On the classic Solution of Fuzzy Linear Matrix Equations

Zhijie Jin*  Jieyong Zhou†  Qixiang He‡

May 15, 2021

Abstract

Since most time-dependent models may be represented as linear or non-linear dynamical systems, fuzzy linear matrix equations are quite general in signal processing, control and system theory. A uniform method of finding the classic solution of fuzzy linear matrix equations is presented in this paper. Conditions of solution existence are also studied here. Under the framework, a numerical method to solve fuzzy generalized Lyapunov matrix equations is developed. In order to show the validation and efficiency, some selected examples and numerical tests are presented.

Keyword: The general form of fuzzy linear matrix equations; classic solution; numerical solution; fuzzy generalized Lyapunov matrix equations.

1 Introduction

Let $A_i$ and $B_i$, $i = 1, 2, \cdots, l$, are $n \times s$ and $t \times m$ real or complex matrices, respectively. The general form of linear matrix equations is

$$A_1XB_1 + A_2XB_2 + \cdots + A_lXB_l = C,$$

where $C$ is an $n \times m$ matrix. This general form covers many kinds of linear matrix equations such as Lyapunov matrix equations, generalized Lyapunov matrix equations, Sylvester matrix equations, generalized Sylvester matrix equations, etc. Those matrix equations arise from a lot of applications including control
theory, stability of differential equation, generalized eigenvalue solution, etc. [5, 11, 12].

In applications, when the information of the corresponding model is imprecise, fuzzy matrix equations are introduced. Solution of fuzzy linear systems (matrix equation or linear equations) is more complicated than solution of crisp linear systems. Since a fuzzy number can be seen as a union of a series of closed intervals, we can implement the fuzzy number arithmetic according to the interval arithmetic. In this situation, the solution we obtain is called the classic solution of fuzzy linear systems [3]. If the crisp solution of 1-cut of a fuzzy linear system is extended to a fuzzy solution, the solution is called the new solution of a fuzzy linear system [3]. The classic solution always satisfies its fuzzy linear system, but the new solution may, or may not, satisfy the system. However, the new solution always exists but the classic solution may fail exist.

the literature of solving fuzzy linear matrix equations are abundant, we list some of them here. In [10], the fuzzy Sylvester matrix equation

\[ A\tilde{X} + \tilde{X}B = \tilde{C}, \]

where \( A \) and \( B \) are crisp coefficient matrices, \( \tilde{C} \) is a fuzzy matrix, and \( \tilde{X} = (\tilde{x}_{ij}) \) is an unknown fuzzy matrix, is considered by using Kronecker product to transfer it to a fuzzy linear equation, and then its classic solution is considered. Because the Kronecker product is employed there, the method enlarges the scale of the matrix equation, so it can not be used to solve large scale fuzzy Sylvester matrix equations. In [8], authors consider the classic solution of fuzzy Sylvester matrix equations based on the interval arithmetic. their method keeps the scale of the matrix equations unchanged, so it can be used to solve large scale fuzzy Sylvester matrix equations. Moreover, the classic solution of fully fuzzy Sylvester matrix equations, where all of the matrices in (1) are fuzzy matrices, is considered in [9]. The method there can also keep the scale of the equation unchanged. The new solutions of the fuzzy and the fully fuzzy Sylvester matrix equations are studied in [6] and [15], respectively. But there are many other kinds of linear matrix equations occurring in fuzzy situation. For example, the generalized Lyapunov matrix equation

\[ MXS^T + SXM + \sum_{j=1}^{N} A_j X A_j^T = B, \]

where \( M, B \) and \( N \) are an \( n \times n \) matrices, \( A_j \) are \( n \times n \) low rank matrices. This matrix equation naturally arises in the context of model order reduction of bilinear control systems and linear parameter-varying systems as well as for the stability analysis of linear stochastic differential equations; see, e.g., [2, 4, 16] references therein. Thus, a uniform method to solve fuzzy linear matrix equations is needed.

In this paper, we consider the classic solution of the general form of fuzzy linear matrix equation

\[ A_1\tilde{X} B_1 + A_2\tilde{X} B_2 + \cdots + A_l\tilde{X} B_l = \tilde{C}, \]
where $A_i$ and $B_i$, $i = 1, 2, \cdots, l$ are $n \times s$ and $t \times m$ crisp real matrices, respectively, $\tilde{C}$ is an $n \times m$ fuzzy matrix and $\tilde{X}$ is the unknown. We try to develop an extension method by the interval arithmetic to keep the scale of (3) unenlarged. This method can provide a uniform framework to solve the classic solution of the general form of fuzzy linear matrix equations. Especially, based on this framework, we present a numerical method to solve the generalized fuzzy Lyapunov equation

$$M \tilde{X} S^T + S \tilde{X} M + \sum_{j=1}^{N} A_j \tilde{X} A_j^T = \tilde{B}, \quad (4)$$

where $M$, $S$, and $A_j$ are crisp $n \times n$ real matrices, $\tilde{B}$ is an $n \times n$ fuzzy matrix and $\tilde{X}$ is the fuzzy unknown.

The rest of this paper is organized as follows. In section 2, some preliminaries and notation are presented. In section 3, the extension method of the general form of fuzzy linear matrix equations is presented. In section 4, conditions of the classic solution existence are studied. In section 5, A numerical method for the classic solution of fuzzy general Lyapunov matrix equations is developed and some corresponding numerical tests are provided.

## 2 Preliminaries

In this section, we mainly introduce some basic knowledge of this paper.

**Definition 2.1.** A fuzzy number is a function $u : \mathbb{R} \mapsto [0, 1]$ satisfies:

1. $u$ is an upper semi-continuous function on $\mathbb{R}$,
2. $u(x) = 0$ outside some interval $[a, d]$,
3. for $a, b, c, d \in \mathbb{R}$ with $a \leq b \leq c \leq d$ satisfies
   
   (a) $u(x)$ is a monotonic increasing function on $[a, b]$,
   
   (b) $u(x)$ is a monotonic decreasing function on $[c, d]$,
   
   (c) $u(x) = 1, \forall x \in [b, c]$.

According to Definition 2.1, we can represent a fuzzy number in the following form:

$$u(x) = \begin{cases} 
  u_L(x), & a \leq x \leq b; \\
  1, & b \leq x \leq c; \\
  u_R(x), & c < x \leq d; \\
  0, & \text{elsewhere}
\end{cases}$$

where $u_L : [a, b] \mapsto [0, 1]$ and $u_R : [c, d] \mapsto [0, 1]$ are the left and right membership function of fuzzy number $u$. A pair of $u_L$ and $u_R$ corresponds to a fuzzy number.

Another definition of fuzzy number is as follows:
Definition 2.2. For \( r \in [0, 1] \), a fuzzy number can be expressed by a pair function \((\underline{u}(r), \overline{u}(r))\), \(0 \leq r \leq 1\) satisfies:

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

When \( \underline{u}(r) = \overline{u}(r) \), the fuzzy number is a crisp number. Especially, a trapezoidal fuzzy number can be presented by \( u(x_0, y_0, \alpha, \beta) \) in the form:

\[
u(x) = \begin{cases}
\frac{1}{\alpha}(x - x_0 + \alpha), & x_0 - \alpha \leq x \leq x_0, \\
1, & x_0 \leq x \leq y_0, \\
\frac{1}{\beta}(y_0 - x + \beta), & y_0 \leq x \leq y_0 + \beta, \\
0, & \text{elsewhere}.
\end{cases}
\]

This form corresponds to \( u(r) = x_0 - \alpha + \alpha r, \overline{u}(r) = y_0 + \beta - \beta r \). When \( x_0 = y_0 \), it is a triangular fuzzy number.

The arithmetic operations for fuzzy numbers \( e = (x(r), \overline{x}(r)), e = (y(r), \overline{y}(r)) \) and real number \( k \) are the following:

1. \( x = y \) if and only if \( \underline{x}(r) = \underline{y}(r), \overline{x}(r) = \overline{y}(r) \),
2. \( x + y = (x(r) + y(r), \overline{x}(r) + \overline{y}(r)) \),
3. \( kx = \begin{cases}
(\alpha kx(r), k\overline{x}(r)), & k \geq 0, \\
(k\overline{x}(r), k\overline{x}(r)), & k < 0.
\end{cases}
\]

3 An extension method of the general form of fuzzy linear matrix equation

In this section, we drive an extension method of (3). Since \( \widetilde{C} = \sum_{k=1}^{l} A_k \widetilde{X} B_k \), then we obtain:

\[
\widetilde{c}_{ij} = \sum_{k=1}^{l} \sum_{q=1}^{r} (A_k X)_{ij} b_{qj}^{(k)} = \sum_{k=1}^{l} \sum_{q=1}^{r} \left( \sum_{s=1}^{r} a_{is}^{(k)} \widetilde{x}_{sq} b_{qj}^{(k)} \right). \tag{5}
\]

We define matrices \( A_k^+ = (a_{ij}^{(k)+}) \) and \( A_k^- = (a_{ij}^{(k)-}) \) as the following:

\[
\begin{align*}
&\begin{cases}
\alpha_{ij} > 0 \Rightarrow a_{ij}^{(k)+} = a_{ij}, a_{ij}^{(k)-} = 0, \\
\alpha_{ij} < 0 \Rightarrow a_{ij}^{(k)+} = 0, a_{ij}^{(k)-} = -a_{ij}.
\end{cases}
\end{align*}
\]

We have \( A_k = A_k^+ - A_k^- \) and \(|A_k| = A_k^+ + A_k^-\). By the similar definitions of \( B_k^+ = (b_{ij}^{(k)+}) \) and \( B_k^- = (b_{ij}^{(k)-}) \), we also have \( B_k = B_k^+ - B_k^- \) and \(|B_k| = \)
Then we can rewrite (5) as the following:

\[ \tilde{c}_{ij} = \sum_{k=1}^{l} \sum_{q=1}^{r} \left( \sum_{s=1}^{r} a_{is}^+ x_{sq} b_{qj}^+ - \sum_{s=1}^{r} a_{is}^- x_{sq} b_{qj}^- \right) (b_{qj}^+ - b_{qj}^-) \]

By doing the following operations:

\[ \tilde{c}_{ij} = \sum_{k=1}^{l} \sum_{q=1}^{r} \left( \sum_{s=1}^{r} a_{is}^+ x_{sq} b_{qj}^+ - \sum_{s=1}^{r} a_{is}^- x_{sq} b_{qj}^- \right) \]

\[ = \left( \sum_{k=1}^{l} \sum_{q=1}^{r} \sum_{s=1}^{r} a_{is}^+ x_{sq} b_{qj}^+ + \sum_{k=1}^{l} \sum_{q=1}^{r} \sum_{s=1}^{r} a_{is}^- x_{sq} b_{qj}^- \right) \]

and let \( \tilde{x}_{sq} = (\pi_{sq}, \pi_{sq}) \) and \( \tilde{c}_{ij} = (\tilde{c}_{ij}, \tilde{c}_{ij}) \), we have

\[ \tilde{c}_{ij} = \left( \sum_{q=1}^{r} \sum_{s=1}^{r} a_{is}^+ x_{sq} b_{qj}^+ + \sum_{q=1}^{r} \sum_{s=1}^{r} a_{is}^- x_{sq} b_{qj}^- \right) - \]

\[ \left( \sum_{q=1}^{r} \sum_{s=1}^{r} a_{is}^+ x_{sq} b_{qj}^- + \sum_{q=1}^{r} \sum_{s=1}^{r} a_{is}^- x_{sq} b_{qj}^+ \right) \]

and

\[ \tilde{c}_{ij} = \left( \sum_{q=1}^{r} \sum_{s=1}^{r} a_{is}^+ x_{sq} b_{qj}^+ + \sum_{q=1}^{r} \sum_{s=1}^{r} a_{is}^- x_{sq} b_{qj}^- \right) - \]

\[ \left( \sum_{q=1}^{r} \sum_{s=1}^{r} a_{is}^+ x_{sq} b_{qj}^- + \sum_{q=1}^{r} \sum_{s=1}^{r} a_{is}^- x_{sq} b_{qj}^+ \right) \].

By the fact that \( \tilde{X} = (X, \bar{X}) \), where \( X = (x_{sq})_{s \times t} \) and \( \bar{X} = (\pi_{sq})_{s \times t} \), we have

\[ C = \left( \sum_{k=1}^{l} A_k^+ X B_k^+ + \sum_{k=1}^{l} A_k^- X B_k^- \right) - \left( \sum_{k=1}^{l} A_k^+ X B_k^- + \sum_{k=1}^{l} A_k^- X B_k^+ \right) \]
and
\[
\bar{C} = \left( \sum_{k=1}^{l} A^{+}_{k} X B^{+}_{k} + \sum_{k=1}^{l} A^{-}_{k} X B^{-}_{k} \right) - \left( \sum_{k=1}^{l} A^{+}_{k} X B^{-}_{k} + \sum_{k=1}^{l} A^{-}_{k} X B^{+}_{k} \right). \tag{9}
\]

If we consider these two expressions in the way of matrix multiplication, we have
\[
\sum_{k=1}^{l} \begin{pmatrix} A^{+}_{k} & -A^{-}_{k} \\ -A^{+}_{k} & A^{-}_{k} \end{pmatrix} \begin{pmatrix} X & \overline{X} \end{pmatrix} \begin{pmatrix} B^{+}_{k} \\ -B^{-}_{k} \end{pmatrix} + \sum_{k=1}^{l} \begin{pmatrix} A^{-}_{k} & -A^{+}_{k} \\ -A^{-}_{k} & A^{+}_{k} \end{pmatrix} \begin{pmatrix} X & \overline{X} \end{pmatrix} \begin{pmatrix} B^{-}_{k} \\ -B^{+}_{k} \end{pmatrix} = \begin{pmatrix} C & \overline{C} \end{pmatrix}. \tag{10}
\]

We can extend the (10) such that
\[
\sum_{k=1}^{l} \begin{pmatrix} A^{+}_{k} & -A^{-}_{k} \\ -A^{+}_{k} & A^{-}_{k} \end{pmatrix} \begin{pmatrix} X & \overline{X} \end{pmatrix} \begin{pmatrix} B^{+}_{k} \\ -B^{-}_{k} \end{pmatrix} + \sum_{k=1}^{l} \begin{pmatrix} A^{-}_{k} & -A^{+}_{k} \\ -A^{-}_{k} & A^{+}_{k} \end{pmatrix} \begin{pmatrix} X & \overline{X} \end{pmatrix} \begin{pmatrix} B^{-}_{k} \\ -B^{+}_{k} \end{pmatrix} = \begin{pmatrix} C & \overline{C} \end{pmatrix}. \tag{11}
\]

Let \( S_{k} = \begin{pmatrix} A^{+}_{k} & -A^{-}_{k} \\ -A^{+}_{k} & A^{-}_{k} \end{pmatrix}, \ T_{k} = \begin{pmatrix} B^{+}_{k} & -B^{-}_{k} \\ -B^{+}_{k} & B^{-}_{k} \end{pmatrix}, \ Y = \begin{pmatrix} X & \overline{X} \end{pmatrix} \) and \( G = \begin{pmatrix} C & \overline{C} \end{pmatrix}. \) We can write (11) in the following form:
\[
\sum_{k=1}^{l} S_{k} Y T_{k} = G. \tag{12}
\]

So the general form of fuzzy linear matrix equations (3) has been successfully transformed into the general form of crisp linear matrix equations (12), with the scales of the coefficient matrices to be \( 2n \times 2s \) and \( 2t \times 2m. \) Let’s take some examples to explain the method.

**Example 3.1.** Let
\[
A = \begin{pmatrix} 1 & 2 \\ -1 & 3 \end{pmatrix}, \quad B = \begin{pmatrix} -1 & 1 \\ 3 & 2 \end{pmatrix},
\]
and
\[
\bar{C} = \begin{pmatrix} -7 + 21r, 25 - 17r \\ -27 + 23r, 14 - 27r \end{pmatrix}, \quad \begin{pmatrix} 8 + 23r, 38 - 19r \\ -7 + 26r, 31 - 30r \end{pmatrix}.
\]
So the equation is

\[
\begin{pmatrix}
1 & 2 \\
-1 & 3
\end{pmatrix}
\begin{pmatrix}
-1 & 1 \\
2 & 3
\end{pmatrix}
= 
\begin{pmatrix}
-7 + 21r & 25 - 17r \\
-27 + 23r & 14 - 27r
\end{pmatrix}
\begin{pmatrix}
8 + 23r & 38 - 19r \\
-7 + 26r & 31 - 30r
\end{pmatrix}.
\]

According to our method, we obtain

\[
A^+ = \begin{pmatrix}
1 & 2 \\
0 & 3
\end{pmatrix},
A^- = \begin{pmatrix}
0 & 0 \\
1 & 0
\end{pmatrix},
B^+ = \begin{pmatrix}
0 & 1 \\
3 & 2
\end{pmatrix},
\text{and } B^- = \begin{pmatrix}
1 & 0 \\
0 & 0
\end{pmatrix}.
\]

Put these matrices into (13), then we have

\[
\left( A^+ X B^+ + A^- X B^- \right) - \left( A^+ X B^- + A^- X B^+ \right)
= \begin{pmatrix}
1 & 2 \\
0 & 3
\end{pmatrix}
\begin{pmatrix}
0 & 1 \\
3 & 2
\end{pmatrix}
\begin{pmatrix}
0 & 0 \\
1 & 0
\end{pmatrix}
\begin{pmatrix}
1 & 0 \\
0 & 0
\end{pmatrix}
- \begin{pmatrix}
1 & 2 \\
0 & 3
\end{pmatrix}
\begin{pmatrix}
1 & 0 \\
0 & 0
\end{pmatrix}
\begin{pmatrix}
0 & 0 \\
1 & 0
\end{pmatrix}
\begin{pmatrix}
0 & 1 \\
3 & 2
\end{pmatrix}
= \begin{pmatrix}
-7 + 21r & 8 + 23r \\
-27 + 23r & -7 + 26r
\end{pmatrix} = C.
\]

Similarly, by (9) we have

\[
\left( A^+ X B^+ + A^- X B^- \right) - \left( A^+ X B^- + A^- X B^+ \right)
= \begin{pmatrix}
1 & 2 \\
0 & 3
\end{pmatrix}
\begin{pmatrix}
0 & 1 \\
3 & 2
\end{pmatrix}
\begin{pmatrix}
0 & 0 \\
1 & 0
\end{pmatrix}
\begin{pmatrix}
1 & 0 \\
0 & 0
\end{pmatrix}
- \begin{pmatrix}
1 & 2 \\
0 & 3
\end{pmatrix}
\begin{pmatrix}
1 & 0 \\
0 & 0
\end{pmatrix}
\begin{pmatrix}
0 & 0 \\
1 & 0
\end{pmatrix}
\begin{pmatrix}
0 & 1 \\
3 & 2
\end{pmatrix}
= \begin{pmatrix}
25 - 17r & 38 - 19r \\
14 - 27r & 31 - 30r
\end{pmatrix} = \overline{C}.
\]

We can see that all the fuzzy matrices has been transformed into crisp matrices in (13) and (14), so it’s possible to obtain the values of \(X\) and \(\overline{X}\) by the crisp way of matrix calculation. The result is

\[
\overline{X} = \begin{pmatrix}
1 + 2r & 2r \\
1 + r & 1 - r
\end{pmatrix}
\quad \text{and} \quad
\overline{X} = \begin{pmatrix}
2 - r & 4 - r & 3 - r \\
3 - r & 1 - r & 3 - 2r
\end{pmatrix}.
\]

So the unknown fuzzy matrix is

\[
\begin{pmatrix}
(r, 2 - r) & (1 + 2r, 4 - r) & (2r, 3 - r) \\
(r, 3 - r) & (1 + r, 1 - r) & (2 + 2r, 3 - 2r)
\end{pmatrix}.
\]
which is exactly the answer of the equation.

**Example 3.2.** Let

\[
A_1 = \begin{pmatrix} 0 & 2 & -1 \\ 0 & 0 & -1 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 3 & 1 & 4 \\ 0 & 0 & 0 \end{pmatrix}, \quad A_3 = \begin{pmatrix} -2 & 1 & 2 \\ 0 & 0 & 0 \end{pmatrix},
\]

\[
B_1 = \begin{pmatrix} 3 & 1 \\ 0 & 2 \end{pmatrix}, \quad B_2 = \begin{pmatrix} 1 & -1 \\ -1 & 0 \end{pmatrix}
\]

and \(\bar{C} = ((-28 + 54r, 29 - 48r), (-38 + 43r, 16 - 52r))\), then we have an equation as the following:

\[
\sum_{k=1}^{3} A_k \bar{X} B_k
\]

\[
= ((-28 + 54r, 29 - 48r), (-38 + 43r, 16 - 52r)).
\]

According to our method, we obtain the following crisp linear matrix equation:

\[
\begin{pmatrix} 0 & 2 & 0 \\ 0 & 0 & -1 \end{pmatrix} (\bar{X} \bar{X}) \begin{pmatrix} 3 & 1 \\ 0 & 2 \\ 0 & 0 \end{pmatrix} + \\
\begin{pmatrix} 0 & 0 & 1 \\ 0 & -2 & 0 \end{pmatrix} (\bar{X} \bar{X}) \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & -1 \end{pmatrix} + \\
\begin{pmatrix} 3 & 1 & 4 \\ 0 & 0 & 0 \end{pmatrix} (\bar{X} \bar{X}) \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & -1 \end{pmatrix} + \\
\begin{pmatrix} 0 & 0 & 0 \\ -3 & -1 & -4 \end{pmatrix} (\bar{X} \bar{X}) \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ -1 & 0 \end{pmatrix} + \\
\begin{pmatrix} 0 & 1 & 2 \\ -2 & 0 & 0 \end{pmatrix} (\bar{X} \bar{X}) \begin{pmatrix} 2 & 2 \\ 0 & 0 \\ -1 & -2 \end{pmatrix} + \\
\begin{pmatrix} 2 & 0 & 0 \\ 0 & -1 & -2 \end{pmatrix} (\bar{X} \bar{X}) \begin{pmatrix} 0 & 0 \\ 1 & 2 \\ -2 & -2 \end{pmatrix} = \\
\begin{pmatrix} -28 + 54r & -38 + 43r \\ 29 - 48r & 16 - 52r \end{pmatrix}
\]
After calculation, we obtain that

\[
X = \begin{pmatrix}
  r & 2 + r \\
  2r & -1 + r \\
  2 + r & 2 + 2r
\end{pmatrix}
\text{ and } \bar{X} = \begin{pmatrix}
  1 - r & 3 - 2r \\
  2 - r & 1 - 2r \\
  3 - r & 4 - r
\end{pmatrix}.
\]

So the unknown fuzzy matrix is

\[
\bar{X} = \begin{pmatrix}
  (r, 1 - r) & (2 + r, 3 - 2r) \\
  (2r, 2 - r) & (-1 + r, 1 - 2r) \\
  (2 + r, 3 - r) & (2 + 2r, 4 - r)
\end{pmatrix}.
\]

By the foregoing discussion, we’ve learnt the way to transform a fuzzy linear matrix equations into a crisp linear matrix equations to solve the classic solution. We find that, however, the method may lose efficiency when the scale of (3) is large. Because it actually doubles the scale in the transformation. To fix the problem, we will improve the method further so that it can be performed better in the case of the large scale of (3). By diagonalizing the right hand side of (11), we have

\[
\frac{1}{\sqrt{2}} \begin{pmatrix}
  I & I \\
  I & -I
\end{pmatrix}
\begin{pmatrix}
  C & \overline{C} \\
  \overline{C} & C
\end{pmatrix}
\begin{pmatrix}
  I & I \\
  I & -I
\end{pmatrix}
\frac{1}{\sqrt{2}} = \begin{pmatrix}
  \frac{1}{2} (C + \overline{C}) & 0 \\
  0 & \frac{1}{2} (C - \overline{C})
\end{pmatrix}.
\]

For the left side of (11), we diagonalize every matrix in the same way. Since

\[
\frac{1}{\sqrt{2}} \begin{pmatrix}
  I & I \\
  I & -I
\end{pmatrix} \frac{1}{\sqrt{2}} = I_{2n},
\]

the left hand side becomes

\[
\sum_{k=1}^{l} \begin{pmatrix}
  A_k^+ - A_k^- & 0 \\
  0 & A_k^+ + A_k^-
\end{pmatrix}
\begin{pmatrix}
  X + \overline{X} & 0 \\
  0 & X - \overline{X}
\end{pmatrix}
\begin{pmatrix}
  B_k^+ - B_k^- & 0 \\
  0 & B_k^+ + B_k^-
\end{pmatrix}.
\]

Thus, the equation (11) becomes

\[
= \sum_{k=1}^{l} \begin{pmatrix}
  A_k & 0 \\
  0 & |A_k|
\end{pmatrix}
\begin{pmatrix}
  X + \overline{X} & 0 \\
  0 & X - \overline{X}
\end{pmatrix}
\begin{pmatrix}
  B_k & 0 \\
  0 & |B_k|
\end{pmatrix}
= \begin{pmatrix}
  C + \overline{C} & 0 \\
  0 & C - \overline{C}
\end{pmatrix}.
\]

9
So (18) can be divided into two crisp linear matrix equations with the scale of (3) as the following:

\[
\sum_{k=1}^{t} A_k Y_1 B_k = C_1 \tag{19}
\]
and

\[
\sum_{k=1}^{t} |A_k| Y_2 |B_k| = C_2, \tag{20}
\]
where \( C_1 = \bar{C} + C \) and \( C_2 = \bar{C} - C \). Obviously, we can obtain \( \bar{X}, \overline{X} \) by doing the following operations:

\[
\bar{X} = \frac{1}{2}(Y_1 + Y_2),
\]
and

\[
\overline{X} = \frac{1}{2}(Y_1 - Y_2).
\]
This method makes computation of large scale fuzzy linear matrix equation feasible. Then, we can conclude the framework of the method to solve (3) as follows.

**Algorithm 3.1. Extension Method (EM)**

1. Calculate \( |A_k|, |B_k|, C_1 \) and \( C_2 \) in (19) and (20);
2. Solve (19) and (20) to derive \( Y_1 \) and \( Y_2 \);
3. Obtain \( \bar{X} = \frac{1}{2}(Y_1 + Y_2) \) and \( \overline{X} = \frac{1}{2}(Y_1 - Y_2) \).

Obviously, The Extension Method here can provide a uniform framework to solve fuzzy linear matrix equation.

### 4 Existence problem

In this section, we will discuss conditions of the classic solution existence. By (8) and (9), we have

\[
\bar{C} - C = \sum_{k=1}^{t} A_k^+(\bar{X} - \overline{X})(B_k^+ + B_k^-) + \sum_{k=1}^{t} A_k^-(\bar{X} - \overline{X})(B_k^+ + B_k^-) \tag{21}
\]

\[
= \sum_{k=1}^{t} (A_k^+ + A_k^-)(\bar{X} - \overline{X})(B_k^+ + B_k^-).
\]
Let \( \Delta C = \bar{C} - C \) and \( \Delta X = \bar{X} - X \), then (21) becomes:

\[
\Delta C = \sum_{k=1}^{l} |A_k| \Delta X |B_k|.
\]  (22)

Noting that \( \Delta X \) has to be a nonnegative matrix, which means every element in \( \Delta X \) should be nonnegative.

We can transfer (22) to a linear equations by using kronecker product.

\[
\sum_{k=1}^{l} (|B_k|^T \otimes |A_k|) V ec(\Delta X) = V ec(\Delta C).
\]

When \( n = r \) and \( s = m \), all the matrices \( A_k \) and \( B_k \) are square matrices. Obviously, matrices \( |B_k|^T \otimes |A_k| \) are also square matrices.

Let \( \lambda_1^{(k)}, \lambda_2^{(k)}, \ldots, \lambda_n^{(k)} \) be the eigenvalues of matrices \( |A_k| \) and \( \mu_1^{(k)}, \mu_2^{(k)}, \ldots, \mu_m^{(k)} \) be the eigenvalues of matrix \( |B_k| \). According to the property of Kronecker product, for every \( |B_k|^T \otimes |A_k| \) will have \( mn \) eigenvalues with the form such that \( \lambda_i^{(k)} \mu_j^{(k)} (i = 1, 2, \ldots, n; j = 1, 2, \ldots, m) \). When \( \sum_{k=1}^{l} \lambda_i^{(k)} \mu_j^{(k)} \neq 0 \) (\( i = 1, 2, \ldots, n; j = 1, 2, \ldots, m \)), \( \sum_{k=1}^{l} (|B_k|^T \otimes |A_k|) \) is nonsingular. So we have

\[
V ec(\Delta X) = (\sum_{k=1}^{l} (|B_k|^T \otimes |A_k|))^{-1} V ec(\Delta C).
\]

**Lemma 4.1.** [13] The inverse of a nonnegative matrix \( A \) is nonnegative if and only if \( A \) is a generalized permutation matrix.

As we know, a matrix that has the same zero pattern as a permutation matrix is called generalized permutation matrix.

Let \( H = \sum_{k=1}^{l} (|B_k|^T \otimes |A_k|) \), then in the following case, \( H \) can be easily identified as a generalized permutation matrix:

**Theorem 4.1.** Let \( |A_k| = (|a_{ij}^{(k)}|)_{n \times n} \geq 0, |B_k| = (|b_{ij}^{(k)}|)_{n \times n} \geq 0, \) and \( H = \sum_{k=1}^{l} (|B_k|^T \otimes |A_k|) \). Then \( H \) is nonsingular and \( H^{-1} \geq 0 \) if for all \( k = 1, 2, \ldots, l \),

1. \( A_k \) and \( B_k \) are generalized permutation matrices;
2. nonzero entries of every \( A_k \) are in the same positions;
3. nonzero entries of every \( B_k \) are in the same positions.

**Proof.** Since \( |A_k| \) and \( |B_k| \) are nonnegative generalized permutation matrices, \( |B_k|^T \otimes |A_k|, k = 1, 2, \ldots, l \), are also generalized nonnegative permutation matrices. Since all of \( A_k, k = 1, 2, \ldots, l \) have same zero pattern and so do all of \( B_k \), \( H \) is a generalized nonnegative permutation matrices.

Then according to lemma 4.1, \( H \) is nonsingular and \( H^{-1} \geq 0 \). 

\[ \square \]
Clearly, Theorem 3.2 in [8] is the special case of this theorem. Conditions in Theorem 4.1 are sufficient, they can not be a necessary conditions, please see the following example.

Let \( l = 2 \), and \( E_1 = E_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \), \( D_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \), and \( D_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \). Then, we have

\[
H = \sum_{k=1}^{2} (|B_k|^T \otimes |A_k|) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}.
\]

We find that although \( D_1 \) and \( D_2 \) are not generalized permutation matrix, \( H \) has one and only one positive number in every row and column. So conditions in theorem 4.1 are not necessary for \( H \) to be a generalized permutation matrix, which makes \( H^{-1} \) a nonnegative matrix.

5 Numerical solution of fuzzy general Lyapunov equation

In this section, we will develop a numerical method to solve fuzzy generalized Lyapunov matrix equations. Fuzzy generalized Lyapunov matrix equations have the following form:

\[
A\bar{X} + \bar{X}B + \sum_{k=1}^{l} F_k\bar{X}F_k^T = \bar{C}.
\]

The equation is a special case of (3). We can transform it into the following two crisp generalized Lyapunov matrix equations by the Extension Method:

\[
AY_1 + Y_1B + \sum_{k=1}^{l} F_kY_1F_k^T = C_1,
\]

and

\[
|A|Y_2 + Y_2|B| + \sum_{k=1}^{l} |F_k|Y_2|F_k|^T = C_2.
\]

where \( C_1 = \bar{C} + \bar{C} \) and \( C_2 = \bar{C} - \bar{C} \). Since both (24) and (25) are crisp generalized Lyapunov equations. A stationary iterative method can be applied to them. For any initial guess \( X^{(0)} \), we do the following iteration:

\[
AY_1^{(m+1)} + Y_1^{(m+1)}B = - \sum_{k=1}^{l} F_kY_1^{(m)}F_k^T + C_1,
\]
and
\[
|A|^2_2 + Y_2^{(m+1)}|B| = -\sum_{k=1}^l |F_k|Y_2^{(m)}|F_k|^T + C_2. \tag{27}
\]

When \(F_k, k = 1, \cdots, l\) are low rank matrices, the iterative method converges [16]. At every iteration, we can use one of Lyapunov matrix equation solvers [18] to solve (26) and 27. Thus, we have the algorithm 5.1 to solve the classic solution of the fuzzy generalized Lyapunov matrix equations.

**Algorithm 5.1.** Solving fuzzy generalized Lyapunov matrix equation with iteration

1. For given \(A, B, F_1, \cdots, F_l,\) and \(X_0,\) compute \(a = \|A\|_F + \|B\|_F + \sum_{k=1}^l \|F_k\|_F^2.\)
2. \(m = 0\) to itermax do:
   (a) solve (26) and (27);
   (b) compute \(R_1^{(m+1)}\) and \(R_2^{(m+1)}\) which are residuals of (26) and (27), respectively;
   (c) If \(\frac{\|R_1^{(m+1)}\|_a}{a} < \text{tolerance}\) and \(\frac{\|R_2^{(m+1)}\|_a}{a} < \text{tolerance}\)
   (d) \(X = \frac{1}{2}\left(Y_1^{(m+1)} + Y_2^{(m+1)}\right)\) and \(\bar{X} = \frac{1}{2}\left(Y_1^{(m+1)} - Y_2^{(m+1)}\right);\)
   (e) Break;
   (f) else do:
   (g) \(Y_1^{(m)} = Y_1^{(m+1)}\) and \(Y_2^{(m)} = Y_2^{(m+1)}\)
3. end if
4. end for

In practical implementation of this algorithm, If the scale of (3) is small, we can employ the direct method developed in [1] to solve (26) and (27). If the scale of (3) is large, we can employ iterative methods in [14, 17] to solve (26) and (27).

In order to show the validation and efficiency, please see the following numerical tests.

**Example 5.1.** In this example, we consider the following example:
\[
A\bar{X} + \bar{X}B + F_1\bar{X}F_1^T + F_2\bar{X}F_2^T = \bar{C}, \tag{28}
\]
Where \(A = MM^T\) and \(B = SS^T.\) Here, \(M\) and \(S\) are two \(n \times n\) matrices generated randomly. Let \(F_1 = f_{11}f_{12}^T\) is a low rank sparse matrix, where \(f_{11}\) and \(f_{12}\) are generalized by two low rank \(n \times 3\) sparse matrices created by function sprand() in MATLAB, so is \(F_2.\) \(\bar{C} = (1 + 2r, 5 - 2r) * \text{ones}(n),\) where \(\text{ones}(n)\) is an \(n \times n\) matrix whose entries are all one.
The computer we use to test the algorithm have physical memory of 32GB, and has Intel(R) Core(TM) i7-10875H CPU @2.30GHz to do the calculation. We test this example 1000 times for n=10, 100, 150, and 200, we record the times of the classic solution existing in Table 1.

To show the efficiency of this algorithm, we will compare the Algorithm 5.1 with the method that transfers (28) to a fuzzy linear equations by using Kronecker Product, and the transferred fuzzy linear equations is computed by the method in [7]. In the rest of this paper, we call this method the Kron method. We compare the average running time of the two methods with computing (28) 30 times in Table 2.

By this table, we find that the performance of Algorithm 5.1 is much better than that of the Kron method. Especially, when n is bigger than 150, we found that the computer’s memory, which is 32GB, is too small to implement the Kron method.

6 conclusion

The general form of fuzzy linear matrix equations covers many types of fuzzy matrix equations. Here, an extension method to solve its classic solution is presented. Conditions of the classic solution existence are studied. Thus, we have a uniform method to solve fuzzy linear matrix equations. Under the framework, we present a numerical method to solve fuzzy generalized Lyapunov matrix equations. Selected examples and numerical tests are shown to illustrate the validation and efficiency.

Declaration:

**Funding:** This study was funded by National Natural Science Foundation of China under grant NSF12071088.

**Conflict of Interest:** All authors of this paper have no conflict of interest.

**Ethical approval:** This article does not contain any studies with human participants or animals performed by any of the authors.

**Author Contributions:** The idea of this paper comes from our seminar. The methodology of this paper is done by the second author and the third author. The numerical experiment is done by the first author. The writing of this paper is done by the second author.

References

[1] Bartels, R. and Stewart, G. *Solution of the matrix equation AX + XB = C*, Communications of the ACM, 1972

[2] Peter Benner and Tobias Breiten, *Low rank methods for a class of generalized Lyapunov equations and related issues*, Numerische Mathematik, 124:441-470, 2013.
[3] James J. Buckley, Esfandiar Eslami and Thomas Feuring, *Fuzzy Mathematics in Economics and Engineering*, Studies in Fuzziness and Soft Computing, Springer-Verlag Berlin Heidelberg GmbH, 2002.

[4] Tobias Damm, *Direct methods and ADI-preconditioned Krylov subspace methods for generalized Lyapunov equations*, Numerical Linear Algebra with Applications, 15:853-871, 2008.

[5] Tobias Damm and Diederich Hinrichsen, *Newton's method for a rational matrix equation occurring in stochastic control*, Linear Algebra and its Applications, 332-334:81-109, 2001.

[6] K. Dookhitram, R. Lollchund, R. K. Tripathi and M. Bhuruth, *Fully fuzzy Sylvester matrix equation*, J. Intell. Fuzzy Syst. 28 (2015) 21992211.

[7] Friedman, M.; Ming, M. and Kandel, A, *Fuzzy linear systems*, Fuzzy Sets and Systems, 1998, 96, 201-209

[8] Q. He, L. Hou and J. Zhou, *The solution of fuzzy Sylvester matrix equation*, Soft Comput. (2018) 22, pp.65156523.

[9] Hou, L., Zhou, J. and He, Q. *An extension method for fully fuzzy Sylvester matrix equation*. Soft Comput (2021). https://doi.org/10.1007/s00500-021-05573-z

[10] X.-B. Guo, *Approximate solution of fuzzy sylvester matrix equations*, in: 2011 Seventh International Conference on Computational Intelligence and Security, IEEE, 2011, pp. 5256.

[11] Kleinman DL, *On the stability of linear stochastic systems*, IEEE Transactions on Automatic Control, 1969, AC14, pp.429-430.

[12] Peter Lancaster, *Explicit Solutions of Linear Matrix Equations*, SIAM Review, 1970.

[13] Minc. H *Nonnegative matrices*. Wiley, New York, 1988.

[14] T. Penzl, *A cyclic low-rank Smith method for large sparse Lyapunov equations*. SIAM J. Sci. Comput., 21(4):14011418, 2000

[15] D. K. Salkuyeh, *on the solution of the fuzzy Sylvester matrix equation*, Soft Comput. 15 (5) (2011) 953961.

[16] Shank S. D., Simoncini V. and Szyld D. B., *Efficient low-rank solution of generalized Lyapunov equations*, Numerische Mathematik, 2016, 134(2):327-342.

[17] Article (Simoncini.V2007) Simoncini, V, *A new iterative method for solving large-scale Lyapunov matrix equations* SIAM J. Scient. Computing, 2007, 29(3):1268-1288.
[18] Simoncini V, *Computational methods for linear matrix equations*, SIAM Review, SIAM, 2016, 58, 377-441
Table 1: The times of the classic solution existing

| n  | The times of the classic solution existing |
|----|--------------------------------------------|
| 10 | 235                                        |
| 100| 148                                        |
| 150| 166                                        |
| 200| 159                                        |
| 400| 138                                        |

Table 2: The running condition with different matrix size $n$

| n  | average Running time of Algorithm 5.1 | average Running time by Kron |
|----|--------------------------------------|-------------------------------|
| 100| 0.69                                 | 3.17                          |
| 150| 1.41                                 | 18.14                         |
| 200| 2.30                                 | NULL                          |
| 400| 12.32                                | NULL                          |
| 800| 55.25                                | NULL                          |
| 1000| 93.83                               | NULL                          |
| 2000| 734.98                               | NULL                          |