Interval Computations
No 2(4), 1992

A NEW CLASS OF ALGORITHMS FOR OPTIMAL SOLUTION OF INTERVAL LINEAR SYSTEMS

Sergey P.Shary

In this paper a new class of *sequentially guaranteeing* algorithms is constructed for finding optimal solutions of interval linear algebraic systems. Estimates of the computational complexity are given for the algorithms developed.

НОВЫЙ КЛАСС АЛГОРИТМОВ ДЛЯ ОПТИМАЛЬНОГО РЕШЕНИЯ ИНТЕРВАЛЬНЫХ ЛИНЕЙНЫХ СИСТЕМ

С.П.Шарый

В работе построен новый класс *последовательно гарантирующих* алгоритмов для нахождения оптимальных решений интервальных систем линейных алгебраических уравнений. Указаны оценки вычислительной сложности этих алгоритмов.

Let the following objects be given: $\mathbf{A}$ an interval $n \times n$-matrix and $\mathbf{b}$, an interval n-vector. The set of all possible solutions of real systems $Ax = b$ where $A \in \mathbf{A}$ and $b \in \mathbf{b}$, that is the set

$$X^* = X^*(\mathbf{A}, \mathbf{b}) = \{x \in \mathbb{R}^n \mid (\exists A \in \mathbf{A})(\exists b \in \mathbf{b})(Ax = b)\}$$

is called, as is well known, a *united solution set* (USS) to an interval linear algebraic system (ILAS), formally written as

$$\mathbf{A}x = \mathbf{b}.$$

The purpose of this paper is to introduce a new computational approach to the solution of the so called "outer problem" for ILAS (according to the terminology of [4, 28] et al), stated traditionally in the following form:

find an interval vector $\mathbf{V}$ that
contains the USS of the given ILAS.

Generally, here the nonsingularity of all real matrices $A \in \mathbf{A}$ is assumed. This guarantees the boundedness of the united solution set. It is easy to understand that the "outer problem" for ILAS is equivalent to the requirement to find a lower estimate of $\min\{x_k \mid x \in X^*(\mathbf{A}, \mathbf{b})\}$ and an upper estimate of $\max\{x_k \mid x \in X^*(\mathbf{A}, \mathbf{b})\}$ for $k = 1, 2, \ldots, n$. Later, we shall concentrate on the computation of $\min\{x_k \mid x \in X^*(\mathbf{A}, \mathbf{b})\}$ (for an arbitrary but fixed component k, in what follows), since

$$\max\{x_k \mid x \in X^*(\mathbf{A}, \mathbf{b})\} = -\min\{x_k \mid x \in X^*(\mathbf{A}, -\mathbf{b})\}.$$

Of particular interest will be optimal (exact) estimates of the USS, since these estimates are the most valuable both in theory and practice. The "outer problem" for ILAS is one of the classical problems of interval analysis, and hundreds of publications are devoted to different aspects of its solution, from the early 1960's to now. Nevertheless, at present, only a few practical algorithms (see [12,25,28]) guarantee the optimality of solutions to the "outer problem" obtained by them in the general case, and the complexity of these algorithms is very high.

Beeck in [12] seems to be the first to point out the remarkable fact that the exact values $\min\{x_k \mid x \in X^*(\mathbf{A}, \mathbf{b})\}$ and $\max\{x_k \mid x \in X^*(\mathbf{A}, \mathbf{b})\}$, $k = 1, 2, \ldots, n$, are attained in the extreme values of the matrix $A \in \mathbf{A}$ and the vector $b \in \mathbf{b}$ (see also [23]). Based on this result, Nickel in [23] proposed a method for solving the "outer problem" for ILAS. He used brute force to look over all possible combinations of endpoints of elements of $\mathbf{A}$ and $\mathbf{b}$ and subsequently solved the real systems $Ax = b$ resulting from this process. The practical significance of this algorithm is not considerable because of disastrous growth of its computational complexity as the dimension of the ILAS in question increases.

Thus, even for systems with a 5×5 matrix, in the general case, there are $2^{5\times(5+1)} = 2^{30} \cong 10^9$ real linear systems of the same dimension to solve. Furthermore, Nickel's algorithm is only *finally guaranteeing*, i.e., the algorithm is certain to give outer component-wise estimates of

the USS only after it terminates, that is, after it examines all possible combinations of endpoints of intervals of the ILAS $\mathbf{A}x = \mathbf{b}$. We shall demonstrate below how one can transform this impractical approach into an attractive computational procedure.

Let us fix some method E for solving the "outer problem" for the ILAS (call it a *basic method*), and let $E(\mathbf{U},\mathbf{v})$ be an interval solution vector of the "outer problem" to the system $\mathbf{U}x = \mathbf{v}$, $E(\mathbf{U},\mathbf{v}) \supseteq X^*(\mathbf{U},\mathbf{v})$, obtained by this method. Then the left endpoint of its k-th component, denoted by $E_k(\mathbf{U},\mathbf{v})$, provides a numerical lower estimate to the sought for value $\min\{x_k \mid x \in X^*(\mathbf{U},\mathbf{v})\}$. We can take any known algorithm for the basic method. (See the books [1,4–6,18,19], the survey [20], which is good but reflects the situation of middle 1980's; the most significant, in our opinion, recent works [13,16,17,21,26,27], as well as the excellent book by Neumaier [22]). The only requirements for a basic method will be low computational complexity (for example, the complexity of its realization would be at most polynomial with respect to the dimension of the problem), and that the condition

$$\begin{aligned}&\text{the estimate } E_k(\mathbf{U},\mathbf{v}) \text{ is monotone with respect}\\ &\text{to inclusion in the arguments } \mathbf{U} \text{ and } \mathbf{v}, \text{ that is,}\\ &\text{for all } \mathbf{U}', \mathbf{U}'' \in I\mathbb{R}^{n\times n} \ , \ \mathbf{v}', \mathbf{v}'' \in I\mathbb{R}^n \text{ for}\\ &\mathbf{U}' \subseteq \mathbf{U}'' \text{ and } \mathbf{v}' \subseteq \mathbf{v}'', \text{ the inequality}\\ &E_k(\mathbf{U}',\mathbf{v}') \geqslant E_k(\mathbf{U}'',\mathbf{v}'') \text{ is valid.}\end{aligned} \qquad \text{(C1)}$$

holds. For the most of the popular algorithms, solving the "outer problem" for ILAS, including the interval Gauss method [1,4,5], the interval sweep methods [5], Gay's methods [13] and various modifications of the simple iteration method [1,4,5,16,17,27 and other], the feasibility of Condition (C1) is easily deduced from the monotonicity property of interval arithmetic with respect to inclusion [1,4,5,18,19].

If an entry $\mathbf{a}_{ij}$ of the matrix $\mathbf{A}$ has a nonzero width, we denote by $\mathbf{A}'$ and $\mathbf{A}''$ the matrices obtained from $\mathbf{A}$ by replacing $\mathbf{a}_{ij}$ by $\underline{\mathbf{a}}_{ij}$ and $\overline{\mathbf{a}}_{ij}$, that is, the left and right endpoints of the interval $\mathbf{a}_{ij}$ respectively. By the Beeck-Nickel result,

$$\min\{x_k \mid x \in X^*(\mathbf{A},\mathbf{b})\} \ = \ (A_0^{-1}b_0)_k$$

for some $A_0 \in \mathbb{R}^{n\times n}$ and $b_0 \in \mathbb{R}^n$, composed of endpoints of components of $\mathbf{A}$ and $\mathbf{b}$. Furthermore,

$$E_k(A_0, b_0) \leqslant (A_0^{-1}b_0)_k,$$

and Condition (C1) implies the inequalities

$$E_k(\mathbf{A}, \mathbf{b}) \leqslant E_k(\mathbf{A}', \mathbf{b}) \leqslant E_k(A_0', b_0)$$

and

$$E_k(\mathbf{A}, \mathbf{b}) \leqslant E_k(\mathbf{A}'', \mathbf{b}) \leqslant E_k(A_0'', b_0),$$

where A_0' and A_0'' are real matrices obtained from A_0 by replacing its (i,j)-th entry by $\underline{\mathbf{a}}_{ij}$ and $\overline{\mathbf{a}}_{ij}$ respectively. Therefore,

$$E_k(\mathbf{A}, \mathbf{b}) \leqslant \min\{E_k(\mathbf{A}', \mathbf{b}),\ E_k(\mathbf{A}'', \mathbf{b})\} \leqslant \min\{x_k \mid x \in X^*(\mathbf{A}, \mathbf{b})\}.$$

Thus, solving two "descendant" ILAS $\mathbf{A}'x = \mathbf{b}$ and $\mathbf{A}''x = \mathbf{b}$, we obtain, in the general case, a more exact lower estimate for $\min\{x_k \mid x \in X^*(\mathbf{A}, \mathbf{b})\}$ of the form $\min\{E_k(\mathbf{A}', \mathbf{b}),\ E_k(\mathbf{A}'', \mathbf{b})\}$.

The partitioning of any interval component $\mathbf{b}_i$ into $\underline{\mathbf{b}}_i$ and $\overline{\mathbf{b}}_i$ in the vector $\mathbf{b}$ of the right-hand side has a similar result. Therefore, for uniformity, let us agree to denote henceforth by $\mathbf{A}'x = \mathbf{b}'$ and $\mathbf{A}''x = \mathbf{b}''$ the descendants of ILAS, obtained from $\mathbf{A}x = \mathbf{b}$ by partitioning some component of the matrix $\mathbf{A}$ or of the vector $\mathbf{b}$ into endpoints.

The procedure of improving the estimate for $\min\{x_k \mid x \in X^*(\mathbf{A}, \mathbf{b})\}$ by partitioning the original ILAS can be repeated with respect to the descendant systems $\mathbf{A}'x = \mathbf{b}'$ and $\mathbf{A}''x = \mathbf{b}''$, to improve the estimate some more, and so on. We shall arrange this procedure of successive improvement of the lower estimate for $\min\{x_k \mid x \in X^*(\mathbf{A}, \mathbf{b})\}$ according to the strategy of the well known in the combinatorial optimization "branch and bound algorithm," as in solving optimization problems in [8,11,15,24] and other works. First, we form the list L of all the systems $\mathbf{U}x = \mathbf{v}$, together with their estimates $E_k(\mathbf{U}, \mathbf{v})$ arising during the procedure of partitioning the original ILAS. Second, we shall partition in each step that descendant ILAS that gives the best current estimate for $\min\{x_k \mid x \in X^*(\mathbf{A}, \mathbf{b})\}$.

Thus, during the work, the algorithm maintains a list L composed of records in the form of triplets $(\mathbf{U}, \mathbf{v}, E_k(\mathbf{U}, \mathbf{v}))$, where $\mathbf{U}$ is an interval

$n \times n-$ matrix, $\mathbf{U} \subseteq \mathbf{A}$, $\mathbf{v}$ is an interval n-vector, $\mathbf{v} \subseteq \mathbf{b}$. In addition, to work more effectively with the list L, we shall put its records in order of ascending values of the estimate $E_k(\mathbf{U}, \mathbf{v})$ and shall refer to the first record $(\mathbf{Q}, \mathbf{r}, E_k(\mathbf{Q}, \mathbf{r}))$ of the list, as well as the ILAS $\mathbf{Q}x = \mathbf{r}$ and the estimate $E_k(\mathbf{Q}, \mathbf{r})$ (the least in L) *leading* at the given step. Before starting the Algorithm, there is only one record in L, namely, $(\mathbf{A}, \mathbf{b}, E_k(\mathbf{A}, \mathbf{b}))$. A sequence of steps (iterations of the Algorithm) is then launched. Each such sequence consists of the following instructions:

1. If the leading ILAS $\mathbf{Q}x = \mathbf{r}$ is a point, stop the computation.
2. Choose an interval component $\mathbf{s}$ of the matrix $\mathbf{Q} = (\mathbf{q}_{ij})$ and the vector $\mathbf{r} = (\mathbf{r}_i)$ that has the greatest width, that is, $\mathbf{s} = \mathbf{q}_{\mu\nu}$ or $\mathbf{s} = \mathbf{r}_\nu$ and
   width of $s = \max\{\max_{i,j}\{\text{width of } \mathbf{q}_{i,j}\}, \max_i\{\text{width of } \mathbf{r}_i\}\}$.
3. Generate interval descendant systems $\mathbf{Q}'x = \mathbf{r}'$ and $\mathbf{Q}''x = \mathbf{r}''$:
   if $\mathbf{s} = \mathbf{q}_{\mu\nu}$ for some $\mu, \nu \in \{1, 2, \ldots, n\}$ we let $\mathbf{q}'_{ij} = \mathbf{q}''_{ij} = \mathbf{q}_{ij}$ for $(i,j) \neq (\mu,\nu)$, $\mathbf{q}'_{\mu\nu} = \underline{\mathbf{q}}_{\mu\nu}$, $\mathbf{q}''_{\mu\nu} = \overline{\mathbf{q}}_{\mu\nu}$, $\mathbf{r}' = \mathbf{r}'' = \mathbf{r}$;
   if $\mathbf{s} = \mathbf{r}_\nu$ for some $\nu \in \{1, 2, \ldots, n\}$ we let $\mathbf{Q}' = \mathbf{Q}'' = \mathbf{Q}$, $\mathbf{r}' = \mathbf{r}'' = \mathbf{r}$ for $i \neq \nu$, $\mathbf{r}'_\nu = \underline{\mathbf{r}}_\nu$, $\mathbf{r}''_\nu = \overline{\mathbf{r}}_\nu$.
4. Compute the estimates $E_k(\mathbf{Q}', \mathbf{r}')$ and $E_k(\mathbf{Q}'', \mathbf{r}'')$.
5. Remove from L the ex–leading record $(\mathbf{Q}, \mathbf{r}, E_k(\mathbf{Q}, \mathbf{r}))$.
6. Insert the records $(\mathbf{Q}', \mathbf{r}', E_k(\mathbf{Q}', \mathbf{r}'))$ and $(\mathbf{Q}'', \mathbf{r}'', E_k(\mathbf{Q}'', \mathbf{r}''))$ in the list L in such a way that would preserve its ranking in ascending order of the third field.

This and similar algorithms for solving the "outer problem" for interval systems of algebraic equations based on adaptive partitioning of the parameter set will be called, in abbreviation, *PPS–algorithms*, as opposed to the PSS–algorithms from [28], which also use adaptive partitioning but in the set of solutions of the interval system (i.e. *adaptive Partitioning of the Solution Set*).

If T is the total quantity of interval components (with nonzero width) of the matrix $\mathbf{A}$ and the vector $\mathbf{b}$ of the original ILAS (in the general case, $T \leqslant (n+1)n$), then the above PPS–algorithm stops after no more than 2^T steps and results in a lower estimate G for $\min\{x_k \mid x \in X^*(\mathbf{A}, \mathbf{b})\}$. The closeness of G and $\min\{x_k \mid x \in X^*\}$ depends primarily on the way the estimate $E_k(\mathbf{U}, \mathbf{v})$ is obtained, that is, on the basic method chosen for solving the intermediate ILAS's. In particular, for the estimate G to be optimal in the general case (in other words, for the equality $G =$

$\min\{x_k \mid x \in X^*\}$ to be exactly true) it is necessary and sufficient that the following condition hold:

$$\begin{aligned} &\text{the estimate } E_k(\mathbf{U}, \mathbf{v}) \text{ is exact on real} \\ &\text{linear algebraic systems, that is, } E_k(U, v) \\ &= (U^{-1}v)_k \quad \text{for all} \quad U \in \mathbb{R}^{n\times n}, \\ &v \in \mathbb{R}^n, \quad \det \quad A \neq 0. \end{aligned} \tag{C2}$$

Many solution methods for the "outer problem" for ILAS do not satisfy this condition. In particular, (C2) is not valid for the well known Rump method [26].

However, if the problem is sufficiently large, the PPS–algorithm will, as a rule, never complete, and will be, in practice an iterative method. It generates a sequence of improving lower estimates for $\min\{x_k \mid x \in X^*(\mathbf{A}, \mathbf{b})\}$, converging in the long run to the exact value of this quantity, and the complexity of each iteration can be made low by an appropriate choice of the basic method. Therefore, the PPS–algorithm splits into a series of effectively computable stages, and each of them results in a solution of the given "outer problem" for ILAS. This makes it fundamentally different from the simplest exhausting algorithm. We shall express this fact by saying that the PPS–algorithm is *sequentially guaranteeing.*

Relaxation algorithms that provide an improvement of an estimate of objective function at each step are popular in practical optimization. We shall carry over this definition directly to the algorithms that solve ILAS. Use of these algorithms is particularly attractive when it is known a priori that the number of steps cannot be large, but some meaningful result from the work of the algorithm is necessary. It is easy to see that, for the PPS–algorithm to be a relaxation algorithm, Condition (C1) is sufficient.

It is advisable to consider the prototypical PPS–algorithm described above as a base that can be developed and extended by standard improvements (see [8,11,15,24] and other), which leads to more nearly perfect PPS–methods for solving ILAS. In this instance, we can successfully realize at most the following modifications:

- one constructs an interval extension of the objective function that is better that the original one (that is, in this case the estimate $E_k(\mathbf{U}, \mathbf{v})$);

- the monotonicity test: after detecting the monotonicity of the objective function on the rectangles of the list L in some variables (usually, by estimating derivatives), the dimension of these rectangles diminishes;
- on the basis of specific local properties of the objective function, one uses minimization procedures in appropriate rectangles which are more efficient that the bisection;
- the partitioning strategy for the leading rectangles is modified (taking account of the properties of the objective function);
- simultaneously with calculation of estimating parameters over rectangles, values of the objective function are calculated in the centers of these rectangles; they give the lower bound of the desired global minimum; knowledge of this lower bound makes it possible to clean the list L of records which cannot be leading.

Let us consider in detail the second, and seemingly the most important, improvement in this list.

Let the ILAS $\mathbf{U}x = \mathbf{v}$ be given, and suppose we know $\partial x_k(\mathbf{U},\mathbf{v})/\partial u_{ij}$ and $\partial x_k(\mathbf{U},\mathbf{v})/\partial v_i$, the interval extension of the corresponding derivatives. If the interval $n \times n$ – matrix $\widetilde{\mathbf{U}}$ and n – vector $\widetilde{\mathbf{v}}$ are formed from the components (here "int" is the symbol for the topological interior)

$$\widetilde{\mathbf{u}}_{ij} = \begin{cases} [\underline{\mathbf{u}}_{ij}, \underline{\mathbf{u}}_{ij}], & \text{if } \partial x_k(\mathbf{U},\mathbf{v})/\partial u_{ij} \geqslant 0, \\ [\overline{\mathbf{u}}_{ij}, \overline{\mathbf{u}}_{ij}], & \text{if } \partial x_k(\mathbf{U},\mathbf{v})/\partial u_{ij} \leqslant 0\,, \\ \mathbf{u}_{ij}, & \text{if } \operatorname{int} \partial x_k(\mathbf{U},\mathbf{v})/\partial u_{ij} \ni 0\,, \end{cases} \qquad (*)$$

$$\widetilde{\mathbf{v}}_i = \begin{cases} [\underline{\mathbf{v}}_i, \underline{\mathbf{v}}_i], & \text{if } \partial x_k(\mathbf{U},\mathbf{v})/\partial v_i \geqslant 0\,, \\ [\overline{\mathbf{v}}_i, \overline{\mathbf{v}}_i], & \text{if } \partial x_k(\mathbf{U},\mathbf{v})/\partial v_i \leqslant 0\,, \\ \mathbf{v}_i, & \text{if } \operatorname{int} \partial x_k(\mathbf{U},\mathbf{v})/\partial v_i \ni 0\,, \end{cases} \qquad (**)$$

then, clearly, $\min\{x_k \mid x \in X^*(\mathbf{U},\mathbf{v})\} = \min\left\{x_k \mid x \in X^*(\widetilde{\mathbf{U}},\widetilde{\mathbf{v}})\right\}$. But since the number of interval components (with nonzero width) of $\widetilde{\mathbf{U}}$ and $\widetilde{\mathbf{v}}$ is in general substantially less than that in $\mathbf{U}$ and $\mathbf{v}$, we will simplify the problem of calculating $\min\{x_k \mid x \in X^*(\mathbf{U},\mathbf{v})\}$ when passing from the original ILAS $\mathbf{U}x = \mathbf{v}$ to the solution of the system $\widetilde{\mathbf{U}}x = \widetilde{\mathbf{v}}$.

How do we find the interval extensions of the derivatives appearing in $(*) - (**)$? This is done in the usual way as follows. If $Z = (z_{ij})$ is an

inverse matrix for $U = (u_{ij})$, the derivatives of the solution to the real system $Ux = v$ with respect to its coefficients are given, as is known, by the formulae:

$$\frac{\partial x_k(U,v)}{\partial u_{ij}} = -z_{ki}x_j, \qquad \frac{\partial x_k(U,v)}{\partial v_i} = z_{ki}$$

(see [1,14]). Therefore, in the case where $\mathbf{Z} = (\mathbf{z}_{ij})$ is an "inverse interval matrix" for $\mathbf{U}$ (that is, $\mathbf{Z} \supseteq \{U^{-1} \mid U \in \mathbf{U}\}$), and $\mathbf{x}_j$ is the j-th component of the interval vector $\mathbf{x} \supseteq X^*(\mathbf{U}, \mathbf{v})$, we can set

$$\frac{\partial x_k(\mathbf{U},\mathbf{v})}{\partial u_{ij}} = -\mathbf{z}_{ki}\mathbf{x}_j, \qquad \frac{\partial x_k(\mathbf{U},\mathbf{v})}{\partial v_i} = \mathbf{z}_{ki}\ .$$

Usually, to compute $\mathbf{Z}$ and $\mathbf{x}$, one uses some cheap approximate algorithm (like Hansen's method [18] for localizing the "inverse interval matrix"). However, the procedure of interval estimating the derivatives from $(*) - (**)$, nonetheless remains expensive on the whole†. Therefore, for reasons of efficiency, it is not wise to do this estimating at each step of the algorithm. We shall "freeze" the inverse interval matrix over a fixed number of steps (as is done with the Jacobian in a number of quasi–Newton methods for solving algebraic systems [3]). In doing so, the access to inverse interval matrices and their treatment becomes more complex, since the PPS–algorithm is essentially a branching process.

It is advisable to store inverse interval matrices in the form of a *heap* H , and to access them by pointers (or references) (see [2]). For this, we add one more field p to the records in the form of triplets $(\mathbf{U}, \mathbf{v}, E_k(\mathbf{U}, \mathbf{v}))$ constituting the list L. This field is the pointer to the interval matrix $\mathbf{Z}$ from H which envelops $\mathbf{U}^{-1}$. In this case we shall say that the record $(\mathbf{U}, \mathbf{v}, E_k(\mathbf{U}, \mathbf{v}))$ *is served* by the matrix $\mathbf{Z}$. Two more natural parameters α and β will be associated with each inverse interval matrix $\mathbf{Z}$ so that the heap H actually turns out to consist of records $(\mathbf{Z}, \alpha, \beta)$. The parameter α is a counter of algorithm's steps on which an access to $\mathbf{Z}$ took place; and β counts the number of records in the list L which are served by $\mathbf{Z}$. If α is large, this demonstrates that previously the algorithm made

†Some authors (for example, [15]) recommend using second derivatives (the Hessian) of the objective function in similar global optimization method. However, for our problem such an improvement seems to be unreasonable, since it will only lead to excessive complexity of the algorithm.

a long use of the matrix $\mathbf{Z}$, that is, it is necessary to replace it by new, narrower interval matrix. On the other hand, if $\beta = 0$, the corresponding inverse matrix does not serve any record of the list L, and hence the triplet $(\mathbf{Z}, \alpha, \beta)$ can be removed from H without any influence on the work of the algorithm.

The parameter α varies in the evident manner. The parameter β is set to 2 when the record $(\mathbf{Z}, \alpha, \beta)$ is inserted in H, and later, as is easy to understand, it is recomputed as follows:

- if the record served by $\mathbf{Z}$ is partitioned into descendants without recomputing the inverse matrix, then we augment β by one;
- if the record served by $\mathbf{Z}$ is partitioned into descendants and the inverse matrix is recomputed, then we decrease β by one;
- if the record served by $\mathbf{Z}$ is deleted from L, then we decrease β by one.

As computational experiments show, introducing the above mechanism, that takes account of monotonicity as well as the other modifications into the simplest PPS–algorithm, and careful choice of the basic method lead to efficient computational procedures for finding optimal and near optimal solutions of the "outer problem" for ILAS. Their application is especially attractive for systems that have only a few interval (nonzero width) elements.

To conclude, it is useful to outline the relation the algorithms proposed here have with other methods for solving ILAS. We have already observed a close connection (actually, a duality) between PPS–algorithms and the class of PSS–algorithms introduced in [28]. But we can arrive at the idea of constructing PPS–algorithms for solving interval linear systems in a somewhat different way, bringing to a logical conclusion some of the widespread interval analysis approaches to the solution of ILAS.

The procedure of Kupermann and Hansen [1, 14] may be considered as one of the predecessors of the modified PPS–algorithm. In fact, the PPS–algorithm is obtained by adding to this procedure the technique of partitioning and branching similar to that which is used in the "branch and bound" algorithm.

Next, starting from the result by Beeck-Nickel mentioned above, a semi-heuristic algorithm was proposed by Manusov, Moiseev and Perkov [7], for solving ILAS. This algorithm was based on the detecting, by trial

computations, the dependency between a solution of the system and its coefficients in the bounds of corresponding intervals. In the sequel, similar methods, based upon the results of Kupermann and Hansen [1,14], were developed by Senashov and Yuldashev [9,10]. (They also proposed the name "interval trial method".) But in spite of their high attractiveness, the algorithms of this type suffer from a serious shortcoming: in general, they don't assure guaranteed results.

In our case, this means that the resulting estimate G given by the algorithm does not necessarily satisfy the inequality

$$G \leqslant \min \{x_k \mid x \in X^*(\mathbf{A}, \mathbf{b})\}.$$

This weakening of the properties of the result contradicts the very spirit of interval analysis, and may be intolerable for some practical problems.

In our opinion, the Manusov-Moiseev-Perkov-Senashov-Yuldashev approach may receive further development only through recognition of the fact that it is impossible to make the dependency of a solution from coefficients completely and finally determinate, if for solving ILAS, methods that do not give an optimal solution are applied. The search for an optimal solution requires too much computational complexity, and does not simplify the original problem.

Nevertheless, recognition of the fact that complete determinism is impossible does not mean, in our opinion, that the basic idea of the approach developed in [7,9,10] is useless. One of the possible ways to save the situation is to refuse to reject variants "condemned" by incomplete determining, but to store them during the computations, together with promising ones, since, under more detailed determining, previously condemned variants would turn out to become promising.

Actually, the PPS–algorithm does this. The list L stores all variants occurring during the work (except the clearly hopeless ones), and at each step of the algorithm the "most promising" variant is processed, if this "promise" is measured by the values of the estimate $E_k(\mathbf{U}, \mathbf{v})$.

References

1. Alefeld G., and Herzberger J., *Introduction to interval computations*, Academic Press, New York, 1983.

2. Bauer F. and Goos G., *Informatik*, Springer Verlag, 1973.

3. Dennis J., Jr. and Schnabel R., *Numerical methods for unconstrained optimization and nonlinear equations*, Prentice-Hall, Englewood Cliffs, 1983.

4. Dobronets B.S. and Shaidurov V.V., *Two-sided numerical methods*, Nauka, Novosibirsk, 1990. (in Russian)

5. Kalmykov S.A., Shokin Yu.I., and Yuldashev Z.Kh., *Methods of interval analysis*, Nauka, Novosibirsk, 1986. (in Russian)

6. Kurpel' N.S. and Shuvar B.A., *Two-sided operator inequalities and their applications*, Naukova dumka, Kiev, 1980. (in Russian)

7. Manusov V.Z., Moiseev S.M., and Perkov S.D., *Interval analysis in linear electrotechnic problems*, Information-operative material (interval analysis), preprint 6, Computer Center, Siberian Branch of the USSR Academy of Sciences, Krasnoyarsk, 29–31 (1988). (in Russian)

8. Pankov P.S., *Algorithm of proving stable assertions and global optimization over a bounded domain*, Frunze, 1984, Deposited in VINITI, No.5250-84. (in Russian)

9. Senashov V.I. and Yuldashev Z.Kh., *Interval trial method*, Information-operative material (interval analysis), preprint 9, Computer Center, Siberian Branch of the USSR Academy of Sciences, Krasnoyarsk, 31–36 (1989). (in Russian)

10. Senashov V.I. and Yuldashev Z.Kh., *On solving system of equations of nodal tensions*, in Information-operative material (interval analysis) Part 1, preprint 16, Computer Center, Siberian Branch of the USSR Academy of Sciences, Krasnoyarsk, 25–28 (1990). (In Russian)

11. Asaithambi N.S., Shen Zuhe and Moore R.E., *On computing the range of values*, Computing **28** 3, 225–237 (1982).

12. Beeck H., *Über die Struktur und Abschätzungen der Lösungsmenge von Linearen Gleichungssystemen mit Intervallkoeffizienten*, Computing **10**, 231–244 (1972).

13. Gay D.M., *Solving interval linear equations*, SIAM Journal on Numerical Analysis **19**, 4, 858–870 (1982).

14. Hansen E.R., *On linear algebraic equations with interval coefficients*, Topics in Interval Analysis, ed. E. Hansen (1969), 35–46, Clarendon Press, Oxford.

15. Hansen E.R., *Global optimization using interval analysis – the multi-dimensional case*, Numerische Mathematik **34** 3, 247 – 270 (1980).

16. Kearfott R.B., *Preconditioners for the interval Gauss-Seidel Method*, SIAM Journal on Numerical Analysis **27**, 3, 804–822 (1990).

17. Mayer G., *Enclosing the solutions of systems of linear equations by interval iterative processes*, Computing Supplementum 6, 47–58, Springer Verlag, (1988).

18. Moore R.E., *Interval analysis*, Prentice-Hall, Englewood Cliffs, 1966.

19. Moore R.E., *Methods and applications of interval analysis*, SIAM, Philadelphia (1979).

20. Neumaier A., *Linear interval equations*, Interval Mathematics 1985, ed. K. Nickel; Lecture Notes in Computer Science, **212**, Springer Verlag, 109–120 (1986).

21. Neumaier A., *Overestimation in linear interval equations*, SIAM Journal on Numerical Analysis **24** (1987), 207–214.

22. Neumaier A., *Interval methods for systems of equations*, Cambridge University Press, 1990.

23. Nickel K., *Die Überschätzung des Wertebereiches einer Funktion in der Intervall-rechnung mit Anwendungen auf Lineare Gleichungssysteme*, Computing **18**, 15–36 (1977).

24. Ratschek H., *Inclusion functions and global optimization*, Mathematical Programming **33**, 300–317 (1985).

25. Rohn J., *Systems of linear interval equations*, Linear Algebra and its Applications **126**, 39–78 (1989).

26. Rump S.M., *Solving algebraic problems with high accuracy*, A New Approach to Scientific Computations, eds. U.Kulisch, W.L.Miranker, Academic Press, 27–49 (1983).

27. Schwandt H., *Iterative methods for systems of equations with interval coefficients and linear form*, Computing **38**, 143–161 (1987).

28. Shary S.P., *Optimal solution of interval linear algebraic systems 1*, Interval Computations **2**, 7–30 (1992).

Computer Center of the Siberian
Branch of the Academy of Sciences,
Akademgorodok,
660036 Krasnoyarsk,
Russia