Journal of Al-Qadisiyah for Computer Science and Mathematics VOL.17.(3) 2025,pp.Math 168–179

Available online at www.qu.edu.iq/journalcm

JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS

ISSN:2521-3504(online) ISSN:2074-0204(print)

Bi-Conjugate Gradient Method for Solving Positive Triangular Fully Fuzzy Linear Systems

Sirwan Sherko Rasheed [a] **,** ***Ivan Subhi Latif*** [b]

[a]*Department of Mathematics, College of Education, University of Salahaddin, Erbil, Iraq. Email: sirwan.rasheed@su.edu.krd*

[b]*Department of Mathematics, College of Education, University of Salahaddin, Erbil, Iraq. Email: ivan.latif@su.edu.krd*

ARTICLE INFO

Article history:
Received: 15/07/2025
Rrevised form: 05/08/2025
Accepted : 10/08/2025
Available online: 30/09/2025

Keywords:

Numerical Optimization,
Fully Fuzzy Linear Systems,
Fuzzy Number,
Fuzzy Arithmetic,
Bi-Conjugate Gradient Method

ABSTRACT

Fuzzy linear systems play an important and efficient role across various domains such as mathematics, engineering, physics, chemistry, economics, statistics, and so on. Dealing with these kinds of systems in the real world is still very hard. This paper suggests a new way to solve fully fuzzy linear systems quickly using the Bi-conjugate gradient method (BGM) ^{SI}. The method builds a one-block rate matrix and lets it skip fuzzy arithmetic operations. The focus is on systems in which both the coefficients and the variables are fuzzy, aiming to produce positive results even under highly uncertain conditions. The (BGM) ^{SI} algorithm is efficient in solving FFLS, requiring only a few iterations and converting the system into a crisp linear system first. To test the validity of the proposed method, we ran three numerical experiments, which confirmed its effectiveness and robustness. Unlike Jacobi or Gauss-Seidel, this proposed method has more efficient and quicker convergence. It is most applicable for FFLS when there are stringent demands on precision and system parameters are uncertain.

MSC..

https://doi.org/10.29304/jqcsm.2025.17.32439

1. Introduction

Fuzzy linear systems have been getting considerable attention as an appropriate mathematical platform to describe the uncertainty concerning real-world problems where imprecision and vagueness are present, for example, in engineering, economics, and decision-making processes. Friedman et al. were among the earliest scholars to explore the theoretical framework of FLS and pave the way for later computational methods [1]. With time, the attention moved from partially fuzzy to fully fuzzy linear systems (FFLS), which pose all components, including coefficients and right-side constants, as fuzzy numbers. This shift opened several new theoretical and algorithmic frontiers. Abbasbandy et al. improved fuzzy symmetric positive definite systems with the conjugate gradient method, thus

∗Corresponding author: Sirwan Sherko Rasheed

Email addresses: sirwan.rasheed@su.edu.krd

Communicated by ‘sub etitor’

increasing the numerical precision of the computation [2]. Later on, Dehghan et al. put forth direct and iterative methods for fully fuzzy linear systems (FFLS), which broadened the methods available for finding solutions to the systems [3, 4]. Allahviranloo et al. [5] investigated maximal and minimal symmetric solutions, and Kumar et al. presented novel methods for systems based on triangular fuzzy numbers [6-8]. The work done by Otadi et al. [9] on fuzzy neural networks and their application to numerical solutions illustrates the application of modern computational methods in the field. More recently, Dookhitram et al. [10] developed preconditioning algorithms, and Ezzati et al. modified the formulation to general fuzzy systems, thus extending the scope of the work [11]. To enhance method accuracy and solution refinement, algorithms focusing on specific forms of fuzzy numbers like triangular and trapezoidal fuzzy numbers have been devised [7, 12, 13]. Malkawi and other authors have outlined solution positivity and consistency and provided necessary and sufficient condition frameworks for system solvability [12, 14-16]. With computational performance becoming increasingly critical, faster methods such as the fast iterative method (FIM) were introduced by Abdolmaleki and Edalatpanah [17]. Also, positive triangular FFLS have been targeted by refined iterative techniques in earlier works [18], while fully fuzzy mixed integer linear programming has been tackled with new interactive methods [19]. Hybrid methods combining analytical and heuristic features, including ranking functions and modified fuzzy numbers, have also been developed [20, 21]. In terms of theoretical innovation, Mikaeilvand et al. introduced novel algebraic frameworks for solving fuzzy systems [22], while Ghanbari et al. addressed dual fuzzy systems, extending the FFLS structure further [23]. Allahviranloo and Abbasi contributed a new solution concept aiming to resolve existing limitations in conventional fuzzy arithmetic [24], and Babakordi proposed a fuzzy analog to Cramer's rule for efficient solutions [25]. Recent contributions have pushed the boundary of FFLS research by addressing complex system types and solution uniqueness, including non-negative systems [26], and numerical solutions have also been explored for coupled trapezoidal fully fuzzy Sylvester matrix equations [13]. These studies reflect a mature and still-expanding body of work that aims to bridge the gap between fuzzy theory and practical linear algebra.
This paper introduces the Bi-conjugate gradient method (BGM) [SI], which offers a constructive solution for arbitrary coefficients in FFLS utilizing a one-block matrix framework. This methodology is capable of accommodating systems of any dimension. The structure of this paper is delineated into six sections. Section 2 provides a comprehensive review of the fundamental definitions of fuzzy set theory. In section 3, it introduces the proposed model. In section 4, two numerical experiments are intended to show the efficiency of the proposed iterative method. Section 5 compares the proposed approach with other results to demonstrate its efficiency. The conclusion of the results we obtained is in section 6.

2. Preliminaries

This segment provides a comprehensive review of fundamental definitions within fuzzy set theory

Definition 2.1:[18] Let Z be a universal set. A fuzzy subset $\widehat{K}$ of Z is defined by its membership function

$$\mu_{\widehat{K}}(z): Z \to [0,1],$$

which allocates to each element $z \in Z$ a real number $\mu_{\widehat{K}}(z)$ in the closed interval $[0,1]$. Here, the value of $\mu_{\widehat{K}}(z)$ at z represents the degree of membership of z in the fuzzy subset $\widehat{K}$.

Formally, a fuzzy set $\widehat{K}$ is represented as $\widehat{K} = \{(z, \mu_{\widehat{K}}(z)), z \in Z, \mu_{\widehat{K}}(z) \in [0,1]\}$, where $\mu_{\widehat{K}}(z)$ represents the degree of membership of z in $\widehat{K}$.

Definition 2.2:[19] A fuzzy set $\widehat{K}$ in $Z = \mathbb{R}^n$ is considered convex if, for all $z_1, z_2 \in \mathrm{Z}$, and for all $\lambda \in [0,1]$, the condition $\mu_{\widehat{K}}(\lambda z_1 + (1-\lambda) z_2) \geq \min\{\mu_{\widehat{K}}(z_1), \mu_{\widehat{K}}(z_2)\}$ holds true.

Definition 2.3:[18] A fuzzy set $\widehat{K}$ defined on a universal set $Z = \mathbb{R}^n$ is called normal fuzzy set if there exists at least one element $z \in Z$ such that $\mu_{\widehat{K}}(z) = 1$

Definition 2.4:[8] A fuzzy set $\widehat{K}$, defined over the universal set of real numbers. $\mathbb{R}$ is designated as a fuzzy number if its membership function $\mu_{\widehat{K}}(z)$: $\mathbb{R} \to [0,1]$ satisfies the following properties:

a) **Convexity:** The membership function is convex, meaning for all $z_1, z_2 \in \mathbb{R}$ and for any $\lambda \in [0,1]$, the following inequality holds: $\mu_{\widehat{K}}(\lambda z_1 + (1-\lambda) z_2) \geq \min\{\mu_{\widehat{K}}(z_1), \mu_{\widehat{K}}(z_2)\}$ holds.
b) **Normality:** There exist at least one point $z_0 \in \mathbb{R}$ such that $\mu_{\widehat{K}}(z_0) = 1$.
c) **Piecewise Continuity:** The membership function $\mu_{\widehat{K}}(z)$ is continuous in pieces over the set of real numbers $\mathbb{R}$.

Definition2.5:[20] A fuzzy triangular number $\widehat{K} = (\alpha, \beta, \gamma)$, illustrated in Figure 1, represents a fuzzy quantity characterized by the α lower boundary, the β peak value, and the γ upper boundary, along with its membership function $\mu_{\widehat{K}}(z)$ defined as follows:

$$\mu_{\widehat{K}}(z) = \begin{cases} 0, & z \le \alpha, \\ \frac{z-\alpha}{\beta-\alpha}, & \alpha \le z \le \beta, \\ \frac{\gamma-z}{\gamma-\beta}, & \beta \le z \le \gamma \\ 0, & \gamma \le z. \end{cases} \tag{1}$$

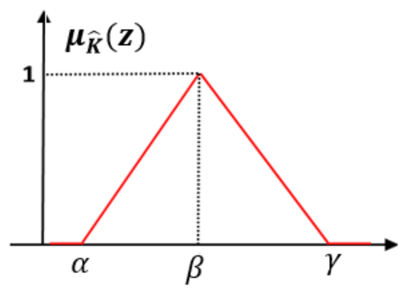

Fig. 1 - Described a triangular fuzzy number

Definition 2.6:[18] Two triangular fuzzy numbers $\widehat{M} = (\alpha, \beta, \gamma)$ and $\widehat{N} = (\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma})$ are considered equal if and only if the conditions $\tilde{\alpha} = \alpha$, $\tilde{\beta} = \beta$, $\tilde{\gamma} = \gamma$ are satisfied.

Definition 2.7:[19] For two triangular fuzzy numbers $\widehat{M} = (\alpha, \beta, \gamma)$ and $\widehat{N} = (\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma})$, we define the following arithmetic operations:

1. Addition: $\widehat{M} \oplus \widehat{N} = (\alpha + \tilde{\alpha}, \beta + \tilde{\beta}, \gamma + \tilde{\gamma})$.
2. Symmetry: $-\widehat{M} = -(\alpha, \beta, \gamma) = (-\alpha, \beta, \gamma)$.
3. Subtraction: $\widehat{M} \ominus \widehat{N} = (\alpha - \tilde{\alpha}, \beta + \tilde{\gamma}, \gamma + \tilde{\beta})$.
4. Multiplication: Suppose $\widehat{M}$ be any triangular fuzzy number and $\widehat{N}$ be non-negative triangular fuzzy number, then we define:

$$\widehat{M} \otimes \widehat{N} \simeq \begin{cases} (\alpha\tilde{\alpha}, \beta\tilde{\beta}, \gamma\tilde{\gamma}), & \alpha \ge 0 \\ (\alpha\tilde{\gamma}, \beta\tilde{\beta}, \gamma\tilde{\gamma}), & \alpha < 0, \gamma \ge 0 \\ (\alpha\tilde{\gamma}, \beta\tilde{\beta}, \gamma\tilde{\alpha}), & \gamma < 0 \end{cases}$$

Definition 2.8:[12] A crisp matrix K is called inverse-nonnegative if $K > 0$ and $K^{-1} > 0$.

Definition 2.9:[7] A matrix $\widehat{K} = (\widehat{k}_{ij})$ is referred to as a fuzzy matrix if every element in $\widehat{K}$ is a fuzzy number. The matrix $\widehat{K}$ is said to be positive (negative) and denoted by $\widehat{K} > 0(\widehat{K} < 0)$ if all of its components are positive (negative) fuzzy numbers. $\widehat{K}$ is said to be non-negative (non-positive) and denoted by $\widehat{K} \ge 0(\widehat{K} \le 0)$ if each of its entries is non-negative (non-positive) fuzzy number. A fuzzy matrix of order $n \times m$, $\widehat{K} = (\widehat{k}_{ij})_{(n\times m)}$, where each $\widehat{k}_{ij}$ is a fuzzy number in from $(k_{ij}, \alpha_{ij}, \beta_{ij})$, using a new notation $\widehat{K} = (K, M, N)$, where $K = (k_{ij}), M = (\alpha_{ij})$ and $N = (\beta_{ij})$ are all $n \times m$ real-valued matrices representing the center, left spread and right spread, respectively.

Definition 2.10:[12] Let $\widehat{K} = (\widehat{k}_{ij})$ and $\widehat{H} = (\widehat{h}_{ij})$ be two $m \times n$ and $n \times q$ fuzzy matrices respectively. We define $\widehat{K} \otimes \widehat{H} = \widehat{G} = (\widehat{g}_{ij})$ which is the $m \times q$ matrix, where
$\widehat{g}_{ij} = \sum^{\oplus}_{P=1,\ldots,n} \widehat{k}_{iP} \otimes \widehat{g}_{Pj}$.

Definition 2.11:[7] Consider the fully fuzzy linear system of equations of size $n \times n$:

$$\begin{cases} (\widehat{k}_{11} \otimes \widehat{z}_1) \oplus (\widehat{k}_{12} \otimes \widehat{z}_2) \oplus \cdots \oplus (\widehat{k}_{1n} \otimes \widehat{z}_{\mathrm{n}}) = \widehat{g}_1 \\ (\widehat{k}_{21} \otimes \widehat{z}_1) \oplus (\widehat{k}_{22} \otimes \widehat{z}_2) \oplus \cdots \oplus (\widehat{k}_{2n} \otimes \widehat{z}_{\mathrm{n}}) = \widehat{g}_2 \\ \vdots \\ (\widehat{k}_{n1} \otimes \widehat{z}_1) \oplus (\widehat{k}_{n2} \otimes \widehat{z}_2) \oplus \cdots \oplus (\widehat{k}_{nn} \otimes \widehat{z}_{\mathrm{n}}) = \widehat{g}_{\mathrm{n}} \end{cases} \tag{2}$$

The above system's matrix form is

$$\widehat{K} \otimes \hat{Z} = \hat{G}, \tag{3}$$

or simply $\widehat{K}\hat{Z} = \hat{G}$ where the coefficient matrix $\widehat{K} = (\hat{k}_{ij})$ and the vector $\hat{G} = (\hat{g}_i)$, $1 \le i, j \le n$ is an $n \times n$ fuzzy matrix and $\hat{z}_i, \hat{g}_i \in F(\mathbb{R})$, $1 \le i \le n$ are fuzzy vectors. We call the system (3) a fully fuzzy linear system (FFLS).

Up to rest of this paper we want to find the positive solution of FFLS $\widehat{K}\hat{Z} = \hat{G}$,

where $\widehat{K} = (K, M, N) \ge 0$, $\hat{G} = (g, b, h) \ge 0$ and $\hat{Z} = (x, y, z) \ge 0$.

So we have

$$(K, M, N) \otimes (x, y, z) = (g, b, h). \tag{4}$$

3. The Proposed Bi-Conjugate Gradient Method (BGM) SI

In this part, a new approach to obtaining a positive solution for FFLS with unknown coefficients will be presented. Examine the positive FFLS that follows.

$\widehat{K} \otimes \hat{Z} = \hat{G}$, where, $\widehat{K} = (\hat{k}_{ij})_{n\times n} = (K, M, N)$, $\hat{G} = (\hat{g}_i)_{n\times 1} = (g, b, h)$, And,

$\hat{Z} = (\hat{z}_j)_{n\times 1} = (x, y, z) \ge 0$, $(K, M, N) \otimes (x, y, z) = (g, b, h)$.

Let $\hat{k}_{ij} = (k_{ij}, \alpha_{ij}, \beta_{ij})$, $\hat{z}_j = (x_j, y_j, z_j)$ and $\hat{g}_i = (g_i, b_i, h_i)$.

Define a block matrix, $S = \begin{pmatrix} K & 0 & 0 \\ N & K & 0 \\ M & 0 & K \end{pmatrix}$,

where K, M and N square matrices are in common size n.

$$K = \begin{pmatrix} k_{11} & k_{12} & \dots & k_{1n} \\ k_{21} & k_{22} & \cdots & k_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ k_{n1} & k_{n2} & \dots & k_{nn} \end{pmatrix}, \quad M = \begin{pmatrix} m_{11} & m_{12} & \dots & m_{1n} \\ m_{21} & m_{22} & \cdots & m_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n1} & m_{n2} & \dots & m_{nn} \end{pmatrix}, \quad N = \begin{pmatrix} n_{11} & n_{12} & \dots & n_{1n} \\ n_{21} & n_{22} & \cdots & n_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ n_{n1} & n_{n2} & \dots & n_{nn} \end{pmatrix},$$

Also let

$Z = Vec(x,\ y,\ z) = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ and $G = Vec(g,\ b,\ h) = \begin{pmatrix} g \\ b \\ h \end{pmatrix}$, Where x, y, z, g, b and h are vectors of n components.

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \quad y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, \quad z = \begin{pmatrix} z_1 \\ \vdots \\ z_n \end{pmatrix}, \ g = \begin{pmatrix} g_1 \\ \vdots \\ g_n \end{pmatrix}, \ b = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}, \ h = \begin{pmatrix} h_1 \\ \vdots \\ h_n \end{pmatrix}.$$

We will appoint a new $3n \times 3n$ linear system, $SZ = G$, in matrix form,

$\begin{pmatrix} K & 0 & 0 \\ M & K & 0 \\ N & 0 & K \end{pmatrix} Vec(x, y, z) = Vec(g, b, h)$, as a result, the new linear system is writable.

$$\begin{pmatrix} K & 0 & 0 \\ M & K & 0 \\ N & 0 & K \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} g \\ b \\ h \end{pmatrix}. \tag{5}$$

Now applying the following Bi-conjugate gradient method (BGM) SI.

Given a linear system of equations represented as: $SZ = G$, where $S \in \mathbb{R}^{n\times n}$ is a (potentially non-symmetric) large sparse matrix, $Z \in \mathbb{R}^n$ is the vector of unknowns, and $G \in \mathbb{R}^n$ is the right-hand side vector, the goal is to find the vector Z. The (BGM) [SI] iteration, in its standard form without preconditioning, begins by choosing an initial guess Z_0 and computing the corresponding residual $R_0 = G - SZ_0$. We then set the shadow residual $\hat{R}_0 = R_0$, which is often taken to be the same as R_0 to simplify implementation. The first set of direction vectors is chosen as $P_0 = R_0$ and $\hat{P}_0 = \hat{R}_0$.

3.1 The Algorithm for the Bi-Conjugate Gradient Method (BGM) [SI]

The basic steps for computation of the Bi-conjugate gradient method (BGM) [SI] are given as follows:

Step 1: Define FFLS $\hat{K} + \hat{Z} = \hat{G}$ and construct a block matrix S and vectors Z and G to represent FFLS in block matrix form, where $S = \begin{pmatrix} K & 0 & 0 \\ N & K & 0 \\ M & 0 & K \end{pmatrix}$, $Z = Vec(x,\ y,\ z)$, $G = Vec(g,\ b,\ h)$. Rewrite FFLS as $SZ = G$.

Step 2: Initialize variables for the Bi-conjugate gradient method (BGM) [SI] set initial guess Z_0, compute initial residual $R_0 = G - S * Z_0$ and set shadow residual $\hat{R}_0 = R_0$, initialize search directions $P_0 = R_0$ and $\hat{P}_0 = \hat{R}_0$.

Step 3: For each iteration $n = 0,1,2, \ldots$ until convergence.

Step 4: Compute step size (α_n), $\alpha_n = \frac{(\hat{R}_n{}^T * R_n)}{(\hat{P}_n{}^T * S * P_n)}$, update solution vector (Z_{n+1}) where $Z_{n+1} = Z_n + \alpha_n * P_n$. update residual $R_{n+1} = R_n - \alpha_n\, S * P_n$ and shadow residual where $\hat{R}_{n+1} = \hat{R}_n - \alpha_n\, S^T * P_n$

Step 5: Check for convergence, if $||R_{n+1}||$ is sufficiently small, go to step 7.

Step 6: If not converged, compute the scalar $\beta_n = \frac{(\hat{R}_{n+1}{}^T * R_{n+1})}{(\hat{R}_n{}^T * R_n)}$, update search directions $P_{n+1} = R_{n+1} + \beta_n * P_n$ and $\hat{P}_{n+1} = \hat{R}_{n+1} + \beta_n * \hat{P}_n$

Step 7: Output the solution Z_{n+1} as the approximate solution.

Step 8: End.

The computational complexity of the new (BGM) [SI] arises in step 3 of the iterative process, where each iteration requires two matrix-vector multiplications Sp_{n-1} and $S^T p_{n-1}$. These operations are expensive for large and sparse matrices because they involve processing all non-zero entries in the matrix. For this reason, the complexity of each iteration is approximate. The efficiency of the method therefore depends on the size of the matrix, its sparsity pattern and the number of iteration required for convergence.

3.2 Convergence of Bi-Conjugate Gradient Method (BGM) [SI] in Fuzzy Context.

The Bi-conjugate Gradient Method (BGM) [SI] is one of the iterative methods used to solve large, sparse linear systems of equations, particularly in cases where there is no symmetry in the coefficient matrix. In contrast to the conventional conjugate gradient method, which is limited to symmetric positive definite matrices, the (BGM) [SI] can effectively tackle issues related to general non-symmetric matrices. Its computational benefit resides in executing matrix-vector multiplication instead of direct factorization or complete matrix storage, rendering it highly applicable in extensive and intricate systems. The rate at which (BGM) [SI] converges is contingent upon the spectral characteristics of the matrix; when the eigenvalues are evenly distributed, or effective preconditioning is utilized, the method demonstrates swift convergence. (BGM) [SI] generates two sets of search directions linked to the matrix and its transpose, thereby satisfying the conjugacy condition required for the iterative procedure. It is the flexibility, low memory requirements, and broad adaptability to diverse scientific and engineering challenges that make the method outstanding. The Bi-conjugate Gradient Method (BGM) [SI] is exceptional in efficiency and accuracy for solving fully fuzzy linear systems (FFLS) within fuzzy contexts. When fuzzy numbers and systems are concerned, how quickly and reliably a method converges is essential. The primary cause of such convergence with the (BGM) [SI] is its ability to reduce the solution of FFLS to a one-block matrix system $SZ = G$, which is solvable by (BGM) [SI] iteratively. The numerical illustrations in the paper show how the (BGM) [SI] is guaranteed convergence, and in addition, the number of iterations is surprisingly small. In Problem (1), for a 2×2 FFLS posed as a problem, the algorithm computed every variable exactly and achieved residual norms of the order 10^{-12} in 4 iterations. Problem (3) contained a 3×3 FFLS, which was more complicated. Despite this, it was able to converge with a residual norm of 10^{-10} within 6 iterations. The rate of convergence remains a key distinguishing feature of this algorithm among other iterative methods, such as the Gauss-Jacobi and Gaussian Saddle methods, which invariably take a large number of iterations to converge for a multitude of variable sets within the same class of fuzzy linear systems. The $(BGM)^{SI}$ algorithm's iterative characteristics make it particularly useful for large systems with uncertainty represented by triangular fuzzy numbers. It is efficient and scalable because it can handle large, sparse, and non-symmetric systems without requiring matrix inversion. Moreover, the $(BGM)^{SI}$ method posits that systems must be solved by ensuring positivity and insensitivity to uncertainties, which is crucial in fuzzy contexts where the solutions have to retain their fuzzy nature and account for unavoidable vagueness. The results prove that the algorithm is very robust to data fuzzification because it reliably converges to a complete, exact solution. The paper notes that convergence was achieved with minimal effort at most, four iterations for all variables in the first problem. This is advantageous compared to more traditional methods that need severe simplifications or transformations, which tend to damage the integrity of the fuzzy solution.

4. Numerical Illustration

In this section, to evaluate the efficacy and precision of the proposed iterative technique outlined in section 3, we will employ Python code software to resolve the following fully fuzzy linear system with a known exact solution.

Problem (1): Consider the following fully fuzzy linear system of equations:

$$(5,1,1) \otimes (x_1, y_1, z_1) \oplus (6,1,2) \otimes (x_2, y_2, z_2) = (50,10,17)$$

$$(7,1,0) \otimes (x_1, y_1, z_1) \oplus (4,0,1) \otimes (x_2, y_2, z_2) = (48,5,7)$$

Using this problem in matrix form, thus we have

$$K = \begin{bmatrix} 5 & 6 \\ 7 & 4 \end{bmatrix}, \; M = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}, \; N = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}$$

$$g = \begin{bmatrix} 50 \\ 48 \end{bmatrix}, \quad b = \begin{bmatrix} 10 \\ 5 \end{bmatrix}, \quad h = \begin{bmatrix} 17 \\ 7 \end{bmatrix}$$

In matrix form $\begin{bmatrix} (5,1,1) & (6,1,2) \\ (7,1,0) & (4,0,1) \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} (50,10,17) \\ (48,5,7) \end{bmatrix}$

To find $S = \begin{bmatrix} K & 0 & 0 \\ M & K & 0 \\ N & 0 & K \end{bmatrix} = \begin{bmatrix} \begin{pmatrix} 5 & 6 \\ 7 & 4 \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} & \begin{pmatrix} 5 & 6 \\ 7 & 4 \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} 5 & 6 \\ 7 & 4 \end{pmatrix} \end{bmatrix} = \begin{pmatrix} 5 & 6 & 0 & 0 & 0 & 0 \\ 7 & 4 & 0 & 0 & 0 & 0 \\ 1 & 1 & 5 & 6 & 0 & 0 \\ 1 & 0 & 7 & 4 & 0 & 0 \\ 1 & 2 & 0 & 0 & 5 & 6 \\ 0 & 1 & 0 & 0 & 7 & 4 \end{pmatrix}$

We obtain matrix Z where $Z = \begin{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \\ \begin{pmatrix} z_1 \\ z_3 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \\ z_1 \\ z_2 \end{pmatrix}$ and $G = \begin{pmatrix} \begin{pmatrix} g_1 \\ g_2 \end{pmatrix} \\ \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \\ \begin{pmatrix} h_1 \\ h_2 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} g_1 \\ g_2 \\ b_1 \\ b_2 \\ h_1 \\ h_2 \end{pmatrix} = \begin{pmatrix} 50 \\ 48 \\ 10 \\ 5 \\ 17 \\ 7 \end{pmatrix}$

Where $SZ = G$. Then $\begin{pmatrix} 5 & 6 & 0 & 0 & 0 & 0 \\ 7 & 4 & 0 & 0 & 0 & 0 \\ 1 & 1 & 5 & 6 & 0 & 0 \\ 1 & 0 & 7 & 4 & 0 & 0 \\ 1 & 2 & 0 & 0 & 5 & 6 \\ 0 & 1 & 0 & 0 & 7 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \\ z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} 50 \\ 48 \\ 10 \\ 5 \\ 17 \\ 7 \end{pmatrix}$

The exact solutions for the fuzzy linear system from the equations $Z = S^{-1}G$ are

$(x_1, y_1, z_1) = (4, 0.09091, 0)$ and $(x_2, y_2, z_2) = (5, 0.09091, 0.5)$

where S is non-symmetric for Bi-conjugate gradient method, first set the starting point with an initial point

$$Z_0 = \begin{pmatrix} 0\\0\\0\\0\\0\\0 \end{pmatrix}. \text{ Therefore } R_0 = G - SZ_0 = \begin{pmatrix} 50\\48\\10\\5\\17\\7 \end{pmatrix} - \begin{pmatrix} 5 & 6 & 0 & 0 & 0 & 0\\ 7 & 4 & 0 & 0 & 0 & 0\\ 1 & 1 & 5 & 6 & 0 & 0\\ 1 & 0 & 7 & 4 & 0 & 0\\ 1 & 2 & 0 & 0 & 5 & 6\\ 0 & 1 & 0 & 0 & 7 & 4 \end{pmatrix} \begin{pmatrix} 0\\0\\0\\0\\0\\0 \end{pmatrix} = \begin{pmatrix} 50\\48\\10\\5\\17\\7 \end{pmatrix}$$

$$\hat{R}_0 = R_0 = \begin{pmatrix} 50\\48\\10\\5\\17\\7 \end{pmatrix}, \quad P_0 = R_0 = \begin{pmatrix} 50\\48\\10\\5\\17\\7 \end{pmatrix}, \quad \hat{P}_0 = \hat{R}_0 = \begin{pmatrix} 50\\48\\10\\5\\17\\7 \end{pmatrix}$$

To find α_0 used algorithm(3.1)

$$\alpha_0 = \frac{\hat{R}_0^{\,T} * R_0}{\hat{P}_0^{\,T} * S * P_0} = \frac{5267}{61402} = 0.0858$$

$$\text{Then } Z_1 = Z_0 + \alpha_0 P_0 = \begin{pmatrix} 0\\0\\0\\0\\0\\0 \end{pmatrix} + \begin{pmatrix} 4.2889\\4.1174\\0.8578\\0.4289\\1.4583\\0.6005 \end{pmatrix} = \begin{pmatrix} 4.2889\\4.1174\\0.8578\\0.4289\\1.4583\\0.6005 \end{pmatrix}$$

$$R_1 = R_0 - \alpha_0 S P_0 = \begin{pmatrix} 50\\48\\10\\5\\17\\7 \end{pmatrix} - \begin{pmatrix} 41.1491\\46.4922\\15.2687\\12.0091\\23.4177\\16.7269 \end{pmatrix} = \begin{pmatrix} 3.8509\\1.5078\\-5.2687\\-7.0091\\-6.4177\\-9.7269 \end{pmatrix}$$

$$\hat{R}_1 = \hat{R}_0 - \alpha_0 S^T \hat{P}_0 = \begin{pmatrix} 50\\48\\10\\5\\17\\7 \end{pmatrix} - \begin{pmatrix} 53.0114\\46.5780\\7.2912\\6.8623\\11.4944\\11.1513 \end{pmatrix} = \begin{pmatrix} -3.0114\\1.4220\\2.7088\\-1.8623\\5.5056\\-4.1513 \end{pmatrix}$$

$$\beta_0 = \frac{\hat{R}_1^{\,T} * R_1}{\hat{R}_0^{\,T} * R_0} = \frac{-5.6254}{5267} = -0.0011$$

$$P_1 = R_1 + \beta_0 P_0 = \begin{pmatrix} 3.8509\\1.5078\\-5.2687\\-7.0091\\-6.4177\\-9.7269 \end{pmatrix} + \begin{pmatrix} -0.0534\\-0.0513\\-0.0107\\-0.0053\\-0.0182\\-0.0075 \end{pmatrix} = \begin{pmatrix} 3.7975\\1.4565\\-5.2793\\-7.0144\\-6.4358\\-9.7344 \end{pmatrix}$$

$$\hat{P}_1 = \hat{R}_1 + \beta_0 \hat{P}_0 = \begin{pmatrix} -3.0114\\1.4220\\2.7088\\-1.8623\\5.5056\\-4.1513 \end{pmatrix} + \begin{pmatrix} -0.0534\\-0.0513\\-0.0107\\-0.0053\\-0.0182\\-0.0075 \end{pmatrix} = \begin{pmatrix} -3.0648\\1.3708\\2.6981\\-1.8677\\5.4875\\-4.1587 \end{pmatrix}$$

Iterations continue until the approximately optimal solution is obtained, which occurs for problem (1) in the fourth step, demonstrating rapid convergence, as explained in Table 1.

Table 1 - Numerical results for the optimal approximate solution of Problem (1) using algorithm (BGM) SI step-by-step.

No. of iterations(n)	Residual Norm	$\hat{z}_1 = (x_1, y_1, z_1)$	$\hat{z}_2 = (x_2, y_2, z_2)$
$n = 0$	1.5159E+01	(4.2889, 0.8578, 1.4583)	(4.1174, 0.4289, 0.6005)
$n = 1$	1.1276E+01	(4.3888, 0.7189, 1.2890)	(4.1557, 0.2444, 0.3444)
$n = 2$	4.5629E-01	(4.0817, -0.0301, 0.0814)	(4.9060, 0.2238, 0.4178)
$n = 3$	3.9151E-12	(4, 0.0909, 0)	(5, 0.0909, 0.5)

It is clear that the sequence of iterations converges to a complete and exact solution.

$$(x_1, y_1, z_1) = (4, 0.09091, 0) \text{ and } (x_2, y_2, z_2) = (5, 0.09091, 0.5)$$

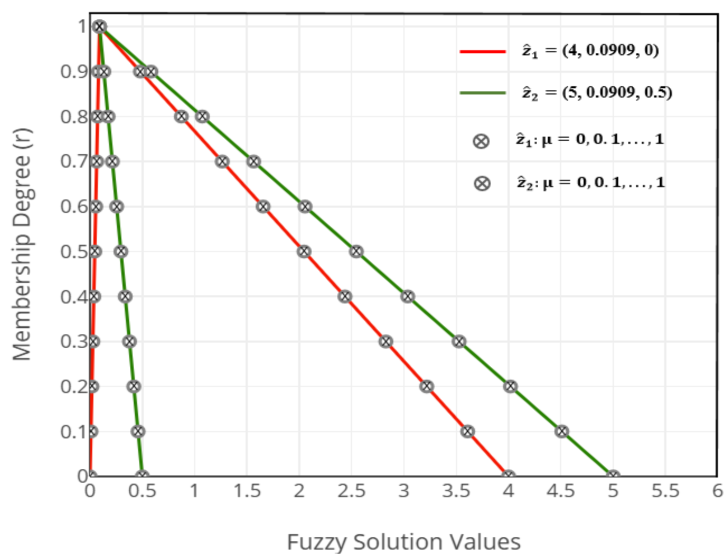

Fig. 2 - Shows the graphical representation of the fuzzy solution ($\hat{z}_1$, $\hat{z}_2$) for problem (1), using the Bi-conjugate gradient method (BGM) SI.

Problem (2): Application of FFLS in manufacturing machines: The Omega Manufacturing Company has resolved to launch three products: the first product, the second product, and the third product. The availability of devices that might restrict production is summarized below.

Type of Machine	Available time (hours of Machine in a month)
Throwing Machine	(124,178,320)
Lathe	(495,741,1222)
Grinder	(890,1349,2164)

The number of machine hours needed for each specific product unit is provided below.

Rate of product (in hours of Machine per unit)

Type of Machine	Product 1	Product 2	Product 3
Throwing Machine	(4,2,6)	(12,12,14)	(18,16,20)
Lathe	(12,10,14)	(45,45,50)	(78,74,80)
Grinder	(18,16,18)	(78,75,80)	(146,146,150)

Now, we need to figure out how much of each product should be produced to fully utilize all the available time.

To illustrate the above problem as a fully fuzzy linear system, we designate Z_1 as the quantity of product 1 manufactured throughout the month. Z_2 and Z_3 denote the totals of products 2 and 3, respectively.

The fully fuzzy linear system related to the problem mentioned above is

$$\begin{pmatrix} (4,2,6) & (12,12,14) & (18,16,20) \\ (12,10,14) & (45,45,50) & (78,74,80) \\ (18,16,18) & (78,75,80) & (146,146,150) \end{pmatrix} \begin{pmatrix} \hat{z}_1 \\ \hat{z}_2 \\ \hat{z}_3 \end{pmatrix} = \begin{pmatrix} (124,178,320) \\ (495,741,1222) \\ (890,1349,2164) \end{pmatrix}$$

Or

$$(4,2,6) \otimes (x_1,y_1,z_1) \oplus (12,12,14) \otimes (x_2,y_2,z_2) \oplus (18,16,20) \otimes (x_3,y_3,z_3) = (124,178,320)$$

$$(12,10,14) \otimes (x_1,y_1,z_1) \oplus (45,45,50) \otimes (x_2,y_2,z_2) \oplus (78,74,80) \otimes (x_3,y_3,z_3) = (495,741,1222)$$

$$(18,16,18) \otimes (x_1,y_1,z_1) \oplus (78,75,80) \otimes (x_2,y_2,z_2) \oplus (146,146,150) \otimes (x_3,y_3,z_3) = (890,1349,2164)$$

The exact solutions for the fuzzy linear system from the equations $Z = S^{-1}G$ are

$(x_1,y_1,z_1) = (4,1,3)$, $(x_2,y_2,z_2) = (3,4,6)$ and $(x_3,y_3,z_3) = (4,1,5)$

Similarly to Problem (1), to solve Problem (2), iterations continue until the approximately optimal solution is obtained, which occurs in the eighth step, demonstrating rapid convergence, as explained in Table 2.

Table 2 - Numerical results for the optimal approximate solution of Problem (2) using algorithm (BGM) SI step-by-step.

No. of iterations(n)	Residual Norm	$\hat{z}_1 = (x_1, y_1, z_1)$	$\hat{z}_2 = (x_2, y_2, z_2)$	$\hat{z}_3 = (x_3, y_3, z_3)$
$n = 0$	3.91529E+02	(0.4533, 0.6507, 1.1698)	(1.8095, 2.7087, 4.4670)	(3.2534, 4.9313, 7.9105)
$n = 1$	3.88076E+01	(0.6826, 0.3751, 1.1474)	(2.6279, 1.4527, 3.8780)	(4.6361, 2.4216, 6.4216)
$n = 2$	4.95695E+00	(1.6642, 1.7104, 4.2498)	(3.6894, 3.7243, 7.2110)	(3.9307, 1.0257, 4.1761)
$n = 3$	2.55683E+00	(2.2218, 0.3932, 3.7029)	(4.2568, 2.7973, 5.6737)	(3.5480, 1.7183, 5.0781)
$n = 4$	2.00976E-01	(3.1815,-1.5485, 26.6075)	(3.5959, 5.0308,-11.1737)	(3.7825, 0.7647, 11.2626)
$n = 5$	4.80962E-03	(4.0000, 1.0000, 3.0000)	(3.0000, 4.0000, 5.9999)	(3.9999, 0.9999, 4.9999)
$n = 6$	3.66167E-04	(4.0000, 1.0000, 3.0000)	(3.0000, 4.0000, 6.0000)	(4.0000, 1.0000, 5.0000)
$n = 7$	2.81369E-10	(4, 1, 3)	(3, 4, 6)	(4, 1, 5)

It is clear that the sequence of iterations converges to a complete and exact solution.

$(x_1,y_1,z_1) = (4,1,3)$, $(x_2,y_2,z_2) = (3,4,6)$ and $(x_3,y_3,z_3) = (4,1,5)$

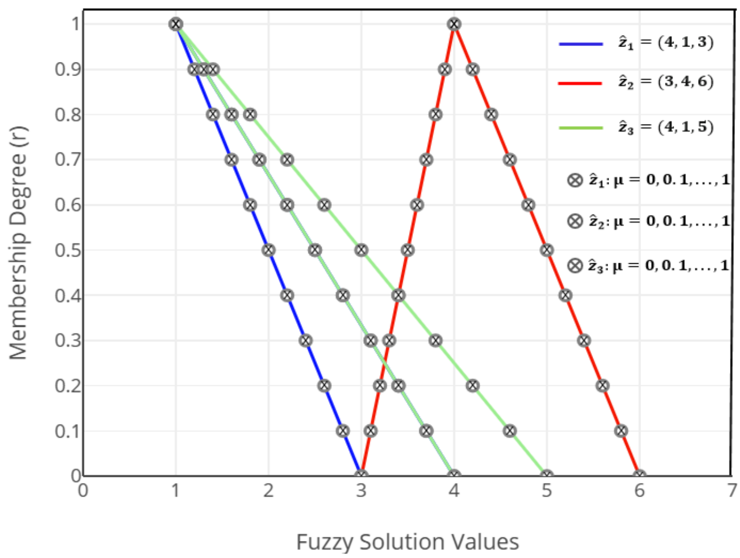

Fig. 3 - Shows the graphical representation of the fuzzy solution ($\hat{z}_1, \hat{z}_2, \hat{z}_3$) for problem (2), using the bi-conjugate gradient method (BGM) ^{SI}.

Problem (3): Examine the totally fuzzy linear system of equations that follows:

$$(4,3,2) \otimes (x_1, y_1, z_1) \oplus (5,2,1) \otimes (x_2, y_2, z_2) \oplus (3,0,3) \otimes (x_3, y_3, z_3) = (71,54,76)$$

$$(7,4,3) \otimes (x_1, y_1, z_1) \oplus (10,6,5) \otimes (x_2, y_2, z_2) \oplus (2,1,1) \otimes (x_3, y_3, z_3) = (118,115,129)$$

$$(6,2,2) \otimes (x_1, y_1, z_1) \oplus (7,1,2) \otimes (x_2, y_2, z_2) \oplus (15,5,4) \otimes (x_3, y_3, z_3) = (155,89,151)$$

The exact solutions for the fuzzy linear system from the equations $Z = S^{-1}G$ are

$(x_1, y_1, z_1) = (4, 2, 2)$, $(x_2, y_2, z_2) = (8, 3, 5)$ and $(x_3, y_3, z_3) = (5, 1, 4)$

Similarly to Problem (1), to solve Problem (3), iterations continue until the approximately optimal solution is obtained, which occurs in the sixth step, demonstrating rapid convergence, as explained in Table 3.

Table 3 - Numerical results for the optimal approximate solution of Problem (3) using algorithm (BGM) ^{SI} step-by-step.

No. of iterations(n)	Residual Norm	$\hat{z}_1 = (x_1, y_1, z_1)$	$\hat{z}_2 = (x_2, y_2, z_2)$	$\hat{z}_3 = (x_3, y_3, z_3)$
$n = 0$	7.46015E+01	(2.7852, 2.1183, 2.9813)	(4.6289, 4.5112, 5.0604)	(6.0803, 3.4913, 5.9234)
$n = 1$	1.95086E+01	(3.8240, 1.5576, 2.1755)	(7.0143, 4.3023, 5.1532)	(6.5406, -0.2394, 3.2277)
$n = 2$	1.05129E+01	(3.9747, 1.6659, 1.9037)	(7.8941, 4.0687, 5.4115)	(5.1436, 0.1006, 3.6099)
$n = 3$	1.86166E+01	(3.9537, 1.7644, 1.9821)	(7.7927, 4.6520, 5.6234)	(5.2807, -0.6298, 3.2903)
$n = 4$	1.54351E-03	(3.9999, 1.9977, 2.0037)	(8.0000, 3.0016, 4.9974)	(5.0000, 1.0001, 3.9997)
$n = 5$	4.45989E-10	(4, 2, 2)	(8, 3, 5)	(5, 1, 4)

It is clear that the sequence of iterations converges to a complete and exact solution.

$(x_1, y_1, z_1) = (4, 2, 2)$, $(x_2, y_2, z_2) = (8, 3, 5)$ and $(x_3, y_3, z_3) = (5, 1, 4)$

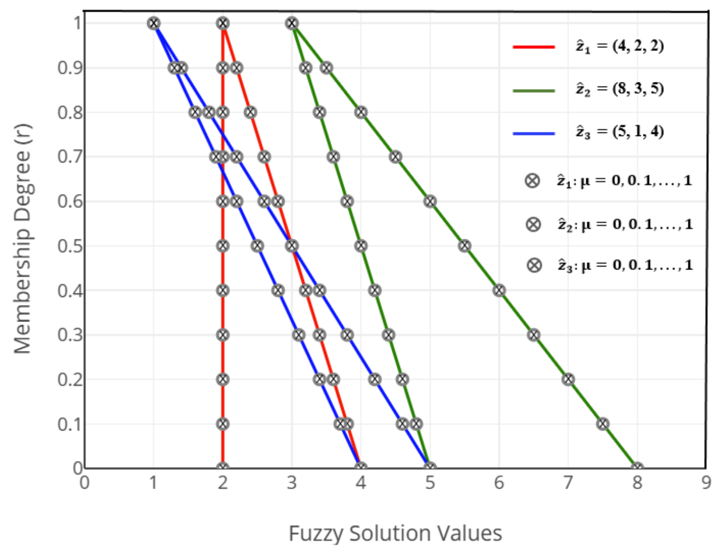

Fig. 4 - Shows the graphical representation of the fuzzy solution ($\hat{z}_1$, $\hat{z}_2$, $\hat{z}_3$) for problem (3), using the bi-conjugate gradient method (BGM) ^{SI}.

5. Result And Discussion

In the first experiment model, the problem is solved using the new algorithm of the Bi-conjugate gradient method (BGM) ^{SI}, which converges to a solution for all variables within four iterations. As in Table 1, all the iterations and results are shown, and Figure 2 shows the solution to the problem. In contrast, when employing alternative iterative methods such as the Gauss-Jacobi method, the convergence rates vary significantly depending on the variable set. In particular, 20 iterations are required to reach convergence for the variables X_1 and X_2, 10 iterations are required for Y_1 and Y_2 to converge to a solution; and 12 iterations are required for Z_1 and Z_2. Likewise, the Gaussian Saddle method's convergence behavior varies depending on the set. Ten iterations are needed for X_1 and X_2, while only four are needed for Y_1 and Y_2. For Z_1 and Z_2 in [18], the Gaussian Saddle method converges in 7 iterations. It takes eight iterations to solve the new algorithm of the (BGM) ^{SI} for the second problem, and six iterations for all variables to solve the third problem. Other approaches, on the other hand, need to solve this issue through more iterations. As in Tables 2 and 3, all the iterations and results are shown, and Figures 3 and 4 show the solution to the problem. The numerical experiments show that the proposed approach for solving positive fully fuzzy linear systems (FFLS) with the Bi-conjugate gradient method (BGM) ^{SI} shows considerable efficiency and accuracy. The method converts the FFLS into a linear system $SZ = G$ and builds a block matrix S from the fuzzy coefficient matrices using the iterative (BGM) ^{SI} algorithm to achieve fast convergence. In the first experiment model, a 2 x 2 FFLS was solved in 4 iterations with a residual norm of the order of 10^{-12} to the solution $(x_1, y_1, z_1) = (4, 0.0909, 0)$ and $(x_2, y_2, z_2) = (5, 0.0909, 0.5)$. In the second experiment model, a 3 x 3 FFLS in 8 iterations with a residual norm of 10^{-10} 8 iterations produced the solution $(x_1, y_1, z_1) = (4, 1, 3)$, $(x_2, y_2, z_2) = (3, 4, 6)$ and $(x_3, y_3, z_3) = (4, 1, 5)$. In the third experiment model, a 3 x 3 FFLS in 6 iterations with a residual norm of 10^{-10} 6 iterations produced the solution $(x_1, y_1, z_1) = (4, 2, 2)$, $(x_2, y_2, z_2) = (8, 3, 5)$, and $(x_3, y_3, z_3) = (5, 1, 4)$. It is clear from the previously mentioned results that the Bi-conjugate gradient method (BGM) outperforms the Gauss-Jacobi and Gaussian Saddle methods in terms of efficiency and convergence time. In relation to other methods studied, the Bi-conjugate gradient method (BGM) ^{SI} is preferable because the convergence was reached with less effort (at most four iterations for all variables). Its capability of solving large, sparse, non-symmetric systems without having to perform a matrix inversion speaks for itself. This method, when compared to conventional approaches that necessitate some forms of drastic simplifications or transformations, is able to provide strong and scalable solutions for FFLS problems that are very highly non-linear. Additionally, the (BGM) ^{SI} algorithm's scope of iteration allows for the solution of larger problems, which is advantageous in operations research, engineering, and economics. In general, this method can be useful for both academic research and solving fuzzy system problems in the real world.

6. Conclusion

One of the newest and most effective approaches to solving fully fuzzy linear systems (FFLS) with fuzzy undefined constants is the one being presented here. Using the Bi-conjugate gradient method $(BGM)^{SI}$. The $(BGM)^{SI}$ method approaches solving the system by ensuring positivity and insensitivity to uncertainties by transforming the system into a block matrix system. Some of the respondents expressed that is most useful in practical life, particularly in engineering fields that require the application of fuzzy systems. The algorithm's iterative nature makes it suitable for larger systems where uncertainty can be modeled with triangular fuzzy numbers. Preconditioning is one direction of potential future work to increase computational efficiency.

References

[1] M. Friedman, M. Ming, and A. Kandel, "Fuzzy linear systems," *Fuzzy sets and systems,* vol. 96, no. 2, pp. 201-209, 1998.

[2] S. Abbasbandy, A. Jafarian, and R. Ezzati, "Conjugate gradient method for fuzzy symmetric positive definite system of linear equations," *Applied mathematics and computation,* vol. 171, no. 2, pp. 1184-1191, 2005.

[3] M. Dehghan, B. Hashemi, and M. Ghatee, "Computational methods for solving fully fuzzy linear systems," *Applied mathematics and computation,* vol. 179, no. 1, pp. 328-343, 2006.

[4] M. Dehghan, B. Hashemi, and M. Ghatee, "Solution of the fully fuzzy linear systems using iterative techniques," *Chaos, Solitons & Fractals,* vol. 34, no. 2, pp. 316-336, 2007.

[5] T. Allahviranloo, S. Salahshour, and M. Khezerloo, "Maximal-and minimal symmetric solutions of fully fuzzy linear systems," *Journal of Computational and Applied Mathematics,* vol. 235, no. 16, pp. 4652-4662, 2011.

[6] A. Kumar, Neetu, and A. Bansal, "A new approach for solving fully fuzzy linear systems," *Advances in Fuzzy Systems,* vol. 2011, no. 1, p. 943161, 2011.

[7] A. Kumar, Neetu, and A. Bansal, "A new computational method for solving fully fuzzy linear systems of triangular fuzzy numbers," *Fuzzy Information and Engineering,* vol. 4, no. 1, pp. 63-73, 2012.

[8] A. Kumar, A. Bansal, and N. Babbar, "Fully fuzzy linear systems of triangular fuzzy numbers (a, b, c)," *International Journal of Intelligent Computing and Cybernetics,* vol. 6, no. 1, pp. 21-44, 2013.

[9] M. Otadi, M. Mosleh, and S. Abbasbandy, "Numerical solution of fully fuzzy linear systems by fuzzy neural network," *Soft computing,* vol. 15, no. 8, pp. 1513-1522, 2011.

[10] K. Dookhitram, M. Sunhaloo, N. Rambeerich, A. Peer, and A. Saib, "A Preconditioning algorithm for the positive solution of fully fuzzy linear system," *Journal of Fuzzy Set Valued Analysis, Article ID jfsva-00123,* 2012.

[11] R. Ezzati, S. Khezerloo, and A. Yousefzadeh, "Solving fully fuzzy linear system of equations in general form," *Journal of Fuzzy Set Valued Analysis,* vol. 2012, pp. 1-11, 2012.

[12] G. Malkawi, N. Ahmad, H. Ibrahim, and B. Alshmari, "Row reduced echelon form for solving fully fuzzy system with unknown coefficients," *Journal of Fuzzy Set Valued Analysis,* vol. 2014, pp. 1-18, 2014.

[13] A. A. Elsayed, N. Ahmad, and G. Malkawi, "Numerical solutions for coupled trapezoidal fully fuzzy sylvester matrix equations," *Advances in Fuzzy Systems,* vol. 2022, no. 1, p. 8926038, 2022.

[14] G. Malkawi, N. Ahmad, and H. Ibrahim, "Solving fully fuzzy linear system with the necessary and sufficient condition to have a positive solution," *Applied Mathematics & Information Sciences,* vol. 8, no. 3, p. 1003, 2014.

[15] G. Malkaw, N. Ahmad, and H. Ibrahim, "An algorithm for a positive solution of arbitrary fully fuzzy linear system," *Computational Mathematics and modeling,* vol. 26, pp. 436-465, 2015.

[16] G. O. Malkawi and H. Y. Alfifi, "The consistency of positive fully fuzzy linear system," in *AIP Conference Proceedings*, 2017, vol. 1905, no. 1: AIP Publishing.

[17] E. Abdolmaleki and S. Edalatpanah, "Fast iterative method (FIM) for solving fully fuzzy linear systems," *Information Sciences and Computing,* no. 1, 2014.

[18] L. Ineirat, A. Daraghmeh, and N. Qatanani, "Iterative Techniques for Solving Positive Triangular Fully Fuzzy Linear System."

[19] G. F. Khalili, S. H. Nasseri, and N. A. Taghi-Nezhad, "A new interactive approach for solving fully fuzzy mixed integer linear programming," *Yugoslav journal of operations research,* vol. 30, no. 1, pp. 71-89, 2020.

[20] S. J. Ghoushchi, E. Osgooei, G. Haseli, and H. Tomaskova, "A novel approach to solve fully fuzzy linear programming problems with modified triangular fuzzy numbers," *Mathematics,* vol. 9, no. 22, p. 2937, 2021.

[21] R. J. Mitlif, "An efficient algorithm for fuzzy linear fractional programming problems via ranking function," *Baghdad Science Journal,* vol. 19, no. 1, pp. 0071-0071, 2022.

[22] N. Mikaeilvand, Z. Noeiaghdam, S. Noeiaghdam, and J. J. Nieto, "A novel technique to solve the fuzzy system of equations," *Mathematics,* vol. 8, no. 5, p. 850, 2020.

[23] M. Ghanbari, T. Allahviranloo, and W. Pedrycz, "A straightforward approach for solving dual fuzzy linear systems," *Fuzzy Sets and Systems,* vol. 435, pp. 89-106, 2022.

[24] F. Abbasi and T. Allahviranloo, "Solving fully fuzzy linear system: a new solution concept," *Information Sciences,* vol. 589, pp. 608-635, 2022.

[25] F. Babakordi and T. Allahviranloo, "A cramer method for solving fully fuzzy linear systems based on transmission average," *Control and Optimization in Applied Mathematics,* vol. 7, no. 2, pp. 115-130, 2022.

[26] D. Behera and S. Chakraverty, "Computational technique for solving imprecisely defined non-negative fully fuzzy algebraic system of linear equations," *International Journal of Fuzzy Logic and Intelligent Systems,* vol. 22, no. 3, pp. 252-260, 2022.