



# Fully Fuzzy Linear Systems in Python Programming

V.Vijayalakshmi, S. Surabhi, A.Karpagam

**Abstract:** This paper proposes the python coding for ST decomposition for Triangular, Trapezoidal, and computing the algorithms for the fully fuzzy linear system in python programming.  $\bar{A}\bar{x} = \bar{b}$  where  $\bar{A}$  is a fuzzy matrix,  $\bar{x}$  and  $\bar{b}$  are fuzzy vectors. ST decompose into a product of symmetric matrix (S) and triangular matrix (T) in the form of triangular and trapezoidal fuzzy number matrices. To best illustrate the proposed methods by python coding algorithm with a new approach Python coding has been adopted. Algorithms have been introduced and the numerical examples have been solved by using python techniques. A study of ST decomposition have been done and the solution is obtained with different algorithms. New numerical problems are presented and an example has been solved for this algorithms and the solutions are obtained.

**Keywords:** Fully Fuzzy Linear System of equations, Triangular fuzzy numbers, Trapezoidal Fuzzy numbers, Python Programming.

## I. INTRODUCTION

L. A. Zadeh introduced the concepts of fuzzy set theory, fuzzy numbers, and fuzzy arithmetic. Fuzzy sets with different types are defined to identify the vagueness of the existing problems. Imprecise values are denoted as fuzzy number.. Fuzzy numbers like a triangular fuzzy number, trapezoidal fuzzy number, Pentagonal, Hexagonal, Octagonal, Pyramid and Diamond fuzzy number have been introduced with its membership function.

A. Kumar et.al [1,2,3] introduced fully fuzzy linear system with arbitrary coefficients, A New Approach for Solving Fully Fuzzy Linear Systems, A New Computational Method for Solving Fully Fuzzy Linear Systems of Triangular Fuzzy Numbers.

Linear system of equations has widely applied in many areas of science, economics, mathematics, physics, operation research, management, and engineering. Researches introduce fuzzy numbers instead of crisp numbers. Mosleh [5,6] proposed the decomposition Method for Solving Fully Fuzzy Linear Systems fully fuzzy linear system.  $A_1x + b_1 = A_2x + b_2$  form in the fully fuzzy linear system of the form by S.Nasseri, M. Matinfar and Z.

Revised Manuscript Received on April 25, 2020.

\* Correspondence Author

**Dr. V.Vijayalakshmi\***, Department of Mathematics, SRM Valliammai Engineering College(SRM group), Kattankulathur, Chennai, India.

**S. Surabhi**, Department of Electronics & Communication Engineering, SRM Valliammai Engineering College(SRM group), Kattankulathur, Chennai, India.

**A.Karpagam**, Department of Mathematics, SRM Valliammai Engineering College(SRM group), Kattankulathur, Chennai, India.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](https://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

Kheiri,[7] introduced. S. Nasseri and M. Sohrabi[8] initiated Cholesky Decomposition for solving the fully fuzzy linear system of equations in International Journal of Applied Mathematics.

Vijayalakshmi V.et.al [10,11,12] introduced the concepts of solving FFLS for triangular, trapezoidal, hexagonal and octagonal fuzzy numbers. Mosleh et.al [5] introduced the Decomposition Method for solving Fully Fuzzy Linear Systems in 2009. Ghassan Malkawi, Nazihah Ahmad an Haslinda Ibrahim [4], developed Solving fully fuzzy linear system with the necessary and sufficient condition to have a positive solution, Applied Mathematics & Information Sciences. Venkatesan, A[9] introduced algorithm for a fully fuzzy linear system with Hexagonal fuzzy number matrix by singular value decomposition, International Journal of Computing Algorithms.

In this paper, a new method is proposed to solve decomposition methods. Python techniques are used to solve the algorithms. Many researchers have proposed solving a fully fuzzy linear system of equations. Python is a versatile interpreted language. Python has evolved into a general interactive prototyping language. It has been applied to a wide range of problems, from replacements for large shell scripts to fancy graphics demos and multimedia applications.

One of Python's strengths is the ability for the user to type some code and immediately to work on it, no compilation or linking is necessary. Interactive performance is further enhanced by Python's concise, clear syntax, it's very high-level data types and its lack of declarations (which is compensated by run-time of the type checking). All this makes programming in Python feel like a leisure trip compared to the hard work involved in writing and debugging even a smallish C program.

This article mainly consists of preliminary definitions with their diagrammatic representation in section 2. Section 3 shows the algorithm, python coding, examples of ST decomposition of a triangular fully fuzzy linear system of equations, Section 4 shows the algorithm, python coding, examples of ST decomposition of a trapezoidal fully fuzzy linear system of equations. Section 5 deals with the conclusion of ST decomposition in different criteria.

## 1. Preliminaries

In this section, basic definitions of fuzzy set theory and fuzzy numbers are reviewed.

### 2.1 Definition Fuzzy Set

Let X be a non-empty set.

A fuzzy set A is characterized by its membership function  $A: X \rightarrow [0,1]$  and  $A(x)$  is interpreted as the degree of membership of an element in fuzzy A for each  $x \in X$ . The value 0 represent non membership and the value 1 represents membership in between values intermediate degrees of membership.

**Definition: Triangular Fuzzy Number**

A set  $(a_1, a_2, a_3)$  and its membership function is given by,

$$\mu_{\bar{a}}(x) = \begin{cases} 0, & x < a_1 \\ \frac{x-a_1}{a_2-a_1}, & a_1 \leq x \leq a_2 \\ \frac{x-a_3}{a_2-a_3}, & a_2 \leq x \leq a_3 \\ 0, & x > a_3 \end{cases}$$

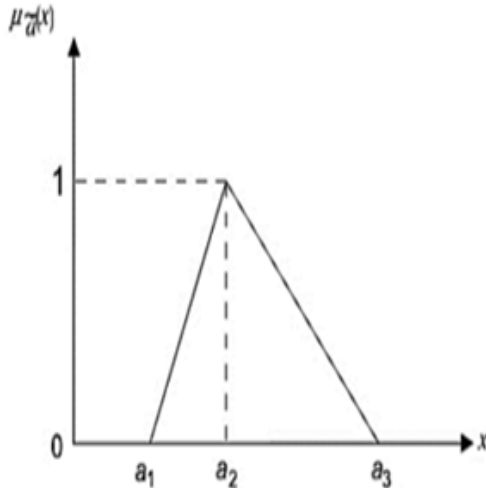


Fig1. Triangle fuzzy number  $\bar{a}$

**2.2 Definition: Trapezoidal Fuzzy Number**

A set  $(a_1, a_2, a_3, a_4)$  and its membership function is given by,

$$\mu_A(x) = \begin{cases} 0 & x < a_1 \\ \frac{x-a_1}{a_2-a_1} & a_1 \leq x \leq a_2 \\ 1 & a_2 \leq x \leq a_3 \\ \frac{a_4-x}{a_4-a_3} & a_3 \leq x \leq a_4 \\ 0 & a_4 > x \end{cases}$$

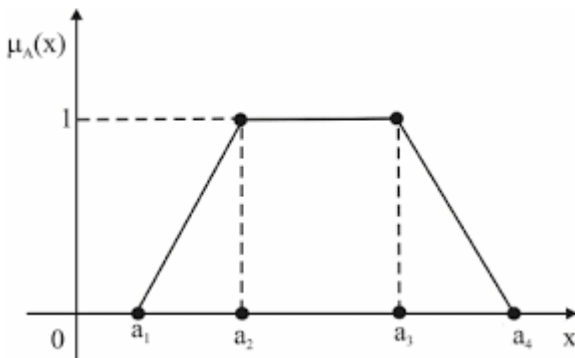


Fig2. Trapezoidal Fuzzy Number

**3 Proposed Method for Triangular Fuzzy Number**

To solve FFLS, in this section LU decomposition and ST decomposition algorithms are proposed.

**ST Decomposition Method**

A new ST decomposition method is proposed.

**STEP I** Consider the nonsingular matrix A such that

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

Where the elements of A are inputs of the matrix. Decompose A into ST where S is the symmetric matrix and T is the triangular matrix

$$A = ST \text{ where } S = \begin{pmatrix} a_{11} & a_{21} & a_{31} \\ a_{21} & s_{22} & s_{23} \\ a_{31} & s_{23} & s_{33} \end{pmatrix} \text{ and } T =$$

$$\begin{pmatrix} 1 & t_{12} & t_{13} \\ 0 & 1 & t_{23} \\ 0 & 0 & 1 \end{pmatrix}$$

**STEP II**

Calculate the unknown elements of the symmetric matrix and triangular matrix by solving the equation.

$$s_{22} = \frac{a_{21}^2 + a_{11}a_{22} - a_{21}a_{12}}{a_{11}}, \quad t_{12} = \frac{a_{12} - a_{21}}{a_{11}}$$

$$s_{23} = \frac{a_{31}a_{21} + a_{11}a_{32} - a_{31}a_{12}}{a_{11}}$$

$$a_{33}a_{21}^2 - a_{33}a_{11}s_{22} - a_{31}a_{21}a_{23} + a_{31}a_{13}s_{22} + 2a_{31}a_{21}s_{23}$$

$$s_{33} = \frac{-a_{31}^2s_{22} - s_{23}a_{13}a_{21} + s_{23}a_{11}a_{23} - a_{11}s_{23}^2}{a_{21}^2 - a_{11}s_{22}}$$

$$t_{13} = \frac{a_{21}a_{23} - a_{13}s_{22} - a_{21}s_{23} + a_{31}s_{22}}{a_{21}^2 - a_{11}s_{22}}$$

$$t_{23} = \frac{a_{21}(a_{13} - a_{31}) + a_{11}(s_{23} - a_{23})}{a_{21}^2 - a_{11}s_{22}}$$

**STEP III**

To solve  $A\bar{x} = \bar{b}$ ,

$$Ax = b$$

$$Mx + Ay = h$$

$$Nx + Az = g$$

**STEP IV**

Taking inverse ,

$$x = A^{-1}b$$

$$y = A^{-1}(h - Mx)$$

$$z = A^{-1}(g - Nx)$$

**STEP V**

Substituting  $A = ST$ , the equations obtained are,

$$x = T^{-1}S^{-1}b$$

$$y = T^{-1}S^{-1}(h - M T^{-1}S^{-1}b)$$

$$z = T^{-1}S^{-1}(g - N T^{-1}S^{-1}b)$$

**3.1 Programming Logic**

STEP 1: Get all the inputs from the user as Matrices a,m,n,b,h,g as stated in theory.  
STEP 2: Using ST\_Decompose function developed based on this algorithm find S and T matrix for the input given  
[s,t]=ST\_Decompose(A)  
STEP 3: Using the inverse function in Python, find the inverse of s and t.  
sinv=np.matrix(inv(s))  
tinv=np.matrix(inv(t))  
STEP 4: Compute values of x,y,z using the function Soln\_of\_ST\_Decompose.  
[x,y,z]=Soln\_of\_ST\_Decompose.(sinv,tinv,b,h,g,m,n)  
STEP 5: Display the Result

**3.2 Sample Code- ST Decomposition**

```
#ST Decomposition for triangular fuzzy numbers 3*3
matrix
#-----
#including modules for matrix operations
import numpy as np
from numpy.linalg import inv
np.set_printoptions(suppress=True)
#input Fuzzy Matrix
a=[[6,5,3],[12,14,8],[24,32,20]]
m=[[1,2,2],[8,12,8],[10,30,19]]
n=[[4,2,1],[20,15,10],[34,30,24]]
b=[[58],[142],[316]]
h=[[30],[139],[297]]
g=[[60],[257],[514]]
#matrix for ST decomposition [A=ST]
s=[[0,0,0],[0,0,0],[0,0,0]]
t=[[0,0,0],[0,0,0],[0,0,0]]
#Algorithm for ST Decomposition
s[0][0]=a[0][0]
s[0][1]=a[1][0]
s[0][2]=a[2][0]
s[1][0]=a[1][0]
s[1][1]=(a[1][0]*a[1][0])+(a[0][0]
*a[1][1])-(a[1][0]*a[0][1])/a[0][0]
s[1][2]=(a[2][0]*a[1][0])+(a[0][0]
*a[2][1])-(a[2][0]*a[0][1])/a[0][0]
s[2][0]=a[2][0]
s[2][1]=s[1][2]
s[2][2]=(a[2][2]*(a[1][0]*a[1][0]))-
(a[2][2]*a[0][0]*s[1][1])-
(a[2][0]*a[1][0]*a[1][2])+(a[2][0]*a[0][2]*s[1][1])+(2*a[2][
0]*s[1][2]*a[1][0])-(a[2][0]*a[2][0]*s[1][1])-
(s[1][2]*a[0][2]*a[1][0])+(a[1][2]
*a[0][0]*s[1][2])-(a[0][0]*s[1][2]*s[1][2]))/(a[1][0]
*a[1][0])-(a[0][0]*s[1][1]))
t[0][0]=1
t[0][1]=(a[0][1]-a[1][0])/a[0][0]
t[0][2]=(a[1][0]*a[1][2])-(a[0][2]*s[1][1])-
(a[1][0]*s[1][2])+(a[2][0]*s[1][1]))/(a[1][0]*a[1][0])-
(a[0][0]*s[1][1])
t[1][0]=0
t[1][1]=1
t[1][2]=(a[0][2]*a[1][0])-(a[0][0]*a[1][2])-
(a[2][0]*a[1][0])+(a[0][0]*s[1][2]))/(a[1][0]*a[1][0])-
(a[0][0]*s[1][1])
t[2][0]=0
t[2][1]=0
```

```
t[2][2]=1
#Displaying S and T matrix
print(s) # Symmetrical matrix
print(t) #Upper Triangular matrix
#Finding Inverse matrix of S and T
sinv=np.matrix(inv(s))
tinv=np.matrix(inv(t))
#Displaying Inverse matrix of S and T
print(sinv)
print(tinv)
#Multiplying inverse(T) and inverse(S)
tinv_sinv=tinv*sinv
#mtinv_sinv=np.matrix(tinv_sinv)
Storing as matrix
print(tinv_sinv)
#Computation of x,y,z values using ST decomposition
Algorithm
x=tinv_sinv*b
y=tinv_sinv*(h-m*x)
z=tinv_sinv*(g-n*x)
#Displaying Solutions of ST Decomposition
print(x)
print(y)
print(z)
```

These codes have been modified as sub programs which are referred as functions. These functions will be called whenever required. Using of those functions in our concept has been described in the Programming Logic section.

**3.3 Practical Application**

Let  $\bar{x}_i$  denotes the two wheeler varieties for  $i = 1,2,3$ . Let  $\bar{x}_1, \bar{x}_2, \bar{x}_3$  denotes the model 1, model 2, model 3 of two wheelers. . Let  $\bar{x}_1=x_1, y_1, z_1$ ,denotes the three parameters are level of satisfaction regarding mileage, Exterior design, safety & comfort for model 1. Let  $\bar{x}_2, \bar{x}_3$  denotes the same parameter level for model 2, model 3.

The study would be carried out by comparing competitors in the two wheeler industry for their three popular models. The relationship between two wheelers and their satisfaction level are given in the form of fully fuzzy linear system of Equations.

$$(6,1,4) \otimes (x_1, y_1, z_1) \oplus (5,2,2) \otimes (x_2, y_2, z_2) \oplus (3,2,1) \otimes (x_3, y_3, z_3) = (58,30,60)$$

$$(12,8,20) \otimes (x_1, y_1, z_1) \oplus (14,12,15) \otimes (x_2, y_2, z_2) \oplus (8,8,10) \otimes (x_3, y_3, z_3) = (142,139,257)$$

$$(24,10,34) \otimes (x_1, y_1, z_1) \oplus (32,30,30) \otimes (x_2, y_2, z_2) \oplus (20,19,24) \otimes (x_3, y_3, z_3) = (316,297,514)$$

**Table 1.Numerical Examples**

SL.NO	NUMERICAL EXAMPLES	ST decomposition
-------	--------------------	------------------



1.	$(6,1,4) \otimes (x_1, y_1, z_1) \oplus (5,2,2)$ $\otimes (x_2, y_2, z_2)$ $\oplus (3,2,1)$ $\otimes (x_3, y_3, z_3)$ $= (58,30,60)$  $(12,8,20) \otimes (x_1, y_1, z_1) \oplus (14,12,15)$ $\otimes (x_2, y_2, z_2)$ $\oplus (8,8,10)$ $\otimes (x_3, y_3, z_3)$ $= (142,139,257)$  $(24,10,34) \otimes (x_1, y_1, z_1) \oplus (32,30,30)$ $\otimes (x_2, y_2, z_2)$ $\oplus (20,19,24)$ $\otimes (x_3, y_3, z_3)$ $= (316,297,514)$	$\begin{pmatrix} (4,1,3) \\ (5,0,5,2) \\ (3,0,5,1) \end{pmatrix}$
2	$(1,2,5) \otimes (x_1, y_1, z_1) \oplus (3,4,4)$ $\otimes (x_2, y_2, z_2)$ $\oplus (0,1,2)$ $\otimes (x_3, y_3, z_3)$ $= (19,68,115)$  $(2,3,5) \otimes (x_1, y_1, z_1) \oplus (0,1,11)$ $\otimes (x_2, y_2, z_2)$ $\oplus (4,5,6)$ $\otimes (x_3, y_3, z_3)$ $= (30,77,261)$  $(2,5,7) \otimes (x_1, y_1, z_1) \oplus (4,6,6)$ $\otimes (x_2, y_2, z_2)$ $\oplus (5,7,10)$ $\otimes (x_3, y_3, z_3)$ $= (61,167,253)$	$\begin{pmatrix} (1,6,7) \\ (2,12,10) \\ (7,14,12) \end{pmatrix}$
3	$(1,4,4) \otimes (x_1, y_1, z_1) \oplus (2,6,6)$ $\otimes (x_2, y_2, z_2)$ $\oplus (3,6,1)$ $\otimes (x_3, y_3, z_3)$ $= (10,36,172)$  $(2,6,6) \otimes (x_1, y_1, z_1) \oplus (8,18,4)$ $\otimes (x_2, y_2, z_2)$ $\oplus (8,12,4)$ $\otimes (x_3, y_3, z_3)$ $= (20,42,216)$  $(3,6,1) \otimes (x_1, y_1, z_1) \oplus (8,12,4)$ $\otimes (x_2, y_2, z_2)$ $\oplus (14,9,4)$ $\otimes (x_3, y_3, z_3)$ $= (30,40,56)$	$\begin{pmatrix} (10,2,5,344) \\ (0, -2.125, 10) \\ (0, -0.75, -74) \end{pmatrix}$

## II. PROPOSED M

### 4 ETHOD FOR TRAPEZOIDAL FUZZY NUMBER

To solve FFLS, in this section ST decomposition algorithms are proposed

#### 4.1 ST Decomposition Method

**STEP I** Consider the nonsingular trapezoidal fuzzy number matrices A.

$$\text{Set } A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}$$

**STEP II** Decompose the matrix A,  $A = ST$  where S is symmetric and T is an upper triangular matrix.

$$S = \begin{pmatrix} a_{11} & a_{21} & a_{31} & a_{41} \\ a_{21} & s_{22} & s_{23} & s_{24} \\ a_{31} & s_{23} & s_{33} & s_{34} \\ a_{41} & s_{24} & s_{34} & s_{44} \end{pmatrix},$$

$$T = \begin{pmatrix} 1 & t_{12} & t_{13} & t_{14} \\ 0 & 1 & t_{23} & t_{24} \\ 0 & 0 & 1 & t_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Calculate the unknown elements of the symmetric matrix and triangular matrix by solving the equation

$$\text{Where } t_{12} = \frac{a_{12} - a_{21}}{a_{11}}, \quad s_{22} = a_{22} - a_{21}t_{12}$$

$$s_{23} = a_{32} - a_{31}t_{12}$$

$$s_{24} = a_{42} - a_{41}t_{12}$$

$$t_{13} = \frac{a_{21}a_{23} - a_{13}s_{22} - a_{21}s_{23} + a_{31}s_{22}}{a_{21}^2 - a_{11}s_{22}}$$

$$t_{23} = \frac{a_{21}(a_{13} - a_{31}) + a_{11}(s_{23} - a_{23})}{a_{21}^2 - a_{11}s_{22}}$$

$$s_{33} = a_{33} - a_{31}t_{13} - s_{23}t_{23}$$

$$s_{34} = a_{43} - a_{41}t_{13} - s_{24}t_{23}$$

$$t_{14} = \frac{s_{23}(a_{14} - a_{41}) + a_{31}(s_{24} - a_{24})}{a_{11}s_{23} - a_{21}a_{31}}$$

$$t_{24} = \frac{s_{33}(a_{24} - s_{24}) + s_{23}(s_{34} - a_{34}) - t_{14}(a_{21}s_{33} - a_{41}s_{23})}{s_{22}s_{33} - s_{23}^2}$$

$$t_{34} = \frac{a_{14} - a_{11}t_{14} - a_{21}t_{24} - a_{41}}{a_{31}}$$

$$s_{44} = a_{44} - s_{41}t_{14} - s_{24}t_{24} - s_{34}t_{34}$$

#### STEP III

On solving  $\bar{A} \otimes \bar{x} = \bar{b}$  we have

$$(A, B, M, N) \otimes (x, y, z, w) = (b, g, h, k)$$

$$Ax = b \Rightarrow x = A^{-1}b$$

$$By = g \Rightarrow y = B^{-1}g$$

$$Az + Mx = h \Rightarrow z = A^{-1}(h - Mx)$$

$$Bw + Ny = k \Rightarrow w = B^{-1}(k - Ny)$$

#### STEP IV

Replace  $A = ST$ ,  $B = S_1T_1$  we have  $x = T^{-1}S^{-1}b$

$$y = \dots = T_1^{-1}S_1^{-1}g, \quad z = T^{-1}S^{-1}(h - Mx), w = T_1^{-1}S_1^{-1}(k - Ny)$$

Using this formula we are finding the solution of x,y,z,w using ST decomposition.

#### 4.2 Programming Logic

**STEP 1:** Get all the inputs from the user as Matrices A, B, M, N,b,g,h,k as stated in theory.

**STEP 2:** Using ST\_Decompose function developed based on this algorithm find S and T matrix for the input(A) given

[sa,ta]=ST\_Decompose(A)

STEP 3:UsingST\_Decompose function developed based on this algorithm find S and T matrix for the input(B) given

[sb,tb]=ST\_Decompose(B)

STEP 4: Using the inverse function in Python, find the inverse of s and t.

sainv=np.matrix(inv(sa))

tainv=np.matrix(inv(ta))

STEP 5:Using inverse function in Python, find inverse of s and t.

sbinv=np.matrix(inv(sb))

tbinv=np.matrix(inv(tb))

STEP 4: Compute values of x,y,z,w using the function Soln\_of\_ST\_Decompose.

[x,y,z,w]=Soln\_of\_ST\_Decompose.(sainv,tainv,sbinv,tbinv, b,h,g,k,M,N)

STEP 5:Display the Result

### 4.3 Sample Code- ST Decomposition

```
#ST Decomposition for trapezoidal fuzzy numbers 4*4 matrix
#~~~~~
#including modules for matrix operations
import numpy as np
from numpy.linalg import inv
np.set_printoptions(suppress=True)
#input Fuzzy Matrix
```

For Trapezoidal fuzzy numbers python coding has been coded similarly like triangular fuzzy numbers as explained in the algorithm.

### 4.4 Practical Applications

An industrialist maintains two companies each having two sectors namely pharmaceutical and chemical sector for company 1 and software and call centre for company 2. Let  $\bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{x}_4$  denotes the profit or loss (income generated) from pharma, software, chemical and call centre sectors respectively. Each sector has four types of industries namely small scale, medium scale, large scale and very large scale. The relationship between industry wise income and sector wise income are given in the form of FFLS.

$$\begin{aligned} &(1,4,4,4) \otimes (x_1, y_1, z_1, w_1) \oplus (2,6,6,6) \otimes (x_2, y_2, z_2, w_2) \\ &\oplus (3,6,1,2) \otimes (x_3, y_3, z_3, w_3) \\ &\oplus (2,4,3,6) \otimes (x_4, y_4, z_4, w_4) \\ &= (10,36,172,160) \\ &(2,6,6,6) \otimes (x_1, y_1, z_1, w_1) \oplus (8,18,4,3) \otimes (x_2, y_2, z_2, w_2) \\ &\oplus (8,12,4,3) \otimes (x_3, y_3, z_3, w_3) \\ &\oplus (6,9,4,6) \otimes (x_4, y_4, z_4, w_4) \\ &= (20,42,216,186) \\ &(3,6,1,2) \otimes (x_1, y_1, z_1, w_1) \oplus (8,12,4,3) \otimes (x_2, y_2, z_2, w_2) \\ &\oplus (14,11,4,2) \otimes (x_3, y_3, z_3, w_3) \\ &\oplus (11,9,4,8) \otimes (x_4, y_4, z_4, w_4) \\ &= (30,40,56,100) \\ &(2,4,3,6) \otimes (x_1, y_1, z_1, w_1) \oplus (6,9,4,6) \otimes (x_2, y_2, z_2, w_2) \\ &\oplus (11,9,4,8) \otimes (x_3, y_3, z_3, w_3) \\ &\oplus (10,13,2,3) \otimes (x_4, y_4, z_4, w_4) \\ &= (40,36,82,82) \end{aligned}$$

Table 2.Numerical Examples

SL_NO	NUMERICAL EXAMPLES	ST Decomposition
1.	$(1,4,4,4) \otimes (x_1, y_1, z_1, w_1) \oplus (2,6,6,6) \otimes (x_2, y_2, z_2, w_2) \oplus (3,6,1,2) \otimes (x_3, y_3, z_3, w_3) \oplus (2,4,3,6) \otimes (x_4, y_4, z_4, w_4)$ $= (10,36,172,160)$ $(2,6,6,6) \otimes (x_1, y_1, z_1, w_1) \oplus (8,18,4,3) \otimes (x_2, y_2, z_2, w_2) \oplus (8,12,4,3) \otimes (x_3, y_3, z_3, w_3) \oplus (6,9,4,6) \otimes (x_4, y_4, z_4, w_4)$ $= (20,42,216,186)$ $(3,6,1,2) \otimes (x_1, y_1, z_1, w_1) \oplus (8,12,4,3) \otimes (x_2, y_2, z_2, w_2) \oplus (14,11,4,2) \otimes (x_3, y_3, z_3, w_3) \oplus (11,9,4,8) \otimes (x_4, y_4, z_4, w_4)$ $= (30,40,56,100)$ $(2,4,3,6) \otimes (x_1, y_1, z_1, w_1) \oplus (6,9,4,6) \otimes (x_2, y_2, z_2, w_2) \oplus (11,9,4,8) \otimes (x_3, y_3, z_3, w_3) \oplus (10,13,2,3) \otimes (x_4, y_4, z_4, w_4)$ $= (40,36,82,82)$	$\begin{bmatrix} (30,30,32,28) \\ (0,4,5,4) \\ (-20,-22,-22,-24) \\ (20,6,18,8) \end{bmatrix}$
2	$(1,2,3,5) \otimes (x_1, y_1, z_1, w_1) \oplus (2,3,4,5) \otimes (x_2, y_2, z_2, w_2) \oplus (5,6,7,8) \otimes (x_3, y_3, z_3, w_3) \oplus (6,7,8,9) \otimes (x_4, y_4, z_4, w_4) = (1,2,3,4)$ $(3,4,7,8) \otimes (x_1, y_1, z_1, w_1) \oplus (4,5,6,8) \otimes (x_2, y_2, z_2, w_2) \oplus (8,9,10,11) \otimes (x_3, y_3, z_3, w_3) \oplus (7,9,11,12) \otimes (x_4, y_4, z_4, w_4) = (2,4,6,8)$ $(9,10,11,12) \otimes (x_1, y_1, z_1, w_1) \oplus (2,4,6,7) \otimes (x_2, y_2, z_2, w_2) \oplus (3,4,6,7) \otimes (x_3, y_3, z_3, w_3) \oplus (5,7,9,11) \otimes (x_4, y_4, z_4, w_4) = (3,5,7,11)$ $(4,6,7,8) \otimes (x_1, y_1, z_1, w_1) \oplus (1,3,4,5) \otimes (x_2, y_2, z_2, w_2) \oplus (2,6,7,8) \otimes (x_3, y_3, z_3, w_3) \oplus (6,7,8,9) \otimes (x_4, y_4, z_4, w_4) = (4,6,8,10)$	$\begin{bmatrix} (7106,1,554,75) \\ (-86,065,-1,239,2,3) \\ (-00065,-1,4187,0247, \\ (-337,1,188,-387,35) \end{bmatrix}$
3	$(1,3,4,6) \otimes (x_1, y_1, z_1, w_1) \oplus (5,4,8,10) \otimes (x_2, y_2, z_2, w_2) \oplus (6,8,11,13) \otimes (x_3, y_3, z_3, w_3) \oplus (8,10,14,17) \otimes (x_4, y_4, z_4, w_4) = (12,14,18,19)$ $(3,4,7,8) \otimes (x_1, y_1, z_1, w_1) \oplus (4,5,8,9) \otimes (x_2, y_2, z_2, w_2) \oplus (5,7,10,11) \otimes (x_3, y_3, z_3, w_3) \oplus (4,8,11,12) \otimes (x_4, y_4, z_4, w_4) = (20,32,36,45)$ $(7,9,10,14) \otimes (x_1, y_1, z_1, w_1) \oplus (12,15,18,19) \otimes (x_2, y_2, z_2, w_2) \oplus (7,10,12,16) \otimes (x_3, y_3, z_3, w_3) \oplus (7,9,10,14) \otimes (x_4, y_4, z_4, w_4) = (8,12,15,19)$ $(4,7,9,11) \otimes (x_1, y_1, z_1, w_1) \oplus (3,7,11,15) \otimes (x_2, y_2, z_2, w_2) \oplus (15,17,19,21) \otimes (x_3, y_3, z_3, w_3) \oplus (13,17,22,23) \otimes (x_4, y_4, z_4, w_4) = (10,12,13,15)$	$\begin{bmatrix} (0,223,4,563,-4,079,-6,563) \\ (4,684,-7,421,-61,228,6,487) \\ (0,198,3,664,20,669,8,413) \\ (-0,916,-0,426,8,413,0,058) \end{bmatrix}$

### III. CONCLUSION

A new approach Python coding has been adopted for this algorithm. Solutions of the fully fuzzy linear system both have positive and negative values. A new innovation python algorithms has been interpreted in order to find out

the unknowns for three and four variables using triangular and trapezoidal fuzzy numbers.



## REFERENCES

1. A. Kumar, Bansal and Neetu, Solution of a fully fuzzy linear system with arbitrary coefficients, International Journal of Applied Mathematics and Computation, **3**, 232-237 (2011).
2. A. Kumar, Neetu and A. Bansal, A New Approach for Solving Fully Fuzzy Linear Systems, Hindawi Publishing Corporation, 1-8 (2011)
3. A. Kumar, Neetu and A. Bansal, A New Computational Method for Solving Fully Fuzzy Linear Systems of Triangular Fuzzy Numbers, Fuzzy Inf. Eng., 63-73 (2012).
4. GhassanMalkawi, Nazihah Ahmad an Haslinda Ibrahim, Solving fully fuzzy linear system with the necessary and sufficient condition to have a positive solution, Applied Mathematics & Information Sciences, **8**, No.3, 1003-1019 (2014)
5. Mosleh, Otadi and. Khanmirzaie, Decomposition Method for Solving Fully Fuzzy Linear Systems, Iranian Journal of Optimization, 188-198 (2009).
6. Mosleh, Otadi Solving fully fuzzy matrix equations, Allied Mathematical Modelling, **36**, 6114–6121(2012).
7. Nasser, Sohrabi, and. Ardil, Solving Fully Fuzzy Linear Systems by use of a Certain Decomposition of the Coefficient Matrix, International Journal of Computational and Mathematical Sciences, **3**, 140-142 (2008).
8. Nasser and M. Sohrabi, Cholesky Decomposition for Solving The Fully Fuzzy Linear System Of Equations, International Journal of Applied Mathematics, **22**, 689-696 (2009).
9. Venkatesan, A Algorithm for a fully fuzzy linear system with Hexagonal fuzzy number matrix by singular value decomposition, International Journal of Computing Algorithms, 2014
10. V Vijayalakshmi, R.Sattanathan, ST decomposition method for solving FFLS” in International Journal of Ultra Scientists of Physical sciences volume 22 Number 3(M) Aug 2011, ISSN: 0970-9150, 747-750 impact factor .045
11. V Vijayalakshmi, Dr. R. Sattanathan, “ST decomposition method for solving FFLS using Gauss Jordan method for Trapezoidal fuzzy numbers” in International Mathematical Forum, Bulgaria, Europe Vol6, 2011 No.45 ISSN: 1312-7594, 2245-2254.
12. V Vijayalakshmi, “Solving FFLSE for Octagonal fuzzy matrices” in International Journal of Pure and Applied Mathematics, Vol 109,2018, ISSN: 1811-8080, 267-275.

## AUTHORS PROFILE



**Dr. V. Vijayalakshmi** is currently working as a Assistant Professor in the Department of Mathematics, SRM Valliammai Engineering College(SRM group), kattankulathur, Chennai, India. She has completed M.Sc., M.Phil., PGDCA., PhD in various universities. She has 21 years of teaching and 10 years of research experience. She has cleared SET (State eligibility test) examination conducted by Government of

Tamil Nadu. She has more than 10 publications in reputed journals on her credit. Her area of research are fuzzy logic, fuzzy mathematics, fuzzy Matrices.



**Ms. S. Surabhi** is currently working as a Assistant Professor in the Department of Electronics and communication Engineering, SRM Valliammai Engineering College(SRM group), kattankulathur, Chennai, India. She has completed M.E,in Anna University M.I She has 5 years of teaching She has more than 10 publications in reputed journals on her credit. Her area of research medical electronics, fuzzy mathematics.



**Ms. A. Karpagam** is currently working as a Assistant Professor in the Department of Mathematics, SRM Valliammai Engineering College(SRM group), kattankulathur, Chennai, India. She has completed M.Sc., M.Phil.in Anna University M.I.T. She has 8 years of teaching and 5 years of research experience. She has cleared SET (State eligibility test) examination conducted by Government of Tamil Nadu. She has more than

10 publications in reputed journals on her credit. Her area of research are fuzzy logic, fuzzy mathematics, fuzzy Operation Research