Iterative solutions of the generalized Sylvester matrix equations by using the hierarchical identification principle

Feng Ding a,*, Peter X. Liu b, Jie Ding a

a Control Science and Engineering Research Center, Jiangnan University (Southern Yangtze University), Wuxi 214122, PR China
b Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada K1S 5B6

Abstract

In this paper, by extending the well-known Jacobi and Gauss–Seidel iterations for \( Ax = b \), we study iterative solutions of matrix equations \( AXB = F \) and generalized Sylvester matrix equations \( AXB + CXD = F \) (including the Sylvester equation \( AX + XB = F \) as a special case), and present a gradient based and a least-squares based iterative algorithms for the solution. It is proved that the iterative solution always converges to the exact solution for any initial values. The basic idea is to regard the unknown matrix \( X \) to be solved as the parameters of a system to be identified, and to obtain the iterative solutions by applying the hierarchical identification principle. Finally, we test the algorithms and show their effectiveness using a numerical example.

Keywords: Matrix equations; Gradient search principle; Jacobi iteration; Gauss–Seidel iteration; Hierarchical identification principle

1. Problem description

Consider a linear system described by a set of equation [1–5],

\[
Ax = b, \quad A \in \mathbb{R}^{m \times n}, \quad b \in \mathbb{R}^m. 
\]

Here, \( x \in \mathbb{R}^n \) is an unknown vector to be solved.

It is well known that for the nonsingular square matrix \( A (m = n) \), (1) has the solution \( x = A^{-1}b \); for the full column-rank nonsquare matrix \( A (m > n) \), we have the (least-squares) solution \( x = (A^T A)^{-1} A^T b \). These conventional solutions, however, involve the computation of matrix inversion. An alternative method is to find the iterative solution \( x(k) \) of \( x, k = 1, 2, 3, \ldots \).
Iterative algorithms are common in the areas of matrix algebra and system identification [1–19]. For example, the Jacobi and Gauss–Seidel methods for \( Ax = b \) are two typical iterative algorithms [1], and Mukaidani et al. discussed an iterative algorithm for generalized algebraic Lyapunov equations [19]. Recently, two gradient based iterative algorithms [2,4] and a least-squares based iterative algorithm [3] were given for general coupled matrix equations and general matrix equations. This paper focuses on gradient based and least-square based iterative solutions for the generalized Sylvester matrix equations [5,20,21], including the Sylvester matrix equation [22] as a special case.

For the full-rank square matrix \( A = [a_{ij}] \in \mathbb{R}^{n \times n} \) with nonzero diagonal elements \((a_{ii} \neq 0)\), the basic idea of the Jacobi iteration and Gauss–Seidel iteration is to split the matrix \( A \) into the sum of a strictly lower triangular submatrix \( L \), a diagonal submatrix \( D \) and a strictly upper triangular submatrix \( U \), i.e., \( A = L + D + U \), where

\[
L = \begin{bmatrix}
0 & 0 & \cdots & \cdots & 0 \\
a_{21} & 0 & \ddots & & \\
a_{31} & a_{32} & 0 & \ddots & \\
& \ddots & \ddots & \ddots & \\
a_{n1} & a_{n2} & \cdots & a_{n,n-1} & 0
\end{bmatrix} \in \mathbb{R}^{n \times n},
\]

\[
D = \text{diag}(a_{11}, a_{22}, \ldots, a_{nn}) \in \mathbb{R}^{n \times n}, \quad \text{and}
\]

\[
U = \begin{bmatrix}
0 & a_{12} & a_{13} & \cdots & a_{1n} \\
0 & 0 & a_{23} & \cdots & a_{2n} \\
& \ddots & \ddots & \ddots & \ddots \\
& & \ddots & \ddots & a_{n-1,n} \\
0 & \cdots & 0 & 0 & 0
\end{bmatrix} \in \mathbb{R}^{n \times n},
\]

and then to compute the iterative solutions \( x(k) \) of \( x \) for the given initial value \( x(0) \) [1]. The details are as follows:

For such a decomposition of \( A \), we have

\[(L + D + U)x = b.\]

If we move \((L + U)x\) to the right-hand side, it gives

\[Dx = -(L + U)x + b.\]

Substituting \( x \) on the left-hand side with \( x(k) \) and \( x \) on the right-hand side with \( x(k-1) \) gets

\[Dx(k) = -(L + U)x(k-1) + b, \quad k = 1, 2, 3, \ldots\]

This is the well-known Jacobi iteration [1].

Similarly, we have the Gauss–Seidel iteration [1],

\[(L + D)x(k) = -Ux(k-1) + b, \quad k = 1, 2, 3, \ldots\]

The main drawback of the Jacobi and Gauss–Seidel iterations is that they cannot guarantee that \( x(k) \) converges to the exact solution \( x = A^{-1}b \). This inspires us to study new iterative methods for matrix equations \( AXB = F \) and generalized Sylvester matrix equations \( AXB + CXD = F \) [20,21], for which the latter includes the Sylvester matrix equation \( AX + XB = F \) [22].

The rest of the paper is organized as follows. In Section 2, we study the iterative solution of \( Ax = b \) and \( AXB = F \) by extending the well-known Jacobi and Gauss–Seidel iterations. In Section 3, we derive iterative algorithms for generalized Sylvester matrix equations and study the convergence properties of the algorithms involved. Section 5 gives an example to illustrate the effectiveness of the algorithms proposed. Section 6 concludes the paper with some remarks.
2. The matrix equation $AXB = F$

In this section, we discuss more general iterative algorithms of $Ax = b$ and $AXB = F$.

Let $G \in \mathbb{R}^{n \times n}$ be a matrix to be determined and let $\mu > 0$ be the step size or the convergence factor. We present a large family of iterative methods for $Ax = b$ as follows [2,3]:

$$x(k) = x(k-1) + \mu G[b - Ax(k-1)].$$

(2)

This family is the extension of the Jacobi and Gauss–Seidel iterations because when $G = D^{-1}$ and $\mu = 1$, it boils down to the Jacobi method; when $G = (L + D)^{-1}$ and $\mu = 1$, we obtain the Gauss–Seidel method.

**Theorem 1.** For the iterative algorithm in (2), if System (1) has a unique solution $x$, then the iterative solution $x(k)$ given by (2) converges to the exact solution $x$ (i.e., $\lim_{k \to \infty} x(k) = x$) for any initial values $x(0)$ if

$$\mu (GA)^T(GA) < (GA)^T + (GA).$$

In fact, if $(GA)^T + (GA)$ is positive definite, a conservative choice of the convergence factor can be given by

$$0 < \mu \leq \frac{\lambda_{\min}[(GA)^T + (GA)]}{\lambda_{\max}[(GA)^T(GA)]}.$$

where $\lambda_{\max}[X]$ ($\lambda_{\min}[X]$) denotes the maximum (resp., minimum) eigenvalue of the square matrix $X$ [2].

From Theorem 1, we can also draw the following corollaries.

**Corollary 1.** If we take $G = A^T$, then the gradient based iterative algorithm [2,3],

$$\begin{align*}
\begin{cases}
x(k) = x(k-1) + \mu A^T[b - Ax(k-1)], \\
0 < \mu < \frac{2}{\max_{i}[A^T]}
\end{cases}
\end{align*}$$

yields $\lim_{k \to \infty} x(k) = x$. Here, $\|X\|^2 = \text{tr}[XX^T]$.

**Corollary 2.** If $A$ is a nonsquare $m \times n$ full column-rank matrix and we take $G = (A^T A)^{-1} A^T$, then the following least-squares-iterative algorithm leads to $\lim_{k \to \infty} x(k) = x$ [2,3]:

$$x(k) = x(k-1) + \mu (A^T A)^{-1} A^T[b - Ax(k-1)], \quad 0 < \mu < 2.$$

Next, we develop iterative algorithms for the matrix equations,

$$AXB = F,$$

(3)

where $A \in \mathbb{R}^{p \times m}, B \in \mathbb{R}^{q \times n}$ and $F \in \mathbb{R}^{p \times q}$ are the given constant matrices, and $X \in \mathbb{R}^{m \times n}$ is the unknown matrix to be solved.

**Lemma 1.** If $A$ is a full column-rank matrix and $B$ is a full row-rank matrix ($p \geq m, n \leq q$), then in the sense of least-squares, (3) has the unique solution:

$$X = (A^T A)^{-1} A^T F B^T (BB^T)^{-1}.$$

The corresponding homogeneous equation $AXB = 0$ has a unique solution $X = 0$.

According to Corollaries 1 and 2, it is easy to get the gradient based iterative algorithm and least-squares based iterative algorithms of (3), which are summarized as the following theorems.

**Theorem 2.** If the conditions of Lemma 1 hold, the gradient based iterative algorithm of (3),

$$X(k) = X(k-1) + \mu A^T[F - AX(k-1) B] B^T,$$

$$0 < \mu < \frac{2}{\lambda_{\max}[A^T A] \lambda_{\max}[B^T B]} \quad \text{or} \quad \mu \leq \frac{2}{\|A\|^2\|B\|^2},$$

yields $X(k) \to X$ [5].
Proof. Define the error matrix,
\[ \tilde{X}(k) := X(k) - X. \]

Using (3) and (4) gives
\[ \tilde{X}(k) = \tilde{X}(k - 1) + \mu A^T AX(k - 1)B^T = \tilde{X}(k - 1) - \mu A^T A\tilde{X}(k - 1)BB^T. \]

Using the formula \( \text{tr}[AB] = \text{tr}[BA] \) and \( \text{tr}[A^T] = \text{tr}[A] \), we have
\[ \|\tilde{X}(k)\|^2 = \text{tr}[\tilde{X}(k)^T\tilde{X}(k)] = \|\tilde{X}(k - 1)\|^2 - 2\mu \text{tr}\{A^T AX(k - 1)BB^T\} + \mu^2 \|A^T AX(k - 1)BB^T\|^2 \]
\[ = \|\tilde{X}(k - 1)\|^2 - 2\mu \|A\tilde{X}(k - 1)B\|^2 + \mu^2 \|A^T AX(k - 1)BB^T\|^2 \]
\[ \leq \|\tilde{X}(k - 1)\|^2 - 2\mu \|A\tilde{X}(k - 1)B\|^2 + \mu^2 \|A^T AX(k - 1)BB^T\|^2 \]
\[ = \|\tilde{X}(k - 1)\|^2 - \mu(2 - \mu \lambda_{\text{max}}[A^T] \lambda_{\text{max}}[B^T B]) \sum_{i=0}^{k-1} \|A\tilde{X}(i - 1)B\|^2. \]

Since
\[ 0 < \mu < \frac{2}{\lambda_{\text{max}}[A^T] \lambda_{\text{max}}[B^T B]}, \]
we have
\[ \sum_{k=1}^{\infty} \|A\tilde{X}(k)B\|^2 < \infty. \]

It follows that
\[ A\tilde{X}(k)B \to 0, \quad \text{as } k \to \infty, \]

According to Lemma 1, we have \( \tilde{X}(k) \to 0 \) as \( k \to \infty \).

Theorem 3. If the conditions of Lemma 1 hold, then the least-squares-iterative algorithm of (3),
\[ X(k) = X(k - 1) + \mu (A^TA)^{-1}A^T[F - AX(k - 1)B]B^T(BB^T)^{-1}, \quad 0 < \mu < 2, \]
yields \( X(k) \to X. \)

The proof of this theorem can be performed using a similar way to that of Theorem 2 and is therefore omitted here.

3. The matrix equation \( AXB + CXD = F \)

In this section, we derive the gradient based and least-squares based iterative solutions of the generalized Sylvester matrix equations,
\[ AXB + CXD = F, \]
where \( A, C \in \mathbb{R}^{m \times n}, B, D \in \mathbb{R}^{n \times n} \) and \( F \in \mathbb{R}^{m \times n} \) are the given constant matrices, and \( X \in \mathbb{R}^{m \times n} \) is the unknown matrix to be solved.

Let \( \lambda_i[M] \) be the \( i \)th eigenvalue of the square matrix \( M \). \( I_n \) is an \( n \times n \) identity matrix, and \( M \otimes N \) be the Kronecker product of \( M \) and \( N \). For an \( m \times n \) matrix
\[ X = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{m \times n}, \]
col[X] is an \( mn \) column vector consisting of the columns of \( X \), i.e.,

\[
\text{col}[X] = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}
\]

**Lemma 2.** (5) has a unique solution if and only if the matrix \( B^T \otimes A + D^T \otimes C \) is nonsingular. Then this solution is given by

\[
\text{col}[X] = [B^T \otimes A + D^T \otimes C]^{-1} \text{col}[F].
\]

The corresponding homogeneous equation \( AXB + CXD = 0 \) has a unique zero solution \( X = 0 \).

The hierarchical identification principle \([2–5,7,8]\) implies that by defining two matrices,

\[
F_1 := F - CBD, \quad \text{and} \quad F_2 := F - AXB.
\]

System (5) can be decomposed into two subsystems,

\[
S_1 : AXB = F_1, \quad \text{and} \quad S_2 : CXD = F_2.
\]

According to Theorem 2, it is easy to get the iterative solutions \( X_1(k) \) and \( X_2(k) \) of two subsystems \( S_1 \) and \( S_2 \) as follows:

\[
\begin{align*}
X_1(k) &= X_1(k-1) + \mu A^T[F_1 - AX_1(k-1)B]B^T, \\
X_2(k) &= X_2(k-1) + \mu C^T[F_2 - CX_2(k-1)D]D^T.
\end{align*}
\]

The above convergence factor \( \mu \) will be given later. Substituting (7) and (8) into the above two equations yields,

\[
\begin{align*}
X_1(k) &= X_1(k-1) + \mu A^T[F - AX_1(k-1)B - CXD]B^T, \\
X_2(k) &= X_2(k-1) + \mu C^T[F - AXB - CX_2(k-1)D]D^T.
\end{align*}
\]

Because the right-hand sides of the above two expressions contain the unknown matrix \( X \), it is impossible to realize this algorithm. Based on the hierarchical identification principle \([2–5,7,8]\), the unknown variable \( X \) is replaced with its estimate at time \((k-1)\). Hence, we have

\[
\begin{align*}
X_1(k) &= X_1(k-1) + \mu A^T[F - AX_1(k-1)B - CX_1(k-1)]B^T, \\
X_2(k) &= X_2(k-1) + \mu C^T[F - AXB - CX_2(k-1)]D^T.
\end{align*}
\]

In fact, we need only an iterative solution \( X(k) \) rather than two solutions \( X_1(k) \) and \( X_2(k) \). Taking the average of \( X_1(k) \) and \( X_2(k) \), we obtain a gradient-iterative (GI) algorithm \([5]\),

\[
X(k) = \frac{X_1(k) + X_2(k)}{2},
\]

\[
\begin{align*}
X_1(k) &= X(k-1) + \mu A^T[F - AX(k-1)B - CX(k-1)]B^T, \\
X_2(k) &= X(k-1) + \mu C^T[F - AX(k-1)B - CX(k-1)]D^T.
\end{align*}
\]

and a conservative choice of the convergence factor is given by

\[
0 < \mu < 2\left\{ \lambda_{\max}[AA^T]\lambda_{\max}[BB^T] + \lambda_{\max}[CC^T]\lambda_{\max}[DD^T] \right\}^{-1} =: \mu_0.
\]

(11–14) form the gradient-iterative algorithm of the generalized Sylvester matrix equation, the initial value may be taken to be \( X(0) = 0 \) or some small real matrix, i.e., \( X(0) = 10^{-6}I_{mn} \), where \( I_{mn} \) denotes an \( m \times n \) matrix with all elements being 1.
Theorem 4. If (5) has a unique (least-squares) solution \( X \), then the iterative solution \( X(k) \) given by the gradient-iterative algorithm (11)-(14) converges to \( X \) (i.e., \( X(k) \to X \)) for any initial value \( X(0) \).

Proof. Referring to [4], we define the error matrices,
\[
\tilde{X}_1(k) = X_1(k) - X, \\
\tilde{X}_2(k) = X_2(k) - X.
\]

Thus, we have
\[
\tilde{X}(k) = X(k) - X = \frac{\tilde{X}_1(k) + \tilde{X}_2(k)}{2}, \tag{15}
\]
\[
||\tilde{X}(k)||^2 = \frac{||\tilde{X}_1(k)||^2}{4} \leq \frac{||\tilde{X}_1(k)||^2 + ||\tilde{X}_2(k)||^2}{2}. \tag{16}
\]

Let
\[
\xi(k) := A\tilde{X}(k-1)B, \tag{17}
\]
\[
\eta(k) := C\tilde{X}(k-1)D. \tag{18}
\]

Using (5), (12) and (13), we have
\[
\tilde{X}_1(k) = \tilde{X}(k-1) + \mu A^T[-A\tilde{X}(k-1)B + C\tilde{X}(k-1)D]B^T = \tilde{X}(k-1) + \mu A^T[-\xi(k) - \eta(k)]B^T, \tag{19}
\]
\[
\tilde{X}_2(k) = \tilde{X}(k-1) + \mu C^T[-A\tilde{X}(k-1)B + C\tilde{X}(k-1)D]D^T = \tilde{X}(k-1) + \mu C^T[-\xi(k) - \eta(k)]D^T. \tag{20}
\]

Using the formula \( \text{tr}[AB] = \text{tr}[BA] \), \( \text{tr}[A^T] = \text{tr}[A] \), (17) and (18), we have
\[
||\tilde{X}_1(k)||^2 = \text{tr}[\tilde{X}_1^T(k)\tilde{X}_1(k)] ||\tilde{X}(k-1)||^2 + 2\mu \text{tr}[\tilde{X}_1^T(k-1)A^T[-\xi(k) - \eta(k)]B^T] + \mu^2 ||A^T[-\xi(k) - \eta(k)]B^T||^2
\]
\[
\leq ||\tilde{X}(k-1)||^2 + 2\mu \text{tr}[\xi^T(k)[-\xi(k) - \eta(k)]] + \mu^2 \lambda_{\text{max}}[AA^T] \lambda_{\text{max}}[BB^T] ||\xi(k) + \eta(k)||^2.
\]

Similarly,
\[
||\tilde{X}_2(k)||^2 = \text{tr}[\tilde{X}_2^T(k)\tilde{X}_2(k)]
\]
\[
\leq ||\tilde{X}(k-1)||^2 + 2\mu \text{tr}[\xi^T(k)[-\xi(k) - \eta(k)]] + \mu^2 \lambda_{\text{max}}[CC^T] \lambda_{\text{max}}[DD^T] ||\xi(k) + \eta(k)||^2.
\]

Substituting the above two equations into (16) yields
\[
||\tilde{X}(k)||^2 \leq ||\tilde{X}(k-1)||^2 - \mu ||\xi(k) + \eta(k)||^2 + \mu^2 \left( \frac{\lambda_{\text{max}}[AA^T] \lambda_{\text{max}}[BB^T]}{2} + \lambda_{\text{max}}[CC^T] \lambda_{\text{max}}[DD^T] \right) \]
\[
\times ||\xi(k) + \eta(k)||^2 ||\tilde{X}(k-1)||^2 - \mu \left( 1 - \frac{\mu}{\mu_0} \right) ||\xi(k) + \eta(k)||^2.
\]
\[
\leq ||\tilde{X}(0)||^2 - \mu \left( 1 - \frac{\mu}{\mu_0} \right) \sum_{i=1}^{k} ||\xi(i) + \eta(i)||^2.
\]

If the convergence factor \( \mu \) is chosen to satisfy
\[
0 < \mu < \mu_0 = \frac{2}{\lambda_{\text{max}}[AA^T] \lambda_{\text{max}}[BB^T] + \lambda_{\text{max}}[CC^T] \lambda_{\text{max}}[DD^T]},
\]
then we have
\[
\mu \left( 1 - \frac{\mu}{\mu_0} \right) \sum_{i=1}^{k} ||\xi(i) + \eta(i)||^2 \leq ||\tilde{X}(0)||^2 < \infty,
\]
or
\[
\sum_{k=1}^{\infty} ||\xi(k) + \eta(k)||^2 < \infty,
\]
It follows that
\[ \xi(k) + \eta(k) \to 0, \quad k \to \infty, \]
or
\[ A\tilde{X}(k - 1)B + C\tilde{X}(k - 1)D \to 0, \quad k \to \infty. \]
According to Lemma 2, we have \( \tilde{X}(k) \to 0 \), as \( k \to \infty. \)

The algorithm in (9) or (10) is known as the single-side iteration, which can not guarantee that \( X(k) \) converges to \( \tilde{X} \); and (11)–(14) give the balanced iterative algorithm, which can be simply written as
\[
X(k) = X(k - 1) + \mu A^T[F - AX(k - 1)B - CX(k - 1)D]B^T/2 + \mu C^T[F - AX(k - 1)B - CX(k - 1)D]D^T/2.
\]
\[ 0 < \mu < \frac{2}{\lambda_{\max}[A^T A] \lambda_{\max}[BB^T] + \lambda_{\max}[C^T C] \lambda_{\max}[DD^T]}.
\]

Next, we show that the convergence rate of the iterative algorithm in (11)–(14) depends on the condition number of the associated system, just like the Jacobi and Gauss–Seidel iterations for linear systems of the form \( Ax = b \) [1]. From (19), (20) and (15), we can get an error equation,
\[
\text{col}[\tilde{X}(k)] = [I_{mn} - \mu \Phi] \text{col}[\tilde{X}(k - 1)],
\]
\[
\Phi := (BB^T) \otimes (A^T A) + (BD^T) \otimes (A^T C) + (DB^T) \otimes (C^T A) + (DD^T) \otimes (C^T C).
\]
From here, the closer the eigenvalues of \( \mu \Phi \) are to 1, the closer the eigenvalues of \( I_{mn} - \mu \Phi \) tend to be zero, and hence, the faster the error \( \text{col}[,\tilde{X}(k)] \) or \( \tilde{X}(k) \) converges to zero. In other words, the gradient-iterative algorithm in (21), (22) has a fast convergence rate for small condition numbers of \( \Phi \).

Similarly, by means of Theorem 3 and the hierarchical identification principle, we can obtain the least-squares-iterative (LSI) algorithm of (5) as follows:
\[
X_1(k) = X(k - 1) + \mu (A^T A)^{-1} A^T[F - AX(k - 1)B - CX(k - 1)D]B^T(BB^T)^{-1},
\]
\[
X_2(k) = X(k - 1) + \mu (C^T C)^{-1} C^T[F - AX(k - 1)B - CX(k - 1)D]D^T(DD^T)^{-1},
\]
\[
X(k) = \frac{X_1(k) + X_2(k)}{2}, \quad 0 < \mu < 4.
\]

4. The Equation \( \sum_{j=1}^{p} A_j X B_j = F \)

In this section, we give the gradient-iterative and least-squares-iterative methods for a more general matrix equation [4],
\[
A_1 X B_1 + A_2 X B_2 + \cdots + A_p X B_p = F,
\]
where \( A_j \in \mathbb{R}^{m \times m} \), \( B_j \in \mathbb{R}^{n \times n} \) and \( F \in \mathbb{R}^{m \times n} \) are given constant matrices, \( X \in \mathbb{R}^{m \times n} \) is the unknown matrix to be solved.

**Lemma 3.** The equation in (26) has a unique solution if and only if the matrix \( \sum_{j=1}^{p} (B_j^T \otimes A_j) \) is nonsingular; in this case the solution is
\[
\text{col}[X] = \left[ \sum_{j=1}^{p} (B_j^T \otimes A_j) \right]^{-1} \text{col}[F];
\]
and if \( F = 0 \), the matrix equation in (26) has a unique solution \( X = 0 \) [4].

Similarly, we can compute the iterative solution \( X(k) \) of (26) by the following gradient-iterative algorithm [4]:
\[
X(k) = [X_1(k) + X_2(k) + \cdots + X_d(k)]/p,
\]
\[ 0 < \mu < 4. \]
\[ X(k) = X(k-1) + \mu A_i^T \left[ F - \sum_{j=1}^{p} A_j X(k-1) B_j \right] B_i, \]

\[ \frac{1}{\mu} = \sum_{j=1}^{p} \lambda_{\text{max}} [A_j A_j^T] \lambda_{\text{max}} [B_j B_j^T] \quad \text{or} \quad \frac{1}{\mu} = \sum_{j=1}^{p} \|A_j \|^2 \|B_j \|^2, \]

or by the following least-squares-iterative (LSI) algorithm:

\[ X(k) = X(k-1) + \mu \sum_{i=1}^{p} (A_i^T A_i)^{-1} A_i^T \left[ F - \sum_{j=1}^{p} A_j X(k-1) B_j \right] B_i (B_i B_i^T)^{-1}, \quad 0 < \mu < 2p. \]

**Theorem 5.** If the matrix equation in (26) has a unique solution \( X \), then the iterative solution \( X(k) \) given by the algorithm in (27)–(29) converges to the solution \( X \) [4].

Convergence of the LSI algorithm in (23)–(25) or LSI algorithm in (30) is also very difficult to prove and still requires studying further.

### 5. Examples

In this section, an example is given to illustrate the convergence properties of the proposed algorithms.

**Example.** Suppose that the generalized Sylvester matrix equations are \( AXB + CXD = F \) with

\[
A = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 2 & -1 \\ 1 & 2 \end{bmatrix},
\]

\[
D = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}, \quad F = \begin{bmatrix} -1 & -5 \\ 16 & 16 \end{bmatrix}.
\]

Then the solution of \( X \) from (6) is

\[
X = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & 5 \end{bmatrix}.
\]

Taking \( X(0) = 10^{-6} I_{2\times2} \), we apply the algorithm in (21), (22) to compute \( X(k) \). The iterative solution of \( X \) is shown in Table 1 with the relative error \( \delta := \|X(k) - X\|/\|X\| \), and the error \( \delta \) versus iteration \( k \) is shown in Fig. 1. From Table 1 and Fig. 1, it is clear that the error \( \delta \) is becoming smaller and approaches zero as iteration time \( k \) increases. This indicates that the proposed algorithm is effective and convergent.

<table>
<thead>
<tr>
<th>( k )</th>
<th>( x_{11} )</th>
<th>( x_{12} )</th>
<th>( x_{21} )</th>
<th>( x_{22} )</th>
<th>( \delta ) (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.42857</td>
<td>1.14286</td>
<td>2.42857</td>
<td>5.28571</td>
<td>28.57142051</td>
</tr>
<tr>
<td>2</td>
<td>0.91837</td>
<td>1.83673</td>
<td>2.75510</td>
<td>4.59184</td>
<td>8.16326300</td>
</tr>
<tr>
<td>3</td>
<td>1.11662</td>
<td>1.93003</td>
<td>2.95335</td>
<td>5.02332</td>
<td>2.33236086</td>
</tr>
<tr>
<td>4</td>
<td>0.99334</td>
<td>1.98667</td>
<td>2.98001</td>
<td>4.96668</td>
<td>0.66638882</td>
</tr>
<tr>
<td>5</td>
<td>1.00952</td>
<td>1.99429</td>
<td>2.99619</td>
<td>5.00190</td>
<td>0.19039680</td>
</tr>
<tr>
<td>6</td>
<td>0.99946</td>
<td>1.99891</td>
<td>2.99837</td>
<td>4.99728</td>
<td>0.05439909</td>
</tr>
<tr>
<td>7</td>
<td>1.00078</td>
<td>1.99953</td>
<td>2.99969</td>
<td>5.00016</td>
<td>0.01554260</td>
</tr>
<tr>
<td>8</td>
<td>0.99996</td>
<td>1.99991</td>
<td>2.99987</td>
<td>4.99978</td>
<td>0.00444074</td>
</tr>
<tr>
<td>9</td>
<td>1.00006</td>
<td>1.99996</td>
<td>2.99997</td>
<td>5.00001</td>
<td>0.00126878</td>
</tr>
<tr>
<td>10</td>
<td>1.00000</td>
<td>1.99999</td>
<td>2.99999</td>
<td>4.99998</td>
<td>0.00036251</td>
</tr>
<tr>
<td>Solution</td>
<td>1.00000</td>
<td>2.00000</td>
<td>3.00000</td>
<td>5.00000</td>
<td></td>
</tr>
</tbody>
</table>
6. Conclusions

This paper presents a set of gradient-iterative algorithms of solving the generalized Sylvester matrix equations by using the Jacobi method and hierarchical identification principle. The analysis indicates that the iterative solution obtained by the gradient-iterative algorithms converges to the exact solution for any initial value. Performances of the least-squares-iterative algorithms is still a future topic to be studied.

References


