An Algorithm for Checking Regularity of Interval Matrices

ARTICLE *in* SIAM JOURNAL ON MATRIX ANALYSIS AND APPLICATIONS · JANUARY 1999

Impact Factor: 1.59 · DOI: 10.1137/S0895479896313978

2 AUTHORS:

Christian Jansson
Technische Universität Hamburg-Harburg
25 PUBLICATIONS **242** CITATIONS
SEE PROFILE

Jiri Rohn
Academy of Sciences of the Czech Republic
128 PUBLICATIONS **1,739** CITATIONS
SEE PROFILE

AN ALGORITHM FOR CHECKING REGULARITY OF INTERVAL MATRICES

CHRISTIAN JANSSON* AND JIRI ROHN†

Abstract. Checking regularity (or singularity) of interval matrices is a known NP-hard problem. In this paper a general algorithm for checking regularity/singularity is presented which is not a priori exponential. The algorithm is based on a theoretical result according to which regularity may be judged from any single component of the solution set of an associated system of linear interval equations. Numerical experiments (with interval matrices up to the size $n = 50$) confirm that this approach brings an essential decrease in the amount of operations needed.

Key words. Interval matrix, regularity, singularity, algorithm, branch and bound, linear interval equations, linear programming

AMS subject classifications. 65F30, 65G10, 90C90

1. Introduction.

An interval matrix

$$A^I = [\underline{A}, \overline{A}] = \{A;\ \underline{A} \leq A \leq \overline{A}\}$$

(where $\underline{A}$, $\overline{A}$ are $n \times n$ matrices and the inequality is to be understood componentwise) is called regular if each $A \in A^I$ is nonsingular, and is said to be singular otherwise (i.e., if it contains a singular matrix). The problem of checking regularity naturally arises in solving linear interval equations, but it is also important in applications since some frequently used properties of interval matrices (as positive definiteness, stability or the P-matrix property) can be verified via checking regularity.

All presently known necessary and sufficient conditions for regularity of interval matrices (summed up in [17], Theorem 5.1) exhibit exponential behaviour: they require solving at least 2^n problems of some sort (as computing determinants, solving linear equations, inverting matrices, checking the P-matrix property etc.), hence they are of little use in practice except for examples of very low dimensions. The exponentiality inherent in all these conditions was explained by the result stating that checking regularity of interval matrices is an NP-hard problem (Poljak and Rohn [15], [16], see also Nemirovskii [13]). Hence, in view of the present status of the complexity theory (the conjecture "P≠NP", Garey and Johnson [4]), there is only little hope that necessary and sufficient conditions verifiable in polynomial time may exist.

The present work was motivated by an attempt to construct an algorithm that would require an exponential number of operations only in the "worst case" examples, and would behave reasonably in the "average" ones. The algorithm presented in this paper, whose main idea is due to Jansson [7], is based on a necessary and sufficient condition of a quite different sort. First, it does not handle solely the interval matrix A^I, but it deals with the solution set $X(A^I, b) = \{x;\ Ax = b \text{ for some } A \in A^I\}$ of an interval linear system $A^I x = b$ with a specially preselected right-hand side b. Second, the necessary and sufficient condition says that for any component C of $X(A^I, b)$ (i.e., a maximal nonempty connected subset of $X(A^I, b)$), A^I is regular if and only if C is bounded. Hence, in order to check regularity, it is sufficient to take (better

* Institute of Computer Science III, Technical University Hamburg-Harburg, Eißendorfer Str. 38, D-21071 Hamburg, Germany (jansson@tu-harburg.de).

† Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic, and Institute of Computer Science, Academy of Sciences, Prague, Czech Republic (rohn@uivt.cas.cz).

said, to generate) a *single* component of $X(A^I, b)$ and to check it for boundedness, which can be done by applying repeatedly a linear programming procedure. The fact that only one component is to be checked, together with a special choice of b aimed at minimizing the number of applications of the linear programming procedure, is decisive for achieving a big reduction in the amount of computations needed. As it will be demonstrated later, we have been able to check in acceptable time regularity of interval matrices up to the size $n = 50$, which would be almost impossible via the classical necessary and sufficient conditions since $2^{50} \approx 10^{15}$.

The paper is organized as follows. To motivate the current research, in section 2 we first show how some properties of interval matrices (positive definiteness, stability and the P-property) can be verified via checking regularity. The NP-hardness result is stated in section 3. In section 4 we derive a necessary and sufficient singularity condition of classical type; it not only demonstrates the inherent exponentiality of the problem, but also shows a way how to try to resolve it. The necessary and sufficient condition employed in the algorithm, formulated in terms of a component of the solution set as explained above, is proved in section 5. The algorithm is described in section 6, and the problem of the choice of an adequate right-hand side b is discussed in section 7. Some examples are given in the last section.

We shall use the following notations. The absolute value of a matrix $A = (a_{ij})$ is denoted by $|A| = (|a_{ij}|)$; the same notation applies to vectors as well. The set of all ± 1-vectors in R^n is denoted by Z, hence Z consists of 2^n vectors. For each $z \in R^n$ we denote

$$T_z := \begin{pmatrix} z_1 & 0 & \dots & 0 \\ 0 & z_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & z_n \end{pmatrix},$$

i.e., T_z is the diagonal matrix with diagonal vector z. For $x \in R^n$ we denote by $\operatorname{sgn} x$ the sign vector of x defined by

$$(\operatorname{sgn} x)_i = \begin{cases} 1 & \text{if } x_i \geq 0, \\ -1 & \text{if } x_i < 0 \end{cases}$$

$(i = 1, \dots, n)$. Hence, $\operatorname{sgn} x \in Z$ for each $x \in R^n$. Moreover, if $z = \operatorname{sgn} x$, then we have $|x| = T_z x$; we shall use this relation later to avoid use of absolute values.

2. Regularity and other properties. To motivate the current research, we shall state briefly here some results showing that a general method for checking regularity may be also employed for checking some other properties of interval matrices.

A real matrix A (not necessarily symmetric) is called positive definite if $x^T A x > 0$ for each $x \neq 0$, stable if $\operatorname{Re} \lambda < 0$ for each eigenvalue λ of A, and P-matrix if all the principal minors of A are positive. An interval matrix A^I is said to have one of these properties if each $A \in A^I$ has this property.

Checking positive definiteness of interval matrices is needed e.g. in global optimization where branch-and-bound methods for nonlinear optimization problems may be accelerated and may be also made finite by using a so-called expansion scheme (see Jansson [6]) which is mainly based on proving positive definiteness of an interval matrix. The following result from [19] reduces checking positive definiteness to checking regularity:

THEOREM 2.1. *An interval matrix* $[\underline{A}, \overline{A}]$ *is positive definite if and only if the interval matrix*

$$\left[\tfrac{1}{2}(\underline{A} + \underline{A}^T), \tfrac{1}{2}(\overline{A} + \overline{A}^T)\right]$$

is regular and contains at least one positive definite matrix.

The problem of checking stability of interval matrices has attracted interest of many researchers in the last decade since it is closely connected to the problem of robust stability of a linear time-invariant system $\dot{x} = Ax$ under data perturbations (see Barmish [1], Lunze [9] and the survey paper by Mansour [10]). Here, reduction to regularity holds for symmetric interval matrices only. An interval matrix $A^I = [\underline{A}, \overline{A}]$ is called symmetric if both the bounds $\underline{A}, \overline{A}$ are symmetric; hence a symmetric interval matrix may contain nonsymmetric matrices. The following result comes again from [19]:

THEOREM 2.2. *A symmetric interval matrix* A^I *is stable if and only if it is regular and contains at least one symmetric stable matrix.*

The study of P-matrices is motivated, among other applications, by the fact that the linear complementarity problem

$$x^+ = Ax^- + b$$

has a unique solution $x = (x_i)$ for each right-hand side b if and only if A is a P-matrix (see Murty [11]). Here the vectors x^+ and x^- are defined by $x_i^+ = \max\{x_i, 0\}$ and $x_i^- = \max\{-x_i, 0\}$ for each i, so that $x = x^+ - x^-$. We have an analogous result [20]:

THEOREM 2.3. *A symmetric interval matrix* A^I *is a* P*-matrix if and only if it is regular and contains at least one symmetric* P*-matrix.*

These results give further motivation for studying the problem of checking regularity of interval matrices.

3. The NP-hardness result. At the end of the 1980's, existence of more than 10 necessary and sufficient conditions for checking regularity of interval matrices, each of which required an exponential amount of time to check (summed up in [17], Theorem 5.1), led to suspicion that the problem might be NP-hard. The corresponding statement was proved by Poljak and Rohn in a report [15] and later in a journal form [16]; it was independently proved by Nemirovskii [13]. In its most recent form [8], the result is stated as follows:

THEOREM 3.1. *Checking regularity is NP-hard in the class of interval matrices of the form*

$$[A - E, A + E],$$

where A *is a nonnegative symmetric positive definite rational matrix and* E *is the matrix of all ones.*

This result shows that checking regularity of interval matrices is a problem at least as hard as the most difficult combinatorial problems in the class NP. Existence of a polynomial-time algorithm for checking regularity of interval matrices would imply, in view of the definition of NP-hardness (see Garey and Johnson [4]), polynomial-time solvability of all problems in the class NP – a possibility which at the present state of human knowledge cannot be fully excluded, but in view of the experience with solving problems from the class NP gathered so far ought to be considered very unlikely. Nevertheless, polynomial-time algorithms may exist for special classes of interval matrices, see [2].

This shows that the problem we are interested in is in its full generality very difficult.

4. A necessary and sufficient condition. Some of the subsequent results (Theorems 4.2 and 5.1) can be derived from the Oettli-Prager theorem [14] for interval linear equations. In order to keep the paper self-contained, we prove them here in another way using the following result (Theorem 4.1) which is of independent interest. For the purpose of its formulation, for an interval matrix $A^I = [\underline{A}, \overline{A}]$ we introduce

$$A_c = \tfrac{1}{2}(\underline{A} + \overline{A})$$

(the center matrix) and

$$\Delta = \tfrac{1}{2}(\overline{A} - \underline{A})$$

(the radius matrix), then A^I can be written in the form

$$A^I = [A_c - \Delta, A_c + \Delta]$$

which we shall use from this point on. We have this result (for its formulation and proof, we refer to the notations introduced at the end of section 1):

THEOREM 4.1. *Let $A^I = [A_c - \Delta, A_c + \Delta]$ be an $n \times n$ interval matrix and let $x \in R^n$. Then*

$$\{Ax;\ A \in A^I\} = [A_c x - \Delta|x|, A_c x + \Delta|x|].$$

Proof. If $A \in A^I$, then $|Ax - A_c x| = |(A - A_c)x| \leq \Delta|x|$, which implies

$$Ax \in [A_c x - \Delta|x|, A_c x + \Delta|x|].$$

Conversely, let $b \in [A_c x - \Delta|x|, A_c x + \Delta|x|]$, so that $|A_c x - b| \leq \Delta|x|$. Define

$$y_i = \begin{cases} (A_c x - b)_i / (\Delta|x|)_i & \text{if } (\Delta|x|)_i \neq 0, \\ 1 & \text{if } (\Delta|x|)_i = 0 \end{cases}$$

$(i = 1, \ldots, n)$, then $|y_i| \leq 1$ and $(A_c x - b)_i = y_i(\Delta|x|)_i$ holds for each i, hence with $z = \operatorname{sgn} x$ we have $A_c x - b = T_y \Delta T_z x$ and thus also $(A_c - T_y \Delta T_z)x = b$, where $A_c - T_y \Delta T_z$ belongs to A^I due to $|T_y \Delta T_z| \leq \Delta$, hence $b \in \{Ax;\ A \in A^I\}$. □

Returning back to our problem, we obtain this characterization of singularity:

THEOREM 4.2. *An interval matrix $A^I = [A_c - \Delta, A_c + \Delta]$ is singular if and only if the inequality*

$$|A_c x| \leq \Delta|x| \tag{4.1}$$

has a nontrivial solution.

Proof. Obviously, A^I is singular if and only if

$$0 \in \{Ax;\ A \in A^I\}$$

holds for some $x \neq 0$, which in view of Theorem 4.1 is equivalent to

$$A_c x - \Delta|x| \leq 0 \leq A_c x + \Delta|x|, \tag{4.2}$$

and thus also to

$$|A_c x| \leq \Delta|x|.$$

□

As we have seen in the proof, the left-hand side absolute value in (4.1) can be removed (Eq. (4.2)), but the right-hand one remains a problem. We can remove it only at the expense of the latent exponentiality in (4.1) becoming apparent, as it can be seen from the next theorem.

THEOREM 4.3. *An interval matrix* $A^I = [A_c - \Delta, A_c + \Delta]$ *is singular if and only if the linear programming problem*

$$\max\{z^T x;\ (A_c - \Delta T_z)x \le 0,\ (A_c + \Delta T_z)x \ge 0,\ T_z x \ge 0\} \tag{4.3}$$

is unbounded for some $z \in Z$.

Proof. If A^I is singular, then by Theorem 4.2 there exists a vector $x \neq 0$ satisfying

$$-\Delta|x| \le A_c x \le \Delta|x|. \tag{4.4}$$

Setting $z = \operatorname{sgn} x$, we have $|x| = T_z x$, hence $T_z x \ge 0$ and from (4.4) it follows $(A_c - \Delta T_z)x \le 0$ and $(A_c + \Delta T_z)x \ge 0$, which shows that x is a feasible solution of (4.3). Moreover, $z^T x = \sum_i |x_i| > 0$. Since αx is a feasible solution of (4.3) for each $\alpha > 0$, we can see that the linear programming problem (4.3) is unbounded.

Conversely, let (4.3) be unbounded for some $z \in Z$. Then there exists an x satisfying $(A_c - \Delta T_z)x \le 0$, $(A_c + \Delta T_z)x \ge 0$, $T_z x \ge 0$ and $z^T x > 0$. Since $T_z x = |x|$, we obtain that (4.1) is satisfied for some $x \neq 0$, hence A^I is singular. □

Let us now discuss this result. If we find out a $z \in Z$ for which (4.3) is unbounded, then A^I is proved to be singular and we are done. However, if A^I is regular, then we must inspect *all* the 2^n linear programming problems of type (4.3) (for each $z \in Z$) in order to be able to prove it. The cornerstone of the exponentiality is the fact that each linear programming problem (4.3) is feasible ($x = 0$ is always a feasible solution), hence none of the problems (4.3) can be *a priori* excluded. In this way we come to the basic idea: to replace the zero vector in the right-hand sides of the constraints $(A_c - \Delta T_z)x \le 0$, $(A_c + \Delta T_z)x \ge 0$ by some nonzero vector b in order to make infeasible as many resulting linear programming problems as possible. We shall exploit and develop this idea in the next section.

5. Theoretical basis of the algorithm. Given an $n \times n$ interval matrix A^I and a vector $b \in R^n$ (we shall specify its choice later), consider the solution set of a system of interval linear equations defined by

$$X(A^I, b) = \{x;\ Ax = b \text{ for some } A \in A^I\}.$$

First we have this description:

THEOREM 5.1. *Let* $A^I = [A_c - \Delta, A_c + \Delta]$ *be an* $n \times n$ *interval matrix and let* $b \in R^n$. *Then*

$$X(A^I, b) = \{x;\ |A_c x - b| \le \Delta|x|\}. \tag{5.1}$$

Proof. $x \in X(A^I, b)$ if and only if

$$b \in \{Ax;\ A \in A^I\}$$

holds, which according to Theorem 4.1 is equivalent to

$$A_c x - \Delta|x| \le b \le A_c x + \Delta|x|,$$

and thereby also to

$$|A_c x - b| \le \Delta|x|.$$ □

Next we shall remove the absolute values from (5.1):

THEOREM 5.2. *Let $A^I = [A_c - \Delta, A_c + \Delta]$ be an $n \times n$ interval matrix and let $b \in R^n$. Then*

$$X(A^I, b) = \bigcup_{z \in Z} X_z(A^I, b),$$

where

$$X_z(A^I, b) = \{x;\ (A_c - \Delta T_z)x \le b,\ (A_c + \Delta T_z)x \ge b,\ T_z x \ge 0\} \tag{5.2}$$

for $z \in Z$.

Proof. If $x \in X(A^I, b)$, then $|A_c x - b| \le \Delta|x|$ and we can easily see that x satisfies

$$(A_c - \Delta T_z)x \le b,\ (A_c + \Delta T_z)x \ge b,\ T_z x \ge 0 \tag{5.3}$$

for $z = \operatorname{sgn} x$, hence $x \in X_z(A^I, b)$. Conversely, if $x \in X_z(A^I, b)$ for some $z \in Z$, then from (5.3) we have $|A_c x - b| \le \Delta|x|$, hence $x \in X(A^I, b)$. □

The set $X_z(A^I, b)$, defined by (5.2), is nothing else than the intersection of the solution set $X(A^I, b)$ with the orthant $\{x \in R^n;\ T_z x \ge 0\}$. Since $X_z(A^I, b)$ is described by a system of linear inequalities, it is a convex polytope; in particular, it is connected (let us recall that a set is called connected [sometimes, path-connected] if any two points of it can be connected by a curve which belongs entirely to the set; in case of a convex set the curve is simply the segment connecting the two points). However, $X(A^I, b)$, as the union of such sets, may be neither convex, nor connected. Consider a one-dimensional example

$$[-1, 1]x_1 = [1, 1].$$

Here $X(A^I, b)$ consists of two disjoint unbounded connected sets

$$X_{-1}(A^I, b) = (-\infty, -1]$$

and

$$X_1(A^I, b) = [1, \infty).$$

We shall see later (Theorem 5.3) that such a situation may occur for singular interval matrices only (indeed, A_c is singular here).

A nonempty connected subset C of $X(A^I, b)$ is called a *component* of $X(A^I, b)$ if it has the property

$$C \subseteq D \subseteq X(A^I, b),\ D \text{ connected } \Rightarrow C = D,$$

i.e., if it is a maximal connected subset of $X(A^I, b)$ with respect to inclusion. Hence each set $X_z(A^I, b), z \in Z$, being connected, must be entirely contained in a single component. This implies that each component C of $X(A^I, b)$ is of the form

$$C = \bigcup_{z \in Y} X_z(A^I, b) \tag{5.4}$$

for some $Y \subseteq Z$. The following theorem shows that regularity or singularity of A^I may be judged from *each* single component of $X(A^I, b)$. The result is due to Jansson [7] where it appeared in another formulation.

THEOREM 5.3. *Let A^I be an $n \times n$ interval matrix, let $b \in R^n$ and let C be an arbitrary component of $X(A^I, b)$. Then A^I is singular if and only if C is unbounded.*

Proof. Both implications are proved by contradiction. If A^I is regular, then $X(A^I, b)$, as the range of the continuous mapping $A \mapsto A^{-1}b$ on a compact set A^I, is connected and bounded, hence $C = X(A^I, b)$, which implies that C is bounded.

Conversely, let C be bounded. Since $C \neq \emptyset$ by definition, there exists an $x_0 \in C$ satisfying $A_0 x_0 = b$ for some $A_0 \in A^I$, and boundedness of C implies nonsingularity of A_0 (if A_0 were singular, then $A_0 \hat{x} = 0$ for some $\hat{x} \neq 0$ and C would contain an unbounded set $\{x_0 + \lambda \hat{x};\ \lambda \in R^1\}$, a contradiction). To prove that A^I is regular, assume to the contrary that A^I contains a singular matrix A_1. For each $t \in [0, 1]$ denote

$$A_t = A_0 + t(A_1 - A_0), \tag{5.5}$$

and let

$$\tau = \inf\{t \in [0, 1];\ A_t \text{ is singular}\}.$$

In view of continuity of the determinant, the infimum is achieved as minimum, hence A_τ is singular and $\tau \in (0, 1]$. For each $t \in [0, \tau)$, A_t is nonsingular, hence

$$x_t = A_t^{-1} b$$

is well defined and the mapping $s \mapsto x_s$, $s \in [0, t]$, defines a curve in $X(A^I, b)$ connecting x_0 with x_t, hence $x_t \in C$ for each $t \in [0, \tau)$. Consider now the sequence of points $\{x_{t_m}\}$ with

$$t_m = (1 - \tfrac{1}{m})\tau, \tag{5.6}$$

$m = 1, 2, \ldots$. Then $\{x_{t_m}\}$ is bounded since C is bounded, hence it contains a convergent subsequence $\{x_{t_{m_k}}\}$, $x_{t_{m_k}} \to x^*$. Then $x^* \in C$ since C is closed due to (5.4), (5.2). As

$$A_{t_{m_k}} x_{t_{m_k}} = b$$

holds for each k, taking $k \to \infty$ we obtain in view of (5.5), (5.6) that

$$A_\tau x^* = b.$$

But since A_τ is singular, there exists an $\tilde{x} \neq 0$ with $A_\tau \tilde{x} = 0$, hence $A_\tau(x^* + \lambda \tilde{x}) = b$ for each $\lambda \in R^1$, which shows that C contains the unbounded set $\{x^* + \lambda \tilde{x};\ \lambda \in R^1\}$, a contradiction. Hence A^I must be regular; this concludes the second part of the proof. □

The result shows that if A^I is singular, then for each $b \in R^n$, *all* components of $X(A^I, b)$ are unbounded; if A^I is regular, then $X(A^I, b)$ has a single component which is equal to $X(A^I, b)$.

It remains to show how to check unboundedness of a component C in the form (5.4):

THEOREM 5.4. *A component*

$$C = \bigcup_{z \in Y} X_z(A^I, b)$$

is unbounded if and only if the linear programming problem

$$\max\{z^T x;\ (A_c - \Delta T_z)x \le b,\ (A_c + \Delta T_z)x \ge b,\ T_z x \ge 0\} \tag{5.7}$$

is unbounded for some $z \in Y$.

Proof. Obviously, C is unbounded if and only if $X_z(A^I, b)$ is unbounded for some $z \in Y$. If $X_z(A^I, b)$ is unbounded, then the optimization problem

$$\max\{|x_i|;\ (A_c - \Delta T_z)x \le b,\ (A_c + \Delta T_z)x \ge b,\ T_z x \ge 0\} \tag{5.8}$$

is unbounded for some i, which implies that (5.7) is unbounded since

$$|x_i| \le \sum_j |x_j| = z^T x$$

holds for each feasible solution x of (5.7), (5.8). Conversely, if (5.7) is unbounded, then (5.8) must be unbounded for some i, which means that the set $X_z(A^I, b)$ is unbounded. □

By comparing Theorems 4.3 and 5.4, we can see that the only (but essential) distinction made by the introduction of the right-hand side b is the difference in quantifiers "$z \in Z$" and "$z \in Y$", where cardinality of Y is equal to the number of the orthants intersected by the component C. Clearly, this number may be influenced by an appropriate choice of b. We shall discuss this question in section 7, but first we shall formulate an algorithm based on theoretical principles of this section.

6. The algorithm. The algorithm is based on Theorems 5.3 and 5.4 and constructs a component C of the form (5.4) implicitly in the following way. At the starting point, a list L of z's to be checked is initialized by setting $z := \operatorname{sgn}(A_c^{-1}b)$. In the current step of the algorithm, if for the current $z \in L$ the problem (5.7) is unbounded, then A^I is proved singular and the algorithm terminates; otherwise, if (5.7) is feasible, then we insert into L all the "neighbouring" vectors from the set

$$N(z) = \{(z_1, \ldots, z_{j-1}, -z_j, z_{j+1}, \ldots, z_n)^T;\ 1 \le j \le n\}$$

that have not been checked yet; to be able to recognize it, we also keep a list K of vectors that have already passed the check. If L becomes empty at some stage, then the component of $X(A^I, b)$ containing the vector $A_c^{-1}b$ has been proved to be bounded, and A^I is regular by Theorem 5.3. Following we give a formal description of the algorithm written in a pseudo-PASCAL code:

sing:=false;
if A_c is singular **then** *sing:=true*
else
  $L := \emptyset$; $K := \emptyset$;
  select b; solve $A_c x = b$;
  $z := \operatorname{sgn} x$; insert z into L;
  repeat
    remove an item z from L;
    insert z into K;
    if (5.7) is unbounded **then** *sing:=true*
    else if (5.7) is feasible **then** $L := L \cup (N(z) - (K \cup L))$
  until (*sing* or $L = \emptyset$);
if *sing* **then** $\{A^I$ is singular$\}$ **else** $\{A^I$ is regular$\}$.

In order to simplify the proof of the main theorem, we first formulate an auxiliary result concerning the case when singularity was not detected during the algorithm. Denote by C_0 the component of $X(A^I, b)$ containing the point $x_c = A_c^{-1}b$ and let Y_0 be the set of ± 1-vectors that were inserted into L in the course of the algorithm. Then Y_0 has this property:

LEMMA 6.1. *If $x \in C_0$ and $T_{z_0}x \geq 0$ for some $z_0 \in Y_0$, then each $z \in Z$ with $T_z x \geq 0$ belongs to Y_0.*

Proof. For the purpose of the proof, denote the linear programming problem (5.7) by $P(z)$. Then x is a feasible solution of some $P(\overline{z})$. Since $T_{z_0}x = |x| = T_{\overline{z}}x$, we can see from the form of (5.7) that x is a feasible solution of $P(z_0)$. Let $T_z x \geq 0$, $z \neq z_0$. Denote

$$J = \{j;\, z_j \neq (z_0)_j\} = \{j_1, \ldots, j_m\}.$$

Since $T_{z_0}x \geq 0$ and $T_z x \geq 0$, it must be $x_j = 0$ for each $j \in J$. Set $z^0 = z_0$ and define vectors $z^k \in Z$, $k = 1, \ldots, m$, in the following way:

$$(z^k)_j = \begin{cases} (z^{k-1})_j & \text{if } j \neq j_k, \\ -(z^{k-1})_j & \text{if } j = j_k \end{cases} \tag{6.1}$$

($k = 1, \ldots, m$, $j = 1, \ldots, n$). We shall prove by induction on $k = 0, \ldots, m$ that $z^k \in Y_0$ and x is a feasible solution of $P(z^k)$. This is obvious for $k = 0$. If the assumption is true for some $k - 1 \geq 0$, then $z^{k-1} \in Y_0$ and $P(z^{k-1})$ is feasible, hence $N(z^{k-1}) - (K \cup L)$ was added to L in the respective step. Since $z^k \in N(z^{k-1})$ by (6.1), z^k was either already present in $K \cup L$, or newly added to L, in both cases $z^k \in Y_0$. Furthermore, since x is a feasible solution of $P(z^{k-1})$ and $T_{z^{k-1}}x = T_{z^k}x$ holds as $x_{j_k} = 0$, it is also a feasible solution of $P(z^k)$. This concludes the proof by induction; since $z^m = z$, we have $z \in Y_0$. □

As it can be seen, this detailed proof is a formalization of the following idea: if we take a path from x_c to $x \in C_0$, then the only change of signs occurs when the path passes through a point with one or more zero components; all the respective sign vectors are added to L in the course of the algorithm, hence they belong to Y_0.

Now we finally prove that the algorithm really performs the task for which it was designed:

THEOREM 6.2. *For each $n \times n$ interval matrix A^I and each $b \in R^n$, the algorithm in a finite number of steps checks regularity or singularity of A^I.*

Proof. First, only elements of the finite set Z are being inserted into L and no element may be reinserted, hence the algorithm terminates in a finite number of steps. If some problem (5.7) is proved unbounded, then A^I is singular by Theorem 5.4. Hence, we only have to prove that if A^I has not been found singular and if the list L becomes empty, then A^I is regular.

The component C_0 may be written in the form (5.4):

$$C_0 = \bigcup_{z \in Y} X_z(A^I, b),$$

where Y may be chosen so that $X_z(A^I, b) \neq \emptyset$ for each $z \in Y$. We shall prove that

$$Y \subseteq Y_0 \tag{6.2}$$

holds. Since $L = \emptyset$, this will imply that all $X_z(A^I, b)$, $z \in Y$ have been checked to be bounded, hence A^I is regular by Theorem 5.4.

To prove (6.2), take a $\overline{z} \in Y$. Choose an $x \in X_{\overline{z}}(A^I, b)$, so that $T_{\overline{z}}x \geq 0$. Since C_0 is connected and contains $x_c = A_c^{-1}b$, there exists a path from x_c to x, contained entirely in C_0. In view of convexity of the sets $X_z(A^I, b), z \in Z$, the path may be chosen in a piecewise linear form $x^0x^1 \dots x^m$, where $x^0 = x_c$, $x^m = x$ and the segment with endpoints x^i, x^{i+1} is always a part of a single orthant ($i = 0, \dots, m-1$). We shall prove by induction on i that for each $i = 0, \dots, m$, if $T_z x^i \geq 0$ for some $z \in Z$, then $z \in Y_0$. Since $z_c = \operatorname{sgn} x_c$ is inserted into L at the beginning of the main loop and $T_{z_c}x_c \geq 0$, the assertion for $x^0 = x_c$ follows from Lemma 6.1. Let the assertion be true for x^i, $i \geq 0$. Since the whole segment with endpoints x^i and x^{i+1} is a part of a single orthant, there exists a $\tilde{z} \in Z$ such that $T_{\tilde{z}}x^i \geq 0$ and $T_{\tilde{z}}x^{i+1} \geq 0$. Then $\tilde{z} \in Y_0$ by assumption concerning x^i, hence from $T_{\tilde{z}}x^{i+1} \geq 0$, $\tilde{z} \in Y_0$ we obtain from Lemma 6.1 that each $z \in Z$ with $T_z x^{i+1} \geq 0$ belongs to Y_0. This concludes the proof by induction. Hence, since $T_{\overline{z}}x^m = T_{\overline{z}}x \geq 0$, we have $\overline{z} \in Y_0$. This proves (6.2). □

In the form presented above, the algorithm suffers a serious setback. If a linear program (5.7) is found bounded (i.e., it has an optimal solution), then *all* the neighbouring vectors $y \in N(z)$ are added to the list L (except those that have been already included). But we can see from the proof of Lemma 6.1 that we only need $N(z)$ to contain all the neighbouring vectors y satisfying $X_y(A^I, b) \cap X_z(A^I, b) \neq \emptyset$. Hence, vectors y with $X_y(A^I, b) \cap X_z(A^I, b) = \emptyset$ need not be included into $N(z)$, without affecting validity of Lemma 6.1 and of Theorem 6.2. In the next theorem we show how such vectors may be identified using information gathered from the current problem.

THEOREM 6.3. *Let for some $z \in Z$ the linear programming problem (5.7) have an optimal solution $\hat{x}$, and let $\hat{A} \in A^I$ be a nonsingular matrix satisfying $\hat{A}\hat{x} = b$. Then we have*

$$X_y(A^I, b) \cap X_z(A^I, b) = \emptyset$$

for each

$$y = (z_1, \dots, z_{j-1}, -z_j, z_{j+1}, \dots, z_n)^T \in N(z) \tag{6.3}$$

such that

$$(z^T\hat{x})(|\hat{A}^{-1}|\Delta e)_j < |\hat{x}_j|, \tag{6.4}$$

where e is the vector of all ones.

Proof. Assume to the contrary that there exists an $x \in X_y(A^I, b) \cap X_z(A^I, b)$. Then in view of (6.3) it must be

$$x_j = 0. \tag{6.5}$$

Since $x \in X(A^I, b)$, there exists an $A \in A^I$ such that $Ax = b$. From the identity

$$\hat{A}(x - \hat{x}) = (\hat{A} - A)x$$

we have

$$|x - \hat{x}| = |\hat{A}^{-1}(\hat{A} - A)x| \leq |\hat{A}^{-1}|\Delta|x|. \tag{6.6}$$

But since $x \in X_z(A^I, b)$, for each $i \in \{1, \dots, n\}$ there holds

$$|x_i| \leq \sum_k |x_k| = z^T x \leq z^T\hat{x},$$

hence

$$|x| \leq (z^T \hat{x})e,$$

and from (6.6) we obtain

$$|x - \hat{x}| \leq (z^T \hat{x})|\hat{A}^{-1}|\Delta e. \tag{6.7}$$

But from (6.4), (6.5) and (6.7) we have

$$|\hat{x}_j| = |x_j - \hat{x}_j| \leq (z^T \hat{x})(|\hat{A}^{-1}|\Delta e)_j < |\hat{x}_j|,$$

which is a contradiction. Hence, $X_y(A^I, b) \cap X_z(A^I, b) = \emptyset$. □

This result shows that the algorithm will remain in force if we include into $N(z)$ only those vectors y of the form (6.3) which *do not* satisfy (6.4). A matrix $\hat{A}$ satisfying $\hat{A}\hat{x} = b$ may be constructed by means of the proof of Theorem 4.1.

In the general description of the algorithm we did not specify the order in which the items are to be removed from the list L. For an efficient implementation it is recommendable, once a problem (5.7) has been solved, to check immediately all the problems (5.7) with y of the form (6.3) for j's not satisfying (6.4), since each of these linear programming problems differs from the original one in one column and one coefficient of the objective function only, so that the previously computed simplex tableau may be easily updated for the new problem. Practical experience shows that in this implementation the algorithm requires $O(pn^4)$ operations, where p is the number of linear programming problems (5.7) solved in the course of the algorithm.

7. The choice of b. It is obvious that the performance of the algorithm depends heavily on the number of orthants intersected by the component C_0. It is therefore a natural idea to try to find a right-hand side vector b such that C_0 would be entirely contained in a single orthant, preferably the nonnegative one. Unfortunately, it turns out that this is generally not possible, even in case of regular interval matrices. The following example is due to Nedoma; we quote here the result only, referring an interested reader to [12] for a proof:

EXAMPLE. Let $A^I = [A_c - \Delta, A_c + \Delta]$ with

$$A_c = \begin{pmatrix} 0 & 2 & 2 \\ 2 & 0 & 4 \\ 1 & 1 & 1 \end{pmatrix},$$

$$\Delta = \begin{pmatrix} 0 & \frac{3}{2} & \frac{3}{2} \\ \frac{3}{2} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Then A^I is regular and there does not exist a $b \in R^3$ satisfying

$$X(A^I, b) \subset (R^3_+)^0,$$

where $(R^3_+)^0$ is the interior of the nonnegative orthant in R^3.

Hence, we must set the goal differently: to find a b such that $X(A^I, b)$ intersects a possibly small number of orthants. Let $x_c = A_c^{-1}b$ as before and let $x \in X(A^I, b)$. Then there exists a matrix $A \in A^I$ with $Ax = b$, hence

$$x = x_c + A_c^{-1}(A_c - A)x. \tag{7.1}$$

We shall try to choose b such that $|x_c| = |A_c^{-1}b|$ is componentwise as large as possible. Then, in view of (7.1), x_c and the perturbed solution x should stay in the same orthant, or differ only in a few signs.

Because $A_c^{-1}(\alpha b) = \alpha A_c^{-1}b$, it follows that $|A_c^{-1}b|$ can be made arbitrarily large. Therefore it is sufficient to look only at right-hand sides $b \in R^n$ with $\|b\|_\infty \leq 1$. This leads to the optimization problem

$$\max\{\gamma;\ |A_c^{-1}b| \geq \gamma e,\ -e \leq b \leq e\}. \tag{7.2}$$

This problem can be solved exactly by applying 2^n linear programming calls. But for construction of an appropriate right-hand side we need only a reasonably good approximation of the optimal solution. Therefore, we proceed in the following way. First, we look at the discrete problem

$$\max\{\gamma;\ |A_c^{-1}b| \geq \gamma e,\ b \in Z\}. \tag{7.3}$$

The solution of (7.3) should be a good approximation of the solution of (7.2). Since each component of $|A_c^{-1}b|$ must be greater than or equal to γ, we may assume that A_c^{-1} is row equilibrated, i.e., the ith row of A_c^{-1} is multiplied by the factor $1/\max_j|(A_c)_{ij}^{-1}|$, $i = 1, \ldots, n$. Row scaling does not change regularity or singularity of an interval matrix A^I.

In order to calculate an approximation of an optimal solution of (7.3), we use a scheme which locally improves γ. We attempt to improve a current approximation b by looking for a superior solution b^{new} in an appropriate neighbourhood of b. In the first step $b, b^{new} \in Z$ are called neighboured if they differ in exactly one entry. The neighbour b^{new} is accepted as a new approximation if the corresponding new objective value γ^{new} is better than the previous one. Following we give a formal description (e_i denotes the ith column of the unit matrix):

$b := e$; $\gamma := \min_j |(A_c^{-1}b)_j|$;
for $k = 1, \ldots, k_1$ **do begin**
  $i := k \bmod(n) + 1$;
  $b^{new} := b - 2b_i e_i$; $\gamma^{new} := \min_j |(A_c^{-1}b^{new})_j|$;
  if $\gamma^{new} > \gamma$ **then begin** $b := b^{new}$; $\gamma := \gamma^{new}$ **end**
end.

This algorithm starts with $b = e$, calculates k_1 neighbours, and compares the corresponding objective values. We have chosen $k_1 = n^2$. Because

$$A_c^{-1}b^{new} = A_c^{-1}b - 2b_i {A_c}^{-1}e_i,$$

the computational costs of the previous algorithm are $O(n^3)$ operations. Notice that the statement $b^{new} := b - 2b_i e_i$ simply changes the ith entry of b from -1 to 1 or conversely.

In the second step we improve the previously calculated right-hand side b by taking the same algorithm, but changing the neighbourhood. Now $b, b^{new} \in Z$ are called neighboured if they differ in exactly two entries. The algorithm runs as follows:

for $k = 1, \ldots, k_2$ **do**
for $l = 1, \ldots, k_2$ **do begin**
  $i := k \bmod(n) + 1$; $h := l \bmod(n) + 1$;
  if $i \neq h$ **then begin**
    $b^{new} := b - 2b_i e_i - 2b_h e_h$; $\gamma^{new} := \min_j |(A_c^{-1}b^{new})_j|$;
    if $\gamma^{new} > \gamma$ **then begin** $b := b^{new}$; $\gamma := \gamma^{new}$ **end**

end
end.
We have chosen $k_2 = n$, and a similar argument as above shows that this algorithm needs $O(n^3)$ operations.

The last calculated b, γ in the previous algorithm, denoted by $\hat{b}, \hat{\gamma}$, serve as an approximation of an optimal solution of the problem (7.3). Because we intended to solve the problem (7.2), we solve in the third step the linear programming problem

$$\max\{\gamma;\, T_z A_c^{-1} b \geq \gamma e,\, -e \leq b \leq e\},$$

where $z = \operatorname{sgn}(A_c^{-1}\hat{b})$. Since $\hat{b}, \hat{\gamma}$ is a feasible solution of this problem, it follows immediately that its optimal solution b^* satisfies

$$\min_j |(A_c^{-1} b^*)_j| \geq \min_j |(A_c^{-1}\hat{b})_j|,$$

and we can use b^* as the desired right-hand side for the main algorithm of section 6.

8. Numerical experiments. In this section results of some numerical experiments are described. To demonstrate how the algorithm works and what are the computational costs of the method, we have also added some examples which are not typical, or where the algorithm does not behave well. The computational costs are proportional to the number p of linear programming problems (5.7) solved in the course of the algorithm. Therefore, we display in the following tables the number p. The program of our algorithm is written in MATLAB, and uses IEEE double floating point arithmetic.

We compare our algorithm with two sufficient regularity conditions. The first one (Beeck [3]) assures regularity of $A^I = [A_c - \Delta, A_c + \Delta]$ if

$$\varrho := \varrho(|A_c^{-1}|\Delta) < 1 \tag{8.1}$$

holds. The second one due to Rump [21], [22] states that A^I is regular if

$$\sigma := \frac{\sigma_{\max}(\Delta)}{\sigma_{\min}(A_c)} < 1, \tag{8.2}$$

where $\sigma_{\max}(\Delta), \sigma_{\min}(A_c)$ denote the maximal singular value of Δ and the minimal singular value of A_c, respectively. We shall see that in many cases both conditions are not satisfied, but A^I is regular. We shall also consider the radius of regularity of an interval matrix $A^I = [A_c - \Delta, A_c + \Delta]$ defined [16] by

$$r(A^I) = \inf\{\varepsilon \geq 0;\, [A_c - \varepsilon\Delta, A_c + \varepsilon\Delta] \text{ is singular}\}.$$

Hence, A^I is regular if and only if $r(A^I) > 1$.

In what follows we present results of computations of eight examples. In all of them A_c is fixed and Δ is varied depending on a real parameter κ; in most cases (examples 2 and 4 through 7) we set $\Delta = \kappa|A_c|$. It can be seen from the tables that p grows, or does not decrease, until the radius of regularity is reached. It is typical for the method that singularity can be proved very fast (often with $p = 1$) for many examples because singular matrices usually occur in almost all orthants. Nevertheless, there exist also problems where p increases even in the singular case (example 6).

EXAMPLE 1. The following example can be viewed as a generalization of the two-dimensional example 3.2 in [17] to the multi-dimensional case. The center matrix

A_c has nonzero entries only in the main diagonal, and above and below the k-th diagonals. We have chosen $n = 50, k = 48$ and A_c of the form

$$(A_c)_{ij} = \begin{cases} 50 & \text{if } i = j, \\ 100 & \text{if } j \geq i + 48, \\ -100 & \text{if } i \geq j + 48, \\ 0 & \text{otherwise} \end{cases}$$

$(i, j = 1, \ldots, 50)$. The radius matrix Δ is defined as follows:

$$(\Delta)_{ij} = \begin{cases} 40 & \text{if } i = j, \\ 0.01 + \kappa & \text{if } j \geq i + 48, \\ 0.01 + \kappa & \text{if } i \geq j + 48, \\ 0.01 & \text{otherwise} \end{cases}$$

$(i, j = 1, \ldots, 50)$. Hence, A^I has large intervals $[10, 90]$ on the diagonal, above and below the k-th diagonals the widths of the intervals are very large (about 2κ), and for the zero entries of A_c we have small intervals $[-0.01, 0.01]$. Table 1 shows behaviour of our algorithm and of the two sufficient regularity conditions for growing κ. We use an additional variable reg which is set to 1 if A^I is regular, and $reg = 0$ in case of singularity.

κ	reg	ϱ	σ	p
8	1	0.9290	1.0595	3
16	1	1.1141	1.3183	3
24	1	1.2979	1.5772	4
32	1	1.4809	1.8361	5
40	1	1.6634	2.0950	6
48	1	1.8454	2.3539	6
56	1	2.0271	2.6128	6
64	1	2.2085	2.8716	6
72	1	2.3897	3.1305	6
80	1	2.5708	3.3894	6
88	1	2.7518	3.6483	6
96	1	2.9327	3.9072	6
104	0	3.1135	4.1661	1
112	0	3.2942	4.4250	1
120	0	3.4748	4.6838	1
128	0	3.6554	4.9427	1
136	0	3.8360	5.2016	1
144	0	4.0165	5.4605	1
152	0	4.1970	5.7194	1
160	0	4.3774	5.9783	1

Table 1

We observe that with exception of $\kappa = 8$ both sufficient regularity conditions (8.1), (8.2) are not satisfied, but our algorithm proves regularity of A^I up to $\kappa = 96$, that is, the intervals above and below the k-th diagonal are $[3.99, 196.01]$. In this case ϱ is about 3, σ is about 4, and $p = 6$. Change from regularity to singularity occurs between $\kappa = 96$ and $\kappa = 104$. In this region we performed some additional experiments with step size 0.5. For the values $\kappa = 96.5, 97, 97.5, 98$ our algorithm proved regularity with $p = 6$, for $\kappa = 99.5, 100, \ldots, 103.5$ singularity was proved with $p = 1$, but for the values

$\kappa = 98.5$ and $\kappa = 99$ the algorithm run was stopped inconclusively after p reached 1000. For $\kappa \geq 104$ singularity of A^I was proved again with $p = 1$ (cf. the remark at the end of section 4). This shows that in a small region around $\kappa = r(A^I)$ the algorithm exhibits exponential behaviour (which is a consequence of the NP-hardness result of section 3), whereas outside this region the computational costs indicated by p are very moderate, and much smaller in case of singularity than in that of regularity. This typical behaviour (confirmed by many other experiments) constitutes the main advantage of our algorithm in contrast to necessary and sufficient regularity conditions [17, Thm. 5.1] which are exponential in n in *each* regularity case.

EXAMPLE 2. Here the center matrix A_c is the Hilbert matrix of dimension $n = 7$. The 2-norm condition number of this matrix is about $4.7 \cdot 10^8$. The radius matrix is $\Delta = \kappa|A_c|$, i.e., we consider relative perturbations.

κ	reg	ϱ	σ	p
1.0e-09	1	0.1185	0.4754	1
2.0e-09	1	0.2369	0.9507	1
3.0e-09	1	0.3554	1.4261	1
4.0e-09	1	0.4738	1.9015	1
5.0e-09	1	0.5923	2.3768	1
6.0e-09	1	0.7108	2.8522	1
7.0e-09	1	0.8292	3.3276	1
8.0e-09	1	0.9477	3.8029	1
9.0e-09	0	1.0661	4.2783	1
1.0e-08	0	1.1846	4.7537	1
1.1e-08	0	1.3031	5.2290	1
1.2e-08	0	1.4215	5.7044	1
1.3e-08	0	1.5400	6.1798	1
1.4e-08	0	1.6585	6.6551	1
1.5e-08	0	1.7769	7.1305	1
1.6e-08	0	1.8954	7.6059	1
1.7e-08	0	2.0138	8.0812	1
1.8e-08	0	2.1323	8.5566	1

Table 2

Table 2 shows that in all cases $p = 1$ and ϱ characterizes the radius of regularity $r(A^I)$ because the inverse Hilbert matrix has a rank 1 sign pattern (see [23]), whereas σ underestimates this radius.

EXAMPLE 3. This example is taken from [25], p. 442. The center matrix is given by

$$(A_c)_{ij} = \begin{cases} 11 - j & \text{if } j \geq i, \\ 10 - j & \text{if } j = i - 1, \\ 0 & \text{if } j < i - 1 \end{cases}$$

$(i, j = 1, \ldots, 10)$. The 2-norm condition number of A_c is about $2.8543 \cdot 10^7$. The coefficients of the radius matrix Δ are defined for all nonzero entries of A_c by $\Delta_{ij} = \kappa|(A_c)_{ij}|$, and are equal to 0.1 otherwise.

κ	reg	ϱ	σ	p
1.0e-08	1	0.1593	4.4539e+05	1
2.0e-08	1	0.2264	4.4539e+05	1
3.0e-08	1	0.2848	4.4539e+05	1
4.0e-08	1	0.3392	4.4539e+05	1
5.0e-08	1	0.3912	4.4539e+05	1
6.0e-08	1	0.4417	4.4539e+05	1
7.0e-08	1	0.4912	4.4539e+05	1
8.0e-08	1	0.5398	4.4539e+05	1
9.0e-08	1	0.5879	4.4539e+05	1
1.0e-07	1	0.6354	4.4539e+05	1
1.1e-07	1	0.6826	4.4539e+05	1
1.2e-07	1	0.7295	4.4539e+05	1
1.3e-07	1	0.7761	4.4539e+05	1
1.4e-07	1	0.8225	4.4539e+05	1
1.5e-07	1	0.8687	4.4539e+05	1
1.6e-07	1	0.9148	4.4539e+05	1
1.7e-07	1	0.9607	4.4539e+05	1
1.8e-07	0	1.0065	4.4539e+05	1
1.9e-07	0	1.0522	4.4539e+05	1
2.0e-07	0	1.0977	4.4539e+05	1

Table 3

Table 3 shows that in all cases $p = 1$, ϱ estimates very well the radius of regularity $r(A^I)$, whereas σ largely underestimates this radius.

EXAMPLE 4. This example is taken from [5], p. 41. The 10-dimensional center matrix is orthogonal and given by

$$(A_c)_{ij} = (2/(n+1))^{1/2} \sin(ij\pi/(n+1)),$$

and the radius matrix is defined by

$$\Delta = \kappa |A_c|.$$

κ	reg	ϱ	σ	p
0.025	1	0.2199	0.0741	1
0.050	1	0.4398	0.1483	1
0.075	1	0.6597	0.2224	1
0.100	1	0.8795	0.2966	1
0.125	1	1.0994	0.3707	10
0.150	1	1.3193	0.4449	10
0.175	1	1.5392	0.5190	10
0.200	1	1.7591	0.5931	35
0.225	1	1.9789	0.6673	45
0.250	1	2.1988	0.7414	74
0.275	1	2.4187	0.8156	113
0.300	1	2.6386	0.8897	145
0.325	1	2.8585	0.9639	202
0.350	1	3.0784	1.0380	249
0.375	0	3.2982	1.1121	15
0.400	0	3.5181	1.1863	2
0.425	0	3.7380	1.2604	1
0.450	0	3.9579	1.3346	1

Table 4

In contrast to the previous two examples we see from Table 4 that σ estimates regularity of A^I very well, whereas ϱ does not (the reason for behaviour of ϱ and σ for orthogonal matrices is explained in Rump [21], [22]). The number p initially grows up to 249, and then p rapidly decreases to one in the case of singularity. Moreover, we see that comparing this matrix with the matrix of example 3 yields types of matrices where both sufficient regularity conditions largely underestimate the radius of regularity, whereas our algorithm proves regularity or singularity of A^I with moderate computational costs.

EXAMPLE 5. The following example is taken from [23]. The center matrix A_c is defined by

$$(A_c)_{ij} = \begin{cases} 1 & \text{if } j = i \text{ or } j = i-1, \\ (-1)^{n+1} & \text{if } i = 1 \text{ and } j = n, \\ 0 & \text{otherwise} \end{cases}$$

$(i, j = 1, \ldots, n)$, and the radius matrix is given by

$$\Delta = \kappa |A_c|, \tag{8.3}$$

i.e., the coefficients of the center matrix are relatively perturbed. For this matrix Rump [24] showed that ϱ underestimates the radius of regularity of A^I by a factor which is equal to the dimension n. This is also demonstrated by the following Table 5 where $n = 10$.

κ	reg	ϱ	σ	p
0.1	1	1.000	0.639	5
0.2	1	2.000	1.279	53
0.3	1	3.000	1.918	61
0.4	1	4.000	2.557	118
0.5	1	5.000	3.196	118
0.6	1	6.000	3.836	118
0.7	1	7.000	4.475	118
0.8	1	8.000	5.114	118
0.9	1	9.000	5.753	118
1.0	0	10.000	6.393	2
1.1	0	11.000	7.032	1
1.2	0	12.000	7.671	1
1.3	0	13.000	8.310	1
1.4	0	14.000	8.949	1
1.5	0	15.000	9.589	1
1.6	0	16.000	10.228	1
1.7	0	17.000	10.867	1
1.8	0	18.000	11.506	1

Table 5

Moreover, we see that σ also underestimates the radius of regularity. Here p grows up to 118 which is about n^2 in case of regularity, and $p \leq 2$ in all cases of singularity. Changing just one coefficient in the above example to $(A_c)_{1n} := n(-1)^{(n+1)}$, and defining Δ again by (8.3), we can see from the following Table 6 that $p \leq 10$ whereas both ϱ and σ underestimate the radius of regularity.

κ	reg	ϱ	σ	p
0.1	1	0.714	5.437	4
0.2	1	1.428	10.873	8
0.3	1	2.141	16.310	9
0.4	1	2.855	21.746	10
0.5	1	3.569	27.183	10
0.6	1	4.283	32.619	10
0.7	1	4.997	38.056	10
0.8	1	5.710	43.492	10
0.9	1	6.424	48.929	10
1.0	0	7.138	54.365	1
1.1	0	7.852	59.802	1
1.2	0	8.566	65.238	1
1.3	0	9.279	70.675	1
1.4	0	9.993	76.111	1
1.5	0	10.707	81.548	1
1.6	0	11.421	86.984	1
1.7	0	12.135	92.421	1
1.8	0	12.848	97.857	1

Table 6

EXAMPLE 6. The center matrix is the 10-dimensional matrix defined by

$$(A_c)_{ij} = \begin{cases} 1 & \text{if } j \geq i, \\ -1 & \text{if } j < i, \end{cases}$$

and the radius matrix is

$$\Delta = \kappa |A_c|.$$

This example is not typical and it shows a behaviour which is contrary to most of our observations. As shown in Table 7, p is maximal in case of singularity, and moreover, in some singularity cases p increases for increasing κ.

κ	reg	ϱ	σ	p
0.02	1	0.2000	0.1975	1
0.04	1	0.4000	0.3951	1
0.06	1	0.6000	0.5926	1
0.08	1	0.8000	0.7902	1
0.10	1	1.0000	0.9877	55
0.12	0	1.2000	1.1852	123
0.14	0	1.4000	1.3828	124
0.16	0	1.6000	1.5803	69
0.18	0	1.8000	1.7778	81
0.20	0	2.0000	1.9754	20
0.22	0	2.2000	2.1729	20
0.24	0	2.4000	2.3705	20
0.26	0	2.6000	2.5680	21
0.28	0	2.8000	2.7655	21
0.30	0	3.0000	2.9631	21
0.32	0	3.2000	3.1606	21
0.34	0	3.4000	3.3581	2
0.36	0	3.6000	3.5557	2

Table 7

EXAMPLE 7. The following example is also not typical for most of our observations. It shows how the computational costs may change dramatically by altering slowly the dimension n. The center matrix is given by

$$(A_c)_{ij} = \begin{cases} 10 & \text{if } i < j, \\ 1 & \text{if } i = j, \\ -10 & \text{if } i > j, \end{cases}$$

and the radius matrix is

$$\Delta = \kappa |A_c|.$$

For $n = 7$ we get $p = 1$ and $\varrho = \sigma$ in all cases (Table 8).

κ	reg	ϱ	σ	p
0.005	1	0.3050	0.3050	1
0.010	1	0.6100	0.6100	1
0.015	1	0.9150	0.9150	1
0.020	0	1.2200	1.2200	1
0.025	0	1.5250	1.5250	1
0.030	0	1.8300	1.8300	1
0.035	0	2.1350	2.1350	1
0.040	0	2.4400	2.4400	1
0.045	0	2.7450	2.7450	1
0.050	0	3.0500	3.0500	1
0.055	0	3.3550	3.3550	1
0.060	0	3.6600	3.6600	1
0.065	0	3.9650	3.9650	1
0.070	0	4.2700	4.2700	1
0.075	0	4.5750	4.5750	1
0.080	0	4.8800	4.8800	1
0.085	0	5.1850	5.1850	1
0.090	0	5.4900	5.4900	1

Table 8

The results for $n = 8$ are displayed in Table 9.

κ	reg	ϱ	σ	p
0.005	1	0.2231	0.1595	1
0.010	1	0.4461	0.3189	3
0.015	1	0.6692	0.4784	5
0.020	1	0.8923	0.6378	5
0.025	1	1.1154	0.7973	10
0.030	1	1.3384	0.9567	10
0.035	1	1.5615	1.1162	10
0.040	0	1.7846	1.2756	1
0.045	0	2.0076	1.4351	1
0.050	0	2.2307	1.5945	1
0.055	0	2.4538	1.7540	1
0.060	0	2.6769	1.9134	1
0.065	0	2.8999	2.0729	1
0.070	0	3.1230	2.2324	1
0.075	0	3.3461	2.3918	1
0.080	0	3.5691	2.5513	1
0.085	0	3.7922	2.7107	1
0.090	0	4.0153	2.8702	1

Table 9

EXAMPLE 8. This example shows, for varying dimension n, results for interval matrices where the coefficients are normally distributed with mean 0 and variance 1. In detail, the center matrix is generated by $A_c = randn(n)$, $\kappa = 0.02 \cdot randn(n)$, and the radius matrix is given by $\Delta = \kappa \cdot randn(n)$. Here, $rand(n)$ is the MATLAB command for the normal random distribution, and for each test set the seed is set to 0. For dimension $n = 20$, we get

κ	reg	ϱ	σ	p
1.5000e-03	1	2.0127e-01	2.5422e-01	1
3.4675e-03	1	2.5216e-01	2.9794e-01	1
5.7830e-03	1	6.1268e-01	8.9690e-01	10
3.8228e-03	0	4.2597e+00	6.6856e+00	1
2.1062e-02	1	1.1923e+00	1.3357e+00	54
8.0028e-03	0	2.8315e+00	3.9616e+00	1
1.9898e-02	0	2.0265e+00	2.4730e+00	1
5.7956e-03	1	1.0729e+00	1.5084e+00	17
4.6480e-04	1	3.5748e-02	4.7923e-02	1
1.0293e-02	1	9.1469e-01	1.1683e+00	20

Table 10

dimension $n = 30$ yields

κ	reg	ϱ	σ	p
3.2528e-04	1	6.2265e-02	8.6597e-02	1
1.4271e-02	0	1.6529e+00	2.1586e+00	1
1.3444e-02	0	6.4642e+00	9.0395e+00	1
2.9834e-03	1	5.8821e-01	7.7027e-01	2
5.1537e-03	0	1.7320e+00	1.9743e+00	1
3.1523e-02	0	2.0272e+01	3.1276e+01	1
7.6462e-03	1	1.2505e+00	1.8812e+00	446
2.7498e-02	0	6.7397e+00	9.4327e+00	1
4.6382e-03	0	5.7781e+00	8.8484e+00	1
7.9532e-03	0	4.1317e+00	6.6051e+00	1

Table 11

and $n = 40$ yields

κ	reg	ϱ	σ	p
1.0966e-02	0	6.6685e+00	1.0145e+01	1
2.4906e-03	1	3.0435e-01	3.2999e-01	1
1.7104e-03	1	2.6830e-01	2.5902e-01	1
6.5490e-03	0	2.1063e+00	3.3164e+00	1
1.4050e-02	0	2.4355e+00	2.8584e+00	1
1.0242e-02	0	5.6521e+00	8.2005e+00	1
2.4428e-03	0	3.8025e+00	5.4907e+00	1
1.0429e-02	0	2.4616e+00	2.8986e+00	1
4.8682e-03	1	1.0540e+00	1.4246e+00	986
1.5887e-02	0	4.1782e+00	5.8441e+00	1

Table 12

We can see that for most of these randomly generated examples $p = 1$, and in the worst cases p is bounded by n^2.

9. Concluding remarks. As it could be seen from the examples presented in the last section, the algorithm proved to be efficient, exhibiting very moderate numbers of calls of the linear programming procedure on the average. This behaviour is to be ascribed to three features: (i) employing a new criterion of Theorem 5.3 which makes the algorithm not a priori exponential, (ii) using a fast heuristic algorithm for choosing a proper right-hand side b, and (iii) avoiding unnecessary calls of the linear programming procedure (Theorem 6.3).

Nevertheless, the computational work is still large compared to checking simple sufficient conditions (8.1) or (8.2). Therefore, for practical computations it is recommendable to use a hybrid algorithm which would first try the sufficient regularity conditions (8.1), (8.2), the sufficient singularity condition

$$\max_{ij}(|A_c^{-1}|\Delta)_{ij}(|A_c^{-1}|\Delta)_{ji} \geq 1$$

([23], Theorem 6.5) and the algorithm for finding a singular matrix in an interval matrix [18], and would resort to the actual algorithm of this paper only if all the previous checks fail. In this way the algorithm can be made even more efficient.

Acknowledgments. The work of the second author was supported by the Czech Republic Grant Agency under grant 201/98/0222. The authors wish to thank the two referees for a number of constructive comments and suggestions that helped to improve the text of this paper.

REFERENCES

[1] B. R. Barmish, *New tools for robustness analysis*, Proceedings of the 27th Conference on Decision and Control, Austin, TX, (1988), pp. 1–6.

[2] I. Bar-On, B. Codenotti and M. Leoncini, *Checking robust nonsingularity of tridiagonal matrices in linear time*, BIT, 36 (1996), pp. 206–220.

[3] H. Beeck, *Zur Problematik der Hüllenbestimmung von Intervallgleichungssystemen*, in Interval Mathematics, K. Nickel, ed., Lecture Notes in Computer Science 29, Berlin, 1975, Springer–Verlag, pp. 150–159.

[4] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP–Completeness*, Freeman, San Francisco, 1979.

[5] R. T. Gregory and D. L. Karney, *Collection of Matrices for Testing Computational Algorithms*, John Wiley & Sons, New York, 1969.

[6] C. Jansson, *On self-validating methods for optimization problems*, in Topics in Validated Computations, J. Herzberger, ed., Amsterdam, 1994, North–Holland, pp. 381–438.

[7] ———, *Calculation of exact bounds for the solution set of linear interval systems*, Linear Algebra and Its Applications, 251 (1997), pp. 321–340.

[8] V. Kreinovich, A. Lakeyev, J. Rohn and P. Kahl, *Computational Complexity and Feasibility of Data Processing and Interval Computations*, Kluwer, Dordrecht, 1997.

[9] J. Lunze, *Robust Multivariable Feedback Control*, Prentice–Hall, Englewood Cliffs, 1989.

[10] M. Mansour, *Robust stability of interval matrices*, Proceedings of the 28th Conference on Decision and Control, Tampa, FL, (1989), pp. 46–51.

[11] K. G. Murty, *Linear Complementarity, Linear and Nonlinear Programming*, Heldermann, Berlin, 1988.

[12] J. Nedoma, *Positively regular vague matrices*. Submitted to *Linear Algebra and Its Applications*.

[13] A. Nemirovskii, *Several NP-hard problems arising in robust stability analysis*, Mathematics of Control, Signals, and Systems, 6 (1993), pp. 99–105.

[14] W. Oettli and W. Prager, *Compatibility of approximate solution of linear equations with given error bounds for coefficients and right–hand sides*, Numerische Mathematik, 6 (1964), pp. 405–409.

[15] S. Poljak and J. Rohn, *Radius of nonsingularity*, Research Report, KAM Series 88–117, Faculty of Mathematics and Physics, Charles University, Prague, December 1988.

[16] ———, *Checking robust nonsingularity is NP–hard*, Mathematics of Control, Signals, and Systems, 6 (1993), pp. 1–9.

[17] J. Rohn, *Systems of linear interval equations*, Linear Algebra and Its Applications, 126 (1989), pp. 39–78.

[18] ———, *An algorithm for finding a singular matrix in an interval matrix*, Journal of Numerical Linear Algebra with Applications, 1 (1992), pp. 43–47.

[19] ———, *Positive definiteness and stability of interval matrices*, SIAM Journal on Matrix Analysis and Applications, 15 (1994), pp. 175–184.

[20] J. Rohn and G. Rex, *Interval P–matrices*, SIAM Journal on Matrix Analysis and Applications, 17 (1996), pp. 1020–1024.

[21] S. M. Rump, *Validated solution of large linear systems*, in Computing Supplementum 9, R. Albrecht, G. Alefeld and H. J. Stetter, eds., Wien, 1993, Springer, pp. 191–212.
[22] ———, *Verification methods for dense and sparse systems of equations*, in Topics in Validated Computations, J. Herzberger, ed., Amsterdam, 1994, North–Holland, pp. 63–135.
[23] ———, *Bounds for the componentwise distance to the nearest singular matrix*, SIAM Journal on Matrix Analysis and Applications, 18 (1997), pp. 83–103.
[24] ———, *Almost sharp bounds for the componentwise distance to the nearest singular matrix*, Linear and Multilinear Algebra, 42 (1998), pp. 93–108.
[25] H. Rutishauser, *Lectures on Numerical Analysis*, Birkhäuser, Basel, 1990.