J. Appl. Math. & Computing Vol. **25**(2007), No. 1 - 2, pp. 119 - 136
Website: http://jamc.net

BLOCK ITERATIVE METHODS FOR FUZZY LINEAR SYSTEMS

KE WANG* AND BING ZHENG

Abstract. Block Jacobi and Gauss-Seidel iterative methods are studied for solving $n \times n$ fuzzy linear systems. A new splitting method is considered as well. These methods are accompanied with some convergence theorems. Numerical examples are presented to illustrate the theory.

AMS Mathematics Subject Classification: 15A06, 65F10, 90C70.
Key words and phrases : Jacobi method, Gauss-Seidel method, fuzzy linear system.

1. Introduction

Many engineering problems, such as equilibrium and steady-state problems, a mechanism using the kinetostatic approach, require the solution of simultaneous algebraic linear equations. However, many real-world engineering systems are too complex to be defined in precise terms, therefore, imprecision is often involved in many engineering design process. Fuzzy systems, which can formulate uncertainty in actual environment, play an essential role in such cases [6, 8–10], and lots of modeling techniques, control problems, and operations-research algorithms have been designed for them since the concept of *fuzzy number* and arithmetic operations with these numbers are first introduced and investigated by Zadeh [14, 15].

One of the major applications using fuzzy number arithmetic is treating systems of simultaneous linear equations whose parameters are all or partially represented by fuzzy numbers. For example, Rao and Chen [11] consider the following system of fuzzy linear equations in engineering analysis:

$$\mathbf{AX} = \mathbf{B}, \tag{1.1}$$

Received August 9, 2006. Revised June 16, 2007. *Corresponding author. This author was supported by China Postdoctoral Science Foundation (No. 20060400634).

The second author is supported by the start-up fund of Lanzhou University and the Natural Science Foundation of Gansu Province (No. 3ZS051-A25-020), P.R. China.

where $\mathbf{X} = (x_1, x_2, \cdots, x_n)^{\mathrm{T}}$ is a fuzzy vector which satisfies equation (1.1), and $\mathbf{A} = (a_{ij})_{n\times n}$ $(i, j = 1, 2, \cdots, n)$ and $\mathbf{B} = (b_1, b_2, \cdots, b_n)^{\mathrm{T}}$ denote the input fuzzy coefficient matrix and fuzzy right-hand-side vector, respectively. They provide a computational method to solve the fuzzy linear system (1.1). This kind of fuzzy linear system also arises in economics and finance as Leontief's input-output model, see [6].

As a special instance of system (1.1), Friedman et al [8] consider an $n \times n$ fuzzy linear system, whose coefficient matrix is crisp and the right-hand side column is an arbitrary fuzzy number vector. By the embedding approach given in [13], the authors replace the original $n \times n$ fuzzy linear system by a $2n \times 2n$ crisp function linear system, i.e., solving the $n \times n$ fuzzy linear system is equal to solving a $2n \times 2n$ crisp function linear system. In general, the $2n \times 2n$ crisp function linear system is large and sparse, see [8]. As is well-known, iterative methods play important roles in solving such crisp function linear systems. Therefore, iterative methods for solving such fuzzy linear systems have been investigated by many authors [1–3,5,7,12], which include classic point iterative methods (such as Jacobi, Richardson, Gauss-Seidel, SOR, SSOR, ESOR, MSOR, AOR etc.), conjugate gradient method, steepest descent method and LU decomposition and Adomian decomposition methods.

In this paper, we consider block iterative methods for $n \times n$ fuzzy linear systems presented in [8], which are efficient and practical because these methods only require the nonsingularity of the coefficient matrix of a fuzzy linear system while point iterative methods mentioned above require the diagonal entries of the coefficient matrix are nonzero. We mainly provide block Jacobi, Gauss-Seidel methods and a new splitting method. By the new method, the augmented system for the original fuzzy linear system falls into two independent subsystems, thus the new method is suitable for solving large-scale fuzzy linear systems in parallel computing environment.

In Section 2 we recall preliminaries for $n \times n$ fuzzy linear systems. The block iterative methods are discussed in Section 3. Numerical examples are given to illustrate our theory in Section 4 and conclusion in Section 5.

2. Preliminaries

Following [13], an arbitrary fuzzy number is represented, in parametric form, by an ordered pair of functions $(\underline{u}(r), \overline{u}(r))$, $0 \leqslant r \leqslant 1$, which satisfy the following requirements:

1. $\underline{u}(r)$ is a bounded left continuous nondecreasing function over $[0, 1]$;
2. $\overline{u}(r)$ is a bounded left continuous nonincreasing function over $[0, 1]$;
3. $\underline{u}(r) \leqslant \overline{u}(r)$, $0 \leqslant r \leqslant 1$.

A crisp number α can be simply expressed as $\underline{u}(r) = \overline{u}(r) = \alpha$, $0 \leqslant r \leqslant 1$.

The addition and scalar multiplication of fuzzy numbers previously defined can be described as follows, for arbitrary $u = (\underline{u}(r), \overline{u}(r))$, $v = (\underline{v}(r), \overline{v}(r))$ and real number λ,

(a) $u + v = (\underline{u}(r) + \underline{v}(r), \overline{u}(r) + \overline{v}(r))$;

(b) $\lambda u = \begin{cases} (\lambda\underline{u}(r), \lambda\overline{u}(r)), & \lambda \geqslant 0, \\ (\lambda\overline{u}(r), \lambda\underline{u}(r)), & \lambda < 0. \end{cases}$

Definition 2.1 [8]. The $n \times n$ linear system

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = y_1, \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = y_2, \\ \qquad\vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = y_n, \end{cases} \tag{2.1}$$

where the coefficient matrix $A = (a_{ij})$, $1 \leqslant i$, $j \leqslant n$ is a crisp matrix and y_i, $1 \leqslant i \leqslant n$ are fuzzy numbers, is called a *fuzzy linear system* (FLS).

Definition 2.2 [8]. A fuzzy number vector $X = (x_1, x_2, \cdots, x_n)^{\mathrm{T}}$ given by

$$x_i = (\underline{x}_i(r), \overline{x}_i(r)), \quad 1 \leqslant i \leqslant n,\ 0 \leqslant r \leqslant 1,$$

is called a *solution* of the fuzzy linear system (2.1) if

$$\begin{cases} \underline{\sum_{j=1}^{n} a_{ij}x_j} = \sum_{j=1}^{n} \underline{a_{ij}x_j} = \underline{y}_i, \\ \overline{\sum_{j=1}^{n} a_{ij}x_j} = \sum_{j=1}^{n} \overline{a_{ij}x_j} = \overline{y}_i. \end{cases} \quad i = 1, 2, \cdots, n. \tag{2.2}$$

Following Friedman et al [8], the system (2.1) can be extended to a $2n \times 2n$ crisp linear system

$$SX = Y \tag{2.3}$$

where

$$S = \begin{array}{c} n \\ n \end{array} \overset{\begin{array}{cc} n & n \end{array}}{\begin{bmatrix} S_1 \geqslant 0 & S_2 \geqslant 0 \\ S_2 \geqslant 0 & S_1 \geqslant 0 \end{bmatrix}},$$

and

$$X = \begin{bmatrix} \underline{X} \\ -\overline{X} \end{bmatrix}, \; Y = \begin{bmatrix} \underline{Y} \\ -\overline{Y} \end{bmatrix},$$

where

$$\underline{X} = \begin{bmatrix} \underline{x}_1 \\ \underline{x}_2 \\ \vdots \\ \underline{x}_n \end{bmatrix}, \; \overline{X} = \begin{bmatrix} \overline{x}_1 \\ \overline{x}_2 \\ \vdots \\ \overline{x}_n \end{bmatrix}, \; \underline{Y} = \begin{bmatrix} \underline{y}_1 \\ \underline{y}_2 \\ \vdots \\ \underline{y}_n \end{bmatrix}, \; \overline{Y} = \begin{bmatrix} \overline{y}_1 \\ \overline{y}_2 \\ \vdots \\ \overline{y}_n \end{bmatrix}.$$

The entries s_{kl} of S are determined as follows

$$\begin{array}{llll} a_{ij} \geqslant 0 & \Rightarrow & s_{ij} = a_{ij}, & s_{i+n,\, j+n} = a_{ij}, \\ a_{ij} < 0 & \Rightarrow & s_{i,\, j+n} = -a_{ij}, & s_{i+n,\, j} = -a_{ij}, \end{array} \quad 1 \leqslant i, j \leqslant n,$$

and any s_{kl} which is not determined by the above items is zero, $1 \leqslant k,\ l \leqslant 2n$.

The following theorem implies when FLS (2.1) has a unique solution.

Theorem 2.1 [8]**.** *The matrix S is nonsingular if and only if the matrices $A = S_1 - S_2$ and $S_1 + S_2$ are both nonsingular.*

Under the conditions of Theorem 2.1, the solution vector of (2.1)

$$X = S^{-1}Y \tag{2.4}$$

is thus unique but may still not be an *appropriate fuzzy vector*. By Theorem 2 of [8], we know that S^{-1} has the same structure as S, i.e.

$$S^{-1} = \begin{bmatrix} T_1 & T_2 \\ T_2 & T_1 \end{bmatrix}.$$

The following result provides a sufficient condition for the unique solution to be a fuzzy vector.

Theorem 2.2 [4]**.** *The unique solution X of* (2.4) *is a fuzzy vector for arbitrary fuzzy vector Y if S^{-1} is nonnegative.*

Restricting the discussion to triangular fuzzy numbers, i.e. $\underline{y}_i(r), \overline{y}_i(r)$ and consequently $\underline{x}_i(r), \overline{x}_i(r)$ are all linear functions of r, and having calculated X which solves (2.3), we can define the fuzzy solution to the original system given by (2.1) as follows.

Definition 2.3. Let $X = \{(\underline{x}_i(r), -\overline{x}_i(r)),\ 1 \leqslant i \leqslant n\}$ denote the unique solution of (2.3). The fuzzy vector $U = \{(\underline{u}_i(r), \overline{u}_i(r)),\ 1 \leqslant i \leqslant n\}$ defined by

$$\begin{aligned} \underline{u}_i(r) &= \min\left\{\underline{x}_i(r), \overline{x}_i(r), \underline{x}_i(1), \overline{x}_i(1)\right\}, \\ \overline{u}_i(r) &= \max\left\{\underline{x}_i(r), \overline{x}_i(r), \underline{x}_i(1), \overline{x}_i(1)\right\} \end{aligned}$$

is called the *fuzzy solution* of $SX = Y$. If $(\underline{x}_i(r), \overline{x}_i(r))$, $1 \leqslant i \leqslant n$ are all fuzzy numbers, then $\underline{u}_i(r) = \underline{x}_i(r)$, $\overline{u}_i(r) = \overline{x}_i(r)$, $1 \leqslant i \leqslant n$ and U is called a *strong fuzzy solution.* Otherwise, U is called a *weak fuzzy solution.*

3. Block Iterative Methods for FLS

For nonsingular system (2.3), where S is nonsingular, that is $A = S_1 - S_2$ and $S_1 + S_2$ are nonsingular, we can use the following splitting,

$$S = D - L - U,$$

where

$$\begin{aligned} D &= \begin{bmatrix} S_1 - S_2 & 0 \\ 0 & S_1 - S_2 \end{bmatrix}, \\ L &= \begin{bmatrix} 0 & 0 \\ -S_2 & 0 \end{bmatrix},\ U = \begin{bmatrix} -S_2 & -S_2 \\ 0 & -S_2 \end{bmatrix}, \end{aligned} \tag{3.1}$$

or

$$\begin{aligned} D &= \begin{bmatrix} S_1 + S_2 & 0 \\ 0 & S_1 + S_2 \end{bmatrix}, \\ L &= \begin{bmatrix} 0 & 0 \\ -S_2 & 0 \end{bmatrix},\ U = \begin{bmatrix} S_2 & -S_2 \\ 0 & S_2 \end{bmatrix}, \end{aligned} \tag{3.2}$$

or

$$\begin{aligned} D &= \begin{bmatrix} S_1 - S_2 & 0 \\ 0 & S_1 + S_2 \end{bmatrix}, \\ L &= \begin{bmatrix} 0 & 0 \\ -S_2 & 0 \end{bmatrix},\ U = \begin{bmatrix} -S_2 & -S_2 \\ 0 & S_2 \end{bmatrix}, \end{aligned} \tag{3.3}$$

or

$$\begin{aligned} D &= \begin{bmatrix} S_1 + S_2 & 0 \\ 0 & S_1 - S_2 \end{bmatrix}, \\ L &= \begin{bmatrix} 0 & 0 \\ -S_2 & 0 \end{bmatrix},\ U = \begin{bmatrix} S_2 & -S_2 \\ 0 & -S_2 \end{bmatrix}, \end{aligned} \tag{3.4}$$

or

$$\begin{aligned} D &= \begin{bmatrix} S_1 - S_2 & 0 \\ 0 & S_1 - S_2 \end{bmatrix}, \\ L &= \begin{bmatrix} 0 & 0 \\ -S_2 & -S_2 \end{bmatrix},\ U = \begin{bmatrix} -S_2 & -S_2 \\ 0 & 0 \end{bmatrix}, \end{aligned} \tag{3.5}$$

or

$$\begin{aligned} D &= \begin{bmatrix} S_1 + S_2 & 0 \\ 0 & S_1 + S_2 \end{bmatrix}, \\ L &= \begin{bmatrix} 0 & 0 \\ -S_2 & S_2 \end{bmatrix},\ U = \begin{bmatrix} S_2 & -S_2 \\ 0 & 0 \end{bmatrix}, \end{aligned} \tag{3.6}$$

or

$$\begin{aligned} D &= \begin{bmatrix} S_1 - S_2 & 0 \\ 0 & S_1 - S_2 \end{bmatrix}, \\ L &= \begin{bmatrix} -S_2 & 0 \\ -S_2 & 0 \end{bmatrix},\ U = \begin{bmatrix} 0 & -S_2 \\ 0 & -S_2 \end{bmatrix}, \end{aligned} \tag{3.7}$$

or

$$
\begin{aligned}
D &= \begin{bmatrix} S_1+S_2 & 0 \\ 0 & S_1+S_2 \end{bmatrix}, \\
L &= \begin{bmatrix} S_2 & 0 \\ -S_2 & 0 \end{bmatrix}, \; U = \begin{bmatrix} 0 & -S_2 \\ 0 & S_2 \end{bmatrix}.
\end{aligned} \tag{3.8}
$$

If S_1 is nonsingular, we may use splitting

$$
\begin{aligned}
S &= \begin{bmatrix} S_1 & 0 \\ 0 & S_1 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ -S_2 & 0 \end{bmatrix} - \begin{bmatrix} 0 & -S_2 \\ 0 & 0 \end{bmatrix} \\
&\equiv D - L - U.
\end{aligned} \tag{3.9}
$$

3.1. Block Jacobi Methods

From splittings (3.1)-(3.9), we can get several different block Jacobi iterative schemes as below.

(1) For splittings (3.1), (3.5) and (3.7), we have scheme

$$X_{k+1} = H_J X_k + D^{-1} Y, \quad k = 0, 1, \cdots , \tag{3.10}$$

where $X_k = \begin{bmatrix} \underline{X_k} \\ -\overline{X}_k \end{bmatrix}$ and

$$
\begin{aligned}
H_J &= D^{-1}(L+U) \\
&= \begin{bmatrix} -(S_1-S_2)^{-1}S_2 & -(S_1-S_2)^{-1}S_2 \\ -(S_1-S_2)^{-1}S_2 & -(S_1-S_2)^{-1}S_2 \end{bmatrix}.
\end{aligned}
$$

(2) For splittings (3.2), (3.6) and (3.8), we have scheme

$$X_{k+1} = H_J X_k + D^{-1} Y, \quad k = 0, 1, \cdots , \tag{3.11}$$

where $X_k = \begin{bmatrix} \underline{X_k} \\ -\overline{X}_k \end{bmatrix}$ and

$$
\begin{aligned}
H_J &= D^{-1}(L+U) \\
&= \begin{bmatrix} (S_1+S_2)^{-1}S_2 & -(S_1+S_2)^{-1}S_2 \\ -(S_1+S_2)^{-1}S_2 & (S_1+S_2)^{-1}S_2 \end{bmatrix}.
\end{aligned}
$$

(3) For splitting (3.3), the scheme is

$$X_{k+1} = H_J X_k + D^{-1} Y, \quad k = 0, 1, \cdots , \tag{3.12}$$

where $X_k = \begin{bmatrix} \underline{X_k} \\ -\overline{X}_k \end{bmatrix}$ and

$$
\begin{aligned}
H_J &= D^{-1}(L+U) \\
&= \begin{bmatrix} -(S_1-S_2)^{-1}S_2 & -(S_1-S_2)^{-1}S_2 \\ -(S_1+S_2)^{-1}S_2 & (S_1+S_2)^{-1}S_2 \end{bmatrix}.
\end{aligned}
$$

(4) For splitting (3.4), the scheme is

$$X_{k+1} = H_J X_k + D^{-1} Y, \quad k = 0, 1, \cdots, \tag{3.13}$$

where $X_k = \begin{bmatrix} \underline{X}_k \\ -\overline{X}_k \end{bmatrix}$ and

$$\begin{aligned} H_J &= D^{-1}(L+U) \\ &= \begin{bmatrix} (S_1+S_2)^{-1}S_2 & -(S_1+S_2)^{-1}S_2 \\ -(S_1-S_2)^{-1}S_2 & -(S_1-S_2)^{-1}S_2 \end{bmatrix}. \end{aligned}$$

(5) For splitting (3.9), the scheme is

$$X_{k+1} = H_J X_k + D^{-1} Y, \quad k = 0, 1, \cdots, \tag{3.14}$$

where $X_k = \begin{bmatrix} \underline{X}_k \\ -\overline{X}_k \end{bmatrix}$ and

$$\begin{aligned} H_J &= D^{-1}(L+U) \\ &= \begin{bmatrix} 0 & -S_1^{-1}S_2 \\ -S_1^{-1}S_2 & 0 \end{bmatrix}. \end{aligned}$$

For the above iterative schemes, we have the following convergence results.

Theorem 3.1. *The block Jacobi iteration* (3.10) *is convergent if and only if* $\rho((S_1-S_2)^{-1}S_2) < \frac{1}{2}$.

Proof. Let $\lambda \in \lambda(H_J)$. Then

$$\begin{aligned} \det(\lambda I - H_J) &= \det\left(\begin{bmatrix} \lambda I + (S_1-S_2)^{-1}S_2 & (S_1-S_2)^{-1}S_2 \\ (S_1-S_2)^{-1}S_2 & \lambda I + (S_1-S_2)^{-1}S_2 \end{bmatrix}\right) \\ &= \lambda^n \cdot \det(\lambda I + 2(S_1-S_2)^{-1}S_2). \end{aligned}$$

Thus,

$$\rho(H_J) = \rho(-2(S_1-S_2)^{-1}S_2) = 2\rho((S_1-S_2)^{-1}S_2),$$

which implies $\rho(H_J) < 1$ if and only if $\rho((S_1-S_2)^{-1}S_2) < \frac{1}{2}$. □

Similarly, we can obtain the next theorem.

Theorem 3.2. *Iteration* (3.11) *converges if and only if* $\rho((S_1+S_2)^{-1}S_2) < \frac{1}{2}$.

Similar to Theorem 3.1 and 3.2, we have

Theorem 3.3. *If* S_1 *is nonsingular, then the iteration* (3.14) *is convergent if and only if* $\rho(S_1^{-1}S_2) < 1$.

Remark 3.1. For schemes (3.12) and (3.13), as we know, the corresponding iterations are convergent if and only if $\rho(H_J) < 1$.

3.2. Block Gauss-Seidel methods

Applying the classic Gauss-Seidel method to splittings (3.1)-(3.4), respectively, we obtain four block Gauss-Seidel iterative schemes.

(1) For splitting (3.1), the scheme is

$$\mathcal{X}_{k+1} = H_{GS}\mathcal{X}_k + (D-L)^{-1}\mathcal{Y}, \quad k = 0, 1, \cdots, \tag{3.15}$$

where $\mathcal{X}_k = \begin{bmatrix} X_k \\ -\overline{X}_k \end{bmatrix}$ and

$$\begin{aligned} H_{GS} &= (D-L)^{-1}U \\ &= \begin{bmatrix} -(S_1-S_2)^{-1}S_2 & -(S_1-S_2)^{-1}S_2 \\ [(S_1-S_2)^{-1}S_2]^2 & [(S_1-S_2)^{-1}S_2]^2 - (S_1-S_2)^{-1}S_2 \end{bmatrix}. \end{aligned}$$

(2) For splitting (3.2), the scheme is

$$\mathcal{X}_{k+1} = H_{GS}\mathcal{X}_k + (D-L)^{-1}\mathcal{Y}, \quad k = 0, 1, \cdots, \tag{3.16}$$

where $\mathcal{X}_k = \begin{bmatrix} X_k \\ -\overline{X}_k \end{bmatrix}$ and

$$\begin{aligned} H_{GS} &= (D-L)^{-1}U \\ &= \begin{bmatrix} (S_1+S_2)^{-1}S_2 & -(S_1+S_2)^{-1}S_2 \\ -[(S_1+S_2)^{-1}S_2]^2 & [(S_1+S_2)^{-1}S_2]^2 + (S_1+S_2)^{-1}S_2 \end{bmatrix}. \end{aligned}$$

(3) For splitting (3.3), the scheme is

$$\mathcal{X}_{k+1} = H_{GS}\mathcal{X}_k + (D-L)^{-1}\mathcal{Y}, \quad k = 0, 1, \cdots, \tag{3.17}$$

where $\mathcal{X}_k = \begin{bmatrix} X_k \\ -\overline{X}_k \end{bmatrix}$ and

$$\begin{aligned} H_{GS} &= (D-L)^{-1}U \\ &= \begin{bmatrix} -(S_1-S_2)^{-1}S_2 & -(S_1-S_2)^{-1}S_2 \\ \\ (S_1+S_2)^{-1}S_2 & (S_1+S_2)^{-1}S_2 \\ \cdot(S_1-S_2)^{-1}S_2 & \cdot[I + (S_1-S_2)^{-1}S_2] \end{bmatrix}. \end{aligned}$$

(4) For splitting (3.4), the scheme is

$$\mathcal{X}_{k+1} = H_{GS}\mathcal{X}_k + (D-L)^{-1}\mathcal{Y}, \quad k = 0, 1, \cdots, \tag{3.18}$$

where $\mathcal{X}_k = \begin{bmatrix} X_k \\ -\overline{X}_k \end{bmatrix}$ and

$$\begin{aligned} H_{GS} &= (D-L)^{-1}U \\ &= \begin{bmatrix} (S_1+S_2)^{-1}S_2 & -(S_1+S_2)^{-1}S_2 \\ \\ -(S_1-S_2)^{-1}S_2 & -(S_1-S_2)^{-1}S_2 \\ \cdot(S_1+S_2)^{-1}S_2 & \cdot[I - (S_1+S_2)^{-1}S_2] \end{bmatrix}. \end{aligned}$$

If S_1 is nonsingular, from splittings (3.5)-(3.9), we have the following five Gauss-Seidel iterations.

(5) For splitting (3.5), the scheme is

$$X_{k+1} = H_{GS}X_k + (D-L)^{-1}Y, \quad k = 0, 1, \cdots, \tag{3.19}$$

where $X_k = \begin{bmatrix} \underline{X}_k \\ -\overline{X}_k \end{bmatrix}$ and

$$\begin{aligned} H_{GS} &= (D-L)^{-1}U \\ &= \begin{bmatrix} -(S_1-S_2)^{-1}S_2 & -(S_1-S_2)^{-1}S_2 \\ S_1^{-1}S_2(S_1-S_2)^{-1}S_2 & S_1^{-1}S_2(S_1-S_2)^{-1}S_2 \end{bmatrix}. \end{aligned}$$

(6) For splitting (3.6), the scheme is

$$X_{k+1} = H_{GS}X_k + (D-L)^{-1}Y, \quad k = 0, 1, \cdots, \tag{3.20}$$

where $X_k = \begin{bmatrix} \underline{X}_k \\ -\overline{X}_k \end{bmatrix}$ and

$$\begin{aligned} H_{GS} &= (D-L)^{-1}U \\ &= \begin{bmatrix} (S_1+S_2)^{-1}S_2 & -(S_1+S_2)^{-1}S_2 \\ -S_1^{-1}S_2(S_1+S_2)^{-1}S_2 & S_1^{-1}S_2(S_1+S_2)^{-1}S_2 \end{bmatrix}. \end{aligned}$$

(7) For splitting (3.7), the scheme is

$$X_{k+1} = H_{GS}X_k + (D-L)^{-1}Y, \quad k = 0, 1, \cdots, \tag{3.21}$$

where $X_k = \begin{bmatrix} \underline{X}_k \\ -\overline{X}_k \end{bmatrix}$ and

$$\begin{aligned} H_{GS} &= (D-L)^{-1}U \\ &= \begin{bmatrix} 0 & -S_1^{-1}S_2 \\ 0 & -S_1^{-1}S_2 \end{bmatrix}. \end{aligned}$$

(8) For splitting (3.8), the scheme is

$$X_{k+1} = H_{GS}X_k + (D-L)^{-1}Y, \quad k = 0, 1, \cdots, \tag{3.22}$$

where $X_k = \begin{bmatrix} \underline{X}_k \\ -\overline{X}_k \end{bmatrix}$ and

$$\begin{aligned} H_{GS} &= (D-L)^{-1}U \\ &= \begin{bmatrix} 0 & -S_1^{-1}S_2 \\ 0 & S_1^{-1}S_2 \end{bmatrix}. \end{aligned}$$

(9) For splitting (3.9), the scheme is

$$X_{k+1} = H_{GS}X_k + (D-L)^{-1}Y, \quad k = 0, 1, \cdots, \tag{3.23}$$

where $\mathcal{X}_k = \begin{bmatrix} X_k \\ -\overline{X}_k \end{bmatrix}$ and

$$\begin{aligned} H_{GS} &= (D-L)^{-1}U \\ &= \begin{bmatrix} 0 & -S_1^{-1}S_2 \\ 0 & (S_1^{-1}S_2)^2 \end{bmatrix}. \end{aligned}$$

Remark 3.2. For schemes (3.15)-(3.18), as we know, the corresponding iterations are convergent if and only if $\rho(H_{GS}) < 1$.

For schemes (3.19)-(3.23), the following result is obtained.

Theorem 3.4. *If S_1 is nonsingular, then the block Gauss-Seidel iterations* (3.19)-(3.23) *are convergent if and only if* $\rho(S_1^{-1}S_2) < 1$.

Proof. For (3.19), let $\lambda \in \lambda(H_{GS})$, then

$$\begin{aligned} \det(\lambda I - H_{GS}) &= \det\left(\begin{bmatrix} \lambda I + (S_1-S_2)^{-1}S_2 & (S_1-S_2)^{-1}S_2 \\ -S_1^{-1}S_2(S_1-S_2)^{-1}S_2 & \lambda I - S_1^{-1}S_2(S_1-S_2)^{-1}S_2 \end{bmatrix}\right) \\ &= \lambda^n \cdot \det(\lambda I + S_1^{-1}S_2). \end{aligned}$$

Thus, $\rho(H_{GS}) = \rho(S_1^{-1}S_2)$, i.e., $\rho(H_{GS}) < 1$ if and only if $\rho(S_1^{-1}S_2) < 1$.
The same applies to (3.20)-(3.23). □

3.3. A new splitting method

If S_1 is nonsingular and $\rho(S_1^{-1}S_2) < 1$, we consider the following splitting:

$$S = M - N, \tag{3.24}$$

where

$$\begin{aligned} M &= \begin{bmatrix} S_1[I-(S_1^{-1}S_2)^2]^{-1} & S_2[I-(S_1^{-1}S_2)^2]^{-1} \\ S_2[I-(S_1^{-1}S_2)^2]^{-1} & S_1[I-(S_1^{-1}S_2)^2]^{-1} \end{bmatrix}, \\ N &= \begin{bmatrix} S_1[I-(S_1^{-1}S_2)^2]^{-1}(S_1^{-1}S_2)^2 & S_2[I-(S_1^{-1}S_2)^2]^{-1}(S_1^{-1}S_2)^2 \\ S_2[I-(S_1^{-1}S_2)^2]^{-1}(S_1^{-1}S_2)^2 & S_1[I-(S_1^{-1}S_2)^2]^{-1}(S_1^{-1}S_2)^2 \end{bmatrix}. \end{aligned}$$

It can be verified that M is invertible and

$$M^{-1} = \begin{bmatrix} S_1^{-1} & -S_1^{-1}S_2S_1^{-1} \\ -S_1^{-1}S_2S_1^{-1} & S_1^{-1} \end{bmatrix}.$$

As is well known, system (2.3) with splitting (3.24) is equal to

$$\mathcal{X} = M^{-1}N\mathcal{X} + M^{-1}\mathcal{Y},$$

which induces iterative scheme

$$\mathcal{X}_{k+1} = M^{-1}N\mathcal{X}_k + M^{-1}\mathcal{Y},$$

that is,

$$\left\{\begin{array}{l} \underline{X}_{k+1} = (S_1^{-1}S_2)^2\underline{X}_k + S_1^{-1}S_2S_1^{-1}\overline{Y} + S_1^{-1}\underline{Y}, \\ \overline{X}_{k+1} = (S_1^{-1}S_2)^2\overline{X}_k + S_1^{-1}S_2S_1^{-1}\underline{Y} + S_1^{-1}\overline{Y}, \end{array}\right. \tag{3.25}$$

viz.

$$X_{k+1} = HX_k + BY, \quad k = 0, 1, \cdots, \tag{3.26}$$

where $X_k = \begin{bmatrix} \underline{X}_k \\ -\overline{X}_k \end{bmatrix}$,

$$H = M^{-1}N = \begin{bmatrix} (S_1^{-1}S_2)^2 & 0 \\ 0 & (S_1^{-1}S_2)^2 \end{bmatrix},$$

and

$$B = M^{-1} = \begin{bmatrix} S_1^{-1} & -S_1^{-1}S_2S_1^{-1} \\ -S_1^{-1}S_2S_1^{-1} & S_1^{-1} \end{bmatrix}.$$

For the new iterative method, we have the convergence result:

Theorem 3.5. *If S_1 is nonsingular, then the block iteration* (3.25) *or* (3.26) *is convergent if and only if* $\rho(S_1^{-1}S_2) < 1$.

Proof. Let $\lambda \in \lambda(H)$. Then

$$\begin{aligned} \det(\lambda I - H) &= \det\left(\begin{bmatrix} \lambda I - (S_1^{-1}S_2)^2 & 0 \\ 0 & \lambda I - (S_1^{-1}S_2)^2 \end{bmatrix}\right) \\ &= \det(\lambda I - (S_1^{-1}S_2)^2) \cdot \det(\lambda I - (S_1^{-1}S_2)^2), \end{aligned}$$

i.e., $\rho(H) = (\rho(S_1^{-1}S_2))^2$. Thus, $\rho(H) < 1$ if and only if $\rho(S_1^{-1}S_2) < 1$. □

Remark 3.3. Note that (3.25) implies that we can compute $\underline{X}$ and $\overline{X}$ independently, thus, the new method is suitable for computing in parallel environment.

4. Numerical examples

In this section, we will give some numerical examples to illustrate the methods presented in this paper. For this purpose, we define a stopping criterion with tolerance $\varepsilon > 0$ as follows

$$\|X_{k+1} - X_k\| \leqslant \varepsilon.$$

As we know, any other type of fuzzy number can be approximated by triangular fuzzy number. Hence the triangular fuzzy numbers are used in all of the following numerical examples. For a triangular fuzzy number $x = (a+br, c+dr)$, we define its norm as

$$\|x\| = \max\{\, |a|, |b|, |c|, |d| \,\},$$

which is equivalent to Hausdorff distance of fuzzy numbers. Thus the norm of

$$X = \left[\begin{array}{c} \underline{X} \\ -\overline{X} \end{array} \right] = \left[\begin{array}{c} \underline{x}_1 \\ \vdots \\ \underline{x}_n \\ -\overline{x}_1 \\ \vdots \\ -\overline{x}_n \end{array} \right] = \left[\begin{array}{c} x_{1a} + x_{1b}r \\ x_{2a} + x_{2b}r \\ \vdots \\ x_{2n,a} + x_{2n,b}r \end{array} \right],$$

where x_{ia} and x_{ib} are crisp numbers, $i = 1, \cdots, 2n$, $0 \leqslant r \leqslant 1$, can be defined as

$$\|X\| = \max_i \{|x_{ia}|, |x_{ib}|\}. \tag{$*$}$$

We use the following examples to illustrate our theory. All runs are performed in MATLAB.

Example 4.1. Consider the 25×25 fuzzy linear system

$$\left\{ \begin{array}{ccccccc} x_1 & + & 2x_2 & - & x_3 & = & (r, 2-r), \\ x_2 & + & 2x_3 & - & x_4 & = & (r, 2-r), \\ \vdots & & & & & & \\ x_{24} & + & 2x_{25} & - & x_1 & = & (r, 2-r), \\ x_{25} & + & 2x_1 & - & x_2 & = & (r, 2-r). \end{array} \right.$$

The extended 50×50 matrix is

$$S = \left[\begin{array}{cc} S_1 & S_2 \\ S_2 & S_1 \end{array} \right]$$

$$= \left[\begin{array}{ccccccc|ccccccc}
1 & 2 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 1 & \cdots & \cdots & 0 & 0 \\
0 & 1 & 2 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & \ddots & \cdots & 0 & 0 \\
0 & 0 & 1 & \ddots & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & \ddots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & 1 & 2 & 0 & 0 & 0 & 0 & \cdots & \cdots & 0 & 1 \\
0 & 0 & 0 & \cdots & 0 & 1 & 2 & 1 & 0 & 0 & \cdots & \cdots & 0 & 0 \\
2 & 0 & 0 & \cdots & 0 & 0 & 1 & 0 & 1 & 0 & \cdots & \cdots & 0 & 0 \\
\hline
0 & 0 & 1 & \cdots & \cdots & 0 & 0 & 1 & 2 & 0 & \cdots & 0 & 0 & 0 \\
0 & 0 & 0 & \ddots & \cdots & 0 & 0 & 0 & 1 & 2 & \cdots & 0 & 0 & 0 \\
0 & 0 & 0 & \cdots & \ddots & 0 & 0 & 0 & 0 & 1 & \ddots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & \cdots & 0 & 1 & 0 & 0 & 0 & \cdots & 1 & 2 & 0 \\
1 & 0 & 0 & \cdots & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 2 \\
0 & 1 & 0 & \cdots & \cdots & 0 & 0 & 2 & 0 & 0 & \cdots & 0 & 0 & 1
\end{array} \right],$$

which is nonsingular. S_1 is also nonsingular, therefore, the exact solution is

$$x_1 = \cdots = x_{25} = (1/4 + r/4, 3/4 - r/4),$$

which is a strong fuzzy solution.

Applying the iterative schemes provided in Section 3 and the point iterative methods in [4] on this system with $X_0 = 0$, we have the numerical results in Table 1.

Example 4.2. Consider the 3×3 fuzzy linear system

$$\begin{cases} 3x_1 - x_3 = (1 + r, 3 - r), \\ x_1 + x_2 + x_3 = (r, 2 - r), \\ -x_2 + 2x_3 = (-3, -2 - r). \end{cases}$$

The extended 6×6 matrix is

$$S = \begin{bmatrix} S_1 & S_2 \\ S_2 & S_1 \end{bmatrix} = \left[\begin{array}{ccc|ccc} 3 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 1 & 0 \\ \hline 0 & 0 & 1 & 3 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 2 \end{array}\right], \text{ which is nonsingular,}$$

where $S_1 = \begin{bmatrix} 3 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 2 \end{bmatrix}$ is nonsingular as well, and the exact solution is

$$\begin{cases} x_1 = (\underline{x}_1(r), \overline{x}_1(r)) = (0.075 + 0.325r, 0.825 - 0.425r), \\ x_2 = (\underline{x}_2(r), \overline{x}_2(r)) = (0.45 + 0.95r, 1.95 - 0.55r), \\ x_3 = (\underline{x}_3(r), \overline{x}_3(r)) = (-0.525 - 0.275r, -0.775 - 0.025r), \end{cases}$$

which is a weak fuzzy solution. In fact x_3 is not a fuzzy number. Therefore the fuzzy solution is

$$\begin{cases} u_1 = (\underline{u}_1(r), \overline{u}_1(r)) = (0.075 + 0.325r, 0.825 - 0.425r), \\ u_2 = (\underline{u}_2(r), \overline{u}_2(r)) = (0.45 + 0.95r, 1.95 - 0.55r), \\ u_3 = (\underline{u}_3(r), \overline{u}_3(r)) = (-0.8, -0.525 - 0.275r). \end{cases}$$

With $X_0 = 0$, we have the iteration results in Table 2.

Example 4.3. Consider the 3×3 fuzzy system

$$\begin{cases} 2x_1 + 3x_2 - x_3 = (-1 + 4r, 6 - 3r), \\ 3x_1 - x_2 + 2x_3 = (4 + 2r, 12 - 6r), \\ x_1 + 2x_2 + 3x_3 = (4 + 5r, 13 - 4r). \end{cases}$$

The extended 6×6 matrix is

$$S = \begin{bmatrix} S_1 & S_2 \\ S_2 & S_1 \end{bmatrix} = \left[\begin{array}{ccc|ccc} 2 & 3 & 0 & 0 & 0 & 1 \\ 3 & 0 & 2 & 0 & 1 & 0 \\ 1 & 2 & 3 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 2 & 3 & 0 \\ 0 & 1 & 0 & 3 & 0 & 2 \\ 0 & 0 & 0 & 1 & 2 & 3 \end{array}\right], \text{ which is invertible,}$$

where $S_1 = \begin{bmatrix} 2 & 3 & 0 \\ 3 & 0 & 2 \\ 1 & 2 & 3 \end{bmatrix}$ is also invertible. The exact solution is

$$\begin{cases} x_1 = (\underline{x}_1(r), \overline{x}_1(r)) = (1, 2-r), \\ x_2 = (\underline{x}_2(r), \overline{x}_2(r)) = (r, 1), \\ x_3 = (\underline{x}_3(r), \overline{x}_3(r)) = (1+r, 3-r). \end{cases}$$

which is a strong fuzzy solution.

With $X_0 = 0$, we have the iteration results in Table 3.

The following three tables show the numerical results of Examples 4.1-4.3, respectively, with the vector norm $(*)$.

Table 1.

scheme	ρ	iterations	approximate solution ($\varepsilon = 10^{-4}$)
Jacobi method			
(3.10)	1.0000	╲	╲
(3.11)	126.8183	╲	╲
(3.12)	63.8941	╲	╲
(3.13)	63.8941	╲	╲
(3.14)	0.9846	10	$x_1 = \cdots = x_{25} =$ $(0.2500 + 0.2500r, 0.7500 - 0.2500r)$
point method	3.0000	╲	╲
Gauss-Seidel			
(3.15)	0.9897	15	$x_1 = \cdots = x_{25} =$ $(0.2500 + 0.2500r, 0.7500 - 0.2500r)$
(3.16)	3.8939×10^3	╲	╲
(3.17)	32.4216	╲	╲
(3.18)	32.4216	╲	╲
(3.19)	0.9846	10	$x_1 = \cdots = x_{25} =$ $(0.2500 + 0.2500r, 0.7500 - 0.2500r)$
(3.20)	0.9846	9	$x_1 = \cdots = x_{25} =$ $(0.2500 + 0.2500r, 0.7500 - 0.2500r)$
(3.21)	0.9846	10	$x_1 = \cdots = x_{25} =$ $(0.2500 + 0.2500r, 0.7500 - 0.2500r)$
(3.22)	0.9846	9	$x_1 = \cdots = x_{25} =$ $(0.2500 + 0.2500r, 0.7500 - 0.2500r)$
(3.23)	0.9694	6	$x_1 = \cdots = x_{25} =$ $(0.2500 + 0.2500r, 0.7500 - 0.2500r)$
point method	4.2368	╲	╲
new method (3.26)	0.9694	6	$x_1 = \cdots = x_{25} =$ $(0.2500 + 0.2500r, 0.7500 - 0.2500r)$

ρ is the spectral radius of the iteration matrix.

╲ indicates the method is not convergent.

Table 2.

scheme	ρ	iterations	approximate solution
Jacobi method			
(3.10)	0.6325	15	$x_1 = (0.0753 + 0.3247r, 0.8247 - 0.4247r)$
$\varepsilon = 10^{-3}$			$x_2 = (0.4491 + 0.9509r, 1.9509 - 0.5509r)$
			$x_3 = (-0.5244 - 0.2756r, -0.7756 - 0.0244r)$
(3.11)	1.0000	\	\
(3.12)	0.6588	26	$x_1 = (0.0750 + 0.3250r, 0.8250 - 0.4250r)$
$\varepsilon = 10^{-4}$			$x_2 = (0.4500 + 0.9500r, 1.9500 - 0.5500r)$
			$x_3 = (-0.5250 - 0.2750r, -0.7750 - 0.0250r)$
(3.13)	0.6588	25	$x_1 = (0.0750 + 0.3250r, 0.8250 - 0.4250r)$
$\varepsilon = 10^{-4}$			$x_2 = (0.4500 + 0.9500r, 1.9500 - 0.5500r)$
			$x_3 = (-0.5250 - 0.2750r, -0.7750 - 0.0250r)$
(3.14)	0.4082	13	$x_1 = (0.0750 + 0.3250r, 0.8250 - 0.4250r)$
$\varepsilon = 10^{-4}$			$x_2 = (0.4500 + 0.9500r, 1.9500 - 0.5500r)$
			$x_3 = (-0.5250 - 0.2750r, -0.7750 - 0.0250r)$
point method	0.8362	39	$x_1 = (0.0749 + 0.3251r, 0.8251 - 0.4251r)$
$\varepsilon = 10^{-3}$			$x_2 = (0.4496 + 0.9504r, 1.9504 - 0.5504r)$
			$x_3 = (-0.5252 - 0.2748r, -0.7748 - 0.0252r)$
Gauss-Seidel			
(3.15)	0.5362	13	$x_1 = (0.0748 + 0.3251r, 0.8253 - 0.4251r)$
$\varepsilon = 10^{-3}$			$x_2 = (0.4496 + 0.9502r, 1.9502 - 0.5501r)$
			$x_3 = (-0.5244 - 0.2753r, -0.7755 - 0.0247r)$
(3.16)	0.7813	43	$x_1 = (0.0750 + 0.3250r, 0.8250 - 0.4250r)$
$\varepsilon = 10^{-4}$			$x_2 = (0.4500 + 0.9500r, 1.9500 - 0.5500r)$
			$x_3 = (-0.5250 - 0.2750r, -0.7750 - 0.0250r)$
(3.17)	0.4441	11	$x_1 = (0.0749 + 0.3250r, 0.8249 - 0.4249r)$
$\varepsilon = 10^{-3}$			$x_2 = (0.4501 + 0.9500r, 1.9502 - 0.5501r)$
			$x_3 = (-0.5250 - 0.2750r, -0.7751 - 0.0250r)$
(3.18)	0.4441	14	$x_1 = (0.0750 + 0.3250r, 0.8250 - 0.4250r)$
$\varepsilon = 10^{-4}$			$x_2 = (0.4500 + 0.9500r, 1.9500 - 0.5500r)$
			$x_3 = (-0.5250 - 0.2750r, -0.7750 - 0.0250r)$
(3.19)	0.4082	9	$x_1 = (0.0753 + 0.3247r, 0.8248 - 0.4248r)$
$\varepsilon = 10^{-3}$			$x_2 = (0.4502 + 0.9498r, 1.9501 - 0.5501r)$
			$x_3 = (-0.5255 - 0.2745r, -0.7749 - 0.0251r)$
(3.20)	0.4082	11	$x_1 = (0.0749 + 0.3250r, 0.8251 - 0.4250r)$
$\varepsilon = 10^{-3}$			$x_2 = (0.4499 + 0.9500r, 1.9500 - 0.5500r)$
			$x_3 = (-0.5248 - 0.2750r, -0.7751 - 0.0250r)$
(3.21)	0.4082	12	$x_1 = (0.0750 + 0.3250r, 0.8250 - 0.4250r)$
$\varepsilon = 10^{-4}$			$x_2 = (0.4500 + 0.9500r, 1.9500 - 0.5500r)$
			$x_3 = (-0.5249 - 0.2750r, -0.7751 - 0.0250r)$
(3.22)	0.4082	10	$x_1 = (0.0748 + 0.3251r, 0.8248 - 0.4249r)$
$\varepsilon = 10^{-3}$			$x_2 = (0.4502 + 0.9500r, 1.9502 - 0.5500r)$
			$x_3 = (-0.5250 - 0.2750r, -0.7750 - 0.0250r)$
(3.23)	0.1667	6	$x_1 = (0.0750 + 0.3250r, 0.8250 - 0.4250r)$
$\varepsilon = 10^{-3}$			$x_2 = (0.4499 + 0.9500r, 1.9500 - 0.5500r)$
			$x_3 = (-0.5249 - 0.2750r, -0.7751 - 0.0250r)$
point method	0.6907	29	$x_1 = (0.0750 + 0.3250r, 0.8250 - 0.4250r)$
$\varepsilon = 10^{-4}$			$x_2 = (0.4500 + 0.9500r, 1.9500 - 0.5500r)$
			$x_3 = (-0.5250 - 0.2750r, -0.7750 - 0.0250r)$
new method	0.1667	6	$x_1 = (0.0750 + 0.3250r, 0.8250 - 0.4250r)$
(3.26)			$x_2 = (0.4500 + 0.9500r, 1.9500 - 0.5500r)$
$\varepsilon = 10^{-3}$			$x_3 = (-0.5250 - 0.2750r, -0.7751 - 0.0250r)$

ρ is the spectral radius of the iteration matrix.

\ indicates the method is not convergent.

Table 3.

scheme	ρ	iterations	approximate solution ($\varepsilon = 10^{-4}$)
Jacobi method			
(3.10)	0.4593	13	$x_1 = (1.0000 - 0.0000r, 2.0000 - 1.0000r)$
			$x_2 = (-0.0000 + 1.0000r, 1.0000 - 0.0000r)$
			$x_3 = (1.0000 + 1.0000r, 3.0000 - 1.0000r)$
(3.11)	0.8495	67	$x_1 = (1.0000 - 0.0000r, 2.0000 - 1.0000r)$
			$x_2 = (-0.0000 + 1.0000r, 1.0000 - 0.0000r)$
			$x_3 = (1.0000 + 1.0000r, 3.0000 + 1.0000r)$
(3.12)	0.5499	20	$x_1 = (1.0000 + 0.0000r, 2.0000 - 1.0000r)$
			$x_2 = (0.0000 + 1.0000r, 1.0000 - 0.0000r)$
			$x_3 = (1.0000 + 1.0000r, 3.0000 - 1.0000r)$
(3.13)	0.5499	19	$x_1 = (1.0000 + 0.0000r, 2.0000 - 1.0000r)$
			$x_2 = (-0.0000 + 1.0000r, 1.0000 - 0.0000r)$
			$x_3 = (1.0000 + 1.0000r, 3.0000 - 1.0000r)$
(3.14)	0.2981	10	$x_1 = (1.0000 - 0.0000r, 2.0000 - 1.0000r)$
			$x_2 = (0.0000 + 1.0000r, 1.0000 - 0.0000r)$
			$x_3 = (1.0000 + 1.0000r, 3.0000 - 1.0000r)$
point method		can't be used directly	
Gauss-Seidel			
(3.15)	0.3692	11	$x_1 = (1.0000 - 0.0000r, 2.0000 - 1.0000r)$
			$x_2 = (-0.0000 + 1.0000r, 1.0000 - 0.0000r)$
			$x_3 = (1.0000 + 1.0000r, 3.0000 - 1.0000r)$
(3.16)	0.4248	14	$x_1 = (1.0000 + 0.0000r, 2.0000 - 1.0000r)$
			$x_2 = (-0.0000 + 1.0000r, 1.0000 + 0.0000r)$
			$x_3 = (1.0000 + 1.0000r, 3.0000 - 1.0000r)$
(3.17)	0.3649	12	$x_1 = (1.0000 + 0.0000r, 2.0000 - 1.0000r)$
			$x_2 = (0.0000 + 1.0000r, 1.0000 - 0.0000r)$
			$x_3 = (1.0000 + 1.0000r, 3.0000 - 1.0000r)$
(3.18)	0.3649	12	$x_1 = (1.0000 - 0.0000r, 2.0000 - 1.0000r)$
			$x_2 = (0.0000 + 1.0000r, 1.0000 - 0.0000r)$
			$x_3 = (1.0000 + 1.0000r, 3.0000 - 1.0000r)$
(3.19)	0.2981	9	$x_1 = (1.0000 - 0.0000r, 2.0000 - 1.0000r)$
			$x_2 = (-0.0000 + 1.0000r, 1.0000 - 0.0000r)$
			$x_3 = (1.0000 + 1.0000r, 3.0000 - 1.0000r)$
(3.20)	0.2981	11	$x_1 = (1.0000 - 0.0000r, 2.0000 - 1.0000r)$
			$x_2 = (-0.0000 + 1.0000r, 1.0000 - 0.0000r)$
			$x_3 = (1.0000 + 1.0000r, 3.0000 - 1.0000r)$
(3.21)	0.2981	10	$x_1 = (1.0000 - 0.0000r, 2.0000 - 1.0000r)$
			$x_2 = (-0.0000 + 1.0000r, 1.0000 - 0.0000r)$
			$x_3 = (1.0000 + 1.0000r, 3.0000 - 1.0000r)$
(3.22)	0.2981	10	$x_1 = (1.0000 + 0.0000r, 2.0000 - 1.0000r)$
			$x_2 = (0.0000 + 1.0000r, 1.0000 - 0.0000r)$
			$x_3 = (1.0000 + 1.0000r, 3.0000 - 1.0000r)$
(3.23)	0.0889	6	$x_1 = (1.0000 - 0.0000r, 2.0000 - 1.0000r)$
			$x_2 = (-0.0000 + 1.0000r, 1.0000 - 0.0000r)$
			$x_3 = (1.0000 + 1.0000r, 3.0000 - 1.0000r)$
point method		can't be used directly	
new method	0.0889	6	$x_1 = (1.0000 - 0.0000r, 2.0000 - 1.0000r)$
(3.26)			$x_2 = (0.0000 + 1.0000r, 1.0000 - 0.0000r)$
			$x_3 = (1.0000 + 1.0000r, 3.0000 - 1.0000r)$

ρ is the spectral radius of the iteration matrix.

5. Conclusion

In this paper we present some block iterative methods for solving $n \times n$ fuzzy linear systems and obtain the necessary and sufficient conditions for the convergence of some iterative schemes. For an $n \times n$ fuzzy linear system, if the extended matrix S by Friedman et al [8] is nonsingular, under the convergence conditions, then for any initial vector X_0, the iterations will converge to $X = \begin{bmatrix} \underline{X} \\ -\overline{X} \end{bmatrix}$, the unique solution of $SX = Y$. The methods are suitable for large systems, even the number of variables is quite high, and, theoretically, the number can be arbitrarily large. As mentioned in Section 2, a crisp number α can be simply expressed as $\underline{u}(r) = \overline{u}(r) = \alpha$, $0 \leqslant r \leqslant 1$, therefore, if there are no fuzzy sets but single numeric values, numeric values of solution will be provided. The numerical examples show that these methods are efficient and applicable for solving such fuzzy linear systems.

Acknowledgement

The authors would like to thank Professor Zengtai Gong, Department of Mathematics, Northwest Normal University, Lanzhou 730070, P.R. China, and the referees for their kind assistance.

References

1. S. Abbasbandy, R. Ezzati and A. Jafarian, *LU decomposition method for solving fuzzy system of linear equations*, Appl. Math. Comput. **172** (2006), 633-643.
2. S. Abbasbandy and A. Jafarian, *Steepest descent method for system of fuzzy linear equations*, Appl. Math. Comput. **175** (2006), 823-833.
3. S. Abbasbandy, A. Jafarian and R. Ezzati, *Conjugate gradient method for fuzzy symmetric positive definite system of linear equations*, Appl. Math. Comput. **171** (2005), 1184-1191.
4. T. Allahviranloo, *Numerical methods for fuzzy system of linear equations*, Appl. Math. Comput. **155** (2004), 493-502.
5. T. Allahviranloo, *The Adomian decomposition method for fuzzy system of linear equations*, Appl. Math. Comput. **163** (2005), 553-563.
6. J.J. Buckley, *Solving fuzzy equations in economics and finance*, Fuzzy Sets and Systems **48** (1992), 289-296.
7. M. Dehghan and B. Hashemi, *Iterative solution of fuzzy linear systems*, Appl. Math. Comput. **175** (2006), 645-674.
8. M. Friedman, M. Ming and A. Kandel, *Fuzzy linear systems*, Fuzzy Sets and Systems **96** (1998), 201-209.
9. J.U. Jeong, *Stability of a periodic solution for fuzzy differential equations*, J. Appl. Math. Comput. **13** (2003), 217-222.
10. H.M. Nehi, H.R. Maleki and M. Mashinchi, *A canonical representation for the solution of fuzzy linear system and fuzzy linear programming problem*, J. Appl. Math. Comput. **20** (2006), 345-354.
11. S.S. Rao and L. Chen, *Numerical solution of fuzzy linear equations in engineering analysis*, Internat. J. Numer. Methods Engrg. **42** (1998), 829-846.
12. K. Wang and B. Zheng, *Symmetric successive overrelaxation methods for fuzzy linear systems*, Appl. Math. Comput. **175** (2006), 891-901.
13. W. Cong-Xin and M. Ming, *Embedding problem of fuzzy number space: Part I*, Fuzzy Sets and Systems **44** (1991), 33-38.
14. L.A. Zadeh, *Fuzzy sets*, Information and Control **8** (1965), 338-353.

15. L.A. Zadeh, *Toward a generalized theory of uncertainty* (GTU)*—an outline*, Inform. Sci. **172** (2005), 1-40.

Ke Wang received his BSc, MSc and PhD degrees from Lanzhou University, China, in July 1999, June 2003 and June 2006, respectively. He was awarded Qiu Shi Graduate Student Scholarship by Qiu Shi Science & Technologies Foundation, Hong Kong, in Dec. 2005 and has been working in Department of Mathematics at East China Normal University, China, as a postdoctor fellow since June 2006. His research interests focus on numerical linear algebra, matrix analysis and related topics.

School of Mathematics and Statistics, Lanzhou University, Lanzhou 730000, P.R. China
Department of Mathematics, East China Normal University, Shanghai 200062, P.R. China
E-mail: kwang@math.ecnu.edu.cn

Bing Zheng received his B.Sc from North West Normal University (China, 1983) and M.Sc from Anhui University (China, 1989). In 2003, he received his Ph.D from Shanghai University in China and is now working at School of Mathematics and Statistics of Lanzhou University in China as an associate professor. During 2001-2002, he is a visiting scholar at Indian Statistical Institute (Delhi Center). His main research area is Numerical Linear Algebra including generalized inverses of matrix and operator in Banach or Hilbert spaces. He has published some papers in various international and national journals including Applied Mathematics and Computation, ELA and etc.

School of Mathematics and Statistics, Lanzhou University, Lanzhou 730000, P.R. China
E-mail: bzheng@lzu.edu.cn