Soft Computing (2019) 23:9647–9655
https://doi.org/10.1007/s00500-018-3529-7

METHODOLOGIES AND APPLICATION

Fuzzy linear systems via boundary value problem

Hewayda M. S. Lotfy[1] · Azza A. Taha[1] · I. K. Youssef[1]

Published online: 17 September 2018

Abstract

Reduction in storage and number of operations are considered through avoiding the representation of zeros in storage as well as in the calculations. The importance of this approach has its effect in large problems that appear in numerical treatments of boundary value problems in general and becomes more effective when fuzzy concepts are considered. We introduce an extended embedding solution model named fuzzy compact storage Gauss–Seidel (FCGS) for solving linear systems of equations with a fuzzy-based right-hand side. The model starts by applying the embedding approach to the $n \times n$ fuzzy linear system, a compact storage technique is then applied to the resultant $2n \times 2n$ de-fuzzification matrix, and finally, a Gauss–Seidel method is applied to the system. The FCGS experimental results and algorithm are clarified on some numerical examples including a fuzzy boundary value problem (FBVP). The error improvements through Gauss–Seidel iterations of fuzzy solution computations are reported. The fuzzy solutions at α-cuts are shown and compared to the exact solutions. FCGS achieved a reduction of at least 50% of storage by using the compact storage concepts and consequently obtain a reduction in the mathematical operations and accordingly the running time especially in FBVP applications.

Keywords Compact storage · Iterative methods · Fuzzy system of linear equations

1 Introduction

Systems of linear equations play important roles in several domains such as in the commercial, scientifical, and engineering domains. Usually, problems in these domains get pruned and reduced into linear systems of equations. When the problem structure becomes imprecise, the linear system of equations becomes no more crisp. Vague values of the system parameters can be demonstrated using probability distributions, intervals, or fuzzy values. Fuzziness deals with imprecise, ambiguous, and unclear inputs that is why it is extremely important to create mathematical models and numerical methods for fuzzy linear system of equations (FLS).

Communicated by V. Loia.

✉ Azza A. Taha
Azzataha@sci.asu.edu.eg

Hewayda M. S. Lotfy
hewayda_lotfy@sci.asu.edu.eg

I. K. Youssef
kaoud22@sci.asu.edu.eg

[1] Department of Mathematics, Faculty of Science, Ain Shams University, Abbassia, Cairo 11566, Egypt

A general model for solving an arbitrary $n \times n$ FLS whose coefficients matrix are crisp and its right-hand side column is an arbitrary fuzzy number vector is given (Friedman et al. 1998). They proposed an embedding approach where the original $n \times n$ FLS is replaced by $2n \times 2n$ crisp linear system. Moreover, another embedding approach (Allahviranloo and Hashemi 2014) replaces the original $n \times n$ FLS by two $n \times n$ crisp linear systems. In both of the mentioned embedding techniques, the size of the computational work is doubled. Solving such crisp large linear systems is a problem in itself, and therefore, the use of iterative techniques becomes extremely useful. The iterative numerical solutions for fuzzy linear system are studied (Allahviranloo 2005; Feng 2008; Yin and Wang 2009). Compact storage is a technique for storing sparse matrices. The main idea is to store only the nonzero elements and to be able to perform the necessary matrix operations. Compact storage schemes allocate contiguous storage in memory for the nonzero elements of the matrix. This requires also a scheme for knowing the places of the elements in the original matrix. There are many alternatives for the compact storage schemes, mentioned for instance (Saad 2003; Day 1977; Barrett et al. 1994).

In this paper, we propose a general model for solving an $n \times n$ FLS whose coefficients are crisp with a fuzzy right-hand side. The FCGS model is utilized and consists of three steps. First, applying an embedding strategy to the FLS yields a $2n \times 2n$ de-fuzzified linear system. Second, a compact storage scheme is applied to its coefficient matrix. Then lastly Gauss–Seidel iterative technique is applied and the solution is generated. In Sect. 2, we introduce the FCGS framework material and methods. In Sect. 3, the model is examined on some numerical examples including a boundary value problem. The discussion on the results of the numerical examples is offered in Sect. 4, and finally, the conclusion is given in Sect. 5.

2 Material and methods

The FCGS model starts by converting the $n \times n$ FLS into a crisp $2n \times 2n$ linear system and then the compact storage is applied to the resulted de-fuzzified coefficient matrix to reduce the storage and the number of operations before applying Gauss–Seidel iterative technique to reduce time. The properties of the crisp matrices (Friedman et al. 1998) are extended to L-matrices and 2-cyclic matrices.

2.1 Fuzzy linear systems

The basic definitions of a fuzzy number and FLS (Dehghan and Hashemi 2006; Guo et al. 2013; Allahviranloo 2004; Goetschel and Voxman 1986) are reintroduced in this section.

Definition 1 A fuzzy number is a fuzzy subset of a universe; let this universe be the set of real numbers $\Re^1$. Let u be a fuzzy subset of $\Re^1$ which is characterized by its membership function $\mu_u : \Re^1 \to [0, 1]$ that is understood as the degree of membership of element x to u, $x \in \Re^1$. The set u is completely determined by a set of ordered pairs; the element and its membership, $u = \{(x, \mu_u(x)) \mid x \in \Re^1)\}$, where

$$\mu_u(x) = \begin{cases} \mu_L(x) & \text{if } x \in [a, c], \\ 1 & \text{if } x \in [c, d], \\ \mu_R(x) & \text{if } x \in [d, b], \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

where $\mu_L(x) : [a, c] \to [0, 1]$ and $\mu_R(x) : [d, b] \to [0, 1]$ are left and right fuzzy membership functions of the fuzzy number u in Fig. 1 denoted by (a, c, d, b) where $a \leq c \leq d \leq b$. Frequently, $u(x)$ is written instead of $\mu_u(x)$.

The previous formula shows that the mapping $u : \Re^1 \to [0, 1]$, has the following properties:

1. u is upper semi-continuous function on $\Re^1$.
2. $u(x) = 0$ outside some interval $[a, b] \subset \Re^1$.
3. There are real numbers c, d such that $a < c \leq d < b$, where

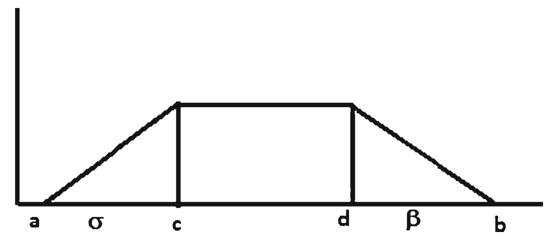

Fig. 1 A general fuzzy number (a, c, d, b)

3.1 $u(x) = \mu_L(x)$ is a real-valued function that is monotonic non-decreasing and continuous on $[a, c]$, where $\mu_L(x) = \frac{x-a}{c-a}$.
3.2 $u(x) = \mu_R(x)$ is a real-valued function that is monotonic non-increasing and continuous on $[d, b]$, where $\mu_R(x) = \frac{b-x}{b-d}$.
3.3 $u(x) = 1$ is a real-valued constant function on $[c, d]$.

This is a description of a trapezoidal fuzzy number when $c \neq d$ and a triangular fuzzy number when $c = d$.

Definition 2 Any fuzzy number u can be written in a parametric form as an ordered pair $(\underline{u}(r), \bar{u}(r))$ of functions, $0 \leq r \leq 1$, which satisfy the following requirements:

1. $\underline{u}(r)$ is a bounded monotonic non-decreasing left-continuous function over [0,1],
2. $\bar{u}(r)$ is a bounded monotonic non-increasing left-continuous function over [0,1].
3. $\underline{u}(r) \leq \bar{u}(r), \quad 0 \leq r \leq 1$

A trapezoidal fuzzy number u is described in the parametric form by the quadrable (c, d, σ, β) where c, d are two de-fuzzifiers and $\sigma > 0$ is the left (width) fuzziness and $\beta > 0$ is the right (width) fuzziness. Hence, the membership function in the parametric form is:

$$u(x) = \begin{cases} \frac{1}{\sigma}(x - c + \sigma) & c - \sigma \leq x \leq c, \\ 1 & c \leq x \leq d, \\ \frac{1}{\beta}(d - x + \beta) & d \leq x \leq d + \beta, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

where the parametric form of a trapezoidal fuzzy number is

$$\underline{u}(r) = c - \sigma + \sigma r, \quad \bar{u}(r) = d + \beta - \beta r \tag{3}$$

For example, the fuzzy number $(1+r, 6-2r)$ has $c = 2, d = 4, \sigma = 1$ and $\beta = 2$.

The trapezoidal fuzzy number becomes a triangular fuzzy number $u(c, \sigma, \beta)$, where there exists exactly one $c \in \Re^1$ with $u(c) = 1$.

The r-level of u, $[u]_r = [\underline{u}(r), \bar{u}(r)], \ \forall r \in [0, 1]$. Hence,

- The 0-level of u is $\overline{[u]_0} = supp(u) = (c - \sigma, d + \beta) = \{x \in \Re^1 : u(x) > 0\}$, and it is called the support of u on $\Re^1$; it is a crisp subset of $\Re^1$ whose elements all have nonzero membership in u where the bar denotes the closure.
- The 1-level is $[u]_1 = [c, d]$. The core of u contains the elements with memberships $u(x) = 1$, core $(u) = \{x \in \Re^1 : u(x) = 1\}$.

A crisp number k is simply represented by $\bar{u}(r) = \underline{u}(r) = k,\ 0 \le r \le 1$. Each crisp number is a single point, while a fuzzy number is a set with degree of membership. Definition 2 yields the same fuzzy number and membership when taking $\sigma = c - a$ and $\beta = b - d$ as in definition 1. By appropriate definitions, the fuzzy number space $\{\underline{u}(r), \bar{u}(r)\}$ becomes a convex cone space E^1. The set E^1 is all (real) fuzzy numbers on $\Re^1$ which are normal, upper semi-continuous, convex and compactly supported fuzzy sets. Fuzzy sets can also be represented by their α-cuts, $\forall \alpha \in [0, 1]$, the crisp set $u_\alpha = \{x \in \Re^1 \mid \mu_u(x) \ge \alpha\}$ is called the α-cut of u.

Definition 3 The operations such as addition, subtraction, and multiplication by a real number k between two fuzzy numbers are defined as follows: Let $u = (\underline{u}(r), \bar{u}(r))$ and $v = (\underline{v}(r), \bar{v}(r))$ be two arbitrary fuzzy numbers; hence,

1. $u = v$ if and only if $\underline{u}(r) = \underline{v}(r)$ and $\bar{u}(r) = \bar{v}(r)$.
2. $u + v = (\underline{u}(r) + \underline{v}(r), \bar{u}(r) + \bar{v}(r))$.
3. $ku = \begin{cases} (k\underline{u}(r), k\bar{u}(r)), & k \ge 0, \\ (k\bar{u}(r), k\underline{u}(r)), & k < 0, \end{cases}$

Definition 4 The $n \times n$ linear system of equations

$$\sum_{j=1}^{n} a_{ij} x_j = y_i,\ i = 1, \dots, n. \tag{4}$$

is a FLS if the coefficient matrix $A = (a_{ij}), 1 \le i, j \le n$ is a crisp $n \times n$ matrix and $y_i \in E^1, 1 \le i \le n$, are fuzzy numbers. When both the elements a_{ij} of the coefficient matrix A and the right-hand side $y_i,\ 1 \le i, j \le n$, are fuzzy numbers, the system is a fully fuzzy linear system (FFLS).

Definition 5 For arbitrary fuzzy numbers $u = (\underline{u}(r), \bar{u}(r))$ and $v = (\underline{v}(r), \bar{v}(r)) \in E^1$, the quantity

$$D^1(u, v) = \int_0^1 (|\underline{u}(r) - \underline{v}(r)| + |\bar{u}(r) - \bar{v}(r)|) dr. \tag{5}$$

is the distance between u and v. The function $D^1(u, v)$ is a metric in E^1. It is shown that (E^1, D^1) is a complete metric space. This distance can be used to define the error between successive calculation of fuzzy solutions in iterative methods where our aim is to maximize the possibility of closeness of exact solutions and our estimated solutions (Guo et al. 2013).

Definition 6 Given a FLS where the coefficient matrix $A = (a_{ij}), 1 \le i, j \le n$ is a crisp $n \times n$ matrix and $y_i \in E^1, 1 \le i \le n$. A fuzzy number vector $(x_1, x_2, \dots, x_n)^T$ given by

$$(\underline{x_i}(r), \overline{x_i}(r)),\ 1 \le i \le n, \quad 0 \le r \le 1,$$

is termed a solution of the FLS if:

$$\begin{cases} \underline{\sum_{j=1}^{n} a_{ij} x_j(r)} = \sum_{j=1}^{n} \underline{a_{ij} x_j}(r) = \underline{y_i}(r), \\ \overline{\sum_{j=1}^{n} a_{ij} x_j(r)} = \sum_{j=1}^{n} \overline{a_{ij} x_j}(r) = \overline{y_i}(r), \\ i = 1, \dots, n \end{cases} \tag{6}$$

If for some i, $a_{ij} > 0$, $1 \le j \le n$, we basically get:

$$\sum_{j=1}^{n} a_{ij} \underline{x_j} = \underline{y_i},\ i = 1, \dots, n$$
$$\sum_{j=1}^{n} a_{ij} \overline{x_j} = \overline{y_i},\ i = 1, \dots, n$$

In general, an arbitrary equation for either $\underline{y_i}$ or $\overline{y_i}$ may incorporate a linear mix of $\underline{x_j}'$s and $\overline{x_j}'$s. Thus, in order to solve the system, one must solve $2n \times 2n$ crisp linear system (Friedman et al. 1998).

Definition 7 A de-fuzzified linear system $SX = Y$ can be built where S is $2n \times 2n$ de-fuzzification matrix which is initialized with zeros and then made from the coefficient matrix A as follows:

$$\begin{aligned} &\text{If } a_{ij} \ge 0 \rightarrow s_{ij} = s_{i+n,j+n} = a_{ij} \\ &\text{If } a_{ij} < 0 \rightarrow s_{i,j+n} = s_{i+n,j} = -a_{ij} \end{aligned} \tag{7}$$

Then, S will have the structure:

$$S = \begin{pmatrix} S_1 & S_2 \\ S_2 & S_1 \end{pmatrix}$$

where S_1 is an $n \times n$ crisp matrix which contains the non-negative elements of the matrix A, and S_2 is an $n \times n$ crisp matrix which contains the absolute values of the negative elements of A, the right-hand side column Y has the arrangement $Y = (\underline{y_1}, \underline{y_2}, \dots, \underline{y_n}, -\overline{y_1}, -\overline{y_2}, \dots, -\overline{y_n})^T$ and therefore the solution vector is expected to be $X = (\underline{x_1}, \underline{x_2}, \dots, \underline{x_n}, -\overline{x_1}, -\overline{x_2}, \dots, -\overline{x_n})^T$

Definition 8 Let $X = \{(\underline{x_i}(r), -\overline{x_i}(r)), 1 \le i \le n\}$ be a set of fuzzy numbers that signify the unique solution of the system $SX = Y$ of the $2n \times 2n$ linear system of equations. The fuzzy number vector $U = \{(\underline{u_i}(r), -\overline{u_i}(r)), 1 \le i \le n\}$ defined by

$$\begin{aligned} &\underline{u_i}(r) = \min\{(\underline{x_i}(r), \overline{x_i}(r), \underline{x_i}(1), \overline{x_i}(1))\}, \\ &\overline{u_i}(r) = \max\{(\underline{x_i}(r), \overline{x_i}(r), \underline{x_i}(1), \overline{x_i}(1))\}, \end{aligned} \tag{8}$$

is known as the fuzzy solution of $SX = Y$. The utilization of $x_i(1)$ is intended to remove the likelihood of fuzzy numbers whose related triangle possesses an angle greater than 90^o. If $(\underline{x}_i(r), \overline{x}_i(r))$, $1 \le i \le n$ are all fuzzy numbers, then

$$\underline{u}_i(r) = \underline{x}_i(r), \overline{u_i}(r) = \overline{x_i}(r),\ \ 1 \le i \le n \tag{9}$$

and U is called a strong fuzzy solution. Otherwise, U is a weak fuzzy solution.

Definition 9 – The matrix A is *positive definite* (Varga 2000), if all its eigenvalues are positive.

– A real matrix A of order n is an L matrix (Young 1971), if:
  - $a_{ii} > 0$ for $i = 1, \ldots, n$
  - $a_{ij} \le 0$ for $i \ne j$ and $i, j = 1, \ldots, n$

– The matrix A has property A, i.e., is 2-cyclic (Varga 2000), if there exists a permutation matrix P such that:

$$PAP^{-1} = \begin{pmatrix} D_1 & H \\ K & D_2 \end{pmatrix}$$

where the diagonal blocks D_1 and D_2 are not necessary of the same order.

It is clear that the elements of the matrix S are all nonnegative real numbers. The properties of the matrix S (Friedman et al. 1998) are considered and in the next theorems we introduce three other properties. In many applications the coefficient matrix A is positive definite (Systems obtained from the discretization of FBVP). Positive definite matrices perform very well with iterative techniques.

Theorem 1 *The matrix S is positive definite if and only if the matrix A is positive definite.*

Proof Let A be positive definite, this means that all its eigenvalues are real and nonnegative, and according to Gershgorin discs theorem (Varga 2000), all the Gershgorin discs are with centers on the positive real axis of the complex domain and with radius guarantees the positive definiteness of A as assumed. Accordingly, the Gershgorin discs of S are coincident with those of A, which completes the proof. □

Theorem 2 *If the matrix A is an L symmetric matrix, then S is a nonnegative symmetric matrix.*

Proof Since A is L matrix, then from Definition 9, $a_{ii} > 0$ and $a_{ij} \le 0$, $\forall i \ne j$ and since A is symmetric then $a_{ij} = a_{ji}$, $\forall i \ne j$.

It is clear from Definition 7 that S is nonnegative, also from Definition 7, we get:

1. If $a_{ij} \ge 0$ then $s_{ij} = s_{i+n,j+n} = a_{ij} = a_{ji} = s_{ji}$
2. If $a_{ij} < 0$ then $s_{ij} = s_{i,j+n} = s_{i+n,j} = -a_{ij} = -a_{ji} = |a_{ji}| = s_{ji}$

then S is symmetric. □

Theorem 3 *If the coefficient matrix A is an L symmetric matrix, then S is two-cyclic block-symmetric.*

Proof From the structure of the S matrix given in Definition 7 and from the definition of L matrix, S_1 is a diagonal matrix and obviously it is block-symmetric. Accordingly, it is clear that S is 2-cyclic. □

Note: The matrix S is promising since it is different from that obtained from the checker board ordering (Young 1971; Saad 2003) that appears in the numerical treatment of Possion's equation. Moreover, the theory of iterative technique introduced by Young can be used.

2.2 Compact storage for FLS

Compact storage techniques are highly recommended for algebraic systems appeared in the numerical treatment of FBVPs due to the sparseness of the coefficient matrix. The situation for the matrix S becomes more than highly recommended. Due to the non-vanishing diagonal elements of the matrix S and to get the most saving in storage, we focus on the compact storage technique named modified sparse row, MSR (Saad 2003). In the MSR format, the coefficient matrix S is stored in only two linear arrays; a real array named AA and an integer array named JA. Usually each element of the real array is represented in memory by eight bytes, while each element of the integer array is represented by one byte. The size of each of these two arrays is equal to the number of all nonzero elements of S plus one. The array AA stores the $2n$ diagonal element of S in order; then the number 0 is used in the $2n + 1$ position as a separator. Later it stores only the nonzero elements of S beginning in the $2n + 2$ position of AA. The elements are determined by scanning S row by row while excluding its diagonal elements. The first $2n$ positions of JA contain the pointer of the beginning of each row in AA. The $2n + 1$ position contains the number of nonzero elements in S plus 1. Afterward, for each nonzero element in $AA(k)$ where $k > 2n + 1$, the integer in $JA(k)$ represents the column index the element $AA(k)$ in S.

2.3 Gauss–Seidel iteration method

The linear system $\sum_{j=1}^{m} a_{ij}x_j = b_i$, $a_{ii} \ne 0$, $i = 1, \ldots, m$, can be solved by the Gauss–Seidel iteration method as:

$$x_i^{[n+1]} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{[n+1]} - \sum_{j=i+1}^{m} a_{ij} x_j^{[n]} \right), \tag{10}$$

$$i = 1, \ldots, m$$

Usually, we write $A = D - L - U$ where D is a diagonal matrix with the same diagonal elements as A, $-L$ and $-U$ are the strictly lower and upper triangular parts of A, respectively. The Gauss–Seidel iteration method in matrix notation is:

$$X^{[n+1]} = T_{GS}X^{[n]} + (D - L)^{-1}b$$
$$T_{GS} = (D - L)^{-1}U$$

where T_{GS} is the Gauss–Seidel iteration matrix. Gauss–Seidel is convergent when the spectral radius, ρ, of the matrix T_{GS} is less than one, $\rho(T_{GS}) < 1$.

3 Numerical examples

To illustrate the introduced FCGS theoretical performance, we consider two examples: the first is straightforward mentioned in many publications and the second is one which can be enlarged to introduce large linear system as appears in the treatment in a FBVP. One of the important sources of structured large linear system is the discretization of boundary value problems. We consider a linear second-order FBVP of the form:

$$-x'' = f(t), 0 \leq t \leq 1$$

$$\begin{aligned} x(0) &= \tilde{A} \\ x(1) &= \tilde{B} \end{aligned} \quad (11)$$

where $\tilde{A}$, $\tilde{B}$ are given fuzzy numbers.

It is proved (Gasilov et al. 2011) that the solution of the FBVP (11) will be fuzzy functions even only when the boundary data are fuzzy numbers and the derivatives are understood in the classical sense. The negative sign in the differential equation of the BVP (11) is to insure the positive definiteness of the differential operator as well as the positive definiteness of the coefficient matrix appears in the discretization of the FBVP (11) and also to exclude zero pivots in the Gauss–Seidel implementation. In this section, we give some numerical experiments to demonstrate the results obtained in the previous section. All the numerical experiments presented in this section were computed in double precision using MATLAB 9 on a PC with a 2.40 GHz 64-bit processor and 8.0 GB memory.

Example 1 Consider the system (Allahviranloo 2004, 2005; Allahviranloo and Hashemi 2014).

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

As described the coefficient matrix A and its extended defuzzification coefficient matrix S are:

$$A = \begin{pmatrix} 1 & -1 \\ 1 & 3 \end{pmatrix}, S = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 3 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 3 \end{pmatrix}$$

It is clear that the matrix S has eight zero elements representing 50% of the elements in S. Usually the matrix S is represented as a real matrix which is represented in memory by 16×8 which is 128 bytes. Using MSR compression technique, S is represented by the two linear arrays AA and JA, the first 4 positions in AA contain the diagonal elements of S in order. The position 5 is not used; we have only four nonzero diagonal elements, 0 is stored in this position. Starting at position 6, the remaining nonzero elements of S are stored row by row. The first 4 positions of JA contain pointers to the position number of each row in AA stored after the zero in AA. For example, $JA(1) = 6$, since the elements of row 1 is stored in AA starting from position 6. The value of $JA(5)$ is 9 indicating the number of nonzero elements in S plus 1. Finally, for each element $AA(k)$, where $k > 5$, the integer $JA(k)$ represents the column index of the element $AA(K)$ in S. The arrays AA and JA are:

$$\begin{aligned} AA &= \begin{pmatrix} 1 & 3 & 1 & 3 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \\ JA &= \begin{pmatrix} 6 & 7 & 8 & 9 & 9 & 4 & 1 & 2 & 3 \end{pmatrix} \end{aligned}$$

The array AA is real and is represented by $9 \times 8 = 72$ bytes, and the array JA is represented as a short integer 9 bytes; therefore, there are 47 bytes saving in this simple example. Moreover, the array AA can be reduced more as follows:

$$AA = \begin{pmatrix} 1 & 3 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Since the first n ($n = 2$ in this example) diagonal elements are repeated.

In Fig. 2, the fuzzy numbers in the right-hand side are shown. Then Gauss–Seidel iteration method is applied on AA, computing the convergence related to an error of 10^{-4}. The resulted fuzzy solutions are shown in Fig. 3, and the error performance is shown in Fig. 4. The fuzzy solutions are:

$$x_1 = (\underline{x_1}(r), \overline{x_1}(r)) = (1.375 + 0.625r, 2.875 - 0.875r),$$
$$x_2 = (\underline{x_2}(r), \overline{x_2}(r)) = (0.875 + 0.125r, 1.375 - 0.375r).$$

Example 2 Consider the FBVP, (Gasilov et al. 2011)

$$\begin{aligned} -x'' + 16x &= 47 - 8t^2 \\ x(0) &= (2, 3, 3.5) \\ x(2) &= (0.5, 1, 1.5) \end{aligned}$$

The fuzzy number, $x(0) = (2, 3, 3.5)$, is converted into its parametric form $(2 + r, 3.5 - 0.5r)$ by using Eq. (3). Similarly, $x(2) = (0.5 + 0.5r, 1.5 - 0.5r)$. The standard finite difference technique with grid spacing $h = 0.2$ gives a

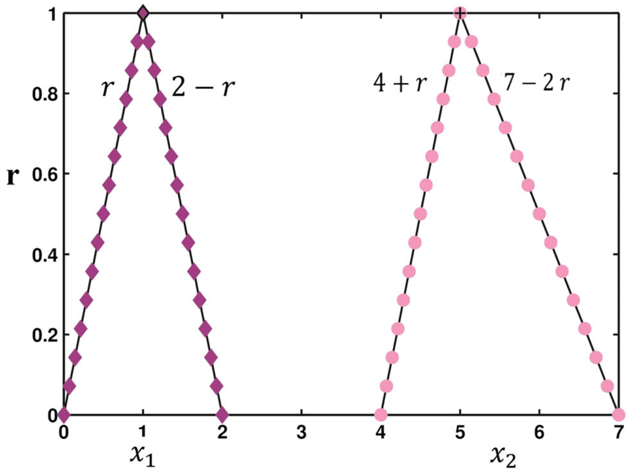

Fig. 2 The fuzzy numbers appears in Example 1, $(r, 2-r)$ and $(4+r, 7-2r)$

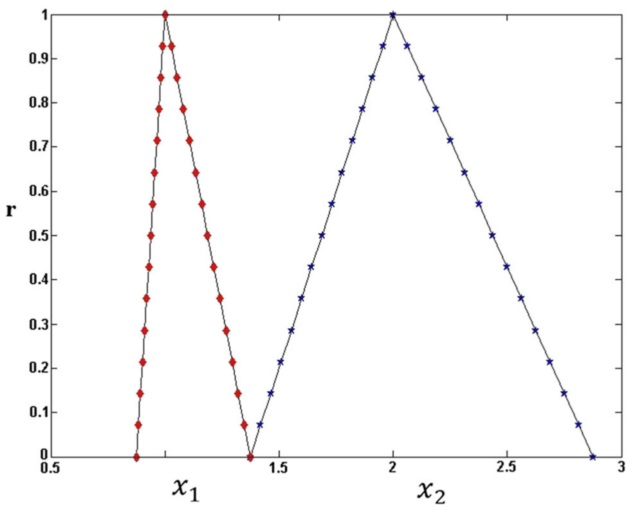

Fig. 3 The solutions x_1 and x_2 of Example 1 after 7 Gauss–Seidel iterations

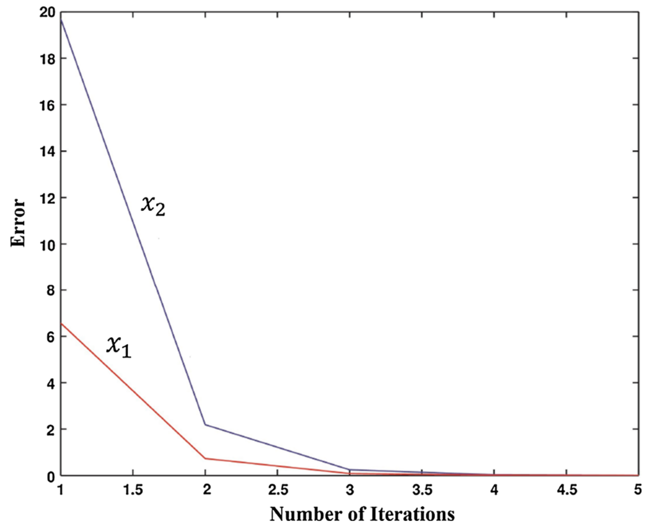

Fig. 4 Gauss–Seidel error improvement while computing the two solutions x_1 and x_2 along with iterations (x-axis) of example 1

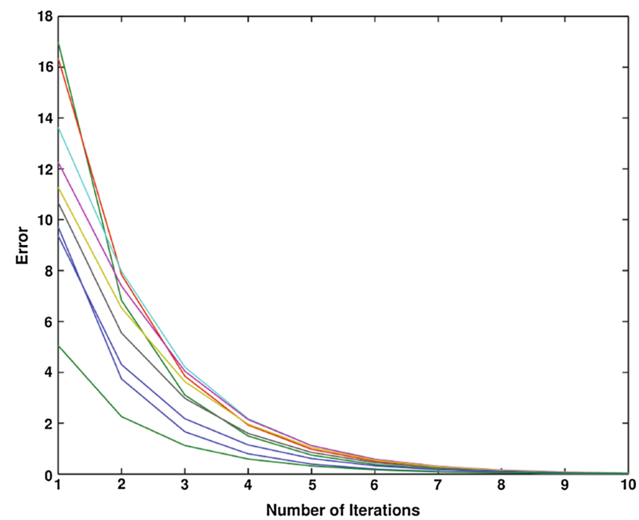

Fig. 5 Gauss–Seidel error improvement while computing the solutions $x_i, i = 1, \ldots, 9$ through the first 14 iterations for Example 2

tri-diagonal system of $n = 9$ linear equations corresponding to the interior points of the grid imposing the domain [0, 2] (Fig. 5).

$$\begin{pmatrix} 2.64 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2.64 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2.64 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2.64 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2.64 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2.64 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2.64 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2.64 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2.64 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{pmatrix} = \begin{pmatrix} (3.8672 + r, 5.3672 - 0.5r) \\ 1.8288 \\ 1.7648 \\ 1.6752 \\ 1.56 \\ 1.4192 \\ 1.2528 \\ 1.0608 \\ (1.3432 + 0.5r, 2.3432 - 0.5r) \end{pmatrix} \tag{12}$$

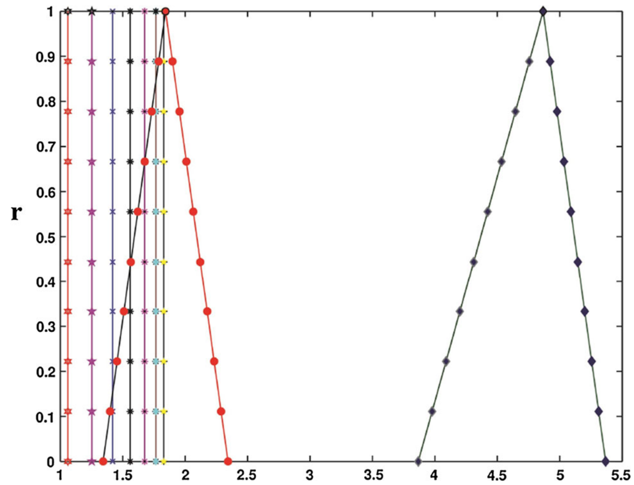

Fig. 6 The fuzzy numbers appears in the RHS of the system of Eq. 12, two fuzzy numbers and 7 crisp numbers represented by straight vertical lines

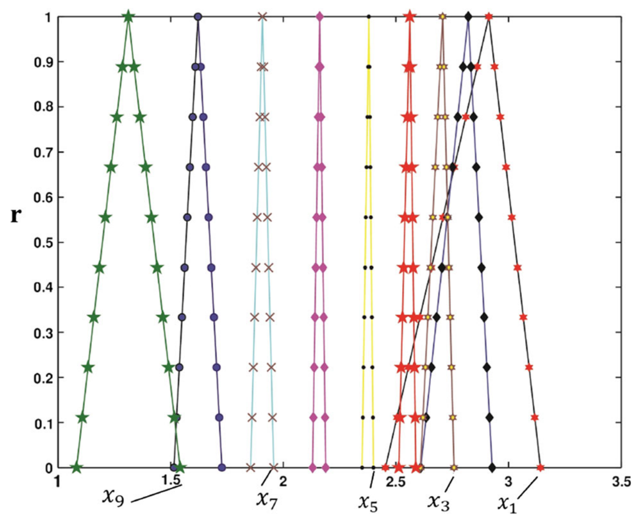

Fig. 7 The solution of the system in Eq. 12 approximates the solution of the FBVP in Example 2

$$S = \begin{pmatrix} S_1 & S_2 \\ S_2 & S_1 \end{pmatrix} \tag{13}$$

The coefficient matrix is crisp and the right-hand side is crisp except the first and ninth components are fuzzy numbers. The seven crisp numbers in Eq. (12) are represented in FCGS as a pair as mentioned in definition 2 and as shown in Fig. 6. This FLS system is then solved using FCGS, and the resulting nine solutions are shown in Fig. 7. The $2n \times 22n$ de-fuzzification matrix S defined in Definition 7 will take the form:
where S_1 and S_2 are defined in (14).

$$S_1 = \begin{pmatrix} 2.64 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2.64 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2.64 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2.64 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.64 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2.64 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2.64 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.64 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.64 \end{pmatrix} \text{ and}$$

$$S_2 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \tag{14}$$

Applying compact storage on the 18×18 matrix S as defined in (13) yields the two arrays, AA and JA (they will not be shown here since they are large but easy to get) where the number of nonzero elements are 50 out of 324 elements. Using compact storage schemes in this case is highly recommended since it will provide a great reduction in the storage and consequently in the performed operations. The solutions obtained are:

$$x_1 = (\underline{x_1}(r), \overline{x_1}(r)) = (2.455 + 0.458r, 3.142 - 0.230r).$$
$$x_2 = (\underline{x_2}(r), \overline{x_2}(r)) = (2.928 - 0.106r, 2.611 + 0.211r).$$
$$x_3 = (\underline{x_3}(r), \overline{x_3}(r)) = (2.609 + 0.098r, 2.758 - 0.050r).$$
$$x_4 = (\underline{x_4}(r), \overline{x_4}(r)) = (2.588 - 0.027r, 2.513 + 0.049r).$$
$$x_5 = (\underline{x_5}(r), \overline{x_5}(r)) = (2.345 + 0.030r, 2.400 - 0.020r).$$
$$x_6 = (\underline{x_6}(r), \overline{x_6}(r)) = (2.188 - 0.027r, 2.130 + 0.031r).$$
$$x_7 = (\underline{x_7}(r), \overline{x_7}(r)) = (1.855 + 0.052r, 1.958 - 0.050r).$$
$$x_8 = (\underline{x_8}(r), \overline{x_8}(r)) = (1.728 - 0.106r, 1.515 + 0.107r).$$
$$x_9 = (\underline{x_9}(r), \overline{x_9}(r)) = (1.083 + 0.230r, 1.542 - 0.230r).$$

The fact is that the odd solutions (e.g., x_1, x_3) are fuzzy numbers, while the even ones (e.g., x_2, x_4) are not fuzzy numbers. The solution in this case is a weak fuzzy number (Friedman et al. 1998), and the even solutions are reintroduced in the form:

$$u_2 = (2.611 + 0.211r, 2.928 - 0.106r).$$
$$u_4 = (2.513 + 0.049r, 2.588 - 0.027r).$$
$$u_6 = (2.130 + 0.031r, 2.188 - 0.027r).$$
$$u_8 = (1.515 + 0.107r, 1.728 - 0.106r).$$

where $u_k = x_k,\ k = 1, 3, 5, 7, 9$.

Figure 5 illustrates the error behavior through 10 iterations, while Fig. 8 displays the solutions at various times t where $t \in [0, 2]$, it shows the solution with the following

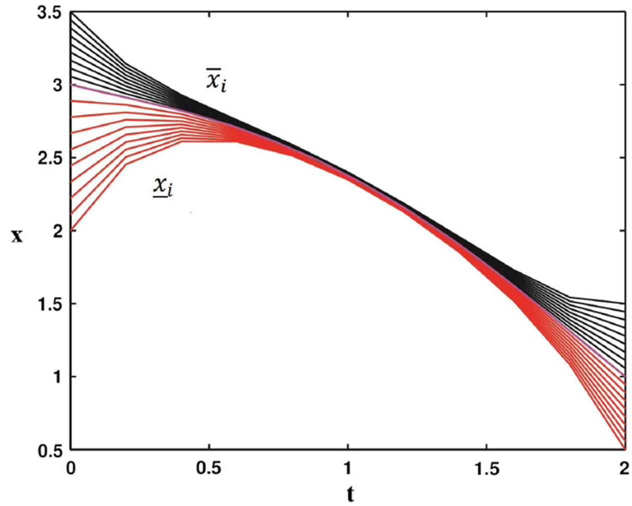

Fig. 8 Fuzzy solutions at various times with α-cut = 0(0.1)1 for Example 2, where black is the higher part of the fuzzy solutions and red is the lower part of the fuzzy solutions

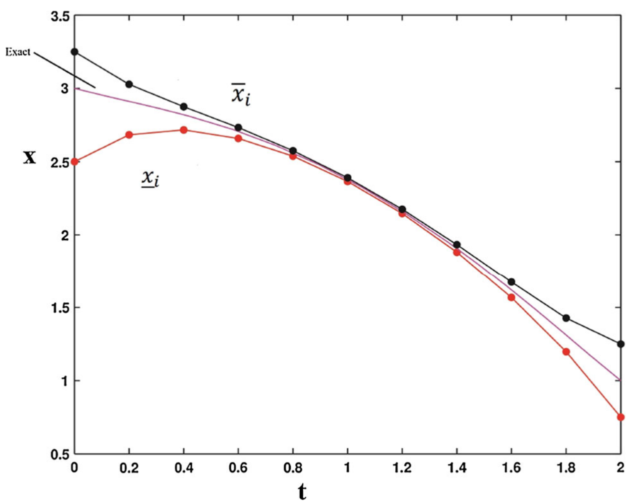

Fig. 9 The fuzzy solutions at α = 0.5 as shown in Fig. 8, the effects of the fuzzy boundary values have propagated along the whole interval. The middle line represents the crisp solution

range of α-cut = 0(0.1)1. The black lines are the higher part of the solutions, and red lines represent the lower part of the solutions. Figure 9 shows the fuzzy solution with respect to α-cut = 0.5, where the exact solution lies between the upper and lower solutions at the same α-cut.

It is easy to show that the matrix S in this example is symmetric, positive definite matrix and moreover it is 2-cyclic consistently ordered matrix.

4 Discussion

Solving large linear system is a fundamental problem in science; generally, many mathematical models involve solution of linear systems explicitly or implicitly. Fuzzy linear systems appear in various realistic applications. The memory requirement is the great problem in solving linear systems. Iterative techniques have reduced the computational work. In solving fuzzy linear systems of size n we have to solve a crisp system of size $2n$. Therefore, the need to use compact storage becomes inevitable. A reduction of at least 50% of the storage is achieved by using the compact storage concepts. Since the number of zeros in the matrix S are at least 50%, the system can be most efficiently solved if the zero elements of S are not stored. Sparse storage schemes allocate contiguous storage in memory for the nonzero elements of the matrix. In addition to a storage reduction, a consequent reduction in a number of the mathematical operations is obtained due to the representation of only nonzero elements. Further reduction can be achieved due to the nature of the problem such as in the tri-diagonal system resulted from the FBVP; this will be our concern in a next subsequent work.

For the first example, the total average elapsed time of FCGS is 0.04 s utilizing a Gauss–Seidel application with compact storage which achieved average elapsed time 0.007 s trying to reach a range of error $= 10^{-16}$. The input maximum iteration is 18, and it reported only 7 iterations are enough to reach error in a range of 10^{-4} as shown in Fig. 4. The maximum error obtained is 19.7 where the error improves fast in the first 4 iterations. The memory storage needed for S is 128 bytes while using FCGS it is reduced to 81 bytes with 37% reduction.

For the second example, the average elapsed time of FCGS is 0.47 s with an application of Gauss–Seidel with compact storage that achieved average elapsed time is 0.066 s trying to reach a range of error $= 10^{-16}$, the input maximum iteration 55 and only 20 iteration where found to be enough in the way to reach error in the range of 10^{-4}. The maximum error reached is 66.4. The memory storage needed for S is 2592 bytes, while using FCGS it is reduced to 459 bytes with 82% reduction. The error improvement through the first 10 iterations for the nine fuzzy numbers calculations is shown in Fig. 5, and it is noticed that a reasonable less number can be selected around 10 iterations. It is apparent that the suitable number of iterations of Gauss–Seidel can be a small number to reach a considerable low error, as shown in Figs. 4 and 5. It is interesting to note that the behavior of the solution for the FBVP with the change of the α values and the crisp solution is the boundary curve between the red and black graphs in Fig. 8 and the middle line in Fig. 9.

5 Conclusion

The iterative techniques are the most appropriate approach for solving fuzzy-related algebraic systems. A simple linear system consists of at least two equations as in Example 1, and this requires solution of a crisp system of size 4 with at least 50% of its coefficients are zeros. The introduced FCGS treatment is effective in storage and computational work and accordingly in the running time as illustrated in Examples 1 and 2.

Fuzzy algebraic systems with positive definite coefficient matrix corresponds to crisp systems with enlarged positive definite coefficient matrix as proved in Theorem 1. Fuzzy algebraic systems with L symmetric matrix give enlarged crisp systems with nonnegative symmetric matrices as in Theorem 2.

Compact storage techniques are more effective when treating BVPs with fuzzy boundary data than the corresponding classical cases; the number of nonzero elements is doubled, while the size of the coefficient matrix is quadrupled; and this has very high effects in large systems as illustrated in Example 2

The FCGS can be efficiently used for other different applications. This approach will be further extended, and further storage schemes can be experimented in future work.

Acknowledgements The authors are grateful to the referees for the careful reading of the manuscript as well as their helpful comments.

Compliance with ethical standards

Conflicts of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Allahviranloo T (2004) Numerical methods for fuzzy system of linear equations. Appl Math Comput 155(2):493–502

Allahviranloo T (2005) Successive over relaxation iterative method for fuzzy system of linear equations. Appl Math Comput 162:189–196

Allahviranloo T, Hashemi A (2014) The embedding method to obtain the solution of fuzzy linear systems. Int J Ind Math 6(3):229–233

Barrett R, Berry M, Chan TF, Demmel J, Donato J, Dongarra J, Eijkhout V, Pozo R, Romine C, Van der Vorst H (1994) Templates for the solution of linear systems: building blocks for iterative methods, 2nd edn. Siam, Philadelphia

Day JD (1977) Compact storage for iterative solution of large sparse linear systems of equations. Int J Math Educ Sci Technol 8(4):385–387

Dehghan M, Hashemi B (2006) Iterative solution of fuzzy linear systems. Appl Math Comput 175:645–674

Feng Y (2008) An iterative method for fuzzy linear systems. In: Fifth international conference on fuzzy systems and knowledge discovery, IEEE Computer Society, pp, 565–569 https://doi.org/10.1109/FSKD.2008.92

Friedman M, Ming M, Kandel A (1998) Fuzzy linear systems. Fuzzy Sets Syst 96:201–209

Gasilov N, Amrahov SE, Fatullayev AG (2011) Linear differential equations with fuzzy boundary values. In: Proceeding of the fifth international conference application of information and communication technologies (AICT2011), Baku, Azerbaijan, pp. 696–700

Goetschel R, Voxman W (1986) Elementary fuzzy calculus. Fuzzy Sets Syst 18:31–43

Guo X, Shang D, Lu X (2013) Fuzzy approximate solutions of second-order fuzzy linear boundary value problems. Boundary Value Probl J 212:1–17

Saad Y (2003) Iterative methods for sparse linear systems, 2nd edn. SIAM, Philadelphia

Varga RS (2000) Matrix iterative analysis, 2nd edn. Springer, Berlin

Yin J-F, Wang K (2009) Splitting iterative methods for fuzzy system of linear equations. Comput Math Model 20(3):326–335

Young DM (1971) Iterative solution of large linear systems. Academic Press, New York