International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor 2025: 7.089

An Algorithm for Solving Interval Full Linear Equation Systems

Duy Cong Le[1], Trung Thanh Nguyen[2]

[1]Faculty of Civil Engineering, Mien Trung University of Civil Engineering, Danang Campus, Vietnam
Corresponding Author Email: *lecongduy[at]muce.ed.vn*

[2]Faculty of Civil Engineering, Mien Trung University of Civil Engineering, Danang Campus, Vietnam
Email: *nguyenthanhtrung[at]muce.ed.vn*

Abstract: ***This paper presents a new algorithm for solving full interval linear equation systems in which both the coefficient matrix and the right-hand-side vector contain interval values. The proposed approach combines interval function optimization with the Differential Evolution algorithm to determine the lower and upper bounds of the interval solution while reducing the overestimation commonly associated with interval arithmetic. The algorithm is implemented in Maple and verified through representative numerical examples. Comparisons with existing methods demonstrate that the proposed approach produces reliable strong interval solutions while maintaining good computational accuracy.***

Keywords: Interval numbers; Interval analysis; Interval arithmetic; Interval matrix; Interval linear equation systems; Differential Evolution; Optimization

1. Introduction

The system of linear interval equations can be presented as follows:

[a]{x}={b} where [a] is an interval matrix, {b} is an interval vector and {x} is an interval solution vector.

Solving the systems of linear equations is considered as a very important part of numerical analysis. The full linear equation systems can be solved by different methods. If the coefficient matrix and right vector are in crisp then this system can be solved by the Cramer's rule, Gauss-Elimination method, Gauss Jacobi iteration method, Gauss-seidel iteration method, matrix inversion method, etc. But in real life situations, parameters of these systems often are charged by different kinds of uncertainties. When the coefficient matrix and the right vector are in the interval form then we may find the solution by Interval Gauss-Elimination type method. In this context different authors developed different methods Such as S.Ning, RB Kearfott [1] developed method by extending a proposed technique of Hansen and Rohn with a formula that bounds the solution set of a system of equations whose coefficient matrix A=\underlineA, \overlineA is an H-matrix; when \bfA is centered about a diagonal matrix. Svetoslav Markov [2] and the authors in [4, 9] developed a method of using an iterative Jacobi type for interval solution of linear parametric systems. Mariusz Pilarek [3] introduced interval extended zero methods for solving systems of linear equations. The authers in [5] also introduced a method on the solution of interval linear systems by using the optimization problem based on gradient vector in order to obtain the lower bound and upper bound of the interval solution. In [6] the authors presented algorithm for the solution of interval linear systems based on interval versions of Gaussian Elimination method for the determinants of interval matrices and the Cramer's rule. And The authors in[7] introduced algorithm for a System of Interval Linear Equations Based on the Constraint Interval Point of View. Esmaeil Siahlooei, SeyedAbolfazl, ShahzadehFazeli[8] presented a method for solving interval linear system based on conjugate gradient algorithm in the context view of interval numbers. T. Nirmala and K. Ganesan [10] introduced an interval version of Gauss Jordon method for solving system of interval linear equations involving generalized interval numbers by applying the generalized interval arithmetic operations. In [11] the authors presented an algorithm using a new type of arithmetic operations and pairing technique on intervals for the interval solution of interval linear equations systems.

In this paper we have proposed a new method for solving interval system of linear equations which is undoubtedly this method always gives strong solution. Specifically, in the paper, we apply the interval function optimization algorithm to calculate the interval functions, determining the variable results of linear equation system. The calculation process is programmed by the authors in Maple.17 software to determine the interval output results.

2. Research Method

2.1 Basic arithmetic operations [12, 13, 14]

A real interval is a non-empty set of real numbers:

$$x = [\underline{x}, \overline{x}] = \left\{ x \in R \middle| \underline{x} \le x \le \overline{x} \right\}$$

where $\underline{x}$ and $\overline{x}$ are lower and upper bounds of the interval $\breve{x}$, x is part of the range $\breve{x}$, R is the set of real numbers. Specifically, an interval A includes two parameters: lower bound a_1, upper bound a_2, denoted as:

A = $[a_1, a_2]$, $a_1 \le a_2$, So Interval arithmetic operators including (+, -, x, /) are performed on intervals.

The sum of two intervals A and B is an interval A+B with the definition:

$A+B = [a_1, a_2] + [b_1, b_2] = [a_1+b_1, a_2+b_2]$; (1)

The subtraction of two intervals A and B results in an interval A-B with the following definition:
$A-B = [a_1, a_2] - [b_1, b_2] = [a_1-b_2, a_2-b_1]$; (2)

The product of two intervals A and B is an interval AxB with the definition:
$AxB = [a_1, a_2]x[b_1, b_2] = [\min(a_1b_1, a_1b_2, a_2b_1, a_2b_2), \max(a_1b_1, a_1b_2, a_2b_1, a_2b_2)]$; (3)

In the case where $a_1, a_2, b_1, b_2 \geq 0$ then: $AxB = [a_1, a_2] \times [b_1, b_2] = [a_1b_1, a_2b_2]$; (4)

The quotient of two intervals A and B when $B \not\ni 0$ is an interval determined through interval multiplication as follows:

$A/B = [a_1, a_2]/[b_1, b_2]=[a_1, a_2]x[1/b_2, 1/b_1] =$
$= [\min(a_1/b_1, a_1/b_2, a_2/b_1, a_2/b_2), \max(a_1/b_1, a_1/b_2, a_2/b_1, a_2/b_2)]$; (5)

In the case where $a_1, a_2, b_1, b_2 \geq 0$ then: $AxB = [a_1, a_2]/[b_1, b_2] = [a_1/b_2, a_2/b_1]$

In many cases, the value boundaries determined by interval arithmetic tend to expand beyond the boundaries of the actual value range, thus making the result inaccurate.

2.2 Interval function [12, 13, 14]

An interval function is a function that takes the interval value of one or more interval parameters. So an interval function maps the value of one or more interval parameters onto an interval. For a function $f(x_1,\ldots,x_n)$, if the interval value function $\breve{f}(\breve{x}_1,\ldots,\breve{x}_n)$ has properties: $\breve{f}(\breve{x}_1,\ldots,\breve{x}_n) = f(x_1,\ldots,x_n)$ for all arguments x; (6)

Then the function $\breve{f}$ is called an interval expansion function of f. In particular, the natural interval expansion function of f can be obtained by replacing each real variable x_i with an interval variable $\breve{x}_j$ and each real operation (+,-,×,÷) with the corresponding interval operations. If the function $\breve{f}$ is an expression with a finite number of interval variables $(\breve{x}_1,\ldots,\breve{x}_n)$ and interval operations (+,-,×,÷) then the function satisfies the basic inclusion property :

If $\breve{x}_1 \subseteq \breve{y}_1,\ldots,\breve{x}_n \subseteq \breve{y}_n$ *then* $\breve{f}(\breve{x}_1,\ldots,\breve{x}_n) \subseteq \breve{f}(\breve{y}_1,\ldots,\breve{y}_n)$; (7)

where $\breve{x} \subseteq \breve{y}$ means that the interval $\breve{x}$ is a subset of the interval $\breve{y}$, only if $\underline{y} \leq \underline{x}$ and $\overline{x} \leq \overline{y}$.

2.3 Implementation of Interval optimization method

This method is implemented based on the method of optimizing the output results when the input parameters contain interval parameters, now instead of using the direct calculation interval arithmetic tool to find the output result range. We perform the optimization of the objective function containing the interval input variables to find the maximum and minimum values with the constrained conditions that the variables of the objective function are limited in the interval of them:

$y_i = f_j(x_1, x_2,\ldots, x_n) \rightarrow \min$, *constrained condition* $a_i \leq x_i \leq b_i$ (8)

$y_i = f_j(x_1, x_2,\ldots, x_n) \rightarrow \max$, *constrained condition* $a_i \leq x_i \leq b_i$ (9)

Solving the problems (8) and (9) to get the minimum and maximum value of the output. Below, the author briefly presents the algorithm to optimize the function containing the interval variable to determine the output in the form of interval number. Using the differential evolution algorithm (DE) [14] set up in the software Mple.17 to solve the programming problem (8) and (9) we get the *maximum and minimum* value of the output result.

2.4 The Differential evolutionary(DE) algorithm[15]

The Differential Evolutionary Algorithm (DEA), developed by Storn and Price [15], is an evolutionary algorithm for solving optimization problems. The general idea of the algorithm is that from a randomly generated population of individuals, new individuals will be generated and selectively compete with the old ones. During this selection process, good individuals will be passed on to the next generations; otherwise, inferior individuals will perish. Here, the instances will be evaluated through an objective function f(x) defined by a particular optimization problem. This process is similar to the process of natural selection described in Darwin's theory of evolution. DEA is an evolved version of the genetic algorithm with the steps of crossover and mutation clearly described by mathematical formulas. Experimentally, DEA is said to have the ability to find the optimal solution very well through mining and exploiting the search space. The DE algorithm is depicted in **Figure 1**.

1: Define the parameters of the algorithm: Number of design variables (D), number of individuals (P), maximum number of iterations (G)
2: Initialize instances of the first population according to (14)
3: **For** g = 1: G
4: Assess the population and identify the best individual xbest
5: **For** i = 1: P
6: Identify the instance mother xi
7: Generate 3 random positive integers r1, r2, r3
8: Determination of mutation coefficient F = N (0.5, 0.22) and the probability of hybridization Cr = 0.8
9: Create mutant vectors that follow (5) or (6)
10: Create vector ci by (7)

```
IF f (ci) <f (xi) THEN xi = ci
IF f (ci) <f (xbest) THEN xbest = ci
End For
End For
Return xbest
```

Figure 1: Differential evolution algorithm

2.5 Algorithm for solving interval linear systems

An interval matrix $[\breve{a}]$ is a matrix whose elements are interval numbers. An interval matrix $[\breve{a}]$ will be written as

$$[\breve{a}]=\begin{bmatrix} \breve{a}_{11} & \breve{a}_{12} & \dots & \breve{a}_{1n} \\ \breve{a}_{21} & \breve{a}_{22} & \dots & \breve{a}_{2n} \\ \dots & \dots & \dots & \dots \\ \breve{a}_{n1} & \breve{a}_{n2} & \dots & \breve{a}_{nn} \end{bmatrix}=[\breve{a}_{ij}]_{nxn};$$

where each $\breve{a}_{ij}=[\underline{a}_{ij},\overline{a}_{ij}]$ for some $\underline{a}_{ij} \le \overline{a}_{ij}$.

The basic full linear system of the interval parameter as:

$$[\breve{a}]\{\breve{x}\}=\{\breve{b}\} \tag{10}$$

Rewriting the equation as follows:

$$\{\breve{x}\}=[\breve{a}]^{-1}.\{\breve{b}\} \tag{11}$$

Expanding equation (11) we have:

$$\{\breve{x}\}=\begin{Bmatrix} \breve{x}_1 \\ \breve{x}_2 \\ \dots \\ \breve{x}_n \end{Bmatrix}=[\breve{a}]^{-1}.\{\breve{b}\}=$$

$$\begin{bmatrix} \breve{a}_{11} & \breve{a}_{12} & \dots & \breve{a}_{1n} \\ \breve{a}_{21} & \breve{a}_{22} & \dots & \breve{a}_{2n} \\ \dots & \dots & \dots & \dots \\ \breve{a}_{n1} & \breve{a}_{n2} & \dots & \breve{a}_{nn} \end{bmatrix}^{-1} \text{x} \begin{Bmatrix} \breve{b}_1 \\ \breve{b}_2 \\ \dots \\ \breve{b}_n \end{Bmatrix} \tag{12}$$

Where $[\breve{\delta}]=[\breve{a}]^{-1}$ is the inverse matrix of $[\breve{a}]$; $[\breve{\delta}]$ is calculated directly by software Maple.17 with the determinant of $[\breve{\delta}]$ is non-zero. Equation system (12) rewrited:

$$\{\breve{x}\}=\begin{Bmatrix} \breve{x}_1 \\ \breve{x}_2 \\ \dots \\ \breve{x}_n \end{Bmatrix}=[\breve{\delta}]\{\breve{b}\}=\begin{bmatrix} \breve{\delta}_{11} & \breve{\delta}_{12} & \dots & \breve{\delta}_{1n} \\ \breve{\delta}_{21} & \breve{\delta}_{22} & \dots & \breve{\delta}_{2n} \\ \dots & \dots & \dots & \dots \\ \breve{\delta}_{n1} & \breve{\delta}_{n2} & \dots & \breve{\delta}_{nn} \end{bmatrix}$$

$$\text{x} \begin{Bmatrix} \breve{b}_1 \\ \breve{b}_2 \\ \dots \\ \breve{b}_n \end{Bmatrix} \tag{13}$$

with $\breve{\delta}_{ij}=\dfrac{1}{\det(\breve{a})}(-1)^{i+j}.\det(\breve{M}_{ij})$;

$\det(\breve{a})$: the determinant of matrix $[\breve{a}]$ is nonnegative;

$\det(\breve{M}_{ij})$: the determinant of $[\breve{M}_{ij}]$; $[\breve{M}_{ij}]$ is the submatrix derived from the matrix $[\breve{a}]$ by removing row i, column j of $[\breve{a}]$.

Equation system (13) converts to a system of linear algebraic equation system:

$$\begin{cases} \breve{x}_1=\breve{\delta}_{11}\breve{b}_1+\breve{\delta}_{12}\breve{b}_2+\dots+\breve{\delta}_{1n}\breve{b}_n \\ \breve{x}_2=\breve{\delta}_{21}\breve{b}_1+\breve{\delta}_{22}\breve{b}_2+\dots+\breve{\delta}_{2n}\breve{b}_n \\ \dots \\ \breve{x}_n=\breve{\delta}_{n1}\breve{b}_1+\breve{\delta}_{n2}\breve{b}_2+\dots+\breve{\delta}_{nn}\breve{b}_n \end{cases} \tag{14}$$

Considering the ith equation of the system of equations (14):

$$\breve{x}_i=\breve{\delta}_{i1}\breve{b}_1+\breve{\delta}_{i2}\breve{b}_2+\dots+\breve{\delta}_{in}\breve{b}_n \tag{15}$$

In equation (15), the left hand side is the i-th interval variable to be found, which is determined from the interval parameters $\breve{\delta}_{ij}$ and $\breve{b}_i$ (i,j = 1,2,…,n). Equation (15) is considered as an interval function that determined the output interval variable. Optimization of the objective function (15) containing the interval input variables to find the maximum and minimum values by using the Differential Evolutionary Algorithm to find the max and min values of $\breve{x}_i$. Doing it for all equations of system (14) will determine all the output variables. The sequence of Algorithm is briefly presented as a block diagram in Figure 2.

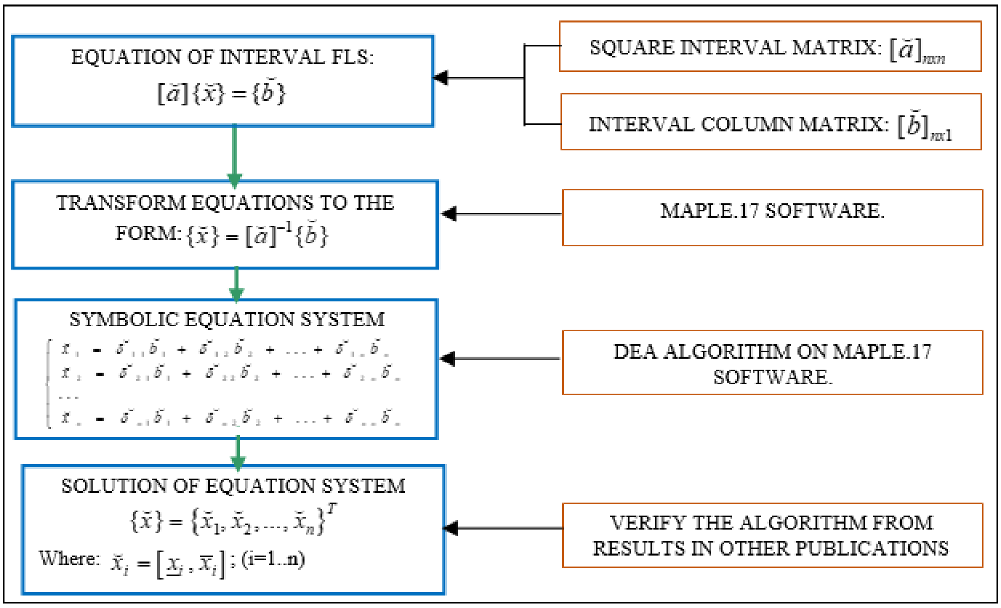

Figure 2: Block diagram of Algorithm steps

3. Numerical Examples

Example 3.1: Consider an interval linear equation system $[\breve{a}]\{\breve{x}\}=\{\breve{b}\}$ discussed in [1, 11], where $[\breve{a}]=$

$$\begin{bmatrix} [3.7,4.3] & [-1.5,-0.5] & [0,0] \\ [-1.5,-0.5] & [3.7,4.3] & [-1.5,-0.5] \\ [0,0] & [-1.5,-0.5] & [3.7,4.3] \end{bmatrix} \in IR^{nxn}$$

and $$\{\breve{b}\}=\begin{Bmatrix} [-14,0] \\ [-9,0] \\ [-3,0] \end{Bmatrix}$$

Applying the proposed algorithm, we obtain a solution set:

$$\{\breve{x}\}=\begin{Bmatrix} \breve{x}_1 \\ \breve{x}_2 \\ \breve{x}_3 \end{Bmatrix}=\begin{Bmatrix} [-6.3776,0] \\ [-6.3983,0] \\ [-3.4047,0] \end{Bmatrix}$$

Comparison study in [1]:

Using interval Gaussian elimination algorithm (IGA), we obtain the solution:

$$\{\breve{x}\}=\begin{Bmatrix} [-6.38,0] \\ [-6.40,0] \\ [-3.40,0] \end{Bmatrix}$$

Using Hansen's technique and Rohn's reformulation, we obtain the same solution:

$$\{\breve{x}\}=\begin{Bmatrix} [-6.38,1.12] \\ [-6.40,1.54] \\ [-3.40,1.40] \end{Bmatrix}$$

Results of methods imlies that the propsed algorithm and the IGA have the same solution. And the solution of proposed algorithm give the smallest box containing the solution set. This allows for confirmation of the reliability of the proposed algorithm.

Example 3.2: Consider an interval linear equation system $[\breve{a}]\{\breve{x}\}=\{\breve{b}\}$, discuss in [8] :

Where $[\breve{a}]=\begin{bmatrix} [2,4] & [-2,1] \\ [-1,2] & [2,4] \end{bmatrix} \in IR^{nxn}$

and $$\{\breve{b}\}=\begin{Bmatrix} [-2,2] \\ [-2,2] \end{Bmatrix}$$

Applying the proposed algorithm, we obtian a solution set:

$$\{\breve{x}\}=\begin{Bmatrix} \breve{x}_1 \\ \breve{x}_2 \end{Bmatrix}=\begin{Bmatrix} [-4,4] \\ [-4,4] \end{Bmatrix}$$

-Using the algorithm in [8], we obtain the same solution:

$$\{\breve{x}\}=\begin{Bmatrix} \breve{x}_1 \\ \breve{x}_2 \end{Bmatrix}=\begin{Bmatrix} [-4,4] \\ [-4,4] \end{Bmatrix}$$

4. Discussion

This study proposed an optimization-based algorithm for solving full interval linear equation systems with interval coefficient matrices and interval right-hand-side vectors. By combining interval function optimization with the Differential Evolution algorithm, the proposed method reduces the excessive interval expansion associated with conventional interval arithmetic while producing reliable strong interval solutions. Numerical examples demonstrate good agreement with published results. Future research may extend the proposed approach to larger-scale systems, singular or nearly singular interval matrices, and more computationally efficient optimization strategies.

Acknowledgments

The research described in this paper was financially supported by the Mien trung University of Civil engineering, Vietnam.

References

[1] Ning. S, Kearfott, R. B. (1997). A comparison of some methods for solving linear interval equations. *SIAM Journal on Numerical Analysis.* Vol 34(4): 1289-1305.
[2] Svetoslav Markov. (1999). An iterative method for algebraic solution to interval equations. *Applied Numerical Mathematics*. 225–239.
[3] Mariusz Pilarek. (2010). Solving systems of linear interval equations using the interval extended zero method and multimedia extensions. *Scientific Research of the Institute of Mathematics and Computer Science*.
[4] Sarangam Majumdar, S. Chakraverty. (2013). Method for the Solution of Interval Systems of Linear Equations. *International Journal of Advances in Applied Sciences (IJAAS*). Vol. 2, No. 2, pp. 73~76.
[5] Shohreh Abolmasoumi, Majid Alavi. (2014). A method for calculating interval linear system. *Journal of mathematics and computer science.*
[6] T. Nirmala, D. Datta, H. S. Kushwaha, K. Ganesan. (2015). On the Solution of System of Interval Linear Equations. *International Journal of Applied Engineering Research.*
[7] Esmaeil Siahlooei, SeyedAbolfazl Shahzadeh Fazeli. (2018). TwoIterative Methods for Solving Linear Interval Systems. *Applied Computational Intelligence and Soft Computing.*
[8] M. Keyanpour, M. Mohaghegh Tabar, W. Lodwick. (2018). A Solution Algorithm for a System of Interval Linear Equations Based on the Constraint Interval Point of View. *Reliable Computing.*
[9] T. Nirmala, K. Ganesan. (2019). Solution of Interval Linear System of Equations-An Iterative Approach. *The 11th National Conference on Mathematical Techniques and Applications.*
[10] T. Nirmala and K. Ganesan. (2021). Solving system of interval linear equations by gauss jordon method using generalized intervalarithmetic. *IOP Conf. Series: Materials Science and Engineering.*
[11] Hema Surya Subramanian, Nirmala Thamaraiselvan, Ganesan Kandasamy. (2023). Implementing Interval Linear Equations Systems for Enhanced Circuit Analysis. *Mathematical Modelling of Engineering Problems.* Vol 10, No.6, 2217-2222.
[12] Gareth I Hargreaves. (2002). Interval analysis in Matlab.
[13] R B Kearfott. (1996). Interval computations introduction uses and resourse. *Euromath Bulletin Journal.*
[14] Hao Zhang.(2005). Nondeterministic linear static finite element analysis. *An Interval Approach, Doctor of Philosophy, School of Civil and Environmental Engineering Georgia Institute of Technology*.
[15] K. V. Price, R. M. Storn, and J.A. Lampinen. (2008). Differential Evolution: A practical Approach to global optimization. *Springer Science & Business, Media, Germany.*

Paper ID: SR26708234828 DOI: https://dx.doi.org/10.21275/SR26708234828