Hindawi
Journal of Applied Mathematics
Volume 2020, Article ID 9874162, 8 pages
https://doi.org/10.1155/2020/9874162

Research Article

Some Hyperbolic Iterative Methods for Linear Systems

K. Niazi Asil and M. Ghasemi Kamalvand

Department of Mathematics, Lorestan University, Khorramabad, Iran

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

Received 18 October 2019; Revised 30 November 2019; Accepted 3 December 2019; Published 7 January 2020

Academic Editor: Qing-Wen Wang

Copyright © 2020 K. Niazi Asil and M. Ghasemi Kamalvand. 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.

The indefinite inner product defined by $J = \mathrm{diag}(j_1, \ldots, j_n)$, $j_k \in \{-1, +1\}$, arises frequently in some applications, such as the theory of relativity and the research of the polarized light. This indefinite scalar product is referred to as hyperbolic inner product. In this paper, we introduce three indefinite iterative methods: indefinite Arnoldi's method, indefinite Lanczos method (ILM), and indefinite full orthogonalization method (IFOM). The indefinite Arnoldi's method is introduced as a process that constructs a J-orthonormal basis for the nondegenerated Krylov subspace. The ILM method is introduced as a special case of the indefinite Arnoldi's method for J-Hermitian matrices. IFOM is mentioned as a process for solving linear systems of equations with J-Hermitian coefficient matrices. Finally, by providing numerical examples, the FOM, IFOM, and ILM processes have been compared with each other in terms of the required time for solving linear systems and also from the point of the number of iterations.

1. Introduction

Nowadays, iterative methods are used extensively for solving general large sparse linear systems in many areas of scientific computing because they are easier to implement efficiently on high-performance computers than direct methods.

Projection methods for solving systems of linear equations have been known for some time. The initial development was done by A. de la Garza [1].

One process by which an approximate solution $\tilde{x}$ of the linear system $Ax = b$ can be found is a projection method onto the subspace $\mathscr{K}$ and orthogonal to $\mathscr{L}$. This method focuses on this requirement that $\tilde{x}$ belongs to $\mathscr{K}$ and the new residual vector $b - A\tilde{x}$ be orthogonal to $\mathscr{L}$ (for more details, refer to [2–5]). Around the early 1950s, the idea of the Krylov subspace iteration was established by Cornelius Lanczos and Walter Arnoldi. Lanczos's method was based on two mutually orthogonal vector sequences and his motivation came from eigenvalue problems. In that context, the most prominent feature of the method is that it reduces the original matrix to tridiagonal form. Lanczos later applied his method to solve linear systems, in particular, symmetric ones. Krylov subspace iterations or Krylov subspace methods are iterative methods which are used as linear system solvers and also iterative solvers of eigenvalue problems. Krylov subspace methods which built up Krylov subspaces look for good approximations to eigenvectors. It is done by keeping all computed approximates and by combining them for a better solution.

In this paper, we introduce three iterative methods in the space with hyperbolic inner product. These methods are indefinite Arnoldi, indefinite Lanczos (ILM), and indefinite full orthogonalization (IFOM), and we define new algorithms to run these hyperbolic versions. By the numerical examples, we will compare these indefinite algorithms with their common definite modes, from the point of the number of iterations and the required time to run the algorithms.

This paper is organized as follows: in Section 2, the indefinite Arnoldi's process is proposed to construct a J-orthonormal basis. In Section 3, we present IFOM to solve the linear system of equations, and in Section 4, we give the ILM. At the end of this section, several numerical examples are expressed to compare the run speed and the number of repetitions. Counting the arithmetic act of multiplication in FOM, IFOM, and ILM algorithms and conclusion are the last two sections, respectively.

2. Indefinite Arnoldi's Method

We know that there are many applications which require a nonstandard scalar product which is usually defined by $[x, y]_J = y^* Jx$, where J is some nonsingular matrix and many of these applications consider Hermitian or skew-Hermitian J and an indefinite scalar product is defined by nonsingular Hermitian indefinite matrix $J \in \mathbb{C}^{n\times n}$ as $[x, y]_J = y^* Jx$. When J is the signature matrix $J = \mathrm{diag}(\pm 1) = \mathrm{diag}(j_{11}, j_{22}, \ldots, j_{nn})$ where $j_{kk} \in \{-1, 1\}$ for all k, the scalar product is referred to as hyperbolic and takes the following form:

$$[x, y]_J = y^* Jx = \sum_{i=1}^{n} j_{ii} x_i \overline{y_i}. \tag{1}$$

Example 1. Let $x = \langle x_1, \ldots, x_n\rangle^T$, $\langle y_1, \ldots, y_n\rangle^T \in \mathbb{C}^n$ and define

$$[x, y] = \sum_{i=1}^{r} x_{\sigma(i)} \overline{y}_{\sigma(i)} - \sum_{i=r+1}^{n} x_{\sigma(i)} \overline{y}_{\sigma(i)}, \tag{2}$$

where σ is a permutation for which $\sigma(i) = j_i$ and $j_i \in \{1, \ldots, n\}$ and r is an arbitrary integer from 0 to n. It is easy to see that $[.,.]$ is an indefinite inner product and its corresponding nonsingular Hermitian matrix is in the form $J = \mathrm{diag}(\pm 1)$, wherein r is the number of $+1$ and $n - r$ is the number of -1. Because if

$$[x, y] = (Jx, y), \tag{3}$$

then

$$\begin{aligned}[x, y] &= (Jx, y) = y^* Jx = \langle \overline{y}_1, \ldots, \overline{y}_n\rangle^T J \langle x_1, \ldots, x_n\rangle \\ &= \sum_{i=1}^{r} x_{\sigma(i)} \overline{y}_{\sigma(i)} - \sum_{i=r+1}^{n} x_{\sigma(i)} \overline{y}_{\sigma(i)},\end{aligned} \tag{4}$$

and conversely, if

$$[x, y] = \sum_{i=1}^{r} x_{\sigma(i)} \overline{y}_{\sigma(i)} - \sum_{i=r+1}^{n} x_{\sigma(i)} \overline{y}_{\sigma(i)}, \tag{5}$$

then it is clear that $[x, y] = (Jx, y)$, for all $x, y \in \mathbb{C}^n$.

Definition 1. If $\mathbb{M}$ is any nonzero subspace of $\mathbb{C}^n$, then the basis $x_1, \ldots, x_k$ for $\mathbb{M}$ is said to be an orthogonal basis with respect to the indefinite inner product $[.,.]$ 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 $i = 1, \ldots, k$. For the hyperbolic inner product, the above definitions of orthogonal basis and orthonormal basis are said to be J-orthogonal and J-orthonormal bases, respectively.

Let C^n be a vector space with an indefinite inner product $[.,.]$. A vector $y \in C^n$ is called nonneutral if $[y, y] \neq 0$. Note first of all that any set of nonneutral vectors $y_1, \ldots, y_m$ which is orthogonal in the sense of the indefinite inner product $[.,.]$ is necessarily linear independent. To see this, suppose that $\sum_{j=1}^{m} g_j y_j = 0$, and hence, for $k = 1, \ldots, m$,

$$\sum_{j=1}^{m} g_j [y_j, y_k] = g_k [y_k, y_k] = 0. \tag{6}$$

Then, it follows that $g_k = 0$.

In this section, we construct the indefinite Arnoldi's method and then we turn it into a practical algorithm.

Definition 2. Let a matrix A and a starting vector v be given. Then, the m-dimensional Krylov subspace $\mathscr{K}_m(A, V)$ is spanned by a sequence of m column vectors:

$$\mathscr{K}_m(A, V) \equiv \mathrm{span}\{v, Av, A^2 v, \ldots, A^{m-1} v\}. \tag{7}$$

It is well known that the construction of a basis with Arnoldi's method for the Krylov subspace $\mathscr{K}_m(A, V)$ leads to an upper Hessenberg matrix that describes the relation between the basis vectors. Each new basis vector can be constructed from the existing set as

$$h_{j+1,j} v_{j+1} = Av_j - \sum_{i=1}^{j} h_{i,j} v_i, \tag{8}$$

where $h_{i,j}$ follows from the orthogonality requirement $v_i^* v_{j+1} = 0$ and $h_{j+1,j}$ follows from the requirement that $\|v_{j+1}\|_2 = 1$. The expression (8) can be formulated in matrix notation as

$$AV_m = V_m H_m + w_m e_m^T, \tag{9}$$

where H_m is an upper Hessenberg matrix. For further details, refer to [6]. Now, the purpose of this section is to construct a J-orthonormal basis for the Krylov subspace (7) and the indefinite Arnoldi's process is an algorithm that brought us to this goal. Furthermore, this process is a tool that condensed the matrix into a Hessenberg form. The indefinite Arnoldi's algorithm for the computation of a J-orthonormal basis of the Krylov (7) subspace is shown in Algorithm 1.

Proposition 1. *Assume that the indefinite Arnoldi's algorithm does not stop before the m-th step. Then, the vectors $v_1, \ldots, v_m$ form a J-orthonormal basis of the Krylov subspace $\mathscr{K}_m(A, v)$.*

Proof. By considering the following expression, the proof is straightforward:

$$t(v_{j+1}) h_{j+1,j} v_{j+1} = Av_j - \sum_{i=1}^{j} t(v_i) h_{ij} v_i, \quad j = 1, \ldots, m. \tag{10}$$

□

Proposition 2. *Define*

(i) H_m, the $m \times m$ Hessenberg matrix whose nonzero entries h_{ij} are defined by indefinite Arnoldi's algorithm

(ii) V_m, the $n \times m$ matrix with column vectors $v_1, \ldots, v_m$

(iii) $J' = diag(t(v_1), \ldots, t(v_m))$

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

Algorithm 1: Indefinite Arnoldi's algorithm.

Then, the following relations are valid:

$$AV_m = V_m \acute{J} H_m + t(v_{m+1}) h_{m+1,m} v_{m+1} e_m^T, \tag{11}$$

$$V_m^* J A V_m = H_m. \tag{12}$$

In particular, if $m = n$, then

$$V_n^{-1} A V_n = \acute{J} H_n. \tag{13}$$

Proof. Indeed, in general, (11) is the matrix representation of (10):

$$\begin{aligned} AV_m &= V_m \acute{J} H_m + w_m e_m^T = V_m \acute{J} H_m + t(v_{m+1}) h_{m+1,m} v_{m+1} e_m^T \\ &= \left(V_m \acute{J} \,\middle|\, v_{m+1} \right) \begin{pmatrix} H_m \\ 0, \ldots, 0, t(v_{m+1}) h_{m+1,m} \end{pmatrix}. \end{aligned} \tag{14}$$

Now, to see (12), left-multiply relation (14) by $V_m^* J$. We earn

$$V_m^* J A V_m = V_m^* J V_m \acute{J} H_m + t(v_{m+1}) h_{m+1,m} V_m^* J v_{m+1} e_m^T. \tag{15}$$

On the other hand, given that the vectors $v_1, \ldots, v_m$ build a J-orthonormal basis, then,

(1) According to the definition of v_{m+1}, $v_{m+1} = 0$ or it is orthogonal to $v_1, \ldots, v_m$, i.e., $[v_{m+1}, v_i] = v_i^* J v_{m+1} = 0$, for $i = 1, \ldots, m$. Thus,

$$V_m^* J v_{m+1} = 0. \tag{16}$$

(2) We have

$$V_m^* J V_m = \left([v_i, v_j] \right)_{i,j} = \operatorname{diag}\left([v_1, v_1], \ldots, [v_m, v_m] \right) = \acute{J}. \tag{17}$$

In other words, $\acute{J} V_m^* J V_m = I$.

Therefore, relation (15) can be summarized as follows:

$$V_m^* J A V_m = H_m, \tag{18}$$

and by left-multiplying by $\acute{J}$, we have

$$\acute{J} V_m^* J A V_m = \acute{J} H_m. \tag{19}$$

Particularly, if $m = n$, relation (17) yields that $\acute{J} V_n^* J = V_n^{-1}$ and thereby,

$$V_n^{-1} A V_n = \acute{J} H_n. \tag{20}$$

It is noteworthy that these concepts are used in [7] to solve an eigenvalue problem. □

3. Indefinite Full Orthogonalization Method

The purpose of this section is to build an algorithm to solve the linear system:

$$Ax = b, \tag{21}$$

where A is an $n \times n$ complex matrix and the inner product of the space is hyperbolic. Let $\mathscr{K}$ and $\mathscr{L}$ be two m-dimensional subspace of $\mathbb{C}^n$. As mentioned in Section 1, one process by which an approximate solution $\tilde{x}$ of the linear system (21) can be found is a projection method onto the subspace $\mathscr{K}$ and orthogonal to $\mathscr{L}$. This method focuses on this requirement that $\tilde{x}$ belongs to $\mathscr{K}$ and the new residual vector $b - A\tilde{x}$ be orthogonal to $\mathscr{L}$. Suppose that $V_m = [v_1, \ldots, v_m]$ is an $n \times m$ matrix whose columns constitute a J-orthonormal basis of the nondegenerated subspace $\mathscr{K}$ and, similarly, $W_m = [w_1, \ldots, w_m]$ is an $n \times m$ matrix whose columns form a J-orthonormal basis of the nondegenerated subspace $\mathscr{L}$ and suppose that the approximate solution of (21) is as follows:

$$\tilde{x} = x_0 + V_m y, \tag{22}$$

where x_0 is an initial guess to the solution of (21). Then, for each $i = 1, \ldots, m$, we should have

$$b - A\tilde{x} [\perp] \mathscr{L}, \tag{23}$$

where $[\perp]$ indicates orthogonality under the hyperbolic product $[.,.]_J$. Thus,

$$\begin{gathered} b - A\tilde{x}[\perp]w_i, \quad \text{for } i = 1, \ldots, m, \\ [b - A\tilde{x}, w_i] = 0 \Longrightarrow [b - A(x_0 + V_m y), w_i] = 0, \end{gathered} \tag{24}$$

and by letting $r_0 = b - Ax_0$, we have

$$[r_0 - AV_m y, w_i] = 0 \Longrightarrow w_i^* J r_0 = w_i^* JAV_m y. \tag{25}$$

This leads to $W_m^* J r_0 = W_m^* JAV_m y$. Now, assuming that the $m \times m$ matrix $W_m^* JAV_m$ is nonsingular, then

$$y = (W_m^* JAV_m)^{-1} W_m^* J r_0, \tag{26}$$

and therewith, for approximate solution $\tilde{x}$, we earn

$$\tilde{x} = x_0 + V_m (W_m^* JAV_m)^{-1} W_m^* J r_0. \tag{27}$$

Now, if $V_m = W_m = \mathscr{K}_m(A, r_0)$, the indefinite full orthogonalization method (IFOM) is a process which seeks for an approximation solution $\tilde{x}$ from the subspace $x_0 + \mathscr{K}_m$ with the proviso that

$$b - A\tilde{x}[\perp]\mathscr{K}_m. \tag{28}$$

Thus, relation (26) changes to $y = (V_m^* JAV_m)^{-1} V_m^* J r_0$. And, by relation (12), it is equal to $y = H_m^{-1} V_m^* J r_0$. On the other hand, by defining $v_1 = (r_0/\beta)$ where, $\beta = \sqrt{|[r_0, r_0]|}$, we have $V_m^* J r_0 = V_m^* J \beta v_1 = t(v_1)\beta e_1$. Thereupon,

$$y = H_m^{-1} t(v_1) \beta e_1, \tag{29}$$

and finally,

$$\tilde{x} = x_0 + V_m H_m^{-1} t(v_1) \beta e_1. \tag{30}$$

Our explanations are summarized in Algorithm 2.

Proposition 3. *The residual vector of the approximate solution x_m calculated by the IFOM algorithm is such that*

$$b - Ax_m = -t(v_{m+1}) h_{m+1,m} e_m^T y_m v_{m+1}. \tag{31}$$

Proof. By using relations (11) and (29), we have the following relations:

$$\begin{aligned} b - Ax_m &= b - A(x_0 + V_m y_m) = r_0 - AV_m y_m \\ &= \beta v_1 - V_m \acute{J} H_m y_m - t(v_{m+1}) h_{m+1,m} e_m^T y_m v_{m+1} \\ &= \beta v_1 - V_m \acute{J} H_m (H_m)^{-1} t(v_1) \beta e_1 \\ &\quad - t(v_{m+1}) h_{m+1,m} e_m^T y_m v_{m+1} \\ &= \beta v_1 - \beta v_1 - t(v_{m+1}) h_{m+1,m} e_m^T y_m v_{m+1} \\ &= -t(v_{m+1}) h_{m+1,m} e_m^T y_m v_{m+1}. \end{aligned} \tag{32}$$

We expect that FOM performs better than IFOM in terms of the number of iterations and the required time to run because in the IFOM method, the product of entries of $\acute{J}$ in the entries of H is added to the calculations, when compared to the FOM method. The following example verifies this expectation. In this example, we have used n instead of m, i.e., we have assumed that n is the maximum number of iterations of the algorithm. But instead, we have considered this requirement that $\|A\hat{x} - b\| < 10^{-8}$. □

Example 2. Let A is a 150×150 tridiagonal matrix and an arbitrary diagonal matrix J is $J = \operatorname{diag}(j_1, \ldots, j_{150})$, $j_i \in \{+1, -1\}$ and suppose that b and v_0 are two vectors in $\mathbb{R}^{150}$ (see Figure 1). Suppose that the entries of these vectors and the nonzero entries of A have been randomly selected from zero to five. Then, the effects of the FOM and IFOM algorithms are as follows:

(i) The IFOM algorithm (Algorithm 2) after 149 iterations and within $t = 1.89$ seconds brings the linear system $Ax = b$ to the following condition:

$$\|A\hat{x} - b\| < 10^{-8}. \tag{33}$$

(ii) The FOM algorithm does the same with 149 replications and at $t = 0.6$ seconds.

Despite the superiority of the FOM on the IFOM, there is an important property of the IFOM algorithm that is shown in the next section.

4. Indefinite Lanczos Method

This section is devoted to the indefinite Lanczos method. As can be seen in the following, this method is expressed as a special case of the indefinite Arnoldi's method in the complex space for the special case when the matrix A is J-Hermitian.

Definition 3. A matrix A is said to be J-Hermitian (J-symmetric) when $A = JA^*J$ ($A = JA^TJ$) and we write $A = A^{[H]}$ ($A = A^{[T]}$).

Proposition 4. *Assume that indefinite Arnoldi's method is applied to a J-Hermitian matrix A. Then, the matrix H_m obtained from the process is tridiagonal and Hermitian.*

Proof. From the indefinite Arnoldi's method, we have relation (12):

$$V_m^* JAV_m = H_m. \tag{34}$$

Thus, $V_m^* A^* J V_m = H_m^*$. On the other hand, that A is J-Hermitian yields $JA = A^*J$. Thus, $H_m^* = H_m$.

Therefore, the resulting matrix H_m by the indefinite Arnoldi's algorithm (Algorithm 1) for J-Hermitian matrix A is an upper Hessenberg and Hermitian matrix. In other words, H_m is a Hermitian tridiagonal matrix. The resulting H_m matrix is shown by T_m, and the diagonal elements are denoted by $\alpha_j := h_{jj}$, and the off-diagonal elements are denoted by $\beta_j := h_{j-1,j} = h_{j,j-1}$,

(1) Compute $r_0 = b - Ax_0$, $\beta := \sqrt{|[r_0, r_0]|}$, and $v_1 := (r_0/\beta)$ and $t(v_1) = [v_1, v_1]$
(2) Define the $m \times m$ matrix $H_m = \{h_{ij}\}_{i,j=1,\dots,m}$; set $H_m = 0$
(3) For $j = 1, \dots, m$ Do:
(4) Compute $w_j := Av_j$
(5) For $i = 1, \dots, j$ Do:
(6) $h_{ij} = [w_j, v_i]$
(7) $w_j := w_j - t(v_i) h_{ij} v_i$
(8) EndDo
(9) $\alpha = \sqrt{|[w_j, w_j]|}$, compute $v_{j+1} = (w_j/\alpha)$
(10) $t(v_{j+1}) = [v_{j+1}, v_{j+1}]$
(11) Compute $h_{j+1,j} = t(v_{j+1})\alpha$. If $\alpha = 0$, set $m := j$ and Goto 12
(12) EndDo
(13) Compute $y_m = H_m^{-1} t(v_1) \beta e_1$ and $x_m = x_0 + V_m y_m$

Algorithm 2: IFOM algorithm.

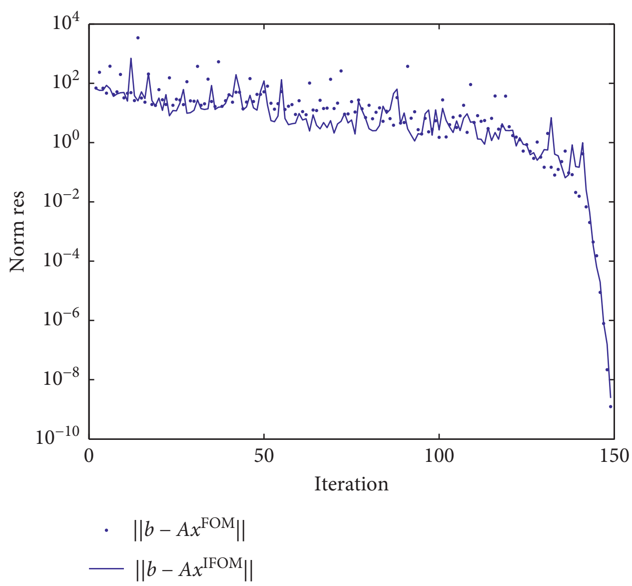

Figure 1

$$T_m = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \beta_m \\ & & & \beta_m & \alpha_m \end{pmatrix}. \tag{35}$$

In fact, we have the following:

$$w_j = Av_j - \sum_{i=1}^{j} t(v_i) h_{ij} v_i, \tag{36}$$

and the above relation turns to

$$w_j = Av_j - t(v_{j-1}) \beta_j v_{j-1} - t(v_j) \alpha_j v_j. \tag{37}$$

Thus,

$$\sqrt{|[w_j, w_j]|}\, v_{j+1} = Av_j - t(v_{j-1}) \beta_j v_{j-1} - t(v_j) \alpha_j v_j. \tag{38}$$

By using the hyperbolic inner product both sides in v_{j+1}, we earn

$$t(v_{j+1}) \sqrt{|[w_j, w_j]|} = [Av_j, v_{j+1}] \Longrightarrow \beta_{j+1} = t(v_{j+1}) \sqrt{|[w_j, w_j]|}. \tag{39}$$

This implies that $\beta_{j+1} = \overline{\beta_{j+1}}$; therefore, T_m is in the above form.

With this explanation, the hyperbolic version of the Hermitian Lanczos algorithm can be formulated as given in Algorithm 3.

Now, consider the linear system $Ax = b$ for which A is a J-Hermitian matrix and x_0 is an initial vector and the indefinite Lanczos vectors $v_1, \dots, v_m$ together with the tridiagonal matrix T_m are given. Then, the approximate solution obtained from an indefinite orthogonal projection method on to $\mathscr{K}_m$, similar to what was seen for the indefinite Arnoldi's method, is given by

$$x_m = x_0 + V_m y_m, \qquad y_m = T_m^{-1}(\beta e_1). \tag{40}$$

Thus, using the above algorithm, Algorithm 4 can be considered as the indefinite Lanczos algorithm for solving a real linear system with the J-Hermitian coefficient matrix.

Similar to what has already been proven for the IFOM algorithm, here also it can be seen that

$$b - Ax_m = -\beta_{m+1} e_m^T y_m v_{m+1}. \tag{41}$$

The advantage of ILM (the indefinite Lanczos method) is that it solves some classes of linear systems with different coefficient matrices, for different choices of matrix J. The following examples explain more in which we use n instead of m. In other words, we assume that the maximum number of the iterations of the algorithm is n. Besides, we use this stop condition that $\|A\widehat{x} - b\| < 10^{-8}$. □

Example 3. Consider the linear system $Ax = b$ in which

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \in M_n(\mathbb{R}), \tag{42}$$

(1) Choose an initial vector v_1 such that $t(v_1) = [v_1, v_1] = \pm 1$
(2) Set $\beta_1 \equiv 0$, $v_0 = 0$, $t(v_0) = 0$
(3) For $j = 1, \dots, m$; do:
(4) $w_j := Av_j - t(v_{j-1})\beta_j v_{j-1}$
(5) $\alpha_j := [w_j, v_j]$
(6) $w_j := w_j - t(v_j)\alpha_j v_j$
(7) $\alpha = \sqrt{|[w_j, w_j]|}$. If $\alpha = 0$, then stop
(8) $v_{j+1} := (w_j/\alpha)$
(9) $t(v_{j+1}) = [v_{j+1}, v_{j+1}]$
(10) $\beta_{j+1} := t(v_{j+1})\alpha$
(11) EndDo

Algorithm 3

(1) Compute $r_0 = b - Ax_0$, $\beta := \sqrt{|[r_0, r_0]|}$, and $v_1 := (r_0/\beta)$
(2) Set $\beta_1 \equiv 0$, $v_0 = 0$, $t(v_0) = 0$, $t(v_1) = [v_1, v_1]$
(3) For $j = 1, \dots, m$. Do
(4) $w_j := Av_j - t(v_{j-1})\beta_j v_{j-1}$
(5) $\alpha_j := [w_j, v_j]$
(6) $w_j := w_j - t(v_j)\alpha_j v_j$
(7) $\alpha = \sqrt{|[w_j, w_j]|}$. If $\alpha = 0$, then stop
(8) $v_{j+1} := (w_j/\alpha)$
(9) $t(v_{j+1}) = [v_{j+1}, v_{j+1}]$
(10) $\beta_{j+1} := t(v_{j+1})\alpha$
(11) EndDo
(12) Set $T_m = \text{tridiag}(\beta_i, \alpha_i, \beta_{i+1})$, and $V_m = [v_1, \dots, v_m]$.
(13) Compute $y_m = T_m^{-1} t(v_1)(\beta e_1)$, and $x_m = x_0 + V_m y_m$.

Algorithm 4

where each block A_{ij} is of order $(n/2) \times (n/2)$ and A_{11} and A_{22} are symmetric matrices and $A_{12}^T = -A_{21}$. By choosing $J = (-I_{n/2}) \oplus I_{n/2}$, it can be seen that $A = A^{[T]}$ (see Figure 2). Now, let $n = 200$ and suppose that A_{11} and A_{22} are diagonal matrices and A_{12} is a tridiagonal matrix where the diagonal entries of A_{11}, A_{22}, and A_{12} are chosen arbitrarily in $(0, 10)$ and b is a vector in $\mathbb{R}^{200}$ that its entries are selected at this distance (see Figure 3). The performance of the algorithm is as follows:

(i) The IFOM algorithm brings the linear system $Ax = b$ to the condition $\|A\hat{x} - b\| < 10^{-8}$, after 123 iterations and within $t = 1.7$ seconds
(ii) The FOM does the same with 118 iterations and within $t = 0.36$ seconds
(iii) The ILM algorithm does the same with 130 iterations and within $t = 0.15$ seconds

It shows that ILM is more efficient than the IFOM and FOM. However, the number of its iterations is higher but less time is required.

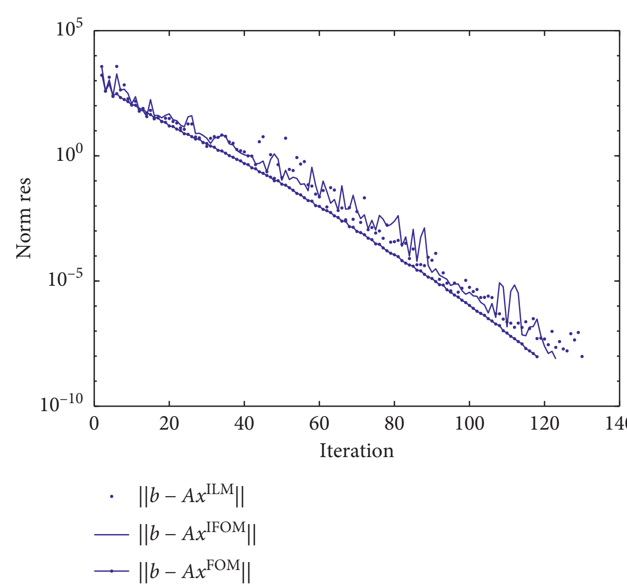

Figure 2

Example 4. Consider the assumptions of the previous example, except that $A \in M_n(\mathbb{C})$ and A_{12} is a tridiagonal matrix with complex entries as $a + ib$ in which $a \in (0, 10)$ and $b \in (1, 5)$ and $A_{12}^* = -A_{21}$. Then, $A = A^{[H]}$ and $n = 200$:

(i) By IFOM: the number of iterations is 140, within $t = 4.06$ seconds
(ii) By FOM: the number of iterations is 136, within $t = 1.11$ seconds

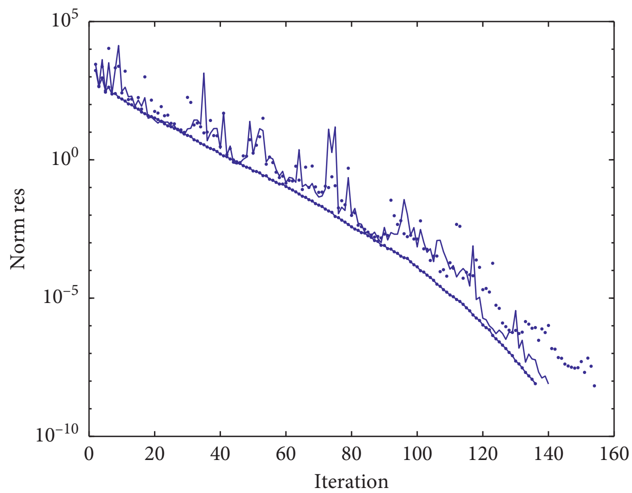

Figure 3

(iii) By ILM: the number of iterations is 154, within $t = 0.33$ seconds

Example 5. Consider the linear system $Ax = b$ in which

$$\begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} \in M_n(\mathbb{R}), \tag{43}$$

where each block A_{ij} is of order $(n/3) \times (n/3)$ and A_{11}, A_{22}, and A_{33} are symmetric matrices and $A_{12}^T = -A_{21}$ and $A_{23} = -A_{32}^T$ and $A_{13}^T = A_{31}$. By choosing $J = (-I_{n/3}) \oplus I_{n/3} \oplus (-I_{n/3})$, it can be seen that A is J-symmetric, and therefore, it is allowed to apply ILM. Now, suppose that

(i) A_{ii}, $i = 1, 2, 3$, are diagonal matrices with diagonal entries between zero and ten and $A_{13}^T = A_{31} = 0$ and $n = 300$

(ii) A_{12} and A_{23} are bidiagonal matrices and the entries of the main diagonals and their below entries are chosen between zero and ten

(iii) b is a vector in $\mathbb{R}^n$ with entries between zero and ten

(iv) x_0 is an initial vector in $\mathbb{R}^n$ with arbitrary entries between zero and ten

Then, to achieve the condition $\|A\hat{x} - b\| < 10^{-8}$, we need to consider the following:

(i) By IFOM: the number of iterations is 175, within $t = 5.6$ seconds

(ii) By FOM: the number of iterations is 167, within $t = 1.09$ seconds

(iii) By ILM: the number of iterations is 190, within $t = 0.39$ seconds

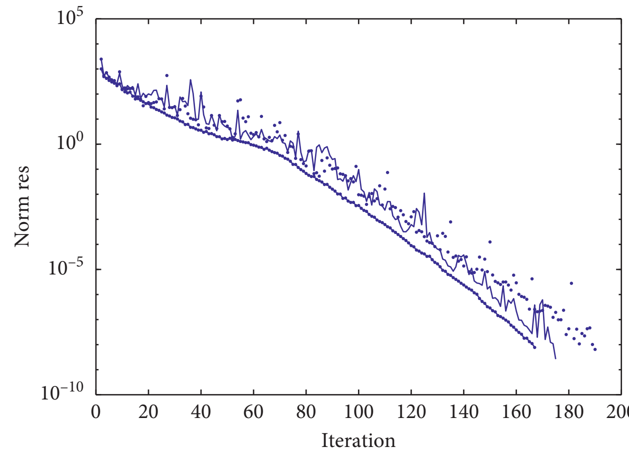

Figure 4

As it is seen, the ILM method is superior to FOM and IFOM methods. It is because of the low length of the recursive relation in its algorithm (tree terms for it) (see Figure 4). In other words, we do not need to do orthogonalizations at each step, on the all earlier vectors of its related Krylov subspace.

5. Counting the Arithmetic Act of Multiplication in FOM, IFOM, and ILM Algorithms

For two n-vectors x, y, we have

$$[x, y] = y^* J x = \sum_{i=1}^{n} j_{ii} \overline{y_i} x_i, \tag{44}$$

where x_i and y_i are the i-th elements of the x, y vectors, respectively, and j_{ii} is the (i, i)-th array of J. Thus, $2n$ multiplication operations are required to perform this indefinite inner product.

Using the above point, the number of multiplication operations required to perform steps (3)–(12) of the IFOM algorithm is equal to

$$mn^2 + \left(m^2 + 5m\right)n + m^2 + 2m, \tag{45}$$

and the number of multiplication operations required to perform steps (3)–(12) of the FOM algorithm is equal to

$$mn^2 + \left(\frac{m^2 + 5m}{2}\right)n + \frac{m^2}{2} + \frac{3}{2}m. \tag{46}$$

However, the number of required multiplication operations to do steps (3)–(11) of the ILM algorithm is

$$mn^2 + 6mn + 2m. \tag{47}$$

Comparison of (46) and (47) shows that, for $m \le 12$, the number of multiplications of the FOM algorithm is further than the ILM algorithm, and by increasing the value of m, this difference will also increase.

In the aforementioned algorithms, the inverses of the upper Hessenberg matrix H_n or the tridiagonal matrix T_n must be calculated. In this regard, it should be noted that calculating the inverse of the matrix T_m in the ILM algorithm requires less number of multiplications than calculating the inverse of the H_m matrix in the FOM algorithm.

What was said above shows that the run speed of the m steps of ILM is faster than that of the FOM algorithm and this is evident in the preceding section.

6. Conclusion

The indefinite inner product defined by $J = \mathrm{diag}(j_1, \ldots, j_n)$, $j_k \in \{-1, +1\}$, arises frequently in applications. It is used, for example, in the theory of relativity and in the research of the polarized light. More on the applications of such products can be found in [8–11]. These applications in other fields of science inspired us to resume Lanczos, FOM, and Arnoldi's methods in the indefinite inner product space. The indefinite Arnoldi's method is a process that constructs a J-orthonormal basis for the nondegenerated Krylov subspace, and the bases that we proved have a particular common property, about the structure of the product of their vectors. In this paper, IFOM process has been introduced and also a process is made which is useful for solving linear systems of equations with J-Hermitian coefficient matrices. This process is the same Lanczos method that has been restored in the hyperbolic inner product space. In this paper, the FOM, IFOM, and ILM processes have been compared with each other in terms of the time required for solving linear systems and the best one is introduced. Indeed, we show that the run speed of the m steps of ILM is faster than that of FOM and IFOM.

Data Availability

All data used to support the findings of this study are accessible and these data are cited at the relevant places within the text. The only exception is MATLAB codes for drawing the figures of the paper, which are also available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] A. de la Garza, "An iterative method for solving systems of linear equations," Report K-731, Union Carbide and Carbon Chemicals Company, K-25 Plant, Oak Ridge, TN, USA, 1951.

[2] C. Brezinski, *Projection Methods for Systems of Equations*, Elsevier, Amsterdam, Netherlands, 1997.

[3] M. Krasnosell'skii, G. Vainikko, P. Zabreiko, Y. Rutitskii, and V. Stetsenko, *Approximate Solution of Operator Equations*, Wolters-Nordhoff, Groningen, Netherlands, 1972.

[4] J. Liesen and Z. Strakos, *Krylov Subspace Methods*, Oxford Science Publications, Oxford, UK, 2013.

[5] R. Torenbeek and K. Vuik, "The Arnoldi and Lanczos methods for approximating the eigenpairs of a matrix," Report 96-44, University of Delft, Delft, Netherlands, 1995.

[6] W. E. Arnoldi, "The principle of minimized iterations in the solution of the matrix eigenvalue problem," *Quarterly of Applied Mathematics*, vol. 9, no. 1, pp. 17–29, 1951.

[7] 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.

[8] A. Kılıçman and Z. A. Zhour, "The representation and approximation for the weighted Minkowski inverse in Minkowski space," *Mathematical and Computer Modelling*, vol. 47, no. 3-4, pp. 363–371, 2008.

[9] B. C. Levy, "A note on the hyperbolic singular value decomposition," *Linear Algebra and Its Applications*, vol. 277, no. 1–3, pp. 135–142, 1998.

[10] R. Onn, A. O. Steinhardt, and A. Bojanczyk, "The hyperbolic singular value decomposition and applications," in *Proceedings of the 32nd Midwest Symposium on Circuits and Systems*, IEEE, Champaign, IL, USA, August 1989.

[11] N. J. Higham, "*J*-orthogonal matrices: properties and generations," *SIAM Review*, vol. 45, no. 3, pp. 504–519, 2003.


Advances in Operations Research
Advances in Decision Sciences
Journal of Applied Mathematics
The Scientific World Journal
Journal of Probability and Statistics
International Journal of Mathematics and Mathematical Sciences
Hindawi
Submit your manuscripts at
www.hindawi.com
Journal of Optimization
International Journal of Engineering Mathematics
International Journal of Analysis
Journal of Complex Analysis
Advances in Numerical Analysis
Mathematical Problems in Engineering
International Journal of Differential Equations
Discrete Dynamics in Nature and Society
International Journal of Stochastic Analysis
Journal of Mathematics
Journal of Function Spaces
Abstract and Applied Analysis
Advances in Mathematical Physics