Indian J Pure Appl Math (2025) 56:1529–1541
https://doi.org/10.1007/s13226-024-00678-1

ORIGINAL RESEARCH

Generalized elimination method and its applications to matrix factorizations

E. Golpar-Raboky

Received: 13 June 2022 / Accepted: 15 July 2024 / Published online: 19 September 2024

Abstract An elimination method is a sequence of elementary row or column operations to divide a matrix into parts with partial zeroing of columns or rows of the matrix leading to a matrix factorization implicitly. We present a generalized elimination algorithm for solving real and Diophantine linear systems. The algorithm uses a zeroing process based on two index sets and provides a mechanism for constructing factorizations such as block LU factorization as well as the WZ and the ZW factorizations under a common framework. We also introduce the octant interlocking factorization method and present two new factorizations named as the ODS and the SDO factorizations. Necessary and sufficient conditions under which the generalized elimination method is carried out are proposed.

Keywords Generalized elimination method · Matrix factorization · Quadrant interlocking factorization · Octant interlocking factorization

Mathematics Subject Classification 15A03 · 15A21 · 15A23 · 34C20 · 49M27

1 Introduction

A block matrix is a matrix whose entries are themselves matrices. Linear algebra is best done with block matrices. Block matrices appear in most modern applications of linear algebra because the notation highlights essential structures in matrix theory. Matrix operations on block matrices can be carried out by treating the blocks as matrix entries. A block algorithm in matrix computation is defined by operations on submatrices rather than matrix elements. Theoretical and practical studies on the numerical stability, complexity, pivoting strategies, parallel implementation and existence conditions of matrix decompositions have been done [5,6,18,20].

The numerical solution of linear systems on the multicore architecture is an important issue for high performance computing. Blocking is a well-known optimization technique for improving the effectiveness of memory hierarchies. See [3,5,15,22,26].

Here, we present a generalized block elimination method which produces some interesting matrix factorizations. Let $\mathbb{R}$ and $\mathbb{R}^{m\times n}$ denote the set of real numbers and the set of $m \times n$ real matrices, respectively.

Definition 1.1 Let $[n] = \{1, \dots, n\}$ and $\alpha_i \subset [n]$, for $i = 1, \dots, s$. We say $\alpha = \{\alpha_1, \dots, \alpha_s\}$ is an index set of $[n]$ if and only if $\alpha_i \bigcap \alpha_j = \emptyset$, whenever $i \neq j$, and $\cup_{i=1}^{s}\alpha_i = [n]$. We denote cardinality of t by $|t|$.

Communicated by Rahul Roy.

E. Golpar-Raboky (✉)
Department of Mathematics, University of Qom, Qom, Iran
E-mail: g.raboky@qom.ac.ir

Let $A \in \mathbb{R}^{n\times n}$ and $\alpha = \{\alpha_1, \ldots, \alpha_s\}$ and $\beta = \{\beta_1, \ldots, \beta_s\}$ be two index sets of $[n]$. Assume that $A(:, \beta_j)$ denote the $n \times |\beta_j|$ submatrix of A containing the columns specified by β_j, $A(\alpha_i, \beta_j)$ denote the $|\alpha_i| \times |\beta_j|$ submatrix of A composed of the rows specified by α_i and the columns specified by β_j.

Note: Throughout the remainder of the paper, we assume that $\alpha = \{\alpha_1, \ldots, \alpha_s\}$, $\beta = \{\beta_1, \ldots, \beta_s\}$ are two index sets of $[n]$ such that $|\alpha_i| = |\beta_i|$ and A is a $s \times s$ block matrix. $A(\alpha_i, \beta_j)$, $1 \le i, j \le s$, denote the (i, j)th block of A respect to the index sets α and β.

The leading principal submatrices of A with respect to the index sets α and β are defined as:

$$A(\cup_{i=1}^{k}\alpha_i, \cup_{j=1}^{k}\beta_j) = \begin{pmatrix} A(\alpha_1, \beta_1) & \ldots & A(\alpha_1, \beta_k) \\ \vdots & \ddots & \vdots \\ A(\alpha_k, \beta_1) & \ldots & A(\alpha_k, \beta_k) \end{pmatrix}, \quad k = 1, \ldots, s.$$

Let $P_\alpha = (e_{\alpha_1}, \ldots, e_{\alpha_s})$ and $P_\beta = (e_{\beta_1}, \ldots, e_{\beta_s})$ denote two partial permutation matrices with respect to the index sets α and β, respectively. Define

$$A_{\alpha,\beta} = P_\alpha^T A P_\beta = \begin{pmatrix} A(\alpha_1, \beta_1) & \ldots & A(\alpha_1, \beta_s) \\ \vdots & \ddots & \vdots \\ A(\alpha_s, \beta_1) & \ldots & A(\alpha_s, \beta_s) \end{pmatrix}. \tag{1.1}$$

We recall that a partitioned nonsingular matrix $A \in \mathbb{R}^{n\times n}$ is said to be block strongly nonsingular if every leading principal block submatrices of A are nonsingular. A block strongly nonsingular matrix has a block LU decomposition [14]. Now, we extend the definition as follows.

Definition 1.2 The matrix $A \in \mathbb{R}^{n\times n}$ is said to be (α, β)-block strongly nonsingular, if and only if $A(\cup_{i=1}^{k}\alpha_i, \cup_{i=1}^{k}\beta_i)$, for $k = 1, \ldots, s$, are nonsingular.

Definition 1.3 [7] The matrix $A \in \mathbb{R}^{n\times n}$ is called column (α, β)-block diagonally dominant (with respect to the matrix norm $\|.\|$) if and only if $A(\alpha_i, \beta_i)$ are nonsingular, and

$$\|A^{-1}(\alpha_j, \beta_j)\|^{-1} - \sum_{j\neq i} \|A(\alpha_i, \beta_j)\| \ge 0, \;\; j = 1, \ldots, s. \tag{1.2}$$

If a strict inequality holds in (1.2) then A is called (α, β)-block strictly diagonally dominant by columns(with respect to the matrix norm $\|.\|$). The matrix A is block diagonally dominant by rows if A^T is block diagonally dominant by columns.

Let A be a (α, β)-block strictly diagonally dominant matrix. Then, $A(\cup_{i=1}^{k}\alpha_i, \cup_{i=1}^{k}\beta_i)$ are nonsingular, for $k = 1, \ldots, s$, and A is a (α, β)-block strongly nonsingular matrix.

Remark 1.1 [18] According to (1.1), $A(\alpha_i, \beta_j)$ is the (i, j)th block of $A_{\alpha,\beta}$ which is also the (i, j)th block of A respect to the index sets α and β. Thus, terms and results that we use to describe the block matrix $A_{\alpha,\beta}$ can be extended to A with respect to the index sets

α and β, by applying two partial permutation matrices P_α and P_β. For instance, the matrix A is (α, β)-block strongly nonsingular ((α, β)-block diagonally dominant) if and only if $A_{\alpha,\beta}$ is block strongly nonsingular (block diagonally dominant).

Example 1.1 Let $\alpha = \{\{3\}, \{2\}, \{1\}\}$, $\beta = \{\{1\}, \{2\}, \{3\}\}$ and

$$A = \begin{pmatrix} 1 & 1 & 3 \\ 1 & 3 & 1 \\ 4 & 1 & 1 \end{pmatrix}.$$

Then,

$$A_{\alpha,\beta} = \begin{pmatrix} 4 & 1 & 1 \\ 1 & 3 & 1 \\ 1 & 1 & 3 \end{pmatrix}.$$

This shows that A is (α, β)-block strongly nonsingular and (α, β)-block strictly diagonally matrix.

Block elementary row and column operations play important roles in linear algebra. The elementary row operations can be generalized as follows. In cases $(i)-(iii)$ below, assume that $|\alpha_i| = |\alpha_j|$.

(*i*) Row switching: a row block within the matrix can be interchanged with another row block:

$$A(\alpha_j, :) \rightleftarrows A(\alpha_i, :),$$

(*ii*) Row multiplication: each element in a row block can be multiplied by a nonzero constant. Let $n_i = |\alpha_i|$ and $M \in \mathbb{R}^{n_i \times n_i}$ be a diagonal matrix, with diagonal entries $m_k \neq 0$,

$$M(k, j) = \begin{cases} m_k & k = j \\ 0 & k \neq j. \end{cases}$$

Then,

$$MA(\alpha_i, :) \to A(\alpha_i, :),$$

is the matrix produced from A by multiplying the row block $A(\alpha_i, :)$ by $m = (m_1, \cdots, m_{n_i})$.

(*iii*) Row addition: a row block can be the sum of that row block and a multiple of another row block:

$$A(\alpha_i, :) + MA(\alpha_j, :) \to A(\alpha_i, :),$$

where, the matrix M is defined in (*ii*). An elimination method is a sequence of elementary row or column operations to divide a matrix into parts with partial zeroing of columns or rows of the matrix leading to a matrix factorization implicitly. In doing this, we need a zeroing process which depends on choosing the index sets t and b as follows:

Let $A^{(1)} = A \in \mathbb{R}^{n\times n}$, $\alpha = \{\alpha_1, \ldots, \alpha_s\}$ and $\beta = \{\beta_1, \ldots, \beta_s\}$ be the index sets of $[n]$, where $|\alpha_i| = |\beta_i| = n_i$, $i = 1, \ldots, s$ and $\sum_{i=1}^{s} n_i = n$. In the kth step we have the matrix $A^{(k)} = M_{k-1} \ldots M_1 A$. If $\det(A^{(k)}(\alpha_k, \beta_k)) \neq 0$, then we compute $M_k A^{(k)}$ for zeroing $A^{(k)}(\alpha_k^c, \beta_k)$, where $\alpha_k^c = [n] - \cup_{i=1}^{k} \alpha_i$. Define $r_k \in \mathbb{R}^{n\times|\beta_k|}$ by

$$r_k(\cup_{i=1}^{k}\alpha_i, :) = 0, \qquad r_k(\alpha_k^c, :) = A^{(k)}(\alpha_k^c, \beta_k) A^{(k)}(\alpha_k, \beta_k)^{-1}, \tag{1.3}$$

and the elementary matrix by

$$M_k = I - r_k I(:, \alpha_k)^T. \tag{1.4}$$

A matrix of the form (1.4) is an elimination transformation and $M_k^{-1} = I + r_k I(:, \alpha_k)^T$. Moreover,

$$M_k A^{(k)} = (I - r_k I(:, \alpha_k)^T) A^{(k)} = A^{(k)} - r_k A^{(k)}(\alpha_k, :), \tag{1.5}$$

and

$$I(\alpha_k^c, :) M_k A I(\alpha_k^c, :) = 0.$$

By repeatedly applying (1.3)–(1.5) we generate a zeroing procedure to obtain a matrix factorization of A. If no singular pivot are encountered, then $\det(A^{(k)}(\alpha_k, \beta_k)) \neq 0$, and $M_{s-1} \ldots M_1 A = C$, then

$$A = BC, \tag{1.6}$$

where

$$\begin{aligned} B &= M_1^{-1} \ldots M_{s-1}^{-1} \\ &= (I - r_1 I(:, \alpha_1)^T)^{-1} \ldots (I - r_{s-1} I(:, \alpha_{s-1})^T)^{-1} \\ &= (I + r_1 I(:, \alpha_1)^T) \ldots (I + r_{s-1} I(:, \alpha_{s-1})^T) \\ &= I + \textstyle\sum_{i=1}^{s-1} r_i I(:, \alpha_i)^T. \end{aligned} \tag{1.7}$$

From the elimination process (1.3)-(1.5), we have

$$B(\alpha_k^c, \beta_k) = r_k(\alpha_k^c, :), \quad C(\alpha_k^c, \beta_k) = 0, \tag{1.8}$$

for $k = 1, \ldots, s-1$.

Here, we present the generalized elimination method for solving the linear system $Ax = d$ and computing the matrix factorization $A = BC$.

Algorithm 1.1 ***Generalized Elimination Algorithm*** *(GEA)*

Input: *$A \in \mathbb{R}^{n\times n}$, $d \in \mathbb{R}^n$ and two index sets $\alpha = \{\alpha_1, \dots, \alpha_s\}$ and $\beta = \{\beta_1, \dots, \beta_s\}$, where $|\alpha_i| = |\beta_i|$, $i = 1, \dots, s$.*

Output: *The matrix factorization $A = BC$ and a solution to $Ax = d$.*

(1) Let $A^{(1)} = A$, $d^{(1)} = d$ and $B = I$, where I is an $n \times n$ identity matrix.
(2) ***for*** *$k = 1, \dots, s-1$ do*

$$\begin{aligned} r_k(\alpha_k, :) = 0,\ r_k(\alpha_k^c, :) &= A^{(k)}(\alpha_k^c, \beta_k) A^{(k)}(\alpha_k, \beta_k)^{-1}, \\ M_k &= I - r_k I(:, \alpha_k)^T,\ B = B + r_k I(:, \alpha_k)^T, \\ A^{(k+1)} &= M_k A^{(k)},\ d^{(k+1)} = M_k d^{(k)}. \end{aligned}$$

end for
(3) Set $C = A^{(s)}$, which solves $Cx = d^{(s)}$ and compute the matrix factorization $A = BC$.

This partitioned algorithm does precisely the same arithmetic operations as any other variant of block Gaussian elimination [18,20]. Algorithm 1.1 is an alternative to block Gaussian elimination with respect to the index sets α and β. Different choices of the parameters will result in different decompositions.

Using (1.1) and Theorem 13.2 in [20], we provide a necessary and sufficient condition under which Algorithm 1.1 can be carried out.

Theorem 1.1 (Factorization theorem) *Let $A \in \mathbb{R}^{n\times n}$. The generalized elimination algorithm can be performed on A to produce the matrix factorization (1.6) if and only if A is a (α, β)-block strongly nonsingular matrix. Moreover, the factorization is unique.*

Proof Suppose $k-1$ steps in Algorithm 1.1 have been executed. At the beginning of step k the matrix A has been overwritten by $M_{k-1} \dots M_1 A = A^{(k-1)}$ and we have

$$|det(A(\cup_{i=1}^k \alpha_i, \cup_{i=1}^k \beta_i))| = |\prod_{i=1}^k det(A^{(k-1)}(\alpha_i, \beta_i))|.$$

Thus, if $A(\cup_{i=1}^k \alpha_i, \cup_{i=1}^k \beta_i)$ is nonsingular, then the kth pivot $A^{(k)}(\alpha_k, \beta_k)$ is also nonsingular.

Now, let be (α, β)-block strongly nonsingular. Then, $A_{\alpha,\beta}$ defined by (1.1) is a block strongly nonsingular matrix and has a unique block LU factorization [18]. This, results in the uniqueness of matrix decomposition of Algorithm 1.1.

It turns out the kth pivot, $A^{(k)}(\alpha_k, \beta_k)$ is singular if $A(\cup_{i=1}^k \alpha_i, \cup_{i=1}^k \beta_i)$ is singular. Therefore, Algorithm 1.1 will fail if the pivot submatrices $A^{(k)}(\alpha_k, \beta_k)$ are singular for some $1 \le k \le s-1$.

Let $A \in \mathbb{R}^{n\times n}$ be a (α, β)-block strongly nonsingular matrix. Then we have the matrix factorization $A = BC$ from Algorithm 1.1 and a block LU factorization $A_{\alpha,\beta} = LU$. Since, $A = P_\alpha A_{\alpha,\beta} P_\beta^T$ we have

$$\begin{aligned} BC &= P_\alpha LU P_\beta^T = (P_\alpha L P_\alpha^T)(P_\alpha P_\beta^T)(P_\beta U P_\beta^T) = L_{\alpha,\alpha} D_{\alpha,\beta} U_{\beta,\beta}, \\ BC &= P_\alpha LU P_\beta^T = (P_\alpha L P_\beta^T)(P_\beta P_\alpha^T)(P_\alpha U P_\beta^T) = L_{\alpha,\beta} D_{\beta,\alpha} U_{\alpha,\beta}. \end{aligned} \tag{1.9}$$

Therefore, the generalized factorization $A = BC$ can be obtained from a block LU factorization of $A_{\alpha,\beta}$.

Remark 1.2 The matrix B is invertible such that $B(\alpha(i), \beta(i)) = I$, for $i = 1, \dots, s$ and according to (1.7), B^{-1} has the same structure as B. Moreover, if $\alpha_i = \beta_i$, for $i = 1, \dots, s$, then C has the same structure as B^T.

There are two main possibilities to compute $r_k(\alpha_k^c, :)$ in step 2 of Algorithm 1.1. We can apply Gaussian elimination with partial pivoting to factorize $A^{(k)}(\alpha_k, \beta_k)$ or compute $A^{(k)}(\alpha_k, \beta_k)^{-1}$ explicitly, which is attractive for parallel machines (see Theorem 13.6 in [20]).

One class of matrices for which GEA can be performed is (α, β)-block diagonally dominant. According to (1.9) and Theorems 13.6, 13.7 and 13.8 in [20], we have the next results.

Theorem 1.2 *Suppose $A \in \mathbb{R}^{n\times n}$ is nonsingular and (α, β)-block diagonally dominant by columns with respect to a subordinate matrix norm. Then, GEA can be performed on A to produce the matrix factorization (1.6). Furthermore, if $A^{(k)}$ denotes the matrix obtained after $k-1$ steps of Algorithm 1.1, then*

$$\max_{k\le i,j\le s} \|A^{(k)}(\alpha_i, \beta_j)\|_2 \le 2 \max_{1\le i,j\le s} \|A(\alpha_i, \beta_j)\|_2 \tag{1.10}$$

Proof According to (1.1), $A_{\alpha,\beta}$ is a block diagonally dominant matrix and has a block LU factorization. It concludes that GEA can be performed on A. The proof of (1.10) follows the lines of the proof for Theorem 13.8 in [20] for partitioned matrix A with respect to the index sets α and β.

For a general matrix the level of instability can be bounded in terms of the condition number and the growth factor for Gaussian elimination without pivoting. A consequence is that the factorization is stable for a matrix A that is symmetric positive definite or point diagonally dominant by rows or columns as long as A is well-conditioned (see [20] for details).

In the following, one main result of Algorithm 1.1 for solving the linear Diophantine system

$$Ax = d,\ A \in \mathbb{Z}^{n\times n},\ x \in \mathbb{Z}^n,\ d \in \mathbb{Z}^n, \tag{1.11}$$

is presented.

Definition 1.4 $A \in \mathbb{Z}^{n\times n}$ is a unimodular matrix if and only if $|\det(A)| = 1$.

Note that if A is unimodular then A^{-1} is also unimodular.

Definition 1.5 We say that $A \in \mathbb{Z}^{n\times n}$ is (α, β)-block strongly unimodular if and only if $A(\cup_{i=1}^{k}\alpha_i, \cup_{i=1}^{k}\beta_i)$, for $k = 1, \dots, s$, are unimodular.

According to Theorem 1.1, we have the next result.

Theorem 1.3 *Let $A \in \mathbb{Z}^{n\times n}$ be (α, β)-block strongly unimodular and $b \in \mathbb{Z}^n$. Then, Algorithm 1.1 produces an integer solution for the linear Diophantine system (1.11) as well as an integer factorization of A.*

Proof Using the process of proving Theorem 1.1, we can show $A^{(k)}$ are integer matrices and $|det(A^{(k)}(\alpha_k, \beta_k))| = 1$, for $k = 1, \dots, s$. Since, M_k are unimodular, for $k = 1, \dots, s$, then C and B are integer matrices. This completes the proof.

Different choices of the index sets α and β in Algorithm 1.1 result in various matrix factorizations. We show how to choose the parameters for computing the LU, WZ and ZW factorizations. We also present two new quadrant interlocking factorizations and show how to set t and b to compute the factorizations.

2 2 × 2 block matrix factorizations

Here, we introduce two index sets of $[n]$.

The first index set is $\alpha = \{\alpha_1, \dots, \alpha_n\}$, such that

$$\alpha_i = \begin{cases} \frac{i+1}{2}, & if\ i\ is\ odd \\ n - \frac{i}{2} + 1, & if\ i\ is\ even. \end{cases} \tag{2.12}$$

We define the second index set $\beta = \{\beta_1, \dots, \beta_n\}$ as follows. If n is an even number, define

$$\beta_i = \begin{cases} \frac{n}{2} - \frac{i+1}{2} + 1, & if\ i\ is\ odd \\ \frac{n}{2} + \frac{i}{2}, & if\ i\ is\ even, \end{cases} \tag{2.13}$$

and if n is an odd number, define

$$\beta_i = \begin{cases} \frac{n+1}{2} - \frac{i}{2}, & if\ i\ is\ even \\ \frac{n+1}{2} + \frac{i-1}{2}, & if\ i\ is\ odd. \end{cases} \tag{2.14}$$

```
function [B,C]=BlockGEA(A,a,b)
clc
[n,n]=size(A);
B=eye(n);
for k=2:2:n-1
 Mk=eye(n) ;
 Mk(a(k+1:n),a(k-1:k))=-A(a(k+1:n),b(k-1:k))*inv(A(a(k-1:k),b(k-1:k)));
 B(a(k+1:n),a(k-1:k))=A(a(k+1:n),b(k-1:k))*inv(A(a(k-1:k),b(k-1:k)));
 A=Mk*A;
end
C=A;
end
```

Fig. 1 Matlab code for computing the 2×2 block factorization

We can also define α_k and β_k from cardinal 2 and compute some special block matrix factorizations. If n is an even number, set $s = \frac{n}{2}$ and define

$$\alpha_k = \{k, n-k+1\},\ \beta_k = \{s-k+1, s+k\},\ k = 1, \ldots, s, \tag{2.15}$$

and if n is an odd number, set $s = \frac{n+1}{2}$ and define

$$\alpha_k = \{k, n-k+1\},\ \alpha_s = \{s\},\ \beta_1 = \{s\},\ \beta_{k+1} = \{s-k, s+k\},\ k = 1, \ldots s-1. \tag{2.16}$$

Here, we give a MATLAB code for Algorithm 1.1, where $A \in \mathbb{R}^{n\times n}$, a and b are two index sets and all the blocks are of size 2×2.

2.1 Quadrant interlocking factorization

The quadrant interlocking factorization (*QIF*) is an alternative to the *LU* factorization, well suited for implementation on parallel computers. Several versions of this factorization have been presented: [8,10–13,19,25]. A direct method, called WZ factorization method, for the solution of the linear system $Ax = b$ was introduced by Evans and Hotzopoulos [8]. The structure of the factors is of butterfly form. The *WZ* factorization conveniently offers a parallel method for solving dense square linear systems and can be faster than the *LU* factorization for large matrices [1,2,23,24]. Several algorithms have been proposed for computing the WZ factorization. One, known as the parallel implicit elimination (PIE), was introduced by Evans [9]. The algorithm uses an elimination method by zeroing the elements of the first and the last columns, simultaneously. Raboky and Mahdavi-Amiri presented several algorithms for computing the WZ factorization, using basic ABS algorithms [16], block scaled ABS algorithms [17] and the Wedderburn rank reduction formula [21].

Here, we introduce the WZ factorization and produce the factorization using the generalized elimination method.

Definition 2.1 Consider two index sets α and β defined by (2.15) or (2.16). A block matrix A is called a W-matrix if the possible nonzero elements $A(\beta_i, \beta_j)$ have $j \geq i$ and it is called a Z-matrix if the possible nonzero elements have $j \geq i$. These matrices have the following forms:

with the gray parts standing for possible nonzeros.

Consider two $n \times n$ permutation matrix using (2.15) and (2.16) as follows:

$$P_\alpha = [e_{\alpha_1}, \ldots, e_{\alpha_s}],\ P_\beta = [e_{\beta_1}, \ldots, e_{\beta_s}] \tag{2.17}$$

where, e_i be the ith column of an $n \times n$ identity matrix.

Let $L, U \in \mathbb{R}^{n\times n}$ be block lower and upper triangular matrices with 2×2 block diagonals, respectively. Then,

$$W = P_\alpha L P_\alpha^T,\ Z = P_\alpha U P_\alpha^T, \tag{2.18}$$

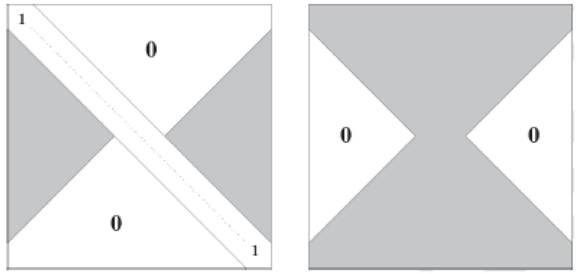


Fig. 2 Structures of the unit **W**-matrix (left) and **Z**-matrix (right)

and

$$Z = P_\beta L P_\beta^T, \; W = P_\beta U P_\beta^T, \tag{2.19}$$

where W is a W-matrix and Z is a Z-matrix. Triangular matrices are similar to W and Z matrices and any structural property of W and Z implies a consequent structural property on L and U [4].

Definition 2.2 We say that a matrix A is factorized in the *WZ* form if

$$A = WZ, \tag{2.20}$$

where the matrix W is a W-*matrix* and Z is a Z-*matrix*.

Here, we show how to choose the index sets such that Algorithm 1.1 computes a WZ factorization. Consider two index sets α and β defined by (2.15) if n is even and (2.16) if n is odd. Let A be a W-matrix, then there exist the possible nonzero elements $A(\alpha_i, \alpha_j)$ for $j \leq i$ and if A is a Z-matrix, then there exist the possible nonzero elements $A(\beta_i, \beta_j)$ for $j \leq i$.

Let $\alpha_k = \beta_k = \{k, n-k+1\}$ and $\alpha_k^c = \beta_k^c = \{k+1 : n-k\}$, for $k = 1, \ldots s$. Then, Algorithm 1.1, zeros $\{k+1 : n-k\}$th elements in the kth column and the $(n-k+1)$th column, simultaneously. This zeroing process leads to a WZ factorization for A as follows.

Theorem 2.1 *Let $A \in \mathbb{R}^{n \times n}$ and the index set α is defined by (2.15) (or (2.16)). Then, GEA computes a WZ factorization if and only if A is (α, α)-block strongly nonsingular. Moreover, the factorization is unique.*

Proof The proof of the existence and uniqueness of WZ factorization follows the lines of the proof for Theorem 1.1. From (1.9), since $P_\alpha P_\alpha^T = I$, we have

$$BC = (P_\alpha L P_\alpha^T)(P_\alpha P_\alpha^T)(P_\alpha U P_\alpha^T),$$

and by (2.18), we have

$$A = BC = WZ$$

The function $BlockGen(A, a, a)$ produces a WZ factorization of A, where a is defined by (2.12).

Definition 2.3 We say that a matrix A is factorized in the form *ZW* if

$$A = ZW, \tag{2.21}$$

where W is a W-matrix, Z is a Z-matrix.

Now, let $\alpha_k = \beta_k = \{s-k+1, s+k\}$ and $\alpha_k^c = \beta_k^c = \{1 : s-k, s+k+1 : n\}$, for $k = 1, \ldots s$. Then, Algorithm 1.1 zeros $1 : s-k-1, s+k : n$th elements in the $(s-k)$th column and the $(s+k-1)$th column, simultaneously. This zeroing process leads to a ZW factorization for A.

Theorem 2.2 *Let $A \in \mathbb{R}^{n \times n}$ and the index set β defined by (2.15) (or (2.16)). Then, GEA computes an ZW factorization if and only if A is (β, β)-block strongly nonsingular. Moreover, the factorization is unique.*

Proof The proof of the existence and uniqueness of ZW factorization follows the lines of the proof for Theorem 1.1. From (1.1), since $P_\beta P_\beta^T = I$, we have

$$BC = (P_\beta L P_\beta^T)(P_\beta P_\beta^T)(P_\beta U P_\beta^T),$$

and by (2.19), we have

$$A = BC = ZW$$

The function $BlockGen(A, b, b)$ produces a ZW factorization of A, where b is defined by (2.13) or (2.14).

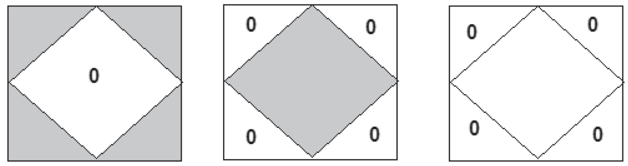


Fig. 3 Structures of the S-matrix (left), O-matrix (middle) and D-matrix (right)

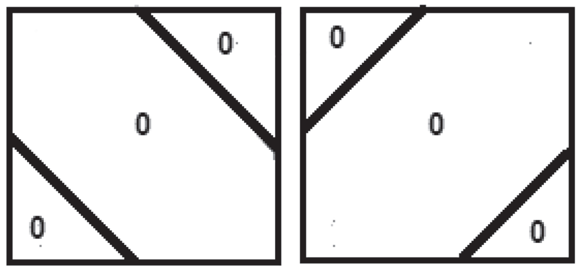


Fig. 4 Special structures of the D-matrix

2.2 Octant interlocking factorization

In this section, we describe the octant interlocking factorization (OIF) method and its existence conditions. The OIF method factorizes A such that the structure of the factors is of rhombus form.

Definition 2.4 Let $\mathbb{R}^{n\times n}$. Consider two index sets α and β defined by (2.15) or (2.16). A matrix A is called a S-matrix if the possible nonzero elements $A(\alpha_i, \beta_j)$ have $j \geq i$ and it is called a O-matrix if the possible nonzero elements $A(\beta_i, \alpha_j)$ have $j \geq i$. A matrix which is both S-matrix and O-matrix, in one of the two forms of Figure 3, is called a D-matrix. These matrices have the following forms:

with the gray parts standing for possible nonzeros.

The OIF is an alternative to the LU factorization. Let L and U be block lower and upper triangular matrices with 2×2 block diagonals, respectively. Then, we have

$$O = P_\alpha L P_\beta^T,\ S = P_\alpha U P_\beta^T, \tag{2.22}$$

and

$$S = P_\beta L P_\alpha^T,\ O = P_\beta U P_\alpha^T, \tag{2.23}$$

where O is an O-matrix, S is an S-matrix. The main idea of the OIF method consists of factorizing A in the form

$$A = ODS \quad or \qquad A = SDO. \tag{2.24}$$

The equations (2.18), (2.19), (2.22) and (2.23) provide interesting relationships between L, U, W, Z, S, O and D matrices as follows:

$$\begin{array}{l} W = P_\alpha L P_\alpha^T,\ W = P_\beta U P_\beta^T\ \ W = DS,\ W = OD, \\ Z = P_\alpha U P_\alpha^T,\ Z = P_\beta L P_\beta^T,\ Z = SD \quad Z = DO, \end{array} \tag{2.25}$$

where $D = P_\alpha P_\beta^T$ or $D = P_\beta P_\alpha^T$ have the following structures:

From Theorem 1.1, we have a necessary and sufficient condition for the existence of an ODS factorization.

Theorem 2.3 *Let $A \in \mathbb{R}^{n\times n}$ and the index sets defined by (2.15) (or (2.16)). Then, GEA computes an ODS factorization if and only if A is a (α, β)-block strongly nonsingular matrix. Moreover, the factorization is unique.*

```
function [B,C]=GenGauss(A,a,b)
[n,n]=size(A);
E=eye(n);
B=E;
for k=1:1:n-1
rk=zeros(n,1);
for j=k:n-1
    rk(a(j+1),:)=A([a(j+1)],b(k))*inv(A(a(k),b(k)));
end
    Mk=eye(n)-rk*(E(:,a(k)))';
    B=B+rk*(E(:,a(k)))';
    A=Mk*A ;
end;
C=A;
end
```

Fig. 5 Matlab code for Algorithm 1.1

Proof The proof of the existence and uniqueness of ODS factorization follows the lines of the proof for Theorem 1.1. From (1.9) we have

$$BC = (P_\alpha L P_\alpha^T)(P_\alpha P_\beta^T)(P_\beta U P_\beta^T),$$

and by (2.25), we have

$$A = BC = WDW = WS = WD^{-1}DS = ODS.$$

The function $BlockGen(A, a, b)$ produces a WS factorization of A, where a and b defined by (2.12) and (2.13) (or (2.14)), respectively.

Corollary 2.1 *Let $A \in \mathbb{Z}^{n\times n}$. If α and β are two index sets defined by (2.15) and (2.16). Then GEA computes an integer ODS factorization if and only if A is a (α, β)-block strongly unimodular matrix.*

The following theorem provides a necessary and sufficient condition for the existence of an SDO factorization.

Theorem 2.4 *Let $A \in \mathbb{R}^{n\times n}$ and the index sets defined by (2.15) (or 2.16). Then, GEA computes an SDO factorization if and only if A is a (β, α)-block strongly nonsingular matrix.*

Proof The proof of the existence and uniqueness of SDO factorization follows the lines of the proof for Theorem 1.1. From (1.9) we have

$$BC = (P_\beta L P_\beta^T)(P_\beta P_\alpha^T)(P_\alpha U P_\alpha^T).$$

From (2.25), we have

$$A = BC = ZDZ = ZO = ZD^{-1}DO = SDO.$$

The function $BlockGen(A, b, a)$ produces a ZO factorization of A, where a and b defined by (2.12) and (2.13)(or (2.14)), respectively.

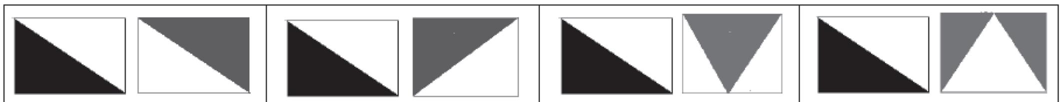
3 Generalized algorithm and some matrix factorizations

Let $A \in \mathbb{R}^{n\times n}$, and $\alpha = \{\alpha_1, \ldots, \alpha_s\}$ and $\beta = \{\beta_1, \ldots, \beta_s\}$ be two index sets of $[n]$. Here, we set the parameters in Algorithm 1.1 to present the associated matrix factorizations $A = BC$.

Here, we give a MATLAB code for Algorithm 1.1, where $A \in \mathbb{R}^{n\times n}$, and a and b are two index sets defined by (2.12) and (2.13)(or (2.14)) , respectively and all the blocks are of size 1×1.

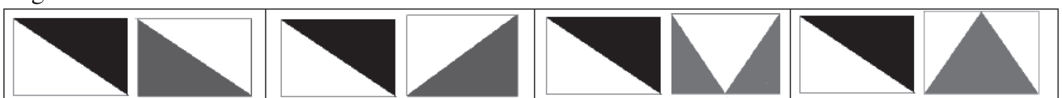
We apply four index sets $\{1, \ldots, n\}$, $\{n, \ldots, 1\}$, $\{\alpha_1, \ldots, \alpha_n\}$ defined by (2.15) and $\{\beta_1, \ldots, \beta_n\}$ defined by (2.16), we compute the following matrix factorizations.

Case (1): Let the first index set equals to $\{1, \ldots, n\}$. By applying the other three index sets we have the following matrix factorizations:



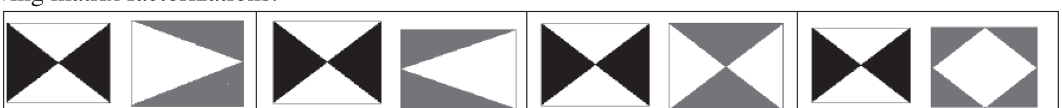
GenGauss(A,1:n,1:n) GenGauss(A,1:n,n:-1:1) GenGauss(A,1:n,a) GenGauss(A,1:n,b)

Case (2): Let the first index set equals to $\{n, \ldots, 1\}$. By applying the other three index sets we have the following matrix factorizations:



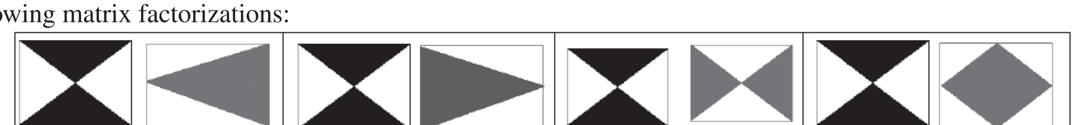
GenGauss(A,n:-1:1,n:-1:1) GenGauss(A,n:-1:1,1:n) GenGauss(A,n:-1:1,b) GenGauss(A,n:-1:1,a)

Case (3): Let the first index set equals to $\{\alpha_1, \ldots, \alpha_n\}$. By applying the other three index sets we have the following matrix factorizations:



GenGauss(A,a,1:n) GenGauss(A,a,n:-1:1) GenGauss(A,a,a) GenGauss(A,a,b)

Case (4): Let the first index set equals to $\{\beta_1, \ldots, \beta_n\}$. By applying the other three index sets we have the following matrix factorizations:



GenGauss(A,b,1:n) GenGauss(A,b,n:-1:1) GenGauss(A,b,b) GenGauss(A,b,a)

4 Numerical illustrations

We give illustrations of our proposed algorithms in computing the QIF factorizations.

Example 4.1 Consider $P = pascal(8)$ as follows:

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 3 & 6 & 10 & 15 & 21 & 28 & 36 \\ 1 & 4 & 10 & 20 & 35 & 56 & 84 & 120 \\ 1 & 5 & 15 & 35 & 70 & 126 & 210 & 330 \\ 1 & 6 & 21 & 56 & 126 & 252 & 462 & 792 \\ 1 & 7 & 28 & 84 & 210 & 462 & 924 & 1716 \\ 1 & 8 & 36 & 120 & 330 & 792 & 1716 & 3432 \end{pmatrix}.$$

By executing $BlockGEA(A, b, a)$, we have an $A = ZO$ factorization of A as follows:

$$B = \begin{pmatrix} 1 & 5 & 5.6429 & 1.5667 & -0.5667 & -0.7937 & -0.3571 & 0 \\ 0 & 1 & 3 & 1.5333 & -0.5333 & -0.2778 & 0 & 0 \\ 0 & 0 & 1 & 1.4 & -0.4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2.2 & 3.2 & 1 & 0 & 0 \\ 0 & 0 & -3 & -6.6 & 7.6 & 5.3333 & 1 & 0 \\ 0 & -2.4 & -15.4286 & -14.7714 & 15.7714 & 17.5238 & 6.8571 & 1 \end{pmatrix}.$$

and

$$C = \begin{pmatrix} 0 & 0 & 0 & -0.0714 & -0.0714 & 0 & 0 & 0 \\ 0 & 0 & 0.2778 & 0.6667 & 0.8333 & 0.5556 & 0 & 0 \\ 0 & -0.6 & -2 & -4 & -6 & -7 & -5.6 & 0 \\ 1 & 4 & 10 & 20 & 35 & 56 & 84 & 120 \\ 1 & 5 & 15 & 35 & 70 & 126 & 210 & 330 \\ 0 & -1.2 & -5-12 & -21 & -28 & -25.2 & 0 & \\ 0 & 0 & 0.6667 & 2 & 3 & 2.3333 & 0 & 0 \\ 0 & 0 & 0 & -0.1143 & -0.1429 & 0 & 0 & 0 \end{pmatrix}.$$

Furthermore, we have an SDO factorization as follows:

$$A = BC = (BD)DC = SDO$$

where,

$$BD = S = \begin{pmatrix} 1.5667 & 5.6429 & 5 & 1 & 0 & -0.3571 & -0.7937 & -0.5667 \\ 1.5333 & 3 & 1 & 0 & 0 & 0 & -0.2778 & -0.5333 \\ 1.4 & 1 & 0 & 0 & 0 & 0 & 0 & -0.4 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -2.2 & 0 & 0 & 0 & 0 & 0 & 1 & 3.2 \\ -6.6 & -3 & 0 & 0 & 0 & 1 & 5.3333 & 7.6 \\ -14.7714 & -15.4286 & -2.4 & 0 & 1 & 6.8571 & 17.5238 & 15.7714 \end{pmatrix}.$$

We also have an WS factorization of A by applying $BlockGEA(A, a, b)$ as follows:

$$B = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3.4286 & 1 & 0 & 0 & 0 & 0 & 0 & 0.0048 \\ 7.1429 & 2.6667 & 1 & 0 & 0 & 0 & 0.0714 & 0.0238 \\ 11.4286 & 4 & 1.6 & 1 & 0 & 0.4 & 0.2857 & 0.0714 \\ 15 & 4 & 1 & 0 & 1 & 1 & 0.6429 & 0.1667 \\ 16 & 2.3333 & 0 & 0 & 0 & 1 & 1 & 0.3333 \\ 12 & 0 & 0 & 0 & 0 & 0 & 1 & 0.6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

and

$$C = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -2.4 & -1.5 & -0.6 & 0 & 0 & -1.2 & -4.6 & -11.8 \\ 1.2 & 0.3 & 0 & 0 & 0 & 0 & 0.7 & 3.9 \\ -0.1 & 0 & 0 & 0 & 0 & 0 & 0 & -0.1 \\ -0.1 & 0 & 0 & 0 & 0 & 0 & 0 & -0.1 \\ 1.9 & 0.6 & 0 & 0 & 0 & 0 & 2.3 & 14.7 \\ -11.6 & -9.8 & -5.6 & 0 & 0 & -25.2 & -117.6 & -355.2 \\ 1 & 8 & 36 & 120 & 330 & 792 & 1716 & 3432 \end{pmatrix},$$

Furthermore, we have an ODS factorization as follows

$$A = BC = (BD)DC = ODS$$

where,

$$BD = O = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 3.4286 & 0.0048 & 0 & 0 & 0 \\ 0 & 1 & 2.6667 & 7.1429 & 0.0238 & 0.0714 & 0 & 0 \\ 1 & 1.6 & 4 & 11.4286 & 0.0714 & 0.2857 & 0.4 & 0 \\ 0 & 1 & 4 & 15 & 0.1667 & 0.6429 & 1 & 1 \\ 0 & 0 & 2.3333 & 16 & 0.3333 & 1 & 1 & 0 \\ 0 & 0 & 0 & 12 & 0.6 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

Example 4.2 Consider the following matrix:

$$A = \begin{pmatrix} 2.6667 & 2.6667 & 3.3333 & 1.3333 & 3.6667 & 1.3333 \\ 3.0000 & 3.6667 & 1.6667 & 1.3333 & 2.6667 & 2.3333 \\ 2.6667 & 2.3333 & 2.0000 & 1.3333 & 2.3333 & 2.0000 \\ 3.3333 & 2.6667 & 1.6667 & 4.0000 & 2.6667 & 1.6667 \\ 3.3333 & 3.3333 & 1.3333 & 3.3333 & 2.3333 & 3.3333 \\ 3.3333 & 3.0000 & 3.0000 & 1.3333 & 2.6667 & 1.3333 \end{pmatrix}.$$

Then, the ZW factorization of A gives:

$$Z = \begin{pmatrix} 1.0000 & 0.3333 & 1.9231 & -0.3077 & -1.3333 & 0 \\ 0 & 1.0000 & 0.7692 & 0.0769 & 0 & 0 \\ 0 & 0 & 1.0000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.0000 & 0 & 0 \\ 0 & 0 & -0.0385 & 0.8462 & 1.0000 & 0 \\ 0 & -1.4444 & 1.6923 & -0.2308 & 1.7778 & 1.0000 \end{pmatrix},$$

and

$$W = \begin{pmatrix} -0.8462 & 0 & 0 & 0 & 0 & 0.4444 \\ 0.6923 & 1.6667 & 0 & 0 & 0.6667 & 0.6667 \\ 2.6667 & 2.3333 & 2 & 1.3333 & 2.3333 & 2 \\ 3.3333 & 2.6667 & 1.6667 & 4.0000 & 2.6667 & 1.6667 \\ 0.6154 & 1.1667 & 0 & 0 & 0.1667 & 2.0000 \\ -0.5043 & 0 & 0.0000 & 0.0000 & 0 & -4.2593 \end{pmatrix}.$$

5 Conclusion

We presented a generalized elimination method for solving real and Diophantine linear systems. We proposed a necessary and sufficient condition under which the method is carried out. We showed that different matrix factorizations such as block LU, the WZ and the ZW factorizations can be derived from the method. We also introduced the octant interlocking factorization which factorizes a nonsingular matrix into rhombus form factors. We presented two octant interlocking factorizations named as the ZO and the WS factorizations.

References

1. B. Bylina and J. Bylina, The Parallel Tiled WZ Factorization Algorithm for Multicore Architectures, International Journal of Applied Mathematics and Computer Science, 29(2) (2019) 407–419.
2. B. Bylina and J. Bylina, The WZ factorization in MATLAB In 2014 Federated Conference on Computer Science and Information Systems (IEEE), (2014) 561-568.
3. J. R. Bunch, Block methods for solving sparse linear systems, Sparse Matrix Computations, Academic Press, 1976, 39–58.
4. B. Codenotti, and F. Romani, A note on quadrant interlocking factorization. IMA journal of numerical analysis, 9(2) (1989) 139–143.
5. J. W. Demmel, N. J. Higham and R. S. Schreiber, Stability of Block LU Factorization, Numerical Linear Algebra with Applications, 2(2) (1995) 173–190.
6. J. W. Demmel, N. J. Higham and R. S. Schreiber, Stability of block algorithms with fast level-3 BLAS, ACM Transaction on Mathematical Software (TOMS), 18(3) (1992) 274-291., Numerical Linear Algebra with Applications, 2(2) (1995) 173–190.
7. C. Echeverria,J. Liesen and R. Nabben, Block diagonal dominance of matrices revisited: bounds for the norms of inverses and eigenvalue inclusion sets. Linear Algebra and its Applications, 553 (2018) 365–383.
8. D.J. Evans and M. Hatzopoulos, A parallel linear system solver, International Journal of Computer Mathematics, 7(3) (1979) 227-238.
9. D.J. Evans, Implicit matrix elimination (IME) schemes, International Journal of Computer Mathematics, 48(3-4) (1993) 229-237.
10. D.J. Evans and A. Hadjidimos, A modification of the quadrant interlocking factorization parallel method, International Journal of Computer Mathematics, 8(2) (1980) 149-166.
11. D.J. Evans, A. Hadjidimos and D. Noutsos, Parallel solution of linear systems by quadrant interlocking factorization methods, Computer Methods in Applied Mechanics and Engineering, 29(1) (1981a) 97-107.
12. D.J. Evans, A. Hadjidimos and D. Noutsos, The parallel solution of banded linear equation by the new quadrant interlocking factorization (QIF) method, International Journal of Computer Mathematics, 9(2) (1981b) 151-161.
13. D.J. Evans and A. Hadjidimos, Parallel solution to certain banded, symmetric and centro-symmetric systems by using the quadrant interlocking factorization method, Mathematics and Computers in Simulation, 23(2) (1981) 180-187.
14. A. Galantai, The rank reduction procedure of Egerváry. Central European Journal of Operations Research, 18(1) (2010) 5.
15. K. A. Gallivan, R. J. Plemmons and A. H. Sameh, Parallel algorithms for dense linear algebra computations, SIAM Review, 32(1) (1990) 54-135.
16. E. Golpar-Raboky and N. Mahdavi-Amiri, WZ factorization via Abaffy-Broyden-Spedicato algorithms, Bulletin of the Iranian Mathematical Society, 40(2) (2014) 399-411.
17. E. Golpar-Raboky and N. Mahdavi-Amiri, A new interpretation of the WZ factorization using block scaled ABS algorithms, Statistics, Optimization and Information Computing, 2(3) (2014) 243-256.
18. G. H. Golub and C. F. Van Loan, Matrix computation (4th edition), The johns hopkins university press, 2013.
19. A. Hadjidimos and D. Noutsos, Special similarity transformations in connection with the quadrant interlocking factorization techniques, Bulletin of the Greek Mathematical Society, 22(22) (1981) 44-66.
20. N. J. Higham, Accuracy and stability of numerical algorithms (2th edition), Society for industrial and applied mathematics, 2002.
21. N. Mahdavi-Amiri, E. Golpar-Raboky, Real and integer Wedderburn rank reduction formulas for matrix decompositions, Optimization Methods and Software, 30(4) (2015) 864-879.
22. R. M. M. Mattheij. Stability of block LU-decompositions of matrices arising from BVP. SIAM Journal on Algebraic Discrete Mathematics, 5(3) (1984) 314-331.
23. S.C.S. Rao, Existence and uniqueness of WZ factorization, Parallel Computing, 23(8) (1997) 1129-1139.
24. S.C.S. Rao and R. Kamra, A hybrid parallel algorithm for large sparse linear systems, Numerical Linear Algebra with Applications, 25(6) (2018) e2210.

25. J. Shanehchi and D. J. Evans, New variants of the quadrant interlocking factorization (QIF) method, International Conference on Parallel Processing, (1981), 493-507.
26. R. Schreiber, Block algorithms for parallel machines, Numerical Algorithms for Modern Parallel Computer Architectures, Springer, New York, 1988.