Iranian Journal of Numerical Analysis and Optimization
Vol. 12, No. 2, pp 301–314
DOI: 10.22067/ijnao.2021.70974.1043
https://ijnao.um.ac.ir/

How to cite this article
Research Article

A generalization of the ABS algorithms and its application to some special real and integer matrix factorizations

E. Golpar-Raboky* and N. Mahdavi-Amiri

Abstract

In 1984, Abaffy, Broyden, and Spediacto (ABS) introduced a class of the so-called ABS algorithms to solve systems of real linear equations. Later, the scaled ABS algorithm, the extended ABS algorithm, the block ABS algorithm, and the integer ABS algorithm were introduced leading to various well-known matrix factorizations. Here, we present a generalization of ABS algorithms containing all matrix factorizations such as triangular, WZ, and ZW. We present the octant interlocking factorization and show that the generalized ABS algorithm is more general to produce the octant interlocking factorization.

AMS subject classifications (2020): 15A03; 15A21; 15A23; 34C20; 49M27.

Keywords: ABS algorithms; Quadrant interlocking factorization; Octant interlocking factorization.

1 Introduction

The basic ABS class of algorithms was first introduced by Abaffy, Broyden, and Spedicato [1] for solving linear systems of equations. 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. Consider the system of linear equations

---

* Corresponding author

Received 14 June 2021; revised 9 November 2021; accepted 24 December 2021

Effat Golpar-Raboky
Department of Mathematics, University of Qom. Qom, Iran. Tel: +98-25-32103791
e-mail:g.raboky@qom.ac.ir

Nezam Mahdavi-Amiri
Faculty of Mathematical Sciences, Sharif University of Technology, Tehran, Iran. e-mail: nezamm@sharif.edu

$$Ax = b, \qquad x \in \mathbb{R}^n, \;\; A \in \mathbb{R}^{m\times n}, \;\; b \in \mathbb{R}^m, \; m \leq n, \tag{1}$$

where $rank(A)$ is arbitrary. With $A = [a_1, \ldots, a_m]^T$, the system is equivalently written as

$$a_i^T x = b_i, \quad i = 1, \ldots, m. \tag{2}$$

An ABS algorithm generates a sequence of approximations x_i such that x_{i+1} is a particular solution of the first i equations and leads to the general solution of a linear system by computing a particular solution and a matrix with rows producing the null space of the coefficient matrix.

An ABS method starts with an arbitrary initial vector $x_1 \in \mathbb{R}^n$ and a nonsingular matrix $H_1 \in \mathbb{R}^{n\times n}$. Given x_i , a solution of the first $i-1$ equations, and H_i a matrix with rows generating the null space of the first $i-1$ rows of the coefficient matrix, an ABS algorithm computes x_{i+1} as a solution of the first i equations and H_{i+1}, with rows generating the null space of the first i rows of the coefficient matrix. Below, we give the class of basic ABS algorithms for solving systems of linear equations (13).

Algorithm 1 (Basic ABS algorithm)**.**
(1) Give $x_1 \in \mathbb{R}^{n\times n}$, arbitrary, $H_1 \in \mathbb{R}^{n\times n}$, arbitrary and nonsingular. Set $i = 1$ and $r = 0$.
(2) Compute $\tau_i = a_i^T x_i - b_i$ and $s_i = H_i a_i$.
(3) **If** ($s_i = 0$ and $\tau_i = 0$), **then** let $x_{i+1} = x_i, H_{i+1} = H_i$ and **go to** (6) (the ith row of A is dependent on its first $i-1$ rows). **If** $s_i = 0$ and $\tau_i \neq 0$, **then stop** (the ith equation and hence the system is incompatible).
(4) Compute the search vector p_i by

$$p_i = H_i^T f_i, \tag{3}$$

where $f_i \in \mathbb{R}^n$ is an arbitrary vector satisfying $a_i^T H_i^T f_i \neq 0$. Compute

$$\alpha_i = \frac{\tau_i}{a_i^T p_i}$$

and

$$x_{i+1} = x_i - \alpha_i p_i.$$

(5) (Update the null space generator) Update H_i by

$$H_{i+1} = H_i - \frac{H_i a_i q_i^T H_i}{q_i^T H_i a_i}, \tag{4}$$

where $q_i \in R^n$ is an arbitrary vector satisfying $s_i^T q_i \neq 0$, and let $r = r + 1$.
(6) **If** $i = m$, **then Stop** (H_{m+1}^T generates the null space of A and r is its rank) **else** let $i = i + 1$ and **go to** (2).

Here, we recall some properties of ABS algorithms; for more details, see [3]. For simplicity, we assume that $A \in \mathbb{R}^{m\times n}$ has full row rank.

1. The system may be incompatible, which can be detected in step (3), when $s_i = 0$ and $\tau_i = b_i - a_i^T x_i \neq 0$.
2. $H_i a_i \neq 0$ if and only if a_i is linearly independent of $a_1, \dots, a_{i-1}$.
3. If $a_1, \dots, a_i$ are linearly independent, then the search vectors $p_1, \dots, p_i$ are linearly independent.
4. If $a_1, \dots, a_i$ are linearly independent, then with $P_i = (p_1, \dots, p_i)$, the implicit factorization $AP_i = L_i$ holds, where L_i is nonsingular lower triangular. Different choices of the parameters H_1, f_i, and q_i lead to different matrix factorizations.

Theorem 1 (LX factorization)**.** Let $A \in \mathbb{R}^{n\times n}$ be a nonsingular matrix and let $H_1 = I$. Then, there exits an index set $k_1 < k_2 < \cdots < k_n$ such that $e_{k_i}^T H_i a_i \neq 0$, for $k = 1, \dots, n$, and the parameter choices $f_i = q_i = e_{k_i}$ are well-defined. Let $[n] = \{1, \dots, n\}$, $B_i = \{k_1, \dots, k_i\}$, and $N_i = [n] \backslash B_i$. Then, we have the following properties:
(1) Every kth row of H_{i+1} with $k \in B_i$ is a null row.
(2) The vector p_i has $n - i$ zero components; its k_ith component is equal to one.
(3) For each $k \in N_i$, the kth column of H_{i+1} is the unit vector e_k, while for each $k \in B_i$, the kth column of H_{i+1} has zero components in the jth position, with $j \in B_i$, implying that only $i(n-i)$ elements need to be computed for H_{i+1}.

Proof. See [17]. □

Corollary 1 (LU factorization)**.** Let $A \in \mathbb{R}^{n\times n}$ be a strongly nonsingular matrix (that is, the determinants of all the principal submatrices are nonzero) and let $H_1 \in \mathbb{R}^{n\times n}$ be the identity matrix. Then
(1) the sequence $\{H_i\}$, where $q_i = \frac{e_i}{e_i^T H_i a_i}$, is well-defined.
(2) the first i rows of H_{i+1} are identically zero and the last $n - i$ columns of H_{i+1} are equal to the last $n - i$ columns of H_i.
(3) $P = (p_1, \dots, p_n)$ is an upper triangular matrix.

Proof. See [3, Theorems 6.3 and 6.5]. □

The scaled ABS method produces a matrix factorization $V^T AP = L$, where L is a lower triangular matrix. Choices of the parameters H_1, f_i, and q_i determine particular methods within the class so that various matrix factorizations are derived; see [1, 3, 4, 18, 16, 17, 19].

Obviously, the original system (13) is equivalent to the following scaled system:

$$V^T Ax = V^T b, \tag{5}$$

where V, the scale matrix, is an arbitrary nonsingular m by m matrix. By replacing a_i with $A^T v_i$ in Algorithm 1, a scaled ABS algorithm is obtained. Chen and Zhou [5] proposed a generalization of the ABS algorithms, named as extended ABS (EABS) class of algorithms, which differs from the ABS

class of algorithms only in updating the Abaffian matrices H_i. The block ABS algorithm was developed by Abaffy and Galantai [2]. Let $n_1, \dots, n_s$ be positive integer numbers such that $n_1 + \cdots + n_s = n$. Consider a block form of A as $A = (A_1^T, \dots, A_s^T)^T$, where $A_i \in \mathbb{R}^{n_i \times n}$, for $i = 1, \dots, s$. The Block ABS methods may be formulated as follows:
(1) Compute $S_i = H_i A_i^T$.
(2) Determine $F_i \in \mathbb{R}^{n \times n_i}$ such that $F_i^T S_i$ is nonsingular and set $P_i = H_i^T F_i$.
(3) Update the Abaffian matrix H_i by

$$H_{i+1} = H_i - H_i A_i^T (Q_i^T H_i A_i^T)^{-1} Q_i^T H_i, \tag{6}$$

where $Q_i \in \mathbb{R}^{n \times n_i}$ is an arbitrary matrix such that $S_i^T Q_i$ is nonsingular. Esmaeili, Mahdavi-Amiri, and Spedicato [7] presented the integer ABS class of algorithms for solving linear Diophantine equations, developed conditions for the existence of an integer solution, and determined all integer solutions [6]. An extension of the integer ABS algorithm using the scaled ABS algorithms was developed by Spedicato et al. [16]. A new class of extended integer ABS algorithms for solving linear Diophantine systems by computing an integer basis for the null space while controlling the growth of intermediate results was developed by Khorramizadeh and Mahdavi-Amiri [13]. Golpar-Raboky and Mahdavi-Amiri [10, 11, 12, 14] presented new ideas for updating the H_i leading to the development of a new class of extended integer ABS algorithms. They also showed how to compute the Smith normal form of an integer matrix using the scaled integer ABS algorithm [10].

For the (not necessarily independent) rows of H_{i+1} to be a generator of null space of the first i rows of A, the extended integer ABS algorithms can always be tuned to produce an integer basis for the integer null space of the coefficient matrix; see Esmaeili, Mahdavi-Amiri, and Spedicato [6].

2 Generalized ABS class of algorithms

The central problem of linear algebra is the solution of linear system of equations. Direct methods used to solve linear systems of equations are based on factorizations of the coefficient matrix into factors to be easy for use in solving the equations. The Gaussian elimination with the corresponding matrix decomposition, the *LU* decomposition, is the most useful method for solving linear equations. The method is well-defined if and only if A is strongly nonsingular, that is, all principal submatrices, $A(1:k, 1:k)$, for all k, are nonsingular. The parallel implicit elimination (PIE) method and the WZ factorization for solving large systems, suitable for parallel computers, have been introduced by Evans [15].

Definition 1. Let $[n] = \{1, \dots, n\}$ and let $\alpha_i \subset [n]$, 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 the cardinality of α by $|\alpha|$.

Let $|\alpha_i| = n_i$, for $i = 1, \dots, s$. Then, an index set $\alpha = \{\alpha_1, \dots, \alpha_s\}$ is a block permutation vector, when the ith block size is equal to n_i and $n = \sum_{i=1}^{s} n_i$.

Let $A \in \mathbb{R}^{m\times n}$ and let α and β be two index sets of $[n]$. Let $A(:, b)$ denote the $m \times |b|$ submatrix of A containing the columns specified by b and let $A(\alpha_i, \beta_j)$, the (i, j)th block of A, 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. If $\alpha_i = \beta_j$, then $A(\alpha_i, \beta_j)$ is a principal submatrix of A and if $\alpha_i = \beta_j = \{1, \dots, k\}$, $1 \le k \le \min\{m, n\}$, then $A(\alpha_i, \beta_j)$ is a leading principal submatrix of A.

Definition 2. Let $A \in R^{n\times n}$, let $t = \{\alpha_1, \dots, \alpha_s\}$ and $\beta = \{\beta_1, \dots, \beta_s\}$ be two index sets of n, and let $A(\alpha_i, \beta_j)$, $1 \le i, j \le s$, denote the (i, j)th block of A. Then,

$$A(\alpha_1, \beta_1) \subset \cdots \subset A(\cup_{i=1}^{s}\alpha_i, \cup_{j=1}^{s}\beta_j) = A \tag{7}$$

is a nested submatrix sequence of A. We say A is (α, β)-block strongly nonsingular if and only if $A(\cup_{i=1}^{k}\alpha_i, \cup_{i=1}^{k}\beta_i)$, for $k = 1, \dots, s$, are nonsingular.

Let P_α and P_β denote the permutation matrices moving the rows and columns of A based on the index sets α and β, respectively, and let

$$A_{\alpha,\beta} = P_\alpha^T A P_\beta. \tag{8}$$

Corollary 2. Let $A \in \mathbb{R}^{n\times n}$. Then A is (α, β)-block strongly nonsingular if and only if $A_{\alpha,\beta}$ is strongly nonsingular.

Note that Theorem 1 provides a relationship between elimination methods and the ABS method. The search vectors p_i in step (4) of Algorithm 1 are aligned sequentially. The generalized elimination method and the ABS method do not produce the same matrix factorizations, generally. In the next section, we present a generalization of the ABS algorithms containing all matrix factorizations produced by the generalized elimination method such as triangular, WZ, and ZW. The generalized ABS method is more general to produce some new matrix factorizations such as the SO and OS factorizations which cannot be produced by the generalized elimination method.

Let $A \in \mathbb{R}^{n\times n}$, let $\alpha = \{\alpha_1, \dots, \alpha_s\}$ and $\beta = \{\beta_1, \dots, \beta_s\}$ be two index sets, and let $A_i \in \mathbb{R}^{|\alpha_i|\times n}$ be such that $A_i = A(\alpha_i, :)$.

Algorithm 2 (Generalized ABS algorithm)**.**

```
Input: $A \in \mathbb{R}^{m\times n}$, an arbitrary nonzero vector $x \in \mathbb{R}^{m\times n}$, an arbitrary nonsingular matrix $H_1 \in \mathbb{R}^{n\times n}$, and two index sets $\alpha = \{\alpha_1, \dots, \alpha_s\}$ and $\beta = \{\beta_1, \dots, \beta_s\}$.
For $i = 1, \dots, s$ do
(1) Compute $S_i = H_i A(\alpha_i, :)$.
(2) Determine $F_i \in \mathbb{R}^{n\times \beta_i}$ such that $F_i^T S_i$ is nonsingular and set $P(:, \alpha_i) = H_i^T F_i$.
(3) Update the approximation for the solution by
```

$$x(\alpha_{i+1}) = x(\alpha_i) - P(:, \alpha_i)d_i,$$

where d_i is the unique solution of the following nonsingular system:

$$A(\alpha_i, :)P(:, \alpha_i)d_i = r_i,$$

and $r_i = A(\alpha_i, :)x_i - \beta_i$.
(3) Update the Abaffian matrix H_i by

$$H_{i+1} = H_i - H_i A(\alpha_i, :)\left(Q_i^T H_i A(\alpha_i, :)\right)^{-1} Q_i^T H_i, \tag{9}$$

where $Q_i \in \mathbb{R}^{n\times\alpha_i}$ is an arbitrary matrix so that $S_i^T Q_i$ is nonsingular.
end for.
(4) Let $P = (p_1, \dots, p_n)$ and compute $C = AP$.

Algorithm 2 provides a null space characterization for the matrix A. From Algorithm 2, we have $H_{i+1}a_j = 0$, for $j \in \cup_{k=1}^{i}\alpha_k$, where a_j^T is the jth row of A. According to (8), Theorem 1 and Corollary 1, we have the following results.

Theorem 2. Let $A \in \mathbb{R}^{n\times n}$, and let $\alpha = (\alpha_1, \dots, \alpha_s)$ and $\beta = (\beta_1, \dots, \beta_s)$ be two index sets. Then, A is (α, β)-block strongly nonsingular if and only if $I(:, b(i))^T H_i A(t(i), :)^T$, for $i = 1, \dots, s$, are nonsingular.

Theorem 3. Let $Q_i = I(:, \beta_i)$, and let H_{i+1} be defined by (9). Then, the following properties hold:
(a) The jth row of H_{i+1} is zero, for $j \in \cup_{k=1}^{i} b(k)$.
(b) The jth column of H_{i+1} is equal to the jth column of H_1, for $j \notin \cup_{k=1}^{i} b(k)$.

Proof. See [3, Theorem 6.3]. □

Now, consider the following definition.

Definition 3. $A \in \mathbb{Z}^{n\times n}$ is a unimodular matrix if and only if $|det(A)| = 1$.

Note that, A is unimodular if and only if A^{-1} is unimodular.

Remark 1. Let $A \in \mathbb{Z}^{n\times n}$. If $A(\alpha_k, \beta_k)$, for $k = 1, \dots, s$, are unimodular, then Algorithm 2 produces an integer matrix factorizatin (B and C are integer matrices).

Algorithm 2 produces a matrix factorization $AP = C$. Different choices of the parameters Q, F, t and b lead to different matrix factorizations. For $\alpha_i = \beta_i = i$, $Q_i = F_i = e_i$, $1 \le i \le n$, Algorithm 2 produces an LU factorization. Next, we show how to choose the parameters in Algorithm 2 to compute the WZ and ZW factorizations. We also discuss the octant interlocking factorization method and present two new factorizations named as the OS and SO factorizations.

3 Quadrant interlocking factorization

A direct method, called the WZ factorization, for solving linear systems of equations $Ax = b$ was introduced by Evans and Hotzopoulos [9]. Let A be an $n \times n$ nonsingular matrix. The WZ factorization of [8] expresses A as $A = WZ$, where W and Z have the following forms:

$$W = \begin{pmatrix} \bullet & \circ & \circ & \circ & \bullet \\ \bullet & \bullet & \circ & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \circ & \bullet & \bullet \\ \bullet & \circ & \circ & \circ & \bullet \end{pmatrix}, \quad Z = \begin{pmatrix} \bullet & \bullet & \bullet & \bullet & \bullet \\ \circ & \bullet & \bullet & \bullet & \circ \\ \circ & \circ & \bullet & \circ & \circ \\ \circ & \bullet & \bullet & \bullet & \circ \\ \bullet & \bullet & \bullet & \bullet & \bullet \end{pmatrix}, \quad X = \begin{pmatrix} \bullet & \circ & \circ & \circ & \bullet \\ \circ & \bullet & \circ & \bullet & \circ \\ \circ & \circ & \bullet & \circ & \circ \\ \circ & \bullet & \circ & \bullet & \circ \\ \bullet & \circ & \circ & \circ & \bullet \end{pmatrix}, \tag{10}$$

where the empty bullets stand for zero and the other bullets stand for possible nonzeros.

The matrix W is called a unit W-matrix if in addition, $w_{ii} = 1$, for $i = 1, \ldots, n$, and $w_{i,n-i+1} = 0$, for $i \neq (n+1)/2$, when n is odd. The transpose of a (unit) *W-matrix* is called a (unit) *Z-matrix* and vise versa. Moreover, A matrix which is both a Z- and a *W-matrix* is called an *X-matrix*.

Note that, we assume that A is nonsingular and of an even size n (without loss of generality) and that $s = \frac{n}{2}$.

Theorem 4. Let A is an $n \times n$ matrix and let n be even. Then A has a WZ factorization if and only if the nested submatrices $A(1 : k, n-k+1 : n, 1 : k, n-k+1 : n)$ are invertible, for $k = 1, \ldots, n/2$.

Proof. See proof of of [15, Theorem 2]. □

Now, we show how to choose the parameters of the generalized ABS algorithm to compute the WZ factorization.
Consider two index sets $\alpha = \{\alpha_1, \ldots, \alpha_s\}$ and $\beta = \{\beta_1, \ldots, \beta_s\}$ such that $\alpha_k = \beta_k = \{k, n-k+1\}$ and $F_k = Q_k = [e_k, e_{n-k+1}]$, for $k = 1, \ldots, s$. Then, P is a Z-matrix, C is a W-matrix, and Algorithm 2 leads to a WZ factorization of A.

Remark 2. Let $A \in \mathbb{Z}^{n\times n}$. If the nested submatrices $A(1 : k, n-k+1 : n, 1 : k, n-k+1 : n)$ are unimodular, for $k = 1, \ldots, \frac{n}{2}$, then A has an integer WZ factorization.

Consider two equal index sets $\alpha = \{\alpha_1, \ldots, \alpha_s\}$ and $\beta = \{\beta_1, \ldots, \beta_s\}$ such that $\alpha_k = \beta_k = \{s-k+1, s+k\}$ and $F_k = Q_k = [e_{s-k+1}, e_{s+k}]$, for $k = 1, \ldots, s$. Then, P is a W-matrix, C is a Z-matrix, and Algorithm 2 leads to a WZ factorization of A.

Theorem 5. Let $A \in \mathbb{R}^{n\times n}$. Then A has a ZW factorization if and only if the nested submatrices $A(s-k+1 : s+k, s-k+1 : s+k)$ are invertible, for $k = 1, \ldots, s$.

Remark 3. Let $A \in \mathbb{Z}^{n\times n}$. If the nested submatrices $A(s-k+1 : s+k, s-k+1 : s+k)$ are unimodular, for $k = 1, \ldots, s$, then A has an integer ZW factorization.

4 Octant interlocking factorization

Here, we first define octant matrices and then present the Octant Interlocking Factorization (OIF). We provide the necessary and sufficient conditions for the existence of OIF and show how to choose the parameters of the generalized ABS algorithm to compute the factorization.

Definition 4. We say $A \in \mathbb{R}^{n\times n}$ is an octant matrix, if it has one of the following structures:

$$S = \begin{pmatrix} \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \circ & \bullet & \bullet \\ \bullet & \circ & \circ & \circ & \bullet \\ \bullet & \bullet & \circ & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet \end{pmatrix}, O = \begin{pmatrix} \circ & \circ & \bullet & \circ & \circ \\ \circ & \bullet & \bullet & \bullet & \circ \\ \bullet & \bullet & \bullet & \bullet & \bullet \\ \circ & \bullet & \bullet & \bullet & \circ \\ \circ & \circ & \bullet & \circ & \circ \end{pmatrix}, \tag{11}$$

with the empty bullets standing for zero and the other bullets standing for possible nonzeros. The matrices in (11) are called the S-matrix and the O-matrix, respectively.

Definition 5. Let $A \in \mathbb{R}^{n\times n}$. We say that A has an octant interlocking factorization, if $A = BC$ such that B and C are octant matrices.

The transpose and the inverse of an S-matrix are an S-matrix and an O-matrix, respectively, as well as the transpose and the inverse of an O-matrix are an O-matrix and an S-matrix, respectively.

Note that, we assume that $A \in \mathbb{R}^{n\times n}$ is nonsingular, that n is an even number, and that $s = \frac{n}{2}$.
Consider two index sets $\alpha = \{\alpha_1, \ldots, \alpha_s\}$ and $\beta = \{\beta_1, \ldots, \beta_s\}$ such that $\alpha_k = \{k, n-k+1\}, \beta_k = \{s-k+1, s+k\}$, and $Q_k = F_k = [e_k, e_{n-k+1}]$. Then, C is an O-matrix and P is an S-matrix.

Now, let $\beta_k = \{k, n-k+1\}$, let $\alpha_k = \{s-k+1, s+k\}$, and let $Q_k = F_k = [e_{s-k+1}, e_{s+k}]$. Then, P is an O-matrix and C is an S-matrix.

Theorem 6. Let $A \in \mathbb{R}^{n\times n}$. Then A has an OS factorization if and only if the nested submatrices $A(s-k+1 : s+k, 1 : k, n-k+1 : n)$ are invertible, for $k = 1, \ldots, s$.

Remark 4. Let $A \in \mathbb{Z}^{n\times n}$. If the nested submatrices $A(s-k+1 : s+k, 1 : k, n-k+1 : n)$, for $k = 1, \ldots, s$, are unimodular, then A has an integer OS factorization.

5 Generalized ABS algorithm and matrix factorizations

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.

Let $A \in \mathbb{R}^{n\times n}$ and let $\alpha = \{\alpha_1, \ldots, \alpha_s\}$ and $\beta = \{\beta_1, \ldots, \beta_s\}$ be two index sets of n. Algorithm 2 computes a matrix factorization $AP = C$, with parameter choices $Q_i = F_i = I(:, \alpha_i)$ and $P(:, \beta_i) = H_i^T F_i$ as follows:

$$A(\alpha_i, :)H_k^T = 0, \quad i = 1, \ldots, k-1,$$

and

$$C(\alpha_i, \beta_k) = A(\alpha_i, :)P(:, \beta_k) = A(\alpha_i, :)H_k^T I(:, \alpha_i) = 0, \qquad i = 1, \ldots, k-1.$$

This means that in the kth step the generalized ABS algorithm performs a partial zeroing such that the elements of the submatrix C corresponding to the columns specified by β_k and the rows specified by α_i, for $i = 1, \ldots, k-1$, turn to zero. Here, we set the parameters in Algorithm 2 to present the associated matrix factorizations $C = AP$.

We consider different choices for α and β such that all the blocks $A(\alpha_i, \beta_j)$ turn to be 1×1 and present the associated matrix factorizations. First, we define two index sets. Let $J = \{j_1, \ldots, j_n\}$ with the J_i as follows:

$$j_i = \begin{cases} \frac{i+1}{2} & \text{if } i \text{ is odd,} \\ n - \frac{i}{2} + 1 & \text{if } i \text{ is even.} \end{cases} \tag{12}$$

We define the index set $K = \{k_1, \ldots, k_n\}$ as follows. If n is an even number, then define

$$k_i = \begin{cases} \frac{n}{2} - \frac{i+1}{2} + 1 & \text{if } i \text{ is odd,} \\ \frac{n}{2} + \frac{i}{2} & \text{if } i \text{ is even,} \end{cases} \tag{13}$$

if n is an odd number, then define

$$k_i = \begin{cases} \frac{n+1}{2} - \frac{i}{2} & \text{if } i \text{ is even,} \\ \frac{n+1}{2} + \frac{i-1}{2} & \text{if } i \text{ is odd.} \end{cases} \tag{14}$$

If $\alpha_i = \beta_i = i$, then C is a lower triangular and P is an upper triangular matrix. For $\alpha_i = \beta_i = n - i + 1$, C is an upper triangular and P is a lower triangular matrix. The different cases are noted in Figure 1

In Figure 2, we give a MATLAB code for Algorithm 2, where $A \in \mathbb{R}^{n\times n}$, $t = \{t_1, \ldots, t_n\}$ and $b = \{b_1, \ldots, b_n\}$ are two index sets, and all the blocks $A(\alpha_i, \beta_i)$ are 1×1, for $i = 1, \ldots, n$.

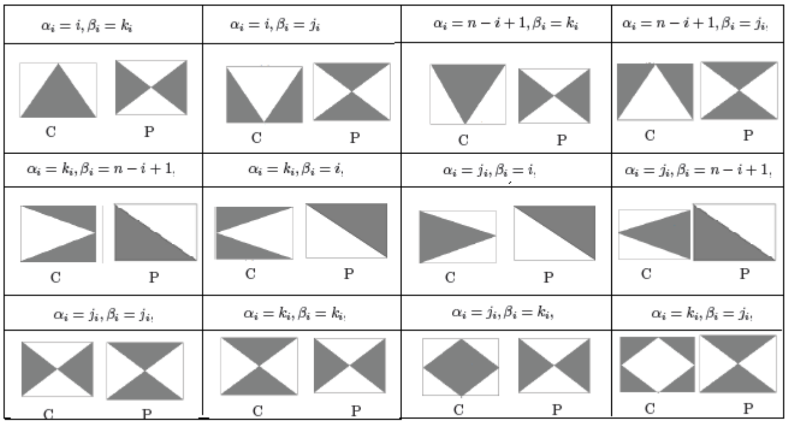

Figure 1: Matrix factorizations associated with different index sets

```
Function    [C,P]=GENABS(A,t,b)
clc
[n,n]=size(A);
E=eye(n);
P=zeros(n);
H=E;
for  i=1:n
      e=E(:,[t(i)]);
      q=E(:,[b(i)]);
      P(:,[b(i)])=H'*q;
      U=e'*A*H'*q;
      H=H-(H*A'*e*inv(U')*q'*H);
end;
C=A*P;
end
```

Figure 2: MATLAB code for Algorithm 2

6 Numerical illustrations

We give illustrations of our proposed algorithms to compute the QIF and OIF factorizations. The algorithms were implemented using $Matlab\ R2020a$.

Example 1. Consider the following matrix:

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 3 & 6 & 10 & 15 & 21 \\ 1 & 4 & 10 & 20 & 35 & 56 \\ 1 & 5 & 15 & 35 & 70 & 126 \\ 1 & 6 & 21 & 56 & 126 & 252 \end{pmatrix}.$$

Let $\alpha_k = \beta_k = \{s-k+1, s+k\}$, for $k = 1,2,3$ and $s = \frac{n}{2} = 3$. Then, the generalized elimination algorithm, Algorithm 1, produces P as a W-matrix and C as a Z-matrix, and we have a ZW factorization as follows:

$$P = \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 & 0 & 3 \\ 2 & -1 & 1 & 0 & 2.5 & -8 \\ -1 & 0.3 & 0 & 1 & -3 & 9 \\ 0.2 & 0 & 0 & 0 & 1 & -4.8 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

and

$$C = \begin{pmatrix} 0.2 & 0.3 & 1 & 1 & 0.5 & 0.2 \\ 0 & 0.2 & 3 & 4 & 0.5 & 0 \\ 0 & 0 & 6 & 10 & 0 & 0 \\ 0 & 0 & 10 & 20 & 0 & 0 \\ 0 & 0.5 & 15 & 35 & 2.5 & 0 \\ 0.2 & 1.8 & 21 & 56 & 10.5 & 1.2 \end{pmatrix}.$$

Example 2. Let

$$A = \begin{pmatrix} 13.8966 & 15.3103 & 14.1873 & 2.3800 & 15.0253 & 10.9443 \\ 6.3420 & 15.9040 & 15.0937 & 9.9673 & 5.1019 & 2.7725 \\ 19.0044 & 3.7375 & 5.5205 & 19.1949 & 10.1191 & 2.9859 \\ 0.6889 & 9.7953 & 13.5941 & 6.8077 & 13.9815 & 5.1502 \\ 8.7749 & 8.9117 & 13.1020 & 11.7054 & 17.8181 & 16.8143 \\ 7.6312 & 12.9263 & 3.2522 & 4.4762 & 19.1858 & 5.0856 \end{pmatrix}.$$

Let $\alpha_k = \{s-k+1, s+k\}$ and $\beta_k = \{k, n-k+1\}$, for $k = 1,2,3$ and $s = 3$. By the generalized ABS algorithm, Algorithm 2, we have

$$P = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1.0000 & -3.3515 & -11.9955 & 0 & 0 \\ 1 & -0.7279 & 4.0998 & 13.5268 & -0.8931 & 0 \\ 0 & 0.0146 & -0.8436 & 1.3279 & -0.2703 & 1 \\ 0 & 0 & -1.2767 & -5.7630 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

and

$$C = \begin{pmatrix} 14.1873 & 5.0185 & -0.4413 & -64.2317 & 1.7108 & 2.3800 \\ 15.0937 & 5.0633 & 0 & 0 & -11.0731 & 9.9673 \\ 5.5205 & 0 & 0 & 0 & 0 & 19.1949 \\ 13.5941 & 0 & 0 & 0 & 0 & 6.8077 \\ 13.1020 & -0.4537 & 0 & 0 & 2.9522 & 11.7054 \\ 3.2522 & 10.6245 & -50.6286 & -210.6023 & 15.0712 & 4.4762 \end{pmatrix}.$$

We realize that P is an O-matrix, that C is an S-matrix, and that $AP = C$.

7 Concluding remarks

We presented a generalized elimination approach for solving linear systems. We established the necessary and sufficient conditions under which the proposed method is applicable. We showed that different matrix factorizations could be derived from the method such as the LU, WZ, and ZW factorizations. We also proposed the octant interlocking factorization to factorize a nonsingular matrix into octant matrices. We presented a generalized ABS algorithm and showed how to choose the parameters of the algorithm to compute the WZ and the ZW factorizations as well as the octant interlocking factorization of real and integer matrices.

Acknowledgements

The first author thanks the Research Council of Qom University and the second author thanks the Research Council of Sharif University of Technology for supporting this work.

References

1. Abaffy, J., Broyden, C.G. and Spedicato, E. *A class of direct methods for linear systems*, Numer. Math. 45 (1984) 361–376.

2. Abaffy, J. and Galantai, A. *Conjugate direction methods for linear and nonlinear systems of algebraic equations*, Colloquia Mathematica Societatis Janos Bolyai 50 (1986) 481–502.

3. Abaffy, J. and Spedicato, E. *ABS projection algorithms, mathematical techniques for linear and nonlinear equations*, Halsted Press, Chichester, 1989.

4. Adib, M., Mahdavi-Amiri, N. and Spedicato, E. *Broyden method as an ABS algorithm,* Publ. Univ. Miskolc Ser. D Nat. Sci. Math. 40 (1999), 3–13.

5. Chen, Y. and Zhou, B. *On g-inverses and nonsingularity of a bordered matrix* $\begin{pmatrix} A & B \\ C & O \end{pmatrix}$, Linear Algebra Appl. 133 (1990) 133–151.

6. Esmaeili, H., Mahdavi-Amiri, N. and Spedicato, E. *Generating the integer null space and conditions for determination of an integer basis using the ABS algorithms*, Bull. Iran. Math. Soc. 27(1) (2001) 1–18.

7. Esmaeili, H., Mahdavi-Amiri, N. and Spedicato, E. *A class of ABS algorithms for linear Diophantine systems*, Numer. Math. 90 (2001) 101–115.

8. Evans, D.J., Hadjidimos, A. and Noutsos, D. *The parallel solution of banded linear equation by the new quadrant interlocking factorization (Q.I.F.) method,* Internat. J. Comput. Math. 9(2) (1981) 151–161.

9. Evans, D.J. and Hatzopoulos, M. *A parallel linear system solver*, Int. J. Comput. Math. 7(3) (1979) 227–238.

10. Golpar-Raboky, E.and Mahdavi-Amiri, N. *Diophantine quadratic equation and Smith normal form using scaled extended integer ABS algorithms,* J. Optim. Theory Appl. 152(1) (2012) 75–96.

11. Golpar-Raboky, E. and Mahdavi-Amiri, N. *WZ factorization via Abaffy-Broyden-Spedicato algorithms*, Bull. Iran. Math. Soc. 40(2) (2014) 1–13.

12. Golpar-Raboky, E. and Mahdavi-Amiri, N. *A new interpretation of the integer and real WZ factorization using block scaled ABS algorithms*, Stat. Optim. Inf. Comput. 2 (2014) 243–256.

13. Khorramizadeh, M. and Mahdavi-Amiri, N. *Integer extended ABS algorithms and possible control of intermediate results for linear Diophantine systems*, 4OR 7 (2009) 145–167.

14. Mahdavi-Amiri, N. and Golpar-Raboky, E. *Real and integer Wedderburn rank reduction formulas for matrix decompositions*, Optim. Methods Softw. 30(4) (2015) 864–879.

15. Rao, S.C.S. *Existence and uniqueness of WZ factorization*, Parallel Comput. 23 (1997) 1129–1139.

16. Spedicato, E., Bodon, E. Del Popolo, A. and Mahdavi-Amiri, N. *ABS methods and ABSPACK for linear systems and optimization: A review*, 4OR 1 (2003) 51–66.

17. Spedicato, E., Bodon, E., Del Popolo, A. and Xia, Z. *ABS algorithms for linear systems and optimization: A review and a bibliography,* Ricerca Operativa 29 (2000) 39–88.

18. Spedicato, E., Bodon, E., Zunquan, X. and Mahdavi-Amiri, N. *ABS methods for continuous and integer linear equations and optimization*, CEJOR Cent. Eur. J. Oper. Res. 18 (2010) 73–95.

19. Spedicato, E., Xia, Z. and Zhang, L. *The implicit LX method of the ABS class,* Optim. Methods Softw. 8 (1997) 99–110.

How to cite this article

E. Golpar-Raboky and N. Mahdavi-Amiri A generalization of the ABS algorithms and its application to some special real and integer matrix factorizations. *Iranian Journal of Numerical Analysis and Optimization*, 2022; 12(2): 301-314. doi: 10.22067/ijnao.2021.70974.1043.