Interval Computations
No 1, 1994

Dedicated to Johannes Weissinger
on the occasion of his 80th birthday

A Computer Aided Existence and Uniqueness Proof for an Inverse Matrix Eigenvalue Problem

Götz Alefeld and Günter Mayer

We present an algorithm which verifies the existence and within some tight interval bounds the uniqueness of a solution for the generalized additive inverse matrix eigenvalue problem.

Посвящается И. Вайсзингеру по
случаю его 80-летия

Компьютерное доказательство существования и единственности решения в обратной матричной проблеме собственных значений

Г. Алефельд, Г. Майер

Представлен алгоритм, проверяющий в некотором узком интервале существование и единственность решения в обобщенной обратной аддитивной матричной проблеме собственных значений.

We thank the Deutsche Forschungsgemeinschaft ('DFG') for supporting this work under grant Al 175/4–1 and Al 175/4–2.

1 Introduction

In [5] the following *generalized additive inverse (matrix) eigenvalue problem* was formulated (cf. also [4, 9, 10, 18, 24, 29]).

Problem 1. Given $n+1$ real $n \times n$ matrices A_i, $i = 0, \ldots, n$, and given n real numbers λ_i, $i = 1, \ldots, n$, find n real numbers c_i^*, $i = 1, \ldots, n$, such that the matrix

$$A(c) := A_0 + \sum_{i=1}^{n} c_i A_i, \quad c := (c_1, \ldots, c_n)^T \tag{1}$$

has the prescribed numbers λ_i as eigenvalues if $c_i = c_i^*$, $i = 1, \ldots, n$.

Often the matrices A_i are assumed to be symmetric (cf. [5], e.g.). We will also do this in our subsequent sections. But at the moment we allow A_i to be unsymmetric, too.

Taking for A_i in (1) the matrix $e^i(e^i)^T$, where e^i denotes the i-th column of the $n \times n$ identity matrix I, results in the following *additive inverse (matrix) eigenvalue problem* [6, 8, 10, 11, 12, 14, 28, 29], which thus is a special version of Problem 1.

Problem 2. Given a real matrix A_0 and given n real numbers λ_i, $i = 1, \ldots, n$, find a diagonal matrix $C^* := \mathrm{diag}(c_1^*, \ldots, c_n^*)$ such that $A := A_0 + C^*$ has the prescribed numbers λ_i as eigenvalues (where often the diagonal entries of A_0 are w.l.o.g. assumed to be zero).

Let $A_0 := 0$ and choose A_i in (1) as null matrix with the exception of the i-th row which is equal to $(e^i)^T A$ for a given real $n \times n$ matrix A. Then (1) yields to the well-known *multiplicative inverse (matrix) eigenvalue problem* [8, 11, 13, 14, 15].

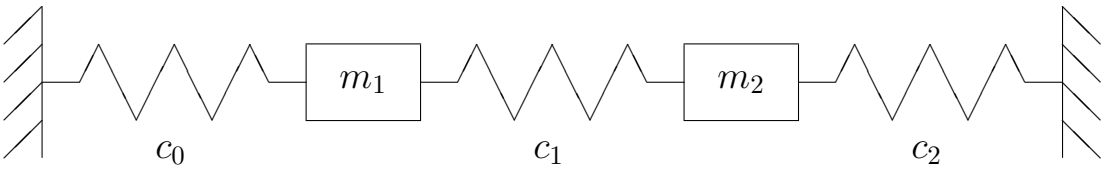
Problem 3. Given a real $n \times n$ matrix A and given n real numbers λ_i, $i = 1, \ldots, n$, find a diagonal matrix $C^* := \mathrm{diag}(c_1^*, \ldots, c_n^*)$ such that $C^* \cdot A$ has the prescribed numbers λ_i as eigenvalues.

Note that for symmetric matrices A Problem 3 is equivalent to finding a diagonal matrix C' such that $A \cdot C'$ has λ_i, $i = 1, \ldots, n$, as eigenvalues.

Often, A is assumed to be symmetric positive definite and C^* is required to have nonnegative diagonal elements (c.f. [8], e.g.).

We illustrate the Problems 1 and 3 by the following little example.

Example 1.1. Assume that we are given two point masses m_1, m_2 which are coupled by a spring of spring constant $c_1 > 0$. Assume that m_1 is coupled by a spring of constant $c_0 > 0$ with some rigid wall and that m_2 is similarly connected by a spring of constant $c_2 > 0$ with a second rigid wall standing parallel to the first one.



Denote by $x_i(t)$, $i = 1, 2$, the displacement of the mass m_i at the time t from its equilibrium position. Neglecting gravity and using Newton's second law [16], p. 55, as well as Hooke's law [16], p. 200, we get—by elementary rules of mechanics (i.e., by the superposition of the forces)—the following equations of motion for our system

$$\left\{ \begin{array}{rcl} m_1\ddot{x}_1(t) & = & -c_0x_1(t) + c_1\{x_2(t) - x_1(t)\} \\ m_2\ddot{x}_2(t) & = & -c_1\{x_2(t) - x_1(t)\} - c_2x_2(t). \end{array} \right. \tag{2}$$

Here, $\ddot{x}(t)$ denotes the second derivative of $x(t)$ with respect to the time t. Let

$$A := \begin{pmatrix} c_0 + c_1 & -c_1 \\ -c_1 & c_1 + c_2 \end{pmatrix},\ B := \begin{pmatrix} m_1 & 0 \\ 0 & m_2 \end{pmatrix},\ x(t) := \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}.$$

Then (2) can be written as

$$B\ddot{x}(t) = -Ax(t) \tag{3}$$

with the solutions

$$\begin{array}{rcl} x(t) & = & \alpha v^{(1)} \cos(\sqrt{\lambda_1}t) + \beta v^{(1)} \sin(\sqrt{\lambda_1}t) \\ & + & \gamma v^{(2)} \cos(\sqrt{\lambda_2}t) + \delta v^{(2)} \sin(\sqrt{\lambda_2}t) \end{array} \tag{4}$$

where λ_1, λ_2 are the eigenvalues of the generalized eigenvalue problem $Av = \lambda Bv$ with the symmetric positive definite matrices A and B. Thus λ_1, λ_2 are positive and eigenvalues of the matrix $B^{-1}A$; $v^{(1)}$, $v^{(2)}$ are any fixed eigenvectors associated with λ_1 and λ_2 respectively, and α, β, γ, δ are any real constants. Normally m_i, c_i are given, and one looks for λ_i, $v^{(i)}$ to get the solutions $x(t)$ of (2). In some cases, however, m_i, λ_i, and c_0 are prescribed whereas c_1, c_2 are to be determined in such a way that the solutions of (2) are given by (4). This means that B is known and A is unknown. Replacing $x(t)$ in (3) by the fundamental solutions $v^{(i)}\cos(\sqrt{\lambda_i}t)$, $v^{(i)}\sin(\sqrt{\lambda_i}t)$, $i = 1, 2$, yields

$$\begin{aligned}
\lambda_i v^{(i)} &= B^{-1}Av^{(i)} \\
&= \left\{ B^{-1}\begin{pmatrix} c_0 & 0 \\ 0 & 0 \end{pmatrix} + c_1 B^{-1}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} + c_2 B^{-1}\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \right\} v^{(i)} \\
&= A(c)v^{(i)}, \quad i = 1, 2
\end{aligned}$$

with

$$\begin{aligned}
A(c) &:= A_0 + c_1 A_1 + c_2 A_2, \quad A_0 := B^{-1}\begin{pmatrix} c_0 & 0 \\ 0 & 0 \end{pmatrix}, \\
A_1 &:= B^{-1}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}, \quad A_2 := B^{-1}\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}
\end{aligned}$$

where $c := (c_1, c_2)^T$ has to be computed. We have thus arrived at a generalized additive inverse eigenvalue problem with matrices A_i, $i = 1, 2, 3$, which are symmetric if $m_1 = m_2$.

Interpreted physically, the problem means the following:

Given the masses, the eigenfrequencies of the system and one spring constant, say c_0, (in order to equal unknowns and equations and thus to keep the chance for uniqueness), adjust the remaining spring constants in such a way that the system oscillates with the prescribed eigenfrequencies $\sqrt{\lambda_1}$, $\sqrt{\lambda_2}$.

Assume now c_i, λ_i to be given and m_i to be determined such that the solutions of (2) can again be expressed by (4). This means, that the spring constants are now given whereas the masses are to be computed to make the system oscillate with the prescribed eigenfrequencies $\sqrt{\lambda_1}$, $\sqrt{\lambda_2}$. We are again led to the problem

$$\lambda_i v^{(i)} = B^{-1}Av^{(i)}, \quad i = 1, 2 \tag{5}$$

where this time the diagonal matrix B is unknown and the matrix A is known. It is obvious that (5) is just a multiplicative inverse eigenvalue problem.

The example above can easily be generalized to a system with n masses and $n+1$ springs with constants c_i, resulting in an $n \times n$ diagonal matrix B and a symmetric $n \times n$ matrix A.

There are several papers which consider existence and uniqueness of the Problems 1–3 provided that the matrices A_i and A are symmetric. We cite [7, 8, 11, 13, 14, 28] without claiming to be exhaustive. Thus [11] derives a necessary criterion for the existence and non-existence, respectively of the Problem 2. In [14] a sufficient criterion is presented for the existence and uniqueness of Problem 2. In [8] it is shown that the Problem 2 has at least one and at most $n!$ solutions if one admits complex values for c_i. It is also remarked that even in the 2×2 case no solution needs to exist for this problem if one restricts c_i to be real. Problem 3 has at least one and at most $n!$ solutions if all the principal minors of A differ from zero and if one admits again complex values for c_i. Note that the first of these two assumptions is satisfied for all symmetric positive definite matrices. Again it is remarked that no solution needs to exist for Problem 3, if one restricts c_i to be real.

In [4, 5, 9, 10, 12, 18], Newton's method is used to construct approximations c_n to a solution c^* of the Problem 1 where c^* is tacitly assumed to exist. As an inherent fact of the Newton method it is remarked that the sequence $\{c_n\}$ converges to c^* if one starts the iteration *sufficiently close* to c^*. The goal of our paper consists in deriving an algorithm which verifies *automatically* the existence and within some interval bounds also the uniqueness for the following slightly specialized version of Problem 1.

Problem 1′. Given $n+1$ real $n \times n$ symmetric matrices A_i, $i = 0, 1, \ldots, n$, and given n real numbers

$$\lambda_1 < \lambda_2 < \cdots < \lambda_n \tag{6}$$

find n real numbers c_i^*, $i = 1, \ldots, n$, such that the matrix $A(c)$ from (1) has the prescribed numbers λ_i as eigenvalues for $c_i = c_i^*$, $i = 1, \ldots, n$.

Our algorithm starts with the Newton method as described in [10], e.g., which, in practice, stops with a vector $\tilde{c}$, viewed to be an approximation of

c^*. Unless additional considerations are done (e.g., check whether the assumptions of Kantorovich's theorem are fulfilled) nothing can be said about the existence of c^* by this purely computational process. In the sequel, we will use interval arithmetic combined with Brouwer's fixed point theorem to verify c^* in some neighbourhood U of $\tilde{c}$ or to get informed that no solution c^* exists in U. In the first case the algorithm even guarantees the uniqueness of c^* with respect to U.

We have organized our paper as follows. In Section 2 we introduce some definitions and notations. In Section 3 we recall some results needed to understand the algorithm which is presented in Section 4. In Section 5 we illustrate this algorithm by numerical examples.

2 Preliminaries

By $\boldsymbol{R}^n$, $\boldsymbol{R}^{m\times n}$, $\boldsymbol{IR}$, $\boldsymbol{IR}^n$, $\boldsymbol{IR}^{m\times n}$ we denote the set of real vectors with n components, the set of real $m \times n$ matrices, the set of intervals, the set of interval vectors with n components and the set of $m \times n$ interval matrices, respectively. By "interval" we always mean a real compact interval. Interval vectors and interval matrices are vectors and matrices, respectively, with intervals as entries. We write intervals in brackets with the exception of degenerate intervals (so-called point intervals) which we identify with the element being contained, and we proceed similarly with interval vectors and interval matrices. Examples are the unit matrix I and its i-th column e^i. If necessary, we identify the elements of $\boldsymbol{R}^{n\times 1}$ and $\boldsymbol{IR}^{n\times 1}$ in the usual way with those of $\boldsymbol{R}^n$ and $\boldsymbol{IR}^n$, respectively. We use the notation $[a] = [\underline{a}, \overline{a}] \in \boldsymbol{IR}$ simultaneously without further reference, and in an analogous way we write $[x] = [\underline{x}, \overline{x}] = ([x_i]) \in \boldsymbol{IR}^n$ and $[A] = [\underline{A}, \overline{A}] = ([\underline{a}_{ij}, \overline{a}_{ij}]) = ([a_{ij}]) \in \boldsymbol{IR}^{n\times n}$. By $\mathrm{int}([x])$ we denote the topological interior of the interval vector $[x]$, by $\check{x}$ we mean its midpoint $\check{x} := (\underline{x} + \overline{x})/2 \in \boldsymbol{R}^n$. We proceed similarly for interval matrices $[A]$ for which we introduce, in addition, the non-negative matrix $|[A]| = (c_{ij}) \in \boldsymbol{R}^{n\times n}$ by $c_{ij} := \max\{|\underline{a}_{ij}|, |\overline{a}_{ij}|\}$.

We equip $\boldsymbol{IR}$ with the usual arithmetic, i.e., we define

$$[a] \circ [b] := \{\tilde{a} \circ \tilde{b} |\ \tilde{a} \in [a],\ \tilde{b} \in [b]\} \tag{7}$$

for $\circ \in \{+, -, \cdot, /\}$, with $0 \notin [b]$ in case of division. It is easily seen that (7) can be expressed by

$$\begin{aligned}
[a] + [b] &= [\underline{a} + \underline{b}, \overline{a} + \overline{b}], \\
[a] - [b] &= [\underline{a} - \overline{b}, \overline{a} - \underline{b}], \\
[a] \cdot [b] &= [\min\ S, \max\ S] \ \text{ with } S := \{\underline{a}\ \underline{b}, \underline{a}\overline{b}, \overline{a}\underline{b}, \overline{a}\overline{b}\}, \\
[a] \,/\, [b] &= [a] \cdot [\frac{1}{\overline{b}}, \frac{1}{\underline{b}}].
\end{aligned}$$

We recall, that $(\boldsymbol{IR}, +, \cdot)$ is not a field. But it turns out (cf. [3]) that $(\boldsymbol{IR}, +)$, $(\boldsymbol{IR}, \cdot)$ are commutative semigroups with the neutral element 0 and 1, respectively. Addition and multiplication are related by the so-called subdistributivity law

$$[a] \cdot ([b] + [c]) \subseteq [a] \cdot [b] + [a] \cdot [c]. \tag{8}$$

Equality seldom holds in (8). We point out, that for non-degenerate intervals, inverses are missing. Thus, equations cannot be handled as usual.

The product $[C] = [A] \cdot [B]$ of two interval matrices $[A] \in \boldsymbol{IR}^{m \times n}$, $[B] \in \boldsymbol{IR}^{n \times p}$ is defined entrywise by

$$[c_{ij}] := \sum_{k=1}^{n} [a_{ik}] \cdot [b_{kj}] \quad (i = 1, \ldots, m;\ j = 1, \ldots, p).$$

It is obvious by (7) that

$$\{A \cdot B |\ A \in [A], B \in [B]\} \subseteq [A] \cdot [B] \tag{9}$$

holds. Simple examples show, that "$\subseteq$" cannot be replaced by "$=$" in (9).

For the symmetric matrix $A(c) = A_0 + \sum_{i=1}^{n} c_i A_i$ of Problem $1'$ we introduce the eigenvalue vector $\lambda(c) = \big(\lambda_i(c)\big) \in \boldsymbol{R}^n$ where $\lambda_i(c)$, $i = 1, \ldots, n$, are the eigenvalues of $A(c)$. For each $c \in \boldsymbol{R}^n$ we assume $\lambda_i(c)$ to be ordered increasingly, i.e., $\lambda_1(c) \leq \lambda_2(c) \leq \cdots \leq \lambda_n(c)$, where multiple eigenvalues are counted according to their multiplicity. Assume just for the subsequent definitions that c^* exists.

By [25], p. 45, $\lambda_i(c)$ depends continuously on c. Observing (6) we therefore get

$$\lim_{c \to c^*} \lambda_i(c) = \lambda_i(c^*) = \lambda_i, \quad i = 1, \ldots, n.$$

In particular, there exists a neighbourhood $U^* \subseteq \boldsymbol{R}^n$ of c^* such that

$$\lambda_1(c) < \lambda_2(c) < \cdots < \lambda_n(c) \quad \text{for } c \in U^*. \tag{10}$$

For these vectors c let $q^i(c) \in \boldsymbol{R}^n$ be the eigenvector of $A(c)$ which corresponds to $\lambda_i(c)$ and which satisfies

$$||q^i(c)||_2^2 := \big(q^i(c)\big)^T q^i(c) = 1 \tag{11}$$

and sign $q_1^i(c) =$ sign $q_1^i(c^*)$ where we assume w.l.o.g. sign $q_1^i(c^*)$ to be non-zero. By [25], p. 45, eigenvectors of algebraic simple eigenvalues depend continuously on the entries of the matrix. Therefore we have

$$\lim_{c \to c^*} q^i(c) = q^i(c^*).$$

Although it is well-known, that $q^i(c)$ depends analytically on c we will not exploit this fact in order to prove the following lemma, which can already be found in [9].

Lemma 2.1. *For the Problem* $1'$ *and* $c \in U^*$ *we get*

$$\frac{\partial\Big(\big(q^i(c)\big)^T\ A(c)\ q^i(c)\Big)}{\partial\ c_j} = \big(q^i(c)\big)^T\ A_j\ q^i(c),\ i,j = 1,\ldots,n.$$

Proof. Let $t_0 > 0$ such that $c_t := c + t\ e^j \in U^*$, $0 \le |t| \le t_0$. Then $A(c_t)q^i(c_t) = \lambda_i(c_t)q^i(c_t)$ implies

$$\begin{aligned}\big(q^i(c)\big)^T A(c_t)q^i(c_t) &= \big(q^i(c)\big)^T A(c)q^i(c_t) - t\big(q^i(c)\big)^T A_j q^i(c_t)\\ &= \big(\lambda_i(c_t) - \lambda_i(c)\big)\big(q^i(c)\big)^T q^i(c_t) + \lambda_i(c)\big(q^i(c)\big)^T q^i(c_t).\end{aligned}$$

Dividing by $t \neq 0$ and taking into account $\big(q^i(c)\big)^T A(c) = \lambda_i(c)\big(q^i(c)\big)^T$ we obtain $\big(q^i(c)\big)^T A_j q^i(c_t) = \dfrac{\lambda_i(c_t) - \lambda_i(c)}{t}\big(q^i(c)\big)^T q^i(c_t)$. Letting t tend to zero and observing the continuity of $q^i(c_t)$ as well as (11) proves the lemma. □

In Section 3 we will use the function $f : U^* \to \boldsymbol{R}^n$ with the entries

$$f_i(c) := \big(q^i(c)\big)^T A(c)q^i(c) - \lambda_i, \quad i = 1,\ldots,n \tag{12}$$

and λ_i from Problem 1′. The zeros of f are clearly the solutions of this problem.

We will also consider the matrix $A(c)$ for interval arguments $[c] = ([c_i]) \in \boldsymbol{IR}^n$ i.e. $A([c]) = A_0 + \sum_{i=1}^n [c]_i A_i \in \boldsymbol{IR}^{n\times n}$. By $\lambda_i([c]) \in \boldsymbol{IR}$ and $q^i([c]) \in \boldsymbol{IR}^n$ we denote any supersets of $\{\lambda_i(c)|\ c \in [c]\}$ and $\{q^i(c)|\ c \in [c]\}$, respectively, where i is kept fixed.

In the subsequent section, we will show how one can get $\lambda_i([c])$ and $q^i([c])$. It is clear by (9) and by Lemma 2.1 that the Jacobian $f'(c)$ of $f(c)$ satisfies

$$f'(c) = \left(\left(q^i(c)\right)^T A_j q^i(c) \right) \in f'([c]) := \left(\left(q^i([c])\right)^T A_j q^i([c]) \right) \tag{13}$$

for $c \in [c]$.

We will use $f'([c])$ in connection with some version of the interval Newton method. To formulate this method, we define the vector

$$\mathrm{IGA}([C],[b]) \in \boldsymbol{IR}^n \tag{14}$$

to denote the vector which results from the interval Gaussian algorithm applied to $[C] \in \boldsymbol{IR}^{n\times n}$ and $[b] \in \boldsymbol{IR}^n$. For simplicity (and in order to be unique) we assume this algorithm to be performed without pivoting. Since $\mathrm{IGA}([C],[b])$ is obtained by formulae which read quite analogously to those of the standard (i.e., non-interval) Gaussian algorithm, it is clear by (7) that

$$\{C^{-1}b|\ C \in [C], b \in [b]\} \subseteq \mathrm{IGA}([C],[b])$$

holds whenever $\mathrm{IGA}([C],[b])$ exists; in particular, C^{-1} exists in this case for any matrix $C \in [C]$. Cf. [3] or [19] for more details concerning the definition, the existence and properties of $\mathrm{IGA}([C],[b])$.

3 The algorithm

We start this section by recalling some basic facts on the interval Newton method

$$[x]^{k+1} = N([x]^k, \tilde{x}^k) \cap [x]^k, \quad k = 0, 1, \ldots \tag{15}$$

with the interval Newton operator

$$N([x],\tilde{x}) := \tilde{x} - \mathrm{IGA}\big(g'([x]), g(\tilde{x})\big), \quad \tilde{x} \in [x] \in \boldsymbol{IR}^n. \tag{16}$$

Here, $g : [x]^0 \in \boldsymbol{IR}^n \to \boldsymbol{IR}^n$ is a real continuously differentiable vector function for which we seek a zero within $[x]^0$;

$$g' : \{[x] \in \boldsymbol{IR}^n |\ [x] \subseteq [x]^0\} \to \boldsymbol{IR}^{n \times n} \tag{17}$$

is an interval function for which

$$\{g'(x) |\ x \in [x]\} \subseteq g'([x])$$

holds if $[x] \subseteq [x]^0$. Assume that g' in (17) is given beforehand in order to make the definition (16) unique. Think, e.g., of an interval arithmetic evalution [3], Ch. 3, for a fixed expression $g'(x)$ of the Jacobian of g. Later on, we will apply the following results on the interval Newton method.

Theorem 3.1 [1]. *Let g, N, $[x]^0$ be given as above and assume $[x]^k$, $k = 1, 2, \ldots$ to be constructed by (15). If $[x]^0$ contains a zero x^* of g then the vectors $[x]^k$, $k = 1, 2, \ldots$ are well defined, each of them contains x^*, and they converge monotonically to the interval vector $[x]^* := \bigcap_{k=0}^{\infty} [x]^k$ which again encloses x^*.*

In [1] it is shown by an example due to Schwandt [30] that $\underline{x}^* \neq \overline{x}^*$ can occur for the bounds of $[x]^*$ from Theorem 3.1. Criteria which guarantee $\underline{x}^* = \overline{x}^* = x^*$ can also be found in [1]. They are fulfilled if $g'(x^*)$ is non-singular and if $[x]^0$ encloses x^* sufficiently tight.

Theorem 3.1 can be applied if (18) of the following theorem holds.

Theorem 3.2 [1]. *Let g and N be defined as above.*

a) If

$$N([x], \tilde{x}) \subseteq [x] \tag{18}$$

then $[x]$ contains exactly one zero of g. This zero is also contained in $N([x], \tilde{x})$.

b) If

$$N([x], \tilde{x}) \cap [x] = \emptyset \tag{19}$$

then $[x]$ contains no zero of g.

Condition (18) can often be fulfilled by using the so-called *ϵ-inflation* (cf. [26]) which modifies (15) in the following way.

Algorithm 3.3.
Choose ϵ as a small positive number and let k_{max} be some positive integer.

Step 1: *Compute an approximation x_{approx} of x^* using any nonlinear system solver. E.g., apply the floating point Newton method.*

Step 2: *Iterate according to the following pseudocode.*

$k := 0$
$[y] := [x_{approx}, x_{approx}]$
repeat
  $k := k + 1$
  $[x] := [y] + [-\epsilon, \epsilon][y] + [-\eta, \eta]$
  choose $\tilde{x} \in [x]$
  $[y] := N([x], \tilde{x})$
  if $[y] \cap [x] = \emptyset$ **then write** *('$[x]$ contains no zero')*
until $[y] \subseteq [x]$ **or** $k = k_{max}$.

Step 3: *If $k < k_{max}$ then continue iterating according to (15) until some stopping criterion is fulfilled. Otherwise stop or improve x_{approx} or change ϵ. In the two latter cases start the inflation process once more.*

In Step 2, η denotes a fixed small number (e.g. the smallest machine number) necessary to guarantee inflation if $[y]_i = 0$.

Without further knowledge on g, there is no guarantee to avoid $k = k_{max}$. But it is a wide numerical experience that (18) is achieved after one or two steps of inflation provided x^* exists and x_{approx} approximates x^* sufficiently well. In addition, there are also criteria which guarantee that ϵ-inflation yields (18) after finitely many steps of iterations according to Step 2 of Algorithm 3.3. Cf. [23] or [27] for details.

We remark that there are also results dealing with (19); cf. [1] and [20], e.g.

It is obvious that the Steps 1–3 form the base of an algorithm to verify a solution of Problem 1′ and to prove its uniqueness within some interval bounds. One only has to replace $g(x)$ by $f(c)$ from (12), $g'([x])$ by $f'([c])$ from (13), and $\tilde{x}$ by $\tilde{c}$. Unfortunately, $f'([c]) = \left(\left(q^i([c])\right)^T A_j q^i([c]) \right)$ is

not given by an arithmetic expression with respect to $[c]$. This complicates the problem. There are several methods to construct enclosures for the eigenvectors $q^i(c)$, $c \in [c]$; cf. [21] for an overview. For our numerical results we chose a method described in [2] for point matrices and generalized in [22] to interval matrices. The essential ideas are the following.

Let $(\tilde{x}, \tilde{\lambda})$ be an approximation of an eigenpair (x^*, λ^*) of a real $n \times n$ matrix A from a given interval matrix $[A]$, which, later on, will be $A([c])$. Without loss of generality assume $\tilde{x}_n = x_n^* = \alpha \neq 0$. For $x \in \boldsymbol{R}^n$ and $\lambda \in \boldsymbol{R}$ introduce the errors $\Delta x := x - \tilde{x}$, $\Delta\lambda := \lambda - \tilde{\lambda}$. Then $\Delta x_n = 0$ and

$$A(\tilde{x} + \Delta x^*) = (\tilde{\lambda} + \Delta\lambda^*)(\tilde{x} + \Delta x^*)$$

hence

$$0 = \tilde{\lambda}\tilde{x} - A\tilde{x} - (A - \tilde{\lambda}I)\Delta x^* + \tilde{x}\Delta\lambda^* + (\Delta\lambda^*)\Delta x^*. \tag{20}$$

Since $\Delta x_n^* = 0$, (20) is equivalent to

$$0 = r - B\Delta y^* + (\Delta y_n^*)(\Delta\hat{y}^*)$$

where

$$\left\{ \begin{array}{rcl} r & := & \tilde{\lambda}\tilde{x} - A\tilde{x}, \\ \Delta y^* & := & (\Delta x_1^*, \ldots, \Delta x_{n-1}^*, \Delta\lambda^*)^T, \\ \Delta\hat{y}^* & := & (\Delta x_1^*, \ldots, \Delta x_{n-1}^*, 0)^T, \end{array} \right. \tag{21}$$

and where $B \in \boldsymbol{R}^{n\times n}$ is equal to $A - \tilde{\lambda}I$ with the exception of the n-th column, which coincides with $-\tilde{x}$. With $C \in \boldsymbol{R}^{n\times n}$, (20) yields the fixed point form

$$\Delta y^* = Cr + (I - CB)\Delta y^* + C(\Delta y_n^*)\Delta\hat{y}^* \tag{22}$$

which is the base of the interval iterative process

$$[\Delta y]^{k+1} = g([\Delta y]^k) \cap [\Delta y]^k, \quad k = 0, 1, \ldots \tag{23}$$

where

$$g([\Delta y]) := Cr + (I - C[B])[\Delta y] + C([\Delta y_n][\Delta\hat{y}]). \tag{24}$$

Analogously to (21), $[\Delta\hat{y}]$ coincides with $[\Delta y]$ in the first $n-1$ components whereas the last component is defined to be zero. The matrix $[B] \in \boldsymbol{IR}^{n\times n}$ is equal to $[A] - \tilde{\lambda}I$ with the exception of the n-th column, which again coincides with $-\tilde{x}$.

With the notation above, and with (9) and (22) the following theorem can easily be proved for (23).

Theorem 3.4. *If $\Delta y^* \in [\Delta y]^0$ then $\Delta y^* \in [\Delta y]^k$, $k = 0, 1, \ldots$, and the iterates $[\Delta y]^k$ converge monotonically to the interval vector $[\Delta y]^* := \bigcap_{k=o}^{\infty} [\Delta y]^k$ which again encloses Δy^*.*

We add another result which resembles Theorem 3.2.

Theorem 3.5. *Let g be defined by (24) and let $[\Delta y] \in \boldsymbol{IR}^n$.*

a) *If*

$$g([\Delta y]) \subseteq int([\Delta y]) \tag{25}$$

then C is non-singular, and for each matrix $A \in [A]$ the vector $(\tilde{x} + [\Delta\hat{y}], \tilde{\lambda} + [\Delta y_n])$ contains exactly one eigenpair (x^*, λ^*), which depends on A, with $\tilde{x}_n = x_n^* = \alpha \neq 0$. In addition, (23) can be performed without intersection, when starting with $[\Delta y]^0 := [\Delta y]$.

b) *If*

$$g([\Delta y]) \cap [\Delta y] = \emptyset \tag{26}$$

then for no matrix $A \in [A]$ the vector $(\tilde{x} + [\Delta\hat{y}], \tilde{\lambda} + [\Delta y_n])$ contains an eigenpair (x^*, λ^*) with $x_n^* = \alpha$.

Proof.

a) is proved in [22] based on results from [26].

b) If the assertion is false then there is a matrix $A \in [A]$ and an eigenpair (x^*, λ^*) of A such that $x_n^* = \alpha$ and $\Delta y^* \in [\Delta y]$. By (22) and (9) we get

$$\Delta y^* = g(\Delta y^*) \in g([\Delta y^*])$$

which contradicts (26). □

Starting with an approximation of an eigenpair, (25) can often be fulfilled using an ϵ-inflation analogously to Algorithm 3.3. In our next theorem we show that ϵ-inflation used as in Algorithm 3.3 is superfluous if $[\Delta y]^0$ can be chosen in a particular way.

Theorem 3.6 [2]. *Denote by $||x||$ the maximum norm for $x \in \boldsymbol{R}^n$, and use the same symbol for its associated matrix norm. Define*

$$\rho := ||Cr||, \quad \sigma := ||\,|I - C[B]|\,||, \quad \tau := ||C||.$$

Assume

$$\sigma < 1 \quad \textit{and} \quad (1-\sigma)^2 - 4\rho\tau > 0. \tag{27}$$

Then

$$\beta_\pm := \frac{1 - \sigma \pm \sqrt{(1-\sigma)^2 - 4\rho\tau}}{2\tau}$$

are real numbers, and

$$[\Delta y] := [-\beta, \beta] \cdot (1, 1, \ldots, 1)^T \tag{28}$$

fulfills (25), provided $\beta \in (\beta_-, \beta_+)$. If $[A]$ is a point matrix and if β is chosen from $\big(\beta_-, (\beta_- + \beta_+)/2\big)$ then $[\Delta y]^* = [\Delta y^*, \Delta y^*]$ for the limit of (23), with Δy^* from (22).

Note that the assumptions (27) certainly hold if $C \approx \check{B}^{-1}$, if the diameter of $[B]$ is sufficiently small, and if $(\tilde{x}, \tilde{\lambda})$ is for each matrix $A \in [A]$ a good approximation of an eigenpair (x^*, λ^*). If the last two conditions hold and if $[A]$ is degenerate, it is shown in [2], that $\check{B}^{-1}$ exists, provided λ^* is an algebraically simple eigenvalue. We remark that (10) guarantees this latter property at least near a solution of Problem $1'$.

If (23) is applied to $A([c])$ with $[\Delta y]^0$ satisfying (25) then one gets iterates $[x]^k := \tilde{x} + [\Delta\hat{y}]^k$ which enclose eigenvectors $x(c)$ of $A(c)$, $c \in [c]$, normalized by $x_n(c) = \alpha$. To get enclosures $q^i([c])$ of $q^i(c)$, $c \in [c]$, we must take into account the normalization (11). This can be done in several ways. To this end define

$$h(x) := \frac{x}{\sqrt{\sum_{j=1}^n x_j^2}} \quad \text{for } x \in \boldsymbol{R}^n \backslash \{0\}.$$

Replacing x by $[x] \in \boldsymbol{IR}^n$ with $0 \notin [x]$, yields an interval function $h([x])$ in which the squares and the root are computed for $[a] \in \boldsymbol{IR}$ as

$$[a]^2 := \{\ \tilde{a}^2 \mid \tilde{a} \in [a]\ \},$$

and

$$\sqrt{[a]} := \{\ \sqrt{\tilde{a}} \mid \tilde{a} \in [a]\ \}, \tag{29}$$

where we assume $\underline{a} \geq 0$ in (29). It is easily seen that $h([x])$ contains all unit vectors (with respect to the Euclidean norm) of the vectors from $[x]$. Unfortunately, $h([x])$, often highly overestimates the range $H := \{h(x) \mid x \in [x]\}$. To get tighter bounds, let $h'(x)$ denote the Jacobian of h at x. Replace x_j by $[x]_j$ in $h'(x)$, $j = 1, \ldots, n$, and denote the result by $h'([x])$. Choose $\tilde{x} \in [x]$. Then it can be shown (see [3], e.g.) that the so-called mean value form

$$h_{mvf}([x]) := h(\tilde{x}) + h'([x])([x] - \tilde{x})$$

encloses H in most cases tighter than $h([x])$, provided $\underline{x}$ is sufficiently close to $\overline{x}$. Therefore we will use the intersection

$$h([x]) \cap h_{mvf}([x]) \tag{30}$$

as enclosure for H.

Now we are able to construct $q^i([c])$ using (23) and (30) in the following way.

Algorithm 3.7.

Step 1: *Compute an approximation $(\tilde{x},\ \tilde{\lambda})$ for the eigenpair $\big(q^i(\check{c}), \lambda_i(\check{c})\big)$ of $A(\check{c})$ using any standard algorithm. E.g., apply a software package like LAPACK.*
Set $\alpha := \tilde{x}_n$.

Step 2: *If (27) is fulfilled, then compute $[\Delta y]$ according to Theorem 3.6 with $\beta \in (\beta_-, \beta_+)$ close to β_-.*
Otherwise use ϵ-inflation starting with $(\tilde{x},\ \tilde{\lambda})$ and proceeding analogously to Step 2 in Algorithm 3.3.

Step 3: *If (25) has been fulfilled in Step 2, then continue iterating according to (23) (without intersection) until some stopping criterion is fulfilled, ending up with some vector $[\Delta y]^{k_0}$.*

Step 4: *Compute*

$$[x] := \tilde{x} + [\Delta \hat{y}]^{k_0}, \quad [\lambda] := \tilde{\lambda} + [\Delta y_n]^{k_0}.$$

Step 5: *Compute*

$$q^i([c]) := h([x]) \cap h_{mvf}([x])$$

as in (30).

We formulate now the complete algorithm for verifying and enclosing c^* of Problem $1'$.

Algorithm 3.8.

a) Approximation $\tilde{c}$ of c^* by using Newton's method

1. *Choose $c := c^0 \in \boldsymbol{R}^n$.*
2. *Compute $\lambda(c)$, $q^i(c)$, $i = 1, \ldots, n$.*
3. *Compute $f(c) = \lambda(c) - \lambda^* \in \boldsymbol{R}^n$ and $f'(c) = \big((q^i(c)^T A_j q^i(c)\big) \in \boldsymbol{R}^{n\times n}$.*
4. *Solve $f'(c)(\tilde{c} - c) = -f(c)$. ("Newton step")*
5. *If $\tilde{c}$ does not fulfill some given stopping criterion, then set $c := \tilde{c}$ and goto Step 2 or stop.*

b) Verification part and improvement of the bounds

6. *$[c] := [\tilde{c}, \tilde{c}]$; inclusion := false.*
7. *$[c] := [c] + [-\epsilon, \epsilon][c] + [-\eta, \eta]$. ("$\epsilon$-inflation")*
8. *Compute $A([c])$ and $A(\check{c})$.*
9. *Compute $\lambda(\check{c})$ and $f(\check{c}) = \lambda(\check{c}) - \lambda^* \in \boldsymbol{R}^n$.*
10. *Compute and normalize $[x]^i([c])$, $i = 1, \ldots, n$, ending up with $q^i([c])$.*
11. *Compute $f'([c]) := \big(q^i([c])^T A_j q^i([c])\big) \in \boldsymbol{IR}^{n\times n}$.*
12. *Compute $[c]' := \check{c} - IGA\big(f'([c]),\ f(\check{c})\big)$. ("Newton step")*
13. *If inclusion = false and if*

$$\begin{cases} [c]' \subseteq [c], & \text{set inclusion := true} \\ & \text{("verification succeeded").} \\ [c]' \not\subseteq [c], & \text{set } [c] := [c]', \text{ goto Step 7 or stop} \\ & \text{("verification failed").} \end{cases}$$

14. Set $[c] := [c]' \cap [c]$ *and goto Step 8 or stop.*

We remark that $\check{c}$ in Step 9 and Step 12 can be replaced by any vector c from $[c]$.

If *inclusion* = *true* for the first time (cf. 13. in Algorithm 3.8) then by Theorem 3.2 we have verified a unique solution c^* of Problem $1'$ within $[c]$ and also within $[c]'$. This finishes the verification part of the algorithm, and the phase of improving the bounds $\underline{c}'$, $\overline{c}'$ for c^* starts by continuing the loop 8–14 until some stopping criterion is fulfilled.

4 Numerical results

We apply now Algorithm 3.8 on several examples. We used the scientific programming language PASCAL–XSC on a HP 720 workstation. PASCAL–XSC allows directed roundings and outward rounded interval arithmetic, cf. [17].

Example 4.1.

$$A_0 = \begin{pmatrix} 6 & 1 & 3 & -2 & 0 \\ 1 & 2 & 2 & 0 & 4 \\ 3 & 2 & 1 & 2 & 0 \\ -2 & 0 & 2 & -2 & 0 \\ 0 & 4 & 0 & 0 & -3 \end{pmatrix}, \quad A_1 = \begin{pmatrix} 2 & 1 & 0 & -1 & 1 \\ 1 & 0 & -4 & -1 & 0 \\ 0 & -4 & -2 & 1 & 3 \\ -1 & -1 & 1 & 0 & 5 \\ 1 & 0 & 3 & 5 & -1 \end{pmatrix},$$

$$A_2 = \begin{pmatrix} 1 & 2 & -3 & 0 & -1 \\ 2 & -1 & -3 & 1 & 0 \\ -3 & -3 & 0 & -2 & 2 \\ 0 & 1 & -2 & 0 & 6 \\ -1 & 0 & 2 & 6 & 1 \end{pmatrix}, \quad A_3 = \begin{pmatrix} 2 & -1 & 0 & 2 & 1 \\ -1 & 2 & 1 & 0 & -6 \\ 0 & 1 & -3 & 8 & -3 \\ 2 & 0 & 8 & 6 & -3 \\ 1 & -6 & -3 & -3 & 4 \end{pmatrix},$$

$$A_4 = \begin{pmatrix} -3 & -2 & 2 & 0 & 4 \\ -2 & 1 & 2 & -4 & 0 \\ 2 & 2 & -2 & -1 & 2 \\ 0 & -4 & -1 & -5 & 0 \\ 4 & 0 & 2 & 0 & 1 \end{pmatrix}, \quad A_5 = \begin{pmatrix} -3 & -1 & -5 & 3 & 2 \\ -1 & 2 & 7 & -1 & -2 \\ -5 & 7 & 5 & -3 & 0 \\ 3 & -1 & -3 & 0 & -2 \\ 2 & -2 & 0 & -2 & 4 \end{pmatrix}.$$

$\lambda^* = (-10, -5, -1, 4, 10)^T$.

Starting vector: $c^0 = (-2.9, 4.1, 0.9, 2.01, -1.01)^T$.

Verified enclosure:

$$[c] = \begin{array}{l} [\ -3.000000000000002E+000,\ -2.999999999999999E+000\ ] \\ [\ \ 3.999999999999999E+000,\ \ 4.000000000000001E+000\ ] \\ [\ \ 9.999999999999996E-001,\ \ 1.000000000000001E+000\ ] \\ [\ \ 1.999999999999999E+000,\ \ 2.000000000000001E+000\ ] \\ [\ -1.000000000000001E+000,\ -9.999999999999997E-001\ ] \end{array}$$

Exact $c^* = (-3, 4, 1, 2, -1)^T$, $A(c^*) = \begin{pmatrix} 3 & 2 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & -5 & 0 & 0 \\ 0 & 0 & 0 & -6 & 8 \\ 0 & 0 & 0 & 8 & 6 \end{pmatrix}$.

Example 4.2.

$$A_0 = \begin{pmatrix} -4 & 4 & -1 & 0 & 0 \\ 4 & -4 & -3 & 2 & -5 \\ -1 & -3 & -4 & -4 & -1 \\ 0 & 2 & -4 & 8 & -2 \\ 0 & -5 & -1 & -2 & 8 \end{pmatrix}, \qquad A_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

$$A_2 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}, \qquad A_3 = \begin{pmatrix} 2 & 0 & 0 & 0 & 1 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix},$$

$$A_4 = \begin{pmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & 0 & 0 & 0 & 1 \\ -1 & 0 & 3 & 2 & 1 \\ -1 & 0 & 2 & 0 & 1 \\ -1 & 1 & 1 & 1 & 2 \end{pmatrix}, \quad A_5 = \begin{pmatrix} -3 & 2 & -1 & 0 & -1 \\ 2 & 1 & -2 & 3 & -4 \\ -1 & -2 & 0 & 2 & 0 \\ 0 & 3 & 2 & 1 & -1 \\ -1 & -4 & 0 & -1 & 2 \end{pmatrix}.$$

$\lambda^* = (-8, -4, -3, 6, 7)^T$.

Starting vector: $c^0 = (-0.9, 1.1, -0.9, 0.9, -0.9)^T$.

Verified enclosure:

$$[c] = \begin{array}{c} [\\ [\\ [\\ [\\ [\end{array} \begin{array}{rcl} -1.000000000000001E+000 & , & -9.999999999999992E-001 \\ 9.999999999999999E-001 & , & 1.000000000000001E+000 \\ -1.000000000000001E+000 & , & -9.999999999999996E-001 \\ 9.999999999999999E-001 & , & 1.000000000000001E+000 \\ -1.000000000000001E+000 & , & -9.999999999999998E-001 \end{array} \begin{array}{c}] \\ ] \\ ] \\ ] \\ ] \end{array}$$

Exact $c^* = (-1, 1, -1, 1, -1)^T$, $A(c^*) = \begin{pmatrix} -4 & 2 & 0 & 0 & 0 \\ 2 & -7 & 0 & 0 & 0 \\ 0 & 0 & -3 & -3 & 0 \\ 0 & 0 & -3 & 5 & 0 \\ 0 & 0 & 0 & 0 & 7 \end{pmatrix}$.

Example 4.3.

$$A_0 = \begin{pmatrix} 0 & 4 & -1 & 1 & 1 & 5 & -1 & 1 \\ 4 & 0 & -1 & 2 & 1 & 4 & -1 & 2 \\ -1 & -1 & 0 & 3 & 1 & 3 & -1 & 3 \\ 1 & 2 & 3 & 0 & 1 & 2 & -1 & 4 \\ 1 & 1 & 1 & 1 & 0 & 1 & -1 & 5 \\ 5 & 4 & 3 & 2 & 1 & 0 & -1 & 6 \\ -1 & -1 & -1 & -1 & -1 & -1 & 0 & 7 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 0 \end{pmatrix}, \quad A_i = e^i (e^i)^T, \quad i = 1, \ldots, 8.$$

$\lambda^* = (10, 20, 30, 40, 50, 60, 70, 80)^T$.

Starting vector: $c^0 = (10, 20, 30, 40, 50, 60, 70, 80)^T$.

Verified enclosure:

$$[c] = \begin{array}{l} [\ 1.190787610247270E+001,\ \ 1.190787610247272E+001\ ] \\ [\ 1.970552150808698E+001,\ \ 1.970552150808700E+001\ ] \\ [\ 3.054549818697703E+001,\ \ 3.054549818697705E+001\ ] \\ [\ 4.006265748844803E+001,\ \ 4.006265748844805E+001\ ] \\ [\ 5.158714029072548E+001,\ \ 5.158714029072551E+001\ ] \\ [\ 6.470213143217948E+001,\ \ 6.470213143217953E+001\ ] \\ [\ 7.017067582089113E+001,\ \ 7.017067582089118E+001\ ] \\ [\ 7.131849917021904E+001,\ \ 7.131849917021909E+001\ ] \end{array}$$

In all three examples the following properties occur:

- All the entries of the matrices A_i are machine representable real numbers.
- The starting vector c^0 differs from the exact solution c^* by 10% approximately.
- The verification process needs only 1 inflation step. It was realized by using the function *blow*$([x], \epsilon)$ of PASCAL–XSC with $\epsilon = 0.1$ in Step 7 of Algorithm 3.8. By Theorem 3.2 it is guaranteed that the listed interval vector $[c]$ contains exactly one solution c^* of the Problem 1′.

- The improvement of the bounds stops whenever $[c]' = [c]$ holds in Algorithm 3.8, due to outward roundings on the computer. This part of the algorithm needs about half of the total time. Thus it is very time consuming. Since the bounds did not become much tighter, the improvement step is (at least for our examples) not very efficient and can be skipped.

Finally we remark, that Example 4.3 is taken from [10], where the approximation $\tilde{c} = (11.90788, 19.70552, 30.54550, 40.06266, 51.58714, 64.70213, 70.17068, 71.31850)^T$ of c^* is given which has been obtained by the Newton method. When the bounds of our verifying vector $[c]$ are rounded (outward) to seven significant digits, then the approximation $\tilde{c}$ in [10] is contained in $[c]$.

References

[1] Alefeld, G. *On the convergence of some interval-arithmetic modifications of Newton's method.* SIAM J. Numer. Anal. **21** (1984), pp. 363–372.

[2] Alefeld, G. *Berechenbare Fehlerschranken für ein Eigenpaar unter Einschluß von Rundungsfehlern bei Verwendung des genauen Skalarprodukts.* Z. angew. Math. Mech. **67** (1987), pp. 145–152.

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

[4] Biegler-König, F. *A Newton iteration process for inverse eigenvalue problems.* Numer. Math. **37** (1981), pp. 349–354.

[5] Bohte, Z. *Numerical solution of the inverse algebraic eigenvalue problem.* Comp. J. **10** (1967), pp. 385–388.

[6] Downing Jr., A. C. and Householder, A. S. *Some inverse characteristic value problems.* J. Assoc. Comput. Mach. **3** (1956), pp. 203–207.

[7] Friedland, S. *Matrices with prescribed off-diagonal elements.* Israel J. Math. **11** (1972), pp. 184–189.

[8] Friedland, S. *Inverse eigenvalue problems.* Linear Algebra Appl. **17** (1977), pp. 15–51.

[9] Friedland, S., Nocedal, J., and Overton, M. L. *Four quadratically convergent methods for solving inverse eigenvalue problems.* In: Griffith, Watson (eds) "Numerical Analysis. Pitman research notes in mathematical series". Longman Scientific and Technical, Harlow Essex, England, 1986, pp. 47–65.

[10] Friedland, S., Nocedal, J., and Overton, M. L. *The formulation and analysis of numerical methods for inverse eigenvalue problems.* SIAM J. Numer. Anal. **24** (1987), pp. 634–667.

[11] Hadeler, K. P. *Ein inverses Eigenwertproblem.* Linear Algebra Appl. **1** (1968), pp. 83–101.

[12] Hadeler, K. P. *Newton-Verfahren für inverse Eigenwertaufgaben.* Numer. Math. **12** (1968), pp. 35–39.

[13] Hadeler, K. P. *Multiplikative inverse Eigenwertprobleme.* Linear Algebra Appl. **2** (1969), pp. 65–86.

[14] Hadeler, K. P. *Existenz- und Eindeutigkeitssätze für inverse Eigenwertaufgaben mit Hilfe des topologischen Abbildungsgrades.* Arch. Rational Mech. Anal. **42** (1971), pp. 317–322.

[15] Heinrich, H. *Ein inverses Eigenwertproblem für endliche Matrizen und seine graphische Lösung für* $n = 3$. Z. Angew. Math. Mech. **40** (1960), pp. 62–64.

[16] Kittel, C., Knight, W. D., and Ruderman, M. A. *Mechanics. Berkeley Physics Course, Vol. 1.* McGraw-Hill, New York, 1965.

[17] Klatte, R., Kulisch, U., Neaga, M., Ratz, D., and Ullrich, Ch. *PASCAL–XSC. Sprachbeschreibung mit Beispielen.* Springer, Berlin, 1991.

[18] Kublanovskaya, W. N. *On an approach to the solution of the inverse eigenvalue problem.* Zapiski nauchnykh seminarov Leningradskogo otdeleniya matematicheskogo instituta im. V. A. Steklova Akademii nauk SSSR (1970), pp. 138–149 (in Russian).

[19] Mayer, G. *Old and new aspects for the interval Gaussian algorithm.* In: Kaucher, E., Markov, S. M., and Mayer, G. (eds) "Computer Arithmetic, Scientific Computation and Mathematical Modelling". IMACS Annals on Computing and Applied Mathematics, **12**, Baltzer, Basel, 1992, pp. 329–349.

[20] Mayer, G. *Some remarks on two interval-arithmetic modifications of the Newton method.* Computing **48** (1992), pp. 125–128.

[21] Mayer, G. *Enclosures for eigenvalues and eigenvectors.* In: Atanassova, L. and Herzberger, J. (eds) "Computer Arithmetic and Enclosure Methods". Elsevier (North-Holland), Amsterdam, 1992, pp. 49–68.

[22] Mayer, G. *A unified approach to enclosure methods for eigenpairs.* Z. angew. Math. Mech. **74** (1994), pp. 115–128.

[23] Mayer, G. *Epsilon-inflation in verification algorithms.* Submitted for publication.

[24] Nocedal, J. and Overton, M. L. *Numerical methods for solving inverse eigenvalue problems.* In: Pereyra, V. and Reinoza, A. (eds) "Numerical Methods. Proceedings, Caracas 1982". Lecture Notes in Mathematics **1005**. Springer, Berlin, 1983, pp. 212–226.

[25] Ortega, J. M. *Numerical analysis. A second course.* SIAM, Philadelphia, 1990.

[26] Rump, S. M. *Solving algebraic problems with high accuracy.* In: Kulisch, U. W. and Miranker, W. L. (eds) "A New Approach to Scientific Computation". Academic Press, New York, 1983, pp. 51–120.

[27] Rump, S. M. *On the solution of interval linear systems.* Computing **74** (1992), pp. 337–353.

[28] Šarman, S. *Bemerkungen zu inversen Eigenwertproblemen.* Computing **4** (1969), pp. 207–215.

[29] Scholtyssek, V. *Iterationsverfahren auf Mannigfaltigkeiten zur Lösung inverser Eigenwertprobleme.* Thesis, University of Karlsruhe, 1992.

[30] Schwandt, H. *Schnelle fast global konvergente Verfahren für die Fünf-Punkt-Diskretisierung der Poissongleichung mit Dirichletschen Randbedingungen auf Rechteckgebieten.* Thesis, Technical University of Berlin, 1981.

Received: September 20, 1993
Revised version: March 7, 1994

G. Alefeld
Institut für Angewandte Mathematik
Universität Karlsruhe
D–76128 Karlsruhe
Germany

G. Mayer
Fachbereich Mathematik
Universität Rostock
D–18051 Rostock
Germany