Intern. J. Fuzzy Mathematical Archive
Vol. 20, No. 1, 2022, 27-43
ISSN: 2320 –3242 (P), 2320 –3250 (online)
Published on 31 August 2022
www.researchmathsci.org
DOI: http://dx.doi.org/10.22457/ijfma.v20n1a03237

Solution of Fuzzy System of Linear Equations using Successive Over-Relaxation Method

Shuvam Saren

Department of Applied Mathematics with Oceanology and Computer Programming
Vidyasagar University, Midnapore-721102, West Bengal, India

Received 30 July 2022; accepted 30 August 2022

Abstract. In this paper, we study the Successive Over-Relaxation (SOR) method to solve a fuzzy system of linear equations (FSLE). This method which is followed by the convergence theorem is discussed in detail and the useful iteration scheme is derived. The applicability of this iteration scheme is shown through supporting theorems and some numerical examples are illustrated by using this iteration scheme.

Keywords: Fuzzy number, parametric form of fuzzy number, fuzzy arithmetic, solution of the FSLE, SOR method.

AMS Mathematics Subject Classification (2010): 46B85, 55M20

1. Introduction

Systems of linear equations have a major contribution in various fields such as mathematics, physics, statistics, economics, engineering and social sciences. Since in many applications at least some of the system's parameters and measurements are represented by fuzzy numbers rather than crisp numbers, it is important to develop mathematical models and numerical procedures that would appropriately treat general fuzzy linear systems and solve them. Fuzzy systems of linear equations with crisp coefficients and fuzzy right-hand side vectors are mathematical models that combine crispness and fuzziness in a linear system of equations. The presence of fuzziness in the right-hand side vector introduces uncertainty and imprecision into the problem, making it more challenging to solve. Traditional methods for solving crisp linear systems are not directly applicable to fuzzy systems.

The concept of fuzzy numbers and arithmetic operations with these numbers was first introduced and investigated by Zadeh. Dubois and Prade investigated various operations on fuzzy numbers. A general model for solving an $n \times n$ FSLE in which the coefficient matrix is crisp and the right-hand side is an arbitrary fuzzy number vector was first proposed by Friedman et al. [1]. Allahviranloo [3, 5] investigated various numerical methods for solving FSLE, which are based on numerical iterative methods such as Jacobi's method, Gauss-Seidal's method and the SOR method. Senthilkumar and Rajendran [12] proposed some algorithms to solve fuzzy linear systems. Inearat and Qatanani [10] discussed the efficiency of fuzzy solutions in these iterative methods.

Different authors have investigated and invented different procedures to solve fuzzy systems of linear equations. Cong-Xing and Ming [7] introduced the embedding method for fuzzy number space. Friedman et al. [1, 8, 15] proposed a general model to solve a fuzzy system of linear equations by using this embedding approach. Wang et al. invented an iterative method for solving a system of linear equations of the form $X = AX + B$. Asady et al. developed different methods of a general fuzzy system using embedding concepts. Vroman et al. [17] solved the general fuzzy linear systems using a parametric form of a fuzzy number. Ezzati [20] developed a new method for solving fuzzy linear systems by using the embedding method and replaced a $n \times n$ fuzzy linear system with two $n \times n$ crisp linear systems. Dehghan and Hashemi [9] investigated various iterative methods for solving FSLE. For fuzzy linear systems, Wang and Zheng [18] developed block iterative methods and Miao et al. [13] developed block SOR methods.

In this paper, we focus on developing a solution methodology for fuzzy systems of linear equations using the Successive Overrelaxation (SOR) method. The SOR method is an iterative numerical technique commonly used for solving crisp linear systems of equations. This paper aims to extend the applicability of the SOR method to solve fuzzy systems of linear equations and provide an efficient and accurate solution. To do so, theoretical and conceptual assistance are taken from various textbooks [2, 4, 21]. By addressing the challenges posed by fuzziness in the right-hand side vector, we strive to provide an effective and efficient solution that can enhance the analysis and decision-making processes in fuzzy systems.

The paper is organized as follows. In Section 1, we introduce the paper and its objective. In Section 2, we recall the preliminary concepts required to develop the result. In Section 3, we discuss the SOR Method in the crisp system of linear equations and the implementation of the SOR Method in FSLE to derive the proposed iteration scheme. In Section 4, we illustrate some numerical examples by using this iteration scheme, varying degrees of fuzziness and system sizes. At last in Section 5, we discuss the conclusion of this paper.

2. Preliminaries

In this section, we recall the basic notations of fuzzy numbers, their arithmetic operations, triangular fuzzy numbers, fuzzy systems of linear equations, and some theorems and concepts related to them. We start by defining the fuzzy number.

2.1. Fuzzy number

A fuzzy number is a fuzzy set defined on the set of real numbers $\mathbb{R}$ like $u: \mathbb{R} \to [0,1]$ which satisfies:

1. u is upper semi-continuous, i.e., for a chosen $\epsilon > 0$ there exists $\delta > 0$ such that $u(x) - u(x_0) < \epsilon$ whenever $|x - x_0| < \delta$, $\forall x, x_0 \in \mathbb{R}$,
2. u is fuzzy convex, i.e., $u(\lambda x + (1-\lambda)y) \geq min\{u(x), u(y)\}$, $\forall x, y \in \mathbb{R}$, $\lambda \in [0,1]$,
3. u is normal, i.e., there exists $x_0 \in \mathbb{R}$ for which $u(x_0) = 1$,
4. $supp(u) = \{x \in \mathbb{R}: u(x) > 0\}$ is the support of u, and its closure $cl\big(supp(u)\big)$ is compact.

Parametric form of a fuzzy number

The parametric form of a fuzzy number u is an ordered pair $(\underline{u}, \overline{u})$ of the functions $\left(\underline{u}(r), \overline{u}(r)\right)$, $0 \leq r \leq 1$, which satisfies the following requirements:

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

For example, the fuzzy numbers $(1+2r, 4-r)$ is shown in Figure 1. A crisp number α is simply represented by $\underline{u}(r) = \overline{u}(r) = \alpha$, $0 \leq r \leq 1$.

A fuzzy number

Let E be the set of all real fuzzy numbers which are upper semi-continuous, normal, convex and compactly supported fuzzy sets. By appropriate, the fuzzy number space $\{\underline{u}(r), \overline{u}(r)\}$ becomes a convex cone E which is isomorphically and isometrically into a Banach space.

Triangular fuzzy number

A triangular fuzzy number (TFN) is a fuzzy number which is denoted by a triplet, i.e., $\tilde{A} = (a, b, c)$ and it is defined by its membership function $\mu_{\tilde{A}}(x)$ which is described as follows:

$$\mu_{\tilde{A}}(x) = \begin{cases} 0, \text{ if } x < a \\ \dfrac{x-a}{b-a}, \text{ if } a \leq x \leq b \\ \dfrac{c-x}{c-b}, \text{ if } b \leq x \leq c \\ 0, \text{ if } x > c \end{cases}$$

Triangular fuzzy number (a, b, c)

The graph of the membership function of a triangular fuzzy number is triangular as shown in Figure 2.

Mean-spread representation of TFNs

Triangular fuzzy number $\tilde{A} = (a, b, c)$ can also be written as $\tilde{A} =< m, \alpha, \beta >$, where m is the centre point or mean value, α and β are called left and right spreads respectively. That is, $b = m$, $\alpha = b - a$, $\beta = c - b$.

If $\alpha = \beta$, then TFN is called symmetric.

Triangular fuzzy number in mean-spread representation

Conversion rule between various forms of TFN

The triplet representation of a TFN (a, b, c) can be written as the parametric form $\left(\underline{u}(r), \overline{u}(r)\right) = (a + (b - a)r, c - (c - b)r)$, where $r \in [0,1]$.

The mean-spread representation of a TFN can be written as the parametric form $\left(\underline{u}(r), \overline{u}(r)\right) = \left((m - \alpha) + \alpha r, (m + \beta) - \beta r\right)$, where $r \in [0,1]$.

Conversely, the parametric form of a TFN $(a + \alpha r, c - \beta r)$, where $r \in [0,1]$, can be written as the triplet representation $(a, a + \alpha, c)$ or $(a, c - \beta, c)$. The parametric form of the same can be written as the mean-spread representation $< a + \alpha, \alpha, \beta >$ or $< c - \beta, \alpha, \beta >$.

Identification of a TFN from its parametric form

The parametric form of a fuzzy number does not necessarily indicate it to be a TFN. Let $(a + \alpha r, c - \beta r)$, where $r \in [0,1]$, be the parametric form of an arbitrary fuzzy number. This fuzzy number will be considered to be a TFN if $\alpha + \beta = c - a$.

If $\alpha + \beta \neq c - a$, then this fuzzy number is not a TFN.

Arithmetic of parametric TFNs

Let $x = \left(\underline{x}(r), \overline{x}(r)\right), y = \left(\underline{y}(r), \overline{y}(r)\right) \in E$, $r \in [0,1]$ and arbitrary $k \in \mathbb{R}$. Then

1. $x = y$ if and only if $\underline{x}(r) = \underline{y}(r)$ and $\overline{x}(r) = \overline{y}(r)$
2. $x + y = \left(\underline{x}(r) + \underline{y}(r), \overline{x}(r) + \overline{y}(r)\right)$
3. $x - y = \left(\underline{x}(r) - \overline{y}(r), \overline{x}(r) - \underline{y}(r)\right)$
4. $kx = \begin{cases} \left(k\underline{x}(r), k\overline{x}(r)\right), \text{if} k \geq 0 \\ \left(k\overline{x}(r), k\underline{x}(r)\right), \text{if} k < 0 \end{cases}$

2.2. Fuzzy system of linear equations

In a system of linear equations, fuzziness occurs in many different ways. The right-hand vector Y may be fuzzy, along with coefficient matrix A and variable vector X are crisp. Alternately, A and Y are fuzzy, X is crisp. A is fuzzy, X and Y are crisp. All A, X and Y are fuzzy matrices. An $n \times n$ linear system is $AX = Y$ or it can be written as

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= y_1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= y_2, \\ &\ldots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n &= y_n, \end{aligned} \tag{1}$$

where the coefficient matrix $A = \left(a_{ij}\right), 1 \leq i, j \leq n$ is an $n \times n$ matrix, $X = (x_1, x_2, \ldots, x_n)^t$ and $Y = (y_1, y_2, \ldots, y_n)^t$. Here we consider, A is a crisp matrix while X

and Y are fuzzy matrices, where $x_i \in E$, $y_i \in E$ for $1 \leq i \leq n$. This system of equations is referred to as a fuzzy system of linear equations.

2.3. Solution Methodology of FSLE

A fuzzy number vector $(x_1, x_2, \dots, x_n)^t$ is given by $x_i = \left(\underline{x_i}(r), \overline{x_i}(r)\right)$, $1 \leq i \leq n$, $0 \leq r \leq 1$, is said to be a solution of the FSLE (1) if

$$\underline{\sum_{j=1}^{n} a_{ij}\, x_j} = \sum_{j=1}^{n} a_{ij}\, \underline{x_j} = \underline{y_j},$$
$$\overline{\sum_{j=1}^{n} a_{ij}\, x_j} = \sum_{j=1}^{n} a_{ij}\, \overline{x_j} = \overline{y_j}.$$

Let us consider the ith equation of the system ([FSLE]) be:

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{ii}x_i + \cdots + a_{in}x_n = y_i$$

That is, $a_{i1}\left(\underline{x_1}, \overline{x_1}\right) + a_{i2}\left(\underline{x_2}, \overline{x_2}\right) + \ldots + a_{ii}\left(\underline{x_i}, \overline{x_i}\right) + \ldots + a_{in}\left(\underline{x_n}, \overline{x_n}\right) = \left(\underline{y_i}(r), \overline{y_i}(r)\right)$

Then we obtain:

$$\begin{aligned} a_{i1}\underline{x_1} + a_{i2}\underline{x_2} + \ldots + a_{ii}\underline{x_i} + \ldots + a_{in}\underline{x_n} &= \underline{y_i}(r), \\ a_{i1}\overline{x_1} + a_{i2}\overline{x_2} + \ldots + a_{ii}\overline{x_i} + \ldots + a_{in}\overline{x_n} &= \overline{y_i}(r), \\ 1 \leq i \leq n, 0 \leq r &\leq 1. \end{aligned}$$

From the above system, we have two crisp $n \times n$ linear systems for all i that can be extended to $(2n) \times (2n)$ crisp linear system as follows:

$$SX = Y$$

This can be written as

$$\begin{bmatrix} S_1 & S_2 \\ S_2 & S_1 \end{bmatrix} \begin{bmatrix} \underline{X} \\ \overline{X} \end{bmatrix} = \begin{bmatrix} \underline{Y} \\ \overline{Y} \end{bmatrix} \tag{2}$$

where S is the extended $(2n) \times (2n)$ matrix, S_1 and S_2 are $n \times n$ matrices with elements being the corresponding non-negative and non-positive elements of the matrix A respectively, $\underline{X} = \left(\underline{x_1}, \underline{x_2}, \dots, \underline{x_n}\right)^t$, $\overline{X} = (\overline{x_1}, \overline{x_2}, \dots, \overline{x_n})^t$, $\underline{Y} = \left(\underline{y_1}, \underline{y_2}, \dots, \underline{y_n}\right)^t$, $\overline{Y} = (\overline{y_1}, \overline{y_2}, \dots, \overline{y_n})^t$.

Thus the FSLE (2) can be extended to a crisp system of linear equation, where $A = S_1 + S_2$.

Therefore, equation (2) can be written as follows:

$$S_1\underline{X} + S_2\overline{X} = \underline{Y}$$
$$S_2\underline{X} + S_1\overline{X} = \overline{Y} \tag{3}$$

If we observe equation (3) in details, we can expend it in the $(2n) \times (2n)$ linear system:

$$s_{11}\underline{x_1} + s_{12}\underline{x_2} + \ldots + s_{1n}\underline{x_n} + s_{1,n+1}\overline{x_1} + s_{1,n+2}\overline{x_2} + \ldots + s_{1,2n}\overline{x_n} = \underline{y_1},$$
$$s_{21}\underline{x_1} + s_{22}\underline{x_2} + \ldots + s_{2n}\underline{x_n} + s_{2,n+1}\overline{x_1} + s_{2,n+2}\overline{x_2} + \ldots + s_{2,2n}\overline{x_n} = \underline{y_2},$$
$$\ldots$$
$$s_{n1}\underline{x_1} + s_{n2}\underline{x_2} + \ldots + s_{nn}\underline{x_n} + s_{n,n+1}\overline{x_1} + s_{n,n+2}\overline{x_2} + \ldots + s_{n,2n}\overline{x_n} = \underline{y_n},$$
$$s_{n+1,1}\underline{x_1} + s_{n+1,2}\underline{x_2} + \ldots + s_{n+1,n}\underline{x_n} + s_{n+1,n+1}\overline{x_1} + s_{n+1,n+2}\overline{x_2} + \ldots + s_{n+1,2n}\overline{x_n} = \overline{y_1},$$
$$s_{n+2,1}\underline{x_1} + s_{n+2,2}\underline{x_2} + \ldots + s_{n+2,n}\underline{x_n} + s_{n+2,n+1}\overline{x_1} + s_{n+2,n+2}\overline{x_2} + \ldots + s_{n+2,2n}\overline{x_n} = \overline{y_2},$$
$$\ldots$$
$$s_{2n,1}\underline{x_1} + s_{2n,2}\underline{x_2} + \ldots + s_{2n,n}\underline{x_n} + s_{2n,n+1}\overline{x_1} + s_{2n,n+2}\overline{x_2} + \ldots + s_{2n,2n}\overline{x_n} = \overline{y_n},$$

where s_{ij} is determined as follows:

$$a_{ij} \geq 0 \text{ then } s_{ij} = a_{ij}, s_{i+n,j+n} = a_{ij},$$
$$a_{ij} \leq 0 \text{ then } s_{i,j+n} = a_{ij}, s_{i+n,j} = a_{ij},$$
$$\text{for } 1 \leq i \leq n, 1 \leq j \leq n.$$

Therefore, in this way, using matrix notation we obtain,

$$SX = Y, \text{ where } S = (s_{ij}), 1 \leq i, j \leq 2n,$$

$$X = [\overline{x_1}, \overline{x_2}, \ldots, \overline{x_n}, \underline{x_1}, \underline{x_2}, \ldots, \underline{x_n}]^t$$

and

$$Y = [\overline{y_1}, \overline{y_2}, \ldots, \overline{y_n}, \underline{y_1}, \underline{y_2}, \ldots, \underline{y_n}]^t.$$

Theorem 1. *A matrix S is non-singular if and only if the matrices $S_1 + S_2$ and $S_1 - S_2$ are both non-singular.*

Let S be a non-singular $(2n) \times (2n)$ matrix of the form $S = \begin{bmatrix} S_1 & S_2 \\ S_2 & S_1 \end{bmatrix}$, where S_1 and S_2 are $n \times n$ matrices with elements being the corresponding non-negative and non-positive elements of the matrix A respectively.

By adding the $(n+i)$th row of S to its ith row for $1 \leq i \leq n$, we obtain

$$S = \begin{bmatrix} S_1 + S_2 & S_1 + S_2 \\ S_2 & S_1 \end{bmatrix} = B(\text{say})$$

Next, we subtract the jth column of S, from its $(n+j)$th column for $1 \leq j \leq n$, we obtain

$$S = \begin{bmatrix} S_1 + S_2 & S_1 + S_2 \\ S_2 & S_1 \end{bmatrix} = \begin{bmatrix} S_1 + S_2 & 0 \\ S_2 & S_1 - S_2 \end{bmatrix} = C(\text{say})$$

It is clear that, $|S| = |B| = |C| = |S_1 + S_2||S_1 - S_2|$.

Therefore $|S| \neq 0$ if and only if $|S_1 + S_2| \neq 0$ and $|S_1 - S_2| \neq 0$.

This completes the proof.

2.4. Fuzzy Solution of FSLE

Let $X = \{\left(\underline{x_i}(r), \overline{x_i}(r)\right), 1 \leq i \leq n\}$ denote the unique solution of the FSLE. The fuzzy number vector $U = \{\left(\underline{u_i}(r), \overline{u_i}(r)\right), 1 \leq i \leq n\}$ defined by

$$\underline{u_i}(r) = min\{\underline{x_i}(r), \overline{x_i}(r), \underline{x_i}(1)\}$$
$$\overline{u_i}(r) = max\{\underline{x_i}(r), \overline{x_i}(r), \underline{x_i}(1)\}$$

is called the fuzzy solution of $SX = Y$.

If $\left(\underline{x_i}(r), \overline{x_i}(r)\right), 1 \leq i \leq n$, are all triangular fuzzy numbers, then $\underline{u_i}(r) = \underline{x_i}(r)$, $\overline{u_i}(r) = \overline{x_i}(r)$, $1 \leq i \leq n$ and U is called a strong fuzzy solution.

Otherwise, U is a weak fuzzy solution.

3. Solution of FSLE using the SOR method

In this section, we study the modified version of the Gauss-Seidel iteration method known as the Successive Over-Relaxation (SOR) method. We also study the implementation of this method in FSLE, the derivation of the iteration scheme to obtain the solution of FSLE using the SOR method and some theorems related to it. We start by describing the SOR method in a crisp system of linear equations and then we extend this to FSLE.

3.1. SOR method in crisp system of linear equations

Assume that all numbers a_{ij}, y_i are real for all $1 \leq i, j \leq n$. Then the ith equation $\sum_{j=1}^{n} a_{ij}\, x_j = y_j$, $i = 1,2,\ldots,n$, can be written as

$$\sum_{j=1}^{i-1} a_{ij}\, x_j + \sum_{j=i}^{n} a_{ij}\, x_j = y_i \qquad (4)$$

Like Gauss-Seidel iteration method, for the solution

$$\left(x_1^{(k+1)}, x_2^{(k+1)}, \ldots, x_{i-1}^{(k+1)}, x_i^{(k)}, x_{i+1}^{(k)}, \ldots, x_n^{(k)}\right)$$

the equation (4) becomes

$$\sum_{j=1}^{i-1} a_{ij}\, x_j^{(k+1)} + \sum_{j=i}^{n} a_{ij}\, x_j^{(k)} = y_i \qquad (5)$$

The residual at the ith equation is then can be computed as

$$r_i = y_i - \sum_{j=1}^{i-1} a_{ij}\, x_j^{(k+1)} - \sum_{j=i}^{n} a_{ij}\, x_j^{(k)} \tag{6}$$

Let $d_i^{(k)} = x_i^{(k+1)} - x_i^{(k)}$ denote the differences of x_i at two consecutive.
In the Successive Over-Relaxation (SOR) method, we assume that

$$a_{ii}\, d_i^{(k)} = w\, r_i, \quad i = 1,2,\dots,n \tag{7}$$

where w is a suitable factor, which is known as the relaxation factor.
Using (6) in (7) we obtain,

$$a_{ii}\left(x_i^{(k+1)} - x_i^{(k)}\right) = w\left(y_i - \sum_{j=1}^{i-1} a_{ij}\, x_j^{(k+1)} - \sum_{j=i}^{n} a_{ij}\, x_j^{(k)}\right)$$

$$\text{or,} a_{ii}x_i^{(k+1)} - a_{ii}x_i^{(k)} = wy_i - w\sum_{j=1}^{i-1} a_{ij}\, x_j^{(k+1)} - w\sum_{j=i}^{n} a_{ij}\, x_j^{(k)}$$

That is

$$a_{ii}x_i^{(k+1)} + w\sum_{j=1}^{i-1} a_{ij}\, x_j^{(k+1)} = (1-w)a_{ii}x_i^{(k)} - w\sum_{j=i+1}^{n} a_{ij}\, x_j^{(k)} + wy_i, \quad i = 1,2,\dots,n;\, k = 0,1,2,\dots \tag{8}$$

and $\left(x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}\right)^t$ is the initial solution.
The method is repeated until the desired accuracy is achieved.

The method is called the over-relaxation method when $1 < w < 2$, and is called the under-relaxation method when $0 < w < 1$. When $w = 1$, the method becomes the Gauss-Seidal iteration method.

3.2. Implementation of the SOR method in FSLE

Recalling the concept of FSLE discussed in the preliminary section, we that in the equation (8), $\{(a_{ij}): 1 \le i \le n, 1 \le j \le n\}$ are crisp numbers and $y_i \in E$, i.e., y_i are real fuzzy numbers for $i = 1,2,\dots,n$.

Without loss of generality, we can extend the equation to and we can convert equation (8) in the matrix form as follows

$$DX^{(k+1)} + wLX^{(k+1)} = (1-w)DX^{(k)} - wUX^{(k)} + wY$$

$$\text{or,} (D + wL)X^{(k+1)} = (1-w)DX^{(k)} - wUX^{(k)} + wY \tag{9}$$

where $D = \begin{bmatrix} D_1 & 0 \\ 0 & D_1 \end{bmatrix}$ with $D_1 = diag(a_{11}, a_{22}, \ldots, a_{nn})$, L is strictly lower triangular matrix defined as $L = \begin{bmatrix} L_1 & 0 \\ S_2 & L_1 \end{bmatrix}$ with L_1 being strictly lower triangular matrix, U is strictly upper triangular matrix defined as $U = \begin{bmatrix} U_1 & S_2 \\ 0 & U_1 \end{bmatrix}$ with U_1 being strictly upper triangular matrix, $X^{(k+1)} = \begin{bmatrix} \underline{X}^{(k+1)} \\ \overline{X}^{(k+1)} \end{bmatrix}$, $X^{(k)} = \begin{bmatrix} \underline{X}^{(k)} \\ \overline{X}^{(k)} \end{bmatrix}$ and $Y = \begin{bmatrix} \underline{Y} \\ \overline{Y} \end{bmatrix}$.

Let $S_1 = D_1 + L_1 + U_1$ and such that $S = \begin{bmatrix} S_1 & S_2 \\ S_2 & S_1 \end{bmatrix}$.

Now, $D + wL = \begin{bmatrix} D_1 & 0 \\ 0 & D_1 \end{bmatrix} + w \begin{bmatrix} L_1 & 0 \\ S_2 & L_1 \end{bmatrix} = \begin{bmatrix} D_1 + wL_1 & 0 \\ wS_2 & D_1 + wL_1 \end{bmatrix}$

Now, from equation (9), it follows

$$\begin{bmatrix} D_1 + wL_1 & 0 \\ wS_2 & D_1 + wL_1 \end{bmatrix} \begin{bmatrix} \underline{X}^{(k+1)} \\ \overline{X}^{(k+1)} \end{bmatrix} = (1-w) \begin{bmatrix} D_1 & 0 \\ 0 & D_1 \end{bmatrix} \begin{bmatrix} \underline{X}^{(k)} \\ \overline{X}^{(k)} \end{bmatrix} - w \begin{bmatrix} U_1 & S_2 \\ 0 & U_1 \end{bmatrix} \begin{bmatrix} \underline{X}^{(k)} \\ \overline{X}^{(k)} \end{bmatrix} + w \begin{bmatrix} \underline{Y} \\ \overline{Y} \end{bmatrix}$$

$$\text{or, } \begin{bmatrix} (D_1 + wL_1)\underline{X}^{(k+1)} \\ wS_2\underline{X}^{(k+1)} + (D_1 + wL_1)\overline{X}^{(k+1)} \end{bmatrix} = \begin{bmatrix} (1-w)D_1\underline{X}^{(k)} \\ (1-w)D_1\overline{X}^{(k)} \end{bmatrix} - \begin{bmatrix} wU_1\underline{X}^{(k)} + wS_2\overline{X}^{(k)} \\ wU_1\overline{X}^{(k)} \end{bmatrix} + \begin{bmatrix} w\underline{Y} \\ w\overline{Y} \end{bmatrix}$$

$$\text{or, } \begin{cases} (D_1 + wL_1)\underline{X}^{(k+1)} = (1-w)D_1\underline{X}^{(k)} - wU_1\underline{X}^{(k)} - wS_2\overline{X}^{(k)} + w\underline{Y}, \\ wS_2\underline{X}^{(k+1)} + (D_1 + wL_1)\overline{X}^{(k+1)} = (1-w)D_1\overline{X}^{(k)} - wU_1\overline{X}^{(k)} + w\overline{Y} \end{cases}$$

$$\text{or, } \begin{cases} (D_1 + wL_1)\underline{X}^{(k+1)} = (1-w)D_1\underline{X}^{(k)} - wU_1\underline{X}^{(k)} - wS_2\overline{X}^{(k)} + w\underline{Y}, \\ (D_1 + wL_1)\overline{X}^{(k+1)} = (1-w)D_1\overline{X}^{(k)} - wU_1\overline{X}^{(k)} - wS_2\underline{X}^{(k+1)} + w\overline{Y} \end{cases} \quad (10)$$

From equation (10), it also follows that

$$\begin{bmatrix} (D_1 + wL_1)\underline{X}^{(k+1)} \\ (D_1 + wL_1)\overline{X}^{(k+1)} \end{bmatrix} = \begin{bmatrix} (1-w)D_1\underline{X}^{(k)} - wU_1\underline{X}^{(k)} - wS_2\overline{X}^{(k)} + w\underline{Y} \\ (1-w)D_1\overline{X}^{(k)} - wU_1\overline{X}^{(k)} - wS_2\underline{X}^{(k+1)} + w\overline{Y} \end{bmatrix}$$

That is

$$(D_1 + wL_1)\begin{bmatrix}\underline{X}^{(k+1)}\\ \overline{X}^{(k+1)}\end{bmatrix} = \begin{bmatrix}(1-w)D_1 - wU_1 & -wS_2\\ -wS_2 & (1-w)D_1 - wU_1\end{bmatrix}\begin{bmatrix}\underline{X}^{(k)}\\ \overline{X}^{(k)}\end{bmatrix} + \begin{bmatrix}w\underline{Y}\\ w\overline{Y}\end{bmatrix}$$

or, $$\begin{bmatrix}\underline{X}^{(k+1)}\\ \overline{X}^{(k+1)}\end{bmatrix} = \begin{bmatrix}(D_1 + wL_1)^{-1}[(1-w)D_1 - wU_1] & -(D_1 + wL_1)^{-1}wS_2\\ -(D_1 + wL_1)^{-1}wS_2 & (D_1 + wL_1)^{-1}[(1-w)D_1 - wU_1]\end{bmatrix}\begin{bmatrix}\underline{X}^{(k)}\\ \overline{X}^{(k)}\end{bmatrix} + \begin{bmatrix}(D_1 + wL_1)^{-1}w\underline{Y}\\ (D_1 + wL_1)^{-1}w\overline{Y}\end{bmatrix}$$

Finally,

$$X^{(k+1)} = PX^{(k)} + C \tag{11}$$

where

$$P = \begin{bmatrix}(D_1 + wL_1)^{-1}[(1-w)D_1 - wU_1] & -(D_1 + wL_1)^{-1}wS_2\\ -(D_1 + wL_1)^{-1}wS_2 & (D_1 + wL_1)^{-1}[(1-w)D_1 - wU_1]\end{bmatrix},$$

$$C = \begin{bmatrix}(D_1 + wL_1)^{-1}w\underline{Y}\\ (D_1 + wL_1)^{-1}w\overline{Y}\end{bmatrix}.$$

We note that, matrix forms like equation (10) and equation (11) are used for the sake of analytical proof of the theorems. But, for computational purpose we must express these to iterative scheme.

Thus from equation (10), it follows

$$s_{ii}\underline{x_i}^{(k+1)} + w\sum_{j=1}^{i-1} s_{ij}\,\underline{x_j}^{(k+1)} = (1-w)s_{ii}\underline{x_i}^{(k)} - w\sum_{j=i+1}^{n} s_{ij}\,\underline{x_j}^{(k)} - w\sum_{j=1}^{n} s_{i,j+n}\,\overline{x_j}^{(k)} + w\underline{y_i},$$
$$s_{ii}\overline{x_i}^{(k+1)} + w\sum_{j=1}^{i-1} s_{ij}\,\overline{x_j}^{(k+1)} = (1-w)s_{ii}\overline{x_i}^{(k)} - w\sum_{j=i+1}^{n} s_{ij}\,\overline{x_j}^{(k)} - w\sum_{j=1}^{n} s_{i,j+n}\,\underline{x_j}^{(k)} + w\overline{y_i},$$

where $i = 1,2,\ldots,n;\quad k = 0,1,2,\ldots$

This is the iteration scheme for the solution of the FSLE using the SOR method.
The stopping criterion of this method with tolerance $\epsilon > 0$ is

$$\frac{||\underline{X}^{(k+1)}|| - ||\underline{X}^{(k)}||}{||\underline{X}^{(k)}||} < \epsilon, \quad \frac{||\overline{X}^{(k+1)}|| - ||\overline{X}^{(k)}||}{||\overline{X}^{(k)}||} < \epsilon, \quad k = 0, 1, 2, \ldots$$

We will illustrate some numerical examples of FSLE using the SOR method in Section 4. But, first, we discuss some theorems related to it.

Theorem 2. *If the SOR method is convergent, then* $0 < w < 2$.

Definition 1. ***(P-regular Splitting)***

Let $A, B, C \in L(\mathbb{R}^n)$. Then $A = B - C$ is a P-regular splitting of A if B is non-singular and $B + C$ is positive definite.

Theorem 3. ***(Stein's Theorem)***
Let $H \in L(\mathbb{R}^n)$ and $A \in L(\mathbb{R}^n)$ be a symmetric positive definite matrix such that $A - H^T AH$ is positive definite. Then $\rho(H) < 1$.

Let λ be any eigen value of H and $u \neq 0$ be a corresponding eigen vector. Then $u^H Au$ and $u^H(A - H^T AH)u$ are real and positive.
Therefore $u^H Au > u^H H^T AHu = (\lambda u)^H A(\lambda u) = |\lambda|^2 u^H Au$, in such a way that $|\lambda|^2 < 1$.

This completes the proof.

Theorem 4. ***(P-regular Splitting Theorem)***
Let $A \in L(\mathbb{R}^n)$ be symmetric positive definite and $A = B - C$ be a P-regular splitting. Then $\rho(B^{-1}C) < 1$.

By Stein's theorem, it is sufficient to show that $Q = A - (B^{-1}C)^T AB^{-1}C$ is positive definite.
Since $B^{-1}C = I - B^{-1}A$, then we have

$$\begin{aligned} Q &= (B^{-1}A)^T A + AB^{-1}A - (B^{-1}A)^T AB^{-1}A \\ &= (B^{-1}A)^T (B + B^T - A)B^{-1}A \end{aligned}$$

But $B + B^T - A = B^T + C$ is positive definite with $B + C$. Therefore Q is positive definite.

This completes the proof.

Theorem 5. ***(Ostrowski-Reich Theorem)***
Let $A \in L(\mathbb{R}^n)$ be a symmetric positive definite matrix and $0 < w < 2$. Then the SOR method converges for any choice of initial approximate vector $X^{(0)}$.
Proof: By fundamental theorem of linear iterative methods (see [2, pp. 118-119]) and P-regular splitting theorem (i.e., Theorem 4), it is sufficient by using the equation (9) to show that

$$A = w^{-1}(D + wL) - w^{-1}[(1 - w)D - wU]$$

is a P-regular splitting of A.
Since the diagonal elements of A are positive, D is positive definite and $D + wL$ is non-singular.
Moreover, the symmetric part of $B + C$ is

$$\begin{aligned} B + B^T - A &= w^{-1}D + L + w^{-1}D + L^T - w^{-1}D - L + w^{-1}D - D - U \\ &= 2w^{-1}D - D + U - U[\because U = L^T] \\ &= 2w^{-1}D - D \\ &= w^{-1}(2 - w)D \end{aligned}$$

which is positive definite, since $0 < w < 2$.
Hence the SOR method converges for any choice of initial approximate vector $X^{(0)}$.

4. Numerical examples

In this section, we illustrate some numerical examples of FSLE using the SOR method. To solve these FSLE, we use the iteration scheme of equation (12) as derived in Section 3.

Example 1. *Let us consider the* 2×2 *FSLE*

$$x_1 - x_2 = (0,1,2)$$
$$x_1 + 3x_2 = (4,5,7)$$

If we recall from Section 2, *the conversion method from triplet form to form of a TFN*,

$$\begin{aligned}(0,1,2) &= (0 + (1-0)r, 2-(2-1)r), r \in [0,1]\\ &= (r, 2-r), r \in [0,1]\end{aligned}$$

and

$$\begin{aligned}(4,5,7) &= (4 + (5-4)r, 7-(7-5)r), r \in [0,1]\\ &= (4+r, 7-2r), r \in [0,1]\end{aligned}$$

Therefore the 2×2 *FSLE now becomes*

$$x_1 - x_2 = (r, 2-r)$$
$$x_1 + 3x_2 = (4+r, 7-2r)$$

The extended 4×4 *matrix is* $S = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 1 & 3 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 1 & 3 \end{bmatrix}$

Also, $Y = \begin{bmatrix} r \\ 4+r \\ 2-r \\ 7-2r \end{bmatrix}$

Let the value of the relaxation factor $w = 1.1$.
Thus from the equation (12), the iteration scheme for SOR method is as follows:

$$\underline{x}_1^{(k+1)} = -0.1\,\underline{x}_1^{(k)} + 1.1\,\overline{x}_2^{(k)} + 1.1\,r$$

$$\overline{x}_1^{(k+1)} = -0.1\,\overline{x}_1^{(k)} + 1.1\,\underline{x}_2^{(k)} + 1.1(2-r)$$

$$3\,\underline{x}_2^{(k+1)} + 1.1\,\underline{x}_1^{(k+1)} = -0.3\,\underline{x}_2^{(k)} + 1.1\,(4+r)$$

$$3\,\overline{x}_2^{(k+1)} + 1.1\,\overline{x}_1^{(k+1)} = -0.3\,\overline{x}_2^{(k)} + 1.1(7-2r)$$

where $k = 0, 1, 2, \ldots$

Let $\underline{x_1}^{(0)} = \overline{x_1}^{(0)} = \underline{x_2}^{(0)} = \overline{x_2}^{(0)} = 0$ and the tolerance be 10^{-3}.
Then the detailed calculations are shown in the following table:

k	$\underline{x_1}$	$\overline{x_1}$	$\underline{x_2}$	$\overline{x_2}$
0	0	0	0	0
1	1.100r	2.200-1.100r	1.467-0.037r	1.760-0.330r
2	1.936+0.627r	3.593-1.030r	0.610+0.140r	1.073-0.323r
3	0.987+0.683r	2.512-0.842r	1.044+0.102r	1.538-0.392r
4	1.594+0.600r	3.097-0.903r	0.778+0.136r	1.277-0.363r
5	1.246+0.641r	2.746-0.860r	0.932+0.118r	1.432-0.382r
6	1.451+0.616r	2.951-0.884r	0.842+0.129r	1.342-0.371r
7	1.331+0.630r	2.831-0.870r	0.895+0.123r	1.395-0.377r
8	1.401+0.622r	2.901-0.878r	0.863+0.126r	1.363-0.374r
9	1.360+0.627r	2.860-0.873r	0.882+0.124r	1.382-0.376r
10	1.384+0.624r	2.884-0.876r	0.871+0.125r	1.371-0.375r
11	1.370+0.626r	2.870-0.874r	0.877+0.125r	1.377-0.375r
12	1.378+0.625r	2.878-0.875r	0.874+0.125r	1.374-0.375r
13	1.373+0.625r	2.873-0.875r	0.876+0.125r	1.376-0.375r
14	1.376+0.625r	2.876-0.875r	0.875+0.125r	1.375-0.375r
15	1.374+0.625r	2.874-0.875r	0.875+0.125r	1.375-0.375r
16	1.375+0.625r	2.875-0.875r	0.875+0.125r	1.375-0.375r

Therefore the solution is:

$$x_1 = \left(\underline{x_1}, \overline{x_1}\right) = (1.375 + 0.625r, 2.875 - 0.875r) = (1.375, 2.000, 2.875)$$

$$x_2 = \left(\underline{x_2}, \overline{x_2}\right) = (0.875 + 0.125r, 1.375 - 0.375r) = (0.875, 1.000, 1.375)$$

Example 2. *Let us consider the* 3×3 *FSLE*

$$9x_1 + 2x_2 + 4x_3 = (19.5, 20, 20.6)$$
$$x_1 + 10x_2 + 4x_3 = (5, 6, 6.6)$$
$$2x_1 - 4x_2 + 10x_3 = (-15.5, -15, -14.4)$$

$$(19.5, 20, 20.6) = (19.5 + 0.5r, 20.6 - 0.6r)$$
$$(5, 6, 6.6) = (5 + r, 6.6 - 0.6r)$$
$$(-15.5, -15, -14.4) = (-15.5 + 0.5r, -14.4 - 0.6r)$$
for $r \in [0,1]$.

Therefore the 3×3 FSLE now becomes

$$9x_1 + 2x_2 + 4x_3 = (19.5 + 0.5r, 20.6 - 0.6r)$$
$$x_1 + 10x_2 + 4x_3 = (5 + r, 6.6 - 0.6r)$$
$$2x_1 - 4x_2 + 10x_3 = (-15.5 + 0.5r, -14.4 - 0.6r)$$

The extended 6×6 matrix is

$$S = \begin{bmatrix} 9 & 2 & 4 & 0 & 0 & 0 \\ 1 & 10 & 4 & 0 & 0 & 0 \\ 2 & 0 & 10 & 0 & -4 & 0 \\ 0 & 0 & 0 & 9 & 2 & 4 \\ 0 & 0 & 0 & 1 & 10 & 4 \\ 0 & -4 & 0 & 2 & 0 & 10 \end{bmatrix}$$

Also, $Y = \begin{bmatrix} 19.5 + 0.5r \\ 5 + r \\ -15.5 + 0.5r \\ 20.6 - 0.6r \\ 6.6 - 0.6r \\ -14.4 - 0.6r \end{bmatrix}$

Let the value of the relaxation factor w = 1.01.

Thus from the equation (12), the iteration scheme for SOR method is as follows:

$$9\underline{x_1}^{(k+1)} = -0.09\underline{x_1}^{(k)} - 1.01\left(2\underline{x_2}^{(k)} + \underline{x_3}^{(k)}\right) + 1.01(19.5 + 0.5r)$$
$$9\overline{x_1}^{(k+1)} = -0.09\overline{x_1}^{(k)} - 1.01\left(2\overline{x_2}^{(k)} + \overline{x_3}^{(k)}\right) + 1.01(20.6 - 0.6r)$$
$$10\underline{x_2}^{(k+1)} + 1.01\underline{x_1}^{(k+1)} = -0.1\underline{x_2}^{(k)} - 4.04\underline{x_3}^{(k)} + 1.01(5 + r)$$
$$10\overline{x_2}^{(k+1)} + 1.01\overline{x_1}^{(k+1)} = -0.1\overline{x_2}^{(k)} - 4.04\overline{x_3}^{(k)} + 1.01(6.6 - 0.6r)$$
$$10\underline{x_3}^{(k+1)} + 2.02\underline{x_1}^{(k+1)} = -0.1\underline{x_3}^{(k)} + 4.04\overline{x_2}^{(k)} + 1.01(-15.5 + 0.5r)$$
$$10\overline{x_3}^{(k+1)} + 2.02\overline{x_1}^{(k+1)} = -0.1\overline{x_3}^{(k)} + 4.04\underline{x_2}^{(k)} + 1.01(-14.4 - 0.6r)$$

where $k = 0,1,2,\ldots$

Let $\underline{x_1}^{(0)} = \overline{x_1}^{(0)} = \underline{x_2}^{(0)} = \overline{x_2}^{(0)} = \underline{x_3}^{(0)} = \overline{x_3}^{(0)} = 0$ and 10^{-3} be the tolerance.

Then the detailed calculations are shown in the following table:

k	$\underline{x}_1$	$\overline{x}_1$	$\underline{x}_2$	$\overline{x}_2$	$\underline{x}_3$	$\overline{x}_3$
0	0	0	0	0	0	0
1	2.188+0.056r	2.312-0.067r	0.284+0.095r	0.433-0.054r	-2.008+0.039r	-1.921-0.047r
2	3.004+0.017r	3.054-0.033r	1.010+0.083r	1.130-0.038r	-1.977+0.025r	-1.937-0.015r
	2.819+0.026r	2.897-0.052r	1.009+0.087r	1.145-0.049r	-1.659+0.030r	-1.612-0.017r
	2.678+0.023r	2.749-0.048r	0.895+0.086r	1.029-0.049r	-1.627+0.026r	-1.586-0.015r
5	2.691+0.025r	2.765-0.049r	0.882+0.087r	1.018-0.049r	-1.677+0.026r	-1.636-0.016r
6	2.716+0.025r	2.790-0.049r	0.899+0.087r	1.035-0.049r	-1.686+0.025r	-1.645-0.015r
7	2.716+0.025r	2.790-0.049r	0.903+0.087r	1.039-0.049r	-1.679+0.026r	-1.638-0.015r
8	2.712+0.025r	2.786-0.049r	0.900+0.087r	1.037-0.049r	-1.677+0.025r	-1.636-0.015r
9	2.712+0.025r	2.786-0.049r	0.900+0.087r	1.036-0.049r	-1.678+0.025r	-1.637-0.015r

Therefore the solution is:

$$x_1 = \left(\underline{x_1}, \overline{x_1}\right) = (2.712 + 0.025r, 2.786 - 0.049r) = (2.712, 2.737, 2.786)$$

$$x_2 = \left(\underline{x_2}, \overline{x_2}\right) = (0.900 + 0.087r, 1.036 - 0.049r) = (0.900, 0.987, 1.036)$$

$$x_3 = \left(\underline{x_3}, \overline{x_3}\right) = (-1.678 + 0.025r, -1.637 - 0.015r) = (-1.678, -1.653, -1.637)$$

5. Conclusion

In conclusion, this paper has explored the solution of fuzzy systems of linear equations using the Successive Over-Relaxation (SOR) method. Throughout this paper, a comprehensive review of the relevant literature has been conducted to understand the theoretical foundations of fuzzy systems and the SOR method. The SOR iteration scheme is adapted and extended to handle fuzzy numbers, allowing for the representation and manipulation of uncertain data in the context of linear equations. The proposed approach is implemented and tested using some numerical examples. The results demonstrate the viability of the SOR method in solving fuzzy systems of linear equations, as it provide accurate solutions while maintaining computational efficiency. The fuzzy nature of the system is appropriately considered, and the iteration scheme has effectively dealt with the imprecision and uncertainty inherent in fuzzy numbers.

Moreover, the paper has addressed the convergence analysis of the SOR method for fuzzy systems, providing insights into its stability and convergence properties. The convergence criteria are established, ensuring the reliability of the numerical solutions obtained. While this paper has achieved its objectives and provided valuable contributions, there are still avenues for further research. Future work could explore the extension of the SOR method to solve more complex fuzzy systems, including higher-dimensional systems and systems with nonlinear relationships. Additionally, investigations into the application of the SOR method in real-world scenarios and comparisons with other existing solution techniques would be beneficial to further validate its effectiveness and performance.

Overall, this paper has successfully addressed the solution of fuzzy systems of linear equations using the SOR method, offering a valuable contribution to the field of fuzzy mathematics and computational techniques. The findings presented here provide a solid foundation for future research and practical applications in various domains where uncertainty and imprecision are prevalent.

REFERENCES

1. M. Friedman, M. Ming, A. Kandel, Fuzzy linear systems, *Fuzzy Sets and Systems*, 96(2) (1998) 201–209.
2. J.M. Ortega, Numerical Analysis: A Second Course, SIAM, 1990.
3. T. Allahviranloo, Numerical methods for fuzzy system of linear equations, *Appl. Math. Comput.*, 155(2) (2004) 493-502.
4. M. Pal, Numerical Analysis for Scientists and Engineers: Theory and C Programs, Narosa, 2007.
5. T. Allahviranloo, Successive over relaxation iterative method for fuzzy system of linear equations, *Appl. Math. Comput.*, 162(1) (2005) 189–196.
6. L. A. Zadeh, Fuzzy sets, *Information and Control*, 8(3) (1965) 338-353.
7. C.-X.Wu and M. Ma, Embedding problem of fuzzy number space: Part I, *Fuzzy Sets and Systems*, 44(1) (1991) 33–38.
8. A.Kandel, M. Friedamn and M. Ming, Fuzzy linear systems and their solution, *IEEE Trans. Fuzzy Systems*, (1996) 336–338.
9. M. Dehghan and B. Hashemi, Iterative solution of fuzzy linear systems, *Appl. Math. Comput.*, 175(1) (2006) 645–674.
10. P. Senthilkumar and G. Rajendran, An algorithmic approach to solve fuzzy linear systems, *J. Inf. Comput. Sci.*, 8(3) (2011) 503–510.
11. K. Wang and B. Zheng, Block iterative methods for fuzzy linear systems, *J. Appl. Math. Comput.*, 25(1) (2007) 119–136.
12. L. Inearat and N. Qatanani, Numerical methods for solving fuzzy linear systems, *Mathematics*, 6(2) (2018) 19.
13. S.-X. Miao, B. Zheng and K. Wang, Block SOR methods for fuzzy linear systems, *J. Appl. Math. Comput.*, 26(1) (2008) 201–218.
14. B. Asady, S. Abbasbandy and M. Alavi, Fuzzy general linear systems, *Appl. Math. Comput.*, 169(1) (2005) 34-40.
15. M. Ming, A. Kandel and M. Friedman, A new approach for defuzzification, *Fuzzy Sets and Systems*, 111(3) (2000) 351–356.
16. D. Dubois and H. Prade, Operations on fuzzy numbers, *International Journal of Systems Science*, 9(6) (1978) 613-626.
17. A.Vroman, G. Deschaijver and E.E. Kerre, Solving system of linear fuzzy equations by parametric functions-an improved algorithm, *Fuzzy Sets and Systems*, 158(14) (2007) 1515-1534.
18. X. Wang, Z. Zhong and M. Ha, Iteration algorithms for solving a system of fuzzy linear equations, *Fuzzy Sets and Systems*, 119(1) (2001) 121-128.
19. T. Allahviranloo, M. Ghanbari, A.A. Hosseinzadeh, E. Haghi and R. Nuraei, A note on "Fuzzy linear systems", *Fuzzy Sets and Systems*, 177(1) (2011) 87–92.

20. R. Ezzati, Solving fuzzy linear systems, *Soft Comput.*, 15(1) (2011) 193–197.
21. H.J. Zimmermann, *Fuzzy Set Theory and its Applications*, Kluwer, Dordrecht, 1996.