SOLVING SYSTEMS OF LINEAR INTERVAL EQUATIONS USING THE "INTERVAL EXTENDED ZERO" METHOD AND MULTIMEDIA EXTENSIONS

Mariusz Pilarek

Institute of Computer and Information Sciences, Czestochowa University of Technology, Poland
mariusz.pilarek@icis.pcz.pl

Abstract. The aim of this paper is to present the implementation of the method for solving linear interval equations using the "interval extended zero" method and multimedia extensions. The "interval extended zero" method allows us to reduce the undesirable excess width effect. Its efficiency was proven before and here we show that it can be used to perform fast calculations using multimedia extensions. Some numerical examples are used to illustrate the efficiency of our implementation in comparison with several numerical libraries for interval arithmetic.

Introduction

Solving the systems of linear equations is considered as a very important part of numerical analysis. But in real life situations, parameters of these systems often are charged by different kinds of uncertainties [1-3]. Leontief's input output model of economy [4] can be taken as an example [5]. In many cases, uncertainty can be represented by intervals. Since the seminal publication by Moore's [6], a rapid development of interval arithmetic had been observed.

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

$$[A][x]=[b] \tag{1}$$

where $[A]$ is an interval matrix, $[b]$ is an interval vector and $[x]$ is an interval solution vector. Generally such a system has no exact solutions. There are, however, methods for approximate solution of (1). The dominant approaches to the solution of interval linear system are based on the treating (1) as a set of real valued equations whose parameters belong to the corresponding intervals [7]. However, these approaches are NP hard problems, and finding a solution of even a small system is a very difficult task. The undesirable feature of known approaches to the solution of (1) is the excess width effect, i.e., the rapid increasing of resulting intervals width as a consequence of interval calculations.

Some methods for reducing excess width effect with the use of various versions of the stationary single-step iteration method were analyzed by V. Zyuzin [8]. Kupriyanova [9] proved the convergence of this iterative process to the so-called maximal inner solution of problem (1) under special (implicit) restrictions on the input data $[A]$, $[b]$ and on the initial approximation. Markov [10, 11] formulated these restrictions explicitly in context of Jacobi type iterative method for the solution of (1). However, in many cases these methods give inverted intervals as a result (when the lower bound is greater than the upper bound). Another approach to reducing the excess width effect is the solution of (1) based on the concept of „interval extended zero" [12, 13]. It was proven that this method gives narrow interval results. In [14], "interval extended zero" method was presented and used as a modification of classic interval division.

In the current report we show that this method can be used to solve big linear interval systems using the so-called SSE [15] multimedia extensions which improves the efficiency of the calculations. Since interval arithmetic operations are based on the bounds of intervals, it seems natural to make those computations in parallel.

In this paper, we propose the implementation of the solution of the systems of interval linear equations with the use of „interval extended zero" method and SSE extensions.

The rest of the paper is set as follows. In Section 1, we recall the fundamentals of „interval extended zero" method and present its interpretation as the modified interval division. In section 2 we present the application of modified interval division to the solution of interval problem (1) and illustrative examples with comparison to other numerical interval packages. Last section concludes with some remarks.

1. Interval arithmetic and „interval extended zero" method for solving linear equations

Let $X=\left[\underline{x},\overline{x}\right]$ and $Y=\left[\underline{y},\overline{y}\right]$ are intervals and $@\in\{+,-,\cdot,/\}$ then according to [6]:

$$X @ Y=\{z=x @ y, x\in X\wedge y\in Y\} \tag{2}$$

On the base of (2), four interval arithmetic operations can be defined as follows:

$$X+Y=\left[\underline{x}+\underline{y},\overline{x}+\overline{y}\right] \tag{3}$$

$$X-Y=\left[\underline{x}-\overline{y},\overline{x}-\underline{y}\right] \tag{4}$$

$$X \cdot Y = \left[\min\left\{ \underline{x} \cdot \underline{y}, \overline{x} \cdot \underline{y}, \underline{x} \cdot \overline{y}, \overline{x} \cdot \overline{y} \right\}, \max\left\{ \underline{x} \cdot \underline{y}, \overline{x} \cdot \underline{y}, \underline{x} \cdot \overline{y}, \overline{x} \cdot \overline{y} \right\} \right], \tag{5}$$

$$\frac{X}{Y} = \left[\underline{x}, \overline{x} \right] \cdot \left[\frac{1}{\overline{y}}, \frac{1}{\underline{y}} \right], 0 \notin \left[\underline{y}, \overline{y} \right]. \tag{6}$$

One of the problems of interval arithmetic is the excess width effect. In [12, 13], the solution of interval linear equations using the method based on the concept of „interval extended zero" was proposed. This method considerably reduces this undesirable effect. Here we present the basics this method.

Let us consider the simplest equation

$$ax - b = 0 \tag{7}$$

where a, b are real values.
Its conventional interval extension leads to the interval equation

$$[a][x] - [b]=0 \tag{8}$$

which seems to be senseless because its left part represents an interval value, whereas the right part is the non-interval degenerated zero. Formally, when extending Eq. (7) one obtains not only interval on its left hand side, but interval zero on the right hand side. Generally, this interval zero cannot be degenerated interval [0,0].

Therefore, the concepts of "interval zero extension" and "interval zero" has been proposed in [13]. The operation of "interval zero extension" provides an "interval zero" in the right hand side of extended Eq. (8). Since "interval zero" is not a degenerated interval, such approach makes it possible to solve the problem of correct interval extension of Eq. (8).

In conventional interval analysis, it is usually assumed that any interval containing zero may be considered as "interval zero". Let us look to this problem from another point of view. Without loss of generality, we can define the degenerated (usual) zero as the result of operation a-a, where a is any real valued number or variable. Hence, in a similar way we can define an "interval zero" as the result of operation $[a]-[a]$, where $[a]$ is an interval. It is easy to see that for any interval $[a]$ we get $[\underline{a}, \overline{a}] - [\underline{a}, \overline{a}] = [\underline{a} - \overline{a}, \overline{a} - \underline{a}] = [-(\overline{a} - \underline{a}), \overline{a} - \underline{a}]$. Therefore, in any case the result of interval subtraction $[a]$-$[a]$, is an interval centered around 0. Thus, if we want to treat a result of subtraction of two identical intervals as "interval zero", then the most general definition of such "zero" will be "interval zero is an interval symmetrical with respect to 0". It must be emphasized that introduced definition says nothing about the width of "interval zero".

Thus, when extending Eq. (8) with previously unknown values of variables in the left hand side, only what we can say about the right hand side is that it should be an interval symmetrical with respect to 0 with not defined width. Hence, as the result of interval extension of Eq. (8) in general case we get

$$\left[\underline{a},\overline{a}\right]\cdot\left[\underline{x},\overline{x}\right]-\left[\underline{b},\overline{b}\right]=\left[-y,y\right] \tag{9}$$

In fact, the right hand side of Eq. (9) is some interval centered around zero, which can be treated as interval extension of the right hand side of Eq. (8), i.e., as an interval extension of 0. The value of y in Eq. (9) is not yet defined since the values of $\underline{x}, \overline{x}$ are also not defined.

For positive interval values we get

$$\begin{cases}\underline{a}\cdot\underline{x}-\overline{b}=-y,\\ \overline{a}\cdot\overline{x}-\underline{b}=y.\end{cases} \tag{10}$$

From Eq. (10) we obtain only one linear equation with two unknown variables $\underline{x}$ and $\overline{x}$:

$$\underline{a}\underline{x}-\overline{b}+\overline{a}\overline{x}-\underline{b}=0 \tag{11}$$

If there are some constraints on the values of unknown variables $\underline{x}$ and $\overline{x}$, then Eq. (11) with these constraints may be considered as the so-called Constraint Satisfaction Problem (*CSP*) [16] and its interval solution may be obtained. The first constraint on the variables $\underline{x}$ and $\overline{x}$ is a solution of Eq. (11) assuming $\underline{x}=\overline{x}$. In this degenerated case we get the solution of Eq. (11) as $x_m=\dfrac{\underline{b}+\overline{b}}{\underline{a}+\overline{a}}$. It is seen that x_m is the upper bound for $\underline{x}$ and the lower bound for $\overline{x}$. The natural low bound for $\underline{x}$ and upper bound for $\overline{x}$ may be defined using basic definitions of interval arithmetic (6) as $\underline{x}=\dfrac{\underline{b}}{\overline{a}}$, $\overline{x}=\dfrac{\overline{b}}{\underline{a}}$. Thus, we have $\left[\underline{x}\right]=\left[\dfrac{\underline{b}}{\overline{a}},x_m\right]$ and $\left[\overline{x}\right]=\left[x_m,\dfrac{\overline{b}}{\underline{a}}\right]$. These intervals can be narrowed taking into account Eq. (11), which in the spirit of *CSP* is treated as a constraint.

From (11) we get:

$$\underline{x}=\frac{\underline{b}+\overline{b}-\overline{a}\overline{x}}{\underline{a}},\overline{x}\in\left[x_m,\frac{\overline{b}}{\underline{a}}\right],\ \overline{x}=\frac{\underline{b}+\overline{b}-\underline{a}\underline{x}}{\overline{a}},\underline{x}\in\left[\frac{\underline{b}}{\overline{a}},x_m\right]\underline{x}\in\left[\frac{\underline{b}}{\overline{a}},x_m\right] \tag{12}$$

Obviously, when $\overline{x}$ is maximal, i.e., $\overline{x}=\dfrac{\overline{b}}{\underline{a}}$, we get the minimal value of $\underline{x}$, i.e., $\underline{x}_{\min}=\dfrac{\underline{b}+\overline{b}}{\underline{a}}-\dfrac{\overline{a}\overline{b}}{\underline{a}^{2}}$. Similarly, from (12) we get the maximal value of $\overline{x}$, i.e., $\overline{x}_{\max}=\dfrac{\underline{b}+\overline{b}}{\overline{a}}-\dfrac{\underline{a}\cdot\underline{b}}{\overline{a}^{2}}$. Since it is possible that $\underline{x}_{\min}<\dfrac{\underline{b}}{\overline{a}}$ and $\overline{x}_{\max}>\dfrac{\overline{b}}{\underline{a}}$, we get the following interval solution:

$$[\underline{x}]=\left[\underline{x}_{\max},\frac{\underline{b}+\overline{b}}{\underline{a}+\overline{a}}\right],\left[\overline{x}\right]=\left[\frac{\underline{b}+\overline{b}}{\underline{a}+\overline{a}},\overline{x}_{\min}\right], \tag{13}$$

where $\underline{x}_{\max}=\max\left(\dfrac{\underline{b}}{\overline{a}},\dfrac{\underline{b}+\overline{b}}{\underline{a}}-\dfrac{\overline{a}\overline{b}}{\underline{a}^{2}}\right)$, $\overline{x}_{\min}=\min\left(\dfrac{\overline{b}}{\underline{a}},\dfrac{\underline{b}+\overline{b}}{\underline{a}}-\dfrac{\underline{a}\cdot\underline{b}}{\overline{a}^{2}}\right)$

Expressions (13) define all possible solutions of Eq. (9). The values of $\overline{x}_{\min}$, $\underline{x}_{\max}$ constitute the interval which produces the widest interval zero after substitution of them in Eq. (9). In other words, the maximum interval solution's width $w_{\max}=\overline{x}_{\min}-\underline{x}_{\max}$ corresponds to the maximum value of y : $y_{\max}=\dfrac{\overline{a}\overline{b}}{\underline{a}}-\underline{b}$. Substitution of degenerated solution $\underline{x}=\overline{x}=x_m$ in Eq. (9) produces the minimum value of y : $y_{\min}=\dfrac{\overline{a}\overline{b}-\underline{a}\underline{b}}{\underline{a}+\overline{a}}$.

Thus, the formal interval solution (13) factually represents the continuous set of nested interval solutions of Eq. (9). It is shown in [13] that this set of interval solutions can be naturally interpreted as a fuzzy number. We can see that values of y characterize the closeness of right hand side of Eq. (9) to the degenerated zero and minimum value $y_{\min}$ is defined exclusively by interval parameters [*a*] and [*b*]. Hence, the values of y may be considered, in a certain sense, as a measure of interval solution's uncertainty caused by the initial uncertainty of Eq. (9).

Therefore we introduce

$$\alpha=1-\frac{y-y_{\min}}{y_{\max}-y_{\min}} \tag{14}$$

which may be treated as a certainty degree of interval solution of Eq. (9). We can see that α rises from 0 to 1 with decreasing of interval's width from maximum value to 0, i.e., with increasing of solution's certainty. Consequently, the values of α may be treated as labels of α-cuts representing some fuzzy solution of Eq. (9). Finally, the solution is obtained in form of triangular fuzzy number

$$\tilde{x} = \left\{ \underline{x}_{\max}, \frac{\underline{b}+\overline{b}}{\underline{a}+\overline{a}}, \overline{x}_{\min} \right\} \tag{15}$$

The resulting fuzzy solution can be reduced to the interval one using well known defuzzification procedures. In our case, defuzzified left and right boundaries of the solution can be represented as

$$\underline{x}_{def} = \frac{\int_{o}^{1} \underline{x}(\alpha)d\alpha}{\int_{0}^{1} d\alpha}, \quad \overline{x}_{def} = \frac{\int_{o}^{1} \overline{x}(\alpha)d\alpha}{\int_{0}^{1} d\alpha} \tag{16}$$

For example, in the case of [*a*], [*b*]>0, in [12, 13] from (10), (14) and (16) the following expressions have been obtained:

$$\underline{x}_{def} = \frac{\overline{b}}{\underline{a}} - \frac{y_{\max} - y_{\min}}{2\underline{a}}, \quad \overline{x}_{def} = \frac{\underline{b}}{\overline{a}} + \frac{y_{\max} + y_{\min}}{2\overline{a}} \tag{17}$$

On the other hand, we can treat it as an approximate solution of the initial interval equation (8), which in turn can be formally presented in the algebraically equivalent form of interval division [x]=[b]/[a]. Therefore, the solution (17) can be formally treated as the result of modified interval division. Hereinafter, such interval solutions can be treated as the results of modified interval division.

It is shown in [12, 13] that proposed method provides a considerable reducing of resulting interval's length in comparison with that obtained using conventional interval arithmetic rules.

2. Solving interval linear systems using „interval extended zero" method

Interval Gaussian elimination algorithm was used to solve Eq. (1). It can be presented as follows:

a) forward substitution:

$$\left[m_{ji}\right] = \left[a_{ji}\right] / \left[a_{ii}\right], \quad \left[a_{jk}\right] = \left[a_{jk}\right] - \left[m_{ji}\right] \cdot \left[a_{ik}\right] \tag{18}$$

$$\left[b_j\right]=\left[b_j\right]-\left[m_{ji}\right]\cdot\left[b_i\right] \tag{19}$$

where i := 1, 2,...n, j:=i+1, ...,n, k:=i,...n, $0 \notin \left[a_{ii}\right]$

b) backward substitution:

$$[s]=\sum_{j=1}^{j=n}\left[a_{ij}\right]\cdot\left[x_j\right] \tag{20}$$

$$\left[x_i\right]=\frac{\left[b_i\right]-[s]}{\left[a_{ii}\right]} \tag{21}$$

where i := 1, 2,...n, $0 \notin \left[a_{ii}\right]$.

Two implementations of the interval Gaussian algorithm have been developed. The modified interval Gaussian elimination algorithm (*MGEA*) with the use of the “interval extended zero” method and the usual Gaussian elimination algorithm (*UGEA*) with the use of the classic interval division operation. In the implementation of *MGEA*, all division operations in (18) and (21) were replaced with modified interval division defined in the previous Section.

Multimedia SSE extensions [15] were used for implementing both *MGEA* and *UGEA*. These processor extensions were designed by Intel to improve the efficiency of the computations. They contain registers that can store integer and floating point numbers and all operations on them can be performed in parallel.

The potential of multimedia extensions has been quickly recognized by interval community. Since intervals are stored in the memory as two floating point numbers (usually double precision), SSE extensions seem to be a very good solution to increase the efficiency of interval computations. Although first experiments with SSE extensions conducted by von Gudenberg [17] were not enthusiastic, Lambov [18] and Goualard [19], proved that SSE extensions can increase the performance of interval computations.

We have tested our implementations using few examples of randomly generated interval matrices and vectors on AMD Dual-Core Opteron 2.2 GHz machine. Three well known and often used within interval community libraries were taken for comparison with our method: RealLib [20], Profil/BIAS [21] and Boost [22]. RealLib is used mostly for real numbers, but it also contains module for interval arithmetic. Similar to our approach, RealLib is implemented with the use of multimedia SSE extensions. Boost is a library for performing fast calculations on real numbers and intervals. Profil/BIAS is designed strictly for interval arithmetic. All these libraries are implemented with the use of classic interval division operation (6), and for all of them usual Gauss elimination algorithm have been implemented.

To estimate the quality of obtained results, we provide the special relative index of uncertainty, *RIU*. It may serve as the quantitative measure of the excess width effect. It was calculated on resulting vectors as a maximal value from all elements

$$RIU = \frac{\max\left(\left(x_m - \underline{x}\right), \left(\overline{x} - x_m\right)\right)}{x_m} \cdot 100\% \tag{22}$$

where $x_m = \left(\underline{x} + \overline{x}\right)/2$. *RIU* was calculated as a maximum value obtained on the elements of the interval solution vectors.

Table 1 presents the results obtained for 3 randomly generated interval matrices and vectors (containing 2000, 4000 and 6000 rows). The *MGEA* column contains results obtained using modified Gaussian elimination algorithm based on the modified interval division, *UGEA* column contains results obtained using usual Gaussian elimination algorithm with the use of classic interval division (6). Next three columns contain results obtained using RealLib (*RealLib*), Profil/BIAS (*BIAS*) and Boost (*Boost*) libraries respectively. Because *RIU* values were equal for RealLib, Profil/BIAS and Boost libraries, there is only one column containing *RIU* values for all of them.

Table 1

Calculations time and *RIU* values for interval Gaussian algorithm

N	*MGEA*		*UGEA*		*RealLib*		*BIAS*	*Boost*
	RIU,%	*Time, s*	*RIU,%*	*Time, s*	*RIU,%*	*Time, s*	*Time, s*	*Time, s*
2000	14.475	103.259	29.234	103.382	29.234	141.710	446.958	413.571
4000	14.218	1852.325	29.237	1851.594	29.237	3321.393	6470.465	6559.811
6000	14.287	8177.590	29.248	8190.385	29.248	16621.638	27870.882	24160.457

Let us first analyze the *RIU* values. All three libraries (RealLib, Profil/BIAS and Boost) are implemented with the use of the classic interval division, just as our *UGEA* implementation. *RIU* values for all of them are equal (*RIU* values calculated for intervals in the input matrices and vectors were equal to 10%). *MGEA* implementation, using “interval extended zero”, method provide much better results: intervals were more than 50% narrower than ones obtained using classic interval division in all cases.

Obtained calculation times confirm the efficiency of our implementation. RealLib library was two times slower for biggest matrix, whereas both Profil/BIAS and Boost libraries were more than three times slower than *MGEA* and *UGEA* implementations.

We can conclude that *MGEA* and *UGEA* implementations are practically equally effective. This proves, that the “interval extended zero” method does not affect the speed of the interval Gaussian elimination algorithm.

Conclusions

A new approach to solve linear interval equation systems based on the concepts of "interval extended zero" method using multimedia SSE extensions is presented. It is shown that this method can be naturally treated as a modified interval division and used in practical applications. This method not only allows us to reduce the undesirable excess width effect, but also perform fast interval computations using the multimedia SSE extensions. Some examples based of randomly generated interval matrices and vectors were used to prove the efficiency of our implementation in solving interval linear systems. The computations using our algorithms *MGEA* and *UGEA* are considerably faster than those of popular RealLib, Profil/BIAS and Boost libraries. Furthermore, the "interval extended zero" method does not affect the speed of the calculations. Both *MGEA* and *UGEA* are equally effective.

References

[1] Buckley J.J., Solving fuzzy equations in economics and finance, Fuzzy Sets & Systems 1992, 48.

[2] Buckley J.J., The fuzzy mathematics of finance, Fuzzy Sets & Systems 1987, 21, 257-273.

[3] Chen S.H., Yang X.W., Interval finite element method for beam structures, Finite Elements in Analysis and Design 2000, 34, 75-88.

[4] Leontief W., Quantitative input output relations in the economic system of the United States, Review of Economics and Statistics 18, 1396, 100-125.

[5] Wu C.C., Chang N.B., Grey input output analysis and its application for environmental cost allocation, European Journal of Operational Research 2003, 145, 175-201.

[6] Moore R.E., Interval Arithmetic and Automatic Error Analysis in Digital Computing. PhD thesis, Stanford University 1962.

[7] Kearfott B., Rigorous Global Search: Continuous Problems, Kluwer Academic Publishers, The Netherlands 1996.

[8] Zyuzin V., An iterative method for solving system of algebraic segment equations of the first order, Differential Equations and the Theory of Functions, Saratov State University, Saratov 1990, 72-82 (in Russian).

[9] Kupriyanova L., Inner estimation of the united solution set to interval linear algebraic system, Reliable Computing 1995, 1(1) 15-31.

[10] Markov S., An iterative method for algebraic solution to interval equations, Applied Numerical Mathematics 1999, 30, 225-239.

[11] Markov S., Popova E., Ullrich C., On the Solution of Linear Algebraic Equations Involving Interval Coefficients, Iterative Methods in Linear Algebra II, IMACS Series in Computational and Applied Mathematics 1996, 4, 216-225.

[12] Dymova L., Sevastjanov P., Fuzzy solution of interval linear equations, Lecture Notes in Computer Science 2008, 4697, 1392-1399.

[13] Dymova L., Sevastjanov P., A new method for solving interval and fuzzy equations: linear case, Information Sciences 2009, 17, 925-937.

[14] Dymova L., Pilarek M., Wyrzykowski R., Solving the systems of interval linear equations with use of modified interval dividing procedure, Lecture Notes in Computer Science 2010, 6068.

[15] Intel SSE4 Programming Reference, http://software.intel.com/file/18187/

[16] Cleary J.C., Logical arithmetica. Future Computing Systems 1987, 2, 125-149.

[17] Wolff von Gudenberg J., Interval Arithmetic on Multimedia Architectures, RR 265, Lehrstuhl fur Informatik II. Universitat Wurzburg, October 2000.

[18] Lambov B., Interval Arithmetic using SSE 2, Lecture Notes in Computer Science 2008, 5045, 102-113.

[19] Goualard F., Fast and Correct SIMD Algorithms for Interval Arithmetic, INRIA, 2008, http://hal.archives-ouvertes.fr/docs/00/28/84/56/PDF/intervals-sse2-long-paper.pdf

[20] Lambov B., RealLib 3 Manual, http://www.brics.dk/ barnie/RealLib/RealLib.pdf

[21] Knuppel O., Profil/BIAS v2.0, http://www.ti3.tu-harburg.de/keil/profil/Profil2.ps.gz

[22] Boost Interval Arithmetic Library,
http://www.boost.org/doc/libs/1_43_0/libs/numeric/interval/doc/interval.htm