J. KSIAM Vol.24, No.4, 375–385, 2020 http://dx.doi.org/10.12941/jksiam.2020.24.375

THE INDEFINITE LANCZOS J-BIOTHOGONALIZATION ALGORITHM FOR SOLVING LARGE NON-J-SYMMETRIC LINEAR SYSTEMS

MOJTABA GHASEMI KAMALVAND[1†] AND KOBRA NIAZI ASIL[1]

[1]DEPARTMENT OF MATHEMATICAL SCIENCES, LORESTAN UNIVERSITY, IRAN
Email address: †ghasemi.m@lu.ac.ir

ABSTRACT. In this paper, a special indefinite inner product, named hyperbolic scalar product, is used and all acquired results have been raised and proved with the proviso that the space is equipped with this indefinite scalar product. The main objective is to be introduced and applied an indefinite oblique projection method, called Indefinite Lanczos J-biorthogonalizatiom process, which in addition to building a pair of J-biorthogonal bases for two used Krylov subspaces, leads to the introduction of a process for solving large non-J-symmetric linear systems, i.e., Indefinite two-sided Lanczos Algorithm for Linear systems.

1. INTRODUCTION

Today, iterative methods are used commonly for solving problems such as large eigenvalue problems and large sparse linear systems and that is because of their easier optimal use in high-performance computers than direct methods. The growing need to solve very large linear systems in some fields of science has led to more attention to iterative techniques. Krylov subspace methods are widely used for the iterative solution of $n \times n$ linear systems of equations and also for solving large eigenvalue problems. For example refer to [1] and [2].

Definition 1.1. *A Krylov subspace is a subspace spanned by a sequence of vectors generated by a given matrix and a vector as follows. Given a matrix A and a starting vector v_0, the Krylov subspace $K_m(A, v_0)$ is spanned by a sequence of m column vectors:*

$$K_m(A, v_0) = span\{v_0, Av_0, A^2v_0, ..., A^{m-1}v_0\}.$$

Recall that the indefinite inner product $[.,.]$ in $\mathbf{C}^n$ has all the 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 $y \in \mathbf{C}^n$, then $x = 0$. This kind of inner product may be applied in some areas of sciences and is commonly

Received by the editors October 23 2020; Revised December 18 2020; Accepted in revised form December 21 2020; Published online December 25 2020.

2010 *Mathematics Subject Classification.* Mathematics Subject Classification. 15A24 - 15B99.

Key words and phrases. Krylov subspace methods, indefinite inner products, Hyperbolic scalar product, non-J-symmetric matrix, J-biorthogonalization, Indefinite Lanczos J-biort-hogonalizatiom Algorithm, Indefinite two-sided Lanczos Algorithm.

† Corresponding author.

defined as $[x, y] = y^* Jx$ where $J \in \mathbf{C}^{n\times n}$ 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:

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

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

$$[x, y] = y^* Jx = \sum\nolimits_{i=1}^{n} j_{ii} x_i \overline{y}_i.$$

More applications of such products can be found in [3, 4, 5, 6]. An excellent and elegant example of working with hyperbolic scalar product can be seen in [7]. In this article it has been investigated the existence of a decomposition which would resemble the tridiagonal Schur decomposition, but with respect to the given hyperbolic scalar product. He proves that a hyperbolic Schur decomposition can be constructed, but not for all square matrices. He also provides sufficient requirements for its existence and offers examples showing why such a decomposition does not exist for all matrices.

In [8], by considering $\mathbf{C}^n$ with the indefinite scalar product (4.2), a number of the Krylov subspace methods have been reviewed and restructured. Algorithms presented there for indefinite case, are already known for definite case. The mentioned methods are Arnoldi, Full orthogonalization and Lanczos. In that paper, the indefinite Arnoldi's process builds a J-orthogonal basis for a nondegenerated Krylov subspace as follows. (You can also see [9])

Algotithm1: Indefinite Arnoldi's process,

(1) Choose a vector x such that $[x, x] \neq 0$
(2) Define $v_1 = \frac{x}{\sqrt{|[x,x]|}}$
(3) For $j = 1, ..., m$ Do:
(4) For $i = 1, ..., j$ 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) $h_{j+1,j} = \sqrt{|[w_j, w_j]|}$
(8) if $h_{j+1,j} = 0$ then stop
(9) $v_{j+1} = \frac{w_j}{h_{j+1,j}}$
(10) EndDo
(11) EndDo

Note that a subspace M is said to be nondegenerate, with respect to the indefinite inner product $[.,.]$ if $x \in M$ and $[x, y] = 0$ for every $y \in M$ imply that $x = 0$. For example, the nondegeneracy property of the indefinite inner product $[.,.]$ yields that $\mathbf{C}^n$ is nondegenerate. On the other hand, for every nondegenerate subspace M it is possible to choose $x \in M$ such that $[x, x] \neq 0$. Because if this is not true, then $[x, x] = 0$ for every $x \in M$. But this ensures that

$$[x, y] = \tfrac{1}{4}\{[x + y, x + y] + i[x + iy, x + iy] - [x - y, x - y] - i[x - iy, x - iy]\}$$

and this shows that $[x, y] = 0$ for all $x, y \in M$. But this is a contradiction. with these explanations, there is no restriction to choose a vector as x such that $[x, x] \neq 0$ in the first line of the above algorithm.

Also, an orthogonal projection method, named indefinite full orthogonalization method (IFOM) finds an approximate solution of the linear system $Ax = b$ and finally, indefinite Lanczos method (ILM) is expressed, a method which considers indefinite Arnoldi's method for J-Hermitian matrices. Indeed, IFOM and ILM are rewritten of full orthogonalization method (FOM) and Lanczos method respectively, by considering the indefinite scalar product (4.2). (see [8])

This paper is organized as follows: In the second section we will recall a brief overview of the Lanczos biorthogonalization procedure and Two sides Lanczos Algorithm for linear systems. In the third section $\mathbf{C}^n$ is equipped with the indefinite scalar product (4.2) and then deals to rewrite the algorithms described in the second section and the results will be proved. At the end, we will give some numerical examples in section 4.

We should point out that throughout this article the capital letters refer to matrices and single index lowercase letters refer to column vectors. A^T and A^* denote the transpose and the conjugate transposed of matrix A, respectively. $A^{[T]}$ implies JA^TJ and the matrix A is said to be non-J-symmetric if $A \neq A^{[T]}$. In sections 3 and 4, $\mathbf{C}^n$ is considered with the indefinite scalar product (4.2). Two sets of vectors $\{v_1, ..., v_n\}$ and $\{w_1, ..., w_n\}$ are said to be J-biorthogonal by considering this scalar product if: $[v_i, w_i] = \pm 1$ and $[v_i, w_j] = 0$ for $i \neq j$, $1 \leq i, j \leq n$.

2. Two-sided Lanczos algorithm

The mentioned methods in the previous section are a number of iterative Krylov subspace methods which get help from orthogonalization of the Krylov vectors to get orthogonal bases for Krylov subspaces or to get an approximate solution of a linear system of equations. Besides these types of methods a number of non-orthogonal projective methods have been developed as a class of Krylov subspace methods where are based on bi-orthogonalization Algorithms. An interesting example of oblique projective methods is proposed by Lanczos in [10] which is considered as one of the most efficient iterative methods for a nonsymmetric given system, and it is used in a variety of application areas. A simple version of his method can be described as follows.

Let v_1 and w_1 be two initial vectors. Then by considering the Krylov spaces $K_m(A, v_1)$ and $K_m(A^T, w_1)$, the non-Hermitian Lanczos algorithm generates successively two sets of vectors $\{v_1, ..., v_n\}$ and $\{w_1, ..., w_n\}$ which span these two spaces and which are bi-orthogonal, i.e., $(v_i, w_j) = \delta_{ij}$ for $1 \leq i, j \leq n$, where δ_{ij} is the Kronecker symbol. The algorithm that achieves this, can be described as following:

Algorithm2: The Lanczos bi-orthogonalization procedure:

(1) Choose two vectors v_1 , w_1 such that $(v_1, w_1) = 1$
(2) Set $\beta_1 = \delta_1 \equiv 0, w_0 = v_0 \equiv 0$
(3) For $j = 1, ..., m$ Do:
(4) $\alpha_j = (Av_j, w_j)$

(5) $\hat{v}_{j+1} = Av_j - \alpha_j v_j - \beta_j v_{j-1}$
(6) $\hat{w}_{j+1} = A^T w_j - \alpha_j w_j - \delta_j w_{j-1}$
(7) $\delta_{j+1} = |(\hat{v}_{j+1}, \hat{w}_{j+1})|^{1/2}$. If $\delta_{j+1} = 0$ Stop
(8) $\beta_{j+1} = (\hat{v}_{j+1}, \hat{w}_{j+1})/\delta_{j+1}$
(9) $w_{j+1} = \hat{w}_{j+1}/\beta_{j+1}$
(10) $v_{j+1} = \hat{v}_{j+1}/\delta_{j+1}$
(11) End Do

Here β_{j+1} and δ_{j+1} is chosen such that $\beta_{j+1}\delta_{j+1} = (\hat{v}_{j+1}, \hat{w}_{j+1})$ and the final outcome is a pair of bi-orthogonal bases $\{v_1, ..., v_m\}$ and $\{w_1, ..., w_m\}$ of $K_m(A, v_1)$ and $K_m(A^T, w_1)$, respectively. But does not provide explicitly an approximate solution for nonsymmetric linear systems. In [10] Lanczos provided an ingenious method to built up such an approximation. Consider the linear system $Ax = b$ in which A is $n \times n$ and nonsymmetric matrix and x_0 is an initial guess for the solution and note $r_0 = b - Ax_0$ as its residual vector. Then the Lanczos algorithm to solve this linear system can be raised as follows:

Algorithm3: Two-sided Lanczos Algorithm (TSL) for linear systems

(1) Compute $r_0 = b - Ax_0$ and $\beta := \|r_0\|_2$
(2) Run m steps of the nonsymmetric Lanczos Algorithm, i.e.,
(3) Start with $v_1 := r_0/\beta$, and any w_1 such that $(v_1, w_1) = 1$
(4) Generate the Lanczos vectors $v_1, ..., v_m, w_1, ..., w_m$
(5) and the tridiagonal matrix T_m from Algorithm 1
(6) Compute $y_m = T_m^{-1}(\beta e_1)$ and $x_m := x_0 + V_m y_m$.

3. INDEFINITE TWO-SIDED LANCZOS ALGORITHM

Assuming the space $\mathbf{C}^n$ with the indefinite scalar product (4.2), the purpose of this section is to rewrite the above two algorithms. In the following, assume that all entries of the matrices and vectors are real.

The final outcome of the following algorithm is a pair of J-biorthogonal bases $\{v_1, ..., v_m\}$ and $\{w_1, ..., w_m\}$ of $K_m(A, v_1)$ and $K_m(A^{[T]}, w_1)$ respectively.

Algorithm4: The indefinite Lanczos J-biorthogonalization procedure:

(1) Choose two vectors x , y such that $[x, y] \neq 0$ and set:

$$v_1 = x/\sqrt{|[x, y]|},\ w_1 = y/\sqrt{|[x, y]|} \text{ and } t_1 := [v_1, w_1].$$

(2) Set $\beta_1 = \delta_1 \equiv 0, w_0 = v_0 \equiv 0$
(3) For $j = 1, ..., m$ Do:
(4) $\alpha_j = [Av_j, w_j]$
(5) $\hat{v}_{j+1} = Av_j - t_j\alpha_j v_j - t_{j-1}\beta_j v_{j-1}$
(6) $\hat{w}_{j+1} = A^{[T]} w_j - t_j\alpha_j w_j - t_{j-1}\delta_j w_{j-1}$
(7) $\delta_{j+1} = |[\hat{v}_{j+1}, \hat{w}_{j+1}]|^{1/2}$. If $\delta_{j+1} = 0$ Stop
(8) $\beta_{j+1} = |[\hat{v}_{j+1}, \hat{w}_{j+1}]|/\delta_{j+1}$
(9) $w_{j+1} = \hat{w}_{j+1}/\beta_{j+1}$
(10) $v_{j+1} = \hat{v}_{j+1}/\delta_{j+1}$

(11) $t_{j+1} = [v_{j+1}, w_{j+1}]$
(12) $\delta_{j+1} = t_{j+1}\delta_{j+1}$
(13) $\beta_{j+1} = t_{j+1}\beta_{j+1}$
(14) End Do

Denote by H_m the following obtained tridiagonal matrix from the algorithm,

$$H_m = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \delta_2 & \alpha_2 & \beta_3 & & \\ & . & . & . & \\ & & \delta_{m-1} & \alpha_{m-1} & \beta_m \\ & & & \delta_m & \alpha_m \end{pmatrix}.$$

Definition 3.1. *The minimal polynomial of a vector v is a nonzero monic polynomial p of lowest degree such that $p(A)v = 0$ and the grade of v means the degree of the minimal polynomial of v with respect to A.*

According to the above algorithm the v_i's belong to $K_m(A, v)$ and the w_i's are in $K_m(A^{[T]}, w_i)$ and the following proposition can be proved.

Proposition 3.1. *The Krylov subspace $K_m(A, v_1)$ is of dimension m if and only if the grade μ of v_1 with respect to A is not less than m. Therefore,*

$$dim(K_m(A, v_1)) = min\{m, grade(v)\}$$

Proof 3.1. *See* [1].

Proposition 3.2. *Assuming that all the matrices and vectors are real, if Algorithm 3 does not breakdown before step m, then the vectors v_i, w_j, $i, j = 1, ..., m$, form a J-biorthogonal system. Moreover, $\{v_1, ..., v_m\}$ is a basis of $K_m(A, v_1)$ and $\{w_1, ..., w_m\}$ is a basis of $K_m(A^{[T]}, w_1)$ and the following relations holds,*

$$AV_m = V_m \acute{J_m} H_m + \delta_{m+1} v_{m+1} e_m^T, \tag{3.1}$$

$$A^{[T]} W_m = W_m \acute{J_m} H_m^T + \beta_{m+1} w_{m+1} e_m^T, \tag{3.2}$$

in which

$$\acute{J_m} = diag(t_1, ..., t_m) \text{ and } t_i := [v_i, w_i],\ i = 1, ..., m.$$

$$W_m^* J A V_m = H_m. \tag{3.3}$$

Proof 3.2. *To show the J-biorthogonality of the vectors v_i and w_i, we use induction. According to the assumption $|[v_1, w_1]| = 1$. Now, suppose that the vectors $\{v_1, ..., v_j\}$ and $\{w_1, ..., w_j\}$ are J-biorthogonal. It will be shown that the vectors $\{v_1, ..., v_{j+1}\}$ and $\{w_1, ..., w_{j+1}\}$ are also J-biorthogonal. First, we prove that $[v_{j+1}, w_i] = 0$, for $i \leq j$. If $i = j$, then by the lines 5 and 10 of the algorithm,*

$$[v_{j+1}, w_j] = [\hat{v}_{j+1}/\delta_{j+1}, w_j] = \delta_{j+1}^{-1}[Av_j - t_j\alpha_j v_j - t_{j-1}\beta_j v_{j-1}, w_j] =$$
$$\delta_{j+1}^{-1}([Av_j, w_j] - t_j\alpha_j[v_j, w_j] - t_{j-1}\beta_j[v_{j-1}, w_j]).$$

The last term in the above equality is zero according to the induction hypothesis and two other terms delete each other by the definition of α_j and that $t_j^2 = 1$. Now, consider $[v_{j+1}, w_i]$, for $i < j$. With regard to the lines 6 and 9 of the algorithm,

$$[v_{j+1}, w_i] = \delta_{j+1}^{-1}([Av_j, w_i] - t_j\alpha_j[v_j, w_i] - t_{j-1}\beta_j[v_{j-1}, w_i]) =$$
$$\delta_{j+1}^{-1}([v_j, \hat{w}_{i+1} + t_i\alpha_i w_i + t_{i-1}\delta_i w_{i-1}] - t_{j-1}\beta_j[v_{j-1}, w_i]) =$$
$$\delta_{j+1}^{-1}([v_j, \beta_{i+1}w_{i+1} + t_i\alpha_i w_i + t_{i-1}\delta_i w_{i-1}] - t_{j-1}\beta_j[v_{j-1}, w_i]).$$

According to the induction hypothesis all of the above inner products vanish, for $i < j-1$. For $i = j-1$, the scalar product is

$$[v_{j+1}, w_{j-1}] = \delta_{j+1}^{-1}([v_j, t_j\beta_j w_j + t_{j-1}\alpha_{j-1}w_{j-1} + t_{j-2}\delta_{j-1}w_{j-2}] - t_{j-1}\beta_j[v_{j-1}, w_{j-1}]) =$$
$$\delta_{j+1}^{-1}(t_j\beta_j[v_j, w_j] - t_{j-1}\beta_j[v_{j-1}, w_{j-1}]) = \delta_{j+1}^{-1}(t_j^2\beta_j - t_{j-1}^2\beta_j) = 0.$$

It can be shown in an identical process that $[v_i, w_{j+1}] = 0$, for $i \leq j$. Finally, $|[v_{j+1}, w_{j+1}]| = 1$. Because,

$$[v_{j+1}, w_{j+1}] = [\hat{v}_{j+1}/\delta_{j+1}, \hat{w}_{j+1}/\beta_{j+1}] = \frac{1}{\delta_{j+1}}\frac{1}{\beta_{j+1}}[\hat{v}_{j+1}, \hat{w}_{j+1}] =$$
$$\frac{1}{\delta_{j+1}}\frac{\delta_{j+1}}{|[\hat{v}_{j+1}, \hat{w}_{j+1}]|}[\hat{v}_{j+1}, \hat{w}_{j+1}] = \frac{[\hat{v}_{j+1}, \hat{w}_{j+1}]}{|[\hat{v}_{j+1}, \hat{w}_{j+1}]|} = \pm 1.$$

This completes the induction proof.

Now, we show that $\{v_1, ..., v_m\}$ is a generating set for $K_m(A, v_1)$. First, we show by induction that each vector v_j is as $q_{j-1}(A)v_1$ where q_{j-1} is a polynomial of degree $j-1$. For $j = 1$, let $q_0(t) \equiv 1$. Then, obviously $v_1 = q_0(A)v_1$. Suppose that the result is true for all integers $\leq j$ and consider v_{j+1}. Then, by considering algorithm 4, we have

$$\delta_{j+1}v_{j+1} = Av_j - t_j\alpha_j v_j - t_{j-1}\beta_j v_{j-1}.$$

This shows that v_{j+1} is expressed as $q_j(A)v_1$ in which q_j is of degree j and completes the proof. On the other hand, by the above proposition, $K_m(A, v_1)$ is a subspace of all vectors in $\mathbf{R}^n$ which can be written as $x = P(A)v$, where P is a polynomial of degree not exceeding $m-1$.

From the above, it follows that $\{v_1, ..., v_m\}$ spans $K_m(A, v_1)$. Thus, that is its basis.

We show (3.1). For this, consider the lines 5 and 10. Then,

$$\delta_{j+1}v_{j+1} = Av_j - t_j\alpha_j v_j - t_{j-1}\beta_j v_{j-1},$$

thus,

$$Av_j = t_j\alpha_j v_j + t_{j-1}\beta_j v_{j-1} + \delta_{j+1}v_{j+1}.$$

and we earn

$$AV_m = V_m\acute{J}_m H_m + \delta_{m+1}v_{m+1}e_m^T.$$

For (3.2), according to the lines 6 and 10, we have the following relationship,

$$\beta_{j+1}w_{j+1} = A^{[T]}w_j - t_j\alpha_j w_j - t_{j-1}\delta_j w_{j-1}.$$

Therefor,

$$A^{[T]}W_m = W_m\acute{J}_m H_m^T + \beta_{m+1}w_{m+1}e_m^T.$$

To get to (3.3), left-multiply the relation (3.1) by $W_m^* J$. Then,

$$W_m^* JAV_m = W_m^* JV_m\acute{J}_m H_m + \delta_{m+1}W_m^* JV_{m+1}e_m^T.$$

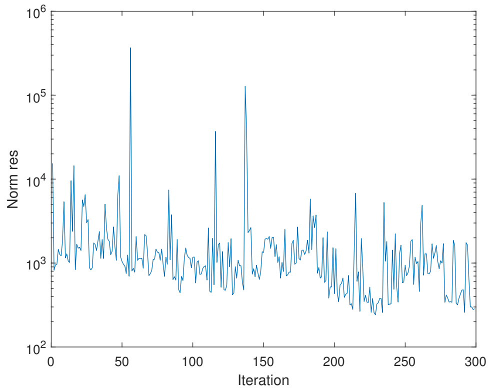

FIGURE 1. Figure for the inefficiency of the ITSL algorithm, for the J-symmetric matrix.

But, according to the J-biorthogonality of the vectors $\{v_i\}_{i=1}^m$ and $\{w_i\}_{i=1}^m$, we have

(1) $W_m^* J V_m = \acute{J}_m$,
(2) $W_m^* J V_{m+1} = 0$.

Thus,

$$W_m^* J A V_m = \acute{J}_m \acute{J}_m H_m + 0 = H_m.$$

Corollary 3.1. *With the assumptions of the above proposition, there is*

$$\acute{J}_m W_m^* J W_m = I_m,$$
$$\acute{J}_m W_m^* J A V_m = \acute{J}_m H_m.$$

Corollary 3.2. *If the algorithm 4 does not stop before the step n then the matrices A and $\acute{J}_n H_n$ are similar.*

In continuing, consider the linear system $Ax = b$ in which A is a real $n \times n$ matrix and is not J-symmetric. The purpose is to provide an approximate solution for that. Let x_0 as an initial guess to the solution and $r_0 = b - Ax_0$ as its residual vector. Set $v_1 = r_0/\beta$ and $\beta = \sqrt{|[r_0, r_0]|}$. The indefinite Lanczos method consists of constructing a sequence of vector x_m defined by the following two conditions:

(1) $x_m - x_0 \in K_m(A, v_1)$,
(2) $x_m = b - Ax_m[\perp]K_m(A^{[T]}, w_1)$,

where $[\perp]$ indicates to orthogonality under the indefinite scalar product (4.2). The condition (1) lets us to write the approximate solution as $x_m = x_0 + V_m y_m$. The second condition causes

$$b - Ax_m[\perp]w_i, \ i = 1, ..., m.$$

Thus, for $i = 1, ..., m$

$$[b - Ax_m, w_i] = 0 \Rightarrow [b - A(x_0 + V_m y), w_i] = 0 \Rightarrow [r_0 - AV_m y, w_i] = 0 \Rightarrow$$
$$[r_0, w_i] = [AV_m y, w_i] \Rightarrow w_i^* J r_0 = w_i^* JAV_m y.$$

Now, assuming that the $m \times m$ matrix $W_m^* JAV_m$ is nonsingular and according to (3.3), we earn

$$y = H_m^{-1} W_m^* J r_0 \tag{3.4}$$

and by the J-biorthogonality of $\{v_i\}_{i=1}^m$ and $\{w_i\}_{i=1}^m$,

$$W_m^* J r_0 = W_m^* J \beta v_1 = t_1 \beta e_1,$$

where $t_1 = [w_1, v_1]$. Thus, $y = H_m^{-1} t_1 \beta e_1$ and $x_m = x_0 + V_m H_m^{-1} t_1 \beta e_1$. These explanations can be organized as follows:

Algorithm5: Indefinite Two-sided Lanczos Algorithm (ITSL) for linear systems:

(1) Compute $r_0 = b - Ax_0$ and $\beta := \sqrt{|[r_0, r_0]|}$
(2) Run m steps of the indefinite Lanczos J-biorthogonalization Algorithm, i.e.,
(3) Start with $v_1 := r_0/\beta$, and any w_1 such that $[v_1, w_1] = \pm 1$
(4) Generate the vectors $v_1, ..., v_m, w_1, ..., w_m$
(5) and the tridiagonal matrix H_m from Algorithm 4
(6) Compute $y_m = H_m^{-1} t_1 \beta e_1$ and $x_m := x_0 + V_m y_m$.

Proposition 3.3. *The residual vector of the approximate solution x_m calculated by Algorithm 5 is such that*

$$b - Ax_m = -\delta_{m+1} e_m^T y_m v_{m+1}.$$

Proof 3.3. *According to (3.1) and (3.4), the argument is straightforward:*

$$b - Ax_m = b - A(x_0 + V_m y_m) = r_0 - AV_m y_m =$$
$$\beta v_1 - V_m \acute{J}_m H_m y_m - \delta_{m+1} e_m^T y_m v_{m+1} = \beta v_1 - V_m \acute{J}_m H_m H_m^{-1} t_1 \beta e_1 - \delta_{m+1} e_m^T y_m v_{m+1}.$$

But

$$V_m \acute{J}_m t_1 \beta e_1 = t_1^2 \beta V_m e_1 = \beta v_1$$

and this completes the proof.

4. NUMERICAL EXAMPLES

It should be noted that we have written MATLAB software programs for algorithms 4 and 5 presented in the previous section and the following examples are calculated with these programs.

Example 4.1. *Consider the block diagonal matrix*

$$J = diag(I_{n/4}, -I_{n/4}, I_{n/4}, -I_{n/4}) \tag{4.1}$$

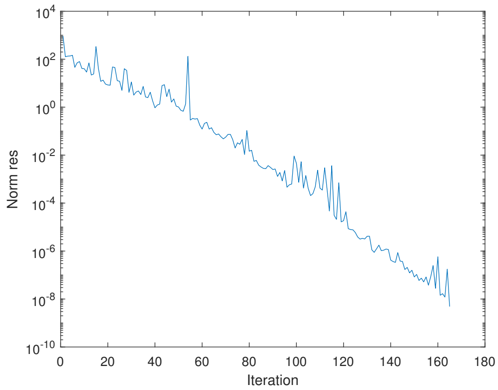

FIGURE 2. Figure for the efficiency of the ITSL algorithm, for the non-J-symmetric matrix.

in which $n = 300$ *and suppose that* A *defined as follows*

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{pmatrix} \tag{4.2}$$

is a J*-symmetric matrix. I.e.,* $A = JA^TJ$ *or equivalently*

$$-A_{12}^T = A_{21},\ -A_{23}^T = A_{32},\ -A_{34}^T = A_{43}$$
$$A_{13}^T = A_{31},\ A_{24}^T = A_{42},\ -A_{14}^T = A_{41}.$$

Now, assume that all A_{ij} *in* A *are diagonal matrices and their diagonal entries have randomly been selected from zero to 10. Except for* $(n/4) \times (n/4)$ *matrix* A_{14} *which if* $j - i < 9n/16 + 3$ *then its* (i, j)*st entry is a nonzero number in the interval* $(0, 10)$ *and the rest of its entries are zero.*

Now, consider the linear system $Ax = b$ *in which* A *is as above and* b *is an* $n \times 1$ *vector with random entries belonging to the interval* $(0, 10)$*. Take the following stop condition* $\|x - x_m\| < \epsilon$ *in which* $\epsilon = 10^{-8}$ *and* x_m *is the approximate solution of the* m*-th stage. With these assumptions, the algorithm does not provide an answer and for that we have the Fig. 1*

Example 4.2. *Consider* J *and* A *as defined in (4.1) and (4.2), respectively and assume that*

$$-A_{12}^T = A_{21},\ -A_{13}^T = A_{31},\ -A_{14}^T = A_{41}$$
$$-A_{23}^T = A_{32},\ -A_{24}^T = A_{42},\ -A_{34}^T = A_{43}.$$

Taking into account these assumptions, the matrix A *is not a* J*-symmetric matrix. Again, like the previous example, suppose that all* A_{ij} *in* A *are diagonal matrices for them the diagonal entries belong to* $(0, 10)$*. Except for* $(n/4) \times (n/4)$ *matrix* A_{14} *which if* $j - i < 9n/16 + 3$ *then its* (i, j)*st entry is a nonzero number in the interval* $(0, 10)$ *and the rest of its entries are*

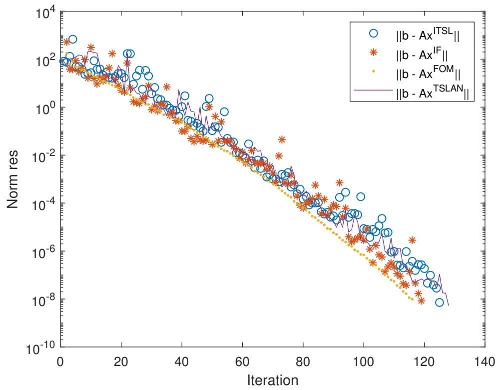

FIGURE 3. Comparison of methods TSL, FOM, ITSL and IFOM.

zero.
Consider the linear system $Ax = b$ in which A is the above matrix and b is a vector with random entries belonging to the interval $(0, 10)$. Take the following stop condition $\|x - x_m\| < \epsilon$ in which $\epsilon = 10^{-8}$ and x_m is the approximate solution of the m-th stage. With these assumptions, the algorithm will be able to calculate the solution of this non-J-symmetric linear system. The result can be seen in the Fig. 2

Example 4.3. *Consider $n \times n$ matrices J and A as follows*

$$J = diag(I_{n/4}, -I_{n/4}, I_{n/4}, -I_{n/4}),$$

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix},$$

and assume that

$$-A_{21}^T = A_{12},\ -A_{11}^T = A_{11},\ -A_{22}^T = A_{22}.$$

Taking into account these assumptions, the matrix A is not a symmetric and J-symmetric matrix. Suppose that A_{11} and A_{22} are tridiagonal matrices for them the entries belong to

TABLE 1. Comparison of methods TSL, FOM, ITSL and IFOM.

Method	ϵ	m	t(s)
TSL	8.7×10^{-9}	128	0.15
FOM	5.7×10^{-9}	116	0.14
ITSL	5.2×10^{-9}	125	0.20
IFOM	3.1×10^{-9}	119	0.34

interval $(0,1)$. *Except for* $(n/2) \times (n/2)$ *matrix* A_{12} *which if* $1 \le i \le n/2$ *then its* $(i, n/2 - i + 1)$*st entry is a nonzero number in the interval* $(0,1)$ *and the rest of its entries are zero. Consider the linear system* $Ax = b$ *in which* A *is the above matrix and* b *is a vector, with random entries belonging to the interval* $(0,1)$. *Take the following stop condition* $\|x - x_m\| < \epsilon$ *in which* $\epsilon = 10^{-8}$ *and* x_m *is the approximate solution of the* m*-th stage. By letting* $n = 200$ *The result can be seen in the Table 1 and Fig. 3*

5. CONCLUSION

Our goal in this paper is to obtain a two-sided Lanczos method in indefinite mode, which is effective for solving large non-J-symmetric linear systems. After obtaining this algorithm, according to numerical examples, it can be seen that this method is acceptable for non-J-symmetric matrices and its performance can be competitive for non-J-symmetric matrices with other methods.

REFERENCES

[1] Y. SAAD, *Iterative methods for sparse linear systems.* Industrial and Applied Mathematics, 3600 University City Science Center Philadelphia, PA. United States, 2003.

[2] Y. SAAD, *Numerical methods for large eigenvalueproblems*. Second edition, 2011.

[3] N.J. HIGHAM, *J-orthogonal matrices: properties and generations.* SIAM, Rev. 45(3):504-519, (2003).

[4] A. KIHÇMAN, Z.A. ZHOUR, *The representation and approximation for the weighted Minkowski inverse in Minkowski space.* Math. Comput. Modelling, 47(3-4):363-371,(2008).

[5] B.C. LEVY, *A note on the hyperbolic singular value decomposition.* Linear Algebra Appl., 277(1-3):135-142, (1998).

[6] R. ONN, A.O. STEINHARDT, A. BOJANCZYK, *The hyperbolic singular value decomposition and applications.* Applied Mathematics and Computing, Trans. 8th Army Conf., Ithaca-NY (USA)(1990), ARO Rep. 91-1, 93-108, (1991).

[7] V. SEGO, *The hyperbolic Schur decomposition, Linear Algebra Appl.* Linear Algebra Appl., 440 (2014), 90-110.

[8] 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.

[9] M. G. KAMALVAND AND K. N. ASIL, *Indefinite Ruhe's Variant of the Block Lanczos Method for Solving the Systems of Linear Equations*, Advances in Mathematical Physics, Volume 2020, Article ID 2439801, 9 pages.

[10] C. LANCZOS, *Solution of systems of linear equations by minimized iteration.* J. Res. Nat. Bureau Standards. 49 (1952), 33-53.

I.GOHBERG, P.LANCASTER, L.RODMAN, *Indefinite linear algebra and applications.* Birkhäuser, 2005.