Solving Two Coupled Fuzzy Sylvester Matrix Equations Using Iterative Least-squares Solutions

Ahmed M. E. Bayoumi and Mohamed A. Ramadan

ABSTRACT

In this paper, five iterative methods for solving two coupled fuzzy Sylvester matrix equations are considered. The two coupled fuzzy Sylvester matrix equations are expressed by using the generalized inverse of the coefficient matrix, then iterative solutions are constructed by applying the hierarchical identification principle and by using the block-matrix inner product (the star product for short). A proposed modification to this algorithm to solve the first coupled fuzzy Sylvester matrix equations is suggested. This proposed modification is compared with the first algorithm where our modification exhibits fast convergence behavior. Also, we suggested two least-squares iterative algorithm by applying a hierarchical identification principle to solve the two coupled fuzzy Sylvester matrix equations. The proposed methods are illustrated by numerical examples.

1. Introduction

Many authors attempt to solve coupled Sylvester matrix equations by various methods. Ding et al. [1] obtained the approximate solutions of the matrix equation $AXB = F$ and the generalized Sylvester matrix equation $AXB + CXD = F$, by extending Jacobi and Gauss–Seidel iteration methods for $Ax = b$. Ding and Chen [2] suggested a least-squares iterative algorithm to solve the generalized coupled Sylvester matrix equation

$$AX + YB = C, \quad DX + YE = F \quad (1)$$

In [3], a large family of iterative methods to solve coupled Sylvester matrix equations by applying a hierarchical identification principle is presented. Iterative algorithms for obtaining the unique least-squares solution were proposed in [2, 3] by introducing the block-matrix inner product.

Efficient numerical algorithms are presented with the gradient-based iterative algorithms [3, 4] and least square-based iterative algorithms [3] for solving coupled matrix equations. Hajarian [5] suggested a conjugate direction (CD) algorithm to find the generalized reflexive solution $X$ and the generalized anti-reflexive solution $Y$ of the coupled
Sylvester matrix equations

\[ AXB + CYD = F_1, \quad EXG + HYN = F_2. \]  

(2)

Zhang [6] constructed a gradient-based iterative algorithm to solve the real coupled matrix equations (2) by using the hierarchical identification principle. Bayoumi et al. [7] suggested a modified gradient based iterative algorithm for solving extended Sylvester-conjugate matrix equations \( AXB + C\bar{X}D = F \).

Friedman et al. [8] proposed a general model for solving an \( n \times n \) fuzzy linear system with a crisp coefficient and an arbitrary vector of fuzzy numbers on the right-hand side column. In [9], fuzzy numbers with a new parametric form are presented. And a new fuzzy arithmetic is defined and applied to fuzzy linear equations and fuzzy calculus. In [10], the common solution pair of fuzzy matrix equations is studied and the Kronecker product and Vec-operator for transforming the system of fuzzy linear matrix equation to a fuzzy linear system are employed. Bayoumi [11] proposed finite iterative Hamiltonian solutions of the generalized coupled Sylvester-conjugate matrix equations. Bayoumi and Ramadan [12] introduced finite iterative Hermitian R-conjugate solutions of the generalized coupled Sylvester-conjugate matrix equations. Behera and Chakraverty [13] proposed a new and simple method to solve fuzzy real system of linear equations with Crisp Coefficients. Wang et al. [14] investigated the least-squares solution with the least norm to a system of tensor equations over the quaternion algebra.

This paper is organized as follows: first, in Section 2, we introduce some notations, definitions, lemmas and theorems that will be needed to develop this work. In Section 3, we suggest five iterative algorithms to obtain the solutions of two coupled fuzzy Sylvester matrix equations. In first algorithm, we investigate the coupled fuzzy Sylvester matrix equations given in (1) by using the generalized inverse of the coefficient matrix, then iterative solutions are constructed by applying the hierarchical identification principle and by using the block-matrix inner product, and we propose a modification to this algorithm in the second algorithm for the same matrix equations. In third algorithm, we introduce least-squares iterative algorithm by applying a hierarchical identification principle to solve coupled fuzzy Sylvester matrix equations given in (1). In fourth algorithm, we investigate the coupled fuzzy Sylvester matrix equations given in (2) by using the generalized inverse of the coefficient matrix, then iterative solutions are constructed by applying the hierarchical identification principle and by using the block-matrix inner product. In fifth algorithm, we introduce least-squares iterative algorithm by applying a hierarchical identification principle to solve coupled fuzzy Sylvester matrix equations given in (2). And we give the convergence properties of these iterative algorithms. In Section4, numerical examples are introduced to illustrate the effectiveness of the proposed algorithms.

2. Preliminaries

The following notations, definitions, lemmas and theorems will be used to develop the proposed work. We use \( A^T \) to denote the transpose of \( A \). The set of all \( m \times n \) real matrices is denoted by \( \mathbb{R}^{m \times n} \). For \( A \in \mathbb{R}^{m \times n} \), \( \text{vec} (A) \) is defined as \( \text{vec} (A) = [a_1^T a_2^T \cdots a_n^T]^T \) where \( a_i \) is the \( i \)th column of the matrix \( A \). The Kronecker product of two matrices \( A = (a_{ij})_{m \times n} \) and \( B \) is denoted by \( A \otimes B \). We have the following well-known property \( \text{vec} (MXN) = (N^T \otimes M) \text{vec} (X) \) for matrices \( M, X, N \).
Definition 2.1: Block-matrix inner product [2]

The block-matrix inner product is called the star product for short, denoted by \( \ast \). Let

\[
X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{bmatrix} \in \mathbb{R}^{mp \times n},
Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_p \end{bmatrix} \in \mathbb{R}^{np \times m},
S_A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1p} \\ A_{21} & A_{22} & \cdots & A_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ A_{p1} & A_{p2} & \cdots & A_{pp} \end{bmatrix},
S_B = \begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1p} \\ B_{21} & B_{22} & \cdots & B_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ B_{p1} & B_{p2} & \cdots & B_{pp} \end{bmatrix}.
\]

Then the block-matrix star product is defined as

\[
X \ast Y = \begin{bmatrix} X_1 Y_1 \\ X_2 Y_2 \\ \vdots \\ X_p Y_p \end{bmatrix},
S_A \ast X = \begin{bmatrix} A_{11} X_1 & A_{12} X_2 & \cdots & A_{1p} X_p \\ A_{21} X_1 & A_{22} X_2 & \cdots & A_{2p} X_p \\ \vdots & \vdots & \ddots & \vdots \\ A_{p1} X_1 & A_{p2} X_2 & \cdots & A_{pp} X_p \end{bmatrix},
X \ast S_B = \begin{bmatrix} X_1 B_{11} & X_1 B_{12} & \cdots & X_1 B_{1p} \\ X_2 B_{21} & X_2 B_{22} & \cdots & X_2 B_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ X_p B_{p1} & X_p B_{p2} & \cdots & X_p B_{pp} \end{bmatrix},
S_A \ast S_B = \begin{bmatrix} A_{11} B_{11} & A_{12} B_{12} & \cdots & A_{1p} B_{1p} \\ A_{21} B_{21} & A_{22} B_{22} & \cdots & A_{2p} B_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ A_{p1} B_{p1} & A_{p2} B_{p2} & \cdots & A_{pp} B_{pp} \end{bmatrix}.
\]

The following basic concepts of fuzzy number arithmetic and fuzzy linear system of equations will be used to develop the proposed work.

Definition 2.2: Fuzzy number [8]

A fuzzy number in parametric form is an ordered pair of functions \( (u(r), \bar{u}(r)) \), \( 0 \leq r \leq 1 \), which satisfies the following requirements:

1) \( u(r) \) is a bounded left continuous non-decreasing function over \([0, 1]\),
2) \( \bar{u}(r) \) is a bounded right continuous non-increasing function over \([0, 1]\),
3) \( u(r) \leq \bar{u}(r) \), \( 0 \leq r \leq 1 \).

A crisp number \( \alpha \) is simply represented by \( u(r) = \bar{u}(r) = \alpha \), \( 0 \leq r \leq 1 \).

The triangular fuzzy numbers are very popular and denoted by \( u = (c, \alpha, \beta) \) and defined by

\[
u(x) = \begin{cases} 
\frac{x - c + \alpha}{\alpha} & c - \alpha \leq x \leq c, \\
\frac{c + \beta - x}{\beta} & c \leq x \leq c + \beta, \\
0 & \text{otherwise}.
\end{cases}
\]

where \( \alpha > 0 \) and \( \beta > 0 \). The parametric form of the number is \( u(r) = r\alpha + c - \alpha \), \( \bar{u}(r) = c + \beta - \beta r \).
The addition and scalar multiplication of fuzzy numbers are defined by the extension principle and can be equivalently represented as follows, see [8, 9].

For arbitrary fuzzy numbers \( v = (v(r), \overline{v}(r)) \) and \( w = (w(r), \overline{w}(r)) \), \( 0 \leq r \leq 1 \) and real number \( k \) as follows:

a) \( v = w \) if and only if \( v(r) = w(r) \) and \( \overline{v}(r) = \overline{w}(r) \),

b) \( v + w = (v(r) + w(r), \overline{v}(r) + \overline{w}(r)) \),

c) \( v - w = (v(r) - w(r), \overline{v}(r) - \overline{w}(r)) \),

d) \( kv = \begin{cases} (kv(r), k\overline{v}(r)) & k \geq 0, \\ (k\overline{v}(r), kv(r)) & k < 0. \end{cases} \)

**Definition 2.3:** Consider the \( p \times q \) linear system of equations

\[
\begin{align*}
\begin{cases}
 a_{11}v_1 + a_{12}v_2 + \cdots + a_{1q}v_q &= w_1, \\
 a_{21}v_1 + a_{22}v_2 + \cdots + a_{2q}v_q &= w_2, \\
 \vdots \\
 a_{p1}v_1 + a_{p2}v_2 + \cdots + a_{pq}v_q &= w_p,
\end{cases}
\end{align*}
\]

(3)

where the coefficient matrix \( A = (a_{ij}) \in \mathbb{R}^{p\times q} \) is given crisp matrix and \( w = (w_1, w_2, \ldots, w_p) \) is given vector of fuzzy numbers and \( v = (v_1, v_2, \ldots, v_q) \) is vector of fuzzy numbers to be determined. This system is called an FSLE.

**Definition 2.4:** A fuzzy number vector \( v = (v_1, v_2, \ldots, v_q) \) where \( v_i = (v_i(r), \overline{v}_i(r)), 0 \leq r \leq 1, i = 1, 2, \ldots, q, \) is called a solution of the fuzzy linear system of equations (3) if

\[
\begin{align*}
\begin{cases}
 \sum_{j=1}^{q} a_{ij}v_j = \sum_{j=1}^{q} a_{ij}v_j &= w_i, \\
 \sum_{j=1}^{q} a_{ij}v_j = \sum_{j=1}^{q} a_{ij}v_j &= \overline{w}_i
\end{cases}
\end{align*}
\]

(4)

In general, an arbitrary equation for either \( w_i \) or \( \overline{w}_i \) is a linear combination of \( v_j \)'s and \( \overline{v}_j \)'s, respectively. Therefore, in order to solve Equation (3) one must solve a \( 2p \times 2q \) crisp linear system of equations (5) as follows:

\[
Sv = w
\]

(5)

where

\[
S = \begin{bmatrix} S_1 & S_2 \\ S_2 & S_1 \end{bmatrix}, \quad v = \begin{bmatrix} v \\ \overline{v} \end{bmatrix}, \quad w = \begin{bmatrix} w \\ \overline{w} \end{bmatrix}
\]

(6)

where the element of \( S = (s_{ij}), 1 \leq i, j \leq 2q \), as follows:

\[
\begin{align*}
\text{if } a_{ij} \geq 0 \quad &\Rightarrow s_{ij} = a_{ij}, \quad s_{i+p,j+q} = a_{ij} \\
\text{if } a_{ij} < 0 \quad &\Rightarrow s_{i+j+q} = -a_{ij}, \quad s_{i+p,j} = -a_{ij}
\end{align*}
\]

(7)
**Theorem 2.1:** [10]: Let matrix \( S \) be in the form (6), then the matrix

\[
S^{(1,3)} = \frac{1}{2} \begin{bmatrix}
(S_1 + S_2)^{(1,3)} + (S_1 - S_2)^{(1,3)} & (S_1 + S_2)^{(1,3)} - (S_1 - S_2)^{(1,3)} \\
(S_1 + S_2)^{(1,3)} - (S_1 - S_2)^{(1,3)} & (S_1 + S_2)^{(1,3)} + (S_1 - S_2)^{(1,3)}
\end{bmatrix}
\]  

(8)
is a \( (1, 3) \)-inverse of the matrix \( S \), where \((S_1 + S_2)^{(1,3)}\) and \((S_1 - S_2)^{(1,3)}\) are \( (1, 3) \)-inverse of the matrices \((S_1 + S_2)\) and \((S_1 - S_2)\), respectively. In particular, the Moore–Penrose inverse of the matrix \( S \) is

\[
S^\dagger = \frac{1}{2} \begin{bmatrix}
(S_1 + S_2)^\dagger + (S_1 - S_2)^\dagger & (S_1 + S_2)^\dagger - (S_1 - S_2)^\dagger \\
(S_1 + S_2)^\dagger - (S_1 - S_2)^\dagger & (S_1 + S_2)^\dagger + (S_1 - S_2)^\dagger
\end{bmatrix}
\]

(9)

**Theorem 2.2:** [10]:

For the consistent system (5) and any \( (1, 3) \)-inverse \( S^{(1,3)} \) of the coefficient matrix \( S \), \( v = S^{(1,3)}w \) is a solution to the system (5).

**Lemma 2.1:** [15]

For matrix equation \( Ax = b \), if \( A \) is a full column-rank matrix, then the following least squares based iterative algorithm leads to \( \lim_{k \to \infty} x(k) = x \)

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

**Lemma 2.2:** [15]

For matrix equation \( AXB = F \), if \( A \) is a full column-rank matrix and \( B \) is a full row-rank matrix, then the iterative solution \( X(k) \) given by the following least squares based iterative algorithm converges to the exact solution \( X \) for any initial values \( X(0) \):

\[
X(k) = X(k - 1) + \mu (A^TA)^{-1}A^T [F - AX(k - 1)B]B^T (BB^T)^{-1} , \quad 0 < \mu < 2.
\]

**Lemma 2.3:** [2]

The coupled fuzzy Sylvester matrix equations given in (1),

\[
AX + YB = C, \quad DX + YE = F,
\]

where \( A, D \in \mathbb{R}^{m \times m} \) and \( B, E \in \mathbb{R}^{n \times n} \) are given crisp matrices and \( C, F \in \mathbb{R}^{m \times n} \) are given fuzzy matrices while \( X, Y \in \mathbb{R}^{m \times n} \) are fuzzy matrices to be determined. Equation (1) has a unique solution if and only if the matrix

\[
Q = \begin{bmatrix}
I_n \otimes A & B^T \otimes I_m \\
I_n \otimes D & E^T \otimes I_m
\end{bmatrix} \in \mathbb{R}^{2mn \times 2mn}
\]

is non-singular; in this case, the unique solution is given by

\[
\begin{bmatrix}
\text{vec}(X) \\
\text{vec}(Y)
\end{bmatrix} = Q^{-1} \begin{bmatrix}
\text{vec}(C) \\
\text{vec}(F)
\end{bmatrix}
\]

and the corresponding homogeneous matrix equation \( AX + YB = 0 \), \( DX + YE = 0 \) has a unique solution \( X = Y = 0 \).
Lemma 2.4: The coupled Sylvester matrix equations given in (2),

\[ AXB + CYD = F_1, \quad EXG + HYN = F_2. \]

where \( A, C, E, H \in \mathbb{R}^{m \times m} \) and \( B, D, G, N \in \mathbb{R}^{n \times l} \) are given crisp matrices and \( F_1, F_2 \in \mathbb{R}^{m \times l} \) are given fuzzy matrices while \( X, Y \in \mathbb{R}^{m \times n} \) are fuzzy matrices to be determined. Equation (2) has a unique solution if and only if the matrix

\[ Q = \begin{bmatrix} B^T \otimes A & D^T \otimes C \\ G^T \otimes E & N^T \otimes H \end{bmatrix} \in \mathbb{R}^{2lm \times 2nm} \]

is non-singular; in this case, the unique solution is given by

\[ \begin{bmatrix} \text{vec}(X) \\ \text{vec}(Y) \end{bmatrix} = Q^{-1} \begin{bmatrix} \text{vec}(F_1) \\ \text{vec}(F_2) \end{bmatrix} \]

and the corresponding homogeneous matrix equation \( AXB + CYD = 0, \quad EXG + HYN = 0 \) has a unique solution \( X = Y = 0 \).

3. The Main Results

In this section, we consider five iterative algorithms to solve two coupled fuzzy Sylvester matrix equations. Algorithm I and algorithm IV adopt the line of the one in [2].

3.1. Iterative Algorithm for Solving the Coupled Fuzzy Sylvester matrix equations (1)

In this section, we present an iterative least-squares algorithm for solving coupled fuzzy Sylvester matrix equations given in (1),

\[ AX + YB = C, \quad DX + YE = F \]

where \( A, D \in \mathbb{R}^{m \times m} \) and \( B, E \in \mathbb{R}^{n \times n} \) are given crisp matrices and \( C, F \in \mathbb{R}^{m \times n} \) are given fuzzy matrices while \( X, Y \in \mathbb{R}^{m \times n} \) are fuzzy matrices to be determined.

The basic idea is to regard Equation (1) as two matrices

\[ R_1 = \begin{bmatrix} C - YB \\ F - YE \end{bmatrix} \quad (10) \]

\[ R_2 = \begin{bmatrix} C - AX, \quad F - DX \end{bmatrix} \quad (11) \]

Hence, Equation (1) can be decomposed into two matrix equations of the form:

\[ K_1X = R_1 \quad (12) \]
\[ YK_2 = R_2 \quad (13) \]

Here \( K_1 = \begin{bmatrix} A \\ D \end{bmatrix} \) and \( K_2 = \begin{bmatrix} B, \quad E \end{bmatrix} \)

Then, we can define the following iterative formulas

\[ X(k) = X(k - 1) + \mu \left( K_1^T K_1 \right)^{-1} K_1^T (R_1 - K_1 X(k - 1)) \quad (14) \]
\[ Y(k) = Y(k - 1) + \mu (R_2 - Y(k - 1) K_2) K_2^T (K_2 K_2^T)^{-1} \]  

(15)

where \( \mu \) is the convergence factor. Substituting from Equations (10) and (11) into Equations (14) and (15) gives

\[
X(k) = X(k - 1) + \mu (K_1^T K_1)^{-1} \left[ A^T D + C - A X(k - 1) - Y B \right],
\]

(16)

\[
Y(k) = Y(k - 1) + \mu \left[ C - A X(k - 1) - Y(k - 1) B, \ F - D X(k - 1) - Y(k - 1) E \right] \left[ B, \ E \right]^T (K_2 K_2^T)^{-1} \]

(17)

The right-hand sides of these equations include the unknown fuzzy matrices \( X \) and \( Y \), so it is impossible to realize the algorithm in Equations (16) and (17). By applying the hierarchical identification principle [4], the unknown fuzzy matrices \( X \) and \( Y \) in these equations is replaced with its estimate \( \hat{X}(k) \) and \( \hat{Y}(k) \). Thus one has

\[
X(k) = X(k - 1) + \mu (K_1^T K_1)^{-1} \left[ A^T D + C - A X(k - 1) - \hat{Y}(k - 1) B \right],
\]

(18)

\[
Y(k) = Y(k - 1) + \mu \left[ C - A X(k - 1) - \hat{Y}(k - 1) B, \ F - D X(k - 1) - \hat{Y}(k - 1) E \right] \left[ B, \ E \right]^T (K_2 K_2^T)^{-1} \]

(19)

where

\[
\mu = \frac{2}{m + n} \quad \text{or} \quad \mu = \frac{2}{\lambda_{\max}(K_1 (K_1^T K_1)^{-1} K_1^T) + \lambda_{\max}(K_2^T K_2)^{-1} K_2}
\]

(20)

In this case, the iterative least-squares solutions of coupled fuzzy Sylvester matrix equations can be written as

\[
\hat{X}(k) = \hat{X}(k - 1) + \mu (K_1^T K_1)^{-1} \left[ A^T D + \overline{C} - A \overline{X}(k - 1) - \overline{Y}(k - 1) B \right],
\]

(21)

\[
\overline{X}(k) = \overline{X}(k - 1) + \mu (K_1^T K_1)^{-1} \left[ A^T D + \overline{C} - A \overline{X}(k - 1) - \overline{Y}(k - 1) B \right],
\]

(22)

\[
\overline{Y}(k) = \overline{Y}(k - 1) + \mu \left[ \overline{C} - A \overline{X}(k - 1) - \overline{Y}(k - 1) B, \ F - D \overline{X}(k - 1) - \overline{Y}(k - 1) E \right] \left[ B, \ E \right]^T (K_2 K_2^T)^{-1} \]

(23)

\[
\overline{Y}(k) = \overline{Y}(k - 1) + \mu \left[ \overline{C} - A \overline{X}(k - 1) - \overline{Y}(k - 1) B, \ F - D \overline{X}(k - 1) - \overline{Y}(k - 1) E \right] \left[ B, \ E \right]^T (K_2 K_2^T)^{-1} \]

(24)

where

\[
\mu = \frac{2}{m + n} \quad \text{or} \quad \mu = \frac{2}{\lambda_{\max}(K_1 (K_1^T K_1)^{-1} K_1^T) + \lambda_{\max}(K_2^T K_2)^{-1} K_2}
\]

We now outline our suggested algorithm.

**Algorithm I**

**Step 1** Input crisp matrices \( A, D \in \mathbb{R}^{m \times m} \) and \( B, E \in \mathbb{R}^{n \times n} \) and input fuzzy matrices \( C, F \in \mathbb{R}^{m \times n} \), and number \( \varepsilon \).
Step 2 Given any two initial fuzzy matrices $X(0), Y(0) \in \mathbb{R}^{m \times n}$.

Step 3 Compute $K_1 = \begin{bmatrix} A \\ D \end{bmatrix}$ and $K_2 = \begin{bmatrix} B, & E \end{bmatrix}$.

Step 4 For $k = 1, 2, \cdots$ until convergence

$$X(k) = X(k-1) + \mu (K_1^T K_1)^{-1} \begin{bmatrix} A^T \\ D \end{bmatrix} \left[ C - AX(k-1) - Y(k-1)B, \quad E - DX(k-1) - Y(k-1)E \right],$$

$$\bar{X}(k) = \bar{X}(k-1) + \mu (K_1^T K_1)^{-1} \begin{bmatrix} A^T \\ D \end{bmatrix} \left[ \bar{C} - A\bar{X}(k-1) - \bar{Y}(k-1)B, \quad \bar{E} - D\bar{X}(k-1) - \bar{Y}(k-1)E \right],$$

$$Y(k) = Y(k-1) + \mu \left[ C - AX(k-1) - Y(k-1)B, \quad E - DX(k-1) - Y(k-1)E \right]$$

$$\times \begin{bmatrix} B, & E \end{bmatrix}^T (K_2 K_2^T)^{-1}$$

$$\bar{Y}(k) = \bar{Y}(k-1) + \mu \left[ \bar{C} - A\bar{X}(k-1) - \bar{Y}(k-1)B, \quad \bar{E} - D\bar{X}(k-1) - \bar{Y}(k-1)E \right]$$

$$\times \begin{bmatrix} B, & E \end{bmatrix}^T (K_2 K_2^T)^{-1}$$

$$\mu = \frac{2}{m+n} \text{ or } \mu = \frac{2}{\lambda_{\max}(K_1(K_1^T K_1)^{-1}K_1^T) + \lambda_{\max}(K_2^T (K_2^T K_2)^{-1} K_2)}$$

Step 5 If $||X(k) - X(k-1)||/||X(k)|| < \varepsilon$, $||\bar{X}(k) - \bar{X}(k-1)||/||\bar{X}(k)|| < \varepsilon$, $||Y(k) - Y(k-1)||/||Y(k)|| < \varepsilon$ and $||\bar{Y}(k) - \bar{Y}(k-1)||/||\bar{Y}(k)|| < \varepsilon$ stop; otherwise go to step 6.

Step 6 Set $k = k + 1$ and return to step 4.

Step 7 End.

Theorem 3.1: If the coupled fuzzy Sylvester matrix equations (1) is consistent and has a unique fuzzy solutions $X^* = (X^*, \bar{X}^*) \in \mathbb{R}^{m \times n}$ and $Y^* = (Y^*, \bar{Y}^*) \in \mathbb{R}^{m \times n}$ and

$$\mu = \frac{2}{m+n} \text{ or } \mu = \frac{2}{\lambda_{\max}(K_1(K_1^T K_1)^{-1}K_1^T) + \lambda_{\max}(K_2^T (K_2^T K_2)^{-1} K_2)}$$

then the iterative sequence $\{X(k), \bar{X}(k), \{Y(k), \bar{Y}(k)\}$ generated by algorithm I converges to $X^*, \bar{X}^*, Y^*$ and $\bar{Y}^*$, that is, $\lim_{k \to \infty} X(k) = X^*$, $\lim_{k \to \infty} \bar{X}(k) = \bar{X}^*$, $\lim_{k \to \infty} Y(k) = Y^*$ and $\lim_{k \to \infty} \bar{Y}(k) = \bar{Y}^*$ for any initial fuzzy matrices $X(0), \bar{X}(0), Y(0)$ and $\bar{Y}(0)$.

Proof: First, we define the estimation error matrices as

$$\xi_1(k) = X(k) - X^*, \quad \xi_2(k) = Y(k) - Y^*, \quad \xi_3(k) = \bar{X}(k) - \bar{X}^* \quad \text{and} \quad \xi_4(k) = \bar{Y}(k) - \bar{Y}^* \quad \text{for} \quad k = 1, 2, \cdots.$$ 

Using algorithm I and the above error matrices, we can obtain

$$\xi_1(k) = \xi_1(k-1) - \mu (K_1^T K_1)^{-1} \begin{bmatrix} A^T \\ D \end{bmatrix} \left[ A\xi_1(k-1) + \xi_2(k-1)B \right]$$

$$\xi_2(k) = \xi_2(k-1) - \mu \left[ A\xi_1(k-1) + \xi_2(k-1)B, \quad D\xi_1(k-1) + \xi_2(k-1)E \right]$$

$$\times \begin{bmatrix} B, & E \end{bmatrix}^T (K_2 K_2^T)^{-1}$$

(25) and

(26) for $k = 1, 2, \cdots$. 

\[ \begin{bmatrix} A \xi_1(k-1) + \xi_2(k-1)B \\ D \xi_1(k-1) + \xi_2(k-1)E \end{bmatrix} = \begin{bmatrix} A \xi_1(k-1) + \xi_2(k-1)B \\ D \xi_1(k-1) + \xi_2(k-1)E \end{bmatrix} \]
Now, by taking the norm of (25) and (26) and using the following formula, we have

\[ \| K_1 [X + (K_1^T K_1)^{-1} \xi_1] \|^2 = \text{tr} \{ [X + (K_1^T K_1)^{-1} \xi_1]^T (K_1^T K_1) [X + (K_1^T K_1)^{-1} \xi_1] \} \]
\[ = \text{tr} \{ X^T (K_1^T K_1) X + 2 X^T \xi_1 + \xi_1^T (K_1^T K_1)^{-1} \xi_1 \} \]
\[ = \| K_1 X \|^2 + 2 \text{tr} [X^T \xi_1] + \| (K_1^T K_1)^{-1} \xi_1 \|^2. \] (27)

Gives

\[ \| K_1 \xi_1(k) \|^2 = \text{tr} \{ \xi_1^T(k) K_1^T K_1 \xi_1(k) \} \]
\[ = \text{tr} \{ \xi_1^T(k-1) K_1^T K_1 \xi_1(k-1) \} \]
\[ - 2 \mu \text{tr} \left\{ \xi_1^T(k-1) \begin{bmatrix} A & B \\ D & E \end{bmatrix}^T \begin{bmatrix} A \xi_1(k-1) + \xi_2(k-1)B \\ D \xi_1(k-1) + \xi_2(k-1)E \end{bmatrix} \right\} \]
\[ + \mu^2 \left\{ \| (K_1^T K_1)^{-1} \xi_1(k) \|^2 \right\} \]
\[ \leq \| K_1 \xi_1(k-1) \|^2 - 2 \mu \text{tr} \{ [A \xi_1(k-1)]^T (A \xi_1(k-1) + \xi_2(k-1)B) \]
\[ + [D \xi_1(k-1)]^T (D \xi_1(k-1) + \xi_2(k-1)E) \}
\[ + \mu^2 \| [A \xi_1(k-1) + \xi_2(k-1)B, D \xi_1(k-1) + \xi_2(k-1)E]^T \|^2 \} \] (28)

Similarly,

\[ \| \xi_2(k) K_2 \|^2 = \text{tr} \{ \xi_2^T(k) K_2^T \xi_2^T(k) \} \]
\[ = \text{tr} \{ \xi_2^T(k-1) K_2^T \xi_2^T(k-1) \} \]
\[ - 2 \mu \text{tr} \left\{ \begin{bmatrix} A \xi_1(k-1) + \xi_2(k-1)B, D \xi_1(k-1) + \xi_2(k-1)E \end{bmatrix} \right\} \]
\[ \times \begin{bmatrix} B, E \end{bmatrix}^T \xi_2^T(k-1) \}
\[ + \mu^2 \left\{ \| [A \xi_1(k-1) + \xi_2(k-1)B, D \xi_1(k-1) + \xi_2(k-1)E]^T \|^2 \right\} \]
\[ \leq \| \xi_2(k-1) K_2 \|^2 - 2 \mu \text{tr} \{ [\xi_2(k-1)B]^T (A \xi_1(k-1) + \xi_2(k-1)B) \}
\[ + \xi_2(k-1)B + [\xi_2(k-1)E]^T (D \xi_1(k-1) + \xi_2(k-1)E) \}
\[ + \mu^2 \| [A \xi_1(k-1) + \xi_2(k-1)B, D \xi_1(k-1) + \xi_2(k-1)E]^T \|^2 \} \] (29)

Define the nonnegative definite function \( \eta(k) \) by

\[ \eta(k) = \| K_1 \xi_1(k) \|^2 + \| \xi_2(k) K_2 \|^2. \]
From (28) and (29), this function can be computed as

\[
\eta(k) \leq ||K_1 \xi_1(k-1)||^2 - 2 \mu \text{tr}[(A\xi_1(k-1))T(A\xi_1(k-1) + \xi_2(k-1)B)
+ [D\xi_1(k-1)]T(D\xi_1(k-1) + \xi_2(k-1)E)]
+ \mu^2 m [||A\xi_1(k-1) + \xi_2(k-1)B||^2 + ||D\xi_1(k-1) + \xi_2(k-1)E||^2]
+ ||\xi_2(k-1)D_2||^2 - 2 \mu \text{tr}[(\xi_2(k-1)B)T(A\xi_1(k-1) + \xi_2(k-1)B)
+ [\xi_2(k-1)E]T(D\xi_1(k-1) + \xi_2(k-1)E)]
+ \mu^2 n [||A\xi_1(k-1) + \xi_2(k-1)B||^2 + ||D\xi_1(k-1) + \xi_2(k-1)E||^2]
\leq ||K_1 \xi_1(k-1)||^2
+ ||\xi_2(k-1)D_2||^2 - 2 \mu \text{tr}[(A\xi_1(k-1) + \xi_2(k-1)B)T(A\xi_1(k-1) + \xi_2(k-1)B)
+(D\xi_1(k-1) + \xi_2(k-1)E)T(D\xi_1(k-1) + \xi_2(k-1)E)]
+ \mu^2 (m + n) [||A\xi_1(k-1) + \xi_2(k-1)B||^2 + ||D\xi_1(k-1) + \xi_2(k-1)E||^2]
\leq \eta(k-1) - 2 \mu [||A\xi_1(i-1) + \xi_2(i-1)B||^2 + ||D\xi_1(i-1) + \xi_2(i-1)E||^2]
+ ||D\xi_1(k-1) + \xi_2(k-1)E||^2]
\]

\[
\eta(k) \leq \eta(1) - \mu [2 - \mu (m + n)] \sum_{i=1}^{k-1} [||A\xi_1(i) + \xi_2(i)B||^2 + ||D\xi_1(i) + \xi_2(i)E||^2].
\]

If the convergence factor \( \mu \) is chosen to satisfy

\[
0 < \mu < \frac{2}{m + n}.
\]

Then

\[
\sum_{k=1}^{\infty} [||A\xi_1(k) + \xi_2(k)B||^2 + ||D\xi_1(k) + \xi_2(k)E||^2] < \infty
\]

Since the matrix equation (1) has a unique fuzzy solution it follows that as \( k \to \infty \)

\[
\lim_{k \to \infty} A\xi_1(k) + \xi_2(k)B = 0 \quad \text{and} \quad \lim_{k \to \infty} D\xi_1(k) + \xi_2(k)E = 0.
\]

According to lemma 2.3, we have

\[
\lim_{k \to \infty} \xi_1(k) = 0 \quad \text{and} \quad \lim_{k \to \infty} \xi_2(k) = 0.
\]

Or

\[
\lim_{k \to \infty} X(k) = X^* \quad \text{and} \quad \lim_{k \to \infty} Y(k) = Y^*.
\]
3.2. A Modified Iterative Algorithm to Solve the Coupled Fuzzy Sylvester Matrix Equations (1)

In this subsection, we propose a modification to algorithm I to solve coupled fuzzy Sylvester matrix equations given in (1),

\[ AX + YB = C, \quad DX + YE = F. \]

where \( A, D \in \mathbb{R}^{m \times m} \) and \( B, E \in \mathbb{R}^{n \times n} \) are given crisp matrices and \( C, F \in \mathbb{R}^{m \times n} \) are given fuzzy matrices while \( X, Y \in \mathbb{R}^{m \times n} \) are fuzzy matrices to be determined. The proposed algorithm is as follows:

**Algorithm 1:**

**Step 1** Input crisp matrices \( A, D \in \mathbb{R}^{m \times m} \) and \( B, E \in \mathbb{R}^{n \times n} \) and input fuzzy matrices \( C, F \in \mathbb{R}^{m \times n} \), and number \( \varepsilon \).

**Step 2** Given any two initial fuzzy matrices \( X(0), Y(0) \in \mathbb{R}^{m \times n} \).

**Step 3** Compute \( K_1 = \begin{bmatrix} A & \varepsilon \\ D \end{bmatrix} \) and \( K_2 = [B, E] \).

**Step 4** For \( k = 1, 2, \ldots \) until convergence

\[
\begin{align*}
X(k) &= X(k-1) + \mu (K_1^T K_1)^{-1} \begin{bmatrix} A & \varepsilon \\ D \end{bmatrix}^T \begin{bmatrix} C - A\bar{X}(k-1) - \bar{Y}(k-1)B & F - D\bar{X}(k-1) - \bar{Y}(k-1)E \\ F - D\bar{X}(k-1) - \bar{Y}(k-1)E & \varepsilon \end{bmatrix}, \\
Y(k) &= Y(k-1) + \mu \begin{bmatrix} C - A\bar{X}(k) - \bar{Y}(k-1)B, & F - D\bar{X}(k) - \bar{Y}(k-1)E \\ F - D\bar{X}(k) - \bar{Y}(k-1)E & \varepsilon \end{bmatrix}, \\
\bar{X}(k) &= \bar{X}(k-1) + \mu (K_1^T K_1)^{-1} \begin{bmatrix} A & \varepsilon \\ D \end{bmatrix}^T \begin{bmatrix} C - A\bar{X}(k-1) - \bar{Y}(k-1)B & F - D\bar{X}(k-1) - \bar{Y}(k-1)E \\ F - D\bar{X}(k-1) - \bar{Y}(k-1)E & \varepsilon \end{bmatrix}, \\
\bar{Y}(k) &= \bar{Y}(k-1) + \mu \begin{bmatrix} C - A\bar{X}(k) - \bar{Y}(k-1)B, & F - D\bar{X}(k) - \bar{Y}(k-1)E \\ F - D\bar{X}(k) - \bar{Y}(k-1)E & \varepsilon \end{bmatrix}, \\
\mu &= \frac{2}{m + n} \text{ or } \mu = \frac{2}{\lambda_{\max}(K_1^T K_1)^{-1} K_1^T K_1 + \lambda_{\max}(K_2^T K_2)^{-1} K_2}.
\end{align*}
\]

**Step 5** If \( ||X(k) - X(k-1)||_{||X(k)||} < \varepsilon, ||X(k) - \bar{X}(k-1)||_{||X(k)||} < \varepsilon, \)

\( ||Y(k) - Y(k-1)||_{||Y(k)||} < \varepsilon \) and \( ||\bar{Y}(k) - \bar{Y}(k-1)||_{||\bar{Y}(k)||} < \varepsilon \) stop; otherwise go to step 6.

**Step 6** Set \( k = k + 1 \) and return to step 4.

**Step 7** End

Note that in the step of computing \( \bar{Y}(k) \), the last approximate solution \( \bar{X}(k) \) has been computed. Hence, we can use the information of \( \bar{X}(k) \) to update the \( Y(k) \). Similarly, in the step of computing \( \bar{Y}(k) \), the last approximate solution \( \bar{X}(k) \) has been computed. Hence, we can use the information of \( \bar{X}(k) \) to update the \( \bar{Y}(k) \).
3.3. Least Squares Based Iterative Solutions of Coupled Fuzzy Sylvester Matrix Equations (1)

In this section, we are studying the least squares based iterative solutions of coupled fuzzy Sylvester matrix equations (1) which can be written as

$$\begin{bmatrix} I_n \otimes A & B^T \otimes I_m \\ I_n \otimes D & E^T \otimes I_m \end{bmatrix} \begin{bmatrix} \text{vec}(X) \\ \text{vec}(Y) \end{bmatrix} = \begin{bmatrix} \text{vec}(C) \\ \text{vec}(F) \end{bmatrix}$$

If $A^+, B^+, D^+, E^+$ contain the positive entries of $A, B, D, E$, respectively, and $A^-, B^-, D^-, E^-$ contain the absolute value of negative entries of $A, B, D, E$, respectively, it is obvious that $A = A^+ - A^-, B = B^+ - B^-, D = D^+ - D^-, E = E^+ - E^-$. So, according to the properties of Kronecker operators it can be written as

$$I_n \otimes A = I_n \otimes (A^+ - A^-) = I_n \otimes A^+ - I_n \otimes A^-,$$

and $B^T \otimes I_m = (B^+ - B^-)^T \otimes I_m = B^{+T} \otimes I_m - B^{-T} \otimes I_m$.

Similarly

$$I_n \otimes D = I_n \otimes D^+ - I_n \otimes D^-,$$

and

$$E^T \otimes I_m = E^{+T} \otimes I_m - E^{-T} \otimes I_m.$$

$$Q = \begin{bmatrix} I_n \otimes A & B^T \otimes I_m \\ I_n \otimes D & E^T \otimes I_m \end{bmatrix}$$

$$Q = \begin{bmatrix} I_n \otimes A^+ - I_n \otimes A^- \\ I_n \otimes D^+ - I_n \otimes D^- \end{bmatrix} - \begin{bmatrix} \frac{B^{+T} \otimes I_m - B^{-T} \otimes I_m}{2} \\ \frac{E^{+T} \otimes I_m - E^{-T} \otimes I_m}{2} \end{bmatrix}$$

$$Q = S_1 - S_2 \quad (30)$$

where

$$S_1 = \begin{bmatrix} I_n \otimes A^+ & B^{+T} \otimes I_m \\ I_n \otimes D^+ & E^{+T} \otimes I_m \end{bmatrix}, \quad S_2 = \begin{bmatrix} I_n \otimes A^- & B^{-T} \otimes I_m \\ I_n \otimes D^- & E^{-T} \otimes I_m \end{bmatrix}.$$ 

Furthermore, it can be concluded that

$$T = \begin{bmatrix} I_n \otimes A^+ & B^{+T} \otimes I_m \\ I_n \otimes D^+ & E^{+T} \otimes I_m \end{bmatrix} + \begin{bmatrix} I_n \otimes A^- & B^{-T} \otimes I_m \\ I_n \otimes D^- & E^{-T} \otimes I_m \end{bmatrix}$$

$$T = S_1 + S_2 \quad (31)$$

Now, the coupled fuzzy Sylvester matrix equations (1) can be written as

$$SM = N \quad (32)$$

where

$$S = \begin{bmatrix} S_1 & S_2 \\ S_2 & S_1 \end{bmatrix}, \quad S_1 = \begin{bmatrix} I_n \otimes A^+ & B^{+T} \otimes I_m \\ I_n \otimes D^+ & E^{+T} \otimes I_m \end{bmatrix}, \quad S_2 = \begin{bmatrix} I_n \otimes A^- & B^{-T} \otimes I_m \\ I_n \otimes D^- & E^{-T} \otimes I_m \end{bmatrix}.$$
By using Lemma 2.1 for the matrix equation (32), then the following least squares based iterative algorithm leads to \( \lim_{k \to \infty} M(k) = M \):

\[
\begin{bmatrix}
\text{vec}(X(k)) \\
\text{vec}(\bar{Y}(k)) \\
-\text{vec}(\bar{X}(k)) \\
-\text{vec}(\bar{Y}(k))
\end{bmatrix}
= \begin{bmatrix}
\text{vec}(X(k-1)) \\
\text{vec}(\bar{Y}(k-1)) \\
-\text{vec}(\bar{X}(k-1)) \\
-\text{vec}(\bar{Y}(k-1))
\end{bmatrix}
+ \mu (S^T S)^{-1} S^T \begin{bmatrix}
\text{vec}(\bar{C}) \\
\text{vec}(\bar{F}) \\
-\text{vec}(\bar{C}) \\
-\text{vec}(\bar{F})
\end{bmatrix}
- \begin{bmatrix}
S_1 & S_2 \\
S_2 & S_1
\end{bmatrix}
\begin{bmatrix}
\text{vec}(X(k-1)) \\
\text{vec}(\bar{Y}(k-1)) \\
-\text{vec}(\bar{X}(k-1)) \\
-\text{vec}(\bar{Y}(k-1))
\end{bmatrix},
\]

\( 0 < \mu < 2 \)

**Corollary 3.1:** Let matrix \( S \) be in the form \( S = \begin{bmatrix} S_1 & S_2 \\ S_2 & S_1 \end{bmatrix} \) where \( S_1 = \begin{bmatrix} I_n \otimes A^+ & B^{+T} \otimes I_m \\ I_n \otimes D^+ & E^{+T} \otimes I_m \end{bmatrix} \), \( S_2 = \begin{bmatrix} I_n \otimes A^- B^{-T} \otimes I_m \\ I_n \otimes D^- E^{-T} \otimes I_m \end{bmatrix} \), then the matrix

\[
S^{[1,3]} = \frac{1}{2} \begin{bmatrix} T^{[1,3]} + Q^{[1,3]} & T^{[1,3]} - Q^{[1,3]} \\ T^{[1,3]} - Q^{[1,3]} & T^{[1,3]} + Q^{[1,3]} \end{bmatrix}
\]  

(33)

is a \([1, 3]\)-inverse of the matrix \( S \), where \( T^{[1,3]} \) and \( Q^{[1,3]} \) are \([1, 3]\)-inverse of the matrices \( T \) and \( Q \), respectively. In particular, the Moore–Penrose inverse of the matrix \( S \) is:

\[
S^+ = \frac{1}{2} \begin{bmatrix} T^+ + Q^+ & T^+ - Q^+ \\ T^+ - Q^+ & T^+ + Q^+ \end{bmatrix}
\]  

(34)

We now outline our suggested algorithm.

**Algorithm III**

**Step 1** Input crisp matrices \( A, D \in \mathbb{R}^{m \times m} \) and \( B, E \in \mathbb{R}^{n \times n} \) and input fuzzy matrices \( C, F \in \mathbb{R}^{m \times n} \), and number \( \varepsilon \).

**Step 2** Given any two initial fuzzy matrices \( X(0), Y(0) \in \mathbb{R}^{m \times n} \).

**Step 3** Compute

\[
S_1 = \begin{bmatrix} I_n \otimes A^+ & B^{+T} \otimes I_m \\ I_n \otimes D^+ & E^{+T} \otimes I_m \end{bmatrix}, \quad S_2 = \begin{bmatrix} I_n \otimes A^- & B^{-T} \otimes I_m \\ I_n \otimes D^- & E^{-T} \otimes I_m \end{bmatrix}, \quad S = \begin{bmatrix} S_1 & S_2 \\ S_2 & S_1 \end{bmatrix}.
\]

where \( A^+, B^+, D^+, E^+ \) contain the positive entries of \( A, B, D, E \), respectively, and \( A^-, B^-, D^-, E^- \) contain the absolute value of negative entries of \( A, B, D, E \), respectively, where \( A = A^+ - A^-, B = B^+ - B^-, D = D^+ - D^-, E = E^+ - E^- \).
Step 4 For \( k = 1, 2, \ldots \) until convergence

\[
\begin{bmatrix}
\text{vec}(X(k)) \\
\text{vec}(\bar{Y}(k)) \\
-\text{vec}(\bar{X}(k)) \\
-\text{vec}(\bar{Y}(k))
\end{bmatrix} = \begin{bmatrix}
\text{vec}(X(k - 1)) \\
\text{vec}(\bar{Y}(k - 1)) \\
-\text{vec}(\bar{X}(k - 1)) \\
-\text{vec}(\bar{Y}(k - 1))
\end{bmatrix} + \mu (S^T S)^{-1} S^T \left( \begin{bmatrix}
\text{vec}(C) \\
\text{vec}(F) \\
-\text{vec}(\bar{C}) \\
-\text{vec}(\bar{F})
\end{bmatrix} - \begin{bmatrix}
S_1 \\
S_2 \\
S_1 \\
-\text{vec}(\bar{Y}(k - 1))
\end{bmatrix} \right),
\]

\[0 < \mu < 2\]

Step 5 If \( \|X(k) - X(k - 1)\|/\|X(k)\| < \varepsilon, \quad \|\bar{X}(k) - \bar{X}(k - 1)\|/\|\bar{X}(k)\| < \varepsilon, \quad \|Y(k) - Y(k - 1)\|/\|Y(k)\| < \varepsilon \) and \( \|ar{Y}(k) - \bar{Y}(k - 1)\|/\|\bar{Y}(k)\| < \varepsilon \) stop; otherwise go to step 6.

Step 6 Set \( k = k + 1 \) and return to step 4.

Step 7 End.

3.4. Iterative Algorithm for Solving the Coupled Fuzzy Sylvester Matrix Equations (2)

In this section, we introduce an iterative least-squares solution of coupled fuzzy Sylvester matrix equations given in (2),

\[AXB + CYD = F_1, \quad EXG + HYN = F_2.\]

where \( A, C, E, H \in \mathbb{R}^{m \times m} \) and \( B, D, G, N \in \mathbb{R}^{n \times l} \) are given crisp matrices and \( F_1, F_2 \in \mathbb{R}^{m \times l} \) are given fuzzy matrices while \( X, Y \in \mathbb{R}^{m \times n} \) are fuzzy matrices to be determined.

The basic idea is to regard Equation (2) as two matrices

\[
R_1 = \begin{bmatrix} F_1 - CYD \\ F_2 - HYN \end{bmatrix}, \quad R_2 = \begin{bmatrix} F_1 - AXB, & F_2 - EXG \end{bmatrix}
\]

Hence, Equation (2) can be decomposed into two matrix equations of the form:

\[
S_1 X \ast T_1 = R_1, \quad S_2 Y T_2 = R_2
\]

where \( S_1 = \begin{bmatrix} A \\ E \end{bmatrix} \) and \( T_1 = \begin{bmatrix} B \end{bmatrix} \)

where \( S_2 = \begin{bmatrix} C, & H \end{bmatrix} \) and \( T_2 = \begin{bmatrix} D, & N \end{bmatrix} \)

Then we can define the following iterative formulas

\[
X(k) = X(k - 1) + \mu \left( S_1^T S_1 \right)^{-1} S_1^T \left[ R_1 - S_1 X(k - 1) \ast T_1 \right] \ast T_1^T \left( T_1 T_1^T \right)^{-1}
\]

\[
Y(k) = Y(k - 1) + \mu \left( S_2^T S_2 \right)^{-1} S_2^T \left[ R_2 - S_2 \ast Y(k - 1) T_2 \right] T_2^T \left( T_2 T_2^T \right)^{-1}
\]

\[0 < \mu < 2\]
where $\mu$ is the convergence factor. Substituting from Equations (35) and (36) into Equations (39) and (40) gives

$$X(k) = X(k - 1) + \mu (S_1^T S_1)^{-1} \begin{bmatrix} \begin{bmatrix} F_1 - AX(k - 1)B - CY(k - 1)D \\ F_2 - EX(k - 1)G - HY(k - 1)N \end{bmatrix} \end{bmatrix} T_1 T_1^T)^{-1}$$

$$Y(k) = Y(k - 1) + \mu (S_1^T S_1)^{-1} \begin{bmatrix} \begin{bmatrix} F_1 - AX(k - 1)B - CY(k - 1)D \\ F_2 - EX(k - 1)G - HY(k - 1)N \end{bmatrix} \end{bmatrix} T_2 T_2^T)^{-1}$$

The right-hand sides of these equations contain the unknown fuzzy matrices $X$ and $Y$, so it is impossible to realize the algorithm in Equations (41) and (42). By applying the hierarchical identification principle [4], the unknown fuzzy matrices $X$ and $Y$ in these equations is replaced with its estimate $X(k)$ and $Y(k)$. Thus one has

$$X(k) = X(k - 1) + \mu (S_1^T S_1)^{-1} \begin{bmatrix} \begin{bmatrix} F_1 - AX(k - 1)B - CY(k - 1)D \\ F_2 - EX(k - 1)G - HY(k - 1)N \end{bmatrix} \end{bmatrix} T_1 T_1^T)^{-1}$$

$$Y(k) = Y(k - 1) + \mu (S_1^T S_1)^{-1} \begin{bmatrix} \begin{bmatrix} F_1 - AX(k - 1)B - CY(k - 1)D \\ F_2 - EX(k - 1)G - HY(k - 1)N \end{bmatrix} \end{bmatrix} T_2 T_2^T)^{-1}$$

In this case, the iterative least-squares solutions of coupled fuzzy Sylvester matrix equations can be written as

$$\bar{X}(k) = \bar{X}(k - 1) + \mu (S_1^T S_1)^{-1} \begin{bmatrix} \begin{bmatrix} F_1 - AX(k - 1)B - CY(k - 1)D \\ F_2 - EX(k - 1)G - HY(k - 1)N \end{bmatrix} \end{bmatrix} T_1 T_1^T)^{-1}$$

$$\bar{Y}(k) = \bar{Y}(k - 1) + \mu (S_1^T S_1)^{-1} \begin{bmatrix} \begin{bmatrix} F_1 - AX(k - 1)B - CY(k - 1)D \\ F_2 - EX(k - 1)G - HY(k - 1)N \end{bmatrix} \end{bmatrix} T_2 T_2^T)^{-1}$$
\[
\begin{bmatrix} C, & H \end{bmatrix}^T \left[ F_1 - A\bar{X}(k - 1)B - C\bar{Y}(k - 1)D, & F_2 - E\bar{X}(k - 1)G - H\bar{Y}(k - 1)N \right] \\
\times \left[ D, & N \right]^T (T_2 T_2^T)^{-1},
\]

\[
\mu = \frac{2}{m + n} \text{ or } \frac{2}{\lambda_{\max}(S_1(S_1^T S_1)^{-1}S_1^T) + \lambda_{\max}(S_2(S_2^T S_2)^{-1}S_2^T) + \lambda_{\max}(T_2(T_2^T T_2)^{-1}T_2^T)}
\]

We now outline our suggested algorithm.

**Algorithm IV**

**Step 1** Input crisp matrices \(A, C, E, H \in \mathbb{R}^{m \times m}\) and \(B, D, G, N \in \mathbb{R}^{n \times l}\) and input fuzzy matrices \(F_1, F_2 \in \mathbb{R}^{m \times l}\), and number \(\epsilon\).

**Step 2** Given any two initial fuzzy matrices \(X(0), Y(0) \in \mathbb{R}^{m \times n}\).

**Step 3** Compute

\[
S_1 = \begin{bmatrix} A \\ E \end{bmatrix}, \quad T_1 = \begin{bmatrix} B \\ G \end{bmatrix}, \quad S_2 = \begin{bmatrix} C, & H \end{bmatrix}, \text{ and } T_2 = \begin{bmatrix} D, & N \end{bmatrix}
\]

**Step 4** For \(k = 1, 2, \ldots \) until convergence

\[
\begin{align*}
\bar{X}(k) &= \bar{X}(k - 1) + \mu (S_1^T S_1)^{-1} \begin{bmatrix} A \\ E \end{bmatrix} \left[ F_1 - A\bar{X}(k - 1)B - C\bar{Y}(k - 1)D, & F_2 - E\bar{X}(k - 1)G - H\bar{Y}(k - 1)N \right] \\
&\quad \times (T_1 T_1^T)^{-1},
\end{align*}
\]

\[
\begin{align*}
\bar{X}(k) &= \bar{X}(k - 1) + \mu (S_1^T S_1)^{-1} \begin{bmatrix} A \\ E \end{bmatrix} \left[ F_1 - A\bar{X}(k - 1)B - C\bar{Y}(k - 1)D, & F_2 - E\bar{X}(k - 1)G - H\bar{Y}(k - 1)N \right] \\
&\quad \times (T_1 T_1^T)^{-1},
\end{align*}
\]

\[
\begin{align*}
\bar{X}(k) &= \bar{X}(k - 1) + \mu (S_1^T S_1)^{-1} \\
&\quad \times \left[ \begin{bmatrix} C, & H \end{bmatrix} \left[ F_1 - A\bar{X}(k - 1)B - C\bar{Y}(k - 1)D, & F_2 - E\bar{X}(k - 1)G - H\bar{Y}(k - 1)N \right] \\
&\quad \times (T_1 T_1^T)^{-1},
\end{align*}
\]

\[
\begin{align*}
\bar{X}(k) &= \bar{X}(k - 1) + \mu (S_1^T S_1)^{-1} \\
&\quad \times \left[ \begin{bmatrix} C, & H \end{bmatrix} \left[ F_1 - A\bar{X}(k - 1)B - C\bar{Y}(k - 1)D, & F_2 - E\bar{X}(k - 1)G - H\bar{Y}(k - 1)N \right] \\
&\quad \times (T_1 T_1^T)^{-1},
\end{align*}
\]

\[
\mu = \frac{2}{m + n} \text{ or } \frac{2}{\lambda_{\max}(S_1(S_1^T S_1)^{-1}) + \lambda_{\max}(S_2(S_2^T S_2)^{-1}) + \lambda_{\max}(T_2(T_2^T T_2)^{-1})}
\]
Step 5 If \(||X(k) - X(k-1)|| / ||X(k)|| < \varepsilon\), \(||X(k) - X(k-1)|| / ||X(k)|| < \varepsilon\), \(||Y(k) - Y(k-1)|| / ||Y(k)|| < \varepsilon\) and \(||Y(k) - Y(k-1)|| / ||Y(k)|| < \varepsilon\) stop; otherwise go to step 6.

Step 6 Set \(k = k + 1\) and return to step 4.

Step 7 End.

**Theorem 3.2:** If the coupled fuzzy Sylvester matrix equations (2) are consistent and has a unique fuzzy solutions \(X^* = (X^*_x, X^*_y) \in \mathbb{R}^{m \times n}\) and \(Y^* = (Y^*_x, Y^*_y) \in \mathbb{R}^{m \times n}\) and

\[
\mu = \frac{2}{m+n} \text{ or } \mu = \frac{2}{\lambda_{\max}(S_1(S_1^T S_1)^{-1}S_1^T)\lambda_{\max}(T_1(T_1^T T_1)^{-1}T_1^T) + \lambda_{\max}(S_2(S_2^T S_2)^{-1}S_2^T)\lambda_{\max}(T_2(T_2^T T_2)^{-1}T_2^T)}
\]

then the iterative sequence \((X(k)), (X(k)), (Y(k))\) and \((\bar{Y}(k))\) generated by algorithm IV converges to \(X^*, \bar{X}^*, Y^*\) and \(\bar{Y}^*\), that is, \(\lim_{k \to \infty} X(k) = X^*, \lim_{k \to \infty} \bar{X}(k) = \bar{X}^*, \lim_{k \to \infty} Y(k) = Y^*\) and \(\lim_{k \to \infty} \bar{Y}(k) = \bar{Y}^*\) for any initial fuzzy matrices \(X(0), \bar{X}(0), Y(0)\) and \(\bar{Y}(0)\).

**Proof:** The proof is similar to Theorem 3.1.

### 3.5. Least Squares Based Iterative Solutions of Coupled Fuzzy Sylvester Matrix Equations (2)

In this subsection, we study least squares based iterative solutions of coupled fuzzy Sylvester matrix equations (2) that can be written as

\[
\begin{bmatrix}
B^T \otimes A \\
G^T \otimes E
\end{bmatrix}
\begin{bmatrix}
\vec{X} \\
\vec{Y}
\end{bmatrix}
=
\begin{bmatrix}
\vec{X}^*_1 \\
\vec{X}^*_2
\end{bmatrix}
\]

If \(A^+, B^+, C^+, D^+, E^+, G^+, H^+, N^+\) contain the positive entries of \(A, B, C, D, E, G, H, N\), respectively, and \(A^-, B^-, C^-, D^-, E^-, G^-, H^-, N^7\) contain the absolute value of negative entries of \(A, B, C, D, E, G, H, N\), respectively, it is obvious that \(A = A^+ - A^-, B = B^+ - B^-, C = C^+ - C^-, D = D^+ - D^-, E = E^+ - E^-, G = G^+ - G^-, H = H^+ - H^-, N = N^+ - N^-\). So, according to the properties of Kronecker operators it can be written as

\[
B^T \otimes A = (B^+ - B^-)^T \otimes (A^+ - A^-)
\]

\[
= (B^+^T \otimes A^+ + B^-^T \otimes A^-) - (B^+^T \otimes A^- + B^-^T \otimes A^+),
\]

Similarly

\[
D^T \otimes C = (D^+^T \otimes C^+ + D^-^T \otimes C^-) - (D^+^T \otimes C^- + D^-^T \otimes C^+),
\]

\[
G^T \otimes E = (G^+^T \otimes E^+ + G^-^T \otimes E^-) - (G^+^T \otimes E^- + G^-^T \otimes E^+),
\]
and $N^T \otimes H = (N^+ T \otimes H^+ + N^- T \otimes H^-) - (N^+ T \otimes H^- + N^- T \otimes H^+)$.

$$Q = \begin{bmatrix} B^T \otimes A & D^T \otimes C \\ G^T \otimes E & N^T \otimes H \end{bmatrix}$$

$$Q = \begin{bmatrix} B^+ T \otimes A^+ + B^- T \otimes A^- & D^+ T \otimes C^+ + D^- T \otimes C^- \\ G^+ T \otimes E^+ + G^- T \otimes E^- & N^+ T \otimes H^+ + N^- T \otimes H^- \end{bmatrix}$$

$$Q = S_1 - S_2$$

(50)

where

$$S_1 = \begin{bmatrix} B^+ T \otimes A^+ + B^- T \otimes A^- & D^+ T \otimes C^+ + D^- T \otimes C^- \\ G^+ T \otimes E^+ + G^- T \otimes E^- & N^+ T \otimes H^+ + N^- T \otimes H^- \end{bmatrix},$$

$$S_2 = \begin{bmatrix} B^+ T \otimes A^- + B^- T \otimes A^+ & D^+ T \otimes C^- + D^- T \otimes C^+ \\ G^+ T \otimes E^- + G^- T \otimes E^+ & N^+ T \otimes H^- + N^- T \otimes H^+ \end{bmatrix}.$$ $

In addition, it can be concluded that

$$T = \begin{bmatrix} B^+ T \otimes A^+ + B^- T \otimes A^- & D^+ T \otimes C^+ + D^- T \otimes C^- \\ G^+ T \otimes E^+ + G^- T \otimes E^- & N^+ T \otimes H^+ + N^- T \otimes H^- \end{bmatrix}$$

$$+ \begin{bmatrix} B^+ T \otimes A^- + B^- T \otimes A^+ & D^+ T \otimes C^- + D^- T \otimes C^+ \\ G^+ T \otimes E^- + G^- T \otimes E^+ & N^+ T \otimes H^- + N^- T \otimes H^+ \end{bmatrix}$$

$$T = S_1 + S_2.$$ (51)

Now, the coupled fuzzy Sylvester matrix equations (2) can be written as $SM = N$, where

$$S = \begin{bmatrix} S_1 & S_2 \\ S_2 & S_1 \end{bmatrix},$$

$$S_1 = \begin{bmatrix} B^+ T \otimes A^+ + B^- T \otimes A^- & D^+ T \otimes C^+ + D^- T \otimes C^- \\ G^+ T \otimes E^+ + G^- T \otimes E^- & N^+ T \otimes H^+ + N^- T \otimes H^- \end{bmatrix},$$

$$S_2 = \begin{bmatrix} B^+ T \otimes A^- + B^- T \otimes A^+ & D^+ T \otimes C^- + D^- T \otimes C^+ \\ G^+ T \otimes E^- + G^- T \otimes E^+ & N^+ T \otimes H^- + N^- T \otimes H^+ \end{bmatrix},$$

$$M = \begin{bmatrix} \text{vec}(X) \\ \text{vec}(Y) \\ -\text{vec}(\overline{X}) \\ -\text{vec}(\overline{Y}) \end{bmatrix},$$

$$N = \begin{bmatrix} \text{vec}(F_1) \\ \text{vec}(F_2) \\ -\text{vec}(F_1) \\ -\text{vec}(F_2) \end{bmatrix}$$
Now, applying Lemma 2.1 for the matrix equation $SM = N$, then the following least squares based iterative algorithm leads to $\lim_{k \to \infty} M(k) = M$

$$
\begin{bmatrix}
  \text{vec}(X(k)) \\
  \text{vec}(Y(k)) \\
  - \text{vec}(X(k)) \\
  - \text{vec}(Y(k))
\end{bmatrix}
= \begin{bmatrix}
  \text{vec}(X(k-1)) \\
  \text{vec}(Y(k-1)) \\
  - \text{vec}(X(k-1)) \\
  - \text{vec}(Y(k-1))
\end{bmatrix}
+ \mu (S^T S)^{-1} S^T \begin{bmatrix}
  \text{vec}(F_1) \\
  \text{vec}(F_2) \\
  - \text{vec}(F_1) \\
  - \text{vec}(F_2)
\end{bmatrix}
- \begin{bmatrix}
  S_1 S_2 \\
  S_2 S_1
\end{bmatrix}
\begin{bmatrix}
  \text{vec}(X(k-1)) \\
  \text{vec}(Y(k-1)) \\
  - \text{vec}(X(k-1)) \\
  - \text{vec}(Y(k-1))
\end{bmatrix},
$$

$0 < \mu < 2$

**Corollary 3.2:** Let matrix $S$ be in the form $S = \begin{bmatrix} S_1 & S_2 \\ S_2 & S_1 \end{bmatrix}$ where

$$
S_1 = \begin{bmatrix}
  B^+ T \otimes A^+ + B^- T \otimes A^- \\
  G^+ T \otimes E^+ + G^- T \otimes E^-
\end{bmatrix}
\begin{bmatrix}
  D^+ T \otimes C^+ + D^- T \otimes C^- \\
  N^+ T \otimes H^+ + N^- T \otimes H^-
\end{bmatrix},
$$

$$
S_2 = \begin{bmatrix}
  B^+ T \otimes A^+ + B^- T \otimes A^- \\
  G^+ T \otimes E^+ + G^- T \otimes E^+
\end{bmatrix}
\begin{bmatrix}
  D^+ T \otimes C^- + D^- T \otimes C^+ \\
  N^+ T \otimes H^- + N^- T \otimes H^+
\end{bmatrix},
$$

then the matrix

$$
S^{[1,3]} = \frac{1}{2} \begin{bmatrix}
  T^{[1,3]} + Q^{[1,3]} \\
  T^{[1,3]} - Q^{[1,3]}
\end{bmatrix}
\begin{bmatrix}
  T^{[1,3]} - Q^{[1,3]} \\
  T^{[1,3]} + Q^{[1,3]}
\end{bmatrix}
$$

(52)

is a $(1, 3)$-inverse of the matrix $S$, where $T^{[1,3]}$ and $Q^{[1,3]}$ are $(1, 3)$-inverse of the matrices $T$ and $Q$, respectively. In particular, the Moore–Penrose inverse of the matrix $S$ is

$$
S^\dagger = \frac{1}{2} \begin{bmatrix}
  T^\dagger + Q^\dagger \\
  T^\dagger - Q^\dagger
\end{bmatrix}
\begin{bmatrix}
  T^\dagger - Q^\dagger \\
  T^\dagger + Q^\dagger
\end{bmatrix}
$$

(53)

We now outline our suggested algorithm.

**Algorithm V**

**Step 1** Input crisp matrices $A, C, E, H \in \mathbb{R}^{m \times m}$ and $B, D, G, N \in \mathbb{R}^{n \times l}$ and input fuzzy matrices $F_1, F_2 \in \mathbb{R}^{m \times l}$, and number $\varepsilon$.

**Step 2** Given any two initial fuzzy matrices $X(0), Y(0) \in \mathbb{R}^{m \times n}$.

**Step 3** Compute

$$
S = \begin{bmatrix} S_1 & S_2 \\ S_2 & S_1 \end{bmatrix},
S_1 = \begin{bmatrix}
  B^+ T \otimes A^+ + B^- T \otimes A^- \\
  G^+ T \otimes E^+ + G^- T \otimes E^-
\end{bmatrix}
\begin{bmatrix}
  D^+ T \otimes C^+ + D^- T \otimes C^- \\
  N^+ T \otimes H^+ + N^- T \otimes H^-
\end{bmatrix},
$$

$$
S_2 = \begin{bmatrix}
  B^+ T \otimes A^+ + B^- T \otimes A^- \\
  G^+ T \otimes E^+ + G^- T \otimes E^+
\end{bmatrix}
\begin{bmatrix}
  D^+ T \otimes C^- + D^- T \otimes C^+ \\
  N^+ T \otimes H^- + N^- T \otimes H^+
\end{bmatrix}
$$
where $A^+, B^+, C^+, D^+, E^+, G^+, H^+, N^+$ contain the positive entries of $A, B, C, D, E, G, H, N$, respectively, and $A^-, B^-, C^-, D^-, E^-, G^-, H^-, N^-$ contain the absolute value of negative entries of $A, B, C, D, E, G, H, N$ where $A = A^+ - A^-, B = B^+ - B^-, C = C^+ - C^-, D = D^+ - D^-, E = E^+ - E^-, G = G^+ - G^-, H = H^+ - H^-, N = N^+ - N^-$. 

**Step 4** For $k = 1, 2, \cdots$ until convergence

$$
\begin{bmatrix}
\text{vec}(X(k)) \\
\text{vec}(Y(k)) \\
- \text{vec}(X(k)) \\
- \text{vec}(Y(k))
\end{bmatrix}
= 
\begin{bmatrix}
\text{vec}(X(k - 1)) \\
\text{vec}(Y(k - 1)) \\
- \text{vec}(X(k - 1)) \\
- \text{vec}(Y(k - 1))
\end{bmatrix}
+ \mu (S^T S)^{-1} S^T
\begin{bmatrix}
\text{vec}(F_1) \\
\text{vec}(F_2) \\
- \text{vec}(F_1) \\
- \text{vec}(F_2)
\end{bmatrix}
- 
\begin{bmatrix} S_1 S_2 \\
S_2 S_1 \\
- \text{vec}(X(k - 1)) \\
- \text{vec}(Y(k - 1))
\end{bmatrix},
$$

$0 < \mu < 2$

**Step 5** If $||X(k) - X(k - 1)||/||X(k)|| < \varepsilon$, $||X(k) - X(k - 1)||/||X(k)|| < \varepsilon$, $||Y(k) - Y(k - 1)||/||Y(k)|| < \varepsilon$ and $||Y(k) - Y(k - 1)||/||Y(k)|| < \varepsilon$ stop; otherwise go to step 6.

**Step 6** Set $k = k + 1$ and return to step 4.

**Step 7** End.

4. Numerical Examples

Numerical examples to demonstrate the efficacy of the proposed algorithms are given in this section.

**Example 4.1:** In this example, we demonstrate our algorithm I and algorithm II theoretical results for solving coupled fuzzy Sylvester matrix equations given in (1),

$$AX + YB = C, \quad DX + YE = F.$$ 

Given

$$A = \begin{bmatrix} 1 & 3 \\ 2 & 5 \end{bmatrix}, \quad D = \begin{bmatrix} 2 & 2 \\ 1 & 3 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 2 & 5 \\ 3 & 4 & 1 \\ 2 & 0 & 1 \end{bmatrix}, \quad E = \begin{bmatrix} 3 & 1 & 2 \\ 4 & 1 & 3 \\ 5 & 2 & 2 \end{bmatrix}, \quad C = \begin{bmatrix} (-36 + 54 r, 36 - 38 r) & (-27 + 30 r, 27 - 30 r) & (-50 + 33 r, 50 - 33 r) \\ (-40 + 38 r, 40 - 38 r) & (-30 + 24 r, 30 - 24 r) & (-42 + 27 r, 42 - 27 r) \end{bmatrix},$$

$$F = \begin{bmatrix} (-63 + 54 r, 63 - 54 r) & (-29 + 21 r, 29 - 21 r) & (-46 + 34 r, 46 - 34 r) \\ (-48 + 33 r, 48 - 33 r) & (-21 + 13 r, 21 - 13 r) & (-36 + 21 r, 36 - 21 r) \end{bmatrix}.$$
This system of coupled fuzzy Sylvester matrix equations has a unique solution

\[
\begin{align*}
X &= \begin{bmatrix}
1 + 2 r, 1 - 2 r \\
(-3 + 2 r, 3 - 2 r) \\
(-4 + 5 r, 4 - 5 r) \\
\end{bmatrix}, \\
Y &= \begin{bmatrix}
-5 + 3 r, 5 - 3 r \\
(-2 + 4 r, 2 - 4 r) \\
(-1 + r, 1 - r) \\
\end{bmatrix}.
\end{align*}
\]

Algorithm I and algorithm II are applied to solve generalized Sylvester matrix equations (1). When selecting the initial matrices as \(X(0), Y(0) = \begin{bmatrix} (1, 1) & (1, 1) & (1, 1) \\
(1, 1) & (1, 1) & (1, 1) \end{bmatrix}\). Algorithm I is convergent for \(0 < \mu < 0.5\) and the iteration process stops at \(k = 195\). While, algorithm II is convergent for \(0 < \mu < 0.5\) and the iteration process stops at \(k = 120\). The iterative solution \(X(k) = (X(k), X(k))\) and \(Y(k) = (Y(k), Y(k))\) for algorithm I is given in Table 1 for \(\mu = 0.5\). And the iterative solution \(X(k) = (X(k), X(k))\) and \(Y(k) = (Y(k), Y(k))\) for algorithm II is given in Table 2 for \(\mu = 0.5\).

We can see that the suggested modified Iterative algorithm (algorithm II) converges faster than iterative algorithm I to solve the coupled fuzzy Sylvester matrix equations (1).

**Example 4.2:** In this example, we demonstrate our algorithm III theoretical results for solving coupled fuzzy Sylvester matrix equations given in (1),

\[AX + YB = C, \quad DX + YE = F.\]
Given

\[ A = \begin{bmatrix} 2 & -1 \\ -3 & 3 \end{bmatrix}, \quad D = \begin{bmatrix} 2 & -3 \\ 3 & -2 \end{bmatrix}, \quad B = \begin{bmatrix} 2 & -3 \\ -1 & 3 \end{bmatrix}, \quad E = \begin{bmatrix} 3 & -1 \\ -2 & 5 \end{bmatrix}, \]

\[ C = \begin{bmatrix} (7r, 18 - 7r) \\ (-17 + 10r, 15 - 10r) \end{bmatrix}, \quad \]

\[ F = \begin{bmatrix} (-8 + 12r, 23 - 12r) \\ (-3 + 12r, 32 - 12r) \end{bmatrix}. \]

This system of coupled fuzzy Sylvester matrix equations has a unique solution

\[ X = \begin{bmatrix} (1 + r, 5 - r) & (2 + r, 4 - r) \\ (-1 + r, 3 - r) & (1 + 2r, 3 - 2r) \end{bmatrix}, \quad Y = \begin{bmatrix} (3 + r, 4 - r) & (1 + 2r, 5 - 2r) \\ (2 + r, 3 - r) & (-3 + 2r, 3 - 2r) \end{bmatrix}. \]

We use algorithm III to solve generalized Sylvester matrix equations (1).
Example 4.3: In this example, we illustrate our theoretical results of algorithm V for solving coupled fuzzy Sylvester matrix equations given in (1),

\[ AXB + CYD = F_1, \quad EXG + HYN = F_2. \]

Given

\[
A = \begin{bmatrix} 2 & -2 \\ -3 & 4 \end{bmatrix}, \quad C = \begin{bmatrix} -1 & -3 \\ -1 & 2 \end{bmatrix}, \quad E = \begin{bmatrix} 2 & 1 \\ -1 & -2 \end{bmatrix}, \quad H = \begin{bmatrix} -2 & 3 \\ 1 & -2 \end{bmatrix},
\]

\[
B = \begin{bmatrix} 3 & 2 \\ 2 & -1 \end{bmatrix}, \quad D = \begin{bmatrix} -1 & 2 \\ -1 & -2 \end{bmatrix}, \quad G = \begin{bmatrix} 3 & -1 \\ 0 & 1 \end{bmatrix},
\]

\[
N = \begin{bmatrix} 5 & -3 \\ 1 & 0 \end{bmatrix}, \quad F_1 = \begin{bmatrix} (-47 + 42r, 37 - 33r) & (-58 + 34r, 14 - 30r) \\ (-37 + 55r, 99 - 71r) & (-51 + 48r, 62 - 57r) \end{bmatrix},
\]

\[
F_2 = \begin{bmatrix} (-17 + 49r, 68 - 42r) & (-40 + 22r, 5 - 29r) \\ (-26 + 27r, 37 - 38r) & (-17 + 21r, 16 - 15r) \end{bmatrix}
\]

This system of coupled fuzzy Sylvester matrix equations has a unique solution

\[
X = \begin{bmatrix} (1 + r, 3 - r) & (1 + 2r, 4 - r) \\ (2 + r, 5 - 2r) & (1 + r, 3 - r) \end{bmatrix}
\]
\[ \mathbf{Y} = \begin{bmatrix}
(3 + r, 5 - 2r) & (2 + r, 3 - r) \\
(2 + r, 4 - r) & (-1 + r, 3 - r)
\end{bmatrix}. \]

We apply algorithm V to solve the generalized Sylvester matrix equations (2).

\[
\mathbf{S}_1 = \begin{bmatrix}
6 & 0 & 4 & 0 & 1 & 3 & 0 & 0 \\
0 & 12 & 0 & 8 & 0 & 4 & 4 & 0 \\
4 & 0 & 0 & 2 & 0 & 0 & 2 & 6 \\
0 & 8 & 3 & 0 & 4 & 0 & 0 & 8 \\
6 & 3 & 0 & 0 & 0 & 15 & 0 & 3 \\
0 & 6 & 0 & 0 & 5 & 0 & 1 & 0 \\
0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 6 & 0 & 0
\end{bmatrix}, \quad \mathbf{S}_2 = \begin{bmatrix}
0 & 6 & 0 & 4 & 0 & 0 & 2 & 6 \\
9 & 0 & 6 & 0 & 2 & 0 & 0 & 8 \\
0 & 4 & 2 & 0 & 2 & 6 & 0 & 0 \\
6 & 0 & 0 & 4 & 0 & 8 & 4 & 0 \\
0 & 0 & 0 & 0 & 10 & 0 & 2 & 0 \\
3 & 0 & 0 & 0 & 0 & 10 & 2 & 0 \\
2 & 1 & 2 & 1 & 0 & 9 & 0 & 0 \\
0 & 2 & 0 & 2 & 3 & 0 & 0 & 0
\end{bmatrix},
\]

\[
\mathbf{T} = \mathbf{S}_1 + \mathbf{S}_2, \quad \mathbf{Q} = \mathbf{S}_1 - \mathbf{S}_2, \quad \mathbf{S} = \begin{bmatrix}
\mathbf{S}_1 & \mathbf{S}_2 \\
\mathbf{S}_2 & \mathbf{S}_1
\end{bmatrix}, \quad \mathbf{S}^\dagger = \begin{bmatrix}
\begin{bmatrix}
\mathbf{vec}(\mathbf{X}) \\
\mathbf{vec}(\mathbf{Y}) \\
-\mathbf{vec}(\overline{\mathbf{X}}) \\
-\mathbf{vec}(\overline{\mathbf{Y}})
\end{bmatrix} \\
\mathbf{vec}(\mathbf{F}_1) \\
\mathbf{vec}(\mathbf{F}_2) \\
-\mathbf{vec}(\overline{\mathbf{F}}_1) \\
-\mathbf{vec}(\overline{\mathbf{F}}_2)
\end{bmatrix} = \begin{bmatrix}
-47 + 42r \\
-37 + 55r \\
-58 + 34r \\
-51 + 48r \\
-17 + 49r \\
-26 + 27r \\
-40 + 22r \\
-17 + 21r \\
-37 + 33r \\
-99 + 71r \\
-14 + 30r \\
-62 + 57r \\
-68 + 42r \\
-37 + 38r \\
-5 + 29r \\
-16 + 15r
\end{bmatrix} = \begin{bmatrix}
0.9999 + 1.0000r \\
2.0000 + 0.9999r \\
0.9999 + 2.0000r \\
1.0000 + 0.9999r \\
3.0000 + 0.9999r \\
1.9999 + 1.0000r \\
1.9999 + 1.0000r \\
-0.9999 + 0.9999r \\
-3.0000 + 1.0000r \\
-4.9999 + 1.9999r \\
-4.0000 + 1.0000r \\
-2.9999 + 0.9999r \\
-4.9999 + 1.9999r \\
-4.0000 + 1.0000r \\
-3.0000 + 1.0000r \\
-2.9999 + 0.9999r
\end{bmatrix}.
\]

When the initial matrices are chosen as \(\mathbf{X}(0), \mathbf{Y}(0) = \begin{bmatrix}
(0, 0) \\
(0, 0)
\end{bmatrix}\). The algorithm V is convergent for \(0 < \mu < 0.7\). After 145 iterations we obtain

\[
\mathbf{X} = \begin{bmatrix}
(1.0000 + 0.9999r, 3.0000 - 1.0000r) \\
(2.0000 + 0.9999r, 4.9999 - 1.9999r)
\end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix}
(3.0000 + 0.9999r, 4.9999 - 1.9999r) \\
(1.9999 + 1.0000r, 3.0000 - 1.0000r)
\end{bmatrix}.
\]

5. Conclusion

In this paper, five iterative algorithms have been constructed to solve two coupled fuzzy Sylvester matrix equations. Two iterative algorithms are based on the generalized inverse of
the coefficient matrix, then iterative solutions are constructed by applying the hierarchical identification principle and by using the block-matrix inner product to solve the two coupled fuzzy Sylvester matrix equations (1) and (2). Also, two least-squares iterative algorithm to solve the two coupled fuzzy Sylvester matrix equations (1) and (2). And a modified iterative algorithm for solving the coupled fuzzy Sylvester matrix equations (1) is proposed. This proposed modification is compared with the first algorithm where our modification exhibits fast convergence behavior. When these two coupled fuzzy Sylvester matrix equations are consistent, for any initial arbitrary fuzzy matrices \(X(0), Y(0)\) the solutions can be obtained. We tested the proposed algorithms using MATLAB and the results verify our theoretical findings.

**Acknowledgments**

The authors would like to express their heartfelt thanks to the editor and anonymous referees for their useful comments.

**Conflict of interest**

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

**Funding**

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

**ORCID**

Ahmed M. E. Bayoumi [http://orcid.org/0000-0002-7691-4111]

**References**

[1] Ding F, Liu PX, Ding J. Iterative solutions of the generalized Sylvester matrix equations by using the hierarchical identification principle. Appl Math Comput. 2008;197(1):41–50.

[2] Ding F, Chen T. Iterative least squares solutions of coupled Sylvester matrix equations. Syst Cont Lett. 2005;54(2):95–107.

[3] Ding F, Chen T. On iterative solutions of general coupled matrix equations. SIAM J Cont Optim. 2006;44(6):2269–2284.

[4] Ding F, Chen T. Gradient based iterative algorithms for solving a class of matrix equations. IEEE Trans Automat Cont. 2005;50(8):1216–1221.

[5] Hajarian M. Generalized reflexive and anti-reflexive solutions of the coupled Sylvester matrix equations via CD algorithm. J Vib Cont. 2018;24(2):343–356.

[6] Zhang H. Iterative solutions of a set of matrix equations by using the hierarchical identification principle. Abst Appl Analy. 2014; 1–10. doi:10.1155/2014/649524. Article ID 649524.

[7] Ramadan MA, El-Danaf TS, E AM. Bayoumi, a modified gradient-based algorithm for solving extended Sylvester-conjugate matrix equations. Asian J Cont. 2018;20(1):228–235.

[8] Friedman M, Ming M, Kandel A. Fuzzy linear systems. Fuzzy Set Syst. 1998;96:201–209.

[9] Ma M, Friedman M, Kandel A. A new fuzzy arithmetic. Fuzzy Set Syst. 1999;108:83–90.

[10] Sadeghi A, Abbaspandy S, Abbasnejad ME. The common solution of the pair of fuzzy matrix equations. World Appl Sci J. 2011;15(2):232–238.
[11] Bayoumi AME. Finite iterative Hamiltonian solutions of the generalized coupled Sylvester-conjugate matrix equations. Trans Ins Meas Cont. 2019;41(4):1139–1148.
[12] Bayoumi AME, Ramadan MA. Finite iterative Hermitian R-conjugate solutions of the generalized coupled Sylvester-conjugate matrix equations. Comp Math Appl. 2018;75:3367–3378.
[13] Behera D, Chakraverty S. Solution to Fuzzy System of Linear Equations with Crisp Coefficients. Fuzzy Inform Eng. 2013;5(2):205–219.
[14] Wang QW, Lv RY, Zhang Y. The least-squares solution with the least norm to a system of tensor equations over the quaternion algebra. Lin Multilin Algebra. 2020. doi:10.1080/03081087.2020.1779172
[15] Ramadan MA, El-Danaf TS, Bayoumi AME. Two iterative algorithms for the reflexive and Hermetian reflexive solutions of the generalized Sylvester matrix equation. J Vib Cont. 2015;21(3):483–492.