

Research Article

Indefinite Ruhe's Variant of the Block Lanczos Method for Solving the Systems of Linear Equations

M. Ghasemi Kamalvand  and K. Niazi Asil 

Department of Mathematics, Lorestan University, Khorramabad, Iran

Correspondence should be addressed to M. Ghasemi Kamalvand; ghasemi.m@lu.ac.ir

Received 10 January 2020; Revised 26 March 2020; Accepted 9 April 2020; Published 26 May 2020

Academic Editor: Soheil Salahshour

Copyright © 2020 M. Ghasemi Kamalvand and K. Niazi Asil. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, we equip \mathbb{C}^n with an indefinite scalar product with a specific Hermitian matrix, and our aim is to develop some block Krylov methods to indefinite mode. In fact, by considering the block Arnoldi, block FOM, and block Lanczos methods, we design the indefinite structures of these block Krylov methods; along with some obtained results, we offer the application of this methods in solving linear systems, and as the testifiers, we design numerical examples.

1. Introduction

First of all, we are going to introduce the inner product space that our work is scheduled to be done in that. Recall that the indefinite inner product $[\cdot, \cdot]$ in \mathbb{C}^n has the all features of a standard inner product except that it may be nonpositive. In the other words, that is linear in the first argument, antisymmetry, and nondegenerate. The latter means that if $[x, y] = 0$ for every $y \in \mathbb{C}^n$, then $x = 0$. This kind of inner product may be applied in some areas of sciences and is commonly defined as $[x, y] = y^* J x$ where $J \in \mathbb{C}^{n \times n}$ is a nonsingular Hermitian matrix, and even in some specific scientific areas such as the theory of relativity or in the research of the polarized light, it may be exclusively as follows:

$$J = \text{diag}(j_1, \dots, j_n), j_k \in \{-1, +1\}. \quad (1)$$

With this particular J , the indefinite inner product $[\cdot, \cdot]$ is referred to as hyperbolic and take the form

$$[x, y] = y^* J x = \sum_{i=1}^n j_{ii} x_i \bar{y}_i. \quad (2)$$

In [1] and [2], by considering \mathbb{C}^n with the indefinite scalar product (2), a number of the Krylov subspace methods

have been reviewed and restructured. These methods are indefinite Arnoldi, indefinite full orthogonalization, and indefinite Lanczos. In this paper, we extend these indefinite Krylov methods to their indefinite block versions which will be discussed more in the subsequent sections.

Considering J as (1), we will need to consider the following definitions:

A subspace \mathbb{M} is said to be nondegenerate, with respect to the indefinite inner product $[\cdot, \cdot]$ if $x \in \mathbb{M}$ and $[x, y] = 0$ for all $y \in \mathbb{M}$ imply that $x = 0$. Otherwise \mathbb{M} is degenerate. For example, the indefinite inner product $[\cdot, \cdot]$ ensures that \mathbb{C}^n itself is always nondegenerate.

If \mathbb{M} is any nondegenerate nonzero subspace of \mathbb{C}^n , then the basis x_1, \dots, x_k for \mathbb{M} is said to be an orthogonal basis with respect to the indefinite inner product $[\cdot, \cdot]$, if $[x_i, x_j] = 0$ for $i \neq j$, and is said to be an orthonormal basis if in addition to orthogonality, $[x_i, x_i] = \pm 1$ for $i = 1, \dots, k$. If the indefinite inner product $[\cdot, \cdot]$ in this definition is the special indefinite inner product presented in (2), then the above definitions of orthogonal basis and orthonormal basis are called J -orthogonal and J -orthonormal bases, respectively.

A matrix A is said to be J -symmetric when $A = JA^T J$ and we write $A = A^{[T]}$.

This paper is classified such that the next section is devoted to recounting the basic definitions and the block Arnoldi and block Arnoldi-Ruhe's variant. In the third

- (1) Choose a unitary matrix V_1 of dimension $n \times p$
- (2) For $j = 1, \dots, m$ Do
- (3) Compute $H_{ij} = V_i^T A V_j, i = 1, \dots, j$
- (4) $W_j = A V_j - \sum_{i=1}^j V_i H_{ij}$
- (5) Compute the QR-factorization of $W_j : W_j = V_{j+1} H_{j+1,j}$
- (6) EndDo

ALGORITHM 1. Block Arnoldi.

section, the indefinite versions of its previous section algorithms are designed and the use of these methods in solving linear systems is discussed. The fourth section offers the indefinite version of the block Lanczos method and its application in solving linear systems with J -symmetric coefficient matrices, and finally, the numerical examples are given in the fifth chapter.

2. Block FOM Method

In some areas of computational sciences, there may be a need to solve large sparse linear systems with several right-hand sides that are given at once. A nonsingular linear systems with p right-hand sides can written as

$$AX = B, \quad (3)$$

with $A \in \mathbb{C}^{n \times n}$ and $X, B \in \mathbb{C}^{n \times p}$. Generally, we call such $n \times p$ matrices block vectors. Block Krylov methods are iterative methods that have been designed for such problems. Note that for $A \in \mathbb{C}^{n \times n}$ and $X \in \mathbb{C}^{n \times p}$, the block Krylov subspaces $\mathbb{K}_n(A, X)$ generated by A from X are

$$\mathbb{K}_n(A, X) := \text{block span}(X, AX, \dots, A^{n-1}X) \subset \mathbb{C}^{n \times p}, \quad (4)$$

where “block span” is defined such that

$$\mathbb{K}_n(A, X) = \left\{ \sum_{k=0}^{n-1} A^k X_{\gamma_k; \gamma_k} \in \mathbb{C}^{p \times p} \gamma, (k = 0, \dots, n-1) \right\}. \quad (5)$$

In general, a block Krylov subspace method acts in the same manner as a Krylov subspace method, but at each iteration, the operator is applied to a block of vectors instead of just one. Most Krylov methods can be generalized to block Krylov space solvers (for example, see [3–7]). Specifically, we recall the algorithms of the three block Krylov methods where our paper is focused on, i.e., the block Arnoldi, block FOM, and block Lanczos algorithms.

The outcome of Algorithm 1 is an orthogonal basis for the block Krylov subspace $\mathbb{K}_n(A, V_1) \equiv \{V_1, AV_1, \dots, A^{n-1}V_1\}$ and also a band upper Hessenberg matrix with p subdiagonals.

The second algorithm which is the resulting of A. Ruhe’s [8] efforts as shown in Algorithm 2.

- (1) Choose p initial vectors $\{v_i\}_{i=1, \dots, p}$
- (2) For $j = p, \dots, m + p - 1$ Do
- (3) Set $k := j - p + 1$
- (4) Compute $w := A v_k$
- (5) For $i = 1, \dots, j$ Do
- (6) $h_{i,k} := (w, v_i)$
- (7) $w := w - h_{i,k} v_i$
- (8) EndDo
- (9) Compute $h_{j+1,k} := \|w\|_2$ and $v_{j+1} = w/h_{j+1,k}$
- (10) EndDo

ALGORITHM 2. Block Arnoldi-Ruhe’s variant.

Especially, the case $p = 1$ coincides with the usual Arnoldi process. According to the algorithm, the vector w satisfies the relation

$$w = A v_k - \sum_{i=1}^j h_{i,k} v_i, \quad (6)$$

where $k = j - p + 1$. By line 9, $w = h_{j+1,k} v_{j+1}$ which yields

$$A v_k = \sum_{i=1}^{k+p} h_{i,k} v_i. \quad (7)$$

Thus,

$$A V_m = V_{m+p} \bar{H}_m, \quad (8)$$

in which the matrix \bar{H}_m is of size $(m + p) \times m$ and V_j represents the $n \times j$ matrix with columns v_1, \dots, v_j .

Another issue on which we will work in the indefinite mode is the solving of linear systems with multiple right-hand sides. The block generalization of FOM and Lanczos methods is defined in straightforward ways to solve linear systems which are defined in spaces with definite inner products. As a short reminder, consider the linear systems

$$A x^{(i)} = b^{(i)}, i = 1, \dots, p, \quad (9)$$

or in matrix notation

$$AX = B, \quad (10)$$

in which $X = (x^{(1)}, \dots, x^{(p)})$ and $X = (b^{(1)}, \dots, b^{(p)})$, and assume that the initial block guess $X_0 = (x_0^{(1)}, \dots, x_0^{(p)}) \in \mathbf{C}^{n \times p}$ of initial guesses $x_0^{(i)}$ is given and the initial block residual R_0 is as follows:

$$R_0 \equiv (r_0^{(1)}, \dots, r_0^{(p)}), \quad (11)$$

in which $r_0^{(i)} := b^{(i)} - Ax_0^{(i)}$. Recall that in the case of a single system, that is when $p = 1$, the approximate solution x_m is chosen such that the correction $x_m - x$ lies in the Krylov subspace

$$K_m(A, r_0) := \text{span}(r_0, Ar_0, \dots, A^{m-1}r_0) \subset \mathbf{C}^n. \quad (12)$$

The block Krylov space method for solving the p systems (3) is an iterative method that generates approximate solution X such that

$$X_m - X_0 \in \mathbb{K}_m(A, R_0), \quad (13)$$

where R_0 is the initial residual as (11).

The block FOM algorithm compute the QR-factorization of R_0 :

$$R_0 = [v_1, \dots, v_p]R, \quad (14)$$

in which matrix $[v_1, \dots, v_p]$ is unitary and R is $p \times p$ upper triangular. This factorization provides the first p vectors of the block Arnoldi basis.

Each of the approximate solutions has the form

$$x^{(i)} = x_0^{(i)} + V_m y^{(i)}. \quad (15)$$

Writing $X = (x^{(1)}, \dots, x^{(p)})$ and $Y = (y^{(1)}, \dots, y^{(p)})$, we have

$$X = X_0 + V_m Y. \quad (16)$$

Let E_1 be the $(m+p) \times p$ matrix whose upper $p \times p$ principal block is an identity matrix. Then, the relation (8) results in

$$\begin{aligned} B - AX &= B - A(X_0 + V_m Y) \\ &= R_0 - AV_m Y = [v_1, \dots, v_p]R - V_{m+p}\bar{H}_m Y \\ &= V_{m+p}(E_1 R - \bar{H}_m Y). \end{aligned} \quad (17)$$

Note that the vector $\bar{g}^{(i)} \equiv E_1 \text{Re}_i$ in \mathbf{C}^{m+p} is a vector whose arrays are zero apart from those from 1 to i which are derived from the i th column of the upper triangular matrix R . The matrix \bar{H}_m is an $(m+p) \times m$ matrix. The block FOM approximation eliminates the last p rows of \bar{H}_m and $\bar{g}^{(i)}$ and then solves the resulting system $H_m y^{(i)} = \bar{g}^{(i)}$. Then, the approximate solution $x^{(i)}$ will be calculated by (15), and finally, from the orthogonality of the column vectors of V_{m+p} , the relation (17) yields

```

(1) Choose a vector  $x$  such that  $[x, x] \neq 0$ 
(2) Define  $v_1 = x/\sqrt{[x, x]}$ 
(3) For  $j = 1, \dots, m$  Do:
(4)   For  $i = 1, \dots, j$  Do:
(5)     Compute  $h_{ij} := [Av_j, v_i]$  and  $t(v_i) = [v_i, v_i]$ 
(6)     Compute  $w_j := Av_j - \sum_{i=1}^j t(v_i)h_{ij}v_i$ 
(7)     If  $\sqrt{[w_j, w_j]} = 0$  then stop
(8)      $v_{j+1} = w_j/\sqrt{[w_j, w_j]}$ 
(9)      $t(v_{j+1}) = [v_{j+1}, v_{j+1}]$ 
(10)     $h_{j+1,j} = t(v_{j+1})\sqrt{[w_j, w_j]}$ 
(11)   EndDo
(12) EndDo

```

ALGORITHM 3. Indefinite Arnoldi's process.

```

(1) Input vectors  $x_1, \dots, x_r$ 
(2)  $r_{11} = \sqrt{[x_1, x_1]}$ 
(3) If  $r_{11} = 0$  stop
(4) else  $q_1 = x_1/r_{11}$  and  $t(q_1) = [q_1, q_1]$ 
(5) For  $j = 2, \dots, r$  Do
(6)   Compute  $r_{ij} := [x_j, q_i]$ , for  $i = 1, \dots, j-1$ 
(7)    $\hat{q} := x_j - \sum_{i=1}^{j-1} t(q_i)r_{ij}q_i$ 
(8)   If  $[\hat{q}, \hat{q}] = 0$  stop else  $q_j = \hat{q}/\sqrt{[\hat{q}, \hat{q}]}$ 
(9)    $t(q_j) = [q_j, q_j]$ 
(10) EndDo

```

ALGORITHM 4.

$$\|b^{(i)} - Ax^{(i)}\|_2 = \|\bar{g}^{(i)} - \bar{H}_m y^{(i)}\|_2. \quad (18)$$

3. Indefinite Block FOM Method

In [2], the indefinite Arnoldi's process builds a J -orthogonal basis for a nondegenerated Krylov subspace as shown in Algorithm 3.

In the following, after recalling the indefinite Gram-Schmidt orthogonalization (Algorithm 4), we will express the block analogue of this algorithm.

Algorithm 4 gives

$$x_j = \sum_{i=1}^{j-1} t(q_i)r_{ij}q_i, \text{ for } j = 1, \dots, m \quad (19)$$

and this implies that $X = Q\hat{J}R$ in which $Q^T J Q = \hat{J}$ and $\hat{J} = \text{diag}(t(q_1), \dots, t(q_m))$. Note that the indefinite modified Gram-Schmidt algorithm is similar to the above algorithm, except that here, the seventh row will be replaced as follows. Now see Algorithm 5.

- (i) For $i = 1, \dots, j-1$
- (ii) $\hat{q} = x_j - t(q_i)r_{ij}q_i$
- (iii) end

- (1) Choose vectors x_1, \dots, x_p such that the indefinite Gram-Schmidt orthogonalization gives result:
 $[x_1, \dots, x_p] = V_1 \hat{J}_1 R_1$ where $V_1^T J V_1 = \hat{J}_1$.
- (2) Define $H_{11} = V_1^T J A V_1$.
- (3) For $j = 1, \dots, m$ Do:
- (4) Compute $H_{ij} = V_i^T J A V_j$, for $i = 1, \dots, j$
- (5) Compute $W_j = A V_j - \sum_{i=1}^j V_i \hat{J}_i H_{ij}$
- (6) EndDo
- (7) Compute the indefinite Q-R factorization of $W_j = V_{j+1} \hat{J}_{j+1} H_{j+1,j}$.

ALGORITHM 5. Indefinite block Arnoldi's process.

- (1) Choose vectors x_1, \dots, x_p such that the indefinite Gram-Schmidt orthogonalization gives result:
 $[x_1, \dots, x_p] = V_1 \hat{J}_1 R_1$ where $V_1^T J V_1 = \hat{J}_1$
- (2) For $j = 1, \dots, m$ Do:
- (3) Compute $W_j := A V_j$
- (4) For $i = 1, \dots, j$ Do
- (5) $H_{i,j} := V_i^T \hat{J}_i W_j$
- (6) $W_j := W_j - V_i \hat{J}_i H_{ij}$
- (7) EndDo
- (8) Compute the indefinite QR-factorization of $W_j = V_{j+1} \hat{J}_{j+1} H_{j+1,j}$.
- (9) EndDo

ALGORITHM 6.

Note that J is the mentioned matrix in the indefinite scalar product (2) and \hat{J}_i is the acquired matrix by the indefinite Gram-Schmidt process. Now, a simple property of the algorithm is proved.

Proposition 1. Denote by I_k the $k \times k$ identity matrix and define the following matrices by the above algorithm:

$$\begin{aligned}
 U_m &= [V_1, \dots, V_m], \\
 H_m &= (H_{ij})_{1 \leq i, j \leq m}, H_{ij} \equiv 0 \text{ for } i > j + 1, \\
 E_m &= \text{matrix of the last } p \text{ columns of } I_{mp}, \\
 \tilde{J}_m &= \text{diag}(\hat{J}_1, \dots, \hat{J}_m), \hat{J}_1 \in \mathbf{C}^{p \times p}, \tilde{J}_m \in \mathbf{C}^{mp \times mp}.
 \end{aligned} \tag{20}$$

Then the following relation holds:

$$\begin{aligned}
 A U_m &= U_m \tilde{J}_m H_m + V_{m+1} \hat{J}_{m+1} V_{m+1, m} E_m^T, \tilde{J}_m U_m^T J A U_m \\
 &= \tilde{J}_m H_m, \\
 \hat{J}_m U_m^T J U_m &= I.
 \end{aligned} \tag{21}$$

Proof. The relation is straightforward by the following equalities which are derived from algorithm:

$$\begin{aligned}
 W_j &= A V_j - \sum_{i=1}^j V_i \hat{J}_i H_{ij} \text{ and } W_j = V_{j+1} \hat{J}_{j+1} H_{j+1, j}, A U_m = [A V_1, \dots, A V_m] \\
 &= \left[V_2 \hat{J}_2 H_{2,1} + V_1 \hat{J}_1 H_{11}, \dots, V_{m+1} \hat{J}_{m+1} H_{m+1, m} + \sum_{i=1}^m V_i \hat{J}_i H_{ij} \right] \\
 &= U_m \tilde{J}_m H_m + V_{m+1} \hat{J}_{m+1} H_{m+1, m} E_m^T.
 \end{aligned} \tag{22}$$

Algorithm 6 is the modified version of Algorithm 5 for which the indefinite QR-factorization is calculated by the indefinite modified Gram-Schmidt process.

As shown in Algorithm 6, we express the indefinite version of the block Arnoldi-Ruhe's variant, as another Krylov subspace method for which A acts on a group of vectors instead of just one vector.

According to Algorithm 7, the vector w is expressed as

$$w = A v_k - \sum_{i=1}^j t_i h_{i,k} v_i, \tag{23}$$

```

(1) Choose matrix  $V_1 = [v_1, \dots, v_p]$  such that for  $i = 1, \dots, p : [v_i, v_j] = 0, i \neq j$  and  $t_i = [v_i, v_i] = \pm 1$ .
(2) For  $j = p, \dots, m + p - 1$  Do
(3)   Set  $k := j - p + 1$ 
(4)   Compute  $w := Avk$ 
(5)   For  $i = 1, \dots, j$  Do
(6)      $h_{i,k} := [w, v_i]$ 
(7)      $w := w - t_i h_{i,k} v_i$ 
(8)   endDo
(9)   Compute  $a := \sqrt{|[w, w]|}$ 
(10)  If  $a = 0$  stop
(11)  Compute  $v_{j+1} = w/a$ 
(12)   $t_{j+1} = [v_{j+1}, v_{j+1}]$ 
(13)   $h_{j+1,k} = t_{j+1} a$ 
(14) End

```

ALGORITHM 7.

where $k = j - p + 1$ and also $w = t_{j+1} h_{j+1,k} v_{j+1}$. Thus,

$$AV_k = \sum_{i=1}^{k+p} t_i h_{i,k} v_i, \quad (24)$$

in which V_j indicates the $n \times j$ matrix with columns v_1, \dots, v_j . Finally, from there

$$AV_m = V_{m+p} \overset{\prime}{J} \bar{H}_m, \quad (25)$$

for $(m+p) \times m$ matrix \bar{H}_m .

Now, similar to the end of the previous section, consider the linear systems

$$Ax^{(i)} = b^{(i)}, \quad i = 1, \dots, p, \quad (26)$$

with matrix notation

$$AX = B, \quad (27)$$

in which $X = (x^{(1)}, \dots, x^{(p)})$ and $B = (b^{(1)}, \dots, b^{(p)})$, and assume that the initial block guess X_0 of initial guesses $x_0^{(i)}$ and the initial block residual R_0 are defined similar to the previous section.

$$R_0 \equiv (r_0^{(1)}, \dots, r_0^{(p)}), \quad (28)$$

in which $r_0^{(i)} := b^{(i)} - Ax_0^{(i)}$. Recall that in the case of a single system, that is when $p = 1$, the approximate solution X_m is chosen such that the correction $X_m - X$ lies in the Krylov subspace

$$K_m(A, r_0) := \text{span}\{r_0, Ar_0, \dots, A^{m-1}r_0\}. \quad (29)$$

A block Krylov space method for solving the p systems (3) is an iterative method that generates approximate solution X such that

$$X_m - X_0 \in \mathbb{K}m(A, R_0), \quad (30)$$

where R_0 is the initial residual as (28).

The indefinite block FOM computes the indefinite QR-factorization of R_0 :

$$R_0 = V_p \overset{\prime}{J} R, \quad (31)$$

such that $V_p = [v_1, \dots, v_p]$ and $V_p^t \overset{\prime}{J} V_p = \overset{\prime}{J}$.

Each of the approximate solutions has the form

$$x^{(1)} = x_0^{(1)} + V_m y^{(1)}. \quad (32)$$

Writing $X = (x^{(1)}, \dots, x^{(p)})$ and $Y = (y^{(1)}, \dots, y^{(p)})$, we have

$$X = X_0 + V_m Y. \quad (33)$$

Let E_1 be the $(m+p) \times p$ matrix as before. Then, the relation (24) results in

$$\begin{aligned} B - AX &= B - A(X_0 + V_m Y) = B - AX_0 - AV_m Y \\ &= R_0 - AV_m Y = [v_1, \dots, v_p] \overset{\prime}{J} R - V_{m+p} \overset{\prime}{J} \bar{H}_m Y \\ &= V_{m+p} \overset{\prime}{J} [E_1 R - \bar{H}_m Y]. \end{aligned} \quad (34)$$

The vector $\bar{g}^{(i)} \equiv E_1 R e_i$ is a vector whose components are zero except those from 1 to i which are extracts from the i th column of the upper triangular matrix R . While H_m and \bar{H}_m are defined as before, the indefinite block FOM (IBFOM) deletes the last p rows of \bar{H}_m and $\bar{g}^{(i)}$ and solves the resulting system $H_m y^{(i)} = g^{(i)}$. Then, the approximate solution $x^{(i)}$ is computed by (32).

4. Block Lanczos and Indefinite Block Lanczos Methods

In 1950, Lanczos proposed an algorithm (Algorithm 8) [9], which designed an orthogonal transformation of a symmetric

```

(1) Choose  $r$  initial orthonormal vectors  $V = [v_1, \dots, v_p]$  and set  $[t_{ij}]_{(m+p) \times m} = 0$ 
(2) For  $j = r, \dots, m + p$ 
(3)    $k = j - p + 1$ 
(4)   Compute  $f = Av_k$ 
(5)   If  $k \leq p$ 
(6)     For  $i = 1, \dots, k - 1$ 
(7)        $c = f^T v_i$ 
(8)        $f = f - cv_i$ 
(9)     end
(10)  else
(11)   For  $i = k - p, \dots, k - 1$ 
(12)      $c = f^T v_i$ 
(13)      $f = f - cv_i$ 
(14)   end
(15)  end
(16)  For  $i = k, \dots, j$ 
(17)    $t_{i,k} = f^T v_i$ 
(18)   If  $i \neq k$ 
(19)      $t_{i,k} = t_{k,i}$ 
(20)   end
(21)    $f = f - t_{ik} v_i$ 
(22)  end
(23)  If  $j < m + p$ 
(24)    $t_{j+1,k} = \|f\|$ 
(25)    $v_{j+1} = f / \|f\|$ 
(26)    $t_{k,j+1} = t_{j+1,k}$ 
(27)  end
(28) end
(29) For  $j = 2, \dots, p$ 
(30)    $k = m + j$ 
(31)    $f = Av_k$ 
(32)   For  $i = k - p, \dots, k - 1$ 
(33)      $c = f^T v_i$ 
(34)      $f = f - cv_i$ 
(35)   end
(36)   For  $i = k, \dots, m + p$ 
(37)      $t_{ik} = f^T v_i$ 
(38)     if  $i \neq k$ 
(39)        $t_{ki} = t_{ik}$ 
(40)     end
(41)   end
(42) end

```

ALGORITHM 8. Block Lanczos (Ruhe's variant).

matrix into a tridiagonal matrix T . It can be applied to Krylov subspace methods for solving the symmetric matrix linear systems as well as the eigenvalue problem of the symmetric matrix. The block Lanczos algorithm was developed by Peter Montgomery and published in 1995, [10]; it is based on, and bears a strong resemblance to, the Lanczos algorithm for finding the eigenvalues of large sparse real matrices.

Due to the login of J and \hat{J} into the IBFOM algorithm, the better performance of the BFOM algorithm can be expected than the IBFOM algorithm, and this seen practically in numerous examples. We know there is a Ruhe's version of the block Lanczos algorithm that is used for the symmetric

coefficient matrix mode. Our purpose in this section is to build its indefinite mode, named indefinite block Lanczos (Ruhe's variant) algorithm, that is used for J -symmetric matrices. First, we express the block Lanczos algorithm and then we design its indefinite version (Algorithm 8).

In Algorithm 9, we suggest the indefinite version of the Algorithm 8:

Algorithm 9 transforms the matrix A in to the matrix T , and as seen for IBFOM algorithm, the following relation satisfies for them:

$$AV_m = V_{m+p} \hat{J} \bar{T}_m. \quad (35)$$

```

(1) Choose matrix  $V = [v_1, \dots, v_p]$  with  $J$ -orthonormal columns and set  $[t_{ij}]_{(m+k) \times m} = 0$  and  $[\hat{J}_{ij}]_{m+p, m+p} = 0$ 
(2) For  $i = 1, \dots, p$ 
(3)    $\hat{J}_{ii} = v_i^T J v_i$ 
(4) end
(5) For  $j = r, \dots, m + p$ 
(6)    $k = j - p + 1$ 
(7)   Compute  $f = A v_k$ 
(8)   If  $k \leq p$ 
(9)     For  $i = 1, \dots, k - 1$ 
(10)       $c = f^T J v_i$ 
(11)       $f = f - \hat{J}_{ii} c v_i$ 
(12)     end
(13)   else
(14)     For  $i = k - p, \dots, k - 1$ 
(15)       $c = f^T J v_i$ 
(16)       $f = f - \hat{J}_{ii} c v_i$ 
(17)     end
(18)   end
(19)   For  $i = k, \dots, j$ 
(20)      $t_{i,k} = f^T J v_i$ 
(21)     If  $i \neq k$ 
(22)        $t_{k,i} = t_{i,k}$ 
(23)     end
(24)      $f = f - \hat{J}_{ii} t_{ik} v_i$ 
(25)   end
(26)   If  $j < m + p$ 
(27)      $t_{j+1,k} = \sqrt{[f, f]}$ 
(28)      $v_{j+1} = f / t_{j+1}$ 
(29)      $\hat{J}_{j+1, j+1} = v_{j+1}^T J v_{j+1}$ 
(30)      $t_{j+1, k} = \hat{J}_{j+1, j+1} t_{j+1, k}$ 
(31)      $t_{k, j+1} = t_{j+1, k}$ 
(32)   end
(33) end
(34) For  $j = 2, \dots, p$ 
(35)    $k = m + j$ 
(36)    $f = A v_k$ 
(37)   For  $i = k - p, \dots, k - 1$ 
(38)      $c = f^T J v_i$ 
(39)      $f = f - J_{ii} c v_i$ 
(40)   end
(41)   For  $i = k, \dots, m + p$ 
(42)      $t_{ij} = f^T J v_i$ 
(43)     if  $i \neq k$ 
(44)        $t_{ki} = t_{ik}$ 
(45)     end
(46)   end
(47) end

```

ALGORITHM 9. Indefinite block Lanczos.

As seen for the IBFOM algorithm, by considering the linear system $AX = B$, we gain some relations similar to what obtained at the end of the third section, and by solving the linear systems $T_m y^{(i)} = g^{(i)}$ and by substituting $y^{(i)}$ in relation (32), the solution $x^{(i)}$ is earned for the system. Here, T_m and \bar{T}_m are defined similar to H_m and \bar{H}_m .

5. Numerical Examples

Suppose that A is an $n \times n$ and J -symmetric matrix and B is a $n \times p$ matrix. Our purpose is to solve the linear system $AX = B$ via three methods: block FOM (BFOM), indefinite block FOM (IBFOM), and indefinite block Lanczos (IBLAN). In the following examples, t will be considered

the time it takes to run each of the algorithms and will be defined as the average of residuals, i.e.,

$$\text{if } \epsilon = \left\| Ax^{(i)} - b^{(i)} \right\|_2 \text{ for } 1 \leq i \leq p, \quad (36)$$

then $\epsilon = \sum_{i=1}^p \epsilon_i / p$. We define m the number of iterations which is chosen arbitrarily.

Example 2. Consider matrix A as follows:

$$A_{n \times n} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \quad (37)$$

in which $A_{11}, A_{12}, A_{21}, A_{22}$ are all tridiagonal matrices with arbitrary arrays belonging to $(0, 1)$. In addition, A_{11}, A_{22} are symmetric and $A_{21} = -A_{12}^T$. With this selection and by defining J as below

$$J = \begin{pmatrix} I_{\frac{n}{2}} & 0 \\ 0 & -I_{\frac{n}{2}} \end{pmatrix}, \quad (38)$$

the matrix A is J -symmetric. If $n = 600$, $m = 300$, and $p = 5$, then the BFOM, IBFOM, and IBLAN method results are as shown in Table 1.

Example 3. Consider the $n \times n$ matrices A and J as follows:

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix}, \quad (39)$$

$$J = \begin{pmatrix} I & 0 & 0 \\ 0 & -I & 0 \\ 0 & 0 & I \end{pmatrix},$$

in which $A_{11}, A_{12}, A_{21}, A_{22}$ are all tridiagonal matrices with arbitrary arrays belonging to $(0, 10)$. Besides, A_{12}, A_{13} , and A_{23} are $(n/3) \times (3/n)$ matrices by zero arrays except for their diagonal and subdiagonal arrays which are randomly selected in $(0, 1)$. Also, suppose that $A_{21} = -A_{12}^T, A_{32} = -A_{23}^T$ and $A_{31} = A_{13}^T$. Under these circumstances, A is J -symmetric. By letting $n = 600$, $m = 300$, and $p = 5$, we the results shown in Table 2.

Example 4. Consider A and J as the previous example. By letting $n = 600$, $m = 500$, and $p = 20$, we obtain the results shown in Table 3.

In the above examples, B and X_0 are $n \times p$ matrices with randomly chosen arrays in $(0, 1)$.

6. Conclusion

As can be seen from the discussed algorithms in this paper, the BFOM and IBFOM algorithms are used to solve the linear

TABLE 1

Method	t	ϵ
BFOM	0.5	4.8×10^{-7}
IBFOM	1.2	3.1×10^{-6}
IBLAN	0.39	2.9×10^{-6}

TABLE 2

Method	t	ϵ
BFOM	0.45	5×10^{-7}
IBFOM	1.2	2×10^{-5}
IBLAN	0.36	5×10^{-7}

TABLE 3

Method	t	ϵ
BFOM	3.3	1.6×10^{-4}
IBFOM	6.2	1.9×10^{-3}
IBLAN	2.2	1.6×10^{-3}

systems with arbitrary coefficients matrices. But since in the IBFOM algorithm the indefinite inner product is used instead of the standard inner product and the matrix J appears in the calculations, the number of arithmetic operations is increased, and as a result, the IBFOM algorithm does not perform better than the BFOM algorithm. But our goal is to solve the block linear systems with the J -symmetric coefficient matrices. So the performance of the IBLAN algorithm for solving such systems is important for us. The block linear systems with the J -symmetric coefficients matrices can only be solved by the BFOM, IBFOM, and IBLAN algorithms. As can be seen in the numerical examples, the performance of the IBLAN algorithm is far better than the IBFOM algorithm and even better than the BFOM algorithm due to the difference in the number of arithmetic operations, and this is what we have theoretically achieved. The result in a sentence is that the best way to solve the block linear systems with the J -symmetric coefficients matrices is to use the IBLAN algorithm.

Data Availability

All data is available.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] M. Aliyari and M. Ghasemi Kamalvand, "A method of indefinite Krylov subspace for eigenvalue problem," *Mathematical Problems in Engineering*, vol. 2018, Article ID 2919873, 5 pages, 2018.

- [2] K. N. Asil and M. G. Kamalvand, "Some hyperbolic iterative methods for linear systems," *Journal of Applied Mathematics*, vol. 2020, Article ID 9874162, 8 pages, 2020.
- [3] D. Loher, *Reliable nonsymmetric block Lanczos algorithms*, [Ph.D. thesis], ETH Zurich, Zurich, Switzerland, 2006, Diss. no. 16337.
- [4] D. P. O'Leary, "The block conjugate gradient algorithm and related methods," *Linear Algebra and its Applications*, vol. 29, pp. 293–322, 1980.
- [5] S. Roland, W. Freund, and M. Malhotra, "A block QMR algorithm for non-Hermitian linear systems with multiple right-hand sides," *Linear Algebra and its Applications*, vol. 254, pp. 119–157, 1997.
- [6] Y. Saad, *Iterative methods for sparse linear systems*, Society for Industrial and Applied Mathematics, 3600 University City Science Center Philadelphia, PA. United States, 2003.
- [7] B. Vital, *Etude de quelques m taille sur multiprocesseur*, [Ph.D. thesis], 1990, Universit [12] ethodes de r esolution de probl'e de Rennes.
- [8] A. Ruhe, "Implementation aspects of band Lanczos algorithms for computation of eigenvalues of large sparse symmetric matrices," *Mathematics of computation*, vol. 33, no. 146, pp. 680–687, 1979.
- [9] C. Lanczos, "An iteration method for the solution of the eigenvalue problem of linear differential and integral operators," *Journal of Research of the National Bureau of Standards*, vol. 45, no. 4, pp. 255–282, 1950.
- [10] P. L. Montgomery, "A block dependencies over GF(2)," in *Proceedings EUROCRYPT95*, vol. 921 of Lecture Notes in Computer Science, p. 106120, Springer-Verlag, Berlin, 1995.