Fundamental Journal of Mathematics and Applications, 6 (1) (2023) 42-50
Research Article

Fundamental Journal of Mathematics and Applications

Journal Homepage: www.dergipark.org.tr/en/pub/fujma
ISSN: 2645-8845
doi: https://dx.doi.org/10.33401/fujma.1195121

Approximate Fuzzy Inverse Matrix Calculation Method using Scenario-based Inverses and Bisection

Hande Günay Akdemir

Department of Mathematics, Faculty of Science and Arts, Giresun University, Giresun, Türkiye

Article Info

Keywords: *Approximate fuzzy inverse, Bisection method, Fully fuzzy linear system of equations, Fuzzy matrix, Fuzzy numerical simulation*
2010 AMS: *03E72, 15B15, 37M05, 90C59*
Received: *26 October 2022*
Accepted: *22 December 2022*
Available online: *9 January 2023*

Abstract

In this paper, we introduce a numerical method to construct the inverse of a square matrix whose elements are trapezoidal or triangular fuzzy numbers (FNs). A set of fuzzy linear equations is required to be solved in order to determine the fuzzy inverse matrix. The proposed technique first iteratively searches the possible solution intervals and then narrows those too-wide estimated intervals via bisection. Using interval arithmetic in left and right matrix multiplication, we aim to approximate the identity matrix as a result of product operations. The dissimilarity of the endpoints of intervals belonging to multiplication matrices with the identity matrix is considered to be an error function to be minimized. In this way, even if the entries of a matrix are uncertain, the fuzzy inverse matrix containing all inverse matrices can be found quickly with the use of computer technology. The method is explained, and comparisons are drawn with inverse stable examples from the literature.

1. Introduction

Real-world models are too complicated to be presented with accurate values of parameters since precise knowledge is often unavailable. To overcome imprecision, we may substitute crisp numbers with FNs provided by experts. For the purpose of solving fully fuzzy linear systems of equations and fully fuzzy matrix equations, an analysis of the fuzzy inverse matrix is beneficial. The narrowest family of matrices that contain all the inverses is known as the inverse of a given fuzzy matrix, whose entries are FNs. Notice that interval and fuzzy matrices are connected since each $\alpha-$cut of a FN is represented by an interval.

Chanas and Nowakowski [1] developed the fuzzy numerical simulation (FNS) technique that allows one to assign an exact numerical value to a fuzzy variable by generating two values from a uniform distribution. Rohn [2] examined the invertibility of interval matrices by providing necessary and sufficient conditions. For the solution of simultaneous linear equations with fuzzy parameters, Rao and Chen [3] provided a computational methodology based on a search algorithm and bisection method. The idea of finding the inverse matrix using scenario-based inverse matrices was first proposed by Dehghan et al. [4]. Basaran [5] presented a method to calculate the inverse of a fuzzy matrix that consists of $LR-$type FNs. Refer to [7] and [8] for suggested alterations and revisions for [5]. Mosleh [6] used fuzzy neural networks with fuzzy weights for fuzzy inputs and fuzzy targets to find approximate solutions to fully fuzzy matrix equations, which can be utilized to determine fuzzy inverse matrices. Guo et al. [9] designed a numerical procedure to derive the fuzzy inverse of the matrix with $LR-$type FNs and supplied a sufficient condition for its existence. Chen and Huang [10] proposed a mathematical programming model to acquire the fuzzy weights of the fuzzy analytical network process by utilizing the fuzzy inverse matrix on the basis of the criterion of the minimum spread of FNs. Babakordi and Taghi-Nezhad [11] also employed linear programming for the calculation of fuzzy inverse matrix. Farahani and Ebadi [12] discussed the sufficient and necessary conditions for the invertibility of a fuzzy matrix by analyzing a system of fuzzy polynomial equations. Akdemir and Kocken [13] proposed an FNS-based bisection procedure

Email address and ORCID number: hande.akdemir@giresun.edu.tr, 0000-0003-3241-1560

for a fuzzy linear regression model with crisp inputs and fuzzy outputs.

Fuzzy matrices whose entries are symmetrical triangular FNs have been considered mostly in the literature. As far as we know, studies on inverse calculations of fuzzy square matrices with asymmetrical and/or trapezoidal fuzzy entries are scarce. Some attempts using approximate operations had high errors. In some cases, inverse stable matrices were not considered. The motivation for this study is to contribute to the literature by proposing a computational technique making use of crisp scenario-based inverses and the bisection method. In this paper, we suggest a new method that first solves a large number of possible equation scenarios and then attains a very large range of solutions. After that, it attempts to reduce the error by narrowing these initial ranges via bisection. For a given square matrix with fuzzy entries, parameters are repeatedly generated via the single-value FNS method, thus the inverses of crisp scenario-based matrices are calculated too many times, different from the approach of Dehghan et al. [4]. This way, we can estimate the intervals in which the inverses are located. But the fuzzinesses of the solutions, namely the lengths of the corresponding ranges of solutions, are too large at the beginning since an excessive number of scenarios is affecting the estimation. After the left and right matrix multiplication, we aim to approximate fuzzy units or fuzzy zeros by increasing the similarity of both sides' endpoints. The fuzzy identity matrix is considered to be a nearly crisp identity matrix.

The rest of this paper is constructed as follows: In the following section, we give fundamental definitions of fuzzy set theory along with the notation we use. In Section 3, we develop a numerical method for the determination of the approximate fuzzy inverse matrix. In Section 4, we provide numerical examples to illustrate our methodology. We conclude in Section 5.

2. Preliminaries

In this section, we review some basic concepts of fuzzy set theory that will be used in the subsequent sections.

Definition 2.1. *A standard fuzzy set $\widetilde{a}$ on the universe of discourse X is defined as*

$$\widetilde{a} = \{ \langle x, \mu_{\widetilde{a}}(x) \rangle | x \in X \}$$

where the membership function $\mu_{\widetilde{a}} : X \to [0,1]$ denotes the grade of belonging of the element x to the set $\widetilde{a}$.

A normal fuzzy subset of the real line, i.e., there exists $\exists x \in \mathbb{R}$ such that $\mu(x) = 1$, with a convex membership function is called a FN.

Definition 2.2 ($LR-$type FN)**.** *Let $L,\ R : \mathbb{R}^+ \to [0,1]$ be decreasing functions with $L(0) = 1,\ R(0) = 1,\ L(1) = 0,\ R(1) = 0,\ L(x),\ R(x) < 1$ for $x > 0$. The FN $\widetilde{a}$ is called $LR-$type if there exist $m_L, m_R\ (m_L < m_R), \alpha > 0$ and $\beta > 0$, with the membership function:*

$$\mu_{\widetilde{a}}(x) = \begin{cases} L\left(\frac{m_L - x}{\alpha}\right), & \text{if } x < m_L \\ 1, & \text{if } m_L \le x < m_R \\ R\left(\frac{x - m_R}{\beta}\right), & \text{if } m_R \le x \end{cases}$$

where α, β are called left spread and right spread, respectively. The lengths of the lower and upper ranges α and β are also known as left fuzziness and right fuzziness. Then, we denote $\widetilde{a} = (m_L, m_R, \alpha, \beta)_{LR}$. Also, the endpoints $\widetilde{a}_L = m_L - \alpha$ and $\widetilde{a}_R = m_R + \beta$ are termed as lower and upper bounds, respectively. The interval $[m_L, m_R]$ is additionally referred to as mode interval.

If $L(x) = R(x) = \max\{0, 1-x\}$, then $\widetilde{a}$ is called a Trapezoidal FN. Similarly, if $L(x) = R(x) = \max\{0, 1-x\}$ and $m_L = m_R$, then we get a Triangular FN. Also, sets consisting of these two types of numbers are closed under the operations given in the definition below.

Definition 2.3. *Let $\widetilde{a_1} = (m_L, m_R, \alpha_1, \beta_1)_{LR}$ and $\widetilde{a_2} = (n_L, n_R, \alpha_2, \beta_2)_{LR}$ be two trapezoidal (or triangular) FNs, and $\lambda \in \mathbb{R}$, then we have:*

- *Addition*

$$\widetilde{a_1} + \widetilde{a_2} = (m_L + n_L, m_R + n_R, \alpha_1 + \alpha_2, \beta_1 + \beta_2)_{LR},$$

- *Scalar Multiplication*

$$\lambda \widetilde{a_1} = \begin{cases} (\lambda m_L, \lambda m_R, \lambda \alpha_1, \lambda \beta_1)_{LR}, & \text{if } \lambda \ge 0 \\ (\lambda m_R, \lambda m_L, -\lambda \beta_1, -\lambda \alpha_1)_{LR}, & \text{if } \lambda < 0 \end{cases}$$

- *Multiplication*

$$\widetilde{a_1} \times \widetilde{a_2} = (l, r, \alpha, \beta)_{LR}, \tag{2.1}$$

where the endpoints of the mode interval and bounds are

$$\begin{aligned} l &= \min\{m_L n_L, m_L n_R, m_R n_L, m_R n_R\}, \\ r &= \max\{m_L n_L, m_L n_R, m_R n_L, m_R n_R\}, \\ l-\alpha &= \min\{(m_L-\alpha_1)(n_L-\alpha_2), (m_L-\alpha_1)(n_R+\beta_2), (m_R+\beta_1)(n_L-\alpha_2), (m_R+\beta_1)(n_R+\beta_2)\}, \\ r+\beta &= \max\{(m_L-\alpha_1)(n_L-\alpha_2), (m_L-\alpha_1)(n_R+\beta_2), (m_R+\beta_1)(n_L-\alpha_2), (m_R+\beta_1)(n_R+\beta_2)\}, \end{aligned}$$

respectively.

Definition 2.4. *[5] A FN $\widetilde{1} = (1,1,\varepsilon,\delta)_{LR}$ is called fuzzy unit if the spreads $\varepsilon, \delta > 0$ are sufficiently small.*

Definition 2.5. *[5] A FN $\widetilde{0} = (0,0,\rho,\sigma)_{LR}$ is called fuzzy zero if the spreads $\rho, \sigma > 0$ are sufficiently small.*

Definition 2.6. *[5] A fuzzy matrix is a fuzzy identity matrix if its main diagonal and remaining elements are respectively fuzzy units and zeros. It is denoted by $\widetilde{I}$ as follows:*

$$\widetilde{I} = \begin{bmatrix} \widetilde{1} & \widetilde{0} & \dots & \widetilde{0} \\ \widetilde{0} & \widetilde{1} & \dots & \widetilde{0} \\ \vdots & \ddots & \ddots & \vdots \\ \widetilde{0} & \widetilde{0} & \dots & \widetilde{1} \end{bmatrix}.$$

3. Proposed method

We attempt to determine the approximate fuzzy (multiplicative) inverse of a square matrix whose parameters are all arbitrary trapezoidal (or triangular) FNs, i.e., not necessarily non-negative. Consider a square fuzzy matrix $\widetilde{A}$, represented by mutually independent $LR-$type fuzzy entries. If there exists an $LR-$type fuzzy matrix $\widetilde{X}$ whose entries are unknown trapezoidal (or triangular) FNs, such that:

$$\widetilde{A} \times \widetilde{X} = \widetilde{X} \times \widetilde{A} = \widetilde{I},$$

we refer the fuzzy matrix $\widetilde{X}$ is the inverse of the fuzzy matrix $\widetilde{A}$ and denote it by $\widetilde{A}^{-1} = \widetilde{X}$. We apply the interval arithmetic-based equation, Equation (2.1) for multiplication.

First, let us consider the right inverse matrix via $\widetilde{A} \times \widetilde{X} = \widetilde{I}$. For the $LR-$type fuzzy entries $\widetilde{a_{ij}} = \left(m_{ij}^L, m_{ij}^R, \alpha_{ij}, \beta_{ij}\right)_{LR}$ and $LR-$type fuzzy unknowns $\widetilde{x_{jk}} = \left(n_{jk}^L, n_{jk}^R, \tau_{jk}, \phi_{jk}\right)_{LR}$, the following fully fuzzy system of linear equations occurred:

$$\begin{aligned} \sum_{j=1}^{n} \widetilde{a_{ij}} \times \widetilde{x_{jk}} &= \sum_{j=1}^{n} \left(m_{ij}^L, m_{ij}^R, \alpha_{ij}, \beta_{ij}\right)_{LR} \times \left(n_{jk}^L, n_{jk}^R, \tau_{jk}, \phi_{jk}\right)_{LR} \\ &= \begin{cases} (1,1,\varepsilon_{ik},\delta_{ik})_{LR}, & \text{if } i = k \\ (0,0,\rho_{ik},\sigma_{ik})_{LR}, & \text{otherwise} \end{cases} \end{aligned}$$

where $i,k = 1,2,...,n$. Similarly, for the left inverse matrix, the following conditions:

$$\sum_{j=1}^{n} \widetilde{x_{ij}} \times \widetilde{a_{jk}} = \begin{cases} \widetilde{1}, & \text{if } i = k \\ \widetilde{0}, & \text{otherwise} \end{cases}$$

must hold for $i,k = 1,2,...,n$. Each system is $n \times n$ by $n \times n$ since there are $n \times n$ unknowns and $n \times n$ equations in total.

The error function we have considered in order to produce close matrices to the fuzzy identity matrix as a result of both left and right multiplication is given as follows:

$$error = \sum_{i=1}^{n} \sum_{k=1}^{n} \sum_{r \in \{R,L\}} \left(e_{ikr}^R + e_{ikr}^L\right) \tag{3.1}$$

where

$$e_{ikr}^{R}=\begin{cases}\left|\left(\sum_{j=1}^{n}\widetilde{a_{ij}}\times\widetilde{x_{jk}}\right)_{r}-1\right|, & \text{if } i=k\\ \left|\left(\sum_{j=1}^{n}\widetilde{a_{ij}}\times\widetilde{x_{jk}}\right)_{r}-0\right|, & \text{otherwise}\end{cases} \tag{3.2}$$

and

$$e_{ikr}^{L}=\begin{cases}\left|\left(\sum_{j=1}^{n}\widetilde{x_{ij}}\times\widetilde{a_{jk}}\right)_{r}-1\right|, & \text{if } i=k\\ \left|\left(\sum_{j=1}^{n}\widetilde{x_{ij}}\times\widetilde{a_{jk}}\right)_{r}-0\right|, & \text{otherwise.}\end{cases} \tag{3.3}$$

Four separate sorts of errors e_{ikR}^{R}, e_{ikL}^{R}, e_{ikR}^{L}, and e_{ikL}^{L} arise for each entry (i,k) of multiplication matrices. The first two of them are a result of the difference between the upper and lower endpoints of the multiplication entry from the right and the corresponding element of the identity matrix. Similarly, the last two of them correspond to left-hand multiplication. In addition, if the spreads ε,δ,ρ, and σ of fuzzy units and zeros are specified as in [5], the error functions (3.2) and (3.3) can be revised accordingly. See also the definition of $\varepsilon-$inverse in [4].
Our approach consists of two phases. At the first one, the lower and upper bounds of potential inverse matrices are searched, and then the absolute error values are calculated. The bisection algorithm is then applied, starting with the spread with the largest total error $\left(e_{ikr}^{R}+e_{ikr}^{L}\right)$ which belongs to the entry (i,k) having the highest r fuzziness, such that $r=R$ and $r=L$ correspond to right and left, respectively. With a specified number of bisection iterations, the original spread is narrowed to a specific extent depending on whether or not the overall error value decreases. The opted narrowing gives a minimum overall error value, amongst others. The bisection process is then performed for the spread with the second-largest inaccuracy. When all spreads' errors are decreased, the process is finished. Let us now explain those two phases of our method separately in Subsections 3.1 and 3.2.

3.1. Search algorithm based on FNS

The foundation of our search algorithm is based on the generation of single values of FNs. We can estimate the intervals in which the inverse entries are located by computing the inverses of crisp matrices, which are obtained by substituting the generated values of the fuzzy parameters. It is obvious that this process needs to be repeated too many times in order to yield meaningful results. The literature contains a wide variety of efficient crisp techniques for calculating inverses or pseudo-inverses. We therefore try to employ these non-fuzzy methods to offer more accurate estimates. To generate given $LR-$type fuzzy entry $\widetilde{a_{ij}}=\left(m_{ij}^{L},m_{ij}^{R},\alpha_{ij},\beta_{ij}\right)_{LR}$, we first simulate two independent uniformly distributed random numbers, u_{ij} and v_{ij}. The weighted arithmetic mean of the endpoints of $u_{ij}-$cut interval is a sample of $\widetilde{a_{ij}}$ as follows:

$$\overline{a_{ij}}=v_{ij}\left[m_{ij}^{L}-\alpha_{ij}L^{-1}\left(u_{ij}\right)\right]+\left(1-v_{ij}\right)\left[\beta_{ij}R^{-1}\left(u_{ij}\right)+m_{ij}^{R}\right] \tag{3.4}$$

where v_{ij} and $(1-v_{ij})$ are considered as weights for $i,j=1,2,...,n$.
The algorithm initially calculates 1-cut inverse matrices by Rohn's method [2], if any. If these inverses are inconsistent, the process is terminated. To find the lower and upper bounds of the inverse matrix's entries, the process first generates single values of the fuzzy parameters and repeatedly computes the inverses of the resulting crisp matrices. For any entry, it is regarded as a potential value for the lower (upper) bound if the value is less (greater) than the left (right) endpoint of the mode interval of the 1-cut inverse. After a large number of endpoints are determined and then averaged, the infimum and supremum of these mean values are used to derive the right and left endpoints, respectively. These bounds, as well as errors, which are too big initially, are required for the execution of the second phase, namely the bisection algorithm. Refer to Algorithm 1 for this phase of our method. Algorithm 1 is a similar process to the search algorithm in [3]. The sequence of the simulation repetitions is represented by the iteration counters s and t, which are integers from 1 to S and 1 to T, respectively.

3.2. Bisection algorithm

The bisection algorithm is applied to the spreads in accordance with the order in which the total errors $\left(e_{ikr}^{R}+e_{ikr}^{L}\right)$, which are obtained at the end of Algorithm 1, are listed in descending order. A one-to-one mapping between the total errors and unknowns is considered. For example, an error $\left(e_{ikr}^{R}+e_{ikr}^{L}\right)$ leads to an update in variable $\left(\widetilde{x_{jk}}\right)_{r}$ where $j=i$. The fuzzy matrix

Algorithm 1 FNS-based Search Algorithm

INPUT: $LR-$type fuzzy entries $\widetilde{a_{ij}} = \left(m_{ij}^L, m_{ij}^R, \alpha_{ij}, \beta_{ij}\right)_{LR}$ where $i, j = 1,2,...,n$, and large positive integers T, S

OUTPUT: $LR-$type fuzzy unknowns $\widetilde{x_{jk}} = \left(n_{jk}^L, n_{jk}^R, \tau_{jk}, \phi_{jk}\right)_{LR}$ and errors e_{ikR}^R, e_{ikL}^R, e_{ikR}^L, and e_{ikL}^L where $i, j, k = 1,2,...,n$

1: **procedure** FNSBASEDSEARCH
2: Compute the inverse matrix of the interval matrix that accepts mode intervals $\left[m_{ij}^L, m_{ij}^R\right]$ as entries by using Rohn's method. Continue if the solutions are consistent, namely $n_{jk}^L \leq n_{jk}^R$ for all $j,k = 1,2,...,n$. Otherwise, report that the inverse matrix does not exist and break.
3: **for** $t \leftarrow 1, T$ **do**
4: **for** $s \leftarrow 1, S$ **do**
5: Simulate $\overline{A}$ using the Formula (3.4)
6: Determine $\left(\overline{A}\right)^{-1}$ using matrix inversion and store the entries as $\widetilde{x_{jk}}^s$ for all $j,k = 1,2,...,n$
7: **end for**
8: For each j and k, store the mean value of the $\widetilde{x_{jk}}^s$ values less (or similarly greater) than n_{jk}^L (or respectively n_{jk}^R) in $\left(\widetilde{x_{jk}}\right)_L^t$ (or $\left(\widetilde{x_{jk}}\right)_R^t$).
9: **end for**
10: Set $\left(\widetilde{x_{jk}}\right)_L := \max_t \left(\widetilde{x_{jk}}\right)_L^t$ and $\left(\widetilde{x_{jk}}\right)_R := \min_t \left(\widetilde{x_{jk}}\right)_R^t$ for all $j,k = 1,2,...,n$
11: Set the fuzzinesses as $\tau_{jk} := n_{jk}^L - \left(\widetilde{x_{jk}}\right)_L$ and $\phi_{jk} := \left(\widetilde{x_{jk}}\right)_R - n_{jk}^R$ for all $j,k = 1,2,...,n$
12: Using Eqs. (3.2) and (3.3), calculate the errors e_{ikR}^R, e_{ikL}^R, e_{ikR}^L, and e_{ikL}^L for the attained inverse
13: **end procedure**

under consideration must therefore be square. Our method consists of applying the bisection algorithm $2n^2$ times after the search algorithm, Algorithm 1. For the obtained inverse after running Algorithm 1, denote the left endpoint matrix as $(\widetilde{x})_L$ and the right endpoint matrix as $(\widetilde{x})_R$. Similarly, denote the matrix consisting of left endpoints of the mode intervals as n^L and the matrix consisting of right endpoints of the mode intervals as n^R, if existed. Refer to Algorithm 2 for this phase of our method. The sequence of the bisection repetitions is expressed by the loop counter m, which ranges from 1 to M. ω, on the other hand, is the order of the bisection iteration with the lowest error value.

4. Illustrative examples

On a computer running MS Windows 10 Pro and equipped with an Intel Core i5-7400 CPU (3.00 GHz) and 4 GB of RAM, all computational tests are carried out using MATLAB R2019a. We use the assumptions that $T = S = 1000$ and $M = 20$ in our computations. MATLAB's "rand" function is used as a pseudo-random generator. We display the values rounded to four significant digits. We take into consideration the following instances to illustrate the method suggested in this paper:

Example 4.1. *Our first example is from [4]. Consider the following fuzzy matrix with triangular entries:*

$$\widetilde{A} = \begin{bmatrix} (80,80,4,8)_{LR} & (25,25,5,5)_{LR} \\ (50,50,10,6)_{LR} & (120,120,6,10)_{LR} \end{bmatrix}.$$

We do not need to use Rohn's method since the entries are triangular FNs and mode intervals consist of only one number. Therefore, inconsistency is never an issue at the line-2 of Algorithm 1, so we simply calculate the inverse of a crisp matrix if it exists. The inverse matrix of the mode matrix is as follows:

$$n^L = n^R = \begin{bmatrix} 80 & 25 \\ 50 & 120 \end{bmatrix}^{-1} = \begin{bmatrix} 0.0144 & -0.0030 \\ -0.0060 & 0.0096 \end{bmatrix}.$$

Dehghan et al. [4] reported the left and right spread matrices for the inverse matrix as follows:

$$\left(\tau_{jk}\right)_1 = \begin{bmatrix} 0.0030 & 0.0013 \\ 0.0021 & 0.0019 \end{bmatrix} \text{ and } \left(\phi_{jk}\right)_1 = \begin{bmatrix} 0.0020 & 0.0016 \\ 0.0032 & 0.0013 \end{bmatrix},$$

respectively. Additionally, the following corresponding spreads are obtained with Rohn's method [2]:

$$\left(\tau_{jk}\right)_2 = \begin{bmatrix} 0.0022 & 0.0013 \\ 0.0020 & 0.0013 \end{bmatrix} \text{ and } \left(\phi_{jk}\right)_2 = \begin{bmatrix} 0.0019 & 0.0011 \\ 0.0022 & 0.0013 \end{bmatrix}.$$

Without applying the bisection algorithm, we find those spreads as follows:

$$\left(\tau_{jk}\right)_3 = \begin{bmatrix} 0.0004 & 0.0002 \\ 0.0002 & 0.0002 \end{bmatrix} \text{ and } \left(\phi_{jk}\right)_3 = \begin{bmatrix} 0.0002 & 0.0002 \\ 0.0004 & 0.0001 \end{bmatrix}.$$

Algorithm 2 Bisection Algorithm

INPUT: Left and right endpoint matrices $(\widetilde{x})_L$, $(\widetilde{x})_R$, boundaries of modes n^L, n^R, errors computed in Algorithm 1, and number of bisection repetitions M
OUTPUT: Updated *error* (3.1), and endpoint $\left(\widetilde{x_{jk}}\right)_r$ where $j = i$

procedure BISECTION(i,k,r)
  Initialize $h_L = 0$ and $h_R = 1$ (If the total error $(e^R_{ikr} + e^L_{ikr})$ is relatively much higher than the other ones, then initialize $h_L = 0.5$ and $h_R = 1$ to speed up the process)
  Set $m := 1$, $\left(\widetilde{x_{jk}}\right)^{new}_{r,m} := \left(\widetilde{x_{jk}}\right)_r$ and $error_m := error$ computed via Eq. (3.1)
  for $m \leftarrow 2, M$ **do**
    $h := (h_L + h_R)/2$
    $\left(\widetilde{x_{jk}}\right)^{new}_{r,m} := hn^r_{jk} + (1-h)\left(\widetilde{x_{jk}}\right)_r$
    Update the *error* using the new value $\left(\widetilde{x_{jk}}\right)^{new}_{r,m}$ instead of $\left(\widetilde{x_{jk}}\right)_r$, and store it in $error_m$
    if $error_m < error_{(m-1)}$ **then**
      $h_L := h$
    else
      $h_R := h$
    end if
  end for
  Return $\left(\widetilde{x_{jk}}\right)_r = \left(\widetilde{x_{jk}}\right)^{new}_{r,\omega}$ such that $error_\omega = \min_{m\in\{1,2,\ldots,M\}} error_m$
  Update *error*
end procedure

The authors [4] used $\alpha-$cuts to view the fuzzy matrix as an interval matrix and investigated five scenario-based matrices for $\alpha \in \{0, 0.25, 0.50, 0.75, 1\}$. In the first phase, on the other hand, we perform $T \times S = 10^6$ inverse calculations.
After the bisection algorithm, we obtain the following spreads:

$$\left(\tau_{jk}\right)_4 = \begin{bmatrix} 7.8317e-10 & 3.0507e-10 \\ 3.8743e-10 & 4.6294e-10 \end{bmatrix} \text{ and } \left(\phi_{jk}\right)_4 = \begin{bmatrix} 3.8099e-10 & 3.8117e-10 \\ 7.8932e-10 & 2.6150e-10 \end{bmatrix}.$$

The right and left multiplications of the inverse matrices obtained by the four methods mentioned above with the matrix $\widetilde{A}$ are calculated as follows:

$$\begin{aligned}
\left(\widetilde{A}\times\widetilde{X}\right)_1 &= \begin{bmatrix} [0.6234, 1.3872] & [-0.2244, 0.2206] \\ [-0.5970, 0.5992] & [0.6370, 1.3610] \end{bmatrix}, \\
\left(\widetilde{X}\times\widetilde{A}\right)_1 &= \begin{bmatrix} [0.6256, 1.3872] & [-0.3310, 0.3324] \\ [-0.4048, 0.3976] & [0.6348, 1.3610] \end{bmatrix}, \\
\left(\widetilde{A}\times\widetilde{X}\right)_2 &= \begin{bmatrix} [0.6872, 1.3584] & [-0.2124, 0.1826] \\ [-0.5520, 0.4796] & [0.7054, 1.3410] \end{bmatrix}, \\
\left(\widetilde{X}\times\widetilde{A}\right)_2 &= \begin{bmatrix} [0.6864, 1.3584] & [-0.3150, 0.2724] \\ [-0.3720, 0.3216] & [0.7062, 1.3410] \end{bmatrix}, \\
\left(\widetilde{A}\times\widetilde{X}\right)_3 &= \begin{bmatrix} [0.8753, 1.1708] & [-0.0908, 0.0792] \\ [-0.2464, 0.1805] & [0.8879, 1.1516] \end{bmatrix}, \\
\left(\widetilde{X}\times\widetilde{A}\right)_3 &= \begin{bmatrix} [0.8844, 1.1705] & [-0.1308, 0.1186] \\ [-0.1713, 0.1206] & [0.8788, 1.1518] \end{bmatrix}, \\
\left(\widetilde{A}\times\widetilde{X}\right)_4 &= \begin{bmatrix} [0.9126, 1.1449] & [-0.0719, 0.0599] \\ [-0.2036, 0.1222] & [0.9246, 1.1257] \end{bmatrix}, \\
\left(\widetilde{X}\times\widetilde{A}\right)_4 &= \begin{bmatrix} [0.9246, 1.1449] & [-0.1018, 0.0898] \\ [-0.1437, 0.0814] & [0.9126, 1.1257] \end{bmatrix}.
\end{aligned}$$

The sum of errors are 6.0826, 5.3212, 2.2564, and 1.7413, respectively. If we make an interpretation based on these findings, our method allows us to get better results that are more similar to the crisp identity matrix than the other two methods.
Now let us discuss some details of the implementation of the method. In the first phase, we reach the initial errors

$$\begin{aligned}
e^R_{11L} + e^L_{11L} &= 0.2403, \quad e^R_{11R} + e^L_{11R} = 0.3412, \\
e^R_{12L} + e^L_{12L} &= 0.2216, \quad e^R_{12R} + e^L_{12R} = 0.1978, \\
e^R_{21L} + e^L_{21L} &= 0.4177, \quad e^R_{21R} + e^L_{21R} = 0.3011, \\
e^R_{22L} + e^L_{22L} &= 0.2333, \quad e^R_{22R} + e^L_{22R} = 0.3034,
\end{aligned}$$

which measure lower and upper bound dissimilarities with the identity matrix. Thus, the bisection order to apply to spreads is as follows: $n_{21}^{L}-(\widetilde{x_{21}})_{L}$, $(\widetilde{x_{11}})_{R}-n_{11}^{R}$, $(\widetilde{x_{22}})_{R}-n_{22}^{R}$, $(\widetilde{x_{21}})_{R}-n_{21}^{R}$, $n_{11}^{L}-(\widetilde{x_{11}})_{L}$, $n_{22}^{L}-(\widetilde{x_{22}})_{L}$, $n_{12}^{L}-(\widetilde{x_{12}})_{L}$, $(\widetilde{x_{12}})_{R}-n_{12}^{R}$.

Example 4.2. *Our second example is from [5]. Consider the following fuzzy matrix with triangular entries:*

$$\widetilde{A}=\begin{bmatrix}(10,10,4,4)_{LR} & (8,8,3,3)_{LR}\\ (6,6,2,2)_{LR} & (4,4,3,3)_{LR}\end{bmatrix}.$$

Basaran [5] specified the fuzzy identity matrix as follows:

$$\widetilde{I}=\begin{bmatrix}(1,1,0.5,0.5)_{LR} & (0,0,0.5,0.5)_{LR}\\ (0,0,0.5,0.5)_{LR} & (1,1,0.5,0.5)_{LR}\end{bmatrix}.$$

The inverse matrices found in [5], [7], [8], and with the proposed method are respectively given as follows:

$$\begin{aligned}
\left(\widetilde{A}^{-1}\right)_{1} &= \begin{bmatrix}(-0.5,-0.5,0.88,0.88)_{LR} & (1,1,2.13,2.13)_{LR}\\ (0.75,0.75,1.63,1.63)_{LR} & (-1.25,-1.25,2.57,2.57)_{LR}\end{bmatrix},\\
\left(\widetilde{A}^{-1}\right)_{2} &= \begin{bmatrix}(-0.5,-0.5,0.875,0.875)_{LR} & (1,1,1.625,1.625)_{LR}\\ (0.75,0.75,0.625,0.625)_{LR} & (-1.25,-1.25,1.125,1.125)_{LR}\end{bmatrix},\\
\left(\widetilde{A}^{-1}\right)_{3} &= \begin{bmatrix}(-0.5,-0.5,0.5,0.11)_{LR} & (1,1,1,0.65)_{LR}\\ (0.75,0.75,0.75,0.35)_{LR} & (-1.25,-1.25,1.25,0.52)_{LR}\end{bmatrix},\\
\left(\widetilde{A}^{-1}\right)_{4} &= \begin{bmatrix}(-0.5,-0.5,2.0191e-06,1.4934e-06)_{LR} & (1,1,2.5932e-06,3.6078e-06)_{LR}\\ (0.75,0.75,1.9759e-06,2.8142e-06)_{LR} & (-1.25,-1.25,4.9302e-06,3.4408e-06)_{LR}\end{bmatrix}.
\end{aligned}$$

The right and left multiplications of the inverse matrices obtained by the four methods given above with the matrix $\widetilde{A}$ *are calculated as follows:*

$$\begin{aligned}
\left(\widetilde{A}\times\widetilde{X}\right)_{1} &= \begin{bmatrix}[-29.0000,31.5000] & [-57.8400,58.3400]\\ [-17.2000,19.7000] & [-35.7800,34.2800]\end{bmatrix},\\
\left(\widetilde{X}\times\widetilde{A}\right)_{1} &= \begin{bmatrix}[-28.3600,30.3600] & [-23.0900,26.0900]\\ [-42.8800,43.8800] & [-36.4200,35.4200]\end{bmatrix},\\
\left(\widetilde{A}\times\widetilde{X}\right)_{2} &= \begin{bmatrix}[-18.6250,20.3750] & [-34.8750,36.1250]\\ [-10.8750,12.6250] & [-21.6250,20.8750]\end{bmatrix},\\
\left(\widetilde{X}\times\widetilde{A}\right)_{2} &= \begin{bmatrix}[-24.2500,26.2500] & [-19.5000,22.5000]\\ [-18.2500,18.7500] & [-16.0000,15.0000]\end{bmatrix},\\
\left(\widetilde{A}\times\widetilde{X}\right)_{3} &= \begin{bmatrix}[-14.0000,9.7600] & [-27.5000,19.4500]\\ [-8.0000,6.1400] & [-17.5000,12.4700]\end{bmatrix},\\
\left(\widetilde{X}\times\widetilde{A}\right)_{3} &= \begin{bmatrix}[-14.0000,10.8600] & [-11.0000,9.6000]\\ [-20.0000,12.4800] & [-17.5000,11.3700]\end{bmatrix},\\
\left(\widetilde{A}\times\widetilde{X}\right)_{4} &= \begin{bmatrix}[-3.2500,5.2500] & [-7.7501,7.7501]\\ [-3.2500,3.2500] & [-4.7500,6.7500]\end{bmatrix},\\
\left(\widetilde{X}\times\widetilde{A}\right)_{4} &= \begin{bmatrix}[-3.0000,5.0000] & [-4.5000,4.5000]\\ [-5.5001,5.5001] & [-5.0000,7.0000]\end{bmatrix}.
\end{aligned}$$

The sum of errors are 542.1400, 328.5000, 213.6300, and 74.0007, respectively.

Example 4.3. *Our third example is from [5]. Consider the following fuzzy matrix with triangular entries:*

$$\widetilde{A}=\begin{bmatrix}(-6,-6,3,3)_{LR} & (-4,-4,2,2)_{LR}\\ (-4,-4,1,1)_{LR} & (-3,-3,1,1)_{LR}\end{bmatrix}.$$

with negative modes. With the same specified fuzzy identity matrix, the inverse matrices found in [5], [7], and with the proposed method are respectively given as follows:

$$\begin{aligned}
\left(\widetilde{A}^{-1}\right)_{1} &= \begin{bmatrix}(-1.5,-1.5,1.5,1.5)_{LR} & (2,2,2,2)_{LR}\\ (2,2,2.25,2.25)_{LR} & (-3,-3,3.5,3.5)_{LR}\end{bmatrix},\\
\left(\widetilde{A}^{-1}\right)_{2} &= \begin{bmatrix}(-1.5,-1.5,6,6)_{LR} & (2,2,8.25,8.25)_{LR}\\ (2,2,7,7)_{LR} & (-3,-3,9.5,9.5)_{LR}\end{bmatrix},\\
\left(\widetilde{A}^{-1}\right)_{3} &= \begin{bmatrix}(-1.5,-1.5,4.6516e-06,3.9282e-06)_{LR} & (2,2,5.5567e-06,6.5507e-06)_{LR}\\ (2,2,5.3942e-06,6.9193e-06)_{LR} & (-3,-3,9.7823e-06,7.6952e-06)_{LR}\end{bmatrix}.
\end{aligned}$$

The sum of errors are 416.5000, 1.4305e+03, and 105.0009, respectively.

Example 4.4. *Our fourth example is from [11]. Consider the following fuzzy matrix with triangular entries:*

$$\widetilde{A} = \begin{bmatrix} (5,5,1,1)_{LR} & (6,6,2,2)_{LR} \\ (4,4,2,2)_{LR} & (7,7,1,1)_{LR} \end{bmatrix}.$$

Babakordi and Taghi-Nezhad [11] reported the following inverse as a non-fuzzy matrix:

$$\widetilde{A}^{-1} = \begin{bmatrix} (0.6364,0.6364,0,0)_{LR} & (-0.5455,-0.5455,0,0)_{LR} \\ (-0.3636,-0.3636,0,0)_{LR} & (0.4545,0.4545,0,0)_{LR} \end{bmatrix}.$$

Their total error is 24.0000. Our inverse with a total error of 24.0001 is as follows:

$$\widetilde{A}^{-1} = \begin{bmatrix} (0.6364,0.6364,0.6306e-06,1.0550e-06)_{LR} & (-0.5455,-0.5455,1.0876e-06,0.6242e-06)_{LR} \\ (-0.3636,-0.3636,0.7781e-06,0.4623e-06)_{LR} & (0.4545,0.4545,0.4354e-06,0.7651e-06)_{LR} \end{bmatrix}.$$

The fuzzy matrix considered here has a non-fuzzy inverse, which is the case where we observe the minimum possible error value. Our method approximates this error.

Example 4.5. *Our next example is from [6]. Consider the following fuzzy matrix with triangular entries:*

$$\widetilde{A} = \begin{bmatrix} (10,10,0.1,0.1)_{LR} & (8,8,0.1,0.01)_{LR} \\ (6,6,0.15,0.1)_{LR} & (4,4,0.09,0.19)_{LR} \end{bmatrix}$$

with relatively small spreads. The author [6] specified the fuzzy identity matrix as follows:

$$\widetilde{I} = \begin{bmatrix} (1,1,0.78,0.4157)_{LR} & (0,0,0.5,0.642)_{LR} \\ (0,0,0.5,0.4183)_{LR} & (1,1,0.6038,0.43)_{LR} \end{bmatrix}.$$

The following fuzzy inverse matrix with an overall error of 0.7035 was given as a result:

$$\widetilde{A}^{-1} = \begin{bmatrix} (-0.5,-0.5,0.05,0.02)_{LR} & (1,1,0.02,0.01)_{LR} \\ (0.75,0.75,0.02,0.02)_{LR} & (-1.25,-1.25,0.023,0.04)_{LR} \end{bmatrix}.$$

With our approximation having a total error of 1.0160, the following inverse matrix is shown below:

$$\widetilde{A}^{-1} = \begin{bmatrix} (-0.5,-0.5,0.0526,0.0163)_{LR} & (1,1,0.0000,0.0115)_{LR} \\ (0.75,0.75,0.0152,0.0169)_{LR} & (-1.25,-1.25,0.0298,0.0369)_{LR} \end{bmatrix}.$$

The 0-cuts of the inverses are respectively given as:

$$\begin{bmatrix} [-0.5500,-0.4800] & [0.9800,1.0100] \\ [0.7300,0.7700] & [-1.2730,-1.2100] \end{bmatrix} \text{ and } \begin{bmatrix} [-0.5526,-0.4837] & [1.0000,1.0115] \\ [0.7348,0.7669] & [-1.2798,-1.2131] \end{bmatrix},$$

which end up being quite similar despite not being able to achieve a better error value.

Example 4.6. *Our last example is adapted from [14]. Consider the following fuzzy matrix with trapezoidal entries:*

$$\widetilde{A} = \begin{bmatrix} (2,4,3,1)_{LR} & (-2,1,0,5)_{LR} \\ (-1,2,4,4)_{LR} & (2,4,2,0)_{LR} \end{bmatrix}.$$

According to [2] and [15], we only need to consider the inverses of the following permutations:

$$A_{11} = \begin{bmatrix} 2 & -2 \\ -1 & 2 \end{bmatrix}, A_{12} = \begin{bmatrix} 4 & 1 \\ 2 & 4 \end{bmatrix}, A_{13} = \begin{bmatrix} 4 & -2 \\ 2 & 2 \end{bmatrix}, A_{14} = \begin{bmatrix} 2 & 1 \\ -1 & 4 \end{bmatrix},$$
$$A_{31} = \begin{bmatrix} 4 & 1 \\ -1 & 2 \end{bmatrix}, A_{32} = \begin{bmatrix} 2 & -2 \\ 2 & 4 \end{bmatrix}, A_{33} = \begin{bmatrix} 2 & 1 \\ 2 & 2 \end{bmatrix}, A_{34} = \begin{bmatrix} 4 & -2 \\ -1 & 4 \end{bmatrix}$$

where

$$\begin{aligned} A_I &= \begin{bmatrix} [2,4] & [-2,1] \\ [-1,2] & [2,4] \end{bmatrix}, \underline{A} = \begin{bmatrix} 2 & -2 \\ -1 & 2 \end{bmatrix}, \overline{A} = \begin{bmatrix} 4 & 1 \\ 2 & 4 \end{bmatrix} \\ A_c &= \frac{\overline{A}+\underline{A}}{2} = \begin{bmatrix} 3 & -0.5 \\ 0.5 & 3 \end{bmatrix}, \Delta = \frac{\overline{A}-\underline{A}}{2} = \begin{bmatrix} 1 & 1.5 \\ 1.5 & 1 \end{bmatrix}, \\ A_{yz} = A_c - T_y \Delta T_z,\ T_1 &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, T_2 = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}, T_3 = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, T_4 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \end{aligned}$$

Thus, the inverse matrix of the interval matrix is as follows:

$$A_I^{-1} = \left[\min_{(y,z)} A_{yz}^{-1}, \max_{(y,z)} A_{yz}^{-1}\right] = \begin{bmatrix} [1/6,1] & [-1/2,1] \\ [-1,1/2] & [1/6,1] \end{bmatrix}.$$

By implementing our algorithm, the following inverse matrix with a total error of 125.0003 is computed as follows:

$$\widetilde{A}^{-1} = \begin{bmatrix} (1/6,1,0.2846,0.2425e-05)_{LR} & (-1/2,1,0.0000,0.1855e-05)_{LR} \\ (-1,1/2,0.0000,0.1881e-05)_{LR} & (1/6,1,0.0936,0.2233e-05)_{LR} \end{bmatrix}.$$

The right and left multiplications of the inverse matrix obtained by our method with the matrix $\widetilde{A}$ are calculated as follows:

$$\begin{aligned} \widetilde{A} \times \widetilde{X} &= \begin{bmatrix} [-7.0000, 8.0000] & [-4.5000, 11.0000] \\ [-9.0000, 8.0000] & [-5.0000, 10.0000] \end{bmatrix}, \\ \widetilde{X} \times \widetilde{A} &= \begin{bmatrix} [-6.0000, 11.0000] & [-4.0000, 10.0000] \\ [-10.0000, 8.5000] & [-6.0000, 7.0000] \end{bmatrix}. \end{aligned}$$

Notice that the error of the Rohn method, 125, is a lower bound for our overall error. The fact that the error term is very close to the lower bound shows the efficiency of the method.

5. Conclusion

We provide a computational approach to find the multiplicative inverse of a square fuzzy matrix where the components are triangular or trapezoidal FNs. This method first obtains a large number of potential inverses to get intervals for the entries. After that, it tries to reduce the error by making these ranges smaller. The algorithm is implemented and successfully tested by solving several examples. At first glance, it seems that our algorithm reduces the total error of the approximate inverses obtained with the existing techniques. It is an effort to improve efficacy by making use of well-known, reliable, efficient, and non-fuzzy techniques. In the literature, fuzzy matrices whose entries are symmetrical triangular FNs have been considered mostly. But, inverse calculations of square fuzzy matrices with asymmetrical and/or trapezoidal fuzzy entries can be achieved with our methodology. Moreover, we are able to reduce the error because we employ interval arithmetic rather than approximate multiplication operations. Future developments of this study include the investigation of solving linear and non-linear equation systems with standard, intuitionistic, or picture fuzzy coefficients and extensional work into the applications of the proposed approach to other areas, such as mathematical programming.

Acknowledgements

The authors would like to express their sincere thanks to the editor and the anonymous reviewers for their helpful comments and suggestions.

Funding

There is no funding for this work.

Availability of data and materials

The data used to support the findings of this study are included within the article. All authors contributed equally to the writing of this paper. All authors read and approved the final manuscript.

References

[1] S. Chanas, M. Nowakowski, *Single value simulation of fuzzy variable*, Fuzzy Sets Syst., **25**(1) (1988), 43-57.
[2] J. Rohn, *Inverse interval matrix*, SIAM J. Numer. Anal., **30**(3) (1993), 864-870.
[3] S. S. Rao, L. Chen, *Numerical solution of fuzzy linear equations in engineering analysis*, Int. J. Numer. Methods Eng., **42**(5) (1998), 829-846.
[4] M. Dehghan, M. Ghatee, B. Hashemi, *Inverse of a fuzzy matrix of fuzzy numbers*, Int. J. Comput. Math., **86**(8) (2009), 1433-1452.
[5] M. A. Basaran, *Calculating fuzzy inverse matrix using fuzzy linear equation system*, Appl. Soft Comput., **12**(6) (2012), 1810-1813.
[6] M. Mosleh, *Evaluation of fully fuzzy matrix equations by fuzzy neural network*, Appl. Math. Modell., **37**(9) (2013), 6364-6376.
[7] M. Mosleh, M. Otadi, *A discussion on "Calculating fuzzy inverse matrix using fuzzy linear equation system"*, Appl. Soft Comput., **28** (2015), 511-513.
[8] J. Kaur, A. Kumar, *Commentary on "Calculating fuzzy inverse matrix using fuzzy linear equation system"*, Appl. Soft Comput., **58** (2017), 324-327.
[9] X. Guo, Y. Wei, Z. Li, *Further investigation to approximate fuzzy inverse*, J. Intell. Fuzzy Syst., **35**(1) (2018), 1161-1168.
[10] C. Y. Chen, J. J. Huang, *Deriving fuzzy weights of the fuzzy analytic network process via fuzzy inverse matrix*, Mathematics, **7**(10) (2019), 914.
[11] F. Babakordi, N. A. Taghi-Nezhad, *Calculating fuzzy inverse matrix using linear programming problem: An improved approach*, Pak. J. Stat. Oper. Res., (2021), 983-996.
[12] H. Farahani, M. J. Ebadi, *Finding fuzzy inverse matrix using Wu's method*, J. Mahani Math. Res. Cent., **10**(1) (2021), 37-52.
[13] H. G. Akdemir, H. G. Kocken, *A new fuzzy linear regression algorithm based on the simulation of fuzzy samples and an application on popularity prediction of Covid-19 related videos*, J. Stat. Manage. Syst., **25**(8) (2022), 2025-2041.
[14] P. V. Saraev, *Interval pseudo-inverse matrices and interval Greville algorithm*, Reliab. Comput., **18** (2013), 147-156.
[15] J. Lebedinska, *On another view of an inverse of an interval matrix*, Soft Comput., **14**(10) (2010), 1043-1046.