

Uzawa Algorithms for Fully Fuzzy Linear Systems

H. Zareamoghaddam¹, A.T. Chronopoulos², M. Nouri Kadijani¹, Z. Zareamoghaddam¹

¹ Department of Mathematics, Kashmar Branch, Islamic Azad University, Kashmar, Iran.

Corresponding author's (H. Zareamoghaddam's) E-mail: zareamoghaddam@yahoo.com

² Department of Computer Science, University of Texas at San Antonio,
1 UTSA Circle, San Antonio, TX 78249, USA.

Received 2 January 2015

Accepted 1 March 2016

Abstract

Recently, there have been many studies on solving different kinds of fuzzy equations. In this paper, the solution of a trapezoidal fully fuzzy linear system (FFLS) is studied. Uzawa approach, which is a popular iterative technique for saddle point problems, is considered for solving such FFLSs. In our Uzawa approach, it is possible to compute the solution of a fuzzy system using various relaxation iterative methods such as Richardson, Jacobi, Gauss-Seidel, SOR, SSOR as well as Krylov subspace methods such as GMRES, QMR and BiCGSTAB. Krylov subspace iterative methods are known to converge for a larger class of matrices than relaxation iterative methods and they exhibit higher convergence rates. Thus, they are more widely used in practical problems. Numerical experiments are to illustrate the performance of our suggested methods.

Keywords: FFLS, Uzawa, iterative methods, trapezoidal fuzzy numbers, fuzzy systems.

1. Introduction

Linear system of equations $A\mathbf{x} = \mathbf{b}$, where A and \mathbf{b} are respectively a crisp (i.e. exact) square matrix and vector, mostly comes from the discretization of some practical partial differential equations. The solution of linear systems is required in various areas such as Physics, Statistics, Engineering and Social Science. Then, several researchers have focused on solving such systems by numerous developed iterative methods over the years. In the past, researchers in numerical computing have developed relaxation iterative methods such as Richardson, Jacobi, Gauss-Seidel, SOR, SSOR. Some of these methods are applicable to only to diagonally dominant and others for symmetric positive definite matrices. They are proved to converge in infinite number of iterations (steps) and their convergence rate is

rather slow for practical problems. Thus more attractive iterative methods are the iterative method based on conjugate gradient (CG). CG is applicable for symmetric positive definite matrices and it is proven to converge in a finite number of steps and in general at higher convergence rates than the relaxation iterative methods. Various extensions of CG that apply to non symmetric and non definite matrices exist and are widely known as Krylov subspace (iterative) methods. Well known Krylov subspace methods are GMRES, Orthodir, (s-step) Orthomin (which are extensions of conjugate gradient method) and QMR, CGS, BiCGSTAB (which are extensions of biconjugate gradient and biorthogonal Lanczos methods). These methods and the theory for their convergence is covered in Refs. 1–9 and references therein.

One difficulty for numerous linear systems is that, in

many practical problems, some parameters of such systems may be vague or imprecise. Because of the importance of these systems, fuzzy mathematics is considered for computing the solution of linear system of equations with vague parameters. Many authors have studied different kinds of fuzzy linear systems and have extended several solver methods for the solutions of these problems because of the large amount of applications of such systems in various areas.

Friedman et al. Ref. 10 studied a general form of fuzzy linear system of equations (FLSE) with a crisp coefficient matrix and an arbitrary right hand side fuzzy vector. They used an embedding approach to replace a $n \times n$ FLSE by a $2n \times 2n$ crisp linear system and the solution of FLSE was extracted from the solution of the new parametric system. Many authors have had different investigations on the solution of such fuzzy problems such as the extended methods based on Friedman et al.'s approach, as well as several novel methods. To know more about these methods refer to Refs. 11–16 and references therein.

Furthermore, there is a secondary kind of fuzzy linear systems of the form

$$\tilde{A}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}, \quad (1)$$

where the components of both the coefficient matrix \tilde{A} , i.e. $\tilde{a}_{i,j}$, $1 \leq i, j \leq n$, and right hand side vector $\tilde{\mathbf{b}}$, i.e. \tilde{b}_i , $1 \leq i \leq n$, are fuzzy numbers, which this system is usually called as fully fuzzy linear system (FFLS). FFLSs have been studied by several authors over the last years. Dehghan et al. focused their studies on FFLSs with LR fuzzy numbers and they proposed some iterative methods in Refs. 17–18 for FFLSs with LR fuzzy numbers. Vroman et al. in Ref. 19 suggested a new approach to compute the solution of FFLSs by Cramer's rule and Allahviranloo et al. in Refs. 20–22 developed different approaches for the solution of these systems.

In this paper, we propose practical implementations, which solve the solution of a trapezoidal FFLSs based on Uzawa approach. At first, Uzawa algorithm was considered as a simple implementation which needs minimal memory requirement for solving

linear saddle point system

$$\begin{pmatrix} A & B^T \\ B & -C \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix}, \quad (2)$$

where $A \in R^{n \times n}$ was a symmetric and positive definite matrix, $C \in R^{m \times m}$ was a symmetric positive semi-definite, $B \in R^{m \times n}$ had full rank, $\mathbf{x}, \mathbf{f} \in R^n$ and $\mathbf{y}, \mathbf{g} \in R^m$. Later on, many investigations, such as the works in Refs. 23–26, have developed different Uzawa approaches for solving different kinds of linear or nonlinear saddle point problems because this sort of problem frequently arises from the discretization of Stokes equations and elasticity problems or from the linearization of Navier-Stokes equations. Generally, the implementation of Uzawa algorithm is as follows.

Suppose $(\mathbf{x}_0; \mathbf{y}_0)$ is an initial guess for the exact pair solution $(\mathbf{x}; \mathbf{y})$ of (2). Uzawa algorithms mainly construct a sequence of approximations \mathbf{x}_k and \mathbf{y}_k for \mathbf{x} and \mathbf{y} based on the following instruction:

Algorithm 1: General Uzawa implementation

For $k = 0$ until convergence do

1. Solve linear system $A\mathbf{x}_{k+1} = \mathbf{f} - B^T \mathbf{y}_k$,
2. $\mathbf{y}_{k+1} = \mathbf{y}_k + \tau(\mathbf{g} - B\mathbf{x}_k - C\mathbf{y}_k)$.

End

In this algorithm, the extrapolation scalar τ is a real number which is named as Uzawa parameter.

This paper's aim is to compute the solution of a trapezoidal FFLS by the above Uzawa instruction based on several different appropriate iterative methods. The rest of this paper is organized as follows. In Section 2, some basic definitions and remarks on fuzzy numbers and FFLSs are reviewed. Uzawa algorithm is discussed more in Section 3. Next, some iterative methods for computing the solution of FFLSs based on Uzawa algorithm are proposed in Section 3. Our proposed algorithms are illustrated by solving some numerical examples in Sections 4 and 5.

2. Preliminaries

In this section, some basic definitions and notes on fuzzy numbers and FFLSs are reviewed Refs. 11, 12, 14, 17, 18 and 20.

Definition 1: A fuzzy number is a fuzzy set $\tilde{u} : R \rightarrow [0, 1]$ which satisfies

1. \tilde{u} is upper semi continuous,
2. $\tilde{u} = 0$ outside some interval $[c, d]$,
3. There are real numbers a, b : $c \leq a \leq b \leq d$ so that
 - (a) $\tilde{u}(x)$ is a monotonic increasing function on $[c, a]$.
 - (b) $\tilde{u}(x)$ is a monotonic decreasing function on $[b, d]$.
 - (c) $\tilde{u}(x) = 1$, $a \leq x \leq b$.

For simplicity the above fuzzy number \tilde{u} , which is called trapezoidal fuzzy number, is represented by $\tilde{u} = (a, b, \alpha, \beta)$ where $0 \leq \alpha = a - c$ and $0 \leq \beta = d - b$. The set of all such fuzzy numbers is denoted by E .

Definition 2: The parametric form of an arbitrary fuzzy number \tilde{u} is also represented by an ordered pair of functions $(\underline{u}(r), \bar{u}(r))$, $0 \leq r \leq 1$, see Ref. 27, that satisfies :

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

A crisp number c is simply represented by $\underline{u}(r) = \bar{u}(r) = c$, $0 \leq r \leq 1$.

Definition 3: The membership function of trapezoidal fuzzy number $\tilde{u} = (a, b, \alpha, \beta)$ is

$$\tilde{u}(x) = \begin{cases} \frac{x-a+\alpha}{\alpha}, & a-\alpha \leq x \leq a, \\ 1, & a \leq x \leq b, \\ \frac{b+\beta-x}{\beta}, & b \leq x \leq b+\beta, \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

and its parametric form is

$$\underline{u}(r) = \alpha r + a - \alpha \text{ and } \bar{u}(r) = b + \beta - \beta r. \quad (4)$$

Furthermore, if $a = b$, then $\tilde{u} = (a, \alpha, \beta)$ is called a triangular fuzzy number.

Remark: For an arbitrary trapezoidal fuzzy number $\tilde{u} = (a, b, \alpha, \beta)$, $0 \leq \tilde{u}$ if $0 \leq a - \alpha$.

Definition 4: For two arbitrary fuzzy numbers $\tilde{u} = (m, n, \alpha, \beta)$ and $\tilde{v} = (p, q, \gamma, \lambda)$ (or in parametric form $\tilde{u}(r) = (\underline{u}(r), \bar{u}(r))$ and $\tilde{v}(r) = (\underline{v}(r), \bar{v}(r))$) and a crisp number c , the following algebraic operations are defined

1. $\tilde{u} + \tilde{v} = (m + p, n + q, \alpha + \gamma, \beta + \lambda)$,
2. $-\tilde{u} = (-n, -m, \beta, \alpha)$,
3. $\tilde{u} - \tilde{v} = (m - q, n - p, \alpha + \lambda, \beta + \gamma)$
4. The multiplication of two trapezoidal fuzzy numbers are defined by

$$\tilde{u} \otimes \tilde{v} = (\min\{mp, nq\}, \max\{mp, nq\}, |m\gamma| + |p\alpha|, |n\lambda| + |q\beta|)$$

where $|x|$ means the absolute value of x .

$$5. \quad c\tilde{u} = \begin{cases} (cm, cn, c\alpha, c\beta), & c \geq 0, \\ (cm, cn, -c\beta, -c\alpha), & c < 0. \end{cases}$$

In parametric form, we have

- i. $\tilde{u}(r) \pm \tilde{v}(r) = (\underline{u}(r) \pm \underline{v}(r), \bar{u}(r) \pm \bar{v}(r))$,
- ii. $c\tilde{u} = \begin{cases} (c\underline{u}(r), c\bar{u}(r)), & c \geq 0 \\ (c\bar{u}(r), c\underline{u}(r)), & c < 0 \end{cases}$,
- iii. $\tilde{u} \otimes \tilde{v} = (\underline{uv}(r), \bar{uv}(r))$, where
 $\underline{uv}(r) = \min\{\underline{u}(r)\underline{v}(r), \underline{u}(r)\bar{v}(r), \bar{u}(r)\underline{v}(r), \bar{u}(r)\bar{v}(r)\}$
and
 $\bar{uv}(r) = \max\{\underline{u}(r)\underline{v}(r), \underline{u}(r)\bar{v}(r), \bar{u}(r)\underline{v}(r), \bar{u}(r)\bar{v}(r)\}$
- iv. $\tilde{u} = \tilde{v}$ iff $\underline{u}(r) = \underline{v}(r)$, $\bar{u}(r) = \bar{v}(r)$, $0 \leq r \leq 1$.

Definition 5: The matrix $\tilde{A} = (\tilde{a}_{i,j})$ is called a fuzzy matrix, if each element of \tilde{A} be a fuzzy number. Similarly, a fuzzy vector is defined.

We can represent the trapezoidal fuzzy matrix $\tilde{A} = (\tilde{a}_{i,j})$, i.e. $\tilde{a}_{i,j} = (a_{i,j}, b_{i,j}, \alpha_{i,j}, \beta_{i,j})$; $1 \leq i, j \leq n$, by $\tilde{A} = (A, B, M, N)$ where $A = (a_{i,j})$, $B = (b_{i,j})$, $M = (\alpha_{i,j})$ and $N = (\beta_{i,j})$.

Definition 6: The following linear system of equations

$$\begin{aligned} \tilde{a}_{1,1} \otimes \tilde{x}_1 + \cdots + \tilde{a}_{1,n} \otimes \tilde{x}_n &= \tilde{b}_1, \\ \tilde{a}_{2,1} \otimes \tilde{x}_1 + \cdots + \tilde{a}_{2,n} \otimes \tilde{x}_n &= \tilde{b}_2, \\ &\vdots \\ \tilde{a}_{n,1} \otimes \tilde{x}_1 + \cdots + \tilde{a}_{n,n} \otimes \tilde{x}_n &= \tilde{b}_n. \end{aligned} \quad (5)$$

where $\tilde{A} = (\tilde{a}_{i,j})$ is a $n \times n$ fuzzy matrix and $\tilde{b}_i \in E$, ($1 \leq i \leq n$) is called a fully fuzzy linear system. The system of equations (5) in matrix form is $\tilde{A} \otimes \tilde{\mathbf{x}} = \tilde{\mathbf{b}}$ with the fuzzy vectors $\tilde{\mathbf{x}} = (\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n)^T \in E^n$ and $\tilde{\mathbf{b}} = (\tilde{b}_1, \tilde{b}_2, \dots, \tilde{b}_n)^T \in E^n$.

In this paper, we look for the possible solution of a fully fuzzy linear system of equations $\tilde{A} \otimes \tilde{\mathbf{x}} = \tilde{\mathbf{b}}$ where $\tilde{A} = (A, B, M, N)$ and $\tilde{\mathbf{b}} = (\mathbf{b}, \mathbf{g}, \mathbf{h}, \mathbf{k})$. So, we have

$$\begin{aligned} \tilde{A} \otimes \tilde{\mathbf{x}} = \tilde{\mathbf{b}} &\Rightarrow (A, B, M, N) \otimes (\mathbf{x}, \mathbf{y}, \mathbf{w}, \mathbf{z}) \\ &= (\mathbf{b}, \mathbf{g}, \mathbf{h}, \mathbf{k}) \end{aligned} \quad (6)$$

From multiplication of two fuzzy numbers (following definition 4), the equation (6) is divided into the following four separate crisp linear systems of equations,

$$A\mathbf{x} = \mathbf{b}, \quad (7)$$

$$B\mathbf{y} = \mathbf{g}, \quad (8)$$

$$A\mathbf{w} + M\mathbf{x} = \mathbf{h}, \quad (9)$$

$$B\mathbf{z} + N\mathbf{y} = \mathbf{k}. \quad (10)$$

Let the crisp matrices A and B are nonsingular, the generalized inverse techniques might be useful for some singular matrices, then

$$\begin{aligned} \mathbf{x} &= A^{-1}\mathbf{b}, \mathbf{w} = A^{-1}(\mathbf{h} - M\mathbf{x}) \\ \mathbf{y} &= B^{-1}\mathbf{g}, \mathbf{z} = B^{-1}(\mathbf{k} - N\mathbf{y}). \end{aligned} \quad (11)$$

3. Uzawa algorithm

One of the simple and efficient algorithms for computing the solution of crisp saddle point problem (2) is the Uzawa implementation (algorithm 1) which finds some approximations x_k and y_k ($k > 0$) for the following system of equations

$$A\mathbf{x} + B^T\mathbf{y} = \mathbf{f}, \quad (12)$$

$$B\mathbf{x} - C\mathbf{y} = \mathbf{g}, \quad (13)$$

computed by the following iterative relations

$$A\mathbf{x}_{k+1} = \mathbf{f} - B^T\mathbf{y}_k, \quad (14)$$

$$\mathbf{y}_{k+1} = \mathbf{y}_k + \tau(\mathbf{g} - B\mathbf{x}_k - C\mathbf{y}_k). \quad (15)$$

where τ is a positive descent parameter.

Based on equation (14), the approximation \mathbf{x}_{k+1} ($k \geq 0$) is determined by solving linear system of equations with different right column $\mathbf{f} - B^T\mathbf{y}_k$ for any iteration by some suitable methods. Next, \mathbf{y}_{k+1} ($k \geq 0$) is computed from equation (15) by an appropriate amount of τ . This scalar has directly a significant effect on the convergence speed of Uzawa algorithm because the algorithm may converge slowly or perhaps may not converge for inappropriate values of τ .

Let us assume that A is a nonsingular matrix. By applying block Gaussian elimination to (2) Ref. 24, the following equation is obtained

$$\begin{pmatrix} A & B^T \\ 0 & BA^{-1}B^T + C \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ BA^{-1}\mathbf{f} - \mathbf{g} \end{pmatrix} \quad (16)$$

From (16) it is easily concluded that the unknown vector x can be removed from the second block equation and y may be computed by solving the linear system

$$(BA^{-1}B^T + C)\mathbf{y} = BA^{-1}\mathbf{f} - \mathbf{g}, \quad (17)$$

instead of computing \mathbf{y} from (15). By applying Uzawa approach on this equation, the approximation $\mathbf{y}_k (k > 0)$ can be obtained from

$$\mathbf{y}_{k+1} = \mathbf{y}_k + \tau(BA^{-1}\mathbf{f} - \mathbf{g} - (BA^{-1}B^T + C)\mathbf{y}_k), \quad (18)$$

which is a Richardson iterative method for the solution of (17).

Theorem 1. *Let suppose that $G = I - \tau(BA^{-1}B^T + C)$. The relation (18) is converges for any arbitrary vector \mathbf{y}_0 , if the spectral radius of G is less than unity, i.e. $\rho(G) < 1$.*

Proof. Let \mathbf{y} is the exact solution of (17), then we have

$$\mathbf{y} = \mathbf{y} + \tau((BA^{-1}\mathbf{f} - \mathbf{g}) - (BA^{-1}B^T + C)\mathbf{y}). \quad (19)$$

Suppose \mathbf{y}_k is the approximate solution of (18) for $0 \leq k$. Subtracting (19) from (18) yields

$$\begin{aligned} \mathbf{y}_{k+1} - \mathbf{y} &= G(\mathbf{y}_k - \mathbf{y}) = \dots \\ &= G^{k+1}(\mathbf{y}_0 - \mathbf{y}), \end{aligned} \quad (20)$$

From $\rho(G) < 1$, it is quickly concluded that $G^{k+1} \rightarrow 0$. Then $\mathbf{y}_{k+1} \rightarrow \mathbf{y}$ for any \mathbf{y}_0 . \square

Theorem 2. *Suppose λ_m and λ_M are the smallest and largest eigenvalues of $BA^{-1}B^T + C$, respectively. Then the spectral radius of G is $\rho(G) = \max\{|1 - \tau\lambda_m|, |1 - \tau\lambda_M|\}$. For the proof refer to Ref. 1.*

The spectral radius of matrix G , which is very important for convergence of (18), is named as "convergence factor". The following theorem gives us some more practical information about Uzawa algorithm and optimum Uzawa parameter.

Theorem 3. (Ref. 23) *Let us suppose that $G = I - \tau(BA^{-1}B^T + C)$ and $(\mathbf{x}; \mathbf{y})$ is the block solution of (2) and also $(\mathbf{x}_k; \mathbf{y}_k)$ are the approximations of $(\mathbf{x}; \mathbf{y})$ computed by the Uzawa algorithm, then the following results are obtained*

i. *The sequences $\mathbf{x} - \mathbf{x}_k$ and $\mathbf{y} - \mathbf{y}_k$ satisfy*

$$\begin{aligned} \left((\mathbf{x} - \mathbf{x}_k)^T A (\mathbf{x} - \mathbf{x}_k) \right)^{1/2} &\leq \lambda_M \|\mathbf{y} - \mathbf{y}_k\|, \\ \|\mathbf{y} - \mathbf{y}_k\| &\leq \rho(G) \|\mathbf{x} - \mathbf{x}_k\|. \end{aligned}$$

ii. *For $\tau < \frac{2}{\lambda_M}$, the Uzawa algorithm is convergent and*

$$\rho(G) = \max\{|1 - \tau\lambda_m|, |1 - \tau\lambda_M|\}.$$

iii. *For $\tau = \frac{2}{\lambda_M}$, the convergence factor (spectral reduce) is*

$$\rho(G) = 1 - \frac{\lambda_m}{\lambda_M}.$$

iv. *The optimal parameter is $\tau_{opt} = \frac{2}{\lambda_m + \lambda_M}$ and the corresponding convergence factor is*

$$\rho_{opt}(G) = \frac{\lambda_M - \lambda_m}{\lambda_M + \lambda_m}.$$

The Uzawa algorithm needs some practical iterative method, which works by repeatedly improving an approximate solution until it is accurate enough.

4. Uzawa Iterative methods for FFLSs

Now, the trapezoidal FFLS $\tilde{A} \otimes \tilde{\mathbf{x}} = \tilde{\mathbf{b}}$ is considered to be solved by the Uzawa approach based on some popular iterative methods like Richardson, Jacobi, Gauss-Seidel, SOR, SSOR, USSOR, as well as iterative methods of Krylov subspace type such as conjugate gradient for symmetric crisp linear systems and its variants for nonsymmetric crisp system of equations and several other popular iterative methods like GMRES Ref. 1. In past, Wang and Wu in Ref. 28 used Uzawa-SOR for solving fuzzy linear system of equations. Here, we proposed several Uzawa approaches for solving fully fuzzy linear system of equations.

As it has been discussed in section 2, the FFLS (1) is divided into four separate crisp systems of equations (7) to (10). Using the Uzawa algorithm, the crisp linear systems (7) and (9) are solved together and similarly the equations (8) and (10) are considered as a separate crisp problem. In a better statement, this FFLS is replaced by the following two problems

$$\text{Pr 1: } A\mathbf{x} = \mathbf{b} \text{ and } A\mathbf{w} + M\mathbf{x} = \mathbf{h}.$$

$$\text{Pr 2: } B\mathbf{y} = \mathbf{g} \text{ and } B\mathbf{z} + N\mathbf{y} = \mathbf{k}.$$

Suppose \mathbf{x}_0 and \mathbf{w}_0 are initial values for the solutions \mathbf{x} and \mathbf{w} of Pr 1 and \mathbf{y}_0 and \mathbf{z}_0 are the initial Guesses for Pr 2. The Uzawa algorithm for the solution of FFLS (1) is as follows:

Algorithm 2: Uzawa algorithm for FFLSs

For $k = 0$ until convergence do

1. Compute \mathbf{x}_{k+1} from $A\mathbf{x}_{k+1} = \mathbf{b}$,
2. Compute \mathbf{w}_{k+1} by
 $\mathbf{w}_{k+1} = \mathbf{w}_k + \tau(\mathbf{h} - A\mathbf{w}_k - M\mathbf{x}_k)$,
3. Compute \mathbf{y}_{k+1} from $B\mathbf{y}_{k+1} = \mathbf{g}$,
4. Compute \mathbf{z}_{k+1} by
 $\mathbf{z}_{k+1} = \mathbf{z}_k + \tau'(\mathbf{k} - B\mathbf{z}_k - N\mathbf{y}_k)$.

End

For the better convergence speed, some appropriate parameters τ and τ' have to be considered for algorithm 2. In section 3, we have discussed the convergence factor and optimum parameter of Uzawa algorithm. So, if A, B are symmetric positive definite matrices, it is concluded that

$$\tau_{opt} = \frac{2}{\lambda_m + \lambda_M} \text{ and } \tau'_{opt} = \frac{2}{\lambda'_m + \lambda'_M} \quad (21)$$

and steps 2 and 4 of algorithm 2 are convergent to the exact solutions w and z for any initial vectors w_0 and z_0 , where λ_m and λ_M are respectively the smallest and the largest eigenvalues of A and λ'_m and λ'_M are similarly the smallest and the largest eigenvalues of B .

The following are some of the possible iterative methods for the solution of such FFLSs.

4.1. Uzawa-Richardson (UR) method

The structure of this iterative algorithm is as follows. At first, the Richardson iterative method needs to compute the solutions of two crisp linear systems (7) and (8), separately. From these primary linear systems we have

$$x = (I - \eta A)x + \eta b, \quad (22)$$

$$y = (I - \eta' B)y + \eta' g, \quad (23)$$

where η and η' are the scalars which called as the extrapolation Richardson parameters.

Using the preceding equations and steps 2 and 4 of algorithm 2, sequentially, the approximate solution of a FFLS is computed. Then the algorithm of UR for the acceleration scalars η, η', τ and τ' is as follows.

Algorithm 3: UR algorithm for FFLSs

For $k = 0$ until convergence do

1. $x_{k+1} = (I - \eta A)x_k + \eta b$,
2. $w_{k+1} = w_k + \tau(h - Aw_k - Mx_k)$,
3. $y_{k+1} = (I - \eta' B)y_k + \eta' g$,
4. $z_{k+1} = z_k + \tau'(k - Bz_k - Ny_k)$.

End

If the square matrices A and B are symmetric and positive definite, the optimal extrapolation parameters η and η' are

$$\eta_{opt} = \frac{2}{\lambda_m + \lambda_M} \text{ and } \eta'_{opt} = \frac{2}{\lambda'_m + \lambda'_M}, \quad (24)$$

and the corresponding spectral radiuses are respectively as

$$\begin{aligned} \rho(I - \eta_{opt} A) &= \frac{\lambda_M - \lambda_m}{\lambda_M + \lambda_m} \text{ and} \\ \rho(I - \eta'_{opt} B) &= \frac{\lambda'_M - \lambda'_m}{\lambda'_M + \lambda'_m}, \end{aligned} \quad (25)$$

To apply some iterative methods like Jacobi, Gauss-Seidel (GS), SOR, SSOR, etc, the following decompositions for the coefficient matrices of A, B are needed

$$A = D + L + U \quad (26)$$

$$B = D' + L' + U', \quad (27)$$

in which D and D' are the diagonals of A, B , L and L' are the strict lower parts and U and U' are the strict upper sections of A and B , respectively.

4.2. Uzawa-Gauss-Seidel (UGS) method

Gauss-Seidel algorithm uses the decompositions (26) and (27) to transform the linear equations (7) and (8) into iterative methods of the forms

$$x_{k+1} = Hx_k + f, \quad (28)$$

$$y_{k+1} = H'y_k + f'. \quad (29)$$

To achieve this, Gauss-Seidel computes the components of approximations x_{k+1} and y_{k+1} by the following iterations

$$\begin{aligned} x_{k+1}^{(i)} &= \frac{1}{a_{i,i}} \left(b_i - \sum_{j=1}^{i-1} a_{i,j} x_{k+1}^{(j)} - \sum_{j=i+1}^n a_{i,j} x_k^{(j)} \right), \\ y_{k+1}^{(i)} &= \frac{1}{b_{i,i}} \left(g_i - \sum_{j=1}^{i-1} b_{i,j} y_{k+1}^{(j)} - \sum_{j=i+1}^n b_{i,j} y_k^{(j)} \right), \\ i &= 1, \dots, n, \end{aligned}$$

where $A = (a_{i,j})$, $B = (b_{i,j})$, $b = (b_i)$ and $g = (g_i)$, or in matrix form by

$$x_{k+1} = -(D+L)^{-1} U x_k + (D+L)^{-1} b, \quad (30)$$

$$y_{k+1} = -(D'+L')^{-1} U' y_k + (D'+L')^{-1} g \quad (31)$$

Proposition 1: These two last iterative relations are convergent for any arbitrary initial vectors if the spectral radius of $(D+L)^{-1} U$ and $(D'+L')^{-1} U'$ are less than unity (theorem 1).

Now, the UGS algorithm is similar to algorithm 3 except for the steps 1 and 3 in which UGS instead computes the approximate solutions x_{k+1} and y_{k+1} by (30) and (31).

4.3. Uzawa-SOR (USOR) method

SOR is an iterative algorithm which is a modified version of Gauss-Seidel. SOR is derived from Gauss-Seidel method by applying an extrapolation parameter ω , which named as "convergence rate", to accelerate the convergence speed of algorithm as much as possible. In USOR, we also have to replace the equations (7) and (8) by two iterative methods of the forms (28) and (29).

Due to the SOR condition, the components of x_{k+1} and y_{k+1} are computed from the following iterations

$$\begin{aligned} x_{k+1}^{(i)} &= \frac{\omega}{a_{i,i}} \left(b_i - \sum_{j=1}^{i-1} a_{i,j} x_{k+1}^{(j)} - \sum_{j=i+1}^n a_{i,j} x_k^{(j)} \right) \\ &\quad + (1-\omega) x_k^{(i)}, \\ y_{k+1}^{(i)} &= \frac{\omega'}{b_{i,i}} \left(g_i - \sum_{j=1}^{i-1} b_{i,j} y_{k+1}^{(j)} - \sum_{j=i+1}^n b_{i,j} y_k^{(j)} \right) \\ &\quad + (1-\omega') y_k^{(i)} \\ i &= 1, \dots, n, \end{aligned}$$

where ω and ω' are the convergence rates. In matrix form, we have

$$\begin{aligned} x_{k+1} &= (D + \omega L)^{-1} ((1-\omega)D - \omega U) x_k \\ &\quad + \omega (D + \omega L)^{-1} b, \end{aligned} \quad (32)$$

$$\begin{aligned} y_{k+1} &= (D' + \omega' L')^{-1} ((1-\omega')D' - \omega' U') y_k \\ &\quad + \omega' (D' + \omega' L')^{-1} g. \end{aligned} \quad (33)$$

Proposition 2: The iterative formulas (32) and (33) are convergent for arbitrary initial vectors x_0 and y_0 , if $\rho(H) < 1$ and $\rho(H') < 1$, where $H = (D + \omega L)^{-1} ((1-\omega)D - \omega U)$ and $H' = (D' + \omega' L')^{-1} ((1-\omega')D' - \omega' U')$ (the proof follows from theorem 1).

Therefore, USOR algorithm differs from UR only in the style of computing the solutions of crisp linear systems (7) and (8). USOR computes these solutions by the above iterative formulas (32) and (33). There are some more similar implementations for Uzawa algorithm based on other iterative methods such as Jacobi, SSOR (Symmetric Successive Over Relaxation), USSOR (UnSymmetric successive Over Relaxation), etc (see Refs. 1–3), which may yield similar or better performances. For Uzawa-Jacobi algorithm, x_{k+1} and y_{k+1} are computed by

$$x_{k+1} = -D^{-1}(L+U) x_k + D^{-1}b, \quad (34)$$

$$y_{k+1} = -D'^{-1}(L'+U') y_k + D'^{-1}g. \quad (35)$$

SSOR and USSOR are two modified versions of SOR algorithm which have no advantage over SOR

as iterative methods except they are mainly more practical used in preconditionings for Krylov subspace iterative methods. If these two methods want to be considered for solving FFLSs by Uzawa approach, their instructions are as follows. For Uzawa-SSOR and Uzawa-USSOR, if $A = Q - P$ and $B = Q' - P'$ the iterative solutions are obtained from

$$x_{k+1} = Q^{-1} P x_k + Q^{-1} b, \quad (36)$$

$$y_{k+1} = Q'^{-1} P y_k + Q'^{-1} g, \quad (37)$$

where in Uzawa-SSOR

$$Q = \frac{1}{\omega(2-\omega)} (D + \omega L) D^{-1} (D + \omega U) \quad \text{and} \quad (38)$$

$$Q' = \frac{1}{\omega'(2-\omega')} (D' + \omega' L') D'^{-1} (D' + \omega' U') \quad (39)$$

while in Uzawa-USSOR

$$Q = \frac{1}{\omega_1 + \omega_2 - \omega_1 \omega_2} (D + \omega_1 L) D^{-1} (D + \omega_2 U), \quad (40)$$

and

$$Q' = \frac{1}{\omega'_1 + \omega'_2 - \omega'_1 \omega'_2} (D' + \omega'_1 L') D'^{-1} \times (D' + \omega'_2 U'). \quad (41)$$

In all these cases, the equations (7) and (8) are replaced by equations of the forms (28) and (29). Next, w_{k+1} and z_{k+1} of the two rest equations (9) and (10) are computed by algorithm 2.

The discussed above iterative methods, which are older algorithms, are named stationary iterative methods with simpler theory and implementations. Nonstationary methods (e.g. Krylov) are more recently developed methods, with harder analysis and implementations, can be highly more effective for crisp linear systems. The following is a discussion about Uzawa algorithm based on some nonstationary iterative methods for solving trapezoidal FFLSs.

4.4. Uzawa algorithms for large sparse FFLSs

Previous iterative methods are more suitable for FFLSs with small matrix dimensions because the convergence speed of these methods is usually slow and they may converge to the solution after too many iterations or may not even converge. Furthermore, the required arithmetic computations of those methods will increase dramatically by raising the dimension of the original problems. As a remedy, a class of popular iterative methods which is called "Krylov subspace methods" is considered because many Krylov subspace approaches are popular for dealing with large sparse crisp linear system of equations. Among these methods, there are some techniques only for solving symmetric linear systems such as Conjugate Gradient (CG), MINRES, etc. as well as several popular iterative algorithms such as FOM, GMRES and several variants of CG algorithms like BiCG, CGS, QMR, BiCGSTAB and many unmentioned algorithms for the solution of unsymmetric linear system of equations (see Refs. 1–3). CG, which generates orthog-

onal residual sequences, is an effective method for a linear system of equations with symmetric positive definite coefficient matrix such that this method stores a limited number of vectors. BiCG generates two CG-like sequences of vectors simultaneously, one for original linear system with coefficient matrix A and one for an alternative system with coefficient matrix A^T . QMR is an updated version of BiCG which mostly avoid the possible breakdown that can occur in BiCG. BiCGSTAB is a variant of BiCG which uses different updates for A^T -sequences. BiCGSTAB has a smoother convergence than BiCG. GMRES is an effective method that requires heavy storage requirement. To avoid this drawback, restarted versions of GMRES are considered for crisp linear systems. Generally, most of the practical iterative methods are based on a projection process. For example, FOM and GMRES depend on an orthogonal projection while BiCG variants are based on biorthogonalization techniques.

Generally, each iterative algorithm has some advantages and disadvantages. It is normally difficult to find an iterative method which could quickly solve

all kinds of crisp linear systems with low memory requirement, especially if the linear systems are ill-posed or singular. Numerous iterative techniques have been developed by several authors, for the solution of such problems, over the years. Also, many researchers have investigated the existing methods to improve their convergence speed. Readers whom are interested in knowing more about these iterative techniques for crisp linear systems are referred to Refs. 1 and 4.

Now, for solving large sparse FFLSs by Uzawa approach similar to the previous discussed methods, at first, two extracted crisp linear equations (7) and (8) have to be solved by a Krylov method. Next, the two reminded iterative solutions w_{k+1} and z_{k+1} are computed by the above mentioned Uzawa approaches of Algorithm 2.

Generally, many iterative methods have been developed and it is impossible to discuss about them all in this section. However, our Uzawa algorithm is flexible to employ any effective iterative method for solving two required crisp linear system of equations.

5. Numerical Experiments

The following numerical experiments are for readers to gain a better understanding of the algorithms discussed in previous section.

Example 1: The following 3×3 FFLS

$$\tilde{A} \otimes \tilde{x} = \tilde{b} \quad (42)$$

is considered to be solved by some discussed Uzawa based algorithms where the system of equations is as follows.

$$\left\{ \begin{array}{l} (4.5, 5, 1, 1)\tilde{x}_1 + (0.5, 1, 0.5, 0.5)\tilde{x}_2 \\ + (1, 3, 1, 0.5)\tilde{x}_3 = (14, 33, 11.75, 22.5), \\ (1, 1, 0.5, 1)\tilde{x}_1 + (5, 8, 2, 3)\tilde{x}_2 \\ + (2, 4, 1, 0.5)\tilde{x}_3 = (28, 67, 25.5, 49.5), \\ (2, 3, 2, 2)\tilde{x}_1 + (3, 4, 1, 1.5)\tilde{x}_2 \\ + (6, 8, 3, 3)\tilde{x}_3 = (34, 65, 33, 62.5). \end{array} \right.$$

From (6), it is quickly concluded that

$$A = \begin{pmatrix} 4.5 & 0.5 & 1 \\ 1 & 5 & 2 \\ 2 & 3 & 6 \end{pmatrix}, B = \begin{pmatrix} 5 & 1 & 3 \\ 1 & 8 & 4 \\ 3 & 4 & 8 \end{pmatrix}$$

$$M = \begin{pmatrix} 1 & 0.5 & 1 \\ 0.5 & 2 & 1 \\ 2 & 1 & 3 \end{pmatrix}, N = \begin{pmatrix} 1 & 0.5 & 0.5 \\ 1 & 3 & 0.5 \\ 2 & 1.5 & 3 \end{pmatrix}$$

and the right hand side vectors are

$$b = \begin{pmatrix} 14 \\ 28 \\ 34 \end{pmatrix}, g = \begin{pmatrix} 33 \\ 67 \\ 65 \end{pmatrix},$$

$$h = \begin{pmatrix} 11.75 \\ 25.5 \\ 33 \end{pmatrix}, k = \begin{pmatrix} 22.5 \\ 49.5 \\ 62.5 \end{pmatrix},$$

One important factor for convergence of Uzawa algorithms for FFLSs is the selection of appropriate Uzawa parameters. Here, we illustrate the effect of choosing different Uzawa parameters on the convergence of (42) by selecting two different set of such parameters for the algorithms of UR, UGS and USOR as follows. The initial Guess for the unknown parameters is $x_0 = y_0 = w_0 = z_0 = (0, 0, 0)^T$.

a) At first, we select $\tau = 0.1$ and $\tau' = 0.1$. After 15 iterations Uzawa-Richardson (UR) computed the following approximations where $\eta = 0.1$ and $\eta' = 0.1$.

$$x_{15} = \begin{pmatrix} 1.9968 \\ 3.9936 \\ 3.0083 \end{pmatrix}, y_{15} = \begin{pmatrix} 2.9896 \\ 5.9945 \\ 4.0100 \end{pmatrix},$$

$$w_{15} = \begin{pmatrix} 0.5014 \\ 2.0082 \\ 1.4904 \end{pmatrix}, z_{15} = \begin{pmatrix} 0.5239 \\ 1.5161 \\ 3.4756 \end{pmatrix}$$

and UR gave us the following results for $\eta = 0.2$ and $\eta' = 0.1$

$$x_{15} = \begin{pmatrix} 2.0100 \\ 4.0200 \\ 3.0308 \end{pmatrix}, y_{15} = \begin{pmatrix} 2.9896 \\ 5.9945 \\ 4.0100 \end{pmatrix},$$

$$w_{15} = \begin{pmatrix} 0.4952 \\ 1.9917 \\ 1.4911 \end{pmatrix}, z_{15} = \begin{pmatrix} 0.5239 \\ 1.5161 \\ 3.4756 \end{pmatrix}$$

The numerical approximations of UGS for the above Uzawa parameters are

$$x_{15} = \begin{pmatrix} 2.0833 \\ 4.1667 \\ 2.5417 \end{pmatrix}, y_{15} = \begin{pmatrix} 3.0000 \\ 6.0000 \\ 4.0000 \end{pmatrix},$$

$$w_{15} = \begin{pmatrix} 0.5269 \\ 1.9283 \\ 1.7025 \end{pmatrix}, z_{15} = \begin{pmatrix} 0.5033 \\ 1.5017 \\ 3.4968 \end{pmatrix}$$

The 15th approximations of FFLS (42) computed by USOR algorithm with SOR parameters $\omega = 1.3$ and $\omega' = 0.3$ were

$$x_{15} = \begin{pmatrix} 2.0833 \\ 4.1667 \\ 2.5417 \end{pmatrix}, y_{15} = \begin{pmatrix} 3.0000 \\ 6.0000 \\ 4.0000 \end{pmatrix},$$

$$w_{15} = \begin{pmatrix} 0.5268 \\ 1.9282 \\ 1.7026 \end{pmatrix}, z_{15} = \begin{pmatrix} 0.5049 \\ 1.5026 \\ 3.4953 \end{pmatrix}$$

b) The second set of Uzawa parameters for computing the solution of (42) is as $\tau = 0.2$, $\tau' = 0.01$. To compare the effect of such parameters on convergence of discussed methods, previous Uzawa approaches are again considered for (42) where $\tau = 0.2$, $\tau' = 0.01$. Similarly, the numerical results of such methods in 15th iteration are written.

UR algorithm gave us the following approximations where $\eta = 0.1$ and $\eta' = 0.1$

$$x_{15} = \begin{pmatrix} 1.9968 \\ 3.9936 \\ 3.0083 \end{pmatrix}, y_{15} = \begin{pmatrix} 2.9896 \\ 5.9945 \\ 4.0100 \end{pmatrix},$$

$$w_{15} = \begin{pmatrix} 0.5062 \\ 2.0152 \\ 1.5134 \end{pmatrix}, z_{15} = \begin{pmatrix} 0.8922 \\ 1.6903 \\ 2.5118 \end{pmatrix},$$

and for $\eta = 0.2$ and $\eta' = 0.1$, UR approximations were

$$x_{15} = \begin{pmatrix} 2.0100 \\ 4.0200 \\ 3.0308 \end{pmatrix}, y_{15} = \begin{pmatrix} 2.9896 \\ 5.9945 \\ 4.0100 \end{pmatrix},$$

$$w_{15} = \begin{pmatrix} 0.3763 \\ 1.7567 \\ 1.1220 \end{pmatrix}, z_{15} = \begin{pmatrix} 0.8922 \\ 1.6903 \\ 2.5118 \end{pmatrix}.$$

UGS computes the following approximations

$$x_{15} = \begin{pmatrix} 2.0833 \\ 4.1667 \\ 2.5417 \end{pmatrix}, y_{15} = \begin{pmatrix} 3.0000 \\ 6.0000 \\ 4.0000 \end{pmatrix},$$

$$w_{15} = \begin{pmatrix} 0.5327 \\ 1.9403 \\ 1.7117 \end{pmatrix}, z_{15} = \begin{pmatrix} 0.8609 \\ 1.6211 \\ 2.5624 \end{pmatrix}.$$

The 15th approximations of algorithm 2 computed by USOR with extrapolation parameters $\omega = 1.3$ and $\omega' = 1.3$ were

$$x_{15} = \begin{pmatrix} 2.0833 \\ 4.1667 \\ 2.5417 \end{pmatrix}, y_{15} = \begin{pmatrix} 3.0000 \\ 6.0000 \\ 4.0000 \end{pmatrix},$$

$$w_{15} = \begin{pmatrix} 0.5331 \\ 1.9411 \\ 1.7130 \end{pmatrix}, z_{15} = \begin{pmatrix} 0.8617 \\ 1.6138 \\ 2.5603 \end{pmatrix}.$$

and for $\omega = 1.3$ and $\omega' = 0.3$ the estimated solutions were

$$x_{15} = \begin{pmatrix} 2.0833 \\ 4.1667 \\ 2.5417 \end{pmatrix}, y_{15} = \begin{pmatrix} 3.0407 \\ 5.9892 \\ 3.9847 \end{pmatrix},$$

$$w_{15} = \begin{pmatrix} 0.5331 \\ 1.9411 \\ 1.7130 \end{pmatrix}, z_{15} = \begin{pmatrix} 0.8560 \\ 1.7289 \\ 2.5899 \end{pmatrix}.$$

It is easy to compare the approximated solutions of different Uzawa algorithms and to observe the effect

of various required parameters with each other. The exact fuzzy solution of FFLS of example 1 is

$$\tilde{x} = \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \tilde{x}_3 \end{pmatrix} = (x, y, w, z) \\ = \begin{pmatrix} (2, 3, 0.5, 0.5) \\ (4, 6, 2.0, 1.5) \\ (3, 4, 1.5, 3.5) \end{pmatrix}$$

The following example is a 6×6 FFLS to be solved by Krylov subspace methods.

Example 2: Consider the problem $\tilde{A} \otimes \tilde{x} = \tilde{b}$ where the trapezoidal fuzzy matrix $\tilde{A} = (A, B, M, N)$ is as follows

$$A = \begin{pmatrix} 5 & 2 & 1 & 3 & 2 & 1 \\ 1 & 4.8 & 1 & 2 & 3 & 1 \\ 2 & 2 & 6 & 0.5 & 1 & 3 \\ 2 & 1 & 3 & 5 & 2 & 1 \\ 1 & 1 & 2 & 1 & 10 & 0.5 \\ 2 & 1 & 3 & 1 & 3 & 6 \end{pmatrix},$$

$$B = \begin{pmatrix} 7 & 2 & 2 & 4 & 3 & 2 \\ 1.5 & 5 & 3 & 3 & 3 & 1 \\ 2 & 3 & 10 & 1 & 2 & 3.5 \\ 3 & 3 & 4 & 12 & 5 & 2 \\ 2 & 1 & 3 & 2 & 11 & 1 \\ 3 & 2 & 3 & 4 & 4 & 9 \end{pmatrix}$$

$$M = \begin{pmatrix} 2 & 1 & 0.5 & 0.1 & 0.2 & 1 \\ 1 & 4 & 1 & 0.5 & 0.3 & 1 \\ 0.1 & 0.3 & 2 & 1 & 0.5 & 0.1 \\ 0.5 & 0.2 & 1 & 3 & 1 & 0.2 \\ 1 & 1 & 0.1 & 0.1 & 4 & 1 \\ 1 & 0.5 & 0.3 & 0.2 & 1 & 2 \end{pmatrix},$$

$$N = \begin{pmatrix} 2 & 1 & 1 & 0.1 & 0.5 & 1 \\ 1 & 4 & 1 & 0.2 & 0.2 & 0.5 \\ 0.2 & 0.3 & 3 & 1 & 0.5 & 0.2 \\ 0.3 & 0.3 & 1 & 3 & 1 & 0.3 \\ 0.5 & 1 & 0.2 & 0.1 & 4 & 1 \\ 1 & 0.5 & 0.1 & 0.2 & 1 & 3 \end{pmatrix}$$

and right hand side fuzzy vector is

$$\tilde{b} = \begin{pmatrix} (33, 69, 25, 44) \\ (36.4, 61, 35.2, 46.5) \\ (28, 72, 25.8, 48.6) \\ (37, 120, 36.7, 64.7) \\ (41.5, 85, 36, 62.1) \\ (39, 97, 34.6, 61.8) \end{pmatrix}.$$

This example is solved by the Uzawa algorithm based on some Krylov subspace methods. For all Uzawa algorithms the parameters $\tau = 0.02$ and $\tau' = 0.01$ are considered and the initial vector for all vectors are zero vector of order 6 (i.e. $x_0 = y_0 = w_0 = z_0 = (0, \dots, 0)^T$).

At first, restarted version of GMRES is considered for two crisp linear systems. In this example, GMRES(3) computed the unknown vectors x and y from the related crisp linear systems and the two rest crisp vectors are estimated by Uzawa approaches with the above parameters. The numerical results of this Uzawa algorithm after 6 iterations were

$$x_6 = \begin{pmatrix} 1.0000 \\ 3.0000 \\ 1.0000 \\ 4.0000 \\ 3.0000 \\ 3.0000 \end{pmatrix}, y_6 = \begin{pmatrix} 2.0000 \\ 3.0000 \\ 3.0000 \\ 5.0000 \\ 5.0000 \\ 4.0000 \end{pmatrix},$$

$$w_6 = \begin{pmatrix} 0.8786 \\ 0.9391 \\ 0.9896 \\ 1.2506 \\ 0.9284 \\ 1.4013 \end{pmatrix}, z_6 = \begin{pmatrix} 0.9117 \\ 0.9524 \\ 1.0386 \\ 1.2433 \\ 1.2396 \\ 1.4235 \end{pmatrix}.$$

The Uzawa algorithm based on QMR computed the approximations in 6 iterations as

$$x_6 = \begin{pmatrix} 1.0000 \\ 3.0000 \\ 1.0000 \\ 4.0000 \\ 3.0000 \\ 3.0000 \end{pmatrix}, y_6 = \begin{pmatrix} 2.0000 \\ 3.0000 \\ 3.0000 \\ 5.0000 \\ 5.0000 \\ 4.0000 \end{pmatrix},$$

$$w_6 = \begin{pmatrix} 0.8349 \\ 0.9473 \\ 1.0230 \\ 1.3778 \\ 0.6032 \\ 1.4566 \end{pmatrix}, z_6 = \begin{pmatrix} 0.8944 \\ 0.9537 \\ 1.0384 \\ 1.2379 \\ 1.2666 \\ 1.4163 \end{pmatrix}.$$

To solve this example by the Uzawa algorithm based on BICGSTAB, the following results have been obtained

$$x_6 = \begin{pmatrix} 1.0000 \\ 3.0000 \\ 1.0000 \\ 4.0000 \\ 3.0000 \\ 3.0000 \end{pmatrix}, y_6 = \begin{pmatrix} 2.0000 \\ 3.0000 \\ 3.0000 \\ 5.0000 \\ 5.0000 \\ 4.0000 \end{pmatrix},$$

$$w_6 = \begin{pmatrix} 0.8886 \\ 0.9586 \\ 1.0126 \\ 1.3182 \\ 0.6717 \\ 1.4601 \end{pmatrix}, z_6 = \begin{pmatrix} 0.9081 \\ 0.9529 \\ 1.0401 \\ 1.2416 \\ 1.2402 \\ 1.4212 \end{pmatrix}.$$

The exact solution of this FFLS is

$$\tilde{x} = \begin{pmatrix} (1, 2, 0.5, 1) \\ (3, 3, 1, 2) \\ (1, 3, 1, 1) \\ (4, 5, 2, 1) \\ (3, 5, 1, 2) \\ (3, 4, 2, 2) \end{pmatrix}.$$

6. Conclusion

In this work, the solution of a trapezoidal fully fuzzy linear system of equations has been studied. Before,

Uzawa algorithm has been widely used for solving system of saddle point problems because of its efficiency and minimal storage requirement. We found out that the mentioned fully fuzzy linear system of equations can be solved by different Uzawa approaches based on relaxation iterative methods such as Richardson, Jacobi, Gauss-Seidel, SOR, SSOR and by krylov subspace methods such as GMRES and BiCGSTAB. We note that the rate of convergence of Krylov subspace methods is higher than that of the relaxation methods and that they apply to a larger class of matrices. Also, Krylov methods are preferred because they have low computational costs. Numerical experiments of previous sections confirm the usefulness of our suggested methods for fully fuzzy linear systems.

References

1. Y. Saad, Iterative Methods for Sparse Linear Systems, SIAM (2003).
2. O. Axelsson, Iterative Solution Methods, Cambridge University Press, (1996).
3. H.A. Van Der Vorst, Iterative Krylov methods for large linear systems, Cambridge University Press, Cambridge (2013).
4. R. Barrett et al, Templates for the Solution of Linear Systems Building Blocks for Iterative Methods, SIAM (1994).
5. A. T. Chronopoulos, A Class of Parallel Iterative Methods Implemented on Multiprocessors, Technical Report UIUCDCS-R-86-1267, Department of Computer Science, University of Illinois, Urbana, Illinois, (1986) 1-116.
6. A. T. Chronopoulos, s-Step Iterative Methods for (Non)Symmetric (In)Definite Linear Systems, SIAM Journal on Numerical Analysis, 28 (6) (1991) 1776-1789.
7. A. Chronopoulos and D. Kincaid, On the Odir iterative method for non-symmetric indefinite linear systems, Numerical Linear Algebra with Applications, 8 (2) (2001) 71-82.
8. A. T. Chronopoulos, On the squared unsymmetric Lanczos method, Journal of Computational and Applied Mathematics, 54 (1) (1994) 65-78.
9. A. T. Chronopoulos and A. Kucherov, Block s-step Krylov iterative methods, Numerical Linear Algebra with Applications, 17 (1) (2010) 3-15.
10. M. Friedman, M. Ming and A. Kandel, Fuzzy linear systems, Fuzzy Sets Syst. 96 (1998) 201-209.
11. 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.
12. T. Allahviranloo et al, On the solution of fuzzy linear system, *Appl. Math. Model.*, 37 (2013) 1170–1176.
13. D. Behera and S. Chakraverty, Solution of Fuzzy System of Linear Equations with Polynomial Parametric Form, *Appl. Appl. Math.*, 7(2) (2012) 648–657.
14. R. Ezzati, Solving fuzzy linear systems, *Soft Comput.*, 15 (2011) 193–197.
15. N. Gasilov et al, A geometric approach to solve fuzzy linear systems, *CMES Comp. Model. Eng.*, 75(3) (2011) 189–203.
16. P. Husek, Stability Margin for Linear Systems with Fuzzy Parametric Uncertainty. *PRICAI*, (2008) 708–717.
17. M. Dehghan, B. Hashemi and M. Ghatee, Computational methods for solving fully fuzzy linear systems, *Appl. Math. Comput.*, 179 (1) (2006) 328–343.
18. M. Dehghan, B. Hashemi and M. Ghatee, Solution of the fully fuzzy linear systems using iterative techniques, *Chaos Soliton Fract.*, 34(2) (2007) 316–336.
19. J. Vroman, G. Deschrijver and E.E. Kerre, Solving systems of linear fuzzy equations by parametric functions, *IEEE T. Fuzzy Syst.* 15 (2007) 370–384.
20. T. Allahviranloo et al, General solutions of fully fuzzy linear systems, *Abstr. Appl. Anal.* (2013) ID: 593274, DOI/10.1155/2013/593274.
21. T. Allahviranloo et al, On the new solutions for a fully fuzzy linear system, *Soft Computing.*, (2013) DOI:10.1007/s00500-013-1037-3.
22. S. Moloudzadeh et al, A new method for solving an arbitrary fully fuzzy linear system, *Soft Comput.* 17(9) (2013) 1725–1731.
23. C. Bacuta, A Unified Approach for Uzawa Algorithms, *SIAM J. Numer. Anal.* 44(6) (2006) 2633–2649.
24. Z.H. Cao, Fast Uzawa algorithm for generalized saddle point problems, *Appl. Numer. Math.* 46 (2003) 157–171.
25. J. Lu, Convergence Analysis of the Modified Nonlinear Inexact Uzawa Algorithm for Saddle Point Problem, *Int. J. Contemp. Math. Sciences*, 22(7) (2012) 1067–1075.
26. J. Zhang and J. Shang, A class of Uzawa-SOR methods for saddle point problems, *Appl. Math. Comput.* 216 (2010) 2163–2168.
27. R. Goetschel and W. Voxman, Elementary fuzzy calculus, *Fuzzy Sets Syst.* 18 (1986) 31–43.
28. K. Wang and Y. Wu, Uzawa-SOR method for fuzzy linear system, *International Journal of Information and Computer Science (IJICS)* 1 (2) (2012) 30–33.