Proceedings of the Conference on Mathematical Foundations of Informatics
MFOI2020, January 12-16, 2021, Kyiv, Ukraine

Quick Recursive QR Decomposition

Gennadi Malaschonok, Andriy Ivashkevich

Abstract

A description of a recursive algorithm for QR-decomposition of a matrix based on Givens rotation matrices is given. Another version of the block-recursive algorithm is obtained by replacing Givens rotations with the Householder transform. These block-recursive algorithms have the same complexity as the matrix multiplication algorithm.

Keywords: recursive QR decomposition, block-recursive matrix algorithms, factorization of matrices, Givens rotation, Householder transformation.

1 Introduction

In connection with the new concept of creating recursive parallel algorithms with dynamic control (see [1]-[3]), we began to look for a recursive algorithm for QR decomposition. When did we get the algebraic equations of such an algorithm, thanks to the personal communication of Prof. Alexandr Tiskin and his publication [3], we found A. Schönhage paper [4] in German, which suggested a similar algorithm, and which was quoted very little in recent years. This convinced us of the correctness of this approach and the whole concept of revising and searching for recursive matrix algorithms.

To simplify the description of the algorithm, we restrict ourselves to the decomposition of matrices over real numbers. At the same time, we note that these matrix factorization algorithms are closed in the sets of algebraic, real and complex numbers and are not closed in the set of rational numbers.

2 Givens algorithm of QR decomposition

Let A be a matrix over a field. It is required to find the upper triangular matrix R and the orthogonal (unitary if the initial field is a field of complex numbers) matrix Q such that $A = Q^T R$.

Consider the case of a 2×2 matrix. The desired decomposition $A = QR$ has the form:

$$\begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} = \begin{pmatrix} c & -s \\ s & c \end{pmatrix} \begin{pmatrix} a & b \\ 0 & d \end{pmatrix},$$

$$Q^T = g_{\alpha,\gamma} = \begin{pmatrix} c & s \\ -s & c \end{pmatrix}.$$

If $\gamma = 0$ then we can set $c = 1,\ s = 0$.

If $\gamma \neq 0$, then $\Delta = \alpha^2 + \gamma^2 > 0$, and we get $c = \alpha/\Delta,\quad s = \gamma/\Delta$. We denote such a matrix Q^T by $g_{\alpha,\gamma}$.

Let the matrix A be given, its elements (i, j) and $(i+1, j)$ be α and γ, and all the elements to the left of them be zero:

$$a_{i,j} = \alpha,\ a_{i+1,j} = \gamma,\ \forall (s < j) : (a_{i,s} = 0)\ \&\ (a_{i+1,s} = 0).$$

Let us denote Givens matrix:

$$G_{i,j} = \mathbf{diag}(I_{i-1}, g_{\alpha,\gamma}, I_{n-i-1}). \tag{1}$$

Then the matrix $G_{i,j}A$ differs from A only in two rows i and $i+1$. All elements in these two rows which are located to the left of the column j, remain zero, and $a_{i+1,j}$ will will become 0.

This property of the Givens matrix allows us to formulate such an algorithm.

(1). First we reset the elements under the diagonal in the left column:

$$A_1 = G_{1,1}G_{2,1}...G_{n-2,1}G_{n-1,1}A.$$

(2). Then we reset the elements that are under the diagonal in the second column:

$$A_2 = G_{2,2}G_{3,2}...G_{n-2,2}G_{n-1,2}A_1.$$

(k). Denote

$$G_{(k)} = G_{k,k}G_{k-1,k}...G_{n-2,k}G_{n-1,k}, \tag{2}$$

for $k = 1, 2, .., n-1$. Then, to calculate the elements of the k-th column, we need to obtain the product of matrices

$$A_k = G_{(k)}A_{k-1}.$$

(n-1). At the end of the calculation, the element in the $n-1$ column will be reseted: $A_{n-1} = G_{(n-1)}A_{n-2} = G_{n-1,n-1}A_{n-2}$.

Let's find the number of operations in this algorithm.

It is necessary to calculate the $(n^2 - n)/2$ Givens matrices and for each of them 6 operations must be performed. When calculating A_1 in (1), the number of multiplications of the Givens matrices into columns of two elements (4 multiplications and 2 additions) is $(n-1)^2$. When calculating A_2 in (2), the number of such multiplications is $(n-2)^2$, and so on. As a result, we get

$$6(n^2 - n)/2 + 6\sum_{i=1..n-1} i^2 = 3n^2 - 3n + 6(n-1)(2n-1)n/6 \approx 2n^3$$

Here we count the number of all arithmetic operations and the operations of extracting the square root.

3 Recursive QR decomposition

We will change the order of actions in the algorithm so that it becomes a recursive algorithm, which contains another recursive algorithm (QP algorithm).

Let a matrix M of size $2n \times 2n$ be divided into four equal blocks:

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix}.$$

3.1 Recursive QR decomposition algorithm.

There are three steps in this algorithm.
(1). The first step is the QR decomposition of the C block: $Q_1 C = C^U$

$$M_1 = \begin{pmatrix} I & 0 \\ 0 & Q_1 \end{pmatrix} \begin{pmatrix} A & B \\ C & D \end{pmatrix} = \begin{pmatrix} A & B \\ C^U & D_1 \end{pmatrix}, \quad D_1 = Q_1 D. \tag{3}$$

(2). The second step is the decomposition of a "parallelogram", which is consist of two triangular blocks: the lower triangular part of the block A and the upper triangular part of the block C^U:

$$M_2 = Q_2 \begin{pmatrix} A & B \\ C^U & D_1 \end{pmatrix} = \begin{pmatrix} A^U & B_1 \\ 0 & D_2 \end{pmatrix}. \tag{4}$$

(3). The third step is the QR decomposition of the D_2 block: $Q_3 D_3 = D^U$.

$$R = \begin{pmatrix} I & 0 \\ 0 & Q_3 \end{pmatrix} \begin{pmatrix} A^U & B_1 \\ 0 & D_2 \end{pmatrix} = \begin{pmatrix} A^U & B_1 \\ 0 & D^U \end{pmatrix}. \tag{5}$$

As a result, we get:

$$M = QR, \quad Q = \mathbf{diag}(I, Q_3) Q_2 \, \mathbf{diag}(I, Q_1). \tag{6}$$

Since the first and third steps are recursive calls of the QR procedures, it remains to describe the procedure for decomposing a "parallelogram". We will call it QP-decomposition.

3.2 Recursive QP-decomposition.

We are looking for the decomposition of the matrix P:

$$Q_2 P = P^U, \; Q_2^T = Q_2^{-1}, \; P = Q_2^T P^U.$$

The matrix $P = \begin{pmatrix} A \\ C^U \end{pmatrix}$ is the left half of the matrix M_1 (3). We divided each of the two blocks that make up matrix P (of size $2n \times n$) into four equal parts. We got 8 blocks, including one zero block and two upper triangular blocks: f^U and h^U (7).

It is required to annul all elements between the upper and lower diagonals of the P matrix, including the lower diagonal. We are interested in a block procedure. Since n is even, we can split the parallelogram formed by the diagonals into 4 parts using its two middle lines. We get 4 equal parallelograms. To decompose each of them, we simply call the QP-procedure 4 times. We will decompose each of them in the following order: lower left (P_{ll}), then simultaneously cancel the upper left (P_{ul}) and the lower right (P_{lr}) and the last we decompose parallelogram in the upper right corner (P_{ur}). The corresponding orthogonal matrices of size $n \times n$ are denoted as Q_{ll}. Q_{ul}. Q_{lr} and Q_{ur}

$$P = \begin{pmatrix} A \\ C^U \end{pmatrix} = \begin{pmatrix} a & c \\ b & d \\ f^U & g \\ 0 & h^U \end{pmatrix}, \quad P^U = \begin{pmatrix} A^U \\ 0 \end{pmatrix} = \begin{pmatrix} a^U & c_1 \\ 0 & d^U \\ 0 & 0 \\ 0 & 0 \end{pmatrix}. \tag{7}$$

$$P_{ll} = Q_{ll} \begin{pmatrix} b & d \\ f^U & g \end{pmatrix} = \begin{pmatrix} b^U & d_1 \\ 0 & g_1 \end{pmatrix}, \tag{8}$$

$$P_{ul} = Q_{ul} \begin{pmatrix} a & c \\ b^U & d_1 \end{pmatrix} = \begin{pmatrix} a^U & c_1 \\ 0 & d_2 \end{pmatrix},$$

$$P_{lr} = Q_{lr} \begin{pmatrix} 0 & g_1 \\ 0 & h^U \end{pmatrix} = \begin{pmatrix} 0 & g^U \\ 0 & 0 \end{pmatrix}, \tag{9}$$

$$P_{ur} = Q_{ur} \begin{pmatrix} 0 & d_2 \\ 0 & g^U \end{pmatrix} = \begin{pmatrix} 0 & d^U \\ 0 & 0 \end{pmatrix}. \tag{10}$$

As result, we get matrices Q_2 (4) and P^U (7):

$$Q_2 = \bar{Q}_{ur}\bar{Q}_2\bar{Q}_{ll}, \quad Q_2 P = P^U, \tag{11}$$

with such factors:

$$\bar{Q}_{ur} = \mathbf{diag}(I_{\frac{n}{2}}, Q_{ur}, I_{\frac{n}{2}}),\ \bar{Q}_2 = \mathbf{diag}(Q_{ul}, Q_{lr}),\ \bar{Q}_{ll} = \mathbf{diag}(I_{\frac{n}{2}}, Q_{ll}, I_{\frac{n}{2}})$$

3.3 Complexity of QP decomposition algorithm

The total number of multiplications of matrix blocks of size $n/2 \times n/2$ is 28: 8 multiplications is necessary for the matrices P_{ll} (8) and P_{ul} (9) and 20 multiplications is necessary for the matrix Q_2 (11).

Hence the total number of operations: $Cp(2n) = 4Cp(n) + 24M(n/2)$. Suppose that for multiplication of two matrices of size $n \times n$ you need γn^β operations and $n = 2^k$, then we get:

$$Cp(2^{k+1}) = 4Cp(2^k) + 28M(2^{k-1}) = 4^k Cp(2^1) + 28\gamma \sum_{i=0}^{k-1} 4^{k-i-1} 2^{i\beta} =$$

$$28\gamma(n^2/4)\frac{2^{k(\beta-2)}-1}{2^{(\beta-2)}-1} + 6n^2 = 7\gamma\frac{n^\beta - n^2}{2^\beta - 4} + 6n^2$$

$$Cp(n) = \frac{7\gamma n^\beta}{2^\beta(2^\beta-4)} + \frac{n^2}{2}(3 - \frac{7\gamma}{2^{\beta+1}-8}). \tag{12}$$

4 The complexity of QR decomposition algorithm

Let us estimate the number of operations $C(n)$ in this block-recursive QR-decomposition algorithm, assuming that the complexity of the matrix multiplication is $M(n) = \gamma n^\beta$, the complexity of QP-decomposition is $Cp(n) = \alpha n^\beta$, where α, β, γ are constants, $\alpha = \frac{7\gamma}{2^\beta(2^\beta-4)}$ and $n = 2^k$:

$$C(n) = 2C(n/2) + Cp(n) + 6M(n/2) = 2C(2^{k-1}) + Cp(2^k) + 6M(2^{k-1}) =$$

$$C(2^0)2^k + \sum_{i=0}^{k} 2^{k-i} Cp(2^i) + 6\sum_{i=0}^{k} 2^{k-i} M(2^{i-1}) =$$

$$\alpha \sum_{i=0}^{k} 2^{k-i} 2^{i\beta} + 6\gamma \sum_{i=0}^{k} 2^{k-i} 2^{(i-1)\beta} =$$

$$(\alpha 2^k + 6\gamma 2^{k-\beta}) \sum_{i=0}^{k} 2^{i(\beta-1)} =$$

$$(\alpha + 6\gamma 2^{-\beta}) \frac{2^\beta n^\beta - 2n}{2^\beta - 2} = \frac{\gamma(2^\beta 6 - 17)(n^\beta - \frac{2n}{2^\beta})}{(2^\beta - 4)(2^\beta - 2)}$$

For the case of $\beta = 3$, $\gamma = 1$, we get $C(n) = (31/24)(n^3 - n/4)$.
For the case of $\beta = \log_2 7$, $\gamma = 1$, we get $C(n) = (5/3)(n^\beta - 2n/7)$.

5 Conclusion

We have presented a recursive algorithm for computing the QR-factorization of a matrix, which is based on Givens rotations and has the complexity of matrix multiplication. It is not difficult to see that Givens rotations can be replaced by Householder transformation [6]. For this, it is necessary to use matrices

$$H_k = I - \frac{2}{||u_k||_2^2} u_k u_k^T$$

(with Householder vector u_k) insted of $G_{(k)}$ (2). All other steps of the recursive algorithm can be left unchanged. An experimental comparison of these two variants of QR-decomposition is of interest. Of particular interest is the implementation of these algorithms for a distributed memory cluster.

References

[1] G. Malaschonok and E. Ilchenko. *Recursive Matrix Algorithms in Commutative Domain for Cluster with Distributed Memory.*2018 Ivannikov Memorial Workshop (IVMEM), Yerevan, Armenia, (2018), pp. 40–46. doi: 10.1109/IVMEM.2018.00015.

[2] G.I. Malaschonok. *Fast Generalized Bruhat Decomposition.* 12th International Workshop on Computer Algebra in Scientific Computing (CASC 2010), LNCS, vol. 6244 (2010), pp. 194–202.

[3] G. Malaschonok. *Generalized Bruhat decomposition in commutative domains.* Computer Algebra in Scientific Computing, CASC2013, LNCS, vol. 8136 (2013), pp. 231–242.

[4] A. Tiskin. *Communication-efficient parallel generic pairwise elimination.* Future Generation Computer Systems, vol. 23, no. 2 (2007), pp. 179–188. doi:10.1016/j.future.2006.04.017

[5] A. Schönhage. *Unitre Transformationen groer Matrizen.* Numerische Mathematik 20, (1973), pp. 409–417

[6] A. S. Householder. *Unitary Triangularization of a Nonsymmetric Matrix.* Journal of the ACM, vol. 5, no. 4, (1958), pp. 339–342. doi:10.1145/320941.320947. MR 0111128.

Gennadi Malaschonok[1], Andriy Ivashkevich[2]

[1] National University of Kyiv-Mohyla Academy, 2 Skovorodi str., Kiev 04070, Ukraine
Email:malaschonok@gmail.com

[2]Kyiv Academic University, 36 Vernadsky blvd., Kyiv 03142, Ukraine
Email:andriyivaskevich99@gmail.com