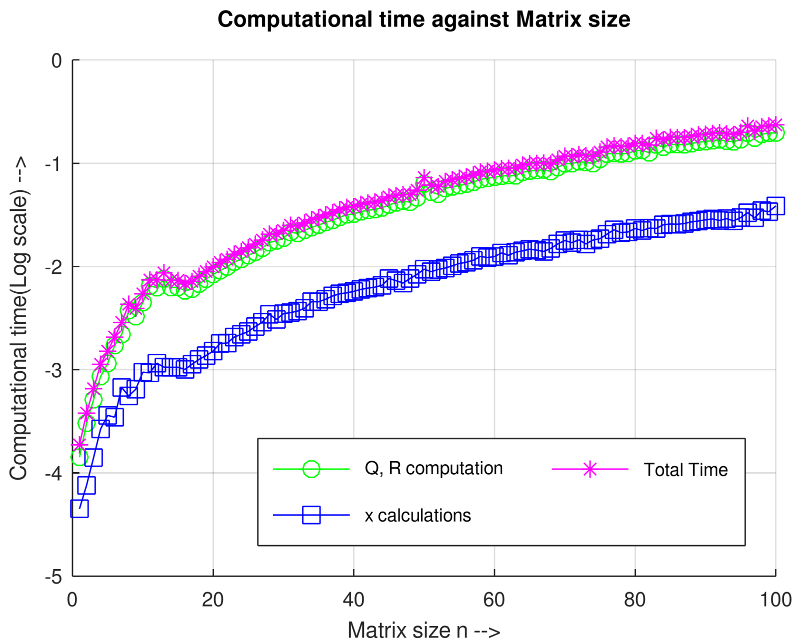

Figure 3: Computational time(Log scale) against Matrix size n.

4 Conclusion

In conclusion, the QR decomposition method for finding a solution to a system of linear equations is extremely useful as it breaks down a complex problem into simple substitutions using an upper triangular matrix.
This method is especially useful if we require x for different vectors b with the same A.
It is widely used in various fields of mathematics and engineering.