Proceedings of the Federated Conference on
Computer Science and Information Systems pp. 561–565

ISBN 978-83-60810-51-4

Analysis and Comparison of QR Decomposition Algorithm in Some Types of Matrix

A. S. Nugraha
Faculty of Computer Science
University of Indonesia
Depok, Indonesia 16424
Email: anggha.satya11@ui.ac.id; stlasn@yahoo.com

T. Basaruddin
Faculty of Computer Science
University of Indonesia
Depok, Indonesia 16424
Email: chan@cs.ui.ac.id

***Abstract*—QR decomposition of matrix is one of the important problems in the field of matrix theory. Besides, there are also so many extensive applications that using QR decomposition. Because of that, there are many researchers have been studying about algorithm for this decomposition. Two of those researchers are Feng Tianxiang and Liu Hongxia. In their paper, they proposed new algorithm to make QR decomposition with the elementary operation that is elementary row operations. This paper gives review of their paper, the analysis and numerical experiment using their algorithm, comparison with other existing algorithms and also suggestion for using other existing better algorithm that also has same features with theirs. Beside of them, we also compare all of these algorithms for some types of matrix. The result can be seen at this paper also.**

I. Introduction

DECOMPOSITION is one of the important subjects in matrix analysis. Decomposition in this case means how we divide a certain matrix into two or more matrices with certain characteristic. There are numerous of objectives when we decompose a matrix, i.e. for maximizing storage optimization, carrying out parallel computation, simplifying problem, etc. One of the famous techniques for conducting the decomposition is QR decomposition. QR decomposition is the decomposition of a matrix (A) into an orthogonal matrix (Q) and an upper triangular matrix (R). Beside being used in the field of theory, QR decomposition can also be used for many practical issues. In signal detection, algorithm using QR decomposition for VBLAST-OFDM systems was studied by Zhong et.al [ZH07]. In another area like wireless applications, digital predistortion (DPD) and etc., QR decomposition is also being used.

However, its decomposition process is usually very complex. This reason becomes a trigger for many researchers for finding new simpler algorithm. Two of them are Feng Tianxiang and Liu Hongxia [FL09]. In [FL09], they presented new algorithm for finding QR decomposition for square and full column rank matrix. For finding the decomposition, they use elementary operation that is elementary row operations. There are at least 2 excesses of full rank matrix, i.e. it has unique QR decomposition and also has unique solution in term of least square problem.

The rest of this paper is organized as follows. Section II shows the basic theory of QR decomposition and the algorithm that was presented in [FL09]. The numerical experiment is shown in Section III while the analysis of their algorithm will be shown in Section IV. Section V gives other suggested algorithm that has better result and also the comparison of those algorithms for some types of matrix. The conclusion of this paper is given in Section VI.

II. Basic Theory and Algorithm

A. Basic Theory

This subsection shows some basic theories that are used for finding the algorithm in [FL09].

Theorem 1: [FL09] If $A \in R^{n\times n}$ is a full column rank matrix, then A^TA is a symmetric positive definite matrix and has unique triangular decomposition $A^TA = LDL^T$ where L is a lower triangular matrix with all diagonal elements are 1 and D is a diagonal matrix with positive diagonal elements.

Theorem 2: [FL09] If $A \in R^{n\times n}$ is a full column rank matrix, then A has QR decomposition $A = QR$ where Q=$A\left(L^{-1}\right)^T D^{-1/2}$ has orthonormal columns and $R = D^{1/2}L^T$ is upper triangular matrix.

Because A is full column rank matrix, then the QR decomposition for A is unique [BP92].

B. The Algorithm

The complete algorithm that proposed by Feng and Liu can be seen in [FL09]. Overall, this algorithm just doing upper triangularization for the composite matrices $A^T \times A$ and $A^T \times \left(A^TA \mid A^T\right)$. This process using elementary row operation. In [FL09], Feng and Liu stated that some advantages from their algorithm are computationally simpler, more elementary, and clearer computational complexity.

Beside those advantages, they were also compare their algorithm with other exist algorithm that is Householder transformation. They stated that it has lower accuracy than the Householder transformation method. However, they did not give their numerical experiment to show that. They also noted in their paper that the algorithm would be used more flexibly to solve some practical problems that need QR decomposition method. It has been known that in practical problems, almost for all problems, it needs very big matrix such as for data storage, 3D image and etc. Unfortunately, from the numerical experiment below, the algorithm in [FL09] have a good chance (just in term of time) just for small matrix whose size not bigger than 15×15.

III. Numerical Experiment

Before doing the numerical experiment, there is simplification in the algorithm without changes the concept. The simplification just for doing the upper triangularization and can be seen in Algorithm 3 below.

Algorithm 3: Simplification of upper triangularization.

$k = 0$
while $k < n - 1$ **do**
  $i = k$
  while $i < n$ **do**
    $i = i + 1$
    temp= $A[i,k] / A[k,k]$
    $j = k - 1$
    while $j < 2 \times n$ **do**
      $A[i,j] = A[i,j]$ −temp× $A[k,j]$
    end while
  end while
end while

The numerical experiment here is done with Matlab R2010b and worked at computer (notebook) with Processor Intel® Core(TM) i5 CPU M460 @ 2.53GHz and RAM 2 GB. Because Feng and Liu compare their algorithm with Householder transformation algorithm, we also compare their performance in term of time and accuracy. The result from the experiment can be seen in Fig. 1 below. The experiment described in Fig. 1 was done for matrix with maximum size 600×600 and repeated for 4 times. The time and error was gotten by getting mean of these loops.

It can be seen from Fig. 1 that the algorithm proposed by Feng and Liu is worse than Householder in term of time and accuracy. This is just the same like what they stated in [FL09]. Fig. 2 shows that their algorithm is "better" than Householder just for matrix whose size not bigger than 15×15. though it is just better in time because the accuracy is still not. The experiment for this small size matrix was repeated 6 times The detail of analysis can be seen in Section IV.

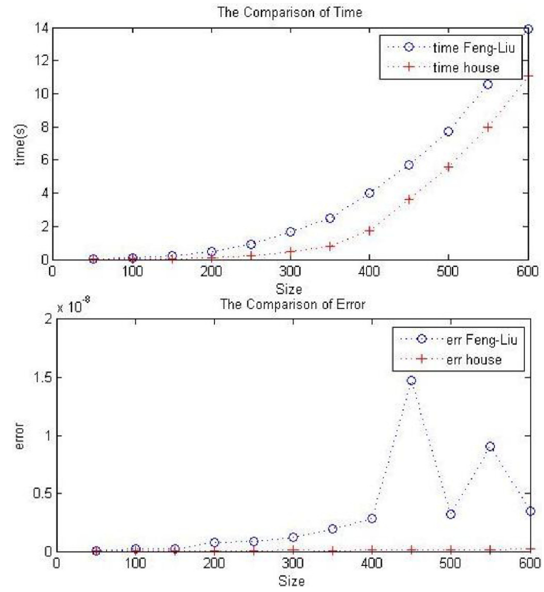

Fig. 1. Comparison time and error for Feng-Liu and Householder algorithm.

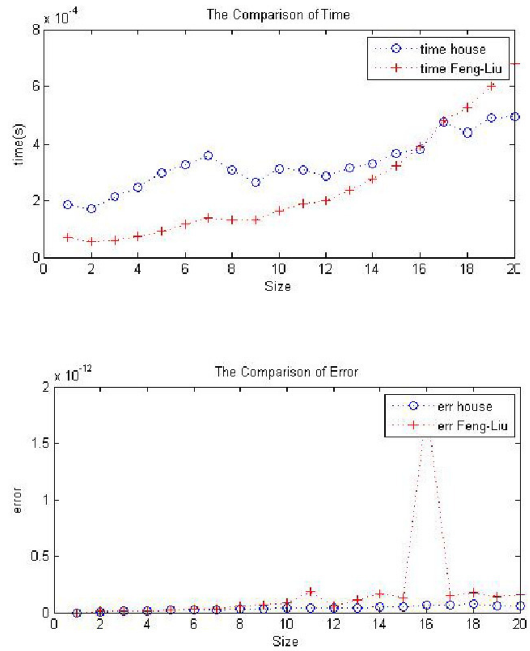

Fig. 2. Comparison time and error for Feng-Liu and Householder algorithm for small size matrix

IV. The Analysis

A. Analysis of Flopping Time

For counting the flopping time, there are 3 steps in the algorithm that must be checked i.e. the multiplication of A and A^T step, upper triangularization step and replacement the j-th row of B by the $B_{jj}^{-1/2}$ times of itself. It is obvious that the multiplication of matrix A and A^T requires $2n^3 - n^2$ flops where n is the size of matrix A.

For upper triangularization step, notice the algorithm in Fig. 3.1. There is an assumption here that each of division or taking square root, it needs 2 flops. The outermost loop executes $n-1$times as the counter k runs from 1 to $n-1$. For each of these loops, the middle loop executes $n - k$ times. For each middle loop, 2 flops are executed (for the division). The innermost loop executes $2n-k+1$ times as the counter j runs from k to $2n$ where in each of it encounters 2 flops. Collecting all of the multiples, the following total number of flops is

$$\sum_{k=1}^{n-1} (n-k)\left\{2 + 2(2n-k+1)\right\}. \tag{1}$$

For solving Equation 1, remember the formulas $\sum_{k=1}^{n} k = \frac{1}{2}n^2 + \frac{1}{2}n$ and $\sum_{k=1}^{n} k^2 = \frac{1}{3}n^3 + \frac{1}{2}n^2 + \frac{1}{6}n$. Using this two formulas, Equation 1 can be solved so that flopping time for upper triangularization step is $\frac{5}{3}n^3 - \frac{5}{3}n$. For the row's replacement, it requires $8n^2$ flops.

By adding all the flops above, the flopping time needed by the algorithm is $\frac{11}{3}n^3 + 7n^2 - \frac{5}{3}n$. However, this flopping time is larger than flopping time needed by Householder transformation. It has been known that for square matrix $n \times n$, if the matrix Q is required, Householder needs $\frac{4}{3}n^3$ flops.

B. Accuracy of The Algorithm

The main step in this algorithm is how the matrices $\left(A^T A \mid A^T\right)$ are transformed into upper triangular form. Feng and Liu use elementary row operation to make this LU factorization. Theorem 4 below quantifies the roundoff errors associated with the computed upper triangular form.

Theorem 4: [Hig96] Assume that A is $m \times n$ matrix. The computed triangular matrices $\widehat{L}$ and $\widehat{Q}$ satisfy $\widehat{L}\widehat{U} = A + \Delta A$ where $\left|\Delta A\right| \leq 3\epsilon\left(n-1\right)\left(\left|A\right| + \left|\widehat{L}\right|\left|\widehat{U}\right|\right) + O\left(\epsilon\right)$.

Proof: For proofing this theorem, we use induction on n. This theorem obviously hold for $n = 1$ because $\left|\Delta A\right| \leq O\left(\epsilon\right)$. Assume it holds for all $(n-1) \times (n-1)$ matrices. Take $A = \begin{bmatrix} \alpha & \omega^T \\ \upsilon & \beta \end{bmatrix}$ is $(n-1) \times (n-1)$ matrices, $\widehat{z} = fl\left(\frac{\upsilon}{\alpha}\right)$ and $\widehat{A_1} = fl\left(\beta - \hat{z}\omega^T\right)$. Therefore, we have $\widehat{z} = \frac{1}{\alpha}\upsilon + f, f \leq \epsilon\frac{|\upsilon|}{|\alpha|} + O\left(\epsilon\right)$ and

$$\widehat{A_1} = \left(\beta - \widehat{z}\omega^T\right) + F, F \leq 2\epsilon\left(\left|\beta\right| + \left|\widehat{z}\right|\left|\omega^T\right|\right) + O\left(\epsilon\right). \quad (2)$$

Now, we are going to do LU decomposition for $\widehat{A_1}$. Because $\widehat{A_1}$ is $(n-1) \times (n-1)$ matrices, the assumption can be used so that the decomposition of $\widehat{A_1}$ satisfy

$$\widehat{L_1}\widehat{U_1} = \widehat{A_1} + \Delta A_1. \quad (3)$$

$$\left|\Delta A_1\right| \leq 3\epsilon\left(n-2\right)\left(\left|\widehat{A_1}\right| + \left|\widehat{L_1}\right|\left|\widehat{U_1}\right|\right) + O\left(\epsilon\right). \quad (4)$$

Thus, $\widehat{L}\widehat{U} \approx \begin{bmatrix} 1 & 0 \\ \widehat{z} & \widehat{L_1} \end{bmatrix}\begin{bmatrix} \alpha & \omega^T \\ 0 & \widehat{U_1} \end{bmatrix} = A + \begin{bmatrix} 0 & 0 \\ \alpha f & H_1 + F \end{bmatrix} \approx A + \Delta A$. From Equation 2, it follows that $\left|\widehat{A_1}\right| \leq (1+2\epsilon)\left(\left|\beta\right| + \left|\widehat{z}\right|\left|\omega^T\right|\right) + O\left(\epsilon\right)$, and by using Equation 3 and Equation 4 we have $\left|\Delta A_1\right| + F \leq 3\epsilon\left(n-1\right)\left(\left|\beta\right| + \left|\widehat{z}\right|\left|\omega^T\right| + \left|\widehat{L_1}\right|\left|\widehat{U_1}\right|\right) + O\left(\epsilon\right)$. Because of $\left|\alpha f\right| \leq \epsilon\left|\upsilon\right|$, then we get $\left|\Delta A\right| \leq 3\epsilon\left(n-1\right)\left(\begin{bmatrix} |\alpha| & |\omega^T| \\ |\upsilon| & |\beta| \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ |\widehat{z}| & |\widehat{L_1}| \end{bmatrix}\begin{bmatrix} |\alpha| & |\omega^T| \\ 0 & |\widehat{U_1}| \end{bmatrix}\right) + O\left(\epsilon\right)$. ■

V. Other Better Algorithm and The Comparison

It can be seen from the numerical experiment in Section III that the algorithm proposed by Feng and Liu in [FL09] is worse than other existing algorithm that is Householder transformation. This result stimulates the writer to make some improvement for their algorithm. Because the main step is the upper triangularization step, then the improvement will be done in there. This improvement just changes the way to make upper triangular form, from with elementary row operation into with the Algorithm 5. The idea of this improvement is taken from [Hig96]. The improvement is like in Algorithm 5.

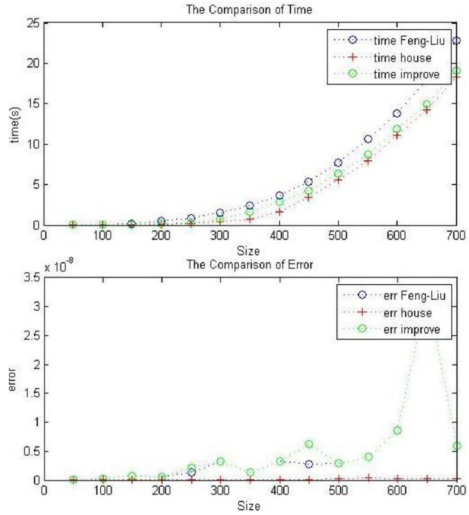

Fig. 3. Comparison time and error Feng-Liu, Householder and the improved algorithm

Algorithm 5: Improvement for Upper Triangularization Step.

while $((k \leq m)\,\&\&\,(B\left[k,k\right] \neq 0))$ **do**
  temp$[k+1:m] = B\left[k+1:m,k\right]/B\left[k,k\right]$;
  $B\left[k+1:m,:\right] = B\left[k+1:m,:\right] -$(temp$\left[k+1:m\right])' \times B\left[k,:\right]$;
  $k = k+1$;
end while

The result from the experiment with this improvement can be seen in Fig. 3. It can be seen from that figure, the improvement improves the time needed for doing QR decomposition, but not the accuracy. Nevertheless, Householder transformation is still better than them.

Because it still weaker than Householder, we try to find another existing algorithm that can be stronger than Householder in term of time and accuracy. Feng and Liu in [FL09] stated that their algorithm has features of simpler calculation and clearer computational complexity. Therefore, the better algorithm must have same features to can be compared.

This existing algorithm is modified Gram-Schmidt (MGS). Bjorck and Paige in [BP92] conclude that MGS method is equivalent, both mathematically and numerically, to Householder for some cases. The MGS algorithm is given in Algorithm 6.

Algorithm 6: MGS algorithm.

$k = 0$
while $k < n$ **do**
  $k = k + 1$

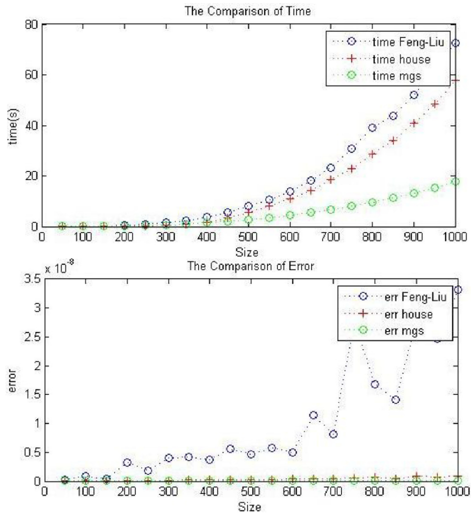

Fig. 4. Comparison time and error between Feng-Liu, Householder and MGS algorithm

$R[k,k] = |A[1:m,k]|$
$Q[1:m,k] = A[1:m,k]/R[k,k]$
$j = k$
while $j < n$ **do**
  $j = j + 1$
  $R[k,j] = Q[1:m,k]' \times A[1:m,j]$
  $A[1:m,j] = A[1:m,j] - Q[1:m,k] \times R[k,j]$
end while
end while

From that algorithm, it is clear that MGS also have features of simpler calculation and clearer complexity. Thus, we compare the algorithm in [FL09] with Householder and also MGS method. The result of this comparison can be seen at Figure 4.

It can be seen from Fig. 4 above that time complexity needed by MGS method is better than Householder and also the accuracy is not worse than Householder. Ref [Bjo67] has shown that MGS produces a computed matrix $\widehat{Q}$ in QR decomposition that satisfies $\widehat{Q}^T\widehat{Q} = I + \triangle I_M, \|\triangle I_M\| \approx \epsilon\kappa_2(A)$ whereas the corresponding result for Householder transformation is $\widehat{Q}^T\widehat{Q} = I + \triangle I_H, \|\triangle I_H\| \approx \epsilon$. In our case ($A$ is full column rank matrix), $\kappa_2(A)$ is very small because the columns of A are independent. From the definition of $\kappa_2(A)$ in our case, if orthonormality is critical, then MGS should be suggested used. Moreover, [Hig96] mentions that the computed matrix $\widehat{Q}$ and $\widehat{R}$ produced by MGS satisfies $\|A - \widehat{Q}\widehat{R}\| \approx \epsilon|A|$ and there exists a Q with perfectly orthonormal columns such that $\|A - Q\widehat{R}\| \approx \epsilon|A|$.

Beside the comparison with Householder and MGS algorithm, we also compare all of these algorithms for some types of matrix. We have compared all of them for some size in previous section. All of the above experiment were done for matrix whose element are integer numbers. For any real numbers, result of the experiment can be seen an Fig. 5. It can be seen that the Feng and Liu's algorithm is still worse than the others.

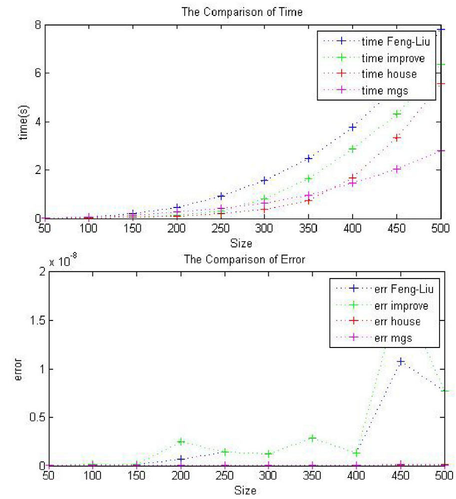

Fig. 5. Comparison time and error between Feng-Liu, improved, Householder and MGS algorithm for small matrix

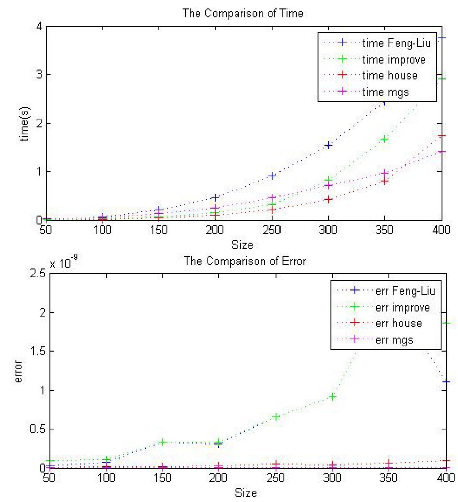

Fig. 6. Comparison time and error between Feng-Liu, improved, Householder and MGS algorithm for sparse matrix

Until now, we have considered about full matrix. But how about sparse matrix? As we now, sparse matrix is very useful in many kind of applications, i.e. image processing, computer graphics, etc. Figure 6 tells about the result. This experiment was repeated for 5 times. Unfortunately, Feng and Liu's algorithm is still worse than the others. The experiment results the same also for matrix whose has magic number.

VI. Concluding Remark

The QR decomposition is often used for counting the eigen values from giant matrix or for solving the least square problem. Besides, there also many practical issues that need QR decomposition to be solved. Therefore, the QR decomposition is not only an important problem in matrix theory, but also has an extensive application prospect. This thing becomes a trigger for many researchers for doing experiment in QR decomposition including Feng Tianxiang and Liu Hongxia. In [FL09], they presented new algorithm for finding $\widehat{Q}$ and $\widehat{R}$ so that $A=\widehat{Q}\widehat{R}$ using elementary operation. Unfortunately, they did not give the numerical analysis and experiment of theirs in [FL09]. This paper gives the analysis of their algorithm and tries to compare their algorithm with other basic algorithm for finding QR decomposition i.e. Householder transformation. The result is, the algorithm in [FL09] is worse than Householder. Without changing their idea, this paper gives improvement for improving their algorithm. The comparison result with this algorithm can be seen in Fig. 3. Because the result is still worse than Householder, then we try to find another existing algorithm that has same features like algorithm proposed by Feng and Liu, i.e. simpler calculation and clearer complexity. The chosen algorithm is modified Gram-Schmidt (MGS). The experiment for comparing their algorithm with Householder and also MGS method can be seen in Fig. 4. The result is MGS method is better than the algorithm in [FL09] and also than Householder if orthonormality is important. After all of the comparison, we are still try to compare all of these algorithms for some types of matrix: matrix with real numbers in its element, sparse matrix, and also matrix with magic number. The result is, Feng and Liu's algorithm is still worse than the others.

REFERENCES

[FL09] Feng Tianxiang and Liu Hongxia, "The Computer Realization of the QR Decomposition on Matrices with Full Column Rank", *Proc. IEEE Int. Conf. on Computer Intelligence and Security.* **NO** (2009), 76 - 79.

[ZH07] Zhong-tang, HE et.al., "Research of QR Factorization Based Signal Detection Algorithms", *Journal of Nanjing University of Posts and Telecommunications (Natural Science).* **27** (2007), 34-38.

[BP92] A. Bjorck and C.C. Paige, "Loss and recapture of orthogonality in the modified Gram-Schmidt algorithm", *SIAM J. Matrix Anal. Appl.* **13** (1992), 176 - 190.

[Hig96] N.J. Higham, "Accuracy and Stability of Numerical Algorithms", SIAM Publications, Philadelphia, PA, 1996.

[Bjo67] A. Bjorck, "Solving linear least squares problems by Gram-Schmidt orthogonalization", *BIT.* **7** (1967), 1-21.