728

PARALLEL STRATEGIES FOR RANK-K UPDATING OF THE QR DECOMPOSITION

ERRICOS. J. KONTOGHIORGHES*† AND DENNIS PARKINSON*

Abstract. Parallel strategies are proposed for updating the QR decomposition of an $n \times n$ matrix after a rank-k change ($k < n$). The complexity analysis of the Givens algorithms is based on the total number of compound disjoint Givens rotations (CDGRs) applied. The first algorithm is an extension of the rank-one updating method. It computes the updating using $2(k+n-2)$ CDGRs with elements annihilated by rotations in adjacent planes. A second rank-one (*greedy*) algorithm that applies rotations in non-adjacent planes is found to use approximately half CDGRs compared with that of the serial rank-one algorithm, when $n = 2^g$. For $k < g$ the efficiency of the *greedy* algorithm which applies approximately $(1 - 2^{-k})2^{(g+1)}$ CDGRs, decreases as k increases, while for $k > g$ it is found to require more steps than the first parallel algorithm. Block generalization of the serial and *greedy* rank-one algorithms are also presented. Both algorithms are rich in level 3 BLAS operations that make them suitable for large scale parallel systems.

Key words. QR decomposition, Givens rotations, parallel algorithms

AMS subject classifications. 15A23, 65F05, 65F25, 65Y05

1. Introduction. Given the QR Decomposition (QRD) of a non-singular $n \times n$ matrix A

$$A = QR \tag{1}$$

the problem of recomputing the QRD of

$$\tilde{A} = A + \sum_{i=1}^{k} x_i y_i^T = A + XY^T \tag{2}$$

is considered, where $x_i, y_i \in \Re^n$, $X = (x_1 \dots x_k)$, $Y = (y_1 \dots y_k)$, $R \in \Re^{n \times n}$ is upper triangular and $Q \in \Re^{n \times n}$ is orthogonal. The matrix XY^T has rank k and the problem is known as the rank-k updating of the QRD (hereafter rank-k UQRD) problem. Observing that $\tilde{A} = Q(R + Q^T XY^T)$, the rank-k UQRD problem is the reduction of

$$\hat{A} = R + ZY^T \tag{3}$$

into upper triangular form by orthogonal transformations, where $Z = Q^T X$. An algorithm for updating the QRD after rank-one change, that is $k = 1$ in (2) , has been described in [6, 7]. The purpose of this work is to develop and analyse parallel strategies for fast rank-k UQRD, which is very important for applications where repeated updating is required [18].

* Department of Computer Science, Queen Mary and Westfield College, Mile End Road, London E1 4NS, UK (ricos@dcs.qmw.ac.uk, dennisp@dcs.qmw.ac.uk).

† Centre for Mathematical Trading and Finance, City University Business School, Frobisher Crescent, Barbican Centre, London EC2Y 8HB, UK

The algorithms will be based on Givens rotations and Householder transformations. Throughout the paper $G_{i,j}^{(k)}$ denotes a Givens rotation in plane (i,j) that reduces to zero the element w_{ik} when it is applied from the left of $W \in \Re^{n\times k}$. It is assumed that the maximum of $\lfloor n/2 \rfloor$ disjoint Givens rotations can be applied simultaneously. The product of these rotations is called CDGR, short for compound disjoint Givens rotation [4, 10, 11, 13]. A single step equivalent to *one time unit* is required to construct and apply a CDGR on matrices of any size, while the time required to compute the rank-k updating $R + ZY^T$ in (2) is assumed to be negligible for any k and n. Under these assumptions the simultaneous application of the CDGR on the orthogonal matrix Q^T in (1) will not have any affect on the time complexity of the algorithms. Therefore, for simplicity the construction of the orthogonal matrix in the updated QRD will not be shown.

In §2 a straight-forward parallelisation of the algorithm in [6] for solving the rank-k UQRD problem is presented when $k > 1$. In §3 a *greedy* algorithm based on recursive doubling is proposed for solving the rank-one UQRD problem. The generalization of the rank-one algorithms and the adaptation of the block-parallel algorithms in [2, 9] are investigated in §4 for solving the updating problem where $k > 1$. Finally, the conclusions and future work is presented in §5.

2. Parallelization of the rank-one updating algorithm. The parallelization of the rank-one UQRD algorithm in [6] is considered for the rank-k case, where $1 \leq k < n$. The first stage of the rank-one algorithm in [6] is to apply the $n-1$ Givens rotations $V^T = G_{2,1}^{(1)} \dots G_{n-1,n-2}^{(1)} G_{n,n-1}^{(1)}$ into the augmented matrix $(Z\ R)$ such that $V^T(Z\ R) = (\zeta e_1\ H)$, where $\zeta^2 = Z^T Z$, e_1 is the first column of the $n \times n$ identity matrix I_n and H is an upper Hessenberg matrix [7]. After computing $\tilde{H} = H + \zeta e_1 Y^T$, the second stage of the algorithm computes the $n-1$ Givens rotations $U^T = G_{n,n-1}^{(n-1)} \dots G_{3,2}^{(2)} G_{2,1}^{(1)}$ to retriangularize the Hessenberg matrix $\tilde{H}$. Thus, the UQRD of $\tilde{A} = A + XY^T = QVUR_n = Q_n R_n$, where R_n is upper triangular and Q_n is orthogonal. A total of $2(n-1)$ steps have been applied using this algorithm.

For the solution of the rank-k UQRD problem when $k > 1$ the above algorithm can be repeated k times using $2k(n-1)$ steps. However, computations on z_i, the ith column of Z, can commence after the last two elements of z_{i-1} $(i = 2, \dots, k)$ have been annihilated. Therefore, the annihilation of the elements in z_i can start at the $(2i-1)$th step and will *fill-in* successively the ith subdiagonal of R. Once z_i has been reduced in the form $\zeta_i e_1$ the rank-one updating $R + \zeta_i e_1 y_i^T$ is performed. This method is a variation of Sameh and Kuck annihilation scheme (hereafter VSK algorithm) in [17] and requires $2k + n - 3$ steps to perform the first stage of the rank-k UQRD problem. Alternatively, the annihilation of the elements of z_i can stop once the ith subdiagonal of R has been *filled-in*, since at this stage the updating $R + z_i y_i^T$ could be performed without creating any further modification of the structure of R. This

(SK algorithm) is equivalent in computing the QRD

$$\tilde{Q}^T Z = \begin{pmatrix} R_z \\ 0 \end{pmatrix} \begin{matrix} k \\ n-k \end{matrix} \tag{4}$$

using the SK annihilation scheme and $H = \tilde{Q}^T R$. The matrix resulting from the rank-k updating

$$\tilde{H} = H + \begin{pmatrix} R_z \\ 0 \end{pmatrix} Y^T \tag{5}$$

will have the same structure as H, that is, its last $n-k-1$ subdiagonals will be zero. The first stage in Fig. 1 illustrates the transformations on Z and the *fill-in* of R, where $n = 8$. At the first stage the integers i in the matrix Z and $\not{i}$ in matrix R denote, respectively, the elements annihilated and the *fill-in* after applying the ith CDGR.

For the retriangularization of $\tilde{H}$ a total of $k+n-2$ CDGRs can be used. At step $k+1-i$ the elements of the ith $(i = 1,\ldots,k)$ subdiagonal start to be annihilated successively by the $n-i$ Givens rotations $G^{(1)}_{i+1,i}, G^{(2)}_{i+2,i+1}, \ldots, G^{(n-i)}_{n,n-1}$. In the case of the VSK algorithm, the triangularization of $\tilde{H}$ can start after the $(k+n-1)$th CDGR has been applied into Z. That is, the VSK algorithm requires one step more than the SK algorithm. The total number of CDGR applied using the SK algorithm is given by

$$T_{\text{SK}}(k,n) = 2(k+n-2).$$

Alternatively, the $\tilde{H}$ matrix can be triangularized by the series of $n-1$ Householder transformations $P^{(1)}, P^{(2)}, \ldots, P^{(n-1)}$, where $P^{(i)}$ annihilates the elements $i+1$ to $\min(k+i,n)$ of the ith column of $\tilde{H}$ $(i = 1,\ldots,n-1)$ by using the ith row of $\tilde{H}$ as a pivot row. Observe that the last Householder reflection is equivalent to a single Givens rotation. Furthermore, the parallel algorithms are identical to the serial rank-one Givens algorithm in [6] when $k = 1$. The second stage of Fig. 1 shows the annihilation pattern of both methods. In the case of Householder reflections an integer i $(i = 1,\ldots,n-1)$ denotes the elements annihilated by the ith Householder transformation.

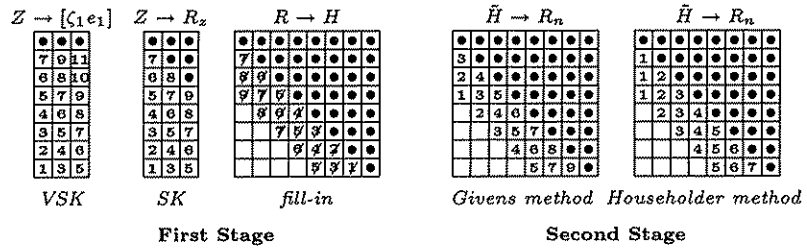


FIG. 1. *Rank-k updating of the QRD by* CDGRs *and Householder transformations.*

The QRD of an $n \times n$ matrix using the SK annihilation scheme requires $2n-3$ CDGRs. Therefore, after explicitly computing the updating $R + ZY^T$ the QRD of $\hat{A}$ in (3) requires $(2k+1)$ less steps than the SK rank-k algorithm, when $k > 1$.

3. The greedy rank-one algorithm. The reduction of the n-element vector Z into the form ζe_1 can be obtained by applying the minimum $\lfloor n/2 \rfloor$ CDGRs using the recursive doubling or *greedy* method [4, 9]. The Givens rotations are not performed on adjacent planes and generally, the application of the single Givens rotation $G_{i,j}^{(k)}$ from the left of the augmented matrix $(Z\ R)$ *fills-in* the elements j to $(i-1)$ of the ith row of R $(i > j)$. However, a sequence of CDGRs can be found which retriangularizes R fast enough so that the total number of steps needed to solve the rank-one UQRD problem is less than the number of steps required by other algorithms.

Assume for simplicity that $n = 2^g$. At the ith $(i = 1, \ldots, g)$ step of the *greedy* algorithm the elements $2^{(g-i)}+1$ to $2^{(g+1-i)}$ of Z are annihilated by the $2^g \times 2^g$ CDGR

$$C^{(i,g)} = \prod_{j=2^{(g-i)}+1}^{2^{(g+1-i)}} G_{j,j-2^{(g-i)}}^{(1)}$$

which can also be written as

$$C^{(i,g)} = \begin{pmatrix} C^{(i-1,g-1)} & 0 \\ 0 & I_{2^{(g-1)}} \end{pmatrix} \quad \text{for } i > 1.$$

It can be proven that the matrix $H^{(g)} = C^{(g,g)} \ldots C^{(1,g)} R$ and $\tilde{H}^{(g)} = H^{(g)} + \zeta e_1 Y^T$ has a special recursive structure which facilitates the development of an efficient Givens algorithm for triangularizing $\tilde{H}^{(g)}$.

THEOREM 3.1. *The structure of the matrix* $H^{(g)} = C^{(g,g)} \ldots C^{(1,g)} R$ *is given by*

$$H^{(g)} = \begin{array}{c} \begin{matrix} 2^{(g-1)} & 2^{(g-1)} \end{matrix} \\ \begin{pmatrix} H^{(g-1)} & \tilde{B}^{(g-1)} \\ R^{(g-1)} & \hat{B}^{(g-1)} \end{pmatrix} \begin{matrix} 2^{(g-1)} \\ 2^{(g-1)} \end{matrix} \end{array}, \tag{6}$$

where $\tilde{B}^{(g-1)}$ *and* $\hat{B}^{(g-1)}$ *are full dense matrices,* $R^{(g-1)}$ *is upper triangular,* $H^{(g-1)}$ *has the same structure as* $H^{(g)}$ *but with* g *replaced by* $g-1$ $(g \geq 1)$ *and* $H^{(0)}$ *is a non-zero scalar.*

Proof. For $g = 1$ the matrix $H^{(1)} = C^{(1,1)} R = G_{2,1}^{(1)} R$ is a 2×2 dense matrix which satisfies the structure (6). The inductive hypothesis is that the structure of $H^{(g)}$ defined in (6) is true. Now, it remains to show that the structure (6) is also true for $n = 2^{(g+1)}$, given that is true for $n = 2^g$. The application of the first CDGR from the left of the R gives

$$C^{(1,g+1)} R = \begin{array}{c} \begin{matrix} 2^g & 2^g \end{matrix} \\ \begin{pmatrix} \bar{R}^{(g)} & \check{B}^{(g)} \\ R^{(g)} & \hat{B}^{(g)} \end{pmatrix} \begin{matrix} 2^g \\ 2^g \end{matrix} \end{array}, \tag{7}$$

where $R^{(g)}$ and $\bar{R}^{(g)}$ are upper triangular and $\hat{B}^{(g)}$ and $\check{B}^{(g)}$ are full dense matrices. This is obvious since the Givens rotation $G_{j,j-2^g}^{(1)}$ *fills-in* the elements $(j - 2^g)$ to

$(j-1)$ of the jth row of R $(j = 2^g+1,\ldots,2^{g+1})$. That is, the *fill-in* of R from the application of $C^{(1,g+1)}$ is a parallelogram of height 2^g. For $\tilde{B}^{(g)} = \prod_{i=1}^{g} C^{(i,g)}\check{B}^{(g)}$ the application of the CDGRs from the left of R can be written as

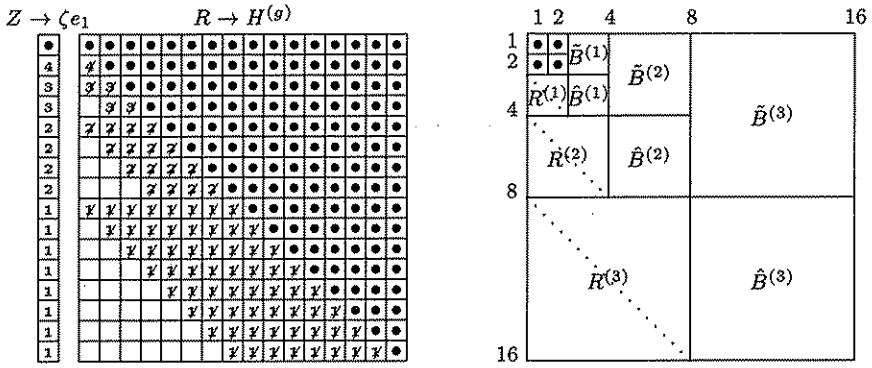
$$\begin{aligned} H^{(g+1)} &= \prod_{i=1}^{g+1} C^{(i,g+1)} R = \begin{pmatrix} \prod_{i=1}^{g} C^{(i,g)} & 0 \\ 0 & I_{2^g} \end{pmatrix} \begin{pmatrix} \bar{R}^{(g)} & \check{B}^{(g)} \\ R^{(g)} & \hat{B}^{(g)} \end{pmatrix} \\ &= \begin{pmatrix} \prod_{i=1}^{g} C^{(i,g)} \bar{R}^{(g)} & \tilde{B}^{(g)} \\ R^{(g)} & \hat{B}^{(g)} \end{pmatrix}, \end{aligned}$$

and using the inductive hypothesis $H^{(g)} = \prod_{i=1}^{g} C^{(i,g)}\bar{R}^{(g)}$ the latter can be expressed in the form

$$H^{(g+1)} = \begin{pmatrix} H^{(g)} & \tilde{B}^{(g)} \\ R^{(g)} & \hat{B}^{(g)} \end{pmatrix},$$

which completes the proof. □

Clearly $\tilde{H}^{(g)}$ has the same structure as $H^{(g)}$. Figure 3 shows the recursive structure of $H^{(g)}$ and the process of reducing $(Z\ R)$ into $(\zeta_1 e_1\ H^{(g)})$ for $g = 4$.



The reduction of $(Z\ R)$ into $(\zeta_1 e_1\ H^{(g)})$ **The recursive structure of $\tilde{H}^{(4)}$**

FIG. 2. *The computation of $(\zeta_1 e_1\ H^{(4)})$ using th greedy algorithm and the structure of $\tilde{H}^{(4)}$.*

The reduction of the matrix $\tilde{H}^{(g)}$ $(g > 1)$ into upper triangular form can be obtained by forming the orthogonal factorizations

$$\tilde{Q}^T\tilde{H}^{(g)} = \tilde{Q}^T \begin{pmatrix} \tilde{H}^{(g-1)} & \tilde{B}^{(g-1)} \\ R^{(g-1)} & \hat{B}^{(g-1)} \end{pmatrix} = \begin{pmatrix} W^{(g-1)} & \tilde{B}_*^{(g-1)} \\ \hat{R}^{(g-1)} & \hat{B}_*^{(g-1)} \end{pmatrix} \begin{matrix} 2^{(g-1)} \\ 2^{(g-1)} \end{matrix}, \tag{8}$$

and

$$\hat{Q}^T \begin{pmatrix} W^{(g-1)} & \tilde{B}_*^{(g-1)} \\ \hat{R}^{(g-1)} & \hat{B}_*^{(g-1)} \end{pmatrix} = R_n, \tag{9}$$

where only the first row of $\left(\tilde{H}^{(g-1)}\ \tilde{B}^{(g-1)}\right)$ is different from its corresponding submatrix in (6), $W^{(g-1)}$ and $\hat{R}^{(g-1)}$ are upper triangular matrices and $(\hat{Q}\tilde{Q})R_n$ is the QRD of $\tilde{H}^{(g)}$.

The factorization of (8) can be computed in $2^{(g-2)}$ steps using a Givens sequence similar to that in [9]. At the jth step the pth subdiagonal of $\tilde{H}^{(g-1)}$ is annihilated by applying simultaneously for $q = 1, \ldots, j-1+2^{(g-2)}$ the Givens rotations $G^{(q)}_{p+q,q+2^{(g-1)}}$, where $p = 2^{(g-2)} + 1 - j$. Notice that prior to the rotations some of the elements of the subdiagonal will already be zero and that the factorization (8) can start after the second CDGR on Z has been applied. The computation of factorization (9) is divided into two stages. In the first stage the upper triangular matrix $\hat{R}^{(g-1)}$ is reduced to zero by applying $2^{(g-1)}$ CDGRs. At the jth ($j = 1, \ldots, 2^{(g-1)}$) step the Givens rotations $G^{(j+p-1)}_{p+2^{(g-1)},j+p-1}$ are applied simultaneously for $p = 1, \ldots, 2^{(g-1)} - j + 1$ in order to annihilate the jth superdiagonal of $\hat{R}^{(g-1)}$, where within this context the main diagonal is equivalent to the first superdiagonal. In the second stage the matrix $\hat{B}^{(g-1)}_*$ is triangularized using $2^g - 3$ CDGRs by employing the SK annihilation scheme. However, the second stage can start after the last two rows of $\hat{R}^{(g-1)}$ have been zeroed. That is, the annihilation of $\hat{B}^{(i)}_*$ starts after the second step of the first stage. Hence, $2^g - 1$ steps are needed to compute factorization (9) . Therefore, the total number of steps required to solve the rank-one UQRD using the *greedy* algorithm is given by

$$T_{\rm gr}(g) \quad = \quad 2 + 2^{(g-2)} + 2^g - 1 = 5 \times 2^{(g-2)} + 1 \quad \text{for } g > 1$$

and by omitting additive constants it follows that $T_{\rm gr}(g)/2^{(g+1)} \simeq 5/8$. The latter shows that the *greedy* algorithm is (approximately) doing 5/8 of the steps performed by that of the serial Givens rank-one algorithm which is equivalent to the SK annihilation scheme for $k = 1$. Figure 3 illustrates the annihilation pattern of the *greedy* based parallel Givens strategies for solving the rank-one UQRD problem, where $g = 4$. The element ④ denotes a zero element of $\tilde{H}^{(g-1)}$ that remains unchanged from the application of the CDGRs.

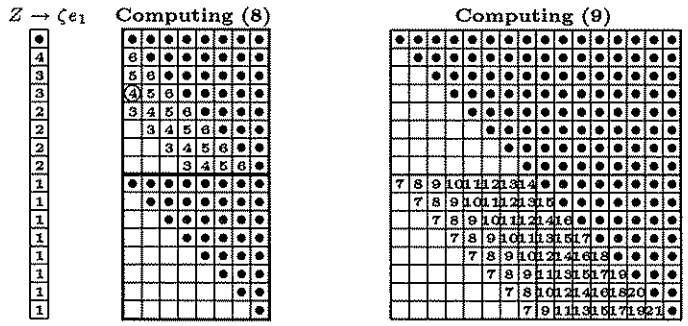


FIG. 3. *Annihilation pattern of the* greedy *rank-one UQRD algorithm, where* $g = 4$.

Recursive derivation of the QRD of $\tilde{H}^{(g)}$ for $g > 3$. Substituting g by $g-1$ in $\tilde{H}^{(g)}$, it gives

$$(10)\quad \tilde{H}^{(g)} = \left(\begin{array}{cc|c} \tilde{H}^{(g-2)} & \tilde{B}^{(g-2)} & \tilde{B}_1^{(g-1)} \\ R^{(g-2)} & \hat{B}^{(g-2)} & \tilde{B}_2^{(g-1)} \\ \hline \multicolumn{2}{c|}{R^{(g-1)}} & \hat{B}^{(g-1)} \end{array}\right)\begin{array}{l} 2^{(g-2)} \\ 2^{(g-2)} \\ 2^{(g-1)} \end{array} \quad \text{where} \quad \tilde{B}^{(g-1)} = \begin{pmatrix} \tilde{B}_1^{(g-1)} \\ \tilde{B}_2^{(g-1)} \end{pmatrix}.$$

Initially the orthogonal factorizations

$$(11)\qquad \tilde{H}^{(g-2)} = \bar{Q}_A \bar{R}^{(g-2)}$$

and

$$(12)\quad \bar{Q}_B^T \left(\begin{array}{cc|c} R^{(g-2)} & \hat{B}^{(g-2)} & \tilde{B}_2^{(g-1)} \\ \hline \multicolumn{2}{c|}{R^{(g-1)}} & \hat{B}^{(g-1)} \end{array}\right) = \left(\begin{array}{cc|c} 0 & \tilde{W}^{(g-2)} & \bar{B}_2^{(g-1)} \\ \hline \multicolumn{2}{c|}{\hat{R}^{(g-1)}} & \check{B}_2^{(g-1)} \end{array}\right)\begin{array}{l} 2^{(g-2)} \\ 2^{(g-1)} \end{array}$$

are computed, where $\bar{R}^{(g-2)}$, $\tilde{W}^{(g-2)}$ and $\hat{R}^{(g-1)}$ are upper triangular. For

$$\check{B}_1^{(g-1)} = \begin{pmatrix} \bar{Q}_A^T \tilde{B}_1^{(g-1)} \\ \bar{B}_2^{(g-1)} \end{pmatrix} \quad \text{and} \quad W^{(g-1)} = \begin{pmatrix} \bar{R}^{(g-2)} & \bar{Q}_A^T \tilde{B}_1^{(g-2)} \\ 0 & \tilde{W}^{(g-2)} \end{pmatrix},$$

the upper triangular factor of the QRD of $\tilde{H}^{(g)}$ is derived after computing the orthogonal factorization

$$(13)\qquad \check{Q}^T \begin{pmatrix} W^{(g-1)} & \check{B}_1^{(g-1)} \\ \hat{R}^{(g-1)} & \check{B}_2^{(g-1)} \end{pmatrix} = R_n.$$

As in factorization (8) the factorization (12) can be computed in $2^{(g-2)}$ steps. The orthogonal matrix $\bar{Q}_B^T$ is defined as the product of the CDGRs $C^{(2^{(g-2)})} \cdots C^{(1)}$, where $C^{(i)} = \prod_{j=1}^{2^{(g-2)}} G_{j,p+2^{(g-2)}}^{(p)}$ for $p = i + j - 1$. The QRD (11) is derived in $T_{\text{gr}}(g-2) = 5 \times 2^{(g-4)} + 1$ steps and factorization (13) is computed in $2^g - 1$ steps. Factorizations (11) and (12) can start, respectively, after the 3rd and 2nd CDGR has been applied from the left of Z. Hence, the total number of steps required to triangularize $\tilde{H}^{(g)}$ using the above method is given by

$$\begin{aligned} \tilde{T}_{\text{gr}}(g) &= \max\Big(1 + T_{\text{gr}}(g-2), 2 + 2^{(g-2)}\Big) + 2^g - 1 \\ &= 21 \times 2^{(g-4)} + 1 \quad \text{for } g > 3 \end{aligned}$$

and $\tilde{T}_{\text{gr}}(g)/T_{\text{gr}}(g) \simeq 21/20$. This indicates that the latter method requires more CDGRs for solving the rank-one UQRD problem compared with that of computing the factorizations (8) and (9) . The same conclusions are expected to be drawn if the above method is used with g replaced by $g - 2^i$ $(i = 0, \ldots, g-1)$ in (10) . Observe that at least $2^{(g-2)}$ steps will be required to compute simultaneously the factorizations equivalent to (11) and (12) , and $2^g - 1$ steps are needed to form the final factorization which corresponds to (13) . That is, the recursive triangularization of $\tilde{H}^{(g)}$ will at least require the same number of steps as $T_{\text{gr}}(g)$.

4. The greedy and block rank-k algorithms. The *greedy* Givens sequence described in [4, 13] can be used to reduce the $n \times k$ matrix Z into upper triangular form using $g + (k-1)\log_2 g$ steps, where $n = 2^g \gg k$ and we only consider the case $k < g$. However, the *greedy* algorithm will result in $\tilde{H}$ having a non-full dense structure that is difficult to exploit. This can be overcomed by using a variation of the *greedy* algorithm called *log-greedy*. At each step the *log-greedy* algorithm annihilates $2^{\lceil \log_2 x_i \rceil}$ elements of the ith column by preserving previously zeroed elements, where x_i is the maximum number of elements in column i that can be possibly annihilated. When $\lceil \log_2 x_i \rceil < \log_2 x_i$, the rotations are chosen so that there is less *fill-in* in R. Obviously the *log-greedy* algorithm requires more steps than the *greedy* algorithm when $k > 1$, while for $k = 1$ and $n = 2^g$ both algorithms are equivalent to the *greedy* rank-one algorithm. The difference in the number of steps between the two algorithms are negligible for $k < g$.

The method has k stages and each with a number of steps. In stage j the algorithm deals with column $(k-j+1)$ of the update. Let $\tilde{G}^{(i,j)}$ and $F^{(i,j)}$ denote, respectively, the CDGR applied at step i in column j of Z and the *fill-ins* of R resulting from the application of $\tilde{G}^{(i,j)}$, where $i \geq j$. All the *fill-ins* of R resulting from the application of $\tilde{G}^{(1,j)}, \tilde{G}^{(2,j)}, \ldots$ are denoted by $F^{(\cdot,j)}$. A *fill-in* has a parallelogram shape and the maximum height of the parallelograms corresponding to $F^{(\cdot,j)}$ is $2^{(g-j)}$. At the jth $(j < k)$ stage the *fill-ins* $F^{(\cdot,k+1-j)}$ are annihilated simultaneously. Using the Givens strategy for computing the factorization (9) a parallelogram of height 2^p can be annihilated in $2^{(p+1)} - 1$ steps. Therefore, the jth stage is completed in $2^{(g-k+j)} - 1$ steps. After the first $k-1$ stages the matrix $\tilde{H}$ will have the same recursive structure as $H^{(g)}$ in (6) . However, the computation of factorization (8) can start prior to the complete annihilation of $F^{(2,2)}$. The number of steps required by the *log-greedy* algorithm to solve the rank-k UQRD problem ($k < g$ and $n = 2^g$) is given approximately by

$$T_{\text{logr}}(k,g) \simeq \sum_{j=1}^{k} (2^{(g-k+j)}) = (1-2^{-k})2^{(g+1)}$$

and $T_{\text{logr}}(k,g)/2^{(g+1)} \simeq 1-2^{-k}$, where the computation of factorization (8) , the QRD of Z and small constants have been ignored. Thus, as k increases the efficiency of the *log-greedy* algorithm decreases compared with that of SK annihilation scheme or the direct parallel QRD of $\hat{A}$ in (3) . For $k > g$ the triangularization of Z using the *greedy* algorithms is inefficient since they result in $\tilde{H}$ having a full dense matrix that requires $O(2^{(g+1)})$ CDGRs to be triangularized.

Figure 4 shows the first stage and the whole process for solving the rank-2 UQRD problem, where $g = 4$. The non-empty $F^{(i,2)}$ $(i = 2, 4, 6)$ is distinguished by bolted frames. Notice that the triangularization of $\tilde{H}$ starts before the QRD of Z is completed and also, the computation which corresponds to factorization (8) starts at step 9.

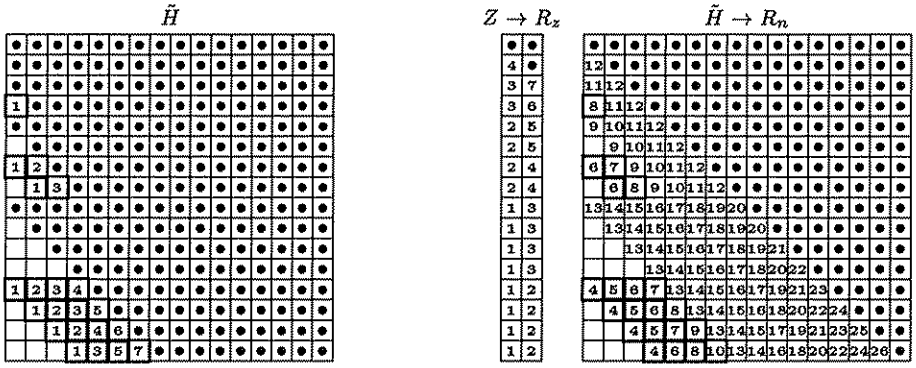


FIG. 4. *Solving the rank-2 UQRD problem using the* log-greedy *method, where* $g = 4$.

4.1. Block parallel strategies. Block generalizations of the rank-one algorithms can be employed to solve the rank-k UQRD problem. The first block parallel algorithm is based on the the serial rank-one Givens algorithm. Although blocks are processed one at a time, the computations within the blocks can be performed in parallel, either by using Householder reflections or CDGRs [9, 10]. Partitioning the matrices Z and R as

$$Z = \begin{pmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_\nu \end{pmatrix} \begin{matrix} n_1 \\ n_2 \\ \\ n_\nu \end{matrix} \quad \text{and} \quad \mathrm{R} = \begin{pmatrix} \overset{n_1}{R_{11}} & \overset{n_2}{R_{12}} & \cdots & \overset{n_\nu}{R_{1\nu}} \\ & R_{22} & \cdots & R_{2\nu} \\ & & \ddots & \vdots \\ & & & R_{\nu\nu} \end{pmatrix} \begin{matrix} n_1 \\ n_2 \\ \\ n_\nu \end{matrix}, \tag{14}$$

let the QRD of Z_ν be given by

$$Q_\nu^T Z_\nu = \begin{pmatrix} W_\nu \\ 0 \end{pmatrix} \begin{matrix} k \\ n_\nu - k \end{matrix} \quad \text{and} \quad \mathrm{Q}_\nu^{\mathrm{T}} \mathrm{R}_{\nu\nu} = \begin{pmatrix} \tilde{R}_{\nu\nu} \\ \hat{R}_{\nu\nu} \end{pmatrix} \begin{matrix} k \\ n_\nu - k \end{matrix} \tag{15}$$

where $n = \sum_{j=1}^{\nu} n_j$ and $n_\nu \geq k$. Computing for $i = \nu - 1, \ldots, 2, 1$ the block-updating QR factorization

$$Q_i^T \begin{pmatrix} \overset{k}{Z_i} & \overset{n_i}{R_{ii}} & \cdots & \overset{n_\nu}{R_{i\nu}} \\ W_{i+1} & 0 & \cdots & \tilde{R}_{i+1\nu} \end{pmatrix} \begin{matrix} n_i \\ k \end{matrix} = \begin{pmatrix} \overset{k}{W_i} & \overset{n_i}{\tilde{R}_{ii}} & \cdots & \overset{n_\nu}{\tilde{R}_{i\nu}} \\ 0 & \hat{R}_{ii} & \cdots & \hat{R}_{i\nu} \end{pmatrix} \begin{matrix} k \\ n_i \end{matrix}, \tag{16}$$

the upper triangular matrix R_z in (4) is given by $R_z = W_1$ and the matrix $\tilde{H}$ in (5) has the block-Hessenberg structure

$$\tilde{H} = \begin{pmatrix} \overset{n_1}{\tilde{H}_{11}} & \overset{n_2}{\tilde{H}_{12}} & \dots & \overset{n_\nu}{\tilde{H}_{1\nu}} \\ \hat{R}_{11} & \hat{R}_{12} & \dots & \hat{R}_{1\nu} \\ & \hat{R}_{22} & \dots & \hat{R}_{2\nu} \\ & & \ddots & \vdots \\ & & & \hat{R}_{\nu\nu} \end{pmatrix} \begin{matrix} k \\ n_1 \\ n_2 \\ \\ n_\nu - k \end{matrix}, \tag{17}$$

where Q_i is orthogonal, W_i is a $k \times k$ upper triangular matrix and

$$\begin{pmatrix} \tilde{H}_{11} & \tilde{H}_{12} & \dots & \tilde{H}_{1\nu} \end{pmatrix} = \begin{pmatrix} \tilde{R}_{11} & \tilde{R}_{12} & \dots & \tilde{R}_{1\nu} \end{pmatrix} + W_1 Y^T.$$

For the triangularization of $\tilde{H}$, initially compute for $i = 1, \dots, \nu - 1$ the factorizations

$$\hat{Q}_i^T \begin{pmatrix} \overset{n_i}{\tilde{H}_{ii}} & \overset{n_{i+1}}{\tilde{H}_{ii+1}} & \dots & \overset{n_\nu}{\tilde{H}_{i\nu}} \\ \hat{R}_{ii} & \hat{R}_{ii+1} & \dots & \hat{R}_{i\nu} \end{pmatrix} \begin{matrix} k \\ n_i \end{matrix} = \begin{pmatrix} \overset{n_i}{R^*_{ii}} & \overset{n_{i+1}}{R^*_{ii+1}} & \dots & \overset{n_\nu}{R^*_{i\nu}} \\ 0 & \tilde{H}_{i+1\,i+1} & \dots & \tilde{H}_{i+1\,\nu} \end{pmatrix} \begin{matrix} n_i \\ k \end{matrix}, \tag{18}$$

where $\hat{Q}_i$ is orthogonal and R^*_{ii} is upper triangular. Then the QRD

$$\hat{Q}_\nu^T \begin{pmatrix} \tilde{H}_{\nu\,\nu} \\ \hat{R}_{\nu\,\nu} \end{pmatrix} = R^*_{\nu\,\nu} \tag{19}$$

is computed, such that the required upper triangular matrix is given by

$$R_n = \begin{pmatrix} \overset{n_1}{R^*_{11}} & \overset{n_2}{R^*_{12}} & \dots & \overset{n_\nu}{R^*_{1\nu}} \\ & R^*_{22} & \dots & R^*_{2\nu} \\ & & \ddots & \vdots \\ & & & R^*_{\nu\nu} \end{pmatrix} \begin{matrix} n_1 \\ n_2 \\ \\ n_\nu \end{matrix}. \tag{20}$$

The second block-parallel algorithm operates in more than one block simultaneously. It is based on the *greedy* rank-one algorithm and the block parallel algorithms in [2, 9]. Assume for simplicity that $n_i = k$ $(i = 1, \dots, \nu)$ and $\nu = 2^g$. Using the partitioning of Z and R in (14) , initially the QRDs

$$Q_{i,0}^T \begin{pmatrix} Z_i & R_{ii} & \dots & R_{i\nu} \end{pmatrix} = \begin{pmatrix} \tilde{W}_i^{(0)} & R_{ii}^{(0)} & \dots & R_{i\nu}^{(0)} \end{pmatrix} \tag{21}$$

are computed simultaneously for $i = 1, \dots, \nu$, where $\tilde{W}_i^{(0)}$ and $R_{ij}^{(0)}$ $(j = i, \dots, \nu)$ are, respectively, upper triangular and full dense square matrices of order k. Then in step $i = 1, \dots, g$ the orthogonal factorizations

$$Q_{j,i}^T \begin{pmatrix} \tilde{W}_j^{(i-1)} R_{jj}^{(i-1)} \dots R_{j\nu}^{(i-1)} \\ \tilde{W}_p^{(i-1)} \quad 0 \quad \dots R_{p\nu}^{(i-1)} \end{pmatrix} = \begin{pmatrix} \tilde{W}_j^{(i)} R_{jj}^{(i)} \dots R_{j\nu}^{(i)} \\ 0 \quad R_{pj}^{(i)} \dots R_{p\nu}^{(i)} \end{pmatrix} \; ; p = j + 2^{(g-i)} \tag{22}$$

are computed simultaneously for $j = 1, \ldots, 2^{(g-i)}$, where $\tilde{W}_j^{(i)}$ is a $k \times k$ upper triangular matrix. After the gth step $R_z = \tilde{W}_1^{(g)}$ is computed. The matrix $\tilde{H}$ in (5) has a recursive structure identical to that of $H^{(g)}$ in (6), with the difference that single elements now correspond to $k \times k$ matrices. In this case the reduction of $\tilde{H}$ into upper triangular form is similar to the annihilation schemes employed to compute the factorizations (8) and (9), but with CDGRs replaced by *Compound Disjoint Orthogonal Matrices* (CDOMs). A CDOM is a product of orthogonal matrices that can be applied simultaneously and each orthogonal matrix is a product of CDGRs or Householder reflections. Notice that a product of disjoint CDGRs is a CDGR and a CDOM is orthogonal.

The process of solving the rank-k UQRD problem using *block-greedy* rank-k algorithm is shown in Fig. 4.1 for $g = 3$. Both ⊠ and ⧅ denote square dense and upper triangular matrices of order k, respectively. It is assumed that a single *Super-step* is required to apply a CDOM. In the first four *Super-steps* both the Z and R matrices are shown, while in the remaining steps the operations are assumed to be performed on the $\tilde{H}$ matrix.

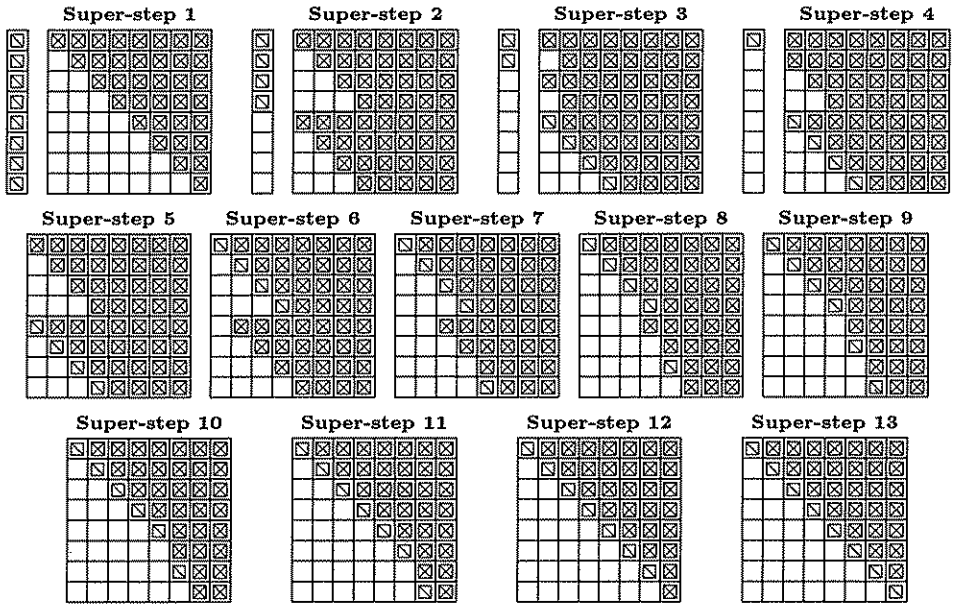


FIG. 5. *Solving the rank-k UQRD problem using the block-greedy rank-k algorithm*

5. Conclusions. Parallel strategies have been presented for retriangularizing an $n \times n$ upper triangular matrix R after a rank-k change. The first two algorithms are based on the SK annihilation scheme in [17] and solve the updating problem by applying $2(k + n - 2)$ CDGRs. However, these algorithms need more steps compared with the $2n-3$ CDGRs required to be applied for computing the QRD of $\hat{A}$ in (3), after forming the updating $R+ZY^T$. A *greedy* rank-one algorithm requiring approximately

half CDGRs than that of the serial algorithm, has also been presented. The algorithm efficiently exploited the recursive structure of the $\tilde{H}$ (*filled-in* R) matrix. However, the efficiency of a modified *greedy* algorithm for solving the rank-k UQRD problem is found to be decreasing for increasing k and performing worst than the SK parallel algorithm for $k > g$, where $n = 2^g$. Two block parallel strategies based on the serial and *greedy* rank-one algorithms have also been described for solving the rank-k UQRD problem.

The block parallel algorithms will be suitable for multiprocessor MIMD systems because of their low communication overheads and rich level 3 BLAS operations [2]. On the other hand the *greedy* parallel algorithms might be found to be efficient for SIMD systems with thousands of processing elements [8].

Future research can be directed towards the use of Fibonacci schemes for generating annihilation schemes to solve the rank-k UQRD problem. The analysis of the algorithms can be more realistically based on the assumption that less than $\lfloor n/2 \rfloor$ CDGRs can be applied simultaneously (limited parallelism) [3]. Furthermore, the rank-k updating algorithms can be extended to solve the block downdating QRD problem and the General linear model (GLM) [5, 10, 11, 12, 15]. One of the methods for solving the block downdating QRD problem requires the QRD of the square matrix B after computing the orthogonal factorization

$$G^T \begin{pmatrix} Q & R \\ Z & 0 \end{pmatrix} = \begin{pmatrix} \pm I & \pm \hat{A} \\ 0 & B \end{pmatrix} \begin{matrix} k \\ n \end{matrix},$$

where $\begin{pmatrix} Q^T & Z^T \end{pmatrix}$ has orthogonal rows, $Z \in \Re^{k \times k}$ and $R \in \Re^{n \times n}$ are upper triangular matrices, G is orthogonal and $\hat{A}$ denotes the data deleted from the original data matrix. Within the context of the numerical solution of the GLM, a *generalised* QRD (GQRD) of the full column rank $Z \in \Re^{n \times k}$ $(n \geq k)$ and an $n \times n$ upper triangular matrix R is computed [1, 11]. The GQRD of Z and R is given by

$$Q^T Z = \begin{pmatrix} R_z \\ 0 \end{pmatrix} \begin{matrix} k \\ n-k \end{matrix}; \qquad (Q^T R)P = R_n,$$

where Q and P are $n \times n$ orthogonal matrices, and R_z and R_n are upper triangular [14, 16]. It can be observed that the similarity between the rank-k UQRD problem and the above factorizations is the retriangularization of a triangular matrix after it has been premultiplied by the orthogonal matrix of a QRD. Block generalization of the parallel strategies reported in [10, 11] are currently considered for solving the downdating and GLM problems by exploiting their special properties.

REFERENCES

[1] E. ANDERSON, Z. BAI, AND J. DONGARRA, *Generalized QR factorization and its applications*, Linear Algebra and its Applications, 162 (1992), pp. 243–271.

[2] M. W. BERRY, J. DONGARRA, AND Y. KIM, *A parallel algorithm for the reduction of a nonsymmetric matrix to block upper-Hessenberg form*, Parallel Computing, 21 (1995), pp. 1189–1211.

[3] M. COSNARD AND M. DAOUDI, *Optimal algorithms for parallel Givens factorization on a coarse-grained PRAM*, Journal of the ACM, 41 (1994), pp. 399–421.

[4] M. COSNARD, J. M. MULLER, AND Y. ROBERT, *Parallel QR decomposition of a rectangular matrix*, Numerische Mathematik, 48 (1986), pp. 239–249.

[5] L. ELDEN AND H. PARK, *Block downdating of least squares solutions*, SIAM Journal of Matrix Analysis and Applications, 15 (1994), pp. 1018–1034.

[6] P. E. GILL, G. H. GOLUB, W. MURRAY, AND M. A. SAUNDERS, *Methods for modifying matrix factorizations*, Mathematics of Computation, 28 (1974), pp. 505–535.

[7] G. H. GOLUB AND C. F. V. LOAN, *Matrix computations*, North Oxford Academic, 1983.

[8] E. J. KONTOGHIORGHES, *Algorithms for linear model estimation on massively parallel systems*, PhD Thesis, University of London, 1993. (Also TR-655, Dept. of Computer Science, Queen Mary and Westfield College, University of London).

[9] ———, *New parallel strategies for block updating the QR decomposition*, Parallel Algorithms and Applications, 5 (1995), pp. 229–239.

[10] E. J. KONTOGHIORGHES AND M. R. B. CLARKE, *Solving the updated and downdated ordinary linear model on massively parallel SIMD systems*, Parallel Algorithms and Applications, 1 (1993), pp. 243–252.

[11] ———, *Solving the general linear model on a SIMD array processor*, Computers and Artificial Intelligence, 14 (1995), pp. 353–370.

[12] S. KOUROUKLIS AND C. C. PAIGE, *A constrained least squares approach to the general Gauss–Markov linear model*, Journal of the American Statistical Association, 76 (1981), pp. 620–625.

[13] J. J. MODI AND M. R. B. CLARKE, *An alternative Givens ordering*, Numerische Mathematik, 43 (1984), pp. 83–90.

[14] B. D. MOOR AND P. V. DOOREN, *Generalizations of the singular value and QR decompositions*, SIAM Journal of Matrix Analysis and Applications, 13 (1992), pp. 993–1014.

[15] C. C. PAIGE, *Numerically stable computations for general univariate linear models*, Communications on Statistical and Simulation Computation, 7 (1978), pp. 437–453.

[16] ———, *Some aspects of generalized QR factorization*, in Reliable Numerical Computations, M. Cox and S. Hammarling, eds., Clarendon Press, Oxford, 1990.

[17] A. H. SAMEH AND D. J. KUCK, *On stable parallel linear system solvers*, Journal of the ACM, 25 (1978), pp. 81–91.

[18] G. M. SHROFF AND C. H. BISHOF, *Adaptive condition estimation for rank-one updates of QR factorizations*, Siam Journal of Matrix Analysis and Applications, 13 (1992), pp. 1264–1278.