An -Najah National University

Faculty of Graduate Studies

Numerical Methods for Solving Fuzzy System of Linear Equations

By

Lubna Labeeb Ahmad Ineirat

Supervisor

Prof . Naji Qatanani

This Thesis is Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Mathematics, Faculty of Graduate Studies, An-Najah National University, Nablus, Palestine.

2017

Numerical Methods for Solving Fuzzy System of Linear Equations

By

Lubna Labeeb Ahmad Ineirat

This Thesis was Defends Successfully on 2 / 2 /2017, and Approved by:

Defense Committee Members	**Signature**
– **Prof. Naji Qatanani /Supervisor**	…..……………..
– **Dr. Abdel Halim Ziqan /External Examiner**	…………………
– **Dr. Mohammad Al-Amleh / Internal Examiner**	…………………

Dedication

This thesis is dedicated to my wonderful family

For their endless love, support and encouragement.

Acknowledgment

First of all, I would like to thank Allah for giving me the strength and determination to carry out this thesis.

I deeply thank my supervisor Prof. Dr. Naji Qatanani for proposing this subject for me, whose help, advice and supervision was invaluable.

My thanks also to my external examiner Dr. Abdel Halim ziqan from the Arab American University-Jenin and my internal examiner Dr. Mohammad Al-Amleh for their useful and valuable comments.

Thanks are also to all faculty members in the department of Mathematics at An -Najah National University.

Finally, and most important, huge thank to my family for full supports and also the Almighty God, for his grace in me.

Thank you very much, everyone!

Lubna Ineirat

Palestine, 2-2-2017.

الإقرار

أنا الموقعة أدناه، مقدمة الرسالة التي تحمل العنوان:

Numerical Methods for Solving Fuzzy System of Linear Equations

أقر بأن ما اشتملت عليه هذه الرسالة إنما هي نتاج جهدي الخاص، باستثناء ما تمت الإشارة اليه حيثما ورد، وأن هذه الرسالة ككل، أو أي جزء منها لم يقدم من قبل لنيل أية درجة علمية أوبحث علمي أو بحثي لدى أية مؤسسة تعليمية أو بحثية أخرى.

Declaration

The work provided in this thesis, unless otherwise referenced, is the research's own work, and has not been submitted elsewhere for any other degree or qualification.

Student's name: **إسم الطالب:**

Signature: **التوقيع:**

Date: **التاريخ:**

Table of Contents

List of Figures

List of Tables

Numerical Methods for Solving Fuzzy System of Linear Equations

By
Lubna Labeeb Ahmad Ineirat
Supervisor
Prof. Naji Qatanani

Abstract

We focus our attention on the analytical and numerical methods for solving the fuzzy linear system (FLS) and fully fuzzy linear system (FFLS).

For the analytical solution of the fuzzy linear system we have presented the following methods: Friedman's proposal, S. Abbasbandy and M. Alavi method, Fuzzy Solution by Using Fuzzy Center, Algorithmic Approach, Embedding method , LU decomposition method, and LU-Decomposition method of Mansouri and Asady. The analytical methods presented for the fully fuzzy linear system include: matrix inversion method, Cramer's rule and LU decomposition method.

For the numerical handling of the fuzzy linear system we have implemented the following techniques, namely: Iterative Jacobi method, Gauss−Sidel methods, and Successive over relaxation iterative method. For the fully fuzzy linear system we have used the Gauss -Jacobi and Gauss- Seidel methods.

To show the efficiency of these numerical techniques we have considered some numerical examples. Numerical results for both (FLS) and (FFLS) have shown to be in a closed agreement with the analytical ones.

We strongly believe that, the Successive over relaxation iterative method(SOR) is one of the most powerful numerical techniques for solving FLS in comparison with other numerical techniques. Moreover, the Gauss-Seidel method is more efficient than the Gauss –Jacobi method for solving FFLS.

Introduction

The subject of Fuzzy System of Linear Equations with a crisp real coefficient matrix and with a vector of fuzzy triangular numbers on the right-hand side and Fully Fuzzy System of Linear Equations where all the parameters of the system are fuzzy numbers arise in many branches of science and technology such as economics, statistics, telecommunications, image processing , physics and even social sciences. In the year of 1965 L.A. Zadeh [27] introduced and investigated the concept of fuzzy numbers that can be used to generalize crisp mathematical concept to fuzzy sets.

There is a vast literature on the investigation of solutions for fuzzy linear systems. Early work in the literature are on to linear equation systems whose coefficient matrix is crisp and the right hand vector is fuzzy, that is known as Fuzzy Linear Equation System (FLS), was first proposed by Friedman et al. [17]. For computing a solution, they used the embedding method and replaced the original fuzzy $n \times n$ linear system by a $2n \times 2n$ crisp linear system. Later, several authors studied FLS. Allahviranloo [4-5], used the Jacobi and Gauss–Seidel iterative methods to compute an approximate solution. He also used the successive over relaxation iterative method for solving FLS. Dehghan & Hashemi [12] investigated the existence of a solution provided that the coefficient matrix is strictly diagonally dominant matrix with positive diagonal entries and then applied several iterative methods for solving FLS. Ezzati [15] developed a new method for solving FLS by using embedding method and replaced an $n \times n$ fuzzy linear system by two $n \times n$ crisp linear system. Furthermore, Muzziolia et al. [22] discussed fuzzy linear systems in the form of $A_1 x + b_1 = A_2 x + b_2$ with A_1, A_2 square matrices of fuzzy coefficients and b_1, b_2 fuzzy number vectors. Abbasbandy and Jafarian [3] proposed the steepest descent method for solving fuzzy system of linear equation.

The crispness of the coefficient matrix makes the modeling of real life problems restricted. Linear systems, whose all the parameters are fuzzy i.e. both coefficient matrix and right hand vector are fuzzy, are named Fully Fuzzy Linear Equation System (FFLS). The main objective of FFLS is to widen the scope of FLS in scientific applications by removing the crispness assumption on the entries of coefficient matrix.

Dehgan et al. [13] have proposed the Adomian decomposition method, iterative methods and some computational methods such as Cramer's rule, Gauss elimination method, LU decomposition method and linear programming approach for finding the solutions of $n \times n$ FFLS. Then, they applied some iterative iterative techniques such as Richardson, Jacobi, Jacobi over relaxation (JOR), Gauss–Seidel, successive over relaxation (SOR), accelerated over relaxation (AOR), symmetric and unsymmetric SOR (SSOR and USSOR) and extrapolated modified Aitken (EMA) ,for solving (FFLS). In addition, they proposed methods from nonlinear Programming, such as Newton, quasi-Newton and conjugate gradient to solve FFLS [14].

Besides FLS and FFLS, there exist the dual forms of both systems in the literature. Generally, both FLS and FFLS are handled under two main headings: square $(n\ x\ n)$ and nonsquare $(mx\ n)$ forms. Most of the works in the literature deal with square form. For example, Asady et al. [8], extended the model of Friedman for $n\ x\ n$ fuzzy linear system to solve general $m\ x\ n$ rectangular fuzzy linear system for $m\ x\ n$,where coefficients matrix are crisp and the right-hand side column is a fuzzy number vector, they replaced the original fuzzy linear system $m\ x\ n$ by a crisp linear system $2m\ x\ 2n$. And they investigated conditions for the existence of a fuzzy solution.

Fuzzy elements of these systems can be taken as triangular, trapezoidal or generalized fuzzy numbers in general or parametric form. While triangular fuzzy numbers are widely used in earlier works, trapezoidal fuzzy numbers are neglected for a long time. Besides, there exist lots of works using the parametric and level cut representation of fuzzy numbers. Another classification for FFLS can be made also depending on whether FFLS has sign restrictions on its parameters. Having sign restrictions for FFLS means that all parameters of FFLS are assumed as positive. Since the parameters are assumed as positive in the most of the papers, further work is needed for FFLS with arbitrary (no restrictions on sign) fuzzy numbers.

This thesis is organized as follows:

In chapter one, we introduce some basic concepts of fuzzy sets, crisp sets, fuzzy numbers, and fuzzy linear system.

Chapter two investigates some analytical methods for solving the Fuzzy Linear System of Equations. These methods are: Friedman's proposal, S. Abbasbandy and M. Alavi Method, Fuzzy Solution by Using Fuzzy Center, Algorithmic Approach, Embedding method, LU decomposition method, and

LU-Decomposition Method of Mansouri and Asady. For the Fully Fuzzy System of Linear Equations we presented the analytical methods: matrix inversion method, Cramer's rule and LU decomposition method.

In chapter three, we employ some numerical methods to solve fuzzy system of linear equations. These are: Iterative Jacobi, Gauss−Sidel methods, and Successive over relaxation iterative method. And we employ Jacobi and Gauss−Sidel methods for fully fuzzy system of linear equations.

In chapter four, MATLAP software has been used to solve numerical examples to demonstrate the efficiency of these numerical schemes introduced in chapter three.

Finally, we draw a comparison between analytical and numerical solutions for some numerical examples.

\

Chapter One

Mathematical Preliminaries

1.1 Crisp Sets

The concept of a set is fundamental in mathematics and it can be described as a collection of objects possibly linked through some properties.

Definition (1.1) [9]: Characteristic function:

Let X be a set and A be a subset of $X(A \subseteq X)$. Then the characteristic function of the set A in X is defined by:

$$\chi_A(x) = \begin{cases} 1 & if\ x \in A, \\ 0 & if\ x \notin A. \end{cases}$$

Classical sets and their operations can be represented by their characteristic functions.

Indeed, Let us consider the union

$A \cup B = \{x \in X \mid x \in A \ or\ x \in B\}$. Its Characteristic function is

$$\chi_{A \cup B}(x) = max\{\chi_A(x), \chi_B(x)\}$$

For the intersection

$A \cap B = \{ x \in X | x \in A\ and\ x \in B\}$ the characteristic function is

$$\chi_{A \cap B}(x) = min\{\chi_A(x), \chi_B(x)\}.$$

If we consider the complement of $A\ in\ X$,

$A^c = \{x \in X \mid x \notin A\}$ it has the characteristic function

$$\chi_{A^c}(x) = 1 - \chi_A(x).$$

1.2 Fuzzy Sets

Zadeh in [27] extended the definition of the characteristic functions by replacing the set $\{0,1\}$ by the closed interval $[0,1]$which is the bases to the new definition of fuzzy sets.

Definition (1.2) [27]: Fuzzy set:

A fuzzy set A(fuzzy subset of X) is defined as a mapping

$$A : X \to [0,1],$$

where $A(x)$ is the membership degree function of x to the fuzzy set A. We denote the collection of all fuzzy subsets of X by $\mathcal{F}(X)$.

In the case of the characteristic function $\chi_A : X \to \{0,1\}$ if $\chi_A(x)=0$ then; the grade of membership is 0; and that means x doesn't belong to A, if the characteristic function $\chi_A(x)=1$, then the grade of membership is equal to 1; and that means x belongs to A. While, in the case of fuzzy sets: $\mu_A(x)$could be any other number from 0 to 1.

We identify a fuzzy set with its membership function. Other notations that can be used the following $\mu_A(x) = A(x)$.

Example 1.1:

$\mu_A(x)=0.95$ may means that x is more likely to be in μ_A, or if $\mu_A(x)=0.5$ then x may be half way between belonging to A and not belonging to A. It is clear that regular subsets of X are a special case of fuzzy sets called crisp fuzzy sets where $\mu_A(x) \in \{0,1\} \subseteq [0,1]$.

We use different ways to display a fuzzy subset of X. In the next example we describe some of those ways:

Example 1.2:

Consider the regular set $X = \{a, b, c, d, e\}$ and let μ_A be the fuzzy subset of X that maps X to [0,1] by the following mapping:

$$a \to 0.2, \quad b \to 0.83, \quad c \to 0.5, \quad d \to 0, \quad and\ e \to 0.6$$

We may write μ_A as the set of ordered pairs:

$\mu_A = \{(a, 0.2), (b, 0.83), (c, 0.5), (d, 0), (e, 0.6)\}$ using notation of regular set, or we may represent it as $\mu_A = \{a_{0.2}, b_{0.83}, c_{0.5}, d_0, e_{0.6}\}$. This last form will be mostly used in this manuscript.

Operations on Fuzzy Sets

Zadeh in his first publication [27], define the operations for fuzzy sets by generalize the theoretic operations of crisp sets (the reader should realize that the set theoretic operations intersection, union and complement correspond to the logical operators and, inclusive or and negation).

Definition (1.3) [28]: Operations for fuzzy sets:

Let E and D be two fuzzy sets, then:

1) The intersection of E and D is the fuzzy set C with

$$C(x) = (E \cap D)(x) = min\{E(x), D(x)\} = E(x) \wedge D(x), \quad \forall x \in X.$$

2) The union of E and D is the fuzzy set C, where

$$C(x) = (E \cup D)(x) = max\{E(x), D(x)\} = E(x) \vee D(x), \quad \forall x \in X.$$

3) The complement of E is the fuzzy set D, where

$$D(x) = E^c(x) = 1 - E(x), \forall x \in X.$$

4) Difference $(E - D)(x) = (E \cap D^c)(x) = min\{E(x), 1 - D(x)\}$.

5) equilibrium points $E(x) = E^c(x)$.

In the following examples we illustrate the previous definitions.

Example 1.3:

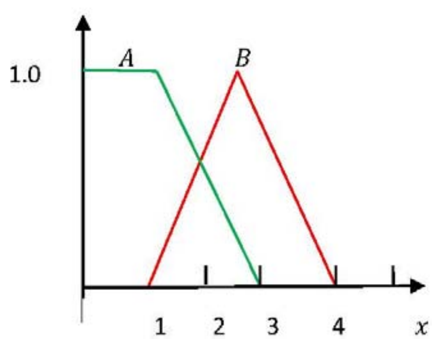
Consider the following two fuzzy sets:

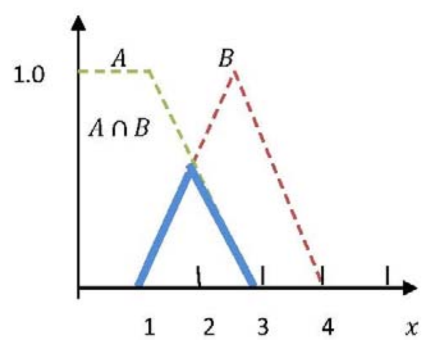
A={$a_{1.0}$, $b_{0.5}$, $c_{0.3}$, $d_{0.2}$} and B={$a_{0.5}$, $b_{0.7}$, $c_{0.2}$, $d_{0.4}$ }.

1. Complement A^c ={$a_{0.0}$, $b_{0.5}$, $c_{0.7}$, $d_{0.8}$}.
2. Complement B^c ={$a_{0.5}$, $b_{0.3}$, $c_{0.8}$, $d_{0.6}$}.
3. Union: $A \cup B$ ={$a_{1.0}$, $b_{0.7}$, $c_{0.3}$, $d_{0.4}$}.
4. Intersection: $A \cap B$ ={$a_{0.5}$, $b_{0.5}$, $c_{0.2}$, $d_{0.2}$}.
5. Difference $A - B = A \cap B^c$ =={$a_{0.5}$, $b_{0.3}$, $c_{0.3}$, $d_{0.2}$}.

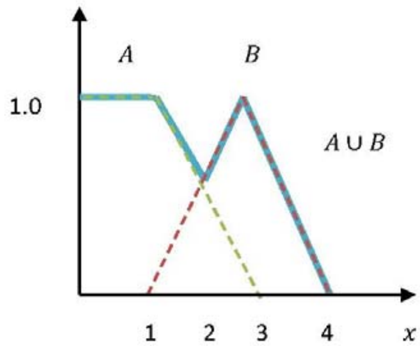
For the continuous graph case:

Take $X = [0,4]$, $A(x)$ and $B(x)$ are as follows:

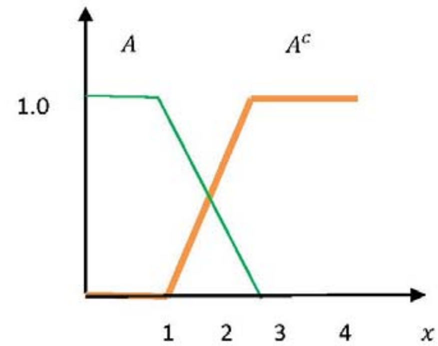




Intersection of fuzzy sets



Union of fuzzy sets



Complement of fuzzy sets

Now we can compare two fuzzy subsets of a set X as one of them containing the other as follows:

Definition (1.4) [27]: The containment:

Let A, B be two fuzzy subsets of X, we say $A \le B$ to mean $A(x) \le B(x)$ for all $x \in X$.

For example: Consider $X = \{a, b, c, d\}$ and let $A=\{a_{0.4}, b_{0.8}, c_{0.1}, d_0\}$ and $B=\{a_{0.1}, b_{0.8}, c_0, d_0\}$, then clearly $B \le A$.

Definition (1.5) [9]: The support of the fuzzy set :

The support of the fuzzy set A is defined by:

$$supp(A) = \{x \in X : A(x) > 0\}.$$

Definition (1.6) [18]: $\boldsymbol{\alpha}$ –cut:

An α –level set of a fuzzy set A of X is a non-fuzzy set denoted by A^α and is defined by:

$$A^\alpha = \begin{cases} x \in X : A(x) \ge \alpha & , if \alpha \in (0,1] \\ cl(supp(A)) & , if \alpha = 0 \end{cases}$$

where $cl(supp(A)) = Closure\ of\ the\ support\ A$.

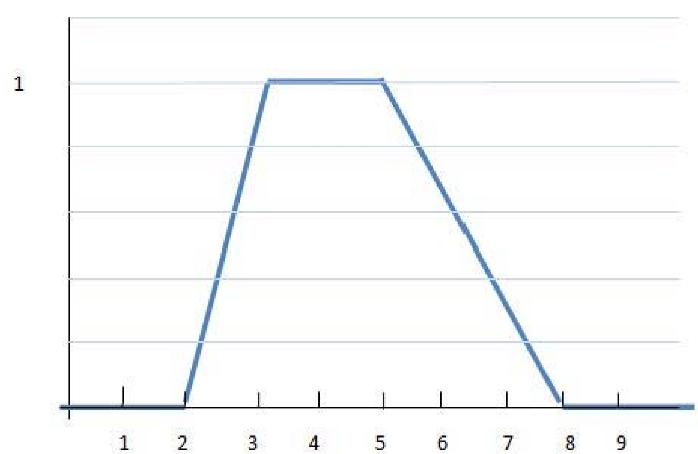
Example 1.4:

The following example displays some $\alpha -$ levels of some fuzzy subsets:

Let A ={ $a_{0.4}$, $b_{0.7}$, $c_{0.3}$, $d_{0.2}$ }be a fuzzy subset of X={a, b, c, d}then the 0.3-level=$A^{0.3}$={a, b, c}, the 0.1-level=$A^{0.1}$={a, b, c, d}. And the support of A $supp(A) = X$ ={a, b, c, d}.

Example 1.5:

The following represents the graph of a fuzzy subset of $R = (-\infty, \infty)$ with its function representation.



where $A(x) = \begin{cases} x-2 & if\ x \in [2,3] \\ 1 & if\ x \in [3,5] \\ \frac{8-x}{3} & if\ x \in [5,8] \\ 0 & elsewhere \end{cases}$

The 0.4 level of this fuzzy set is, $A^{0.4} = \{x \in X: A(x) \geq 0.4\}$

$0.4 \leq x - 2 \Rightarrow x \geq 2.4$

$0.4 \geq \dfrac{8-x}{3} \Rightarrow x \leq 6.8\ so\ A^{0.4} = [2.4 , 6.8]$

In general, the α-level can be found as follows:

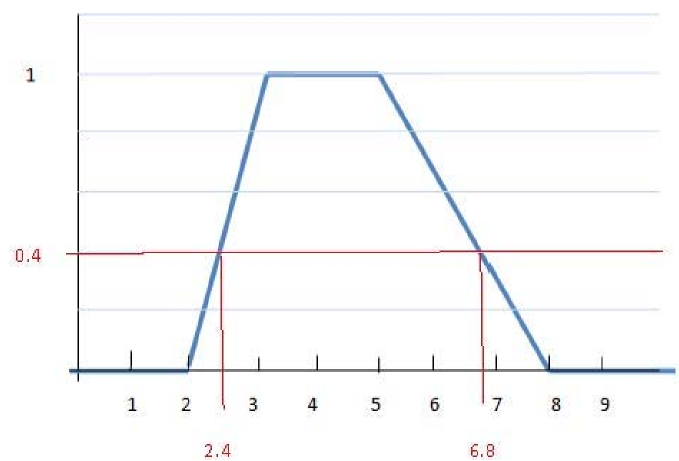
A^{α}=[x_1^{α} , x_2^{α}]

Now, $\alpha = x_1^{\alpha} - 2$, and this implies that $x_1^{\alpha} = \alpha + 2$

And $\alpha = \frac{8-x_2^{\alpha}}{3}$ which means $x_2^{\alpha} = 8 - 3\alpha$

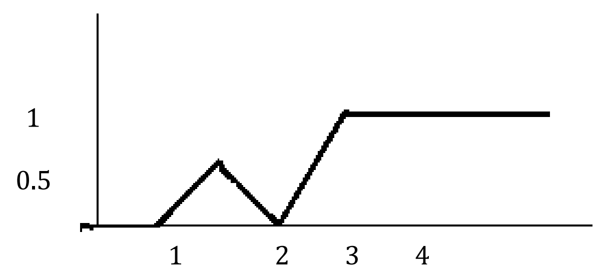
So $A^{\alpha} = [\alpha + 2, 8 - 3\alpha]$

For $\alpha = 0.4$, $A^{0.4} = [2.4 , 6.8]$



Example 1.6:

Let A defined as the following



$Supp(A) = (1 , 2) \cup (2 , \infty)$

$A^{0.4}$ ={$x \in X : A(x) \geq 0.4$} = $[1.8 , 2.2] \cup [3.4 , \infty)$.

Definition (1.7) [23]: Normal fuzzy set:

A fuzzy set A is called normal if there is at least one point $x \in R$ with $A(x) = 1$.

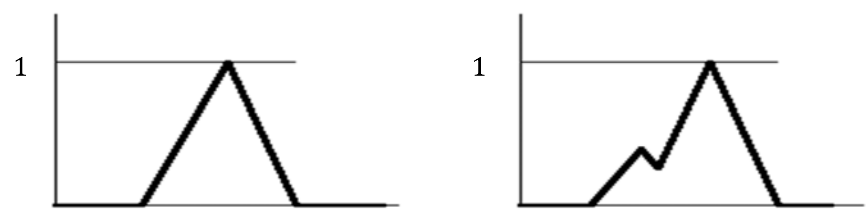
Definition (1.8) [23]: convex fuzzy set:

A fuzzy set E is convex if each of its α −level are convex set,

i.e. $E^{\alpha} = \{x \in X : E(x) \geq \alpha\}$ are convex $\forall \alpha \in (0,1]$.

An alternative definition of convexity: we call E convex if and only if

$$E(\tau\, y + (1 - \tau)\, z) \geq min\,\{E(y), E(z)\}, \forall y, z \in X, \quad \tau \in [0,1].$$



Convex Fuzzy Set   Non-convex Fuzzy Set

Figure (1.1)

1.3 Interval Arithmetic [11]

An interval is a subset of R such that $A = [a_1, a_2] = \{t : a_1 \leq t \leq a_2, a_{1,} a_2 \in R\}$.

If $A = [a_1, a_2]\ and\ B = [b_1, b_2]$ are two intervals, thus the arithmetic operations are:

Addition:

$[a_1, a_2] + [b_1, b_2] = [a_1 + b_1, a_2 + b_2]$

Subtraction

$[a_1, a_2] - [b_1, b_2] = [a_1 - b_2, a_2 - b_1]$

Product

$[a_1, a_2].[b_1, b_2] =$

$[\min(a_1 b_1, a_1 b_2, a_2 b_1, a_2 b_2), max(a_1 b_1, a_1 b_2, a_2 b_1, a_2 b_2)]$

Division

$[a_1, a_2]/[b_1, b_2]$=

$[\min(a_1/b_1, a_1/b_2, a_2/b_1, a_2/b_2), max(a_1/b_1, a_1/b_2, a_2/b_1, a_2/b_2)]$

$b_1, b_2 \neq 0$

1.4 Fuzzy Numbers

A way to describe the vagueness and lack of precision of data is a fuzzy number. The theory of fuzzy numbers is based on the theory of fuzzy sets which was introduced by Zadeh [27] in 1965. The concept of a fuzzy number was first used by Nahmias in the United States and by Dubois and Prade in France in the late 1970's. Our definition of a fuzzy number is illustrating in the following.

Definition (1.9) [16]: fuzzy number:

A fuzzy number is a fuzzy set $v: R \rightarrow [0,1]$ which satisfies:

- v is upper semi continuous.
- $v(x) = 0$ outside some interval $[a, d]$.
- There are real numbers $b, c:\ a \leq b \leq c \leq d$ for which

a. $v(x)$ is monotonic increasing on $[a, b]$,

b. $v(x)$ is monotonic decreasing on $[c, d\ ]$,

c. $v(x) = 1,\ b\ \leq\ x\ \leq\ c$.

$$\text{i.e. } v(x) = \begin{cases} 0\ , & x \leq a \\ f(x)\ , & a \leq\ x \leq b \\ 1\ , & b \leq x \leq c \\ g(x), & c \leq\ x \leq d \\ 0, & x \geq d \end{cases} \qquad (1.1)$$

where f is an increasing function and is called the left side, while g is a decreasing function and is called the right side.

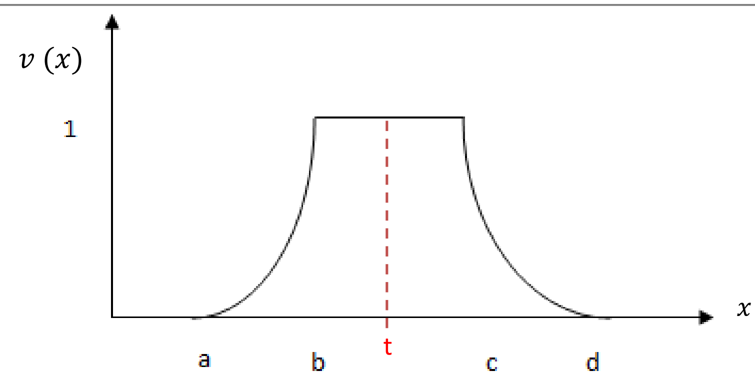


Figure (1.2)

- Also v is called symmetric fuzzy number if $v(t\ + x)\ =\ v(t\ -\ x)$ for all $x\ \in\ R$, where $t\ =\ \frac{b+c}{2}$.
- The set of all the fuzzy numbers is denoted by E^1.
- If $v(x)$ in the intervals $[a, b]$ and $[c, d]$ is linear then it is called a trapezoidal fuzzy number(which we will discuss later) and we write

$$v(x)\ \ = (\ a, b, c, d\ ).$$

Definition (1.10) [24]: Parametric form of fuzzy number:

An arbitrary fuzzy number in parametric form is represented by an ordered pair of functions $\left(\overline{v(r)}, \underline{v(r)}\right)$, $0 \leq r \leq 1$, which satisfy the following requirements:

1. $\underline{v(r)}$ is a bounded left-continuous non-decreasing function over $[0, 1]$.

2. $\overline{v(r)}$ is a bounded left-continuous non-increasing function over $[0, 1]$.

3. $\underline{v(r)} \leq \overline{v(r)}$; $0 \leq r \leq 1$.

Remark (1.1) [15]:

A crisp number α is simply represented by
$\underline{v(r)} = \overline{v(r)} = \alpha, 0 \leq r \leq 1$.

Also $v = (\underline{v}, \overline{v})$ is called a symmetric fuzzy number in parametric form if $v^c(r) = \frac{\underline{\underline{v(r)}}+\overline{v(r)}}{2}$ is a real constant for all $0 \leq r \leq 1$.

For example:

$u = (2 + r, 5 - 2r)$ is a fuzzy number and $v = (1 + r, 3 - r)$ is a symmetric fuzzy number in parametric form.

1.4.1. Types of a Fuzzy Number

Here we will talk about most popular types of fuzzy numbers, namely:

1) Triangular Fuzzy Number

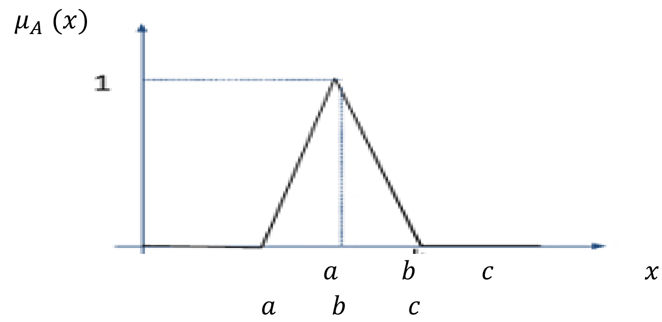


Figure (1.3)

A triangular fuzzy number (TFN) as illustrated in Figure (1.3) is a special type and the most common of fuzzy number and its membership function $\mu_A\ (x)$ is given by:

$$\mu_A\ (x) = \begin{cases} 0\ , & x \le a, \\ \frac{x-a}{b-a}\ , & a \le x \le b, \\ \frac{c-x}{c-b}\ , & b \le x \le c, \\ 0\ , & c \le x\ . \end{cases}$$

2) Trapezoidal Fuzzy Number

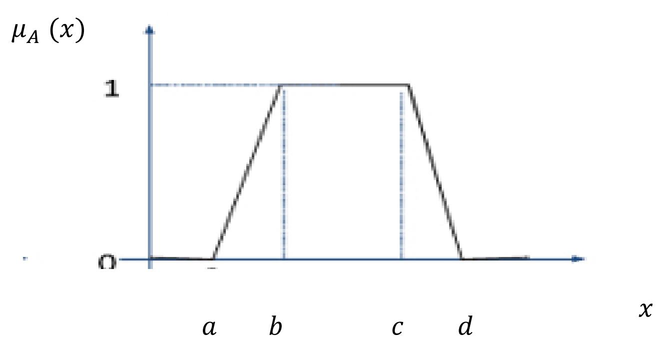


Figure (1.4)

A trapezoidal fuzzy number (Tr F N) which illustrated in Figure (1.4) is a special type of fuzzy number and its membership function $\mu_A\ (x)$ is given by

$$\mu_A(x) = \begin{cases} 0, & x \le a, \\ \frac{x-a}{b-a}, & a \le x \le b, \\ 1, & b \le x \le c, \\ \frac{d-x}{d-c}, & c \le x \le d, \\ 0, & d \le x. \end{cases}$$

1.4.2 Conversion from Fuzzy Number to Interval Using Alpha Cut

The α −cut operation can be also applied to the fuzzy number. If we denote α − cut interval for fuzzy number A as A^{α}, the obtained interval A^{α} is defined as following

$$A^{\alpha} = [x_1^{\alpha}, x_2^{\alpha}]$$

We can also know that it is an ordinary crisp interval

1) Conversion Triangular Fuzzy Number to Interval

Let, a triangular fuzzy number defined as $= (a_1, a_2, a_3)$, then to find α −cut of A , $\forall\alpha \in [0, 1]$ we first set α equal to the left and right membership function of A . That is,

$\alpha = \frac{x_1^{\alpha} - a_1}{a_2 - a_1}$ and $\alpha = \frac{a_3 - x_2^{\alpha}}{a_3 - a_2}$

Expressing x^{α} in terms of α we have, $x_1^{\alpha} = \alpha(a_2 - a_1) + a_1$ and

$x_2^{\alpha} = -\alpha(a_3 - a_2) + a_3$

Therefore, we can write the fuzzy interval in terms of α −cut interval as:

$$A^{\alpha}=[\alpha(a_2 - a_1) + a_1, -\alpha(a_3 - a_2) + a_3].$$

Example 1.7:

Let $A = (1, 2, 3)$, $B = (-3, -2, -1)$ and $C = (3, 4, 5)$

Then $A^{\alpha} = [1 + \alpha, 3 - \alpha]$, $B^{\alpha} = [-3 + \alpha, -1 - \alpha]$, $C^{\alpha} = [3 + \alpha, 5 - \alpha]$.

Example 1.8:

In the case of the triangular fuzzy number $A = (-5, -1, 1)$, the membership function value will be,

$$\mu_A\,(x) = \begin{cases} 0 & x \le -5 \\ \frac{x+5}{4} & -5 \le x \le -1 \\ \frac{1-x}{2} & -1 \le x \le 1 \\ 0 & 1 \le x \end{cases}$$

α −cut interval from this fuzzy number is

$$\frac{x+5}{4} = \alpha \Rightarrow x = 4\alpha - 5$$

$$\frac{1-x}{2} = \alpha \Rightarrow x = -2\alpha + 1$$

$$A^{\alpha} = [a_1^{\alpha}\ , a_2^{\alpha}\ ] = [4\alpha - 5, -2\alpha + 1]$$

If $\alpha = 0.5$, substituting 0.5 for α, we get $A^{0.5}$

$$A^{0.5} = [a_1^{0.5}\ , a_2^{0.5}\ ] = [-3, 0]$$

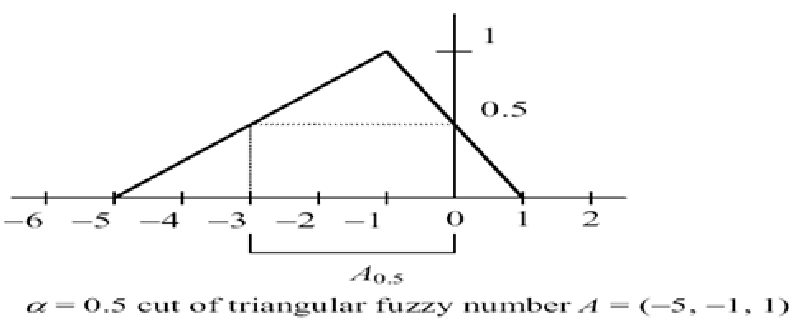


Figure (1.5)

2) Conversion Trapezoidal Fuzzy Number to Interval

Let, a trapezoidal fuzzy number defined as $A = (a_1, a_2, a_3, a_4)$

By following the similar procedure as above, we can write the fuzzy interval in terms of α –cut interval as following:

$$A^{\alpha} = [\alpha(a_2 - a_1) + a_1\ , -\alpha(a_4 - a_3) + a_4].$$

1.4.3. Fuzzy Arithmetic

Since A^{α} is now interval, so fuzzy addition, subtraction, multiplication and division are the same as interval arithmetic.

Definition (1.11) [20]:

As discussed above, fuzzy numbers may be transformed into an interval through parametric form. So, for any arbitrary fuzzy number $x = (\underline{x}(\alpha), \overline{x}(\alpha))$, $y = (\underline{y}(\alpha), \overline{y}(\alpha))$ and scalar k ,we have the interval based fuzzy arithmetic as

i. $x = y$ if and only if $\underline{x}(\alpha) = \underline{y}(\alpha)$ and $\overline{x}(\alpha) = \overline{y}(\alpha)$.

ii. $x + y = (\underline{x}(\alpha) + \underline{y}(\alpha) , \overline{x}(\alpha) + \overline{y}(\alpha))$.

iii. $x - y = (\underline{x}(\alpha) - \overline{y}(\alpha),\ \overline{x}(\alpha) - \underline{y}(\alpha))$.

iv. $x \times y = [min(\underline{x}(\alpha)\underline{y}(\alpha), \underline{x}(\alpha)\overline{y}(\alpha), \overline{x}(\alpha)\underline{y}(\alpha), \overline{x}(\alpha)\overline{y}(\alpha)),$

$max(\underline{x}(\alpha)\underline{y}(\alpha), \underline{x}(\alpha)\overline{y}(\alpha), \overline{x}(\alpha)\underline{y}(\alpha), \overline{x}(\alpha)\overline{y}(\alpha))]$.

v. $x/y = ((\underline{x}(\alpha), \overline{x}(\alpha)))/((\underline{y}(\alpha), \overline{y}(\alpha)) = (\underline{x}(\alpha)/\overline{y}(\alpha), \overline{x}(\alpha)/\underline{y}(\alpha))$.

provided $\underline{y}(\alpha) = \overline{y}(\alpha) \neq 0$

vi. $kx = \begin{cases} [k\underline{x}(\alpha), k\overline{x}(\alpha)] & ,k \geq 0 \\ [k\overline{x}(\alpha), k\underline{x}(\alpha)] & ,k < 0. \end{cases}$

Definition (1.12) [23]: Positive fuzzy number:

A fuzzy number A is called positive, denoted by $A > 0$, if its membership function $\mu_A(x)$satisfies $\mu_A(x) = 0, \forall x \leq 0$.

Definition (1.13) [23]: Nonnegative fuzzy number:

A fuzzy number A is called nonnegative, denoted by ≥ 0 , if its membership function $\mu_A(x)$satisfies $\mu_A(x)$=0, $\forall x < 0$.

Definition (1.14) [25]: Equality in fuzzy numbers:

Two triangular fuzzy numbers $N = (m, \gamma, \beta)$ and $M = (n, \alpha, \delta)$ are said to be equal, if and only if $m = n, \gamma = \alpha$ and $\beta = \delta$.

1.5 Fuzzy Linear System

In 1965[27] Lotfi Zadeh was submit fuzzy logic, which has had achieved many successful applications in several areas that one can imagine. The reason behind that they are many real-world applications problems are involved the systems in which at least some parameters are represented by fuzzy numbers rather than crisp numbers. Moreover a system of fuzzy linear equations may appear in a wide variety of problems in various areas such as engineering, mathematics, physics, statistic and social sciences.

A linear system of fuzzy equations divided into three categories

$$Ax = b \tag{1.2}$$

– In the first category, the coefficient matrix arrays are crisp numbers, the right-hand side column is an arbitrary fuzzy vector and the unknowns are fuzzy numbers.

– In the second category, the coefficient matrix arrays are fuzzy numbers, the right-hand side column is an arbitrary fuzzy vector and the unknowns are crisp numbers.

– In the third category, all the coefficient matrix arrays, the right-hand side arrays and the unknowns, are fuzzy numbers.

Definition (1.15) [1]: Fuzzy linear system:

The $n \times n$ linear system

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ \dots \ + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \ \dots \ + a_{2n}x_n = b_2, \\ \quad \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \ \dots \ + a_{nn}x_n = b_n, \end{cases} \tag{1.3}$$

where the coefficients matrix $A = (a_{ij}), 1 \le i,j \le n$ is a crisp $n \times n$ matrix and each $b_i \in E^1, 1 \le i \le n$, is fuzzy number, is called a fuzzy linear system (FLS).

Definition (1.16) [1]: Solution of fuzzy linear system:

A fuzzy number vector $X = (x_1, x_2, \dots, x_n)^t$ given by $x_i = \left(\underline{x_i}(r), \overline{x_i}(r)\right),\ 1 \le i \le n,\ 0 \le r \le 1$ is called (in parametric form) a solution of the FLS(1.3) if

$$\underline{\sum_{j=1}^{n} a_{ij}x_j} = \sum_{j=1}^{n} \underline{a_{ij}x_j} = \underline{b_i},$$

$$\overline{\sum_{j=1}^{n} a_{ij}x_j} = \sum_{j=1}^{n} \overline{a_{ij}x_j} = \overline{b_i}. \tag{1.4}$$

Following Friedman et al (1998) [17] we introduce the notations below:

$x = (\underline{x_1}, \underline{x_2}, \dots \underline{x_n}, -\overline{x_1}, -\overline{x_2}, \dots - \overline{x_n})^t$

$b = (\underline{b_1}, \underline{b_2}, \dots \underline{b_n}, -\overline{b_1}, \overline{-b_2}, \dots - \overline{b_n})^t$

$S = (s_{ij}), 1 \le i,j \le 2n, where\ s_{ij}$ are determined as follows:

$$a_{ij} \ge 0 \Rightarrow s_{ij} = a_{ij}, s_{i+n,j+n} = a_{ij},$$

$$a_{ij} < 0 \Rightarrow s_{i,j+n} = -a_{ij}, s_{i+n,j} = -a_{ij}. \tag{1.5}$$

and any s_{ij} which is not determined by (1.5) is zero. Using matrix notation we have

$$SX = b \tag{1.6}$$

The structure of S implies that $s_{ij} \geq 0$ and thus

$$S = \begin{pmatrix} B & C \\ C & B \end{pmatrix} \tag{1.7}$$

where B contains the positive elements of A , C contains the absolute value of the negative elements of A and $A = B - C$. An example in the work of Friedman et al (1998) shows that the matrix S may be singular even if A is nonsingular.

Theorem (1.1) [17]:

(Friedman et al (1998)) The matrix S is nonsingular matrix if and only if the matrices $A = B - C$ and $B + C$ are both nonsingular.

Proof. By subtracting the jth column of S, from its $(n + j)$th column for $1 \leq j \leq n$ we obtain

$$S = \begin{pmatrix} B & C \\ C & B \end{pmatrix} \rightarrow \begin{pmatrix} B & C - B \\ C & B - C \end{pmatrix} = S_1.$$

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

$$S_1 = \begin{pmatrix} B & C - B \\ C & B - C \end{pmatrix} \rightarrow \begin{pmatrix} B + C & 0 \\ C & B - C \end{pmatrix} = S_2.$$

Clearly, $|S| = |S_1| = |S_2| = |B + C||B - C| = |B + C||A|$.

Therefore

$|S| \neq 0$ if and only if $|A| \neq 0$ and $|B + C| \neq 0$,

Which concludes the proof.

Corollary 1.1 [17]:

If a crisp linear system does not have a unique solution, the associated fuzzy linear system does not have one either.

Definition (1. 17) [7]: Strong solution:

If $X = (\underline{x_1}, \underline{x_2}, \dots \underline{x_n}, -\overline{x_1}, -\overline{x_2}, \dots, -\overline{x_n})^{\mathrm{T}}$ is a solution of (1.6) and for each $1 \leq i \leq n$, when the inequalities $\underline{x_i} \leq \overline{x_i}$ hold, then the solution $X = (\underline{x_1}, \underline{x_2}, \dots \underline{x_n}, -\overline{x_1}, -\overline{x_2}, \dots, -\overline{x_n})^{\mathrm{T}}$ is called a strong solution of the system (1.6).

Definition (1.18) [7]: weak solution:

If $X = (\underline{x_1}, \underline{x_2}, \dots \underline{x_n}, -\overline{x_1}, -\overline{x_2}, \dots, -\overline{x_n})^{\mathrm{T}}$ is a solution of (1.6) and for some $i \in [1, n]$, when the inequality $\underline{x_i} \geq \overline{x_i}$ hold, then the solution $X = (\underline{x_1}, \underline{x_2}, \dots \underline{x_n}, -\overline{x_1}, -\overline{x_2}, \dots, -\overline{x_n})^{\mathrm{T}}$ is called a weak solution of the system (1.6).

Theorem (1.2) [7]:

Let $S = \begin{pmatrix} B & C \\ C & B \end{pmatrix}$ be a nonsingular matrix. Then the system (1.6) has a strong solution if and only if $(B + C)^{-1}(\underline{b} - \overline{b}) \leq 0$.

Theorem (1.3) [7]:

The FLS (1.3) has a unique strong solution if and only if the following conditions hold:

1) The matrices

$A = B - C$ and $B + C$ are both invertible matrices .

2) $(B + C)^{-1}(\underline{b} - \overline{b}) \leq 0$.

1.6 Fully Fuzzy Linear System of Equations

Definition (1.19) [13]:

Consider the $n \times n$ fully fuzzy linear system of equations:

$$\begin{cases} (\tilde{a}_{11} \otimes \tilde{x}_1) \oplus (\tilde{a}_{12} \otimes \tilde{x}_2) \oplus \cdots \oplus (\tilde{a}_{1n} \otimes \tilde{x}_n) = \tilde{b}_1, \\ (\tilde{a}_{21} \otimes \tilde{x}_1) \oplus (\tilde{a}_{22} \otimes \tilde{x}_2) \oplus \cdots \oplus (\tilde{a}_{2n} \otimes \tilde{x}_n) = \tilde{b}_2, \\ \vdots \\ (\tilde{a}_{n1} \otimes \tilde{x}_1) \oplus (\tilde{a}_{n2} \otimes \tilde{x}_2) \oplus \cdots \oplus (\tilde{a}_{nn} \otimes \tilde{x}_n) = \tilde{b}_n. \end{cases} \quad (1.8)$$

the matrix form of the above system is

$$\tilde{A} \otimes \tilde{x} = \tilde{b} \quad (1.9)$$

where the coefficient matrix $\tilde{A} = (\tilde{a}_{ij})$, $1 \leq i, j \leq n$ is an $n \times n$ fuzzy matrix and $\tilde{x}_i, \tilde{b}_i$, $1 \leq i \leq n$ are fuzzy vectors. This system is called fully fuzzy linear system (FFLS).

Let us review some important definitions and arithmetic of fuzzy number.

We symbolically represent the Triangular fuzzy number as $\widetilde{m} = (m, \alpha, \beta)$.

In addition we denote the set of all Triangular fuzzy number by F(R).

Definition (1.20) [13]: Positive fuzzy number:

A fuzzy number $\widetilde{m}$ is said to positive (negative), shows as $\widetilde{m} > 0 (\widetilde{m} < 0)$ where its membership function satisfies $\mu_{\widetilde{m}}(x) = 0, \forall x < 0 (\forall x > 0)$. Consequently, a Triangular fuzzy number as $\widetilde{m} = (m, \alpha, \beta)$ is said to be positive if and only if $m - \alpha \geq 0$.

Definition (1.21) [13]: Arithmetic operations on fuzzy numbers:

For two fuzzy numbers $\tilde{m} = (m, \alpha, \beta)$ and $\tilde{n} = (n, \gamma, \delta)$ we define

1. Addition: $\tilde{m} \oplus \tilde{n} = (m, \alpha, \beta) \oplus (n, \gamma, \delta) = (m + n, \alpha + \gamma, \beta + \delta)$.

2. Opposite: $-\tilde{m} = -(m, \alpha, \beta) = (-m, \beta, \alpha)$.

3. Multiplication of two fuzzy numbers : If $\tilde{m} > 0$ and $\tilde{n} > 0$, then

$$(m, \alpha, \beta) \otimes (n, \gamma, \delta) = (mn, m\gamma + n\alpha, m\delta + n\beta).$$

4. Scalar multiplication:

$$k \otimes (m, \alpha, \beta) = \begin{cases} (km, k\alpha, k\beta), & k \geq 0 \\ (km, -k\alpha, -k\beta), & k < 0 \end{cases}$$

Chapter Two

Analytical Methods for Solving Linear Fuzzy Systems

We will discuss some analytical methods for solving the first category of fuzzy linear systems (1.2) where the coefficient matrix are crisp numbers and the right-hand side column is an arbitrary fuzzy vector and the unknowns are fuzzy numbers. Moreover, we introduce some analytical methods for solving the third category of fuzzy linear systems (1.2) which is called fully fuzzy linear system, where all the coefficient matrix arrays, the right-hand side arrays and the unknowns are fuzzy numbers.

2.1 Analytical Methods for Solving Fuzzy Systems of Linear Equations (FLS)

2.1.1 Friedman's Proposal [17]

The idea of this approach is replacing the original system with matrix A by $(2n)\times(2n)$ crisp linear system with matrix S which may be singular matrix even if A is nonsingular matrix.

Consider the ith equation of the system (1.3):

$$a_{i1}\left(\underline{x}_1,\overline{x}_1\right)+\cdots+a_{ii}\left(\underline{x}_i,\overline{x}_i\right)+\cdots+a_{in}\left(\underline{x}_n,\overline{x}_n\right)=\left(\underline{y_i}(r),\overline{y_i}(r)\right),$$

we have

$$\begin{aligned}&\underline{a_{i1}x_1}+\cdots+\underline{a_{ii}x_i}+\cdots+\underline{a_{in}x_n}=\underline{y_i}(r)\\ &\overline{a_{i1}x_1}+\cdots+\overline{a_{ii}x_i}+\cdots+\overline{a_{in}x_n}=\overline{y_i}(r),\ 1\le i\le n,\ 0\le r\le 1.\end{aligned}\tag{2.1}$$

From (2.1) we have two crisp $n \times n$ linear systems for all i that means we can extended the fuzzy system (1.3) to a $2n \times 2n$ crisp linear system as follows:

$$SX = Y, \tag{2.2}$$

where s_{ij} are determined as follows:

$$\begin{aligned} &a_{ij} \geq 0 \Rightarrow s_{ij} = a_{ij}, s_{i+n,j+n} = a_{ij}, \\ &a_{ij} < 0 \Rightarrow s_{i,j+n} = -a_{ij}, s_{i+n,j} = -a_{ij}. \end{aligned} \tag{2.3}$$

and any s_{ij}which is not determined by equation(2.3) is zero and.

$$X=\begin{bmatrix} \underline{x}_1 \\ \vdots \\ \underline{x}_n \\ -\overline{x}_1 \\ \vdots \\ -\overline{x}_n \end{bmatrix}, \quad Y=\begin{bmatrix} \underline{y}_1 \\ \vdots \\ \underline{y}_n \\ -\overline{y}_1 \\ \vdots \\ -\overline{y}_n \end{bmatrix}. \tag{2.4}$$

The structure of $S = (s_{ij}), 1 \leq i, j \leq 2n$ implies $s_{ij} \geq 0$ and that

$$S=\begin{bmatrix} B & C \\ C & B \end{bmatrix} \tag{2.5}$$

where B contains the positive elements of A, and C the absolute values of the negative entries of A, that is, $A = B - C$.

now the system (2.2) yields to

$$SX = Y \rightarrow \begin{bmatrix} B & C \\ C & B \end{bmatrix}\begin{bmatrix} \underline{X} \\ -\overline{X} \end{bmatrix} = \begin{bmatrix} \underline{Y} \\ -\overline{Y} \end{bmatrix}. \tag{2.6}$$

Thus fuzzy linear system (1.3) is extended to a crisp (2.6) which also can be written as the following:

$$\begin{cases} B\underline{X} + C(-\overline{X}) = \underline{Y}, \\ C(-\underline{X}) + B\overline{X} = \overline{Y}. \end{cases} \tag{2.7}$$

Example 2.1:

Consider the 2× 2 fuzzy linear system

$3x_1 - 4x_2 = y_1,$

$5x_1 + 2x_2 = y_2.$

The 4×4 system is

$$\begin{aligned}
3\underline{x}_1 \qquad\qquad + 4(-\overline{x}_2) &= \underline{y}_1,\\
5\underline{x}_1 + 2\underline{x}_2 \qquad\qquad &= \underline{y}_2,\\
4\,\underline{x}_2 + 3(-\overline{x}_1) \qquad &= -\overline{y}_1,\\
5(-\overline{x}_1) + 2(-\overline{x}_2) &= -\overline{y}_2,
\end{aligned}$$

i.e.

$$S = \begin{pmatrix} 3 & 0 & 0 & 4 \\ 5 & 2 & 0 & 0 \\ 0 & 4 & 3 & 0 \\ 0 & 0 & 5 & 2 \end{pmatrix}$$

The linear system of equation(2.2) is now a $(2n) \times (2n)$ crisp linear system and can be uniquely solved for X, if and only if the matrix S is non-singular. On the other hand, the following example contradicts the notable fact that S may be singular even if the original matrix A is not.

Example 2.2:

The matrix A of the following fuzzy linear system

$2x_1 + 2x_2 = y_1,$

$3x_1 - 3x_2 = y_2$

is nonsingular matrix, while

$$S = \begin{pmatrix} 2 & 2 & 0 & 0 \\ 3 & 0 & 0 & 3 \\ 0 & 0 & 2 & 2 \\ 0 & 3 & 3 & 0 \end{pmatrix}$$

is singular. So a fuzzy linear system represented by a nonsingular matrix A may have no solution or an infinite number of solutions.

The next result eliminate the possibility of a unique fuzzy solution, whenever the crisp system is not uniquely solved, i.e. whenever A is singular.

Theorem 2.1 [17]:

If S^{-1} exists it must have the same structure as S, i.e.

$$S^{-1} = \begin{pmatrix} D & E \\ E & D \end{pmatrix} \tag{2.8}$$

Now, to calculate E and D we write

$$SS^{-1} = \begin{pmatrix} B & C \\ C & B \end{pmatrix} \begin{pmatrix} D & E \\ E & D \end{pmatrix} = \begin{pmatrix} \mathrm{I} & 0 \\ 0 & \mathrm{I} \end{pmatrix}$$

then we get

$$BD + CE = \mathrm{I}, \qquad CD + BE = 0. \tag{2.9}$$

By subtracting and adding the two parts of Equation(2.9) we obtain

$$D - E = (B - C)^{-1}, \qquad D + E = (B + C)^{-1}, \tag{2.10}$$

then we get,

$$D = \frac{1}{2}[(B + C)^{-1} + (B - C)^{-1}],$$

$$E = \frac{1}{2}[(B + C)^{-1} - (B - C)^{-1}]. \tag{2.11}$$

the solution vector actually is unique but may still not be a suitable fuzzy vector.

The next result provides necessary and sufficient conditions for the unique solution to be a fuzzy vector.

Theorem 2.2[17]:

The unique solution X of equation($X = S^{-1}Y$) is a fuzzy vector for arbitrary Y if and only if S^{-1} has nonnegative entries.

Proof: see [17].

Theorem 2.3 [17]:

The inverse of nonnegative matrix A is nonnegative if and only if A is a permutation matrix.

To define the fuzzy solution of the crisp linear system, we consider the following theorem:

Theorem 2.4 [19]:

Let $X = \left\{\left(\underline{x_i}(r), -\overline{x_i}(r)\right),\ 1 \le i \le n\right\}$ denote the unique solution of the $2n \times 2n$ crisp linear system(2.2). The fuzzy number vector $U = \left\{\left(\underline{u_i}(r), \overline{u_i}(r)\right),\ 1 \le i \le n\right\}$ defined by

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

is called the fuzzy solution of crisp system $SX = Y$. If $\left(\underline{x_i}(r), -\overline{x_i}(r)\right)$, $1 \le i \le n$, are all fuzzy numbers then $\underline{u_i}(r) = \underline{x_i}(r)$, $\overline{u_i}(r) = \overline{x_i}(r), 1 \le i \le n$ and U is called a strong fuzzy solution. Otherwise, U is a weak fuzzy solution.

Example 2.3:

Consider the 2×2 fuzzy system

$$2x_1 + 3x_2 = (2 + 2r, 8 - 4r),$$
$$5x_1 - x_2 = (4r,\ 6 - 2r).$$

The extended 4×4 matrix is

$$S = \begin{pmatrix} 2 & 3 & 0 & 0 \\ 5 & 0 & 0 & 1 \\ 0 & 0 & 2 & 3 \\ 0 & 1 & 5 & 0 \end{pmatrix}$$

and the solution of equation (2.2) is

$$X = \begin{bmatrix} \underline{x}_1(r) \\ \underline{x}_2(r) \\ -\overline{x}_1(r) \\ -\overline{x}_2(r) \end{bmatrix} = S^{-1}Y =$$
$$\begin{bmatrix} -2/221 & 45/221 & -15/221 & 6/221 \\ 75/221 & -30/221 & 10/221 & -4/221 \\ -15/221 & 6/221 & -2/221 & 45/221 \\ 10/221 & -4/221 & 75/221 & -30/221 \end{bmatrix} \begin{bmatrix} 2+2r \\ 4r \\ 4r-8 \\ 2r-6 \end{bmatrix},$$

i.e.

$$\underline{x}_1(r) = \frac{80 + 128r}{221}, \quad \overline{x}_1(r) = \frac{284 - 76r}{221}$$
$$\rightarrow x_1 = \left(\frac{80 + 128r}{221}, \frac{284 - 76r}{221}\right),$$

$$\underline{x}_2(r) = \frac{94 + 62r}{221}, \quad \overline{x}_2(r) = \frac{400 - 244r}{221}$$
$$\rightarrow x_2 = \left(\frac{94 + 62r}{221}, \frac{400 - 244r}{221}\right).$$

Here $\underline{x}_1 \leq \overline{x}_1,\ \underline{x}_2 \leq \overline{x}_2;\ \underline{x}_1,\ \underline{x}_2$ are monotonic decreasing functions. Thus the fuzzy solution $x_1,\ x_2$ is a strong fuzzy solution.

2.1.2 S. Abbasbandy and M. Alavi Method

This is an efficient method for solving an $n \times n$ system of fuzzy linear equations. The original system with matrix A is replaced by two $n \times n$ crisp function linear systems (in comparison with Friedman's procedure [17]). The obtained solution vector will be symmetric solution if the right hand side vector is symmetric [1].

Now, we will clarify the fuzzy solution

The i^{th} equation in (1.3) can be represent in the following equivalent form:

$$\sum_{a_{ij} \geq 0} a_{ij}\, \underline{x}_j + \sum_{a_{ij} < 0} a_{ij}\, \overline{x}_j = \underline{y}_i \qquad (2.12a)$$

$$\sum_{a_{ij} \geq 0} a_{ij}\, \overline{x}_j + \sum_{a_{ij} < 0} a_{ij}\, \underline{x}_j = \overline{y}_i \qquad (2.12b)$$

thus,

$$\sum_{a_{ij} \geq 0} a_{ij}(\overline{x}_j - \underline{x}_j) - \sum_{a_{ij} < 0} a_{ij}(\overline{x}_j - \underline{x}_j) = \overline{y}_i - \underline{y}_i \qquad (2.13)$$

If we assume $w_j = \overline{x}_j - \underline{x}_j$ and $v_i = \overline{y}_i - \underline{y}_i$ then Equation(2.13) has the form

$$\sum_{a_{ij} \geq 0} a_{ij} w_j - \sum_{a_{ij} < 0} a_{ij} w_j = v_i, \qquad i = 1,2, \dots, n,$$

and in the matrix form

$$(B + C)W = V,$$

Where $W = (w_1, w_2, \dots, w_n)^t$, $V = (v_1, v_2, \dots, v_n)^t$ and $A = B - C$. Let $X^c = (x_1^c, x_2^c, \dots, x_3^c)$ and $Y^c = (y_1^c, y_2^c, \dots, y_3^c)$ where $x_i^c = (\underline{x}_i(r) + \overline{x}_i(r))/2$ and $y_i^c = (\underline{y}_i(r) + \overline{y}_i(r))/2$ for $1 \leq i \leq n$.

Theorem 2.5 [1]:

Let X be a fuzzy solution of FLS (1.3) where coefficients matrix A is nonsingular matrix and Y is a fuzzy number vector. Then $AX^c = Y^c$.

Proof: Based on the equation (2.12), we have for each $i, 1 \leq i \leq n$

$$\sum_{a_{ij}\geq 0}(a_{ij}\frac{(\overline{x}_j(r)+\underline{x}_j(r))}{2})+\sum_{a_{ij}<0}(a_{ij}\frac{(\overline{x}_j(r)+\underline{x}_j(r))}{2})=\frac{(\overline{y}_i(r)+\underline{y}_i(r))}{2}$$

hence,

$$\sum_{a_{ij}\geq 0} a_{ij}x_j^c+\sum_{a_{ij}<0} a_{ij}x_j^c = y_i^c,$$

i.e., $(B-C)X^c = Y^c$, which conclude the proof.

Remark 2.1 [1]:

In previous Theorem, if Y is symmetric fuzzy vector then X is symmetric fuzzy vector.

Remark 2.2 [1]:

For finding the solution of FLS (1.3), we must solve the following crisp linear systems,

$$\begin{cases} (B+C)W = V, \\ (B-C)X^c = Y^c. \end{cases} \qquad (2.14)$$

And after solving (2.14), it is enough to take

$$\underline{x}_i = x_i^c - 0.5w_i$$

$$\overline{x}_i = x_i^c + 0.5w_i \qquad \text{for each } i,\ 1 \leq i \leq n.$$

Example 2.4:

Consider the 2×2 symmetric fuzzy linear system

$$x_1 - x_2 = (2r, 4 - 2r)$$

$$x_1 + 2x_2 = (6 + 3r, 12 - 3r)$$

thus

$$\underline{x}_1 - \overline{x}_2 = 2r, \qquad \underline{x}_1 + 2\underline{x}_2 = 6 + 3r,$$

$$\overline{x}_1 - \underline{x}_2 = 4 - 2r, \quad \overline{x}_1 + 2\overline{x}_2 = 12 - 3r,$$

then

$$\begin{cases} (\overline{x}_1 - \underline{x}_1) + (\overline{x}_2 - \underline{x}_2) = 4 - 4r, \\ (\overline{x}_1 - \underline{x}_1) + 2(\overline{x}_2 - \underline{x}_2) = 6 - 6r, \end{cases}$$

which is equivalent to

$$\begin{cases} w_1 + w_2 = v_1, \\ w_1 + 2w_2 = v_2, \end{cases} \qquad (i)$$

where $v_1 = 4 - 4r$ and $v_2 = 6 - 6r$. Another crisp linear system is

$$\begin{cases} x_1^c - x_2^c = 2 = y_1^c, \\ x_1^c + 2x_2^c = 9 = y_2^c, \end{cases} \qquad (ii)$$

By solving (i) and (ii), we have $w_1 = 2 - 2r,\ w_2 = 2 - 2r,\ x_1^c = \frac{13}{3},\ x_2^c = \frac{7}{3}$ and therefore

$$\underline{x}_1 = \frac{13}{3} - \frac{1}{2}(2 - 2r), \ \ \overline{x}_1 = \frac{13}{3} + \frac{1}{2}(2 - 2r),$$
$$\underline{x}_2 = \frac{7}{3} - \frac{1}{2}(2 - 2r), \ \ \overline{x}_2 = \frac{7}{3} + \frac{1}{2}(2 - 2r).$$

Here $\underline{x}_1 \leq \overline{x}_1,\ \underline{x}_2 \leq \overline{x}_2$; and $\overline{x}_1, \overline{x}_2$ are monotonic non-increasing and $\underline{x}_1,\ \underline{x}_2$ are monotonic non-decreasing functions. Thus the obtained solution $x_1,\ x_2$ is a strong fuzzy solution.

In case of weak solution, we will take in our consideration Theorem (2. 4), a weak fuzzy solution will be obtained in the next example .

Example 2.5:

Consider the 3×3 non-symmetric fuzzy system

$$x_1 - 2x_2 + 3x_3 = (2r, 5 - 3r),$$

$$x_1 - x_2 + x_3 = (-3, -2 - r),$$

$$3x_1 + x_2 + x_3 = (1 + 2r, 3).$$

The two crisp linear systems are

$$\begin{cases} w_1 + 2w_2 + 3w_3 = 5 - 5r, \\ w_1 + w_2 + w_3 = 1 - r, \\ 3w_1 + w_2 + w_3 = 2 - 2r, \end{cases}$$

and

$$\begin{cases} x_1^c - 2x_2^c + 3x_3^c = \dfrac{5 - r}{2}, \\ x_1^c - x_2^c + x_3^c = \dfrac{-5 - r}{2}, \\ 3x_1^c + x_2^c + x_3^c = 2 + r. \end{cases}$$

The solution vectors in parametric form are $W = (0.5 - 0.5r, -3 + 3r, 3.5 - 3.5r)^t$

And $X^c = (-2.5833 - 0.083r, 4.8333 + 0.833r, 4.9167 + 0.4167r)^t$, then we obtain

$$x_1 = (0.1667r - 2.833, -0.333r - 2.333),$$

$$x_2 = (-0.667r + 6.333, 2.333r + 3.333),$$

$$x_3 = (2.167r + 3.167, -1.333r + 6.667).$$

The fact that x_2 is not fuzzy number because. W_2 is negative, the fuzzy solution in this case is a weak solution given by

$$u_1 = (0.1667r - 2.833, -0.333r - 2.333),$$

$$u_2 = (2.333r + 3.333, -0.667r + 6.333\ ),$$

$$u_3 = (2.167r + 3.167, -1.333r + 6.667).$$

2.1.3 Fuzzy Solution by Using Fuzzy Center

This proposed method is based on the use of graphical method for solving a system of n fuzzy linear equations with n variables by using fuzzy center. The original system is replaced by a crisp linear system in which the graphical method can be used to solve it. This method was applied for both symmetric and non-symmetric fuzzy linear system. In comparison with other methods, this method is efficient to obtain the solution, when the number of variables in the fuzzy linear system is large [26].

Remark 2.3 [26]:

By Theorem 2.7, the fuzzy center x_i^c satisfies equation(1.3), consequently we can find x_i^c from the equation (1.3) by using ordinary method.

We can represented the i^{th} equation in (1.3) by the following equivalent form

$$\sum_{a_{ij}\geq 0} a_{ij}\,\underline{x}_j + \sum_{a_{ij}<0} a_{ij}\,\overline{x}_j = \underline{y}_i \qquad (2.15a)$$

$$\sum_{a_{ij}\geq 0} a_{ij}\,\overline{x}_j + \sum_{a_{ij}<0} a_{ij}\,\underline{x}_j = \overline{y}_i \qquad (2.15b)$$

where $x_i^c = \frac{\underline{x_j}(r)+\overline{x_j}(r)}{2}$ and $y_i^c = \frac{\underline{y_j}(r)+\overline{y_j}(r)}{2}$ for $1 \leq i \leq n$.

Theorem 2.6 [26]

The extreme points on the monotonic decreasing solution vector $(\overline{x}_1, \overline{x}_2, \dots\dots\dots, \overline{x}_n)$ can be obtained by replacing $\underline{x}_j$ in terms of $\overline{x}_j$ by using fuzzy center in (2.15b) at $r = 0$ and $r = 1$.

Proof:

As we know $x_i^c = \frac{\underline{x_j} + \overline{x_j}}{2}$, which yields $\underline{x}_j = 2x_j^c - \overline{x}_j$

Replace $\underline{x}_j$ by using the above result in (2.15b) we get

$$\sum_{a_{ij}\geq 0} a_{ij}\,\overline{x}_j + \sum_{a_{ij}<0} a_{ij}\left(2x_j^c - \overline{x}_j\right) = \overline{y}_i\,(r), \quad i = 1,2,\dots,n.$$

which gives,

$$\sum_{a_{ij}\geq 0} a_{ij}\,\overline{x}_j - \sum_{a_{ij}<0} a_{ij}\,\overline{x}_j = \overline{y}_i\,(r) - 2\sum_{a_{ij}<0} a_{ij}\,x_j^c \quad ,i = 1,2,\dots,n \qquad (2.16)$$

Obviously the above equations in (2.16) represents a crisp system when $r = 0$ and $r = 1$.

The crisp system can be solved by ordinary method, thus we have a solution vector

$(\overline{x}_1, \overline{x}_2, \dots\dots\dots, \overline{x}_n)$ at $r = 0$ and $r = 1$.

Theorem 2.7[26]:

The extreme points on the monotonic increasing solution vector $(\underline{x}_1, \underline{x}_2, \dots\dots\dots, \underline{x}_n)$ can be obtained by replacing $\overline{x}_j$ in terms of $\underline{x}_j$ by using fuzzy center in (2.15a) at $r = 0$ and $r = 1$.

In similar manner we can prove the theorem (see [26] for more details).

After identifying the points in the graph by using the previous theorems, so it is possible to find the equation of straight line joining the points by ordinary method. That will give the complete solution to the given system.

The following examples are used to explain the above method.

Example 2.6:

Consider the 2×2 symmetric fuzzy linear system

$$x_1 - x_2 = (2r, 4 - 2r)$$

$$x_1 + 2x_2 = (6 + 3r, 12 - 3r)$$

By using the Theorem 2.7, we have

$$x_1^c - x_2^c = 2$$

$$x_1^c + 2x_2^c = 9$$

Solving, we get

$$x_1^c = \frac{13}{3}$$

$$x_2^c = \frac{7}{3}$$

Now, by using equation(2.15a) we write

$$\underline{x}_1 - \overline{x}_2 = 2r$$

$$\underline{x}_1 + 2\underline{x}_2 = 6 + 3r$$

By replacing $\overline{x}_2$ by $2x_2^c - \underline{x}_2$ in $\underline{x}_1 - \overline{x}_2 = 2r$, and substitute the value of x_2^c we get

$$\underline{x}_1 + \underline{x}_2 = \frac{14}{3} + 2r$$

$$\underline{x}_1 + 2\underline{x}_2 = 6 + 3r$$

Put $r = 0$, thus the above system reduces to a crisp system that can be solved to give

$$\underline{x}_1 at\ r = 0\ is\ \frac{10}{3}$$

$$\underline{x}_2\ at\ r = 0\ is\ \frac{4}{3}$$

Similarly, Put $r = 1$, the above system reduces to a crisp system that gives

$$\underline{x}_1 at\ r = 1\ is\ \frac{13}{3}$$

$$\underline{x}_2\ at\ r = 1\ is\ \frac{7}{3}$$

Now, by plotting the points $(\frac{10}{3}, 0)$ and $(\frac{13}{3}, 1)$ and finding the equation of the straight line joining the two points, we get the required solution for $\underline{x}_1$.

$$\underline{x}_1 = r + \frac{10}{3}.$$

similarly, by plotting the points $(\frac{4}{3}, 0)$ and $(\frac{7}{3}, 1)$ and finding the equation of the straight line joining the two points, we get the required solution for $\underline{x}_2$.

$$\underline{x}_2 = r + \frac{4}{3}.$$

Finally, we use similar method to find $\overline{x}_1$ and $\overline{x}_2$.

$$\overline{x}_1 = \frac{16}{3} - r$$

$$\overline{x}_2 = \frac{10}{3} - r.$$

The graphical solution is shown below in Figure (2.1).

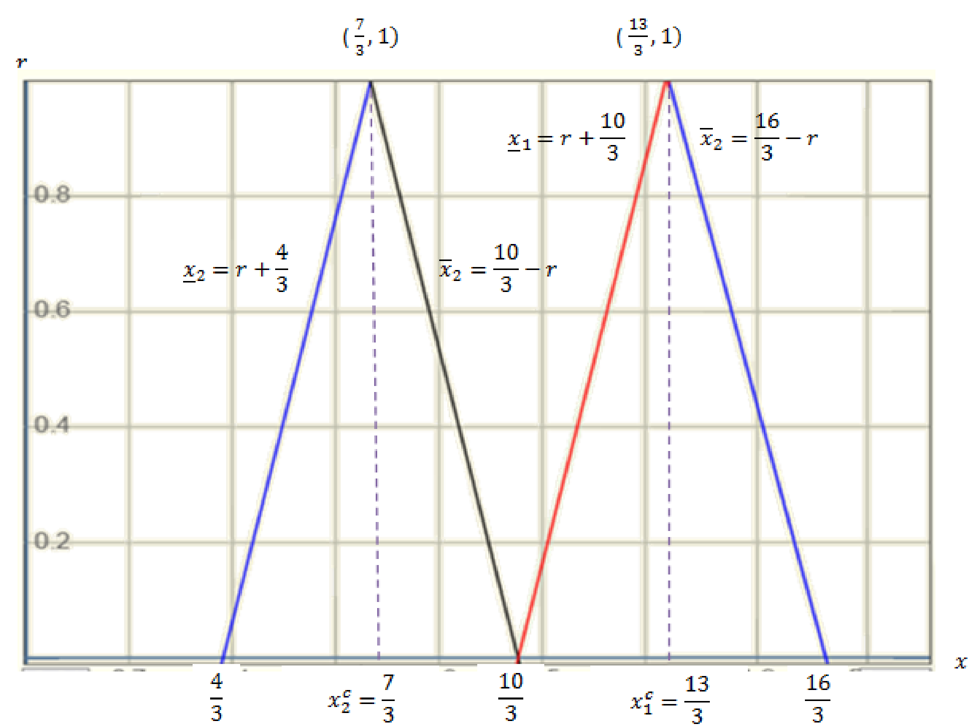


Figure (2.1) : Graphical representation of the solution of example 2.6.

Example 2.7:

Consider the 2×2 non- symmetric fuzzy linear system

$$x_1 - x_2 = (2r, 4 - 2r)$$

$$x_1 + 2x_2 = (6 + 3r, 13 - 4r)$$

By using the Theorem 2.7, we have

$$x_1^c - x_2^c = 2$$
$$x_1^c + 2x_2^c = \frac{19 - r}{2}$$

Solving, we get

$$x_1^c = \frac{27 - r}{6}$$
$$x_2^c = \frac{15 - r}{6}$$

Now, by using equation (2.18) we write

$$\underline{x}_1 - \overline{x}_2 = 2r$$

$$\underline{x}_1 + 2\underline{x}_2 = 6 + 3r$$

By replacing $\overline{x}_2$ by $2x_2^c - \underline{x}_2$ in $\underline{x}_1 - \overline{x}_2 = 2r$, and substitute the value of x_2^cwe get

$$\underline{x}_1 + \underline{x}_2 = \frac{15 + 5r}{3}$$

$$\underline{x}_1 + 2\underline{x}_2 = 6 + 3r$$

Put $r = 0$, thus the above system reduces to crisp system and then solve, we get

$$\underline{x}_1 at\ r = 0\ is\ 4$$

$$\underline{x}_2\ at\ r = 0\ is\ 1$$

Similarly, Put $r = 1$, the above system reduces to crisp system and then solve, we get

$$\underline{x}_1 \, at \; r = 1 \; is \; \frac{13}{3}$$

$$\underline{x}_2 \; at \; r = 1 \; is \; \frac{7}{3}$$

Now, by plotting the points $(4\,,0\,)$and $(\,\frac{13}{3},1)$ and finding the equation of the straight line joining the two points, we get the required solution for $\underline{x}_1$.

$$\underline{x}_1 = \frac{1}{3}r + 4\,.$$

similarly, by plotting the points $(1,0\,)$ and$(\frac{7}{3},1)$ and finding the equation of the straight line joining the two points, we get the required solution for $\underline{x}_2$.

$$\underline{x}_2 = \frac{4}{3}r + 1.$$

Finally, we use similar method to find $\overline{x}_1$ and $\overline{x}_2$.

$$\overline{x}_1 = 5 - \frac{2}{3}r$$

$$\overline{x}_2 = 4 - \frac{5}{3}r$$

The graphical solution is shown below in Figure (2.2).

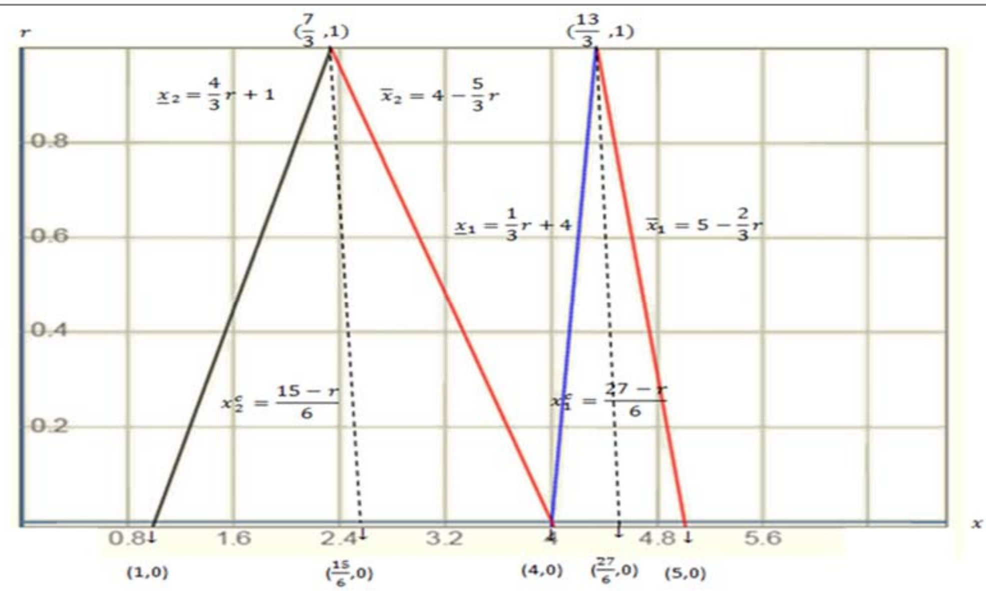


Figure (2.2) : Graphical representation of the solution of example 2.7.

2.1.4 Algorithmic Approach

In this technique the original system is reduced into two equivalent crisp linear systems which can be solved by given algorithm. Also we have showed that this method is applicable for both symmetric and non-symmetric system in addition is suitable to obtain the solution of fuzzy linear system, when the number of variables involved in the linear system is large.

Let's introduce the new technique for getting the solution of linear systems in fuzzy environment. Consider the ith equation of the system (1.3):

$$a_{i1}\left(\underline{x}_1, \overline{x}_1\right) + \ldots + a_{ii}\left(\underline{x}_i, \overline{x}_i\right) + \ldots + a_{in}\left(\underline{x}_n, \overline{x}_n\right) = \left(\underline{y}_i(r), \overline{y}_i(r)\right),$$

we have

$$a_{i1}\underline{x_1} + \cdots + a_{ii}\underline{x_i} + \cdots + a_{in}\underline{x_n} = \underline{y_i}(r)$$

$$a_{i1}\overline{x_1} + \cdots + a_{ii}\overline{x_i} + \cdots + a_{in}\overline{x_n} = \overline{y_i}(r), \qquad 1 \le i \le n, \quad 0 \le r \le 1.$$

As a result of this we have two crisp $n \times n$ linear systems $A\underline{X} = \underline{Y}$ and $A\overline{X} = \overline{Y}$ for all $i, 1 \le i \le n$. Thus, the above system can be extended to two sets of linear systems such as $A\underline{X^0} = \underline{Y^0}$, $A\underline{X^1} = \underline{Y^1}$and $A\overline{X^0} = \overline{Y^0}$, $A\overline{X^1} = \overline{Y^1}$ by replacing $r = 0$ and $r = 1$.

Remark 2.4 [24]:

If $a_{ij} < 0$, then the method can be continued after replacing $\overline{x_i}$ by $2x^c - \underline{x_i}$ in $A\underline{X} = \underline{Y}$ and $\underline{x_i}$ by $2x^c - \overline{x_i}$ in $\overline{X} = \overline{Y}$.

We have introduced the following propositions, to solve the above system.

Proposition 2.1[24]:

The crisp system $A\underline{X} = \underline{Y}$can be divided into two crisp linear systems such as $A\underline{X^0} = \underline{Y^0}$and $A\underline{X^1} = \underline{Y^1}$ by replacing $r = 0$ and $r = 1$ respectively. The extreme solution $\underline{x^0} = (\underline{x_1^0}, \underline{x_2^0}, \dots \underline{x_n^0})$ and $\underline{x^1} = (\underline{x_1^1}, \underline{x_2^1}, \dots \underline{x_n^1})$ can be obtained by directly from the above two crisp systems.

Proposition 2.2[24]:

The crisp system $A\overline{X} = \overline{Y}$ can be divided into two crisp linear systems such as $A\overline{X^0} = \overline{Y^0}$and $A\overline{X^1} = \overline{Y^1}$ by replacing $r = 0$ and $r = 1$ respectively. The extreme solution $\overline{x^0} = (\overline{x_1^0}, \overline{x_2^0}, \dots \overline{x_n^0})$ and $\overline{x^1} = (\overline{x_1^1}, \overline{x_2^1}, \dots \overline{x_n^1})$ can be obtained by directly from the above two crisp systems.

Proposition 2.3[24]:

Let$(\underline{x^0}, \overline{x^0})$ and$(\underline{x^1}, \overline{x^1})$ be the extreme crisp solution at $r = 0$ and $r = 1$ respectively. Then the solution of the fuzzy linear systems (1.3) is obtained by using the extreme solution as $\underline{x_i} = \left(\underline{x_i^1} - \underline{x_i^0}\right) r + \underline{x_i^0}$ and $\overline{x_i} = \left(\overline{x_i^1} - \overline{x_i^0}\right) r + \overline{x_i^0}$ for $i = 1,2, \dots, n$.

Now, to find the solution of the fuzzy linear system (1.3) we will introduce the following algorithm. First, from the matrix $AX = Y$ by using the fuzzy linear system. Extend $n \times n$ system $AX = Y$ into two systems such as $A\underline{X} = \underline{Y}$ and $A\overline{X} = \overline{Y}$. By replacing r as 0 in the above system, we obtain $A\underline{X^0} = \underline{Y^0}$, $A\overline{X^0} = \overline{Y^0}$. Now, this crisp system can be solved by the direct method, we get the extreme crisp solutions $\underline{x^0}$ and $\overline{x^0}$. Repeat the same steps for $r =$

1, we get $\underline{x^1}$ and $\overline{x^1}$. We employ the extreme solution $X^0 = (\underline{x^0}, \overline{x^0})$ and $X^1 = (\underline{x^1}, \overline{x^1})$ to find the solution vector $X = (\underline{x}, \overline{x})$ by $\underline{x_i} = \left(\underline{x_i^1} - \underline{x_i^0}\right) r + \underline{x_i^0}$ and $\overline{x_i} = \left(\overline{x_i^1} - \overline{x_i^0}\right) r + \overline{x_i^0}$ for $i = 1, 2, \ldots, n$.

The following flow chart will illustrate the procedure to find the solution of FLS

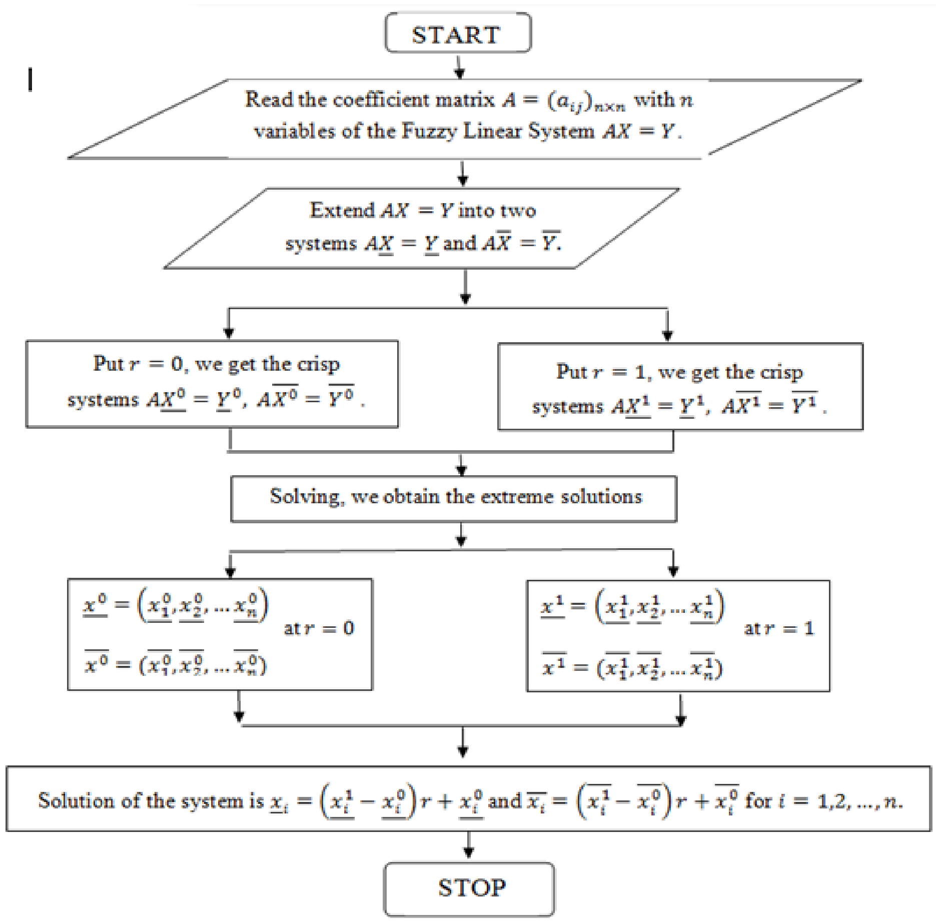


Figure (2.3): Procedure to find the solution of FLS

Thus, we present an example to illustrate the above algorithm.

Example 2.8: Consider the fuzzy linear system

$$x_1 - x_2 = (2r, 4 - 2r)$$

$$x_1 + 2x_2 = (6 + 3r, 12 - 3r)$$

The above system can be written as $SX = Y$

where $S = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 2 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 2 \end{bmatrix}$, $X = \begin{bmatrix} \underline{x_1} \\ \underline{x_2} \\ -\overline{x}_1 \\ -\overline{x}_2 \end{bmatrix}$ and $Y = \begin{bmatrix} 2r \\ 6 + 3r \\ 2r - 4 \\ 3r - 12 \end{bmatrix}$.

By replacing r as 0 in the above system, we get the following crisp system

$$SX^0 = Y^0,$$

where $S = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 2 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 2 \end{bmatrix}$, $X^0 = \begin{bmatrix} \underline{x_1^0} \\ \underline{x_2^0} \\ -\overline{x_1^0} \\ -\overline{x_2^1} \end{bmatrix}$ and $Y^0 = \begin{bmatrix} 0 \\ 6 \\ -4 \\ -12 \end{bmatrix}$.

From the augmented matrix for the system $SX^0 = Y^0$ and solve the system by Gauss Elimination method, we have

$$\underline{x_1^0} = \frac{10}{3}, \qquad \overline{x_1^0} = \frac{16}{3}$$

$$\underline{x_2^0} = \frac{4}{3}, \qquad \overline{x_2^0} = \frac{10}{3}$$

Similarly by replacing r as 1 in the same system, we get the following crisp system $SX^1 = Y^1$,

where $S = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 2 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 2 \end{bmatrix}$, $X^1 = \begin{bmatrix} \underline{x_1^1} \\ \underline{x_2^1} \\ -\overline{x_1^1} \\ -\overline{x_2^1} \end{bmatrix}$ and $Y^1 = \begin{bmatrix} 2 \\ 9 \\ -2 \\ -9 \end{bmatrix}$.

Solving the system $SX^1 = Y^1$by the same method, we have

$$\underline{x_1^1} = \frac{13}{3}, \qquad \overline{x_1^1} = \frac{13}{3}$$

$$\underline{x_2^1} = \frac{7}{3}, \qquad \overline{x_2^1} = \frac{7}{3}$$

By using the following formula

$$\underline{x_i} = \left(\underline{x_i^1} - \underline{x_i^0}\right) r + \underline{x_i^0} \text{ and } \overline{x_i} = \left(\overline{x_i^1} - \overline{x_i^0}\right) r + \overline{x_i^0} \text{ for } i = 1,2,\ldots,n.$$

We get the solution of the given system as

$$\underline{x_1} = r + \frac{10}{3}, \qquad \overline{x_1} = -r + \frac{16}{3},$$

$$\underline{x_2} = r + \frac{4}{3}, \qquad \overline{x_2} = -r + \frac{10}{3}.$$

The graphical representation of the obtained solution is shown in Figure (2.4)

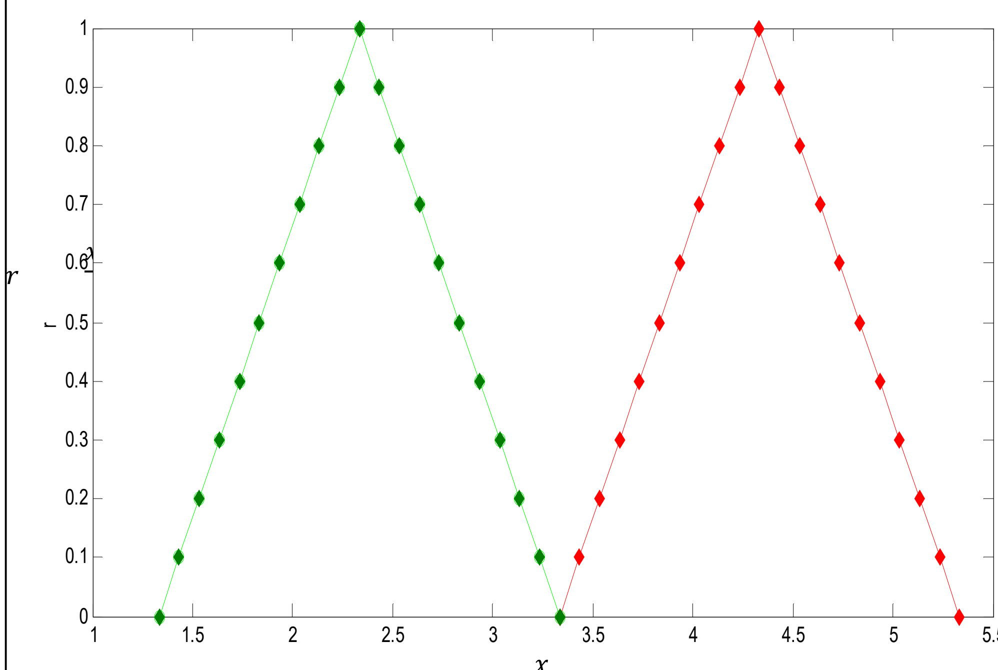


Figure (2.4): Graphical representation of the solution of example 2.8

2.1.5 Embedding Method

In the first we are going to define an embedding map to form a new crisp system.

Definition 2.1[6]:

For an arbitrary fuzzy number $\tilde{x}$ in parametric form the embedding $\pi: \mathbb{R}^2 \to \mathbb{R}^2$ is defined as follows

$$\pi\left(\underline{x}(r), \overline{x}(r)\right) = \left(\overline{x}(r) - \underline{x}(r), \overline{x}(r) + \underline{x}(r)\right). \quad (2.17)$$

Lemma 2.1 [31]:

Let $\check{x} = \left(\underline{x}(r), \overline{x}(r)\right)$, $\check{y} = \left(\underline{y}(r), \overline{y}(r)\right)$ are arbitrary fuzzy numbers and let k be a real number. Then

(i) $\tilde{x} = \tilde{y}$ if and only if $\pi(\tilde{x}) = \pi(\tilde{y})$

(ii) $\pi(\tilde{x} + \tilde{y}) = \pi(\tilde{x}) + \pi(\tilde{y})$

$(iii) \pi(k\tilde{x}) = \pi\left(k\left(\underline{x}(r), \overline{x}(r)\right)\right) = (|k|(\overline{x}(r) - \underline{x}(r), k(\overline{x}(r) + \underline{x}(r)))$

Proof: see[6].

By employ the previous lemma 2.1, system(1.3) can be replaced by the following parametric system:

$$\pi\left(\sum_{j=1}^{n}\left(a_{ij}\left(\underline{x}_j(r), \overline{x}_j(r)\right)\right)\right) = \pi\left(\underline{b}_i(r), \overline{b}_i(r)\right), i = 1,2, \dots, n. \quad (2.18)$$

$$\sum_{j=1}^{n}\left(\pi\left(a_{ij}\left(\underline{x}_j(r), \overline{x}_j(r)\right)\right)\right) = \left(\overline{b}_i(r) - \underline{b}_i(r), \overline{b}_i(r) + \underline{b}_i(r)\right),$$

$$i = 1,2, \dots, n. \quad (2.19)$$

$$\sum_{j=1}^{n}(|a_{ij}|\left(\overline{x}_j(r)-\underline{x}_j(r)\right),a_{ij}\left(\overline{x}_j(r)+\underline{x}_j(r)\right))$$
$$=\left(\overline{b}_i(r)-\underline{b}_i(r),\overline{b}_i(r)+\underline{b}_i(r)\right),i=1,2,\dots,n. \qquad (2.20)$$

$$(\sum_{j=1}^{n}|a_{ij}|\left(\overline{x}_j(r)-\underline{x}_j(r)\right),\sum_{j=1}^{n}a_{ij}\left(\overline{x}_j(r)+\underline{x}_j(r)\right))$$
$$=\left(\overline{b}_i(r)-\underline{b}_i(r),\overline{b}_i(r)+\underline{b}_i(r)\right),i=1,2,\dots,n. \qquad (2.21)$$

So we have now the following equations:

$$\sum_{j=1}^{n}|a_{ij}|\left(\overline{x}_j(r)-\underline{x}_j(r)\right)=\overline{b}_i(r)-\underline{b}_i(r),\quad i=1,2,\dots,n \qquad (2.22)$$

$$\sum_{j=1}^{n}a_{ij}\left(\overline{x}_j(r)+\underline{x}_j(r)\right)=\overline{b}_i(r)+\underline{b}_i(r),\quad i=1,2,\dots,n \qquad (2.23)$$

Thus in order to solve the fuzzy linear system (1.3) we must solve two $(n\times n)$ crisp linear system of equation (2.22) and (2.23).

the matrix form of systems (2.22) and (2.23) is as following:

$$BU=Z,\ AY=W \qquad (2.24)$$

where the coefficients matrix $B=\left[|a_{ij}|\right]_{i,j=1}^{n}$ and $A=\left[a_{ij}\right]_{i,j=1}^{n}$ are crisp $n\times n$ matrices and the right hand side columns are the vectors

$Z=\left(\overline{b}_1(r)-\underline{b}_1(r),\overline{b}_2(r)-\underline{b}_2(r),\dots,\overline{b}_n(r)-\underline{b}_n(r)\right)^T,$

$W=(\overline{b}_1(r)+\underline{b}_1(r),\overline{b}_2(r)+\underline{b}_2(r),\dots,\overline{b}_n(r)+\underline{b}_n(r))^T.$

$U=\left(\overline{x}_1(r)-\underline{x}_1(r),\overline{x}_2(r)-\underline{x}_2(r),\dots,\overline{x}_n(r)-\underline{x}_n(r)\right)^T$ and

$Y=\left(\overline{x}_1(r)+\underline{x}_1(r),\overline{x}_2(r)+\underline{x}_2(r),\dots,\overline{x}_n(r)+\underline{x}_n(r)\right)^T$ are the solutions of the crisp linear systems of equation(2.24).

Theorem 2.8[6]:

The fuzzy linear system (1.3) has a unique solution if and only if the matrices A and B are both nonsingular.

For the proof it is obvious.

Hence the solution vector is unique but it is still not an appropriate fuzzy number vector.

So the following theorems will explain guarantied conditions for receiving fuzzy number vector solution.

In order to obtain an appropriate solution we will use the following theorems.

Theorem 2.9[6]:

The unique solution X of equation(2.22) is nonnegative for arbitrary Z if and only if B^{-1} is nonnegative.

Proof: see [6].

Theorem 2.10 [6]:

The inverse of a nonnegative matrix A is nonnegative if and only if A is a generalized permutation matrix.

Theorem 2.11 [6]:

The fuzzy linear system (1.3) has a fuzzy solution if B^{-1}, $B^{-1} - A^{-1}$, $B^{-1} + A^{-1}$ are nonnegative matrices.

Proof: let $B^{-1} = \left(t_{ij}\right)$and $A^{-1} = \left(s_{ij}\right),\ 1 \leq i,j \leq n$ then

$$U = B^{-1}Z,\ Y = A^{-1}W \tag{2.25}$$

$u_i = \overline{x}_i(r) - \underline{x}_i(r)$ and $y_i = \overline{x}_i(r) + \underline{x}_i(r),\ 1 \le i,j \le n,$ are the solution of equation (2.22) and equation (2.23) respectively. Thus we can write: $\overline{x}_i = \frac{1}{2}(y_i + u_i)$

$$\overline{x}_i = \frac{1}{2}\left(\sum_{j=1}^{n} s_{ij}w_{ij} + \sum_{j=1}^{n} t_{ij}z_{ij}\right) \tag{2.26}$$

With replacement $z_j = \left(\overline{b}_j(r) - \underline{b}_j(r)\right)$ and $w_j = (\overline{b}_j(r) + \underline{b}_j(r))$ in equation (2.26), then we obtain the next result

$$\overline{x}_i = \frac{1}{2}\left(\sum_{j=1}^{n}(s_{ij} + t_{ij})\overline{b}_j + \sum_{j=1}^{n}(s_{ij} - t_{ij})\underline{b}_j\right) \tag{2.27}$$

Since $\overline{b}_j$ is monotonically decreasing and $\underline{b}_j$ is monotonically increasing for all j, and according to assumptions of theorem, $\overline{x}_i$ to be monotonically decreasing. In a similar way: $\underline{x}_i = \frac{1}{2}(y_i - u_i)$is monotonically increasing.

Theorem 2.12 [6]:

with notation of theorem (2.11), the fuzzy linear system (1.3) has a fuzzy number solution, if and only if

$$\begin{cases} u_i \ge 0 \\ \left|\frac{dy_i}{dr}\right| \le -\frac{du_i}{dr} \end{cases} \tag{2.28}$$

where $U = B^{-1}Z$ and $Y = A^{-1}W$.

Proof: let the fuzzy linear system (1.3) has a fuzzy number solution vector $X = (x_1, x_2, \dots, x_n)^T$ which $x_i = (\underline{x}_i(r), \overline{x}_i(r))$. Thus, $u_i = \overline{x}_i(r) - \underline{x}_i(r) \geq 0,\ i = 1,2, \dots, n.$ Since $\overline{x}_i = \frac{1}{2}(y_i + u_i)$ is monotonically decreasing and $\underline{x}_i = \frac{1}{2}(y_i - u_i)$ is monotonically increasing, then $\frac{d\overline{x}_i}{dr} \leq 0$ and $\frac{d\underline{x}_i}{dr} \geq 0$. Therefore $\frac{d(y_i+u_i)}{dr} \leq 0$, $\frac{d(y_i-u_i)}{dr} \geq 0$ i.e. $-\frac{du_i}{dr} \geq \frac{dy_i}{dr}$ and $-\frac{du_i}{dr} \geq -\frac{dy_i}{dr}$. Consequently, $\left|\frac{dy_i}{dr}\right| \leq -\frac{du_i}{dr}$. Conversely is obvious.

Example 2.9:

Consider the 2×2 fuzzy linear system

$$x_1 - x_2 = (2r, 4 - 2r)$$

$$x_1 + 2x_2 = (6 + 3r, 13 - 4r)$$

$det(A) = 3$ and $\det(B) = 1$, consequently, equation(2.22) and equation (2.23) will have solution as follow:

$$U = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} \overline{x}_1(r) - \underline{x}_1(r) \\ \overline{x}_2(r) - \underline{x}_2(r) \end{pmatrix} = B^{-1}Z = \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} 4 - 4r \\ 7 - 7r \end{pmatrix} = \begin{pmatrix} 1 - r \\ 3 - 3r \end{pmatrix}$$

$$Y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} \overline{x}_1(r) + \underline{x}_1(r) \\ \overline{x}_2(r) + \underline{x}_2(r) \end{pmatrix} = A^{-1}W = \begin{pmatrix} \frac{2}{3} & \frac{1}{3} \\ -\frac{1}{3} & \frac{1}{3} \end{pmatrix} \begin{pmatrix} 4 \\ 19 - r \end{pmatrix}$$

$$= \begin{pmatrix} 9 - \frac{r}{3} \\ 5 - \frac{r}{3} \end{pmatrix}$$

$\forall r, 0 \leq r \leq 1,\ u_1 = 1 - r$ and $u_2 = 3 - 3r,$ both are nonnegative.

Also $\forall r, 0 \leq r \leq 1, \left|\frac{dy_i}{dr}\right| \leq -\frac{du_i}{dr}$, $i = 1,2.$ So the result will be

$$\overline{x}_1 = \frac{1}{2}(y_1 + u_1) = 5 - 0.667r,$$

$$\underline{x}_1 = \frac{1}{2}(y_1 - u_1) = 4 + 0.333r,$$

$$\overline{x}_2 = \frac{1}{2}(y_2 + u_2) = 4 - 1.667r,$$
$$\underline{x}_2 = \frac{1}{2}(y_2 - u_2) = 1 + 1.333r.$$

Therefore, the fuzzy number solution is

$$x_1 = \left(\underline{x}_1(r), \overline{x}_1(r)\right) = (4 + 0.333r, 5 - 0.667r),$$
$$x_2 = \left(\underline{x}_2(r), \overline{x}_2(r)\right) = (1 + 1.333r, 4 - 1.667r).$$

A weak fuzzy solution will be obtained in the next example.

Example 2.10:

consider the 3×3 fuzzy system

$$-x_1 + x_2 + x_3 = (-2, -1 - r),$$
$$x_1 - 2x_2 + x_3 = (2 + r, 3),$$
$$3x_1 + x_2 + 2x_3 = (r, 2 - r),$$

$det(A) = 13$ and $\det(B) = -1$, consequently, equation (2.22) and equation (2.23) will have solution as follow:

$$U = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} \overline{x}_1(r) - \underline{x}_1(r) \\ \overline{x}_2(r) - \underline{x}_2(r) \\ \overline{x}_3(r) - \underline{x}_3(r) \end{pmatrix} = B^{-1}Z = \begin{pmatrix} -3 & 1 & 1 \\ -1 & 1 & 0 \\ 5 & -2 & -1 \end{pmatrix} \begin{pmatrix} 1 - r \\ 1 - r \\ 2 - 2r \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 - r \end{pmatrix}$$

$$Y = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} \overline{x}_1(r) + \underline{x}_1(r) \\ \overline{x}_2(r) + \underline{x}_2(r) \\ \overline{x}_3(r) + \underline{x}_3(r) \end{pmatrix} = A^{-1}W = \begin{pmatrix} -0.385 & -0.077 & 0.231 \\ 0.077 & -0.385 & 0.154 \\ 0.538 & 0.308 & 0.077 \end{pmatrix} \begin{pmatrix} -3 - r \\ 5 + r \\ 2 \end{pmatrix}$$
$$= \begin{pmatrix} 1.232 - 0.308r \\ -1.848 - 0.462r \\ 0.08 - 0.23r \end{pmatrix}$$

$\forall r, 0 \le r \le 1,\ u_1 = 0, u_2 = 0$ and $u_3 = 1 - r$ are nonnegative.

Also $\quad \forall r, 0 \le r \le 1, \left|\frac{dy_1}{dr}\right| \ge -\frac{du_1}{dr}, \left|\frac{dy_2}{dr}\right| \ge -\frac{du_2}{dr}, \left|\frac{dy_3}{dr}\right| \le -\frac{du_3}{dr}$

according to that this (FLS) will not have fuzzy number solution.

2.1.6 *LU* Decomposition Method

Theorem 2.13 [2]:

Let A be an $n \times n$ matrix with all non-zero leading principal minors. Then A has a unique factorization:

$$A = LU,$$

Where L is unit lower triangular matrix and U is upper triangular matrix.

In order to decomposition of matrix S, we must find both matrices L and U such that $S = LU$, where

$$L = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix}, \qquad U = \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix},$$

Where L_{11} and L_{22} are lower triangular matrices, U_{11} and U_{22} are upper triangular matrices.

Now we suppose that $A = B - C$ has LU decomposition. So we have

$$S = \begin{bmatrix} B & C \\ C & B \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix},$$

then

$$B = L_{11}U_{11}, \tag{2.29}$$

$$C = L_{11}U_{12} \Rightarrow U_{12} = L_{11}^{-1}C,$$

$$C = L_{21}U_{11} \Rightarrow L_{21} = CU_{11}^{-1},$$

$$B = L_{21}U_{12} + L_{22}U_{22},$$

Now we can write

$$B - CB^{-1}C = L_{22}U_{22}. \tag{2.30}$$

From (2.29) and (2.30) if B and $B - CB^{-1}C$ both have LU decomposition, then S has LU decomposition.

Theorem 2.14 [2]:

Let S be an $n \times n$ symmetric positive definite matrix then there exists a unique lower triangular matrix L with positive diagonal entries such that $S = LL^T$.

Therefore if the matrix S be a symmetric positive definite matrix then we have

$$S = \begin{bmatrix} B & C \\ C & B \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} L_{11}^T & L_{21}^T \\ 0 & L_{22}^T \end{bmatrix},$$

then

$$B = L_{11}L_{11}^T, \tag{2.31}$$

$$C = L_{11}L_{21}^T \Rightarrow L_{21}^T = L_{11}^{-1}C,$$

$$C = L_{21}L_{11}^T \Rightarrow L_{21} = C(L_{11}^T)^{-1},$$

$$B = L_{21}L_{21}^T + L_{22}L_{22}^T,$$

thus

$$B - CB^{-1}C = L_{22}L_{22}^T. \tag{2.32}$$

By using Theorem (2.14) in LU decomposition method, the matrices B and $B - CB^{-1}C$ should be symmetric positive definite.

Example 2.11:

Consider the 2×2 non- symmetric fuzzy linear system

$$2x_1 + 3x_2 = (2 + 2r, 8 - 4r),$$

$$5x_1 - x_2 = (4r,\ 6 - 2r).$$

The extended 4×4 matrix is

$$S = \begin{pmatrix} 2 & 3 & 0 & 0 \\ 5 & 0 & 0 & 1 \\ 0 & 0 & 2 & 3 \\ 0 & 1 & 5 & 0 \end{pmatrix}$$

and

$$B = \begin{pmatrix} 2 & 3 \\ 5 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 4 & 1 \end{pmatrix} \begin{pmatrix} 5 & 0 \\ 0 & 3 \end{pmatrix},$$

$$B - CB^{-1}C = \begin{pmatrix} 2 & 3 \\ 5 & 0.133 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0.4 & 1 \end{pmatrix} \begin{pmatrix} 5 & 0.133 \\ 0 & 2.947 \end{pmatrix}$$

and hence

$$S = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0.4 & 1 & 0 & 0 \\ 0 & 0.333 & 1 & 0 \\ 0 & 0 & 0.4 & 1 \end{pmatrix} \begin{pmatrix} 5 & 0 & 0 & 1 \\ 0 & 3 & 0 & -0.4 \\ 0 & 0 & 5 & 0.133 \\ 0 & 0 & 0 & 2.947 \end{pmatrix}$$

Now the exact solution is

$$x_1 = (\underline{x}_1(r), \overline{x}_1(r)) = (0.362 + 0.579r, 1.285 - 0.344r),$$

$$x_2 = (\underline{x}_2(r), \overline{x}_2(r)) = (0.425 + 0.281r, 1.809 - 1.104r).$$

The exact and the approximate solution are show in figure (2.5).

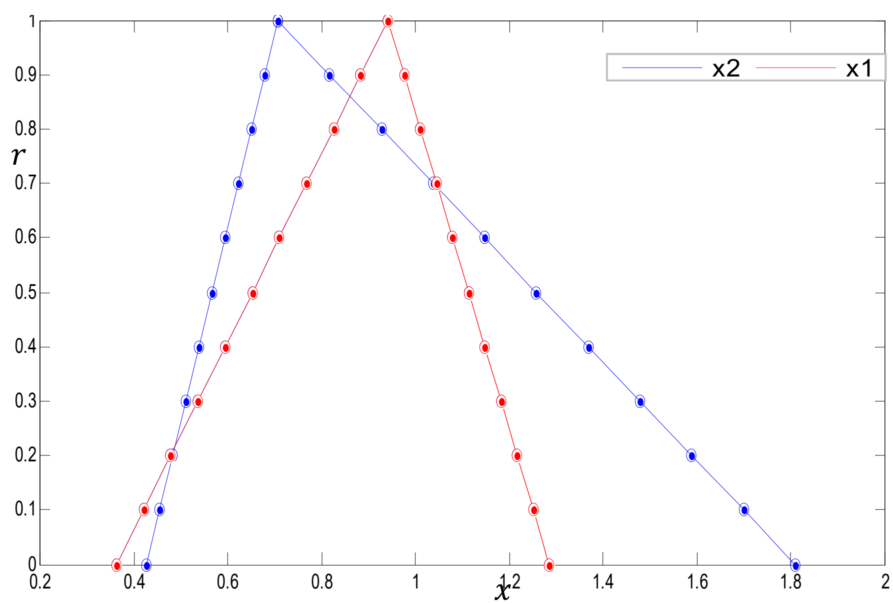


Figure (2.5): Graphical representation of the solution of example 2.11

Example 2.12:

Consider the 2×2 symmetric fuzzy linear system

$$x_1 - x_2 = (2r, 4 - 2r)$$

$$x_1 + 2x_2 = (6 + 3r, 12 - 3r)$$

The extended 4×4 matrix is

$$S = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 2 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 2 \end{pmatrix}$$

and

$$B = \begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix},$$

$$B - CB^{-1}C = \begin{pmatrix} 1 & 0.5 \\ 1 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0.5 \\ 0 & 1.5 \end{pmatrix}$$

and hence

$$S = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0.5 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 2 & 0 & -1 \\ 0 & 0 & 1 & 0.5 \\ 0 & 0 & 0 & 1.5 \end{pmatrix}$$

Now the exact solution is

$$x_1 = (\underline{x}_1(r), \overline{x}_1(r)) = (3.333 + r, 5.333 - r),$$

$$x_2 = (\underline{x}_2(r), \overline{x}_2(r)) = (1.333 + r, 3.333 - r).$$

The exact and the approximate solution using LU decomposition are show in figure (2.6).

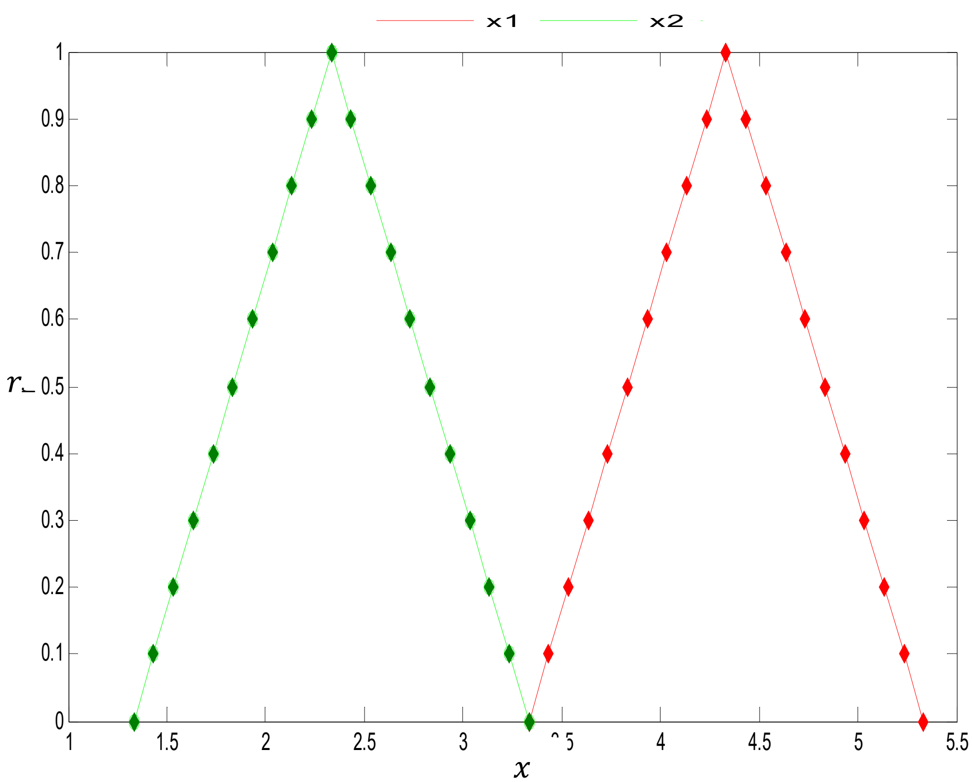


Figure (2.6): Graphical representation of the solution of example 2.12

2.1.7 *LU*-Decomposition Method of Mansouri and Asady

In this subsection we want to proposed the efficient of this method and compared with Abbasbandy [2] for solve a large linear system and extension to very large system.

Theorem 2.15 [21]:

Suppose A is nonsingular square matrix, then A has a unique decomposition such that

$A = LU$

where L is a unit lower triangular matrix and U is upper triangular matrix.

Theorem 2.16 [21]:

Let A be an $n \times n$ symmetric positive definite matrix then there exists a unique lower triangular matrix L with positive diagonal entries s.t

$A = LL^t$

Now as we show in the previous subsection (2.1.6), we can factor the matrix A into LU using $l_{ii} = 1$. Thus to solve the linear system $LUX = b$

we solve the system $LZ = b$

$$\begin{cases} z_1 & = b_1, \\ l_{21}z_1 + l_{22}z_2 & = b_2, \\ \vdots & \\ l_{k1}z_1 + l_{k2}z_2 + \cdots\cdots + l_{k(k-1)}z_{k-1} + z_k & = b_k, \\ \vdots & \\ l_{n1}z_1 + l_{n2}z_2 + \cdots\cdots + l_{n(n-1)}z_{n-1} + z_n & = b_n. \end{cases} \tag{2.33}$$

by forward substitution, and we obtain fuzzy solution $Z = (z_1, z_2, \dots, z_n)^t$ which we put it into upper fuzzy linear system $UX = Z$

$$\begin{cases} u_{11}x_1 + u_{12}x_2 + \cdots\cdots + u_{1n}x_n = z_1, \\ u_{22}x_2 + \cdots\cdots + u_{2n}x_n = z_2, \\ \vdots \\ u_{kk}x_k + \cdots\cdots + u_{kn}x_n = z_k, \\ \vdots \\ u_{nn}x_n = z_n, \end{cases} \tag{2.34}$$

and we solve this system using the backward substitution.

Example 2.13:

Consider the 3×3 non-symmetric fuzzy system

$$3x_1 + x_2 - x_3 = (2r, 5 - 3r),$$

$$-x_1 + 3x_2 + 2x_3 = (-3, -2 - r),$$

$$x_1 + x_2 + 3x_3 = (1 + 2r, 3).$$

When we solving this system by using $LU-$decomposition method, we obtain

$$A=\begin{bmatrix}3&1&-1\\-1&3&2\\1&1&3\end{bmatrix}=\begin{bmatrix}1&0&0\\-0.333&1&0\\0.333&0.2&1\end{bmatrix}\begin{bmatrix}3&1&-1\\0&3.333&1.667\\0&0&3\end{bmatrix}$$

To solve the given system, we use forward substitution to solve $LZ=b$, that is

$$\begin{bmatrix}1&0&0\\-0.333&1&0\\0.333&0.2&1\end{bmatrix}\begin{bmatrix}z_1\\z_2\\z_3\end{bmatrix}=\begin{bmatrix}(2r,5-3r)\\(-3,-2-r)\\(1+2r,3\ \ )\end{bmatrix}$$

This yields,

$$Z=\begin{bmatrix}z_1\\z_2\\z_3\end{bmatrix}=\begin{bmatrix}(2r,5-3r)\\(-3+0.667r,-0.333-2r)\\(-0.6+3.4r,3.6-0.8r\ \ )\end{bmatrix}$$

Finally, we solve the system $UX=Z$ using backward substitution, that is

$$\begin{bmatrix}3&1&-1\\0&3.333&1.667\\0&0&3\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}=\begin{bmatrix}(2r,5-3r)\\(-3+0.667r,-0.333-2r)\\(-0.6+3.4r,3.6-0.8r\ \ )\end{bmatrix}$$

Then, we obtain

$$X=\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}=\begin{bmatrix}(-0.0667+4.933r,7.0667-2.2r)\\(-15+3.333r,-1.1667r)\\(-0.2+1.1333r,1.2-0.2667r\ \ )\end{bmatrix}$$

In the following example we will compare Mansouri and Asady methods with Abbasbandy method [2].

Example 2.14:

Consider the 3×3 symmetric fuzzy system

$$4x_1 + x_2 - x_3 = (1 + r, 3 - r),$$

$$-x_1 + x_2 + x_3 = (2 + r, 3),$$

$$2x_1 + x_2 + x_3 = (-2, -1 - r).$$

when we solve this system using LU –decomposition method, we obtain

$$A = \begin{bmatrix} 4 & 1 & -1 \\ -1 & 1 & 1 \\ 2 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -0.25 & 1 & 0 \\ 0.5 & 0.4 & 1 \end{bmatrix} \begin{bmatrix} 4 & 1 & -1 \\ 0 & 1.25 & 0.75 \\ 0 & 0 & 1.2 \end{bmatrix}$$

To solve the given system, we use forward substitution to solve $LZ = b$, that is

$$\begin{bmatrix} 1 & 0 & 0 \\ -0.25 & 1 & 0 \\ 0.5 & 0.4 & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} (1 + r, 3 - r) \\ (2 + r, 3) \\ (-2, -1 - r) \end{bmatrix}$$

This yields,

$$Z = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} (1 + r, 3 - r) \\ (2.25 + 1.25r, 3.75 - 0.25r) \\ (-5 + 0.6r, -2.4 - 2r) \end{bmatrix}$$

Finally, we solve the system $UX = Z$ using backward substitution, that is

$$\begin{bmatrix} 4 & 1 & -1 \\ 0 & 1.25 & 0.75 \\ 0 & 0 & 1.2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} (1 + r, 3 - r) \\ (2.25 + 1.25r, 3.75 - 0.25r) \\ (-5 + 0.6r, -2.4 - 2r) \end{bmatrix}$$

Then, we obtain

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} (-2.1667 + 0.25r, -0.5 - 1.1667r) \\ (3 + 2r, 5.5 - 0.5r) \\ (-2.1667 + 0.5r, -0.5 - 1.1667r) \end{bmatrix}$$

Clearly in this example A is nonsingular but the 6×6 crisp matrix S in the following form

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

is a singular matrix, and therefore the proposed methods in (Abbasbandy et al) [2] can not be used to solve this system.

2.2 Analytical Methods for Solving Fully Fuzzy Linear System of Equations(FFLS)

In this section, we will discuss the third category of fuzzy system of linear equations where all the coefficient matrix arrays, the right-hand side arrays and the unknowns, are fuzzy numbers, we will apply the matrix inversion method, Cramer's rule and LU decomposition method.

Our target in this section to obtain a positive solution of a fully fuzzy linear system (1.9) where $\tilde{A} = (A, M, N) > 0, \tilde{b} = (b, g, h) > 0$ and $\tilde{x} = (x, y, z) > 0$. Thus we have

$$(A, M, N) \otimes (x, y, z) = (b, g, h). \tag{2.35}$$

In this section some direct methods to solve the Equation(1.8) is presented:

2.2.1 Matrix Inversion Method [13]

By using the approximation formula for the extended multiplication of two fuzzy numbers Equation(2.35) may be written as

$$(Ax, Ay + Mx, Az + Nx) = (b, g, h)$$

Now using definition (1.14), we get

$$Ax = b,$$
$$Ay + Mx = g, \tag{2.36}$$
$$Az + Nx = h.$$

i.e.

$$Ax = b,$$
$$Ay = g - Mx, \tag{2.37}$$
$$Az = h - Nx.$$

We assume that A is nonsingular matrix, thus equation (2.37) may be written as

$$x = A^{-1}b,$$
$$y = A^{-1}g - A^{-1}Mx, \tag{2.38}$$
$$z = A^{-1}h - A^{-1}Nx.$$

Therefore , the fuzzy solution (x, y, z)can be easily obtained by using the above equation (2.38).

Example 2.15:

Consider the fully fuzzy linear system of equations:

$$(5,1,1) \otimes (x_1, y_1, z_1) \oplus (6,1,2) \otimes (x_2, y_2, z_2) = (50,10,17)$$
$$(7,1,0) \otimes (x_1, y_1, z_1) \oplus (4,0,1) \otimes (x_2, y_2, z_2) = (48,5,7)$$

thus we have

$$A = \begin{bmatrix} 5 & 6 \\ 7 & 4 \end{bmatrix}, \quad M = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}, \quad N = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}$$
$$b = \begin{bmatrix} 50 \\ 48 \end{bmatrix}, \quad g = \begin{bmatrix} 10 \\ 5 \end{bmatrix}, \quad h = \begin{bmatrix} 17 \\ 7 \end{bmatrix}$$

So

$$\begin{bmatrix} 5 & 6 \\ 7 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 50 \\ 48 \end{bmatrix} \Rightarrow \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 4 \\ 5 \end{bmatrix}$$

Similarly

$$\begin{bmatrix}5 & 6\\7 & 4\end{bmatrix}\begin{bmatrix}y_1\\y_2\end{bmatrix}=\begin{bmatrix}5\\10\end{bmatrix}-\begin{bmatrix}0 & 1\\1 & 1\end{bmatrix}\begin{bmatrix}4\\5\end{bmatrix}\Rightarrow \begin{bmatrix}y_1\\y_2\end{bmatrix}=\begin{bmatrix}\frac{1}{11}\\\frac{1}{11}\end{bmatrix}$$

and

$$\begin{bmatrix}5 & 6\\7 & 4\end{bmatrix}\begin{bmatrix}z_1\\z_2\end{bmatrix}=\begin{bmatrix}17\\7\end{bmatrix}-\begin{bmatrix}1 & 2\\0 & 1\end{bmatrix}\begin{bmatrix}4\\5\end{bmatrix}\Rightarrow \begin{bmatrix}z_1\\z_2\end{bmatrix}=\begin{bmatrix}0\\\frac{1}{2}\end{bmatrix}$$

Therefore the solution is

$$\tilde{x}=\begin{bmatrix}\left(4,\frac{1}{11},0\right)\\\left(5,\frac{1}{11},\frac{1}{2}\right)\end{bmatrix}$$

2.2.2 Cramer's rule [13]

Cramer's rule is another method for solving the fully fuzzy linear system of equations, which states that each entry x_i in the solution is a quotient of two determinants.

For solving FFLS (1.9) with this method, consider equation (2.37). So we may write

$$x_i=\frac{\det(A^{(i)})}{\det(A)},\quad i=1,2,\dots,n$$

where $A^{(i)}$ denotes the matrix which obtained from A by replacing its i^{th} column by b . then using solution x , we have

$$y_i=\frac{\det(A'^{(i)})}{\det(A)},\quad i=1,2,\dots,n$$

$$z_i = \frac{\det(A''^{(i)})}{\det(A)}, \quad i = 1,2,\dots,n$$

where$A'^{(i)}$ and $A''^{(i)}$ denotes matrix which obtained from A by replacing its i^{th} column by $g - Mx$ and $h - Nx$, respectively.

Example 2.16:

Consider the following fully fuzzy linear system of equations:

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

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

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

In matrix form

$$\begin{bmatrix} (4,3,2) & (5,2,1) & (3,0,3) \\ (7,4,3) & (10,6,5) & (2,1,1) \\ (6,2,2) & (7,1,2) & (15,5,4) \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} (71,54,76) \\ (118,115,129) \\ (155,89,151) \end{bmatrix}$$

Thus we have

$$A = \begin{bmatrix} 4 & 5 & 3 \\ 7 & 10 & 2 \\ 6 & 7 & 15 \end{bmatrix}, \quad M = \begin{bmatrix} 3 & 2 & 0 \\ 4 & 6 & 1 \\ 2 & 1 & 5 \end{bmatrix}, \quad N = \begin{bmatrix} 2 & 1 & 3 \\ 3 & 5 & 1 \\ 2 & 2 & 4 \end{bmatrix}$$

$$b = \begin{bmatrix} 71 \\ 118 \\ 155 \end{bmatrix}, \quad g = \begin{bmatrix} 54 \\ 115 \\ 89 \end{bmatrix}, \qquad h = \begin{bmatrix} 76 \\ 129 \\ 151 \end{bmatrix}$$

Where $det\ (\ A) = 46$

Now we calculate A^1 ,A^2 and A^3 which obtained from A by replacing its i^{th} column by b .

$$A^1 = \begin{bmatrix} 71 & 5 & 3 \\ 118 & 10 & 2 \\ 155 & 7 & 15 \end{bmatrix} \Rightarrow \det(A^1) = 184$$

$$A^2 = \begin{bmatrix} 4 & 71 & 3 \\ 7 & 118 & 2 \\ 6 & 155 & 15 \end{bmatrix} \Rightarrow \det(A^2) = 368$$

$$A^3 = \begin{bmatrix} 4 & 5 & 71 \\ 7 & 10 & 118 \\ 6 & 7 & 155 \end{bmatrix} \Rightarrow \det(A^3) = 230$$

Therefore we have, $x_1 = \frac{184}{46} = 4,\ x_2 = \frac{368}{46} = 8$ and $x_3 = \frac{230}{46} = 5$

i.e.
$$x = \begin{bmatrix} 4 \\ 8 \\ 5 \end{bmatrix}$$

Now to calculate y and z we first need to calculate $A'^{(i)}$ and $A''^{(i)}$ denotes matrix which obtained from A by replacing its i^{th} column by $g - Mx$ and $h - Nx$, respectively.

$$g - Mx = \begin{bmatrix} 45 \\ 115 \\ 89 \end{bmatrix} - \begin{bmatrix} 3 & 2 & 0 \\ 4 & 6 & 1 \\ 2 & 1 & 5 \end{bmatrix}\begin{bmatrix} 4 \\ 8 \\ 5 \end{bmatrix} = \begin{bmatrix} 26 \\ 46 \\ 48 \end{bmatrix}$$

$$h - Nx = \begin{bmatrix} 76 \\ 129 \\ 151 \end{bmatrix} - \begin{bmatrix} 2 & 1 & 3 \\ 3 & 5 & 1 \\ 2 & 2 & 4 \end{bmatrix}\begin{bmatrix} 4 \\ 8 \\ 5 \end{bmatrix} = \begin{bmatrix} 45 \\ 72 \\ 107 \end{bmatrix}$$

Now,

$$A'^{(1)} = \begin{bmatrix} 26 & 5 & 3 \\ 46 & 10 & 2 \\ 48 & 7 & 15 \end{bmatrix} \Rightarrow \det\left(A'^{(1)}\right) = 92$$

$$A'^{(2)} = \begin{bmatrix} 4 & 26 & 3 \\ 7 & 46 & 2 \\ 6 & 48 & 15 \end{bmatrix} \Rightarrow \det\left(A'^{(2)}\right) = 138$$

$$A'^{(3)} = \begin{bmatrix} 4 & 5 & 26 \\ 7 & 10 & 46 \\ 6 & 7 & 48 \end{bmatrix} \Rightarrow \det\left(A'^{(3)}\right) = 46$$

$$y_1 = \frac{92}{46} = 2,\ y_2 = \frac{138}{46} = 3 \text{ and } y_3 = \frac{46}{46} = 1$$

$$y = \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix}$$

Similarly, $$A''^{(1)} = \begin{bmatrix} 45 & 5 & 3 \\ 72 & 10 & 2 \\ 107 & 7 & 15 \end{bmatrix} \Rightarrow \det\left(A''^{(1)}\right) = 92$$

$$A''^{(2)} = \begin{bmatrix} 4 & 45 & 3 \\ 7 & 72 & 2 \\ 6 & 107 & 15 \end{bmatrix} \Rightarrow \det\left(A''^{(2)}\right) = 230$$

$$A''^{(3)} = \begin{bmatrix} 4 & 5 & 45 \\ 7 & 10 & 72 \\ 6 & 7 & 1073 \end{bmatrix} \Rightarrow \det\left(A''^{(3)}\right) = 187$$

So, $z_1 = \frac{92}{46} = 2,\ z_2 = \frac{230}{46} = 5 \text{ and } z_3 = \frac{184}{46} = 4$

$$z = \begin{bmatrix} 2 \\ 5 \\ 4 \end{bmatrix}$$

Therefore the solution of this problem is

$$\tilde{x} = \begin{bmatrix} (4,2,2) \\ (8,3,5) \\ (5,1,4) \end{bmatrix}$$

2.2.3 *LU* decomposition method for solving FFLS[13]

The coefficients matrix of the linear system of equations in the LU decomposition method is factored into the product of two lower and upper triangular matrices. This method is frequently used to solve a large system of equations. Consider the system of equation (1.9), where $\tilde{A}$ is a non-singular matrix. we start by writing the matrix $\tilde{A}$ as the product of a lower triangular matrix L and an upper triangular matrix U in the following form

$$\tilde{A} = \tilde{L} \otimes \tilde{U},$$

Where $\tilde{A} = (A, M, N)$, $\tilde{L} = (L_1, L_2, L_3)$ and $\tilde{U} = (U_1, U_2, U_3)$.

Thus we have

$$(A, M, N) = (L_1, L_2, L_3) \otimes (U_1, U_2, U_3)$$

$$(A, M, N) = (L_1U_1, L_1U_2 + L_2U_1, L_1U_3 + L_3U_1)$$

i.e.

$$A = L_1U_1, \tag{2.39}$$

$$M = L_1U_2 + L_2U_1, \tag{2.40}$$

$$N = L_1U_3 + L_3U_1. \tag{2.41}$$

In order to obtain the unique solution we either set all the diagonal elements of L as 1 or all the diagonal elements of U as 1. For $U_{ii} = 1,\ i = 1,2, \dots, n$, this method is called the Crout's LU decomposition method and for $L_{ii} = 1$,

$i = 1,2,\dots,n,$ this is called Doolittle's method. Here in this chapter we will use Doolittle's factorization method.

First of all we calculate L_1 and U_1 such that $A = L_1U_1$, where L_1 is a lower triangular crisp matrix, having the diagonal of 1's and U_1 is an upper triangular crisp matrix with the general diagonal.

$$\begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ l_{21} & 1 & 0 & \dots & 0 \\ l_{31} & l_{32} & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \dots & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} & \dots & u_{1n} \\ 0 & u_{22} & u_{23} & \dots & u_{2n} \\ 0 & 0 & u_{33} & \dots & u_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & u_{nn} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{bmatrix}$$

which amounts to n^2 equations in the n^2 unknowns l_{ij} and u_{ij}. The computations runs as the following:

$$u_{1j} = a_{1j}, \quad j = 1,2,\dots,n. \tag{2.42}$$

$$l_{i1}u_{11} = a_{i1} \Rightarrow l_{i1} = \frac{a_{i1}}{u_{11}}, \quad i = 1,2,\dots,n. \tag{2.43}$$

Continuing in a recursive way for $r = 2,3,\dots,n,$ we alternatively get the rows of U_1 and corresponding columns of L_1 to be

$$u_{rj} = a_{rj} - \sum_{k=1}^{r-1} l_{rk}u_{kj}, \qquad j = r, r+1,\dots,n. \tag{2.44}$$

Each row will follow by the corresponding column of L_1

$$l_{rj} = \frac{a_{ir} - \sum_{k=1}^{r-1} l_{ik}u_{kr}}{u_{rr}} \qquad i = r, r+1,\dots,n, \tag{2.45}$$

We place the diagonals of L_2 and L_3 to be consist of 0's not 1's.

By using equation(2.40), and $l_2 = \left(l'_{ij}\right)$ with diagonals of 0's and $U_2 = \left(U'_{ij}\right)$ we may write

$$m_{ij} = \sum_{k=1}^{n} l_{ik} u'_{kj}, \quad 1 \le i, j \le n, \; l'_{ii} = 0 \tag{2.46}$$

Since L_1 and U_1 in hand, we can continue our approach to the second step for L_2 and U_2 as follows:

$$u'_{1j} = m_{1j}, \quad j = 1,2, \dots, n, \tag{2.47}$$

$$l'_{i1} = \frac{m_{i1} - l_{i1} u'_{11}}{u_{11}}, \quad i = 1, \dots, n \tag{2.48}$$

We continue in a recursive way, for $r = 2,3, \dots, n$ we alternatively find the rows of U_1 and corresponding columns of L_1 to be

$$u'_{rj} = m_{rj} - \sum_{k=1}^{r-1} (l_{rk} u'_{kj}), \quad j = r, r+1, \dots, n, \tag{2.49}$$

$$l'_{ir} = \frac{m_{ir} - \sum_{k=1}^{r-1} l'_{ik} u_{kr} - \sum_{k=1}^{r} l_{ik} u'_{kr}}{u_{rr}}, \quad i = r, r+1 \dots, n \tag{2.50}$$

Similarly by equation(2.41), and $L_3 = (l''_{ij})$ and $U_3 = (U''_{ij})$we may write

$$n_{ij} = \sum_{k=1}^{n} l_{ik} u''_{kj} + l''_{ik} u_{kj}, \quad 1 \le i, j \le n. \tag{2.51}$$

By continue our approach to the second step for find L_3 and U_3 as follows:

$$u''_{1j} = n_{1j}, \;\; j = 1, \dots, n,$$

$$l''_{i1} = \frac{n_{i1} - l_{i1}u''_{11}}{u_{11}}, \quad i = 1, \dots, n. \tag{2.52}$$

Finally we find the rows of U_3 and the corresponding columns of L_3 for $r = 2,3, \dots, n$ to be as follow:

$$u''_{rj} = n_{rj} - \sum_{k=1}^{r-1} \left(l_{rk}u''_{kj} + l''_{rk}u_{kj} \right), \quad j = r, r+1, \dots, n,$$

$$l''_{ir} = \frac{n_{ir} - \sum_{k=1}^{r-1} l''_{ik}u_{kr} - \sum_{k=1}^{r} l_{ik}u''_{kr}}{u_{rr}}, \;\; i = r, r+1 \dots, n \tag{2.53}$$

The solution to the problem $\tilde{A} \otimes \tilde{x} = \tilde{b}$ could be obtained by a two step triangular solve process

$$\widetilde{A} \otimes \tilde{x} = \tilde{b}$$

$$\tilde{L} \otimes \widetilde{U} \otimes \tilde{x} = \tilde{b}$$

$$\Rightarrow \quad \tilde{L} \otimes \widetilde{x'} = \tilde{b}$$

$$\text{Where } \; \widetilde{U} \otimes \tilde{x} = \widetilde{x'} \tag{2.54}$$

By solving system (2.54), we obtain the solution x.

Example 2.17:

Consider the following FFSLE

$$\begin{bmatrix} (6,1,4) & (5,2,2) & (3,2,1) \\ (12,8,20) & (14,12,15) & (8,8,10) \\ (24,10,34) & (32,30,30) & (20,19,24) \end{bmatrix} \otimes \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} (58,30,60) \\ (142,139,257) \\ (316,297,514) \end{bmatrix}.$$

in matrix form

$$\widetilde{A} \otimes \tilde{x} = \tilde{b}$$

where

$$A = \begin{bmatrix} 6 & 5 & 3 \\ 12 & 14 & 8 \\ 24 & 32 & 20 \end{bmatrix}, \quad M = \begin{bmatrix} 1 & 2 & 2 \\ 8 & 12 & 8 \\ 10 & 30 & 19 \end{bmatrix}, \quad N = \begin{bmatrix} 4 & 2 & 1 \\ 20 & 15 & 10 \\ 34 & 30 & 24 \end{bmatrix}$$

From equations (2.42), (2.43), (2.44)and(2.45), we can calculate the elements of L_1 and U_1.

$$u_{11} = a_{11} = 6, \qquad u_{12} = a_{12} = 5, \qquad u_{13} = a_{13} = 3.$$

$$l_{11} = \frac{a_{11}}{u_{11}} = \frac{6}{6} = 1, \qquad l_{21} = \frac{a_{21}}{u_{11}} = \frac{12}{6} = 2, \qquad l_{31} = \frac{a_{31}}{u_{11}} = \frac{24}{6} = 4.$$

$$u_{22} = a_{22} - l_{21}u_{12} = 14 - 2 \times 5 = 4,$$

$$u_{23} = a_{23} - l_{21}u_{13} = 8 - 2 \times 3 = 2,$$

$$l_{22} = \frac{a_{22} - l_{21}u_{12}}{u_{22}} = \frac{14 - 2 \times 5}{4} = 1,$$

$$l_{32} = \frac{a_{32} - l_{31}u_{12}}{u_{22}} = \frac{32 - 4 \times 5}{4} = 3,$$

$$u_{33} = a_{33} - l_{31}u_{13} - l_{32}u_{23} = 20 - 4 \times 3 - 3 \times 2 = 2,$$

$$l_{33} = 1.$$

Thus we have,

$$L_1 = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 4 & 3 & 1 \end{bmatrix} \qquad U_1 = \begin{bmatrix} 6 & 5 & 3 \\ 0 & 4 & 2 \\ 0 & 0 & 2 \end{bmatrix}$$

To find the elements of L_2 and U_2we use equations (2.47), (2.48), (2.49) and (2.50).

$$u'_{11} = m_{11} = 1, \qquad u'_{12} = m_{12} = 2, \qquad u'_{13} = m_{13} = 2.$$

$$l'_{11} = \frac{m_{11} - l_{11}u'_{11}}{u_{11}} = \frac{0}{6} = 0, \qquad l'_{21} = \frac{m_{21} - l_{21}u'_{11}}{u_{11}} = \frac{6}{6} = 1,$$

$$l'_{31} = \frac{m_{31} - l_{31}u'_{11}}{u_{11}} = \frac{6}{6} = 1.$$

$$u'_{22} = m_{22} - l_{21}u'_{12} - l'_{21}u_{12} = 12 - 4 - 5 = 3,$$

$$u'_{23} = m_{23} - l_{21}u'_{13} - l'_{21}u_{13} = 8 - 4 - 3 = 1,$$

$$u'_{33} = m_{33} - l_{31}u'_{13} - l'_{31}u_{13} - l_{32}u'_{23} - l'_{32}u_{23} = 19 - 8 - 3 - 3 - 4 = 1 = 1,$$

$$l'_{32} = \frac{m_{32} - l'_{31}u_{12} - l_{31}u'_{12} - l_{32}u'_{22}}{u_{22}} = \frac{30 - 5 - 8 - 9}{4} = 2,$$

Thus we have,

$$L_2 = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 2 & 0 \end{bmatrix} \qquad U_2 = \begin{bmatrix} 1 & 2 & 2 \\ 0 & 3 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

In similar way, we can use equations (2.47), (2.48), (2.49) and (2.50). To find the elements of L_{32} and U_3 , we obtain

$$L_3 = \begin{bmatrix} 0 & 0 & 0 \\ 2 & 0 & 0 \\ 3 & 1 & 0 \end{bmatrix} \qquad U_3 = \begin{bmatrix} 4 & 2 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 3 \end{bmatrix}$$

Therefore the LU decomposition method of fuzzy matrix $\tilde{A}$ is

$$\tilde{A} = \tilde{L} \otimes \tilde{U} = \begin{bmatrix} (1,0,0) & (0,0,0) & (0,0,0) \\ (2,1,2) & (1,0,0) & (0,0,0) \\ (4,1,3) & (3,2,1) & (1,0,0) \end{bmatrix} \otimes \begin{bmatrix} (6,1,4) & (5,2,2) & (3,2,1) \\ (0,0,0) & (4,3,1) & (2,1,2) \\ (0,0,0) & (0,0,0) & (2,1,3) \end{bmatrix}$$

To solve the fully fuzzy linear system we will start by solving the system of equations $\tilde{L} \otimes \widetilde{x'} = \tilde{b}$

i.e.

$$\begin{bmatrix}(1,0,0) & (0,0,0) & (0,0,0)\\(2,1,2) & (1,0,0) & (0,0,0)\\(4,1,3) & (3,2,1) & (1,0,0)\end{bmatrix} \otimes \begin{bmatrix}x'\\y'\\z'\end{bmatrix} = \begin{bmatrix}(58,30,60)\\(142,139,257)\\(316,297,514)\end{bmatrix}$$

Using Cramer's rule we can easily compute $\tilde{x}'$ as

$$\tilde{x}' = \begin{bmatrix}x'\\y'\\z'\end{bmatrix} = \begin{bmatrix}(58,30,60)\\(26,21,21)\\(6,4,11)\end{bmatrix}$$

Finally we solve the fully fuzzy linear system of equations $\widetilde{U} \otimes \tilde{x} = \widetilde{x'}$

i.e.

$$\begin{bmatrix}(6,1,4) & (5,2,2) & (3,2,1)\\(0,0,0) & (4,3,1) & (2,1,2)\\(0,0,0) & (0,0,0) & (2,1,3)\end{bmatrix}\begin{bmatrix}x\\y\\z\end{bmatrix} = \begin{bmatrix}(58,30,60)\\(26,21,21)\\(6,4,11)\end{bmatrix}$$

Again we obtain by Cramer's rule

$$\tilde{x} = \begin{bmatrix}x\\y\\z\end{bmatrix} = \begin{bmatrix}(4,1,3)\\(5,0.5,2)\\(3,0.5,1)\end{bmatrix}$$

Chapter Three

Numerical Technique for Solving Linear Fuzzy Systems

In this chapter we will present some numerical technique for solving FLS and FFLS.

3.1 Numerical Methods for Solving Fuzzy System of Linear Equations(FLS)

In this section we will apply the following iterative schemes for solving (FLS).

3.1.1 Iterative Jacobi and Gauss−Sidel methods

An iterative technique for solving an $n \times n$ linear system $AX = b$ starts with an initial approximation X^0 to the solution X and then generates a sequence $\{X^{(k)}\}_{k=0}^{\infty}$, which converges to X. Most iterative technique involve a process of converting the system $AX = b$ into an equivalent system $X = TX + C$, where T is an $n \times n$ matrix and C is a column vector. After selecting an initial approximation X^0 we generate a sequence of vectors $\{X^{(k)}\}_{k=0}^{\infty}$ defined by

$X^{(k)} = TX^{(k-1)} + C \quad k \geq 1.$

Definition 3.1[5]: Diagonally Dominant Matrix: A square matrix A is called diagonally dominant if $|a_{ij}| \geq \sum_{i=1, i\neq j}^{n} |a_{ij}|$, $j = 1,2, \dots, n$. A is called strictly diagonally dominant if $|a_{ij}| > \sum_{i=1, i\neq j}^{n} |a_{ij}|$, $j = 1,2, \dots, n$.

At the beginning we are going to presented the following theorems.

Theorem 3.1 [4]:

Let matrix A in equation(1.3) be strictly diagonally dominant then both the Jacobi iterates and Gauss - Sidel iterates are converge to $A^{-1}Y$ for any X^0.

Theorem 3. 2 [4]:

The matrix A in equation(1.3) is strictly diagonally dominant if and only if matrix S be strictly diagonally dominant.

Proof: Let A be column strictly diagonally dominant matrix,

i.e. $|a_{ij}| > \sum_{i=1,i\neq j}^{n} |a_{ij}|, \quad j = 1,2,\dots,n.$

By considering the structure of S we have

$$\begin{aligned} s_{ij} = s_{n+i,n+j} = a_{ij} > 0 \Leftrightarrow s_{n+i,j} = s_{i,n+j} = a_{ij} = 0, \\ s_{ij} = s_{n+i,n+j} = a_{ij} = 0 \Leftrightarrow s_{n+i,j} = s_{i,n+j} = a_{ij} < 0, \end{aligned} \tag{3.1}$$

also

$$\sum_{i=1,i\neq j}^{2n} |s_{ij}| = \sum_{i=1,i\neq j}^{n} |s_{ij}| + \sum_{i=1,i\neq j}^{n} |s_{n+ij}|, \quad j = 1,\dots,2n.$$

If $s_{ij} > 0, i,j = 1,2,\dots,n$, then

$$\sum_{i=1,i\neq j}^{2n} |s_{ij}| = \begin{cases} \sum_{i=1,i\neq j}^{2n} |s_{ij}| \\ \sum_{i=1,i\neq j}^{2n} |s_{i,n+j}| \end{cases}$$

$$= \begin{cases} \sum_{i=1,i\neq j}^{n} |s_{ij}| + \sum_{i=1,i\neq j}^{n} |s_{n+ij}| \\ \sum_{i=1,i\neq j}^{n} |s_{i,n+j}| + \sum_{i=1,i\neq j}^{n} |s_{n+i,n+j}|, \quad j = 1,\dots,n. \end{cases}$$

From (3.1)

$$\sum_{i=1,i\neq j}^{2n} |s_{ij}| = \begin{cases} \sum_{i=1,i\neq j}^{n} |s_{ij}| < |a_{jj}| = |s_{jj}| \\ \sum_{i=1,i\neq j}^{n} |s_{ij}| < |a_{jj}| = |s_{n+i,n+j}|, \quad j = 1, \dots, n. \end{cases}$$

then

$$\sum_{i=1,i\neq j}^{2n} |s_{ij}| < |s_{jj}|, \quad j = 1,2, \dots, n.$$

Now suppose that S be column strictly diagonally dominant, we have

$$\sum_{i=1,i\neq j}^{2n} |s_{ij}| = \sum_{i=1,i\neq j}^{n} |s_{ij}| + \sum_{i=1,i\neq j}^{n} |s_{n+ij}|.$$

Taking into consideration (3.1) and $A = B - C$ we have

$$\sum_{i=1,i\neq j}^{2n} |s_{ij}| = \sum_{i=1,i\neq j}^{n} |a_{ij}| < |s_{jj}| = |a_{jj}|, \quad j = 1,2, \dots, n.$$

since $s_{ij} = 0, j = n, \dots, 2n$. It can be hold for row strictly diagonally dominant too. The proof is complete.

From [4], without loss of generality, suppose that $s_{ii} > 0$ for all $i = 1,2, \dots, 2n$. and let $S = D + L + U$ where

$$D = \begin{bmatrix} D_1 & 0 \\ 0 & D_1 \end{bmatrix}, \qquad L = \begin{bmatrix} L_1 & 0 \\ S_2 & L_1 \end{bmatrix}, \qquad U = \begin{bmatrix} U_1 & S_2 \\ 0 & U_1 \end{bmatrix}$$

$(D_1)_{ii} = s_{ii} > 0,\ I = 1,2, \dots, n$, and assume $S_1 = D_1 + L_1 + U_1$. In the Jacobi method, from the structure of $SX = Y$ we have

$$\begin{bmatrix} D_1 & 0 \\ 0 & D_1 \end{bmatrix} \begin{bmatrix} \underline{X} \\ \overline{X} \end{bmatrix} + \begin{bmatrix} L_1 + U_1 & S_2 \\ S_2 & L_1 + U_1 \end{bmatrix} \begin{bmatrix} \underline{X} \\ \overline{X} \end{bmatrix} = \begin{bmatrix} \underline{Y} \\ \overline{Y} \end{bmatrix}$$

then

$$\underline{X} = D_1^{-1}\underline{Y} - D_1^{-1}(L_1 + U_1)\underline{X} - D_1^{-1}S_2\overline{X},$$

$$\overline{X} = D_1^{-1}\overline{Y} - D_1^{-1}(L_1 + U_1)\overline{X} - D_1^{-1}S_2\underline{X}. \tag{3.2}$$

Thus the Jacobi iterative technique will be

$$\underline{X}^{k+1} = D_1^{-1}\underline{Y} - D_1^{-1}(L_1 + U_1)\underline{X}^k - D_1^{-1}S_2\overline{X}^k,$$

(3.3)

$$\overline{X}^{k+1} = D_1^{-1}\overline{Y} - D_1^{-1}(L_1 + U_1)\overline{X}^k - D_1^{-1}S_2\underline{X}^k, \quad k = 0,1, \dots$$

The elements of $X^{k+1} = \left(\underline{X}^{k+1}, \overline{X}^{k+1}\right)^t$ are

$$\underline{x}_i^{k+1}(r) = \frac{1}{s_{i,i}}\left[\underline{y}_i(r) - \sum_{j=1, j\neq i}^{n} s_{i,j}\underline{x}_j^k(r) - \sum_{j=1}^{n} s_{i,n+j}\overline{x}_j^k(r)\right],$$

$$\overline{x}_i^{k+1}(r) = \frac{1}{s_{i,i}}\left[\overline{y}_i(r) - \sum_{j=1, j\neq i}^{n} s_{i,j}\overline{x}_j^k(r) - \sum_{j=1}^{n} s_{i,n+j}\underline{x}_j^k(r)\right],$$

$$k = 0,1,2, \dots, \quad i = 1,2, \dots, n.$$

The result in the matrix form of the Jacobi iterative technique are $X^{k+1} = PX^k + C$ where

$$P = \begin{bmatrix} -D_1^{-1}(L_1 + U_1) & -D_1^{-1}S_2 \\ -D_1^{-1}S_2 & -D_1^{-1}(L_1 + U_1) \end{bmatrix}, \quad C = \begin{bmatrix} D_1^{-1}\underline{Y} \\ D_1^{-1}\overline{Y} \end{bmatrix}, \quad X = \begin{bmatrix} \underline{X} \\ \overline{X} \end{bmatrix}.$$

In the Gauss− Sidel method, we have:

$$\begin{bmatrix} D_1 + L_1 & 0 \\ S_2 & D_1 + L_1 \end{bmatrix}\begin{bmatrix} \underline{X} \\ \overline{X} \end{bmatrix} + \begin{bmatrix} U_1 & S_2 \\ 0 & U_1 \end{bmatrix}\begin{bmatrix} \underline{X} \\ \overline{X} \end{bmatrix} = \begin{bmatrix} \underline{Y} \\ \overline{Y} \end{bmatrix} \tag{3.4}$$

then

$$\underline{X} = (D_1 + L_1)^{-1}\underline{Y} - (D_1 + L_1)^{-1}U_1\underline{X} - (D_1 + L_1)^{-1}S_2\overline{X},$$

(3.5)

$$\overline{X} = (D_1 + L_1)^{-1}\overline{Y} - (D_1 + L_1)^{-1}U_1\overline{X} - (D_1 + L_1)^{-1}S_2\underline{X}.$$

Thus the Gauss− Sidel iterative technique will be

$$\underline{X}^{k+1} = (D_1 + L_1)^{-1}\underline{Y} - (D_1 + L_1)^{-1}U_1^{-1}\underline{X}^k - (D_1 + L_1)^{-1}S_2\overline{X}^k,$$

(3.6)

$$\overline{X}^{k+1} = (D_1 + L_1)^{-1}\overline{Y} - (D_1 + L_1)^{-1}U_1^{-1}\overline{X}^k - (D_1 + L_1)^{-1}S_2\underline{X}^k,$$

$$k = 0,1, \dots$$

So the elements of $X^{k+1} = \left(\underline{X}^{k+1}, \overline{X}^{k+1}\right)^t$ are

$$\underline{x}_i^{k+1}(r) = \frac{1}{s_{i,i}}\left[\underline{y}_i(r) - \sum_{j=1}^{i-1} s_{i,j}\underline{x}_j^{k+1}(r) - \sum_{j=i+1}^{n} s_{i,j}\underline{x}_j^{k}(r) - \sum_{j=1}^{n} s_{i,n+j}\overline{x}_j^{k}(r)\right],$$

$$\overline{x}_i^{k+1}(r) = \frac{1}{s_{i,i}}\left[\overline{y_i}(r) - \sum_{j=1}^{i-1} s_{i,j}\overline{x}_j^{k}(r) - \sum_{j=i+1}^{n} s_{i,j}\overline{x}_j^{k}(r) - \sum_{j=1}^{n} s_{i,n+j}\underline{x}_j^{k}(r)\right],$$

$$k = 0,1,2,\dots, \quad i == 1,2,\dots,n.$$

The result in the matrix form of the Gauss−Sidel iterative technique are

$X^{k+1} = PX^k + C$ as

$$P = \begin{bmatrix} -(D_1 + L_1)^{-1}U_1 & -(D_1 + L_1)^{-1}S_2 \\ -(D_1 + L_1)^{-1}S_2 & -(D_1 + L_1)^{-1}U_1 \end{bmatrix}, \quad C = \begin{bmatrix} (D_1 + L_1)^{-1}\underline{Y} \\ (D_1 + L_1)^{-1}\overline{Y} \end{bmatrix},$$

$$X = \begin{bmatrix} \underline{X} \\ \overline{X} \end{bmatrix}.$$

From Theorem 3.1and 3. 2, both Jacobi iterates and Gauss− Sidel iterates are converge to the unique solution $X = A^{-1}Y$, for any X^0, where $X \in R^{2n}$ and $(\underline{X}, \overline{X}) \in E^n$.The stopping criterion for a given tolerance $\epsilon > 0$ is

$$\frac{\left\|\overline{X}^{k+1} - \overline{X}^{k}\right\|}{\left\|\overline{X}^{k+1}\right\|} < \epsilon, \qquad \frac{\left\|\underline{X}^{k+1} - \underline{X}^{k}\right\|}{\left\|\underline{X}^{k+1}\right\|} < \epsilon, \quad k = 0,1,\dots$$

3.1.2 Successive over relaxation iterative method

In this section we turn next to a modification of the Gauss–Sidel iteration which known as SOR iterative method. By multiply system of (3.4) in D^{-1}:

$$\begin{bmatrix} I + D_1^{-1}L_1 & 0 \\ S_2 & I + D_1^{-1}L_1 \end{bmatrix}\begin{bmatrix} \underline{X} \\ \overline{X} \end{bmatrix} + \begin{bmatrix} D_1^{-1}U_1 & S_2 \\ 0 & D_1^{-1}U_1 \end{bmatrix}\begin{bmatrix} \underline{X} \\ \overline{X} \end{bmatrix} = \begin{bmatrix} D_1^{-1}\underline{Y} \\ D_1^{-1}\overline{Y} \end{bmatrix} \qquad (3.7)$$

Let $D_1^{-1}U_1 = U_1$, $D_1^{-1}L_1 = L_1$ then

$$\begin{bmatrix} I + L_1 & 0 \\ S_2 & I + L_1 \end{bmatrix}\begin{bmatrix} \underline{X} \\ \overline{X} \end{bmatrix} + \begin{bmatrix} U_1 & S_2 \\ 0 & U_1 \end{bmatrix}\begin{bmatrix} \underline{X} \\ \overline{X} \end{bmatrix} = \begin{bmatrix} D_1^{-1}\underline{Y} \\ D_1^{-1}\overline{Y} \end{bmatrix} \qquad (3.8)$$

hence

$$(I + L_1)\underline{X} = D^{-1}\underline{Y} - U_1\underline{X} - S_2\overline{X},$$

$$(I + L_1)\overline{X} = D^{-1}\overline{Y} - U_1\overline{X} - S_2\underline{X} \tag{3.9}$$

for some parameter ω:

$$(I + \omega L_1)\underline{X} = \omega D^{-1}\underline{Y} - [(1-\omega)I + \omega U_1]\underline{X} - \omega S_2\overline{X},$$

$$(I + \omega L_1)\overline{X} = \omega D^{-1}\overline{Y} - [(1-\omega)I + \omega U_1]\overline{X} - \omega S_2\underline{X}. \tag{3.10}$$

If $\omega = 1$, then clearly X is just the Gauss–Sidel solution (3.9). So the SOR iterative method will be:

$$\underline{X}^{k+1} = (I + \omega L_1)^{-1}\omega D^{-1}\underline{Y} - (I + \omega L_1)^{-1}[(1-\omega)I + \omega U_1]\underline{X}^{k} - (I + \omega L_1)^{-1}\omega S_2\overline{X}^{k},$$

$$\overline{X}^{k+1} = (I + \omega L_1)^{-1}\omega D^{-1}\overline{Y} - (I + \omega L_1)^{-1}[(1-\omega)I + \omega U_1]\overline{X}^{k} - (I + \omega L_1)^{-1}\omega S_2\underline{X}^{k}. \tag{3.11}$$

Consequently the result in the matrix form of the SOR iterative method are $X^{K+1} = PX^{K} + C$ where

$$P = \begin{bmatrix} -(I + \omega L_1)^{-1}[(1-\omega)I + \omega U_1] & -(I + \omega L_1)^{-1}\omega S_2 \\ -(I + \omega L_1)^{-1}\omega S_2 & -(I + \omega L_1)^{-1}[(1-\omega)I + \omega U_1] \end{bmatrix},$$

$$C = \begin{bmatrix} (I + \omega L_1)^{-1}\omega D^{-1} \\ (I + \omega L_1)^{-1}\omega D^{-1} \end{bmatrix}.$$

For $0 < \omega < 1$ this method is called successive under−relaxation method that can be used to achieve convergence for systems that are not convergent by the Gauss−sidel method.

For $\omega > 1$ the method is called successive over−relaxation method (SOR) that can be used to accelerate of convergence of linear systems that are already convergent by the Gauss−sidel method.

Theorem 3.3 [5]:

If S is appositive definite matrix and $0 < \omega < 2$ then the SOR method converges for any choice of initial approximate vector X^0.

3.2 Numerical Methods for Solving Fully Fuzzy Linear System of Equations (FFLS)

In the previous chapters, we have presented some direct methods for solving fully fuzzy linear system of equations. In this section, two iterative methods namely: Gauss-Jacobi, and Gauss-Seidel methods are presented to find the solution of fully fuzzy linear system of equations.

3.2.1 Gauss- Jacobi method

To solve fully fuzzy linear system of equations we already discussed an approach in chapter 2. According to which the positive vectors x, y and z can be found by solving following linear system of equations .

Now consider the FFLS $Ax = b$. by using equation (2.36)

$$\begin{cases} Ax = b \\ Ay + Mx = g \\ Az + Nx = h \end{cases}$$

We can write the previous equation as

$$\begin{cases} a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n = b_i \\ (a_{i1}y_1 + a_{i2}y_2 + \cdots + a_{in}y_n) + (m_{i1}x_1 + m_{i2}x_2 + \cdots + m_{in}x_n) = g_i, 1 \le i \le n \\ (a_{i1}z_1 + a_{i2}z_2 + \cdots + a_{in}z_n) + (n_{i1}x_1 + n_{i2}x_2 + \cdots + n_{in}\, x_n) = h_i \end{cases} \quad (3.12)$$

Using above equations, we can say

$$\begin{cases} a_{ii}x_i = b_i - \sum_{j=1, j\neq i}^{n} a_{ij}x_j \\ a_{ii}y_i = g_i - \left(\sum_{j=1, j\neq i}^{n} a_{ij}y_j + \sum_{j=1}^{n} m_{ij}x_j \right), \quad 1 \le i \le n, a_{ii} \neq 0 \\ a_{ii}z_i = h_i - \left(\sum_{j=1, j\neq i}^{n} a_{ij}z_j + \sum_{j=1}^{n} n_{ij}x_j \right) \end{cases} \quad (3.13)$$

Hence

$$\begin{cases} x_i = \frac{1}{a_{ii}}(b_i - \sum_{j=1, j\neq i}^{n} a_{ij}x_j) \\ y_i = \frac{1}{a_{ii}}(g_i - \left(\sum_{j=1, j\neq i}^{n} a_{ij}y_j + \sum_{j=1}^{n} m_{ij}x_j \right)), \quad 1 \le i \le n \\ z_i = \frac{1}{a_{ii}}(h_i - \left(\sum_{j=1, j\neq i}^{n} a_{ij}z_j + \sum_{j=1}^{n} n_{ij}x_j \right)) \end{cases} \quad (3.14)$$

This can easily be written as

$$\begin{cases} x_i = -\frac{1}{a_{ii}} \sum_{j=1, j\neq i}^{n} a_{ij}x_j + \frac{b_i}{a_{ii}} \\ y_i = -\frac{1}{a_{ii}} \left(\sum_{j=1, j\neq i}^{n} a_{ij}y_j + \sum_{j=1}^{n} m_{ij}x_j \right) + \frac{g_i}{a_{ii}}, \quad 1 \le i \le n \\ z_i = -\frac{1}{a_{ii}} \left(\sum_{j=1, j\neq i}^{n} a_{ij}z_j + \sum_{j=1}^{n} n_{ij}x_j \right) + \frac{h_i}{a_{ii_i}} \end{cases} \quad (3.15)$$

Equations (3.15) can be written in matrix form as the following:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = M \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \beta \tag{3.16}$$

where M is called the iteration matrix and β is a vector.

To solve system(3.16), we can consider initial approximation $X^{(0)}$ of the solution vector and then we substitute it into the right hand side of equation (3.16). The solution of equation (3.16) will give a vector $X^{(1)}$, which is better approximation to the solution than $X^{(0)}$. We continue this process until the successive iteration $X^{(k)}$ converges to the solution up to desired accuracy, which suggests the following iterative process as the Gauss-Jacobi method for solving a fully fuzzy linear system of equations:

$$\begin{cases} x_i{}^{(k+1)} = \dfrac{1}{a_{ii}}(b_i - \displaystyle\sum_{j=1, j\neq i}^{n} a_{ij} x_j{}^{k}) & (3.17) \\ y_i{}^{(k+1)} = \dfrac{1}{a_{ii}}(g_i - \left(\displaystyle\sum_{j=1, j\neq i}^{n} a_{ij} y_j{}^{(k)} + \sum_{j=1}^{n} m_{ij} x_j{}^{(k)}\right)), 1 \leq i \leq n & (3.18) \\ z_i{}^{(k+1)} = \dfrac{1}{a_{ii}}(h_i - \left(\displaystyle\sum_{j=1, j\neq i}^{n} a_{ij} z_j{}^{(k)} + \sum_{j=1}^{n} n_{ij} x_j{}^{(k)}\right)) & (3.19) \end{cases}$$

In general,

$$\begin{bmatrix} x^{(k+1)} \\ y^{(k+1)} \\ z^{(k+1)} \end{bmatrix} = M \begin{bmatrix} x^{(k)} \\ y^{(k)} \\ z^{(k)} \end{bmatrix} + \beta, \quad (k \geq 0) \tag{3.20}$$

Where M is called the iteration matrix of the iterative method and β is a vector. $x^{(k+1)}$ and $x^{(k)}$ denote solution at k^{th} and $(k+1)^{th}$ iteration respectively.

Equation (3.20) can be written in the matrix form as:

$$\begin{bmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \\ \dots \\ \dots \\ x_n^{(k+1)} \end{bmatrix} = -\begin{bmatrix} \frac{1}{a_{11}} & & & & \\ & \frac{1}{a_{22}} & & & \\ & & \frac{1}{a_{33}} & & \\ & & & \dots & \\ & & & & \frac{1}{a_{nn}} \end{bmatrix} \left\{ \begin{bmatrix} 0 & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & 0 & a_{32} & \cdots & a_{2n} \\ a_{31} & a_{32} & 0 & \cdots & a_{3n} \\ \dots & \dots & \cdots & 0 & \cdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & 0 \end{bmatrix} \begin{bmatrix} x_1^{(k)} \\ x_2^{(k)} \\ \dots \\ \dots \\ x_n^{(k)} \end{bmatrix} - \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \cdots \\ b_n \end{bmatrix} \right\}$$

or

$$x^{(k+1)} = -D_A^{-1}(L_A + U_A)x^{(k)} + D_A^{-1}b$$

Similarly, equation (3.18) and (3.19) can be written in matrix form respectively as

$$y^{(k+1)} = -D_A^{-1}(L_A + U_A)y^{(k)} + D_A^{-1}g$$

$$z^{(k+1)} = -D_A^{-1}(L_A + U_A)z^{(k)} + D_A^{-1}h$$

Sufficient condition [14]:

The Gauss – Jacobi iterative method for solving fully fuzzy linear system of equations $\widetilde{A} \otimes \tilde{x} = \tilde{b}$ converges if and only if the classical Gauss- Jacobi iterative method converges for solving the crisp linear system of equations $Ax = b$ derived from the corresponding fully fuzzy linear system of equations.

If the matrix A in the crisp linear system of equations $Ax = b$ is strictly diagonally dominant i.e., $|a_{ii}| > \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|, i = 1,2,3, \dots, n$ then the iterations obtained in classical Gauss- Jacobi iterative method converges for any initial approximation $X^{(0)}$.

3.2.2 Gauss- Seidel method

Another well-known iterative method for solving FFLS is the Gauss–Seidel method.

Equation (3.12) can be written as:

$$\begin{cases} \sum_{j\le i} a_{ij}x_j = b_i - \sum_{j>i}^{n} a_{ij}x_j, \\ \sum_{j\le i} a_{ij}y_i = g_i - \left(\sum_{j>i}^{n} a_{ij}y_j + \sum_{j=1}^{n} m_{ij}x_j\right), \qquad 1 \le i \le n, \\ \sum_{j\le i} a_{ij}z_i = h_i - \left(\sum_{j>i}^{n} a_{ij}z_j + \sum_{j=1}^{n} n_{ij}x_j\right). \end{cases}$$

Thus, Gauss- Seidel method is defined as:

$$\begin{cases} x_i^{(k+1)} = \frac{1}{a_{ii}}\left[b_i - \sum_{j<i}^{n} a_{ij}x_j^{(k+1)} - \sum_{j>i}^{n} a_{ij}x_j^{(k)}\right], \\ y_i^{(k+1)} = \frac{1}{a_{ii}}\left[g_i - \sum_{j<i}^{n} a_{ij}y_j^{(k+1)} - \sum_{j>i}^{n} a_{ij}y_j^{(k)} - \sum_{j=1}^{n} m_{ij}x_j^{(k)}\right], 1 \le i \le n, k \ge 0 \\ z_i^{(k+1)} = \frac{1}{a_{ii}}\left[h_i - \sum_{j<i}^{n} a_{ij}z_j^{(k+1)} - \sum_{j>i}^{n} a_{ij}z_j^{(k)} - \sum_{j=1}^{n} n_{ij}x_j^{(k)}\right] \end{cases} \qquad (3.21)$$

or, in matrix form the system can be written as

$$\begin{cases} (D_A + L_A)x^{(k+1)} = b - U_A x^{(k)} \\ (D_A + L_A)y^{(k+1)} = g - U_A y^{(k)} - Mx^{(k)} \\ (D_A + L_A)z^{(k+1)} = h - U_A z^{(k)} - Mx^{(k)} \end{cases}$$

where D_A, L_A, U_A are diagonal, lower triangular and upper triangular matrices respectively.

Therefore the Gauss-Seidel iterative method for solving fully fuzzy linear system of equations is as follows:

$$\begin{cases} x^{(k+1)} = -(D_A + L_A)^{-1}U_A x^{(k)} + (D_A + L_A)^{-1}b \\ y^{(k+1)} = -(D_A + L_A)^{-1}U_A y^{(k)} + (D_A + L_A)^{-1}Mx^{(k)} + (D_A + L_A)^{-1}\,g \\ z^{(k+1)} = -(D_A + L_A)^{-1}U_A z^{(k)} + (D_A + L_A)^{-1}Nx^{(k)} + (D_A + L_A)^{-1}h \end{cases}$$

In this method also, if A is strictly diagonally dominant then the iteration always converges. Gauss-Seidel method will generally converge if the Jacobi method converges and will converge at a faster speed.

Chapter Four

Numerical Examples and Results

4.1 Numerical Examples and Results for Fuzzy System of Linear Equations(FLS)

To demonstrate the efficiency and accuracy of the numerical schemes which we discuss it in chapter three, we will use MATLAB software to solve some numerical examples, then draw a comparison between approximate solution and exact ones for the following schemes: Jacobi method, Gauss−Sidel method, and Successive over relaxation iterative method.

Example 4.1.

Consider the 2×2 non- symmetric fuzzy linear system

$$\begin{aligned} 2x_1 - 2x_2 &= (2r, 4 - 2r) \\ 2x_1 + 6x_2 &= (8 + 2r, 14 - 4r) \end{aligned} \tag{4.1}$$

Numerical Solution of Equation (4.1) using Jacobi Method

The extended 4×4 matrix is

$$S = \begin{pmatrix} 2 & 0 & 0 & -2 \\ 2 & 6 & 0 & 0 \\ 0 & -2 & 2 & 0 \\ 0 & 0 & 2 & 6 \end{pmatrix}$$

$$X=\begin{pmatrix}\underline{x}_1(r)\\ \underline{x}_2(r)\\ -\overline{x}_1(r)\\ -\overline{x}_2(r)\end{pmatrix}=S^{-1}Y=\begin{pmatrix}0.5625 & -0.0625 & 0.1875 & -0.1875\\ -0.1875 & 0.1875 & -0.0625 & 0.0625\\ 0.1875 & -0.1875 & 0.5625 & -0.0625\\ -0.0625 & 0.0625 & -0.1875 & 0.1875\end{pmatrix}\begin{pmatrix}2r\\ 8+2r\\ 2r-4\\ 4r-14\end{pmatrix}$$

The exact solution is

$$x_1=\left(\underline{x_1}(r),\overline{x_1}(r)\right)=\left(\frac{11}{8}+\frac{5}{8}r,\frac{23}{8}-\frac{7}{8}r\right),$$
$$x_2=\left(\underline{x_2}(r),\overline{x_2}(r)\right)=\left(\frac{7}{8}+\frac{1}{8}r,\frac{11}{8}-\frac{3}{8}r\right).$$

The exact and approximate solutions are shown in Figure (4.1).

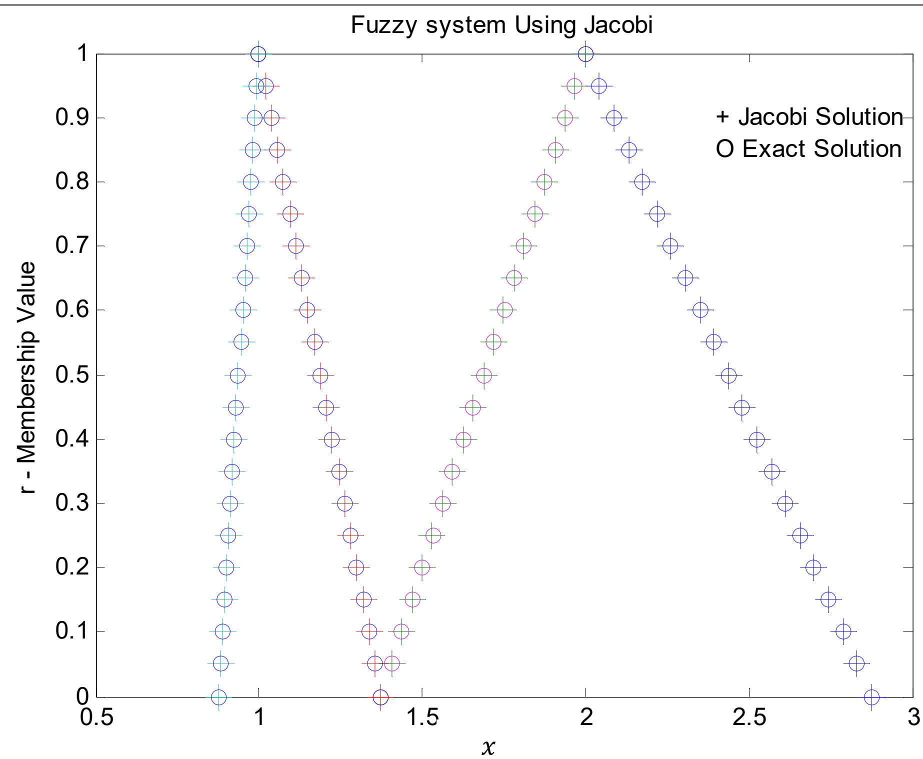
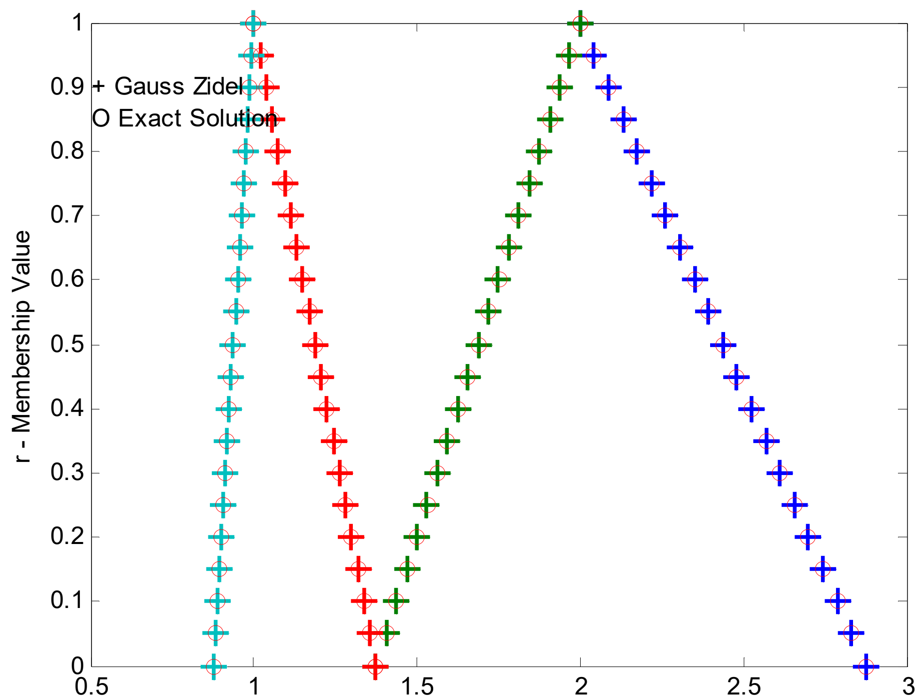


Figure (4.1): Graphical representation of the solution of example 4.1 using Jacobi method

Numerical Solution of Equation(4.1) using Gauss- Seidel Method

The exact and approximated solutions are plotted and compared in Figure(4.2).



Figure(4.2): Graphical representation of the solution of example 4.1 using Gauss-Seidel method

Numerical Solution of Example (4.1) using Successive over relaxation iterative method

The exact and approximated solutions are plotted and compared in Figure(4.3)

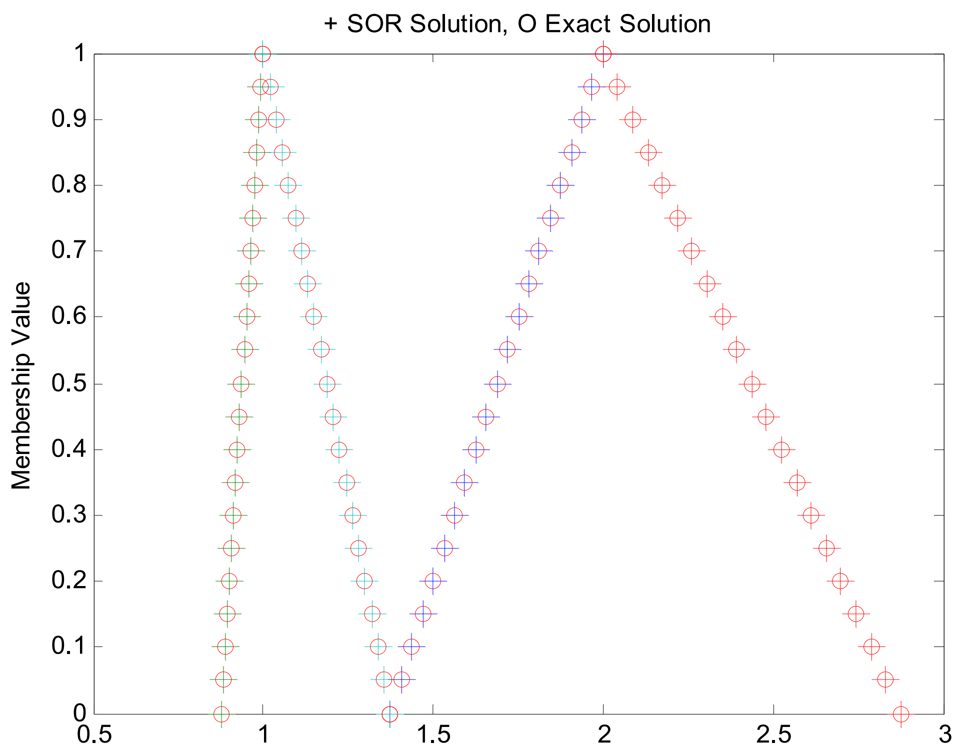


Figure (4.3): Graphical representation of the solution of example 4.1 using SOR method

Comparison results between three methods

Numerical Method	Number of Iterations	Total CPU Time in Seconds
Jacobi	16	16.8
Gauss- Seidel	9	7.9
Successive over relaxation	6	12.5

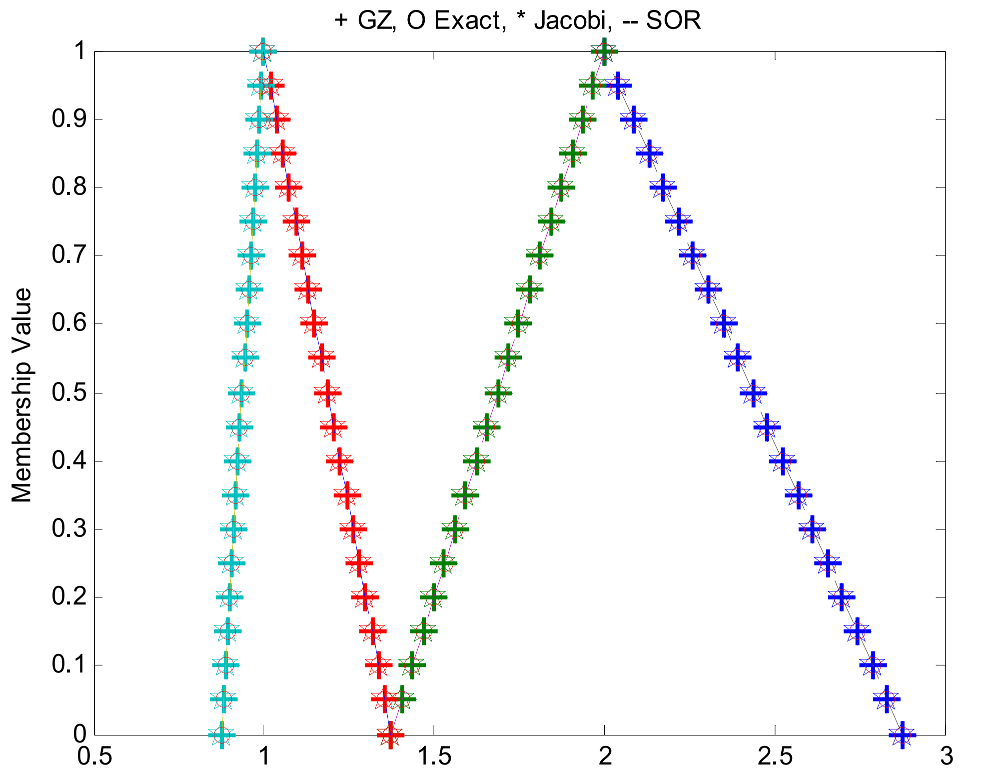


Figure (4.4): comparison between three methods.

4.2 Numerical Examples and Results for Fully Fuzzy System of Linear Equations(FFLS)

Numerical Solution by Gauss- Jacobi iterative method

Example 4.2:

Solve the following system of equations using Gauss- Jacobi method

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

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

So, from the above system we have

$A = \begin{bmatrix} 5 & 6 \\ 7 & 4 \end{bmatrix}$ $\quad M = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$ $\quad N = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}$ and

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

To solve the above problem by using Gauss-Jacobi method, first of all we obtain the following equations by the method explained in chapter 3.

$$\begin{bmatrix} 5 & 6 \\ 7 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 50 \\ 48 \end{bmatrix} \tag{4.2}$$

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 5 & 6 \\ 7 & 4 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 10 \\ 5 \end{bmatrix} \tag{4.3}$$

$$\begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 5 & 6 \\ 7 & 4 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 17 \\ 7 \end{bmatrix} \tag{4.4}$$

Equation(4.2) can be written as:

$5x_1 + 6x_2 = 50$

$7x_1 + 4x_2 = 48$

Since$|5| \ngtr |6|$ and $|4| \ngtr |7|$, therefore the above system of equations is not diagonally dominant. So writing the above system in diagonally dominant form as:

$$7x_1 + 4x_2 = 48$$

(4.5)

$$5x_1 + 6x_2 = 50$$

So, the above system of linear equations is in diagonally dominant form as $|7| > |4|$ and $|6| > |5|$. Now, to find the solution by Gauss-Jacobi method first of all (4.5) can also be written as

$$x_1 = \frac{1}{7}(48 - 4x_2)$$

(4.6)

$$x_2 = \frac{1}{6}(50 - 5x_1)$$

Thus, the Gauss-Jacobi's methods when applied to the above system, it gives

$$x_1{}^{(k+1)} = \frac{1}{7}\left(48 - 4x_2{}^{(k)}\right)$$

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

$$x_2{}^{(k+1)} = \frac{1}{6}\left(50 - 5x_1{}^{(k)}\right)$$

Now, starting with initial approximation vector $x^{(0)} = (0,0)$, we get

$$x_1{}^{(1)} = \frac{1}{7}\left(48 - 4x_2{}^{(0)}\right) = \frac{48}{7} = 6.8571$$

$$x_2^{(1)} = \frac{1}{6}\left(50 - 5x_1{}^{(0)}\right) = \frac{50}{6} = 8.3333$$

i.e. $x^{(1)} = (6.8571,\ 8.3333)$

hence continuing with this we obtain

Table (4.1): the sequence $x^{(k)}, k = 0, 1, 2, \ldots$ generated by the Jacobi method, with $\varepsilon = 1 \times 10^{-2}$.

k	$x = (x_1, x_2)$	Error $\|x - x_0\|$
1	$x^{(1)} = (6.8571, 8.3333)$	10.7919
2	$x^{(2)} = (2.0952, 2.6190)$	7.4383
3	$x^{(3)} = (5.3605, 6.5873)$	5.1390
4	$x^{(4)} = (3.0930, 3.8662)$	3.5421
5	$x^{(5)} = (4.6479, 5.7559)$	2.4471
6	$x^{(6)} = (3.5681, 4.4601)$	1.6867
7	$x^{(7)} = (4.3085, 5.3599)$	1.1653
8	$x^{(8)} = (3.7943, 4.7429)$	0.8032
9	$x^{(9)} = (4.1469, 5.1714)$	0.5549
10	$x^{(10)} = (3.9021, 4.8776)$	0.3825
11	$x^{(11)} = (4.0700, 5.0816)$	0.2642
12	$x^{(12)} = (3.9534, 4.9417)$	0.1821
13	$x^{(13)} = (4.0333, 5.0389)$	0.1258
14	$x^{(14)} = (3.9778, 4.9722)$	0.0867
15	$x^{(15)} = (4.0159, 5.0185)$	0.0599
16	$x^{(16)} = (3.9894, 4.9868)$	0.0413
17	$x^{(17)} = (4.0076, 5.0088)$	0.0285
18	$x^{(18)} = (3.9950, 4.9937)$	0.0197
19	$x^{(19)} = (4.0036, 5.0042)$	0.0136
20	$x^{(20)} = (3.9976, 4.9970)$	0.0094

Since we have already found the exact solution of the above system in chapter two, Example 2.15 and is found to be $(4, 5)$. It seems that the sequence $x^{(k)}, k = 0,1,2, \ldots$ generated by the Jacobi method will converge to the exact solution. Hence up to two decimal places we obtain

$$x = (x_1, x_2) = (4.00, 5.00)$$

Now, putting the value of (x_1, x_2) in the equations (4.3) and (4.4) we obtain

$$\begin{bmatrix} 5 & 6 \\ 7 & 4 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 5 & 6 \\ 7 & 4 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

i.e.
$$\begin{aligned} 5y_1 + 6y_2 &= 1 \\ 7y_1 + 4y_2 &= 1 \end{aligned}$$
and
$$\begin{aligned} 5z_1 + 6z_2 &= 3 \\ 7z_1 + 4z_2 &= 2 \end{aligned}$$

Since the above equation are not in the form of diagonally dominant form. So converting them to diagonally dominant form as:

$$\begin{aligned} 7y_1 + 4y_2 &= 1 \\ 5y_1 + 6y_2 &= 1 \end{aligned} \tag{4.7}$$

Now, solving the above equations by the same procedure that is used to solve the system (4.2), we obtain:

$$y_1 = \frac{1}{7}(1 - 4y_2)$$

$$y_2 = \frac{1}{6}(1 - 5y_1)$$

Taking the initial approximation as $y^{(0)} = (0,0)$ and continuing with Gauss- Jacobi method we obtain

Table (4.2): the sequence $y^{(k)}, k = 0, 1, 2, \ldots$ generated by the Jacobi method, with $\varepsilon = 1 \times 10^{-2}$.

k	$y = (y_1, y_2)$	Error $\lvert y - y_0 \rvert$
1	$y^{(1)} = (0.143200, 0.167570)$	0.2204
2	$y^{(2)} = (0.047448, 0.048233)$	0.1530
3	$y^{(3)} = (0.115640, 0.128030)$	0.1050
4	$y^{(4)} = (0.070042, 0.071201)$	0.0729
5	$y^{(5)} = (0.102510, 0.109200)$	0.0500
6	$y^{(6)} = (0.080801, 0.082138)$	0.0347
7	$y^{(7)} = (0.096264, 0.100230)$	0.0238
8	$y^{(8)} = (0.085924, 0.087346)$	0.0165
9	$y^{(9)} = (0.093288, 0.095962)$	0.0113
10	$y^{(10)} = (0.088364, 0.089826)$	0.0079

at 10^{th} iteration we obtain $y^{(10)} = (0.0862,\ 0.0862)$ which is very close to exact solution $\left(\frac{1}{11}, \frac{1}{11}\right)$. Hence the value of the optimal solution up to two decimal place is:

$$y = (y_1, y_2) = (0.09, 0.09)$$

Similarly solving

$$5z_1 + 6z_2 = 3$$

$$(4.8)$$

$$7z_1 + 4z_2 = 2$$

Solving (4.8) we find that the value of z converges at 12^{th} iteration as follows:

Table (4.3): the sequence $z^{(k)}, k = 0, 1, 2, \dots$ generated by the Jacobi method, with $\varepsilon = 1 \times 10^{-2}$.

k	$z = (z_1, z_2)$	Error $\lvert z - z_0 \rvert$
1	$z^{(1)} = (0.28614,\ 0.5014)$	0.5773
2	$z^{(2)} = (-0.00037113, 0.26295)$	0.3728
3	$z^{(3)} = (0.13589, 0.50171)$	0.2749
4	$z^{(4)} = (-0.00054786, 0.38816)$	0.1775
5	$z^{(5)} = (0.064337,\ 0.50186)$	0.1309
6	$z^{(6)} = (-0.00063202, 0.44778)$	0.0845
7	$z^{(7)} = (0.030266, 0.50193)$	0.0623
8	$z^{(8)} = (-0.0006721, 0.47618)$	0.0403
9	$z^{(9)} = (0.014041, 0.50196)$	0.0297
10	$z^{(10)} = (-0.00069118, 0.4897)$	0.0192
11	$z^{(11)} = (0.0063151, 0.50197)$	0.0141
12	$z^{(12)} = (-0.00070027, 0.49614)$	0.0091

Thus the value of z up to two decimal points is

$$z = (z_1, z_1) = (0, 0.5)$$

Hence the solution of given fully fuzzy linear system of equations is as follows:

$$\breve{x} = \begin{bmatrix} (4, 0.09, 0) \\ (5, 0.09, 0.5) \end{bmatrix}$$

Which is the required solution of the given fully fuzzy linear system of equations.

Numerical Solution by Gauss- Seidel iterative method

Consider the system (4.2).

i.e.

$$x_1 = \frac{1}{7}(48 - 4x_2)$$

$$x_2 = \frac{1}{6}(50 - 5x_1)$$

The Gauss-Seidel iterative formula for this system can be written as:

$$x_1^{(k+1)} = \frac{1}{7}\left(48 - 4x_2^{(k)}\right)$$

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

$$x_2^{(k+1)} = \frac{1}{6}\left(50 - 5x_1^{(k+1)}\right)$$

Taking the $x^{(0)} = \left(x_1^{(0)}, x_2^{(0)}\right) = (0,0)$ we get

$$x_1^{(1)} = \frac{1}{7}\left(48 - 4x_2^{(0)}\right) = \frac{48}{7} = 6.8571$$

$$x_2^{(1)} = \frac{1}{6}\left(50 - 5x_1^{(1)}\right) = \frac{15.7145}{6} = 2.6191$$

i.e. $\quad x^{(1)} = (6.8571,\ 2.6191)$

hence continuing with this, we get

Table (4.4): the sequence $x^{(k)}, k = 0, 1, 2, \ldots$ generated by the Gauss-Seidel method, with $\varepsilon = 1 \times 10^{-2}$.

k	$x = (x_1, x_2)$	Error $\lvert x - x_0 \rvert$
1	$x^{(1)} = (6.8571, 2.6190)$	7.3403
2	$x^{(2)} = (5.3605, 3.8662)$	1.9481
3	$x^{(3)} = (4.6479, 4.4601)$	0.9277
4	$x^{(4)} = (4.3085, 4.7429)$	0.4418
5	$x^{(5)} = (4.1469, 4.8776)$	0.2104
6	$x^{(6)} = (4.0700, 4.9417)$	0.1002
7	$x^{(7)} = (4.0333, 4.9722)$	0.0477
8	$x^{(8)} = (3.7943, 4.7429)$	0.0227
9	$x^{(9)} = (4.0159, 4.9868)$	0.0108
10	$x^{(10)} = (4.0036, 4.9970)$	0.0052

Since we can found the exact solution of the above system in chapter two, And is found to be (4,5). It seems that the sequence $x^{(k)}, k = 0,1,2, \ldots$ generated by the Gauss-Seidel method will converges to the exact solution.

So, by the above results it is clear that the value of x up to two decimal points is $x = (x_1, x_2) = (4,5)$ solving (4.7) we obtain

Table (4.5): the sequence $y^{(k)}, k = 0, 1, 2, \dots$ generated by the Gauss-Seidel method, with $\varepsilon = 1 \times 10^{-2}$.

k	$y = (y_1, y_2)$	Error $\lvert y - y_0 \rvert$
1	$y^{(1)} = (0.142340, 0.047947)$	0.1502
2	$y^{(2)} = (0.114940, 0.070779)$	0.0357
3	$y^{(3)} = (0.101900, 0.081652)$	0.1070
4	$y^{(4)} = (0.095685, 0.086829)$	0.0081

Hence the value of y up to two decimal places can be written as

$$y = (y_1, y_2) = (0.09, 0.09)$$

and solving (4.8), using the same method as used for solving the system (4.5) we obtain

Table (4.6): the sequence $z^{(k)}, k = 0, 1, 2, \dots$ generated by the Gauss-Seidel method, with $\varepsilon = 1 \times 10^{-2}$.

k	$z = (z_1, z_2)$	Error $\lvert z - z_0 \rvert$
1	$z^{(1)} = (0.28614, 0.2619500)$	0.53879
2	$z^{(2)} = (0.13646, 0.3866800)$	0.1948
3	$z^{(3)} = (0.06518, 0.4460800)$	0.0928
4	$z^{(4)} = (0.031238, 0.474370)$	0.0442
5	$z^{(5)} = (0.015075, 0.487840)$	0.0210
6	$z^{(6)} = (0.0073784, 0.49425)$	0.0100
7	$z^{(7)} = (0.0037134, 0.49731)$	0.0048

Hence from the above results, we find that the value of z up to two decimal points is found to be:

$$z = (z_1, z_2) = (0, 0.5)$$

Hence the solution of the given fully fuzzy linear system of equation up to two decimal places is found to be

$$\check{x} = \begin{bmatrix} (4,0.09,0) \\ (5,0.09,0.5) \end{bmatrix}$$

Comparison results between Gauss-Jacobi and Gauss- Seidel Method

Numerical Methods	Total CPU Time in seconds	Error $\lvert \breve{x} - x^0 \rvert$
Gauss-Jacobi	4.2	0.009127082636873
Gauss- Seidel	5.9	0.004770825630768

Conclusions

In this thesis, analytical and numerical methods have been used to solve Fuzzy System of Linear Equations where the coefficient matrix arrays are crisp numbers, the right-hand side column is an arbitrary fuzzy vector and the unknowns are fuzzy numbers. Moreover, some analytical and numerical methods have been used to solve Fully Fuzzy System of Linear Equations where all the coefficient matrix arrays, the right-hand side arrays and the unknowns, are fuzzy numbers.

The numerical methods for FLS and FFLS were implemented in a form of algorithms to solve some numerical test cases using MATLAB software.

For FLS the numerical results have shown to be in a close agreement with the analytical ones. Moreover, the SOR iterative method is one of the most powerful numerical technique for solving FLS, in terms of number of iterations and CPU time, as we show in Example (4.1).

For FFLS the numerical results have shown to be in a close agreement with the analytical ones. In fact, the Gauss- Seidel iterative methods is more efficient than the Gauss-Jacobi for solving FFLS in terms of number of iterations, CPU time and the absolute error, as we shown in Example (4.2).

References

[1] S. Abbasbandy and M. Alavi, ***A method for solving fuzzy linear systems,*** **Iranian journal of Fuzzy systems**, 2 (2005), 37-43.

[2] S. Abbasbandy, R. Ezzati, and A. Jafarian, ***LU decomposition method for solving fuzzy system of linear equations,*** **Applied Mathematics and Computation,** vol. 172, no. 1, (2006), pp. 633–643.

[3] S. Abbasbandy and A. Jafarian, ***Steepest descent method for system of fuzzy linear equations,*** **Applied Mathematics and Computation**, Vol. 175,(2006), pp. 823–833.

[4] T. Allahviraloo, ***Numerical methods for fuzzy system of linear equations***, **Applied Mathematics and Computation** 155 (2004) 493-502.

[5] T. Allahviraloo, ***Successive over relaxation iterative method for fuzzy system of linear equations***, **Applied Mathematics and Computation** 62 (2005),pp 189-196.

[6] T. Allahviranloo and A. Hashemi, **The embedding method to obtain the solution of fuzzy linear systems**, Vol. 6, No. 3,(2014) Article ID IJIM-00469, 229-233.

[7] Amrahov, S. Mrah and I. Askerzade, ***Strong solutions of the fuzzy linear systems,*** **CMES-Computer Modeling in Engineering & Sciences,** vol. 76(4), (2011), pp. 207-216.

[8] B. Asady, S. Abbasbandy and M. Alavi, ***Fuzzy general linear systems,*** **Applied Mathematics and Computation** 169 (2005) 34–40.

[9] B. Bede, ***Mathematics of fuzzy sets and fuzzy logic***, **STUDFUZZ 295, Springer-Verlag Berlin Heidelberg:** (2013), 1-12.

[10] D. Behera and S. Chakraverty, ***Solution of fuzzy system of linear equations with polynomial parametric form***, **Applications and Applied Mathematics** 7 (2),(2012), 648-657.

[11] S. Das and S. Chakraverty**,** ***Numerical solution of interval and fuzzy system of linear equations,*** **National Institute of Technology Rourkela, India,** Vol. 7, Issue 1 (June 2012), pp. 334 – 356.

[12] M. Dehghan and B. Hashemi, ***Iterative solution of fuzzy linear systems,*** **Applied Mathematics and Computation** 175 (2006) 645–674.

[13] M. Dehghan, B. Hashemi, and M. Ghatee, ***Computational methods for solving fully fuzzy linear systems***, **Applied Mathematics and Computation** 179.1 (2006): 328-343.

[14] M. Dehghan, B. Hashemi and M. Ghatee, ***Solution of the fully fuzzy linear systems using iterative techniques,*** **Chaos Solutions and Fractals**, Volume 34, Issue 2, October(2007), 316–336 .

[15] R. Ezzati, ***Solving fuzzy linear systems***, **Soft Comput** (2011) 15:193–197.

[16] M. Ma, M. Friedman and A. Kandel, ***A new fuzzy arithmetic***, **Fuzzy sets and systems** 108 (1999), 83-90.

[17] M. Friedman, M. Ming and A. Kandel, ***Fuzzy linear systems,*** **Fuzzy Sets and Systems** 96(1998),pp 201-209.

[18] SH. Gao. Z. Zhang and C. Cao, ***Multiplication operation on fuzzy number,*** **Journal of Software**, Vol. 4, No. 4, 2009, 331-338.

[19] M. Ghanbari and R. Nuraei, ***Convergence of a semi-analytical method on the fuzzy Linear systems,*** **Iranian Journal of Fuzzy Systems** Vol. 11, No. 4, (2014), 45-60.

[20] S. Majumdar, ***On fuzzy system of linear equations,*** **Frontiers of Mathematics and Its Applications** (2014), 34-37.

[21] P. Mansouri and B. Asady, ***Iterative methods for solving fuzzy linear systems,*** **Australian Journal of Basic and Applied Sciences**, 5(7), (2011), pp. 1036-1049.

[22] S. Muzzioli and H. Reynaerts, ***Fuzzy linear systems of the form*** $\boldsymbol{A_1 x + b_1 = A_2 x + b_2}$,**Fuzzy Sets and Systems**, Vol. 157, (2006),pp. 939–951.

[23] H. Nasseri, ***Fuzzy numbers: positive and nonnegative,*** **International Mathematical Forum,** 3, (2008), no. 36, 1777 – 1780.

[24] P. Senthilkumar and G. Rajendran, ***An algorithmic approach to solve fuzzy linear systems,*** **Journal of Information & Computational Science** 8: 3 (2011) 503–510.

[25] P. Senthilkumar and G. Rajendran, ***New approach to solve symmetric fully fuzzy linear systems,*** **Sadhana** Vol. 36, Part 6, December (2011), 933–940.

[26] P. Senthilkumar and G. Rajendran, ***Solution of fuzzy linear systems by using fuzzy centre,*** **Applied Mathematical Sciences**, Vol. 3, No. 49, (2009), 2411 – 2419.

[27] L. A. Zadeh, ***Fuzzy sets*, Information Control**, vol. 8, (1965), pp.338-353.

[28] H. J. Zimmermann, ***Fuzzy set theory***, **John Wiley & Sons, Inc. WIREs Comp Stat** 2 (2010) 317–332.

Appendix

MATLAB Code for Jacobi Iterative Techniques for Solving Fuzzy System of Linear Equations

```
% Solving Fuzzy System of Linear Equations (FSLE)
% Jacobi Method


clc
clear all
close all
syms r


%% Inputs;
S = [1  0  0 -1;
     1  3  0  0;
     0 -1  1  0;
     0  0  1  3];
Y = [  r;
     4+r;
     2-r;
     7-2*r];


% S = [10 0   0 -4;
```

```
%      0  16 -4  0;
%      0 -4  10  0;
%     -4  0   0 16]; % Change for your example
% Y = [26+2*r;
%      34+2*r;
%      31-3*r;
%      38-2*r]; % Change for your example

% S = [ 4  1  0  0  0 -1;
%       0  3  1 -1  0  0;
%       2  1  3  0  0  0;
%       0  0 -1  4  1  0;
%      -1  0  0  0  3  1;
%       0  0  0  2  1  3];
% Y = [   +r ;
%        2+r ;
%       -2   ;
%        2-r ;
%        3   ;
%       -1-r];

N= length(Y)/2;
Y_U = Y(1:N);
Y_L = Y(N+1:2*N);
```

```
x0_U=zeros(N,1);
x0_L=zeros(N,1);
toler = 0.001;   % Change for your example
err = 1;
max = 500;   % Change to biger, if the solution didn't converge
% x_axis_name = 'Put a name';    % Change for your example
y_axis_name = 'r - Membership Value';

%% Start Coding with Jacobi Method;
T=0;
for K = 1:max
%% Check if the matrix A is diagonally dominant
for i = 1:2*N
   j = 1:2*N;
   j(i) = [];
   B = abs(S(i,j));
   Check(i) = abs(S(i,i)) - sum(B); % Is the diagonal value greater than the
remaining row values combined?
  if Check(i) < 0
     T=1;
     fprintf('The matrix is not strictly diagonally dominant at row %2i\n\n',i)
  end
end
```

```
if T == 1
   break
end

%% Jacobi Iteration; Based on Paper {5,4}
for I = 1:N
   sumL1=0;
   sumL2=0;
   for J = 1:N
      if J ~= I
      sumL1 = sumL1 + S(I,J)*x0_L(J);
      end
      sumL2 = sumL2 + S(I,N+J)*x0_U(J);
   end
   x_L(K,I) = eval((Y_L(I) - sumL1 - sumL2)/S(I,I));

   sumU1=0;
   sumU2=0;
   for J = 1:N
      if J ~= I
      sumU1 = sumU1 + S(I,J)*x0_U(J);
      end
      sumU2 = sumU2 + S(I,N+J)*x0_L(J);
   end
```

```
    x_U(K,I) = eval((Y_U(I) - sumU1 - sumU2)/S(I,I));
end

%% Cheching Error;
  ML1 = Noreta(x_L(K,:));
  MU1 = Noreta(x_U(K,:));
  ML2 = Noreta(x0_L);
  MU2 = Noreta(x0_U);
err1 = vpa(subs(abs(ML1 - ML2),r,0:0.1:1));
err2 = vpa(subs(abs(MU1 - MU2),r,0:0.1:1));
err = [err1,err2];
if  double(err) <= toler
    XL = x_L(end,:);
    XU = x_U(end,:);
    Error = err;
    NoIterations = K;
    break
end
 x0_L = x_L(K,:)';
 x0_U = x_U(K,:)';
end

%% Puting Answers in one vector X (XL1 XU1 XL2 XU2 . . . )
j=1;
```

```
for i = 1:2:2*N
    X(i) = XL(j);
    j= j +1;
end
j=1;
for i = 2:2:2*N
    X(i) = XU(j);
    j= j +1;
end

%% Showing Answers
ANS = X;
NoIterations = K

for i = 1:2*N
F(i,:) = subs(ANS(i),r,(0:0.05:1));
plot(F(i,:),(0:0.05:1),'+','markersize',11)
hold all
end
% xlabel(x_axis_name)
ylabel(y_axis_name)
title('Fuzzy system Using Jacobi')
% grid on
% the colour gradiant starts from Blue
```

```
%% Exact Solution based on X = (1/S)*Y
Ans = vpa(inv(S)*Y);
i=1;
j=1;
for i = 1:length(Ans)
   if i <= length(Ans)/2
       v1(i)= Ans(i);
   else v2(j)=Ans(i);
       j = j + 1;
   end
   i=i+1;
end

for i = 1:length(v1)
   Ans_F(i,:) = [v1(i) v2(i)];
end
Ans_F
for i = 1:2*N
TRR(i,:) = subs(Ans(i),r,(0:0.05:1));
plot(TRR(i,:),(0:0.05:1),'o')
hold on
end
text(2.4,0.9,'+ Jacobi Solution')
text(2.4,0.85,'O Exact Solution')
set(text,'linewidth',2)
format long
```

MATLAB Code for Gauss - Sidel Iterative Techniques for Solving Fuzzy System of Linear Equations

```
% Solving Fuzzy System of Linear Equations (FSLE)
% Gauss Seidel Method

clc
clear all
close all
syms r

%% Inputs;
% S = [2  1  3  0  0  0;
%      4  1  0  0  0 -1;
%      0  3  1 -1  0  0;
%      0  0  0  2  1  3;
%      0  0 -1  4  1  0;
%     -1  0  0  0  3  1];
% Y = [11+08*r;
%      27-08*r;
%     -23+10*r;
%     -05-08*r;
%      10+05*r;
%      27-12*r];
```

```
% S = [1  0  0 -1;
%      1  3  0  0;
%      0 -1  1  0;
%      0  0  1  3];
% Y = [  r;
%      4+r;
%      2-r;
%      7-2*r];

S = [ 4  1  0  0  0 -1;
      0  3  1 -1  0  0;
      2  1  3  0  0  0;
      0  0 -1  4  1  0;
     -1  0  0  0  3  1;
      0  0  0  2  1  3];
Y = [   r ;
       2+r ;
      -2   ;
       2-r ;
       3   ;
      -1-r];

% S = [10 0   0 -4;
%      0  16 -4  0;
```

```
%      0 -4  10  0;
%     -4  0   0 16]; % Change for your example
% Y = [26+2*r;
%      34+2*r;
%      31-3*r;
%      38-2*r]; % Change for your example

N= length(Y)/2;
Y_U = Y(1:N);
Y_L = Y(N+1:2*N);

x0_U=zeros(N,1);
x0_L=zeros(N,1)';
toler = 0.001;   % Change for your example
err = 1;
max = 500;   % Change to biger, if the solution didn't converge
% x_axis_name = 'Put a name';    % Change for your example
y_axis_name = 'r - Membership Value';

%% Starting Program
% Check if the matrix A is diagonally dominant
T=0;
for i = 1:2*N
   j = 1:2*N;
```

```
    j(i) = [];
    B = abs(S(i,j));
    Check(i) = abs(S(i,i)) - sum(B); % Is the diagonal value greater than the
remaining row values combined?
  if Check(i) < 0
      T=1;
      fprintf('The matrix is not strictly diagonally dominant at row %2i\n\n',i)
  end
end

if T == 1
    break
end

% Gauss-Seidel Iteration Technique; Based on Paper {5}
for K = 1:max
for I = 1:N
    sumL1=0;
    sumL2=0;
    sumL3=0;
    for J = 1:I-1
        sumL1 = sumL1 + S(I,J)*x_L(K,J);
    end
    for J = I+1:N
```

```
        sumL2 = sumL2 + S(I,J)*x0_L(J);
    end
    for J = 1:N
        sumL3 = sumL3 + S(I,J+N)*x0_U(J);
    end
    x_L(K,I) = eval((Y_L(I) - sumL1 - sumL2 - sumL3)/S(I,I));

    sumU1=0;
    sumU2=0;
    sumU3=0;
    for J = 1:I-1
        sumU1 = sumU1 + S(I,J)*x_U(K,J);
    end
    for J = I+1:N
        sumU2 = sumU2 + S(I,J)*x0_U(J);
    end
    for J = 1:N
        sumU3 = sumU3 + S(I,J+N)*x0_L(J);
    end
    x_U(K,I) = eval((Y_U(I) - sumU1 - sumU2 - sumU3)/S(I,I));
end

% Cheching Error;
  ML1 = Noreta(x_L(K,:));
```

```
  MU1 = Noreta(x_U(K,:));
  ML2 = Noreta(x0_L);
  MU2 = Noreta(x0_U);
err1 = vpa(subs(abs(ML1 - ML2),r,0:0.1:1));
err2 = vpa(subs(abs(MU1 - MU2),r,0:0.1:1));
err = [err1,err2];
if  double(err) <= toler
    XL = x_L(end,:);
    XU = x_U(end,:);
    Error = err;
    NoIterations = K;
    break
end
 x0_L = x_L(K,:)';
 x0_U = x_U(K,:)';
end

%% Puting Answers in one vector X (XL1 XU1 XL2 XU2 . . . )
j=1;
for i = 1:2:2*N
    X(i) = XL(j);
    j= j +1;
end
j=1;
```

```
for i = 2:2:2*N
    X(i) = XU(j);
    j= j +1;
end

%% Showing Answers
ANS = vpa(X);
NoIterations = K

for i = 1:2*N
F(i,:) = subs(ANS(i),r,(0:0.05:1));
plot(F(i,:),(0:0.05:1),'+','lineWidth',2,'markersize',11)
hold all
end
% xlabel(x_axis_name)
ylabel(y_axis_name)
% title('+ Gauss Zidel, O Exact Solution')
% grid on
%% the colour gradiant starts from Blue

%% Exact Solution based on X = (1/S)*Y
Ans = vpa(inv(S)*Y);

i=1;
```

```
j=1;
for i = 1:length(Ans)
   if i <= length(Ans)/2
       v1(i)= Ans(i);
   else v2(j)=Ans(i);
       j = j + 1;
   end
   i=i+1;
end

for i = 1:length(v1)
   Ans_F(i,:) = [v1(i) v2(i)];
end
Ans_F
for i = 1:2*N
TRR(i,:) = subs(Ans(i),r,(0:0.05:1));
plot(TRR(i,:),(0:0.05:1),'or')
hold on
end
text(0.5,0.9,'+ Gauss Zidel')
text(0.5,0.85,'O Exact Solution')
set(text,'linewidth',2)
format long
```

MATLAB Code for $LU-$ decomposition Method for Solving Fuzzy System of Linear Equations

```
function [X,S1,S2,L,U] = LUFLLE(A,Y)
   % solving Fuzzy number system using LU factorization.
   % [X,S1,S2,L,U] = LUFLLE(A,Y). A: the system matrix.
   % Y: Fuzzy numbers matrix (n-by-2).
   % returns: X: the solution vector, the matrices S1 and S2,
   % and the LU factorization.
   %% Initialization and Pre-setting
   An = A(:);
   S1 = [];
   S2 = [];
   Y = Y(:);

   for i = 1:length(An)
      if (An (i) > 0)
         S1(i) = An (i);
         S2(i) = 0;
      else
         S2(i) = An (i);
         S1(i) = 0;
      end
   end
```

```
S1 = reshape(S1,size(A));
S2 = reshape(S2,size(A));
S = [S1 S2;S2 S1];

%% LU factorization Process
for i = 1 : size(S,1)
   M(i) = det(S(1:i,1:i));
   if (M(i) == 0)
      fprintf('the principal minor %i is zero, LU fact. is not unique!!!',i);
      break;
   end
end
a = (M(1:length(M)-1) <= 0);
if sum(a) == 0
   disp(' S has a unique LL\^t Factorization');
end
[L,U,X] = LUfact(S,Y); % calling the function of LU-fact.

%% Display
X2 = X; % some resetting
for i = 1 : length(X)
   X3 = inline(X2(i));
   if (((X3(1) - X3(0)) < 0) && i <= length(X)./2)
      X(i) =  X(i);
```

```
        end
        if (((X3(1) - X3(0) > 0)) && i > length(X)./2)
            X(i) =  X(i);
        end
    end


    X = simplify(X); % simplification
    fprintf('\n\t X = \n')
    for i=1: length(Y)./2
        fprintf('\t\t ( %s , %s )\n',char(X(i)),char(X(i+length(Y)/2)));
    end
    figure('color','w')
    myplots(X); % calling myplot function
    hold off;
end
```

MATLAB code for $LU-$ factorization for Fuzzy System of Linear Equations

```
function [L,U,x]=LUfact(A,b)
  sa = size(A);
  sb = size(b);
  if(sa(1)==sa(2) && sa(2)==sb(1) && sb(2)==1)
    n=sa(1);
    %% LU Factorization...
    for i=1:n
      U(i,i)=1;
    end
    L(1,1)=sqrt(A(1,1));
    U(1,1)=L(1,1); %% L11*U11=A11;
    if(U(1,1)*L(1,1)==0)
      error(' ** no possible LU factorization!!!');
    else
      for(j=2:n)
        U(1,j)=A(1,j)/L(1,1);
        L(j,1)=A(j,1)/U(1,1);
      end
      for(i=2:n-1)
        s=0;
        for k=1:i-1
          s=s+L(i,k)*U(k,i);
        end
```

```
    L(i,i)=sqrt(A(i,i)-s);
    U(i,i)=L(i,i);
    if(L(i,i)*U(i,i)==0)
        error(' ** no possible LU factorization!!!');
        t=0;
    else
        t=1;
        for(j=i+1:n)
            su=0; sl=0;
            for(k=1:i-1)
                su=su+L(i,k)*U(k,j);
                sl=sl+L(j,k)*U(k,i);
            end
            U(i,j)=(A(i,j)-su)/L(i,i);% i-th row of U.
            L(j,i)=(A(j,i)-sl)/U(i,i);%i-th column of L.
        end
    end
end
if(t==1)
    s=0;
    for (k=1:n-1)
        s=s+L(n,k)*U(k,n);
    end
    L(n,n)=sqrt(A(n,n)-s);
    U(n,n)=L(n,n);
    if(L(n,n)*U(n,n)==0)
        disp(' ** the matrix is singular!!!');
```

```
            end
         end
      end
      %% Forward substitution...
      y(1,1)=b(1,1)/L(1,1);
      for (i=2:n)
         s=0;
         for(j=1:i-1)
            s=s+L(i,j)*y(j,1);
         end
         y(i,1)=(b(i,1)-s)/L(i,i);
      end
      %% Backward substitution...
      x(n,1)=y(n,1)/U(n,n);
      for(i=n-1:-1:1)
         s=0;
         for(j=i+1:n)
            s=s+U(i,j)*x(j,1);
         end
         x(i,1)=(y(i,1)-s)/U(i,i);
      end
   end
end
```

MATLAB Code for SOR Iterative Techniques for Solving Fuzzy System of Linear Equations

```
% Solving Fuzzy System of Linear Equations (FSLE)
% Successive over/under Relaxation ()SOR


clc
clear all
close all
syms r
t = cputime;

% Inputs;
S =  [ 2  0  0 -2;
       2  6  0  0;
       0 -2  2  0;
          0 0 2 6];



Y = [   2*r ;
      8+2*r ;
       4-2*r ;
      14-4*r];
```

```
% S = [10 0   0 -4;
%      0  16 -4  0;
%      0 -4  10  0;
%     -4  0   0 16]; % Change for your example
% Y = [26+2*r;
%      34+2*r;
%      31-3*r;
%      38-2*r]; % Change for your example

% S = [8 2 1 0 0 0 0 0 0 3;
%      0 5 1 0 1 2 0 0 1 0;
%      1 0 5 1 1 0 1 0 0 0;
%      0 0 0 4 2 0 0 1 0 0;
%      1 0 0 0 3 0 2 0 0 0;
%      0 0 0 0 3 8 2 1 0 0;
%      2 0 0 1 0 0 5 1 0 1;
%      0 1 0 0 0 1 0 5 1 1;
%      0 0 1 0 0 0 0 0 4 2;
%      0 2 0 0 0 1 0 0 0 3];
%  Y = [r;
%       4+r;
%       1+2*r;
%       1+r;
%       3*r;
```

```
%       2-r;
%       7-2*r;
%       6-3*r;
%       3-r;
%       6-3*r];

N= length(Y)/2;
Y_U = Y(1:N);
Y_L = Y(N+1:2*N);
maxEter = 300;
Toler = 0.001;

omega = 0.4;  % [0,1]

%% Getting D,L,U from S
d = diag(S);
for i=1:2*N
    for j=1:2*N
        if i~=j
        D(i,j) = 0;
        else
            D(i,j) = d(i);
        end
    end
```

```
end
L = tril(S);
U = triu(S);
L1 = L(1:N,1:N);
D1 = D(1:N,1:N);
U1 = U(1:N,1:N);
C = L(N+1:2*N,1:N);
s = D + L + U;
B = D1 + L1 + U1;

%% Iteration
x0_L = zeros(N,1);
x0_U = zeros(N,1);
% Equations from paper Iterative solution of fuzzy linear systems
for k = 1:maxEter
x_L = inv(D1+omega*L1)*(omega*Y_L + ((1-omega)*D1-
omega*U1)*x0_L -omega*C*x0_U);
x_U = inv(D1+omega*L1)*(-omega^2*C*inv(D1+omega*L1)*Y_L +
omega*Y_U + ((1-omega)*D1-
omega*U1+omega^2*C*inv(D1+omega*L1)*C)*x0_U -
omega*C*inv(D1+omega*L1)*((1-omega)*D1-omega*U1)*x0_L);
i=0;
for jj = 0:0.05:1
   i=i+1;
```

```
err1(:,i) = subs(abs(x_L - x0_L),r,jj);
err2(:,i) = subs(abs(x_U - x0_U),r,jj);
end
err = [err1,err2];
if max(double(err)) <= Toler
   break
end

x0_L = x_L;
x0_U = x_U;
end

%% Puting Answers in one vector X (XL1 XU1 XL2 XU2 . . . )
j=1;
for i = 1:2:2*N
   X(i) = x_L(j);
   j= j +1;
end
j=1;
for i = 2:2:2*N
   X(i) = x_U(j);
   j= j +1;
end
```

```
%% Exact Solution based on X = (1/S)*Y
GGG = vpa(inv(S)*Y);
i=1;
j=1;
for i = 1:length(GGG)
    if i <= length(GGG)/2
         v1(i)= GGG(i);
    else v2(j)=GGG(i);
         j = j + 1;
    end
    i=i+1;
end

for i = 1:length(v1)
    Ans_F(i,:) = [v1(i) v2(i)];
end
Ans_F

%% Showing Answers
ANS = X;
NoIterations = k

for i = 1:2*N
F(i,:) = subs(Ans_F(i),r,(0:0.05:1));
```

```
plot(F(i,:),(0:0.05:1),'+','markersize',11)
hold all
end
% xlabel('put a name')
ylabel('Membership Value')
title('+ SOR Solution, O Exact Solution')
grid on

for i = 1:2*N
KKK(i,:) = subs(Ans_F(i),r,(0:0.05:1));
plot(KKK(i,:),(0:0.05:1),'or')
hold on
end
t_CPU_SOR = cputime-t
% t_CPU_SOR = cputime
```

MATLAB Code for Jacobi and Gauss-Sidel Iterative Techniques for Solving Fully Fuzzy System of Linear Equations

```
function [X,Y,Z] = iter(A,M,N,b,g,h,X0,Y0,Z0,eps)
   % [X,Y,Z] = jaco(A,M,N,b,g,h,X0,Y0,Z0,eps), Takes the fully Fuzzy
system
   % Matrices, A,M,N,b,g,h and the precision "eps" and the initial values
X0,
   % Y0 and Z0, and returns the solution vectors X,Y,Z.

   disp(' Select Which Methood to use:'); % selection of the desired
method.
   s = lower(input (' enter "J" for Jacobi, or "S" for Gauss-Seidal:... ','s'));

   % initialization.
   [n,m] = size(A);
   L = zeros(n,m);
   U = zeros(n,m);
   % Calculate the permutation matrix E.
   E = zeros(n,m);
   [v,inda] = max(A);
   for i = 1:n
      [r,p] = max (v);
      E(inda(i),p) = 1;
```

```
   v(p) = -exp(-10);
end
% Transform the system to Diagonally dominant system.
A = E * A;
% Calculate Da, La, and Ua.
D = diag( diag (A));
for i = 1:n
   for j = 1:m
      if (i > j)
         L(i,j) = A(i,j);
      elseif (i < j)
         U(i,j) = A(i,j);
      end
   end
end

Di = inv(D);
DLi = inv(D+L);
if (det(D) == 0)
   error('Da is Singular !!!');
end

% start the iterations.
e = 1000;
```

```
i = 0;
if (s == 'j') % Jacobi Method.
   fprintf('\n\t\t\t******* Starting Jacobi Method ******* ');
   pause
   fprintf('\n\n \t ** solution for X: ')
   while (e > eps) % Solving for X
      i = i + 1;
      X = - Di*((L+U)*X0 - E*b);
      e = norm(X - X0);
      X0 = X;
      fprintf('\n iteration %d: \t (%s) ,\t error: %.4f',i,num2str(X',5),e);
      if (e > (50/eps))
         error ('The process is Diverging !!!');
         Break;
      end
   end
   i = 0; e = 1000;
   fprintf('\n\n \t ** solution for Y: ')
   while (e > eps) % Solving for Y
      i = i + 1;
      Y = - Di*((L+U)*Y0 + E*(M*X - g));
      e = norm(Y - Y0);
      Y0 = Y;
      fprintf('\n iteration %d: \t (%s) ,\t error: %.4f',i,num2str(Y',5),e);
```

```
    end
    i = 0; e = 1000;
    fprintf('\n\n \t ** solution for Z: ')
    while (e > eps) % Solving for Z
        i = i + 1;
        Z = - Di*((L+U)*Z0 + E*(N*X - h));
        e = norm(Z - Z0);
        Z0 = Z;
        fprintf('\n iteration %d: \t (%s) ,\t error: %.4f',i,num2str(Z',5),e);
    end
    fprintf('\n\n the approximation error is: %f', e);

elseif (s == 's') % Gauss-Seidal Method.
    fprintf('\n\t\t\t******* Starting Gauss-Seidal Method ******* ');
    pause
    fprintf('\n\n \t ** solution for X: ')
    while (e > eps) % Solving for X
        i = i + 1;
        X = - DLi*(U*X0 - E*b);
        e = norm(X - X0);
        X0 = X;
        fprintf('\n iteration %d: \t (%s) ,\t error: %.4f',i,num2str(X',5),e);
        if (e > (50/eps))
            error ('The process is Diverging !!!');
```

```
            Break;
        end
    end
    i = 0; e = 1000;
    fprintf('\n\n \t ** solution for Y: ')
    while (e > eps) % Solving for Y
        i = i + 1;
        Y = - DLi*(U*Y0 + E*(M*X - g));
        e = norm(Y - Y0);
        Y0 = Y;
        fprintf('\n iteration %d: \t (%s) ,\t error: %.4f',i,num2str(Y',5),e);
    end
    i = 0; e = 1000;
    fprintf('\n\n \t ** solution for Z: ')
    while (e > eps) % Solving for Z
        i = i + 1;
        Z = - DLi*(U*Z0 + E*(N*X - h));
        e = norm(Z - Z0);
        Z0 = Z;
        fprintf('\n iteration %d: \t (%s) ,\t error: %.4f',i,num2str(Z',5),e);
    end
    fprintf('\n\n the approximation error is: %f \n\n', e);
  end

end
```

جامعة النجاح الوطنية
كلية الدراسات العليا

الطرق العددية لحل نظام من المعادلات الخطية الضبابية

إعداد
لبنى لبيب احمد انعيرات

إشراف
أ.د. ناجي قطناني

قدمت هذه الأطروحة استكمالاً لمتطلبات الحصول على درجة الماجستير في الرياضيات بكلية الدراسات العليا، في جامعة النجاح الوطنية، نابلس – فلسطين.
2017

الطرق العددية لحل نظام من المعادلات الخطية الضبابية

إعداد

لبنى لبيب احمد انعيرات

إشراف

أ.د. ناجي قطناني

الملخص

في هذه الاطروحة ركزنا على حل نظام من المعادلات الخطية الضـــبابية بحيث تكون معاملات النظام اعداد حقيقية اما الثوابت والمتغيرات فهي اعداد ضـبابية ، وركزنا على مناقشـة بعض الطرق لحل نظام من المعادلات الكاملة الخطية الضــــبابية بحيث تكون كلاً من المعاملات والمتغيرات والثوابت اعداد ضبابية.

الطرق التحليلية لحل نظام من المعادلات الخطية الضـبابية شـملت طريقة فريدمان، طريقة سـعيد ابســـبندي وماجد علوي، طريقة كريمر ، طريقة تحليل مصـــفوفة المعاملات لمصـــفوفة عليا ودنيا وطريقة الاستبعاد لجاوس.

اما الطرق العددية التي تناولناها هي: طريقة جاكوبي، طريقة جاوس ســــايدل وطريقة التتابع. تم تنفيذ الامثلة العددية بهذه الطرق وتم وضـــع مقارنة بينها حيث اظهرت لنا النتائج العددية ان طريقة التتابع كانت اكثر كفاءة بالمقارنة مع الجاكوبي والجاوس سايدل فقد تم الوصول الى الحل بتكرارات وزمن اقل.

الطرق التحليلية لحل نظام من المعادلات الكاملة الخطية الضبابية كانت طريقة المعاكس المباشر ، طريقة كريمر وطريقة تحليل مصفوفة المعاملات لمصفوفة عليا ودنيا.

والطرق العديية اشتملت على طريقة الجاكوبي والجاوس سايدل. تم تنفيذ الامثلة العددية بهذه الطرق وتم وضع مقارنة بين هذه الطرق العددية حيث اظهرت لنا النتائج العددية ان طريقة الجاوس سايدل اكثر كفاءة بالمقارنة مع الجاكوبي فقد تم الوصول الى الحل بتكرارات وزمن وخطأ اقل بالمقارنة مع طريقة الجاكوبي.