IEICE Proceeding Series

All Solution Algorithm for Parameter-Dependent Nonlinear Equations Using Affine Arithmetic

Kosuke Ihara, Yuchi Kanzawa

Vol. 1 pp. 419-422
Publication Date: 2014/03/17
Online ISSN: 2188-5079

2012 International Symposium on Nonlinear Theory and its Applications
NOLTA2012, Palma, Majorca, Spain, October 22-26, 2012

All Solution Algorithm for Parameter-Dependent Nonlinear Equations Using Affine Arithmetic

Kosuke Ihara† and Yuchi Kanzawa†

†Shibaura Institute of Technology 3-7-5 Toyosu Koto-ku, Tokyo 135-8548, Japan
Email: ma11021@shibaura-it.ac.jp, kanzawa@sic.shibaura-it.ac.jp

Abstract—A new all solution algorithm is proposed for parameter-dependent nonlinear equations. In this algorithm, affine arithmetic[1], which is more accurate than interval arithmetic[2], is used for an existence test and two non-existence tests of a solution. The efficiency of the proposed algorithm is verified by some numerical examples.

1. Introduction

In this paper, we consider the parameter-dependent nonlinear equations

$$f : D \subset \mathbf{R}^{n+m} \to \mathbf{R}^n, \quad f(x) = 0, \quad x \in \mathbf{R}^{n+m} \tag{1}$$

in a given domain $T \subset D$. Assume that f is continuously differentiable and can be estimated by interval arithmetic (IA)[2] and affine arithmetic (AA)[1].

Algorithms based on IA are well-known computational methods for finding all solutions of Eq.(1)(e.g., [2]). In this method, we need to take the hyper-rectangle as the test domain that is parallel to axes, and we need to select m parameters from $(n + m)$ variables. Subsequently, we guarantee the existence of a solution with IA. However, the main drawback of this method is that it involves extremely high computational costs. To improve this algorithm, it is necessary to develop an efficient existence test for a solution. An efficient non-existence test using AA[3], proposed for Eq.(1) with $m = 0$, could be used with $m > 0$. Although an existence test using AA for Eq.(1) with $m = 0$ was proposed[4], it cannot be used for Eq.(1) with $m > 0$.

The purpose of this paper is to propose an existence test using AA for Eq.(1) with $m > 0$ and to propose an all solution algorithm for Eq.(1), including the proposed existence test. First, we present an existence theorem for Eq.(1) using AA, where parameter selection is not needed; in contrast, parameter selection is needed in the test using IA. Then, we propose an all solution algorithm including the proposed existence test and the non-existence test proposed in [3]. Furthermore, we consider an all solution algorithm including two existence tests, one using AA and the other using IA. By developing an efficient existence test for a solution using AA, which is more accurate than that using IA, we can efficiently find all solutions for Eq.(1). Finally, we verify the efficiency of the proposed algorithm through some numerical examples.

The remainder of this paper is organized as follows. In section 2, we explain some notations and definitions for AA, and we introduce a non-existence test using AA. In section 3, we propose an existence theorem for a solution of Eq.(1) using AA and an all solution algorithm using AA. In section 4, we present some numerical examples. Finally, in section 5, we conclude the paper.

2. Preliminaries

In this section, we briefly explain the notations and definitions that will be used in this paper, and we introduce a non-existence test using AA.

2.1. Affine arithmetic

The essential drawback of IA is that it causes unexpected overestimation of calculation results because correlation between quantities is ignored. In order to overcome this problem, affine arithmetic (AA)[1], an extension of IA, was proposed in 1994. AA is a variant of IA, which maintains correlation between quantities. Thus, AA is able to avoid the extreme increase in interval width often observed in IA.

In AA, a quantity x is represented in the affine form[1]

$$x = x_0 + \sum_{i=1}^{n} x_i \varepsilon_i + x_{n+1}\delta, \quad (x_i \in \mathbf{R}), \tag{2}$$

where $\varepsilon_i (i = 1, \ldots, n)$ and δ are symbolic real variables in the interval $[-1, 1]$. Conventional AA[1] adds a new term ε_{n+1} with each nonlinear operation, whereas in this paper, we sum up such terms into δ. Vector whose elements are affine forms is called affine form vector. Interval $[x]$ is transformed into the following affine form:

$$x = \frac{\underline{x} + \overline{x}}{2} + \frac{\overline{x} - \underline{x}}{2}\varepsilon_1, \tag{3}$$

where $\underline{x} \in \mathbf{R}$ and $\overline{x} \in \mathbf{R}$ are the lower and upper endpoint of the $[x]$, respectively. The affine form expressed in Eq.(2) can always be returned to an interval by the following formula:

$$[x_0 - \Delta, x_0 + \Delta], \quad \Delta = \sum_{i=1}^{n+1} |x_i|. \tag{4}$$

For affine forms

$$x = x_0 + x_1\varepsilon_1 + \cdots + x_n\varepsilon_n + x_{n+1}\delta, \tag{5}$$
$$y = y_0 + y_1\varepsilon_1 + \cdots + y_n\varepsilon_n + y_{n+1}\delta, \tag{6}$$

linear operations are defined naturally as follows:

$$x \pm y = (x_0 \pm y_0) + \sum_{i=1}^{n} (x_i \pm y_i)\varepsilon_i + (x_{n+1} + y_{n+1})\delta. \tag{7}$$

In this paper, we consider affine arithmetic with modified multiplication (AAM)[5] and affine arithmetic with narrowest multiplication (AAN)[6].

The non-existence test for Eq.(1) with $m = 0$ proposed in [3] can be easily used for Eq.(1) with $m > 0$ as follows:

Theorem 2.1 [3] Let $f : D \subset \mathbf{R}^{n+m} \to \mathbf{R}^n$, and let $I \in \mathbf{R}^{n+m} \subset D$ be an affine form vector. If

$$0 \notin f(I) \tag{8}$$

holds, the solution of Eq. (1) does not exist in I.

3. All solution algorithm using affine arithmetic

In this section, we present an existence theorem for a solution of Eq.(1) using AA, and we propose an all solution algorithm for Eq.(1).

We propose an existence theorem for a solution of Eq.(1) using AA as follows:

Theorem 3.1 Let $f : D \subset \mathbf{R}^{n+m} \to \mathbf{R}^n$, $c \in \mathbf{R}^{n+m}$, $s \in \mathbf{R}^m_+$, and $t \in \mathbf{R}^n_+$. Let $W \in \mathbf{R}^{(n+m)\times n}$ and $V \in \mathbf{R}^{(n+m)\times m}$ be composed of the left n columns and right m columns of an $(n+m)$-dimensional non-singular matrix, respectively. Let $\mathrm{diag}(t) \in \mathbf{R}^{n\times n}$ and $\mathrm{diag}(s) \in \mathbf{R}^{m\times m}$ be the diagonal matrices

$$\mathrm{diag}(t) = \begin{pmatrix} t_1 & & 0 \\ & \ddots & \\ 0 & & t_n \end{pmatrix} \tag{9}$$

and

$$\mathrm{diag}(s) = \begin{pmatrix} s_1 & & 0 \\ & \ddots & \\ 0 & & s_m \end{pmatrix}, \tag{10}$$

respectively. Let $\varepsilon^{(n)} \subset \mathbf{R}^n$ and $\varepsilon^{(m)} \subset \mathbf{R}^m$ be the affine form vectors

$$\varepsilon^{(n)} = (\varepsilon_1, \cdots, \varepsilon_n)^{\mathsf{T}} \tag{11}$$

and

$$\varepsilon^{(m)} = (\varepsilon_{n+1}, \cdots, \varepsilon_{n+m})^{\mathsf{T}}, \tag{12}$$

respectively. If $R \in \mathbf{R}^{n\times n}$ is a non-singular matrix and

$$\varepsilon^{(n)} - \mathrm{diag}^{-1}(t)Rf(c + W\mathrm{diag}(t)\varepsilon^{(n)} + V\mathrm{diag}(s)\varepsilon^{(m)}) \subset \varepsilon^{(n)} \tag{13}$$

holds, then there exists a solution for Eq.(1) not only in affine form vector $c + W\mathrm{diag}(t)\varepsilon^{(n)} + V\mathrm{diag}(s)\varepsilon^{(m)}$ but also in $c + W\mathrm{diag}(t)\varepsilon^{(n)} - Rf(c + W\mathrm{diag}(t)\varepsilon^{(n)} + V\mathrm{diag}(s)\varepsilon^{(m)}) + V\mathrm{diag}(s)\varepsilon^{(m)}$.

Although V and W have optionality in Theorem 3.1, the simplicity of an existence guarantee of a solution is considered, and it is set as follows. Let $V \in \mathbf{R}^{(n+m)\times m}$ be composed of an orthonormal basis that spans the null space of $f'(c)$, i.e.,

$$\begin{cases} f'(c)V = O, \\ V^{\mathsf{T}}V = E, \end{cases} \tag{14}$$

where O is an $n\times m$ zero matrix and E is an $m\times m$ identity matrix. Let $W \in \mathbf{R}^{(n+m)\times n}$ be composed of an orthonormal basis that spans the null space of V^{T}, i.e.,

$$\begin{cases} V^{\mathsf{T}}W = \tilde{O}, \\ W^{\mathsf{T}}W = \tilde{E}, \end{cases} \tag{15}$$

where $\tilde{O}$ is an $m\times n$ zero matrix and $\tilde{E}$ is an $n\times n$ identity matrix.

Similarly, although s and t have optionality, they must be determined such that the domain $c + W\mathrm{diag}(t)\varepsilon^{(n)} + V\mathrm{diag}(s)\varepsilon^{(m)}$ includes interval $I \in \mathbf{R}^{n+m}$ arising from the division in the all solution algorithm. In order to take the optimal domain, let s_i be the length of the projection of I to the i-th column vector of V, i.e.,

$$s_i = \sum_{j=1}^{n+m} \mathrm{rad}(I_j)|V_{j,i}|, \ (i = 1, \ldots, m), \tag{16}$$

where $\mathrm{rad}(I_j)$ is the radius of the interval I_j. and let t_i be the length of the projection of I to the i-th column vector of W, i.e.,

$$t_i = \sum_{j=1}^{n+m} \mathrm{rad}(I_j)|W_{j,i}|, \ (i = 1, \ldots, n). \tag{17}$$

A conceptual diagram is shown in Fig.1. Fig.1 shows that the affine form vector I' includes the test domain I of the interval vector using V, W, s, and t. Note that the hyper-parallelohedron represented as the affine form vector with V, W, s, and t cannot be represented by an interval vector.

Using the proposed existence theorem 3.1 and the non-existence test 2.1, we propose an all solution algorithm using AA as follows:

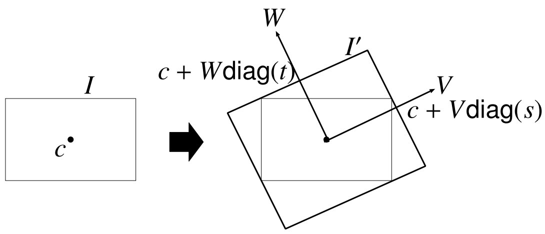

Figure 1: Conceptual diagram of the test domain

Algorithm 3.1 Let $f : D \subset \mathbf{R}^{n+m} \rightarrow \mathbf{R}^n$, and let the hyper-rectangular domain $T \subset D$ be the initial domain of search.

Step 1: The list $\mathcal{L}$ of the search intervals is initialized as $\mathcal{L} = \{T\}$.

Step 2: If $\mathcal{L}$ is empty, end. If not, the head element of $\mathcal{L}$ is defined as I. I is deleted from $\mathcal{L}$.

Step 3: If $0 \notin f(I)$ holds, the solution does not exist in I; go to Step 2.

Step 4: Let $c = \mathrm{mid}(I)$, where $\mathrm{mid}(I)$ is the midpoint of the I. If $0 \notin f(c) + f'(I)(I - c)$ holds, the solution does not exist in I; go to Step 2.

Step 5: Let interval vector I be transformed into an affine form vector. If $0 \notin f(I)$ holds, the solution does not exist in I; go to Step 2.

Step 6: Calculate $V \in \mathbf{R}^{(n+m)\times m}$ from Eq.(14).

Step 7: Calculate $W \in \mathbf{R}^{(n+m)\times n}$ from Eq.(15).

Step 8: Let $R \simeq (f'(c)W)^{-1}$. If it does not exist at all, go to Step 12.

Step 9: Calculate $s \in \mathbf{R}^m$ and $t \in \mathbf{R}^n$ from Eq.(16) and Eq.(17), respectively.

Step 10: Let $t'_i = \max[t_i, \max_{\beta\in\{-1,1\}} \{2Rf(c + V_j s_j \beta)\}_i]$, $i = 1, \ldots, n$, and $j = 1, \ldots, m$. Let $I' = c + W\mathrm{diag}(t')\varepsilon^{(n)} + V\mathrm{diag}(s)\varepsilon^{(m)}$.

Step 11: If $\varepsilon^{(n)} - \mathrm{diag}^{-1}(t')Rf(I') \subset \varepsilon^{(n)}$ holds, there exists a solution in I'; go to Step 2. If $\varepsilon^{(n)} - \mathrm{diag}^{-1}(t')Rf(I') \cap \varepsilon^{(n)} = \emptyset$ holds, the solution does not exist in I'; go to Step 2. Otherwise, go to Step 13.

Step 12: Let the affine form vector $I'' = I' - Rf(c + W\mathrm{diag}(t)\varepsilon^{(n)} + V\mathrm{diag}(s)\varepsilon^{(m)})$ be transformed into the interval vector. Let $I = I \cap I''$. I is divided into two intervals I_1 and I_2, and they are added to the tail of $\mathcal{L}$. Then, go to Step 2. □

In this algorithm, the affine form vector I', which wraps the test domain I, is parallel to each column vector of V, that is, the linearized variety of the expected solution, and highly accurate AA is employed for the existence guarantee of a solution. Therefore, the proposed algorithm is expected to be more efficient than the conventional method.

4. Numerical Examples

In this section, to verify the efficiency of the proposed method(Algorithm 3.1), some numerical examples are implemented for the following algorithms.

Algorithm A An algorithm including an existence test using IA and three non-existence tests using IA.

Algorithm B An algorithm including an existence test using AAM, two non-existence tests using AAM, and two non-existence tests using IA(Algorithm 3.1).

Algorithm C An algorithm including an existence test using AAN, two non-existence tests using AAN, and two non-existence tests using IA(Algorithm 3.1).

Algorithm D An algorithm including two existence tests, one using AAM and the other using IA, two non-existence tests using AAM, and three non-existence tests using IA.

Algorithm E An algorithm including two existence tests, one using AAN and the other using IA, two non-existence tests using AAN, and three non-existence tests using IA.

Our computer environment has the following specifications. CPU: Pentium Dual-Core 2.0 GHz, memory: 2.5 GB, OS: Free BSD 6.3, and compiler: g++ 3.4.6. In Tables 5 and 6, the notation "—" indicates that memory over occurred.

Experiment 4.1 We consider the nonlinear equations

$$\begin{cases} v_{\mathsf{in}} - 3.2g_1(v_1) - (v_1 + v_2) = 0, \\ g_1(v_1) - g_2(v_2) = 0, \end{cases} \tag{18}$$

where

$$\begin{cases} g_1(v_1) = 0.43v_1^3 - 2.69v_1^2 + 4.56v_1 + \alpha_1, \\ g_2(v_2) = 2.5v_2^3 - 10.5v_2^2 + 11.8v_2 + \alpha_2, \end{cases} \tag{19}$$

which is derived from [2], and g_1 and g_2 are fluctuated with α. First, α is fixed at 0, that is, Eq.(18) is one-parameter dependent. For the initial domain $(v_{\mathsf{in}}, v_1, v_2) \in ([0, 20], [0.5], [0, 5])$, the number of searched domains and the computation time for each algorithm are listed in Tables 1 and 2, respectively. Next, α is considered to be in an internal, that is, Eq.(18) is three-parameters dependent. For the initial domain $(v_{\mathsf{in}}, v_1, v_2, \alpha_1, \alpha_2) \in ([0, 20], [0.5], [0, 5], [-0.05, 0.05], [-0.05, 0.05])$, the number of searched domains and the computation time for each algorithm are listed in Tables 3 and 4, respectively. □

Table 1: Number of searched domains of the efficiencies for Eq.(18) with $(\alpha_1, \alpha_2) = (0, 0)$

Number of searched domains				
A	B	C	D	E
166595	4857	3560	4857	3560

Table 2: Computation time of the efficiencies for Eq.(18) with $(\alpha_1, \alpha_2) = (0, 0)$

Computation time(s)				
A	B	C	D	E
4.048×10^1	2.487×10^0	5.605×10^0	2.956×10^0	5.966×10^0

Table 3: Number of searched domains of the efficiencies for Eq.(18) with $(\alpha_1, \alpha_2) = ([-0.05, 0.05], [-0.05, 0.05])$

Number of searched domains				
A	B	C	D	E
234329	44392	15208	1402	845

Table 4: Computation time of the efficiencies for Eq.(18) with $(\alpha_1, \alpha_2) = ([-0.05, 0.05], [-0.05, 0.05])$

Computation time(s)				
A	B	C	D	E
9.058×10^1	4.124×10^1	5.933×10^1	1.469×10^0	3.269×10^0

Experiment 4.2 We consider the m-parameter dependent nonlinear equations

$$g(x_i)+x_1+x_2+\cdots+x_{i+j}-i = 0, \quad i = 1, \ldots, n, \quad j = 1, \ldots, m, \tag{20}$$

where

$$g(x_i) = 2.5x_i^3 - 10.5x_i^2 + 11.8x_i.$$

Let the initial domain be $x_k = [-2.0, 2.0](k = 1, \ldots, n+m)$, and let n and m vary. First, let $m = 1$, that is, Eq.(20) is one-parameter dependent. For $n \in \{2, \ldots, 11\}$, the number of searched domains and the computation time for each algorithm are listed in Tables 5 and 6, respectively. Similarly, let $m = 2, 3, 4$, and 5, that is, Eq.(20) is two, three, four, and five-parameter dependent, respectively. For $m \in \{2, 3, 4, 5\}$, the number of searched domains for each algorithm are listed in Tables 7, 9, 11 and 13, respectively. The corresponding computation time for $m \in \{2, 3, 4, 5\}$ are listed in Tables 8, 10, 12 and 14, respectively. □

Table 5: Number of searched domains of the efficiencies for Eq.(20) with $m = 1$

n	Number of searched domains				
	A	B	C	D	E
2	68	75	69	75	69
3	7562	510	369	510	369
4	49621	3108	2104	3108	2104
5	620189	24677	15765	24677	15765
6	2865907	122923	75200	122923	75200
7	9170371	387857	256058	387857	256058
8	—	1069459	752765	1069459	752765
9	—	2236031	1907621	2236031	1907621
10	—	4492597	4481557	4492597	4481557
11	—	11256385	11256226	11256385	11256226

Table 6: Computation time of the efficiencies for Eq.(20) with $m = 1$

n	Computation time(s)				
	A	B	C	D	E
2	1.442×10^{-1}	3.220×10^{-2}	8.215×10^{-2}	3.794×10^{-2}	8.591×10^{-2}
3	2.460×10^0	3.564×10^{-1}	8.651×10^{-1}	4.362×10^{-1}	9.181×10^{-1}
4	2.527×10^1	3.285×10^0	8.684×10^0	4.078×10^0	9.237×10^0
5	4.889×10^2	3.833×10^1	1.061×10^2	4.905×10^1	1.129×10^2
6	3.241×10^3	2.666×10^2	7.870×10^2	3.491×10^2	8.370×10^2
7	1.469×10^4	1.145×10^3	3.978×10^3	1.529×10^3	4.227×10^3
8	—	4.209×10^3	1.699×10^4	5.734×10^3	1.800×10^4
9	—	1.170×10^4	6.025×10^4	1.603×10^4	6.387×10^4
10	—	3.075×10^4	1.959×10^5	4.268×10^4	2.076×10^5
11	—	9.673×10^4	6.557×10^5	1.397×10^5	6.854×10^5

In all tables, Algorithm B is faster than Algorithm C and Algorithm D is faster than Algorithm E, which shows the low computational cost of AAM beats the high accuracy of AAN. In Tables 2 and 6, Algorithm B is the fastest. In Tables 4, 12, and 14, Algorithm D is the fastest. In Tables 8 and 10, Algorithm D is the fastest in $n = 1$ while Algorithm B is the fastest in $n \geq 2$. From these results, algorithms including the proposed existence test using AA, whatever these are with the one using IA or not, have less

Table 7: Number of searched domains of the efficiencies for Eq.(20) with $m = 2$

n	Number of searched domains				
	A	B	C	D	E
1	68	413	372	7	276
2	881939	6197	3065	6197	3065
3	—	69178	29585	69178	29585
4	—	544141	216512	544141	216512
5	—	3186594	1240287	3186594	1240287
6	—	—	4879212	—	4879212

Table 8: Computation time of the efficiencies for Eq.(20) with $m = 2$

n	Computation time(s)				
	A	B	C	D	E
1	8.292×10^{-3}	1.212×10^{-1}	2.357×10^{-1}	3.751×10^{-3}	1.836×10^{-1}
2	2.267×10^{2}	3.392×10^{0}	4.671×10^{0}	4.006×10^{0}	4.968×10^{0}
3	—	6.091×10^{1}	8.527×10^{1}	7.504×10^{1}	9.149×10^{1}
4	—	7.215×10^{2}	1.055×10^{3}	9.254×10^{2}	1.133×10^{3}
5	—	6.008×10^{3}	9.715×10^{3}	8.095×10^{3}	1.053×10^{4}
6	—	—	5.741×10^{4}	—	6.279×10^{4}

Table 9: Number of searched domains of the efficiencies for Eq.(20) with $m = 3$

n	Number of searched domains				
	A	B	C	D	E
1	834	17664	8286	27	17
2	—	705042	263489	705042	263489

Table 10: Computation time of the efficiencies for Eq.(20) with $m = 3$

n	Computation time(s)				
	A	B	C	D	E
1	1.150×10^{-1}	6.602×10^{0}	6.615×10^{0}	1.269×10^{-2}	1.534×10^{-2}
2	—	4.789×10^{2}	4.817×10^{2}	5.121×10^{2}	5.684×10^{2}

Table 11: Number of searched domains of the efficiencies for Eq.(20) with $m = 4$

n	Number of searched domains				
	A	B	C	D	E
1	14066	707392	248652	171	103

Table 12: Computation time of the efficiencies for Eq.(20) with $m = 4$

n	Computation time(s)				
	A	B	C	D	E
1	2.301×10^{0}	2.583×10^{2}	3.307×10^{2}	9.412×10^{-2}	1.065×10^{-1}

Table 13: Number of searched domains of the efficiencies for Eq.(20) with $m = 5$

n	Number of searched domains				
	A	B	C	D	E
1	436653	—	—	1678	1323

Table 14: Computation time of the efficiencies for Eq.(20) with $m = 5$

n	Computation time(s)				
	A	B	C	D	E
1	7.556×10^{1}	—	—	1.038×10^{0}	1.574×10^{0}

the numbers of searched domains and lower computational time than the conventional algorithm(Algorithm A). Mortifyingly, the proposed existence test didn't always beat the conventional one. The reason why Algorithm D and E are faster than Algorithm B and C is considered to be that the existence test using IA was more accurate than that using AA in some domains. However, the case that the proposed existence test lose the conventional one is only low dimension. From Tables 8 and 10, it is considered that Algorithm B is the fastest in high-dimensional problems. In order to exemplify this, higher-dimensional case in Tables 12 and 14 will be experienced as the future work.

5. Conclusion

In this paper, a new all solution algorithm was proposed for parameter-dependent nonlinear equations using AA. For five all solution algorithms, some numerical examples were presented. We showed that an algorithm including the proposed existence test using AA is more efficient than the conventional algorithm, with the help of numerical examples. However, an algorithm including only the proposed existence test using AA was not efficient for all problems. Accordingly, it is necessary to propose an efficient existence test using AA and to propose an algorithm including it.

References

[1] Andrade, M. V. A., Comba, J. L. D., Stolfi, J.: "Affine Arithmetic", INTERVAL'94, St.Petersburg(Russia), March 5-110 (1994).

[2] Kanzawa, Y., Kashiwagi, M., Oishi, S.: "Algorithm for finding all solutions of parameter-dependent nonlinear equations with guaranteed accuracy", Electronics and Communications in Japan, Vol.82, No.10, pp.33–39 (1999).

[3] Kikuchi, T., Kashiwagi, M.: "Elimination of non-existence regions of the solution of nonlinear equations using affine arithmetic", Proc. NOLTA2001, (2001).

[4] Miyazima, S., Kashiwagi, M.: "Existence test for solution of nonlinear systems applying affine arithmetic", Journal of Computational and Applied Mathematics, Vol.199, No.2, pp.304–309 (2007).

[5] Miyazima, S., Miyata, T., Shirai, T., Kashiwagi, M.: " On the Multiplication and the Division of the Affine Arithmetic", The Transactions of the Institute of Electronics, Information and Communication Engineers, Vol.J86-A, No.3, pp.232–240 (2003).

[6] Miyazima, S., Miyata, S., Kashiwagi, M.: "On the Best Multiplication of the Affine Arithmetic", The Transactions of the Institute of Electronics, Information and Communication Engineers, Vol.J86-A, No.2, pp.150–159 (2003).