Verified Solutions of Inverse Symmetric Eigenvalue Problems*

Shinya Miyajima
Faculty of Science and Engineering, Iwate University,
4-3-5 Ueda, Morioka-shi, Iwate 020-8551, Japan
miyajima@iwate-u.ac.jp

Abstract

An iterative algorithm for computing an interval vector containing the solution to the inverse symmetric eigenvalue problem is proposed. The iterative process in this algorithm involves only cubic complexity per iteration. Uniqueness of the contained solution can moreover be verified by the algorithm. Numerical resultsillustrate the properties of the algorithm. As an application of the proposed algorithm, an algorithm for enclosing a solution to an inverse singular value problem is also sketched.

Keywords: Inverse symmetric eigenvalue problems, Nonlinear systems, Numerical verification
AMS subject classifications: 65F15, 65G20, 65H17

1 Introduction

Let $A(c)$ be the affine family

$$A(c) = A_0 + \sum_{i=1}^{n} c_i A_i,$$

where $c \in \mathbb{R}^n$ and $A_0, \ldots, A_n$ are real symmetric $n \times n$ matrices. Denote the eigenvalues of $A(c)$ by $\lambda_1(c), \ldots, \lambda_n(c)$, where $\lambda_1(c) \leq \cdots \leq \lambda_n(c)$. The following is called the inverse symmetric eigenvalue problem treated in this paper:

Problem 1 *Given real numbers $\lambda_1^* < \cdots < \lambda_n^*$, find $c^* \in \mathbb{R}^n$ such that $\lambda_i(c^*) = \lambda_i^*$, $i = 1, \ldots, n$.*

Problem 1 arises in a variety of applications such as inverse Sturm-Liouville problems, the inverse vibrating string problem, nuclear spectroscopy and molecular spectroscopy [10]. There is a large amount of literature (e.g. [8, 9]) on conditions for existence and uniqueness of the solution to Problem 1.

*Submitted: October 4, 2016; Revised: April 12, 2017; Accepted: September 5, 2017.

Solving Problem 1 is equivalent to solving the equation $f(c) = 0$ in $\mathbb{R}^n$, where the function $f : \mathbb{R}^n \to \mathbb{R}^n$ is defined by

$$f(c) := (\lambda_1(c) - \lambda_1^*, \ldots, \lambda_n(c) - \lambda_n^*)^T. \tag{1}$$

If c^* exists, $\lambda_i(c)$, $i = 1, \ldots, n$ are continuously differentiable around c^* (see e.g. [21]). Therefore, so is f. The Jacobian matrix $J(c)$ of $f(c)$ is given such that (cf. [10])

$$J(c)_{ij} = \frac{\partial \lambda_i(c)}{\partial c_j} = q^{(i)}(c)^T A_j q^{(i)}(c), \quad i, j = 1, \ldots, n, \tag{2}$$

where $q^{(i)}(c)$ is a unit 2-norm eigenvector corresponding to $\lambda_i(c)$. For solving Problem 1, Newton and Newton-like methods exploiting $f(c)$ and $J(c)$ have been proposed (see [10] for example).

The work presented in this paper addresses the problem of computing verified solutions to Problem 1, specifically, computing an interval vector which is guaranteed to contain the solution. Numerical algorithms cannot provide exact solutions in general, and usually give approximations. One approach to providing reliability for the computed approximation is to numerically compute an interval containing the solution. Since the midpoint of the interval is usually the approximation, the radius can be regarded as an error bound. If the radius is small, therefore, we can conclude that the approximation is reliable. By utilizing the intervals containing the solutions, moreover, we can *mathematically prove* that the solution to Problem 1 for specific $A_0, \ldots, A_n$ and $\lambda_1^*, \ldots, \lambda_n^*$ is not unique. Specifically, if two intervals containing the solutions are disjoint, then we can assert *with mathematical rigor* that the specific problem has at least two solutions. No matter how accurate two numerical solutions are, distinctness of them does not prove non-uniqueness. In fact, non-uniqueness of the solution to an example of Problem 1 is proved in [1, Section 5] based on the verified computation, which is not noted in the previous paper [10].

Pioneering work for computing verified solutions to Problem 1 seems to be the iterative algorithms in [1, 2]. The algorithms in [1] and [2] are based on the Krawczyk [13] and interval Newton operators, respectively. Let $\tilde{c}$ and $\boldsymbol{z}$ be a numerical solution to Problem 1 and an interval n-vector, respectively. In both of the algorithms, an interval n-vector and $n \times n$ interval matrix containing $f(\tilde{c})$ and $\{J(c) : c \in \tilde{c} + \boldsymbol{z}\}$, respectively, are required. Since $f(c)$ and $J(c)$ are not *explicitly* represented as a function of c, computing these intervals is not trivial. In [2], the algorithm in [14] is adopted for obtaining the interval vector and matrix. The algorithm in [14] computes intervals containing an eigenvalue and its corresponding eigenvector of an interval matrix, and is summarized in [2, Algorithm 3.7]. This algorithm is also mentioned in [1] as one of the methods for obtaining the interval vector and matrix mentioned above. The iterative processes of the algorithms in [1, 2] require $\mathcal{O}(n^4)$ operations per iteration.

The purpose of this paper is to propose an iterative algorithm for computing verified solutions to Problem 1. This algorithm is also based on the Krawczyk operator, and requires the above interval vector and matrix. We develop new methods for obtaining them. With the aid of the new methods, the iterative process of the algorithm requires only $\mathcal{O}(n^3)$ operations per iteration, although it contains a process involving $\mathcal{O}(n^4)$ operations before the iterative process. The new methods are applications of the results developed in [18, 19]. Although new error bounds for the eigenvalue perturbation have been developed in past decades (see [15, 16, 17, 20, 28], e.g.), utilizing the results in [18, 19] seems to be more suitable for our purpose, since $A(c)$ is real

symmetric and all eigenvalues and eigenvectors must be considered. The proposed algorithm moreover verifies uniqueness of the contained solution.

As an application of the proposed algorithm, we also sketch an algorithm for enclosing a solution to an inverse singular value problem, which is stated as follows:

Problem 2 *Let $A(c) := A_0 + \sum_{i=1}^{n} c_i A_i$, where $c \in \mathbb{R}^n$, and $A_0, \ldots, A_n$ are real $m \times n$ matrices with $m \geq n$. Denote the singular values of $A(c)$ by $\sigma_1(c), \ldots, \sigma_n(c)$, where $\sigma_1(c) \leq \cdots \leq \sigma_n(c)$. Given real numbers $0 < \sigma_1^* < \cdots < \sigma_n^*$, find $c^* \in \mathbb{R}^n$ such that $\sigma_i(c^*) = \sigma_i^*$, $i = 1, \ldots, n$.*

Problem 2 has practical applications, such as construction of Toeplitz-related matrices from prescribed singular values [4], the inverse problem in some quadratic group [22], passivity enforcement in nonlinear circuit simulation [25], and the optimal sequence designed for direct-spread code division multiple access [29].

This paper is organized as follows: In Section 2, notation and theories used in this paper are introduced. In Sections 3 and 4, the methods for computing the above interval vector and matrix are established, respectively. In Section 5, the verification algorithm is proposed. In Section 6, numerical results are reported. In Section 7, the verification algorithm for the solution to Problem 2 is sketched. Section 8 finally summarizes the results in this paper and highlights possible extension and future work.

2 Preliminaries

Let $\mathbb{IR}$, $\mathbb{IR}^n$ and $\mathbb{IR}^{n\times n}$ be the sets of all real intervals, interval n-vectors and $n \times n$ interval matrices, respectively. For $\boldsymbol{a} \in \mathbb{IR}$, let $\mathrm{mid}\,(\boldsymbol{a})$ and $\mathrm{rad}\,(\boldsymbol{a})$ be the midpoint and radius of $\boldsymbol{a}$, respectively, and $|\boldsymbol{a}| := \max_{a\in\boldsymbol{a}} |a|$. For $v \in \mathbb{R}^n$, define $\|v\|_2 := \sqrt{\sum_i v_i^2}$. For $M = (M_{ij}) \in \mathbb{R}^{n\times n}$, let $|M| := (|M_{ij}|)$, $\|M\|_\infty := \max_i \sum_j |M_{ij}|$ and $\|M\|_1 := \max_j \sum_i |M_{ij}|$. We can then define $|\boldsymbol{v}| \in \mathbb{R}^n$ and $|\boldsymbol{M}| \in \mathbb{R}^{n\times n}$ for $\boldsymbol{v} \in \mathbb{IR}^n$ and $\boldsymbol{M} \in \mathbb{IR}^{n\times n}$, respectively. Define $\|\boldsymbol{v}\|_2 := \||\boldsymbol{v}|\|_2$, $\|\boldsymbol{M}\|_\infty := \||\boldsymbol{M}|\|_\infty$ and $\|\boldsymbol{M}\|_1 := \||\boldsymbol{M}|\|_1$. Let `eps`, `realmin` and I be machine epsilon, the smallest positive normalized floating point number (especially `eps` $= 2^{-52}$ and `realmin` $= 2^{-1022}$ in IEEE 754 double precision), and the $n \times n$ identity matrix, respectively, and $e := (1, \ldots, 1)^T \in \mathbb{R}^n$. For $M^c, M^r \in \mathbb{R}^{n\times n}$ with $\min_{i,j} M_{ij}^r \geq 0$, $\langle M^c, M^r \rangle$ denotes the interval matrix whose midpoint and radius are M^c and M^r, respectively. The notation $\mathrm{fl}(\cdot)$ denotes a result of floating point computation, where all operations inside the parentheses are executed by ordinary floating point arithmetic in rounding to nearest mode. The notations $\overline{\mathrm{fl}}(\cdot)$ and $\underline{\mathrm{fl}}(\cdot)$ denote rigorous upper and lower bounds for the insides of the parentheses obtained by rounding mode controlled floating point computations, respectively.

We cite Lemmas 1 to 5, and present Lemma 6. Lemma 1 is a modification of the Rump's theorem, whose statement and proof can be found in [20, Theorem 1].

Lemma 1 (Miyajima et al. [18]) *Let $A \in \mathbb{R}^{n\times n}$ be symmetric, $\tilde{q}^{(i)} \in \mathbb{R}^n \setminus \{0\}$ and $\tilde{\lambda}_i \in \mathbb{R}$, $i = 1, \ldots, n$ with $\tilde{\lambda}_1 \leq \cdots \leq \tilde{\lambda}_n$ be given, λ_i be the eigenvalues of A such that $\lambda_1 \leq \cdots \leq \lambda_n$, $\tilde{Q} := (\tilde{q}^{(1)}, \ldots, \tilde{q}^{(n)})$, $\tilde{D} := \mathrm{diag}(\tilde{\lambda}_1, \ldots, \tilde{\lambda}_n)$, $R := A\tilde{Q} - \tilde{Q}\tilde{D}$ and $G := I - \tilde{Q}^T\tilde{Q}$. If $\|G\|_\infty < 1$, then*

$$|\lambda_i - \tilde{\lambda}_i| \leq \delta, \quad i = 1, \ldots, n, \quad \text{where } \delta := \frac{\sqrt{\|R\|_\infty \|R\|_1}}{1 - \|G\|_\infty}.$$

Lemma 2 (Wilkinson [30]) *Let A, $\tilde{q}^{(i)}$, $\tilde{\lambda}_i$, λ_i and R be as in Lemma 1, and $r^{(i)}$ be the i-th column of R. It then holds that*

$$\min_j |\lambda_j - \tilde{\lambda}_i| \le \varepsilon_i, \quad i = 1, \dots, n, \quad \text{where } \varepsilon_i := \frac{\|r^{(i)}\|_2}{\|\tilde{q}^{(i)}\|_2}.$$

Lemma 3 (Miyajima et al. [18]) *Let δ and ε_i be as in Lemmas 1 and 2, respectively. Then, $\varepsilon_i \le \delta$ holds for all i.*

Lemma 4 (Miyajima et al. [18]) *Let λ_i and $\tilde{\lambda}_i$, $i = 1, \dots, n$ be sequences of real numbers such that $\lambda_1 \le \cdots \le \lambda_n$ and $\tilde{\lambda}_1 \le \cdots \le \tilde{\lambda}_n$, respectively. Assume $|\lambda_i - \tilde{\lambda}_i| \le \delta$ for all i, and*

$$\begin{cases} \tilde{\lambda}_{i+1} - \tilde{\lambda}_i > 2\delta & \\ \tilde{\lambda}_i - \tilde{\lambda}_{i-1} > 2\delta \quad \text{and} \quad \tilde{\lambda}_{i+1} - \tilde{\lambda}_i > 2\delta & \\ \tilde{\lambda}_i - \tilde{\lambda}_{i-1} > 2\delta & \end{cases} \begin{matrix} (i = 1) \\ (2 \le i \le n-1) \\ (i = n) \end{matrix}$$

for some i. Then, $\min_j |\lambda_j - \tilde{\lambda}_i| = |\lambda_i - \tilde{\lambda}_i|$ for some i.

Lemma 5 (Miyajima et al. [18]) *Let λ_i, $\tilde{\lambda}_i$ and δ be as in Lemma 4. Assume $\min_j |\lambda_j - \tilde{\lambda}_i| \le \varepsilon_i$ for each i, and some partial sequence $\tilde{\lambda}_{\underline{k}}, \dots, \tilde{\lambda}_{\overline{k}}$ with $1 \le \underline{k} < \overline{k} \le n$ are clustered such that $\tilde{\lambda}_{\underline{k}} - \tilde{\lambda}_{\underline{k}-1} > 2\delta$, $\tilde{\lambda}_{\overline{k}+1} - \tilde{\lambda}_{\overline{k}} > 2\delta$ and $\tilde{\lambda}_{k+1} - \tilde{\lambda}_k \le 2\delta$ for all $k = \underline{k}, \dots, \overline{k} - 1$. If $\varepsilon_k + \varepsilon_{k+1} < \tilde{\lambda}_{k+1} - \tilde{\lambda}_k$ for all $k = \underline{k}, \dots, \overline{k} - 1$, then $\min_j |\lambda_j - \tilde{\lambda}_k| = |\lambda_k - \tilde{\lambda}_k|$ for all $k = \underline{k}, \dots, \overline{k}$.*

Lemma 6 is a modification of [19, Theorem 6] suited for enclosing eigenvectors of a real symmetric matrix having unit 2-norm.

Lemma 6 *Let A, $\tilde{q}^{(i)}$, $\tilde{\lambda}_i$ and λ_i be as in Lemma 1, ε_i be as in Lemma 2, $q^{(i)} \in \mathbb{R}^n \backslash \{0\}$ with $\|q^{(i)}\|_2 = 1$ be an eigenvector corresponding to λ_i, $\rho_i \in \mathbb{R}$ fulfills $0 < \rho_i \le \min_{j \ne i} |\lambda_j - \tilde{\lambda}_i|$, and $\xi_i := \varepsilon_i / \rho_i$. If $\xi_i \le 1$, then*

$$\left\| q^{(i)} - \frac{1}{\|\tilde{q}^{(i)}\|_2} \tilde{q}^{(i)} \right\|_2 \le \omega_i, \quad i = 1, \dots, n, \quad \text{where } \omega_i := \sqrt{2}\sqrt{1 - \sqrt{1 - \xi_i^2}}.$$

Proof Since A is real symmetric, we can take $\{q^{(1)}, \dots, q^{(n)}\}$ as an orthonormal basis. Then, there exist $d_1, \dots, d_n \in \mathbb{R}$ such that $(1/\|\tilde{q}^{(i)}\|_2)\tilde{q}^{(i)} = \sum_j d_j q^{(j)}$. By updating $q^{(i)} = -q^{(i)}$ if necessary, we can assume $d_i \ge 0$ without loss of generality. From $\|(1/\|\tilde{q}^{(i)}\|_2)\tilde{q}^{(i)}\|_2^2 = \|\sum_j d_j q^{(j)}\|_2^2$, we have $1 = \sum_j d_j^2$, so that $(1 + d_i)(1 - d_i) = \sum_{j \ne i} d_j^2$. From $d_i \ge 0$, we obtain $1 + d_i > 0$, which gives

$$1 - d_i = \frac{\sum_{j \ne i} d_j^2}{1 + d_i} \ge 0. \tag{3}$$

Since $q^{(1)}, \dots, q^{(n)}$ are the orthonormal eigenvectors, we moreover have

$$\begin{aligned} \left\| A\left(\frac{1}{\|\tilde{q}^{(i)}\|_2}\tilde{q}^{(i)}\right) - \tilde{\lambda}_i\left(\frac{1}{\|\tilde{q}^{(i)}\|_2}\tilde{q}^{(i)}\right) \right\|_2^2 &= \left\| A\left(\sum_j d_j q^{(j)}\right) - \tilde{\lambda}_i\left(\sum_j d_j q^{(j)}\right) \right\|_2^2 \\ &= \sum_j d_j^2 (\lambda_j - \tilde{\lambda}_i)^2 \ge \sum_{j \ne i} d_j^2 (\lambda_j - \tilde{\lambda}_i)^2 \ge \min_{k \ne i} (\lambda_k - \tilde{\lambda}_i)^2 \sum_{j \ne i} d_j^2. \end{aligned}$$

From this and $0 < \rho_i \le \min_{j\neq i} |\lambda_j - \tilde{\lambda}_i|$, we obtain $\varepsilon_i^2 \ge \rho_i^2 \sum_{j\neq i} d_j^2$, so that $\sum_{j\neq i} d_j^2 \le \xi_i^2$, which shows $d_i^2 = 1 - \sum_{j\neq i} d_j^2 \ge 1 - \xi_i^2$. Thus, $d_i \ge 0$ and $\xi_i \le 1$ yield $d_i \ge \sqrt{1-\xi_i^2}$, so that (3) gives $(1-d_i)^2 \le (1-\sqrt{1-\xi_i^2})^2$. This and $\sum_{j\neq i} d_j^2 \le \xi_i^2$ finally show

$$\begin{aligned}\left\| q^{(i)} - \frac{1}{\|\tilde{q}^{(i)}\|_2}\tilde{q}^{(i)} \right\|_2^2 &= \left\| (1-d_i)q^{(i)} - \sum_{j\neq i} d_j q^{(j)} \right\|_2^2 = (1-d_i)^2 + \sum_{j\neq i} d_j^2 \\ &\le \left(1-\sqrt{1-\xi_i^2}\right)^2 + \xi_i^2 = 2\left(1-\sqrt{1-\xi_i^2}\right) = \omega_i^2,\end{aligned}$$

which completes the proof. □

3 An Interval Enclosing the Residual

As mentioned in Section 1, solving Problem 1 is equivalent to solving $f(c) = 0$, where $f(c)$ is defined in (1). A standard approach for computing an interval vector containing a solution to nonlinear systems is the Krawczyk method, which is based on the following theorem:

Theorem 1 (Krawczyk [13]) *Assume $f : D \subset \mathbb{R}^n \to \mathbb{R}^n$ is continuously differentiable in D. Let $\tilde{c} \in D$ and $\boldsymbol{z} \in \mathbb{IR}^n$ satisfy $\tilde{c} + \boldsymbol{z} \subseteq D$, let $J(c)$ be the Jacobian matrix of f at the point $c \in D$, and let $S \in \mathbb{R}^{n\times n}$. Suppose $\boldsymbol{J} \in \mathbb{IR}^{n\times n}$ satisfies $\boldsymbol{J} \supseteq \{J(c) : c \in \tilde{c} + \boldsymbol{z}\}$ and define $\boldsymbol{k}(\tilde{c}, \boldsymbol{z}) := -Sf(\tilde{c}) + (I - S\boldsymbol{J})\boldsymbol{z}$. If $\boldsymbol{k}(\tilde{c}, \boldsymbol{z}) \subseteq \mathrm{int}(\boldsymbol{z})$, then f has a zero in $\tilde{c} + \boldsymbol{k}(\tilde{c}, \boldsymbol{z}) \subseteq \tilde{c} + \boldsymbol{z}$, and this zero is unique in $\tilde{c} + \boldsymbol{z}$.*

In practical applications, $\tilde{c}$ is a numerical solution to $f(c) = 0$, S is an approximate inverse of the midpoint of $\boldsymbol{J}$, and an interval vector enclosing $f(\tilde{c})$ is required for computing $\boldsymbol{k}(\tilde{c}, \boldsymbol{z})$. As mentioned in Section 1, computing $\boldsymbol{J}$ and the interval vector is not trivial, since $f(c)$ and $J(c)$ in Problem 1 are not *explicitly* represented as the function of c. In this section, we develop a new method for computing the interval vector enclosing $f(\tilde{c})$. A new method for computing $\boldsymbol{J}$ will be proposed in Section 4.

Let $\tilde{c}$ be as the above. Assume as a result of numerical spectral decomposition of $A(\tilde{c})$, we have $\tilde{D}, \tilde{Q} \in \mathbb{R}^{n\times n}$ with $\tilde{D} = \mathrm{diag}(\tilde{\lambda}_1, \ldots, \tilde{\lambda}_n)$ such that $A(\tilde{c})\tilde{Q} \approx \tilde{Q}\tilde{D}$ and $\tilde{Q}$ is approximately orthogonal. We consider computing $\eta_i \in \mathbb{R}$ satisfying $\lambda_i(\tilde{c}) \in \langle \tilde{\lambda}_i, \eta_i \rangle$, i.e., $|\lambda_i(\tilde{c}) - \tilde{\lambda}_i| \le \eta_i$, $i = 1, \ldots, n$. If η_i can be obtained, we can enclose $f_i(\tilde{c})$ such that $f_i(\tilde{c}) \in \langle \tilde{\lambda}_i, \eta_i \rangle - \lambda_i^*$.

Let $R := A(\tilde{c})\tilde{Q} - \tilde{Q}\tilde{D}$, $G := I - \tilde{Q}^T\tilde{Q}$, and $\tilde{q}^{(i)}$ and $r^{(i)}$ be the i-th columns of $\tilde{Q}$ and R, respectively, for $i = 1, \ldots, n$. We can then expect $R \approx 0$, $G \approx 0$ and $r^{(i)} \approx 0$. Suppose $\|G\|_\infty < 1$ and $\tilde{q}^{(i)} \neq 0$, and define $\delta := \sqrt{\|R\|_1 \|R\|_\infty}/(1 - \|G\|_\infty)$ and $\varepsilon_i := \|r^{(i)}\|_2 / \|\tilde{q}^{(i)}\|_2$. Then, Lemma 1 gives $|\lambda_i(\tilde{c}) - \tilde{\lambda}_i| \le \delta$ for all i. From Lemma 2, moreover, $\min_j |\lambda_j(\tilde{c}) - \tilde{\lambda}_i| \le \varepsilon_i$. Taking Lemma 3 into account, we compute the above η_i such that

$$\eta_i = \begin{cases} \varepsilon_i & (\text{If } \min_j |\lambda_j(\tilde{c}) - \tilde{\lambda}_i| = |\lambda_i(\tilde{c}) - \tilde{\lambda}_i| \text{ is proved}) \\ \delta & (\text{otherwise}) \end{cases}.$$

To verify $\min_j |\lambda_j(\tilde{c}) - \tilde{\lambda}_i| = |\lambda_i(\tilde{c}) - \tilde{\lambda}_i|$, we can apply Lemmas 4 and 5 to $\lambda_i := \lambda_i(\tilde{c})$.

4 An Interval Containing the Jacobian Matrix

As mentioned in Section 3, we need to compute $\boldsymbol{J} \in \mathbb{IR}^{n\times n}$ such that $\boldsymbol{J} \supseteq \{J(c) : c \in \tilde{c} + \boldsymbol{z}\}$ for $J(c)$ in (2). We formulate and prove Theorem 2 for this purpose.

Theorem 2 *Let $\tilde{c}$, $\tilde{\lambda}_i$, $\tilde{q}^{(i)}$ and $r^{(i)}$ be as in Section 3, $\boldsymbol{z} \in \mathbb{IR}^n$ be given, $\boldsymbol{B}(\boldsymbol{z}) := \sum_{i=1}^n \boldsymbol{z}_i A_i$, $\boldsymbol{r}^{(i)}(\boldsymbol{z}) := r^{(i)} + \boldsymbol{B}(\boldsymbol{z})\tilde{q}^{(i)}$, $\eta_i(\boldsymbol{z})$ satisfy $|\lambda_i(c) - \tilde{\lambda}_i| \le \eta_i(\boldsymbol{z})$, $\forall c \in \tilde{c} + \boldsymbol{z}$,*

$$\rho_i(\boldsymbol{z}) := \begin{cases} \tilde{\lambda}_2 - \tilde{\lambda}_1 - \eta_2(\boldsymbol{z}) & (i=1) \\ \min\left(\tilde{\lambda}_i - \tilde{\lambda}_{i-1} - \eta_{i-1}(\boldsymbol{z}), \tilde{\lambda}_{i+1} - \tilde{\lambda}_i - \eta_{i+1}(\boldsymbol{z})\right) & (2 \le i \le n-1) \\ \tilde{\lambda}_n - \tilde{\lambda}_{n-1} - \eta_{n-1}(\boldsymbol{z}) & (i=n) \end{cases}, \quad (4)$$

and $\varepsilon_i(\boldsymbol{z}) := \|\boldsymbol{r}^{(i)}(\boldsymbol{z})\|_2 / \|\tilde{q}^{(i)}\|_2$ for $i = 1, \ldots, n$. Assume $\rho_i(\boldsymbol{z}) > 0$ and define $\xi_i(\boldsymbol{z}) := \varepsilon_i(\boldsymbol{z})/\rho_i(\boldsymbol{z})$. Suppose $\xi_i(\boldsymbol{z}) \le 1$ and let $\omega_i(\boldsymbol{z}) := \sqrt{2}\sqrt{1 - \sqrt{1-\xi_i(\boldsymbol{z})^2}}$. Then, for $J(c)$ in (2), $\{J(c)_{ij} : c \in \tilde{c} + \boldsymbol{z}\} \subseteq \boldsymbol{J}_{ij}$, where

$$\boldsymbol{J}_{ij} := \left\langle \frac{\tilde{q}^{(i)^T} A_j \tilde{q}^{(i)}}{\tilde{q}^{(i)^T}\tilde{q}^{(i)}}, \frac{2\omega_i(\boldsymbol{z}) e^T |A_j \tilde{q}^{(i)}|}{\|\tilde{q}^{(i)}\|_2} + \omega_i(\boldsymbol{z})^2 e^T |A_j| e \right\rangle, \quad i,j = 1, \ldots, n.$$

Proof Let c be an arbitrary n-vector included in $\tilde{c} + \boldsymbol{z}$ and $s^{(i)}(c) := q^{(i)}(c) - (1/\|\tilde{q}^{(i)}\|_2)\tilde{q}^{(i)}$. Since $A_j^T = A_j$, we have

$$\begin{aligned} q^{(i)}(c)^T A_j q^{(i)}(c) &= \left(\frac{1}{\|\tilde{q}^{(i)}\|_2}\tilde{q}^{(i)} + s^{(i)}(c)\right)^T A_j \left(\frac{1}{\|\tilde{q}^{(i)}\|_2}\tilde{q}^{(i)} + s^{(i)}(c)\right) \\ &= \frac{\tilde{q}^{(i)^T} A_j \tilde{q}^{(i)}}{\tilde{q}^{(i)^T}\tilde{q}^{(i)}} + \frac{2 s^{(i)}(c)^T A_j \tilde{q}^{(i)}}{\|\tilde{q}^{(i)}\|_2} + s^{(i)}(c)^T A_j s^{(i)}(c) \\ &\in \left\langle \frac{\tilde{q}^{(i)^T} A_j \tilde{q}^{(i)}}{\tilde{q}^{(i)^T}\tilde{q}^{(i)}}, \frac{2|s^{(i)}(c)|^T |A_j \tilde{q}^{(i)}|}{\|\tilde{q}^{(i)}\|_2} + |s^{(i)}(c)|^T |A_j| |s^{(i)}(c)| \right\rangle \\ &\subseteq \left\langle \frac{\tilde{q}^{(i)^T} A_j \tilde{q}^{(i)}}{\tilde{q}^{(i)^T}\tilde{q}^{(i)}}, \frac{2\|s^{(i)}(c)\|_2 e^T |A_j \tilde{q}^{(i)}|}{\|\tilde{q}^{(i)}\|_2} + \|s^{(i)}(c)\|_2^2 e^T |A_j| e \right\rangle. \end{aligned}$$

From this and (2), it suffices to show $\|s^{(i)}(c)\|_2 \le \omega_i(\boldsymbol{z})$, $\forall c \in \tilde{c} + \boldsymbol{z}$. From $|\lambda_i(c) - \tilde{\lambda}_i| \le \eta_i(\boldsymbol{z})$, $\forall c \in \tilde{c} + \boldsymbol{z}$ and (4), we have $\rho_i(\boldsymbol{z}) \le \min_{j\ne i} |\lambda_j(c) - \tilde{\lambda}_i|$, $\forall c \in \tilde{c} + \boldsymbol{z}$ (see Figure 1).

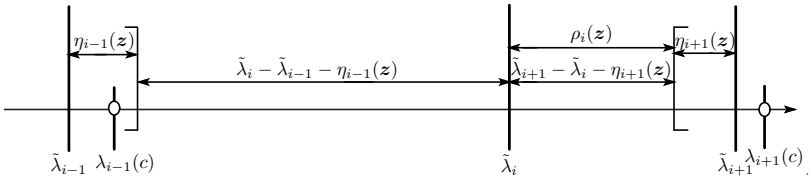

Figure 1: The explanation for $\rho_i(\boldsymbol{z}) \le \min_{j\ne i} |\lambda_j(c) - \tilde{\lambda}_i|$, $\forall c \in \tilde{c} + \boldsymbol{z}$.

Any c can be written as $c = \tilde{c} + z$, where $z \in \boldsymbol{z}$, so that $A(c) = A(\tilde{c}) + A(z) - A_0$ and

$$\|A(c)\tilde{q}^{(i)} - \tilde{\lambda}_i \tilde{q}^{(i)}\|_2 = \|r^{(i)} + (A(z) - A_0)\tilde{q}^{(i)}\|_2 \le \|r^{(i)} + \boldsymbol{B}(\boldsymbol{z})\tilde{q}^{(i)}\|_2 = \|\boldsymbol{r}^{(i)}(\boldsymbol{z})\|_2,$$

which shows

$$\frac{\|A(c)\tilde{q}^{(i)} - \tilde{\lambda}_i \tilde{q}^{(i)}\|_2}{\|\tilde{q}^{(i)}\|_2} \le \varepsilon_i(\boldsymbol{z}), \quad \forall c \in \tilde{c} + \boldsymbol{z}. \quad (5)$$

Thus, $0 < \rho_i(\boldsymbol{z}) \le \min_{j\neq i} |\lambda_j(c) - \tilde{\lambda}_i|$, $\forall c \in \tilde{c} + \boldsymbol{z}$, $\xi_i(\boldsymbol{z}) \le 1$ and Lemma 6 gives $\|s^{(i)}(c)\|_2 \le \omega_i(\boldsymbol{z})$, $\forall c \in \tilde{c} + \boldsymbol{z}$. $\square$

In practical applications, we need to numerically compute $\eta_i(\boldsymbol{z})$ in Theorem 2. Let $\tilde{D}$, $\tilde{Q}$, R and G be as in Section 3, $\boldsymbol{R}(\boldsymbol{z}) := R + \boldsymbol{B}(\boldsymbol{z})\tilde{Q}$ and

$$\delta(\boldsymbol{z}) := \sqrt{\|\boldsymbol{R}(\boldsymbol{z})\|_\infty \|\boldsymbol{R}(\boldsymbol{z})\|_1}/(1 - \|G\|_\infty).$$

We then have

$$A(c)\tilde{Q} - \tilde{Q}\tilde{D} = R + (A(z) - A_0)\tilde{Q} \in R + \boldsymbol{B}(\boldsymbol{z})\tilde{Q} = \boldsymbol{R}(\boldsymbol{z}), \quad \forall c \in \tilde{c} + \boldsymbol{z},$$

so that $\|A(c)\tilde{Q} - \tilde{Q}\tilde{D}\|_\infty \le \|\boldsymbol{R}(\boldsymbol{z})\|_\infty$ and $\|A(c)\tilde{Q} - \tilde{Q}\tilde{D}\|_1 \le \|\boldsymbol{R}(\boldsymbol{z})\|_1$ hold for all $c \in \tilde{c} + \boldsymbol{z}$. This and Lemma 1 imply $|\lambda_i(c) - \tilde{\lambda}_i| \le \delta_i(\boldsymbol{z})$, $\forall c \in \tilde{c} + \boldsymbol{z}$. From (5) and Lemma 2, moreover, $\min_j |\lambda_j(c) - \tilde{\lambda}_i| \le \varepsilon_i(\boldsymbol{z})$, $\forall c \in \tilde{c} + \boldsymbol{z}$. Taking these inequalities and Lemma 3 into account, $|\lambda_i(c) - \tilde{\lambda}_i| \le \eta_i(\boldsymbol{z})$, $\forall c \in \tilde{c} + \boldsymbol{z}$ holds by determining $\eta_i(\boldsymbol{z})$ such that

$$\eta_i(\boldsymbol{z}) = \begin{cases} \varepsilon_i(\boldsymbol{z}) & (\text{If } \min_j |\lambda_j(c) - \tilde{\lambda}_i| = |\lambda_i(c) - \tilde{\lambda}_i|, \forall c \in \tilde{c} + \boldsymbol{z} \text{ is proved}) \\ \delta(\boldsymbol{z}) & (\text{otherwise}) \end{cases}.$$

To verify $\min_j |\lambda_j(c) - \tilde{\lambda}_i| = |\lambda_i(c) - \tilde{\lambda}_i|, \forall c \in \tilde{c} + \boldsymbol{z}$, we can again apply Lemmas 4 and 5 to $\lambda_i := \lambda_i(c)$, $\delta := \delta(\boldsymbol{z})$ and $\varepsilon_i := \varepsilon_i(\boldsymbol{z})$.

5 Proposed Algorithm

We first consider the representation of $\boldsymbol{J}$ in Theorem 2, with a view towards its efficient computation. Let $J_{\mathrm{mid}} := (\tilde{q}_i^T A_j \tilde{q}_i/(\tilde{q}_i^T \tilde{q}_i))$, $v := (e^T|A_1|e, \ldots, e^T|A_n|e)^T$,

$$U := 2\mathrm{diag}(\|\tilde{q}^{(1)}\|_2, \ldots, \|\tilde{q}^{(n)}\|_2)^{-1} \begin{pmatrix} e^T|A_1\tilde{q}^{(1)}| & \cdots & e^T|A_n\tilde{q}^{(1)}| \\ \vdots & \ddots & \vdots \\ e^T|A_1\tilde{q}^{(n)}| & \cdots & e^T|A_n\tilde{q}^{(n)}| \end{pmatrix},$$

and $J_{\mathrm{rad}} := \mathrm{diag}(\omega_1(\boldsymbol{z}), \ldots, \omega_n(\boldsymbol{z}))U + (\omega_1(\boldsymbol{z})^2, \ldots, \omega_n(\boldsymbol{z})^2)^T v^T$. We then have

$$\begin{pmatrix} \dfrac{2\omega_1(\boldsymbol{z})e^T|A_1\tilde{q}^{(1)}|}{\|\tilde{q}^{(1)}\|_2} & \cdots & \dfrac{2\omega_1(\boldsymbol{z})e^T|A_n\tilde{q}^{(1)}|}{\|\tilde{q}^{(1)}\|_2} \\ \vdots & \ddots & \vdots \\ \dfrac{2\omega_n(\boldsymbol{z})e^T|A_1\tilde{q}^{(n)}|}{\|\tilde{q}^{(n)}\|_2} & \cdots & \dfrac{2\omega_n(\boldsymbol{z})e^T|A_n\tilde{q}^{(n)}|}{\|\tilde{q}^{(n)}\|_2} \end{pmatrix} = \mathrm{diag}(\omega_1(\boldsymbol{z}), \ldots, \omega_n(\boldsymbol{z}))U,$$

$$\begin{pmatrix} \omega_1(\boldsymbol{z})^2 e^T|A_1|e & \cdots & \omega_1(\boldsymbol{z})^2 e^T|A_n|e \\ \vdots & \ddots & \vdots \\ \omega_n(\boldsymbol{z})^2 e^T|A_1|e & \cdots & \omega_n(\boldsymbol{z})^2 e^T|A_n|e \end{pmatrix} = \begin{pmatrix} \omega_1(\boldsymbol{z})^2 \\ \vdots \\ \omega_n(\boldsymbol{z})^2 \end{pmatrix} v^T,$$

so that $\boldsymbol{J}$ can be written as $\boldsymbol{J} = \langle J_{\mathrm{mid}}, J_{\mathrm{rad}} \rangle$. Based on this representation and Sections 3 and 4, we propose Algorithm 1 for enclosing c^*.

Algorithm 1 *Let $k_{\max}$ and ϱ be a positive integer and real number, respectively. This algorithm computes $\boldsymbol{c} \in \mathbb{IR}^n$ such that $\boldsymbol{c} \ni c^*$. The uniqueness of the solution contained in $\boldsymbol{c}$ is also verified.*

Step 1: *Compute $\tilde{c}$ by numerically solving Problem 1 via a known algorithm.*

Step 2: *Compute an interval matrix enclosing $A(\tilde{c})$. Calculate $\tilde{D}$ and $\tilde{Q}$ by performing a numerical spectral decomposition of the midpoint of the interval matrix.*

Step 3: *Compute an interval matrix enclosing G (during this process, intervals enclosing $\tilde{q}^{(i)^T}\tilde{q}^{(i)}$, $i = 1, \ldots, n$ are also obtained). Calculate $\overline{\mathrm{fl}}(\|G\|_\infty)$ using this interval matrix. If $\overline{\mathrm{fl}}(\|G\|_\infty) \geq 1$, terminate with failure.*

Step 4: *Compute an interval matrix enclosing R reusing the interval matrix in Step 2. Calculate $\overline{\mathrm{fl}}(\delta)$ and $\overline{\mathrm{fl}}(\varepsilon_i)$ using this interval matrix and $\overline{\mathrm{fl}}(\|G\|_\infty)$. Determine η_i by applying Lemmas 4 and 5 to $\lambda_i := \lambda_i(\tilde{c})$, $\delta := \overline{\mathrm{fl}}(\delta)$ and $\varepsilon_i := \overline{\mathrm{fl}}(\varepsilon_i)$.*

Step 5: *Compute interval matrices enclosing $A_i\tilde{Q}$, $i = 1, \ldots, n$.*

Step 6: *Compute $\overline{\mathrm{fl}}(U)$, $\overline{\mathrm{fl}}(v)$ and $\boldsymbol{J}_{\mathrm{mid}} \in \mathbb{IR}^{n\times n}$ such that $\boldsymbol{J}_{\mathrm{mid}} \ni J_{\mathrm{mid}}$ using the intervals containing $\tilde{q}^{(i)^T}\tilde{q}^{(i)}$ and interval matrices in Step 5. Calculate S such that $S = \mathrm{fl}(\mathrm{mid}\,(\boldsymbol{J}_{\mathrm{mid}})^{-1})$.*

Step 7: *Compute $\boldsymbol{g} \in \mathbb{IR}^n$ such that $\boldsymbol{g} \ni -Sf(\tilde{c})$ using η_i, and initialize $\boldsymbol{z}$ and k such that $\boldsymbol{z} = \langle 0, \varrho\rangle e$ and $k = 1$, respectively.*

Step 8: *If $k = k_{\max}$, terminate with failure. Otherwise, compute $\overline{\mathrm{fl}}(\delta(\boldsymbol{z}))$ and $\overline{\mathrm{fl}}(\varepsilon_i(\boldsymbol{z}))$ reusing $\overline{\mathrm{fl}}(\|G\|_\infty)$ and the interval matrix in Step 4, determine $\eta_i(\boldsymbol{z})$ by applying Lemmas 4 and 5 to $\lambda_i = \lambda_i(c)$, $\delta := \overline{\mathrm{fl}}(\delta(\boldsymbol{z}))$ and $\varepsilon_i := \overline{\mathrm{fl}}(\varepsilon_i(\boldsymbol{z}))$, and calculate $\underline{\mathrm{fl}}(\rho_i(\boldsymbol{z}))$.*

Step 9: *If $\min_i \underline{\mathrm{fl}}(\rho_i(\boldsymbol{z})) \leq 0$, terminate with failure. Otherwise, compute $\overline{\mathrm{fl}}(\xi_i(\boldsymbol{z}))$.*

Step 10: *If $\max_i \overline{\mathrm{fl}}(\xi_i(\boldsymbol{z})) > 1$, terminate with failure. Otherwise, compute $\overline{\mathrm{fl}}(\omega_i(\boldsymbol{z}))$.*

Step 11: *Compute $\overline{\mathrm{fl}}(J_{\mathrm{rad}})$ using $\overline{\mathrm{fl}}(\omega_i(\boldsymbol{z}))$, $\overline{\mathrm{fl}}(v)$ and $\overline{\mathrm{fl}}(U)$. Calculate $\overline{\boldsymbol{J}} := \boldsymbol{J}_{\mathrm{mid}} + \langle 0, \overline{\mathrm{fl}}(J_{\mathrm{rad}})\rangle$. Then $\overline{\boldsymbol{J}} \supseteq \boldsymbol{J}$.*

Step 12: *Compute $\boldsymbol{w} := \boldsymbol{g} + (I - S\overline{\boldsymbol{J}})\boldsymbol{z}$. If $\boldsymbol{w} \subseteq \mathrm{int}(\boldsymbol{z})$, go to Step 13. Otherwise, update $\boldsymbol{z}$ and k such that $\boldsymbol{z} = \langle 1, \mathtt{eps}\rangle\boldsymbol{w} + \langle 0, \mathtt{realmin}\rangle e$ and $k = k+1$, respectively, and go back to Step 8.*

Step 13: *Compute $\boldsymbol{c} = \tilde{c} + \boldsymbol{w}$. Terminate.*

The update of $\boldsymbol{z}$ in Step 12 is based on epsilon inflation [6, 23], which has precise theoretical justification [23]. Computing enclosures of $\boldsymbol{B}(\boldsymbol{z})$, $\boldsymbol{B}(\boldsymbol{z})\tilde{Q}$ and $S\overline{\boldsymbol{J}}$ involves $\mathcal{O}(n^3)$ operations. The other computations in Steps 8 to 12 are possible with $\mathcal{O}(n^2)$ operations by reusing the matrices obtained in Steps 1 to 7. Hence, Steps 8 to 12 require only $\mathcal{O}(n^3)$ operations, i.e., the iterative process in Algorithm 1 involves only cubic complexity per iteration. On the other hand, Step 5 involves $\mathcal{O}(n^4)$ operations. If the number of iterations is $\mathcal{O}(n)$, therefore, Algorithm 1 requires $\mathcal{O}(n^4)$ operations.

6 Numerical Results

We used a computer with Intel Core 1.2GHz CPU, 16GB RAM, MATLAB R2012a with Intel MKL and IEEE 754 double precision. Throughout this section, let `lam` $:= (\lambda_1^*, \ldots, \lambda_n^*)^T$, and $\alpha \in \mathbb{R}$ satisfy $0 < \alpha \leq 1$. We set `lam` such that `lam` $= (1, \mathrm{fl}(1 + \alpha), 3, \ldots, n)^T$. Observe that λ_1^* and λ_2^* become closely clustered when α is small.

We denote the compared algorithms as follows:

`AM`: [2, Algorithm 3.8],

`AGM`: The algorithm in [1, Section 4], and

`M`: Algorithm 1.

In `AM`, we used `intgauss.m` in [12, Section A] to execute the interval Gaussian algorithm, and we skipped the improvement step based on the discussion in [2, Section 4]. In `AM` and `AGM`, we adopted [2, Algorithm 3.7] to compute intervals containing all eigenvalues and unit 2-norm eigenvectors of interval matrices. In Step 1 of `M`, we executed the Newton method discussed in [1, 2] with the stopping criterion (18) in [1, Section 4]. We set $k_{\max}$ and ϱ in `M` as $k_{\max} = 50$ and $\varrho = \eta_k$, respectively, where η_k is defined as in (13) in [1, Section 4] and satisfies (18) in the paper. See `http://web.cc.iwate-u.ac.jp/~miyajima/ISEP.zip` for details of the implementations, where the INTLAB [24] codes of the compared algorithms (denoted by `AM.m`, `AGM.m` and `M.m`) are uploaded.

Let `A0` $:= A_0$, `A` be a 3-dimensional array storing $A_1, \ldots, A_n$, and `ex_sol` $:= (1, \ldots, n)^T$. In the examples below, we obtained `A0` from `A` and `ex_sol` by the following code for making c^* approximately satisfy $c^* \approx$ `ex_sol`:

```
setround(0);
Ac = zeros(n);  for i = 1:n, Ac = Ac + ex_sol(i)*A(:,:,i);  end;
[Q,D] = eig(Ac); d = diag(D); A0 = Q*diag(lam - d)*Q'; A0 = (A0 + A0')/2;
```

Let $\boldsymbol{c} \in \mathbb{IR}^n$ contain c^* in Problem 1. To assess the qualities of the enclosure, define the maximum radius (MR) as $\max_i \mathrm{rad}\,(\boldsymbol{c})_i$. In some examples, the compared algorithms failed. The reasons for the failure of `AM` and `M` are that `intgauss.m` caused error, and that $\min_i \underline{\mathrm{fl}}(\rho_i(\boldsymbol{z})) \leq 0$ in Step 9 occurred, respectively.

Example 1

Consider the case where A_i, $i = 1, \ldots, n$ are Toeplitz matrices such that

$$A_1 = I, \quad A_2 = \begin{pmatrix} 0 & 1 & 0 & \ldots & 0 \\ 1 & 0 & 1 & \ddots & \vdots \\ 0 & 1 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 & 1 \\ 0 & \ldots & 0 & 1 & 0 \end{pmatrix}, \quad \ldots, \quad A_n = \begin{pmatrix} 0 & 0 & \ldots & 0 & 1 \\ 0 & \ddots & \ddots & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ldots & \ddots & \ddots & 0 \\ 1 & 0 & \ldots & 0 & 0 \end{pmatrix}.$$

The matrix set $\{A_i\}_{i=1}^n$ has been used extensively; see for instance [3, 7, 26, 27]. We set the initial guess `c0` in the Newton method by `setround(0); c0 = ex_sol + 0.01* min(abs(ex_sol))*ones(n,1);`. Table 1 displays the MR and CPU times (sec) of the algorithms for various α and n.

We see from Table 1 that the MR by `M` was larger and smaller than that by `AGM` when n was small and large, respectively. The algorithm `M` was faster than the other algorithms. This result coincides with the fact that the iterative process in `M` requires only $\mathcal{O}(n^3)$ operations per iteration, whereas those in the other algorithms involve $\mathcal{O}(n^4)$ operations per iteration.

Table 1: The MR (left part) and CPU times (right part) in Example 1.

α	n	AM	AGM	M	AM	AGM	M
1	10	2.1e–12	2.1e–14	4.1e–14	5.0e–1	2.0e–1	1.6e–2
1	20	5.0e–9	1.0e–13	2.1e–13	1.2e+0	4.1e–1	2.6e–2
1	30	failed	3.1e–13	6.2e–13	failed	7.1e–1	4.4e–2
1	310	failed	1.0e–8	2.8e–10	failed	3.9e+1	8.4e+0
1	320	failed	1.2e–8	3.4e–10	failed	4.1e+1	8.8e+0
1	330	failed	failed	4.2e–10	failed	failed	9.9e+0
2^{-9}	10	3.2e–12	2.1e–14	3.9e–14	6.2e–1	2.3e–1	1.8e–2
2^{-9}	20	5.7e–9	1.4e–13	2.9e–13	2.0e+0	4.5e–1	2.8e–2
2^{-9}	30	failed	4.7e–13	1.1e–12	failed	8.0e–1	4.5e–2
2^{-9}	310	failed	8.6e–9	9.5e–10	failed	3.8e+1	7.9e+0
2^{-9}	320	failed	1.0e–8	1.3e–9	failed	4.1e+1	8.6e+0
2^{-9}	330	failed	failed	1.8e–9	failed	failed	9.7e+0
2^{-18}	10	2.7e–12	9.8e–14	1.7e–13	9.2e–1	2.4e–1	1.9e–2
2^{-18}	20	failed	8.6e–12	1.8e–11	failed	4.6e–1	4.2e–2
2^{-18}	30	failed	7.0e–11	1.8e–10	failed	8.2e–1	5.1e–2
2^{-18}	310	failed	failed	failed	failed	failed	failed
2^{-18}	320	failed	failed	failed	failed	failed	failed
2^{-18}	330	failed	failed	failed	failed	failed	failed

Example 2

Consider the case where $A_1 = (1/m_1)e^{(1)}e^{(1)^T}$,

$$A_i = \left(\frac{1}{\sqrt{m_1}}e^{(1)} - \frac{1}{\sqrt{m_i}}e^{(i)}\right)\left(\frac{1}{\sqrt{m_1}}e^{(1)} - \frac{1}{\sqrt{m_i}}e^{(i)}\right)^T, \quad i = 2, \ldots, n,$$

$e^{(i)}$ is the i-th column of I, $m_1 = 2$, and $m_2 = \cdots = m_n = 0.2$. The set $\{A_i\}_{i=1}^n$ is an extension of that in [26, Example 4.2]. We computed A_i via floating point operations in rounding to nearest mode, and set `c0` by `setround(0); c0 = ex_sol + 0.001* min(abs(ex_sol))*ones(n,1);`. Table 2 shows the quantities similar to Table 1.

It can be seen from Table 2 that in some examples, `M` succeeded whereas the other algorithms failed. This result shows the robustness of `M`.

7 Application to Inverse Singular Value Problems

Let $A(c)$, A_i, $\sigma_i(c)$, σ_i^* for $i = 1, \ldots, n$ and c^* be as in Problem 2, let $\tilde{c}$ be a numerical solution to Problem 2, let $\boldsymbol{z} \in \mathbb{IR}^n$ be given, and let $A(c) = U(c)\Sigma(c)V(c)^T$ be a singular value decomposition of $A(c)$, where $\Sigma(c) = \mathrm{diag}(\sigma_1(c), \ldots, \sigma_n(c)) \in \mathbb{R}^{m\times n}$, and $U(c) = (u^{(1)}(c), \ldots, u^{(m)}(c)) \in \mathbb{R}^{m\times m}$ and $V(c) = (v^{(1)}(c), \ldots, v^{(n)}(c)) \in \mathbb{R}^{n\times n}$ are orthogonal. Assume as a result of numerical singular value decomposition of $A(\tilde{c})$, we have $\tilde{U} = (\tilde{u}^{(1)}, \ldots, \tilde{u}^{(m)}) \in \mathbb{R}^{m\times m}$, $\tilde{\Sigma} = \mathrm{diag}(\tilde{\sigma}_1, \ldots, \tilde{\sigma}_n) \in \mathbb{R}^{m\times n}$ and

Table 2: The MR (left part) and CPU times (right part) in Example 2.

α	n	AM	AGM	M	AM	AGM	M
1	10	4.6e–14	4.5e–14	6.3e–14	4.7e+0	2.0e–1	1.7e–2
1	20	2.5e–9	1.4e–10	1.9e–10	2.9e+0	6.1e–1	6.1e–2
1	21	6.3e–9	3.5e–10	4.8e–10	8.8e+0	6.7e–1	7.4e–2
2^{-18}	10	4.6e–14	1.1e–13	7.7e–14	2.2e+0	2.3e–1	1.9e–2
2^{-18}	20	2.9e–9	failed	2.0e–10	6.0e+0	failed	8.8e–2
2^{-18}	21	failed	failed	4.9e–10	failed	failed	1.4e–1
2^{-21}	10	4.6e–14	5.4e–13	2.7e–13	1.5e+0	2.1e–1	1.9e–2
2^{-21}	20	failed	failed	2.3e–10	failed	failed	2.6e–1
2^{-21}	21	failed	failed	failed	failed	failed	failed

$\tilde{V} = (\tilde{v}^{(1)}, \ldots, \tilde{v}^{(n)}) \in \mathbb{R}^{n\times n}$ such that $A(\tilde{c}) \approx \tilde{U}\tilde{\Sigma}\tilde{V}^T$, and $\tilde{U}$ and $\tilde{V}$ are approximately orthogonal.

Solving Problem 2 is equivalent to solving $f(c) = 0$, where

$$f(c) := (\sigma_1(c) - \sigma_1^*, \ldots, \sigma_n(c) - \sigma_n^*)^T.$$

The function f is differentiable around c^*, and the singular vectors corresponding to $\{\sigma_i(c)\}_{i=1}^n$ are continuous around c^* (cf. [5]). The Jacobian matrix $J(c)$ of $f(c)$ is given such that (see [5] for example)

$$J(c)_{ij} = \frac{\partial \sigma_i(c)}{\partial c_j} = u^{(i)}(c)^T A_j v^{(i)}(c), \quad i, j = 1, \ldots, n.$$

From these and Sections 3 and 4, the verified computation for c^* is possible if $\sigma_i(\tilde{c})$ can be enclosed, and upper bounds for $\|u^{(i)}(c) - (1/\|\tilde{u}^{(i)}\|_2)\tilde{u}^{(i)}\|_2$ and $\|v^{(i)}(c) - (1/\|\tilde{v}^{(i)}\|_2)\tilde{v}^{(i)}\|_2$, $\forall c \in \tilde{c} + \boldsymbol{z}$ can be computed for $i = 1, \ldots, n$. Note that upper bounds for $\|u^{(j)}(c) - (1/\|\tilde{u}^{(j)}\|_2)\tilde{u}^{(j)}\|_2$, $\forall c \in \tilde{c} + \boldsymbol{z}$, $j = n+1, \ldots, m$ are not required.

Let $U_1(c) := (u^{(1)}(c), \ldots, u^{(n)}(c))$, $U_2(c) := (u^{(n+1)}(c), \ldots, u^{(m)}(c))$, $\tilde{U}_1 := (\tilde{u}^{(1)}, \ldots, \tilde{u}^{(n)})$, $\tilde{U}_2 := (\tilde{u}^{(n+1)}, \ldots, \tilde{u}^{(m)})$,

$$\begin{aligned}
\mathcal{A}(c) &:= \begin{pmatrix} 0 & A(c)^T \\ A(c) & 0 \end{pmatrix}, \quad \mathcal{Q}(c) := \frac{1}{\sqrt{2}} \begin{pmatrix} V(c) & V(c) & 0 \\ U_1(c) & -U_1(c) & \sqrt{2}U_2(c) \end{pmatrix}, \\
\mathcal{D}(c) &:= \mathrm{diag}(\sigma_1(c), \ldots, \sigma_n(c), -\sigma_1(c), \ldots, -\sigma_n(c), \underbrace{0, \ldots, 0}_{m-n}), \\
\tilde{\mathcal{Q}} &:= \frac{1}{\sqrt{2}} \begin{pmatrix} \tilde{V} & \tilde{V} & 0 \\ \tilde{U}_1 & -\tilde{U}_1 & \sqrt{2}\tilde{U}_2 \end{pmatrix}, \\
\tilde{\mathcal{D}} &:= \mathrm{diag}(\tilde{\sigma}_1, \ldots, \tilde{\sigma}_n, -\tilde{\sigma}_1, \ldots, -\tilde{\sigma}_n, \underbrace{0, \ldots, 0}_{m-n}),
\end{aligned}$$

and let $q^{(i)}(c)$ and $\tilde{q}^{(i)}$ be the i-th columns of $\mathcal{Q}(c)$ and $\tilde{\mathcal{Q}}$, respectively, for $i = 1, \ldots, m+n$. We then have $\mathcal{A}(c)\mathcal{Q}(c) = \mathcal{Q}(c)\mathcal{D}(c)$ (see [11]), i.e., the singular values and vectors of $A(c)$ can be obtained by considering eigenvalues and eigenvectors of $\mathcal{A}(c)$, respectively. Therefore, intervals containing $\sigma_i(\tilde{c})$, $i = 1, \ldots, n$ can be obtained

by enclosing the largest n eigenvalues of $\mathcal{A}(\tilde{c})$ based on Section 3 using $\tilde{\mathcal{Q}}$ and $\tilde{\mathcal{D}}$. Moreover, the upper bounds for $\|u^{(i)}(c) - (1/\|\tilde{u}^{(i)}\|_2)\tilde{u}^{(i)}\|_2$ and $\|v^{(i)}(c) - (1/\|\tilde{v}^{(i)}\|_2)\tilde{v}^{(i)}\|_2$, $\forall c \in \tilde{c} + \boldsymbol{z}$, $i = 1, \ldots, n$ can be computed by calculating an upper bound for $\|q^{(i)}(c) - (1/\|\tilde{q}^{(i)}\|_2)\tilde{q}^{(i)}\|_2$, $\forall c \in \tilde{c} + \boldsymbol{z}$, $i = 1, \ldots, n$ based on Section 4.

8 Conclusion

In this paper, we have proposed an algorithm for computing an interval containing the solution to Problem 1, we have reported the numerical results, and have sketched the algorithm for enclosing the solution to Problem 2. If A_i, $i = 1, \ldots, n$ have the special structure which is taken into account to reduce the computational cost, then the cost of Algorithm 1 may become $\mathcal{O}(n^3)$ operations. Our future work will be to develop a verification algorithm which is applicable even when $\{\lambda_1^*, \ldots, \lambda_n^*\}$ includes multiple eigenvalues.

Acknowledgments

The author acknowledges the reviewers for valuable comments. This work was partially supported by JSPS KAKENHI Grant Number JP16K05270.

References

[1] G. Alefeld, A. Gienger, and G. Mayer. Numerical validation for an inverse matrix eigenvalue problem. *Computing*, 53:311–322, 1994.

[2] G. Alefeld and G. Mayer. A computer aided existence and uniqueness proof for an inverse matrix eigenvalue problem. *Interval Computations*, 1:4–27, 1994.

[3] Z.J. Bai, R.H. Chan, and B. Morini. An inexact Cayley transform method for inverse eigenvalue problems. *Inverse Probl.*, 20:1675–1689, 2004.

[4] Z.J. Bai, X.Q. Jin, and S.W. Vong. On some inverse singular value problems with Toeplitz-related structure. *Numer. Algebra Control Optim.*, 2:187–192, 2012.

[5] Z.J. Bai and S.F. Xu. An inexact Newton-type method for inverse singular value problems. *Linear Algebra Appl.*, 429:527–547, 2008.

[6] O. Caprani and K. Madsen. Iterative methods for interval inclusion of fixed points. *BIT*, 18:42–51, 1978.

[7] R.H. Chan, H.L. Chung, and S.F. Xu. The inexact Newton-like method for inverse eigenvalue problem. *BIT*, 43:7–20, 2003.

[8] M.T. Chu and G.H. Golub. Structured inverse eigenvalue problems. *Acta Numer.*, 11:1–71, 2002.

[9] M.T. Chu and G.H. Golub. *Inverse Eigenvalue Problems*. Oxford University Press, New York, 2005.

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

[11] G.H. Golub and C.F. Van Loan. *Matrix Computations, third ed.* The Johns Hopkins University Press, Baltimore and London, 1996.

[12] G.I. Hargreaves. Interval analysis in Matlab. *MIMS EPrint*, 2009.1:1–49, 2002. `http://eprints.ma.man.ac.uk/1204/`.

[13] R. Krawczyk. Newton-algorithmen zur bestimmung von nullstellen mit fehlerschranken. *Computing*, 4:187–201, 1969.

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

[15] S. Miyajima. Fast enclosure for all eigenvalues in generalized eigenvalue problem. *J. Comp. Appl. Math.*, 233:2994–3004, 2010.

[16] S. Miyajima. Numerical enclosure for each eigenvalue in generalized eigenvalue problem. *J. Comp. Appl. Math.*, 236:2545–2552, 2012.

[17] S. Miyajima. Fast enclosure for all eigenvalues and invariant subspaces in generalized eigenvalue problems. *SIAM J. Matrix Anal. Appl.*, 35:1205–1225, 2014.

[18] S. Miyajima, T. Ogita, and S. Oishi. Fast verification for respective eigenvalues of symmetric matrix. *Lecture Notes Comput. Sci.*, 3718:306–317, 2005.

[19] S. Miyajima, T. Ogita, and S. Oishi. Numerical verification for each eigenpair of symmetric matrix. *Trans. JSIAM*, 16:535–552, 2006 (in Japanese).

[20] S. Miyajima, T. Ogita, S.M. Rump, and S. Oishi. Fast verification for all eigenpairs in symmetric positive definite generalized eigenvalue problems. *Reliab. Comput.*, 14:24–45, 2010.

[21] J.M. Ortega. *Numerical Analysis.* Academic Press, New York, 1972.

[22] T. Politi. A discrete approach for the inverse singular value problem in some quadratic group. *Lecture Notes Comput. Sci.*, 2658:121–130, 2003.

[23] S.M. Rump. Verification methods for dense and sparse systems of equations. In *Topics in Validated Computations - Studies in Computational Mathematics (J. Herzberger ed.)*, pages 63–136. Elsevier, Amsterdam, 1994.

[24] S.M. Rump. INTLAB - INTerval LABoratory. In *Developments in Reliable Computing (T. Csendes ed.)*, pages 77–104. Kluwer Academic Publishers, Dordrecht, 1999.

[25] C.S. Saunders, J. Hu, C.E. Christoffersen, and M.B. Steer. Inverse singular value method for enforcing passivity in reduced-order models of distributed structures for transient and steady-state simulation. *IEEE Trans. Microw. Theory Tech.*, 59:837–847, 2011.

[26] W. Shen. A generalized inexact Newton method for inverse eigenvalue problems. *Abstr. Appl. Anal.*, 2014:Article ID 721346, 6 pages, 2014.

[27] W.P. Shen, C. Li, and X.Q. Jin. A Ulm-like method for inverse eigenvalue problems. *Appl. Numer. Math.*, 61:356–367, 2011.

[28] K. Toyonaga. Numerical enclosure for multiple eigenvalues of an Hermitian matrix whose graph is a tree. *Linear Algebra Appl.*, 431:1989–1999, 2009.

[29] J.A. Tropp, I.S. Dhillon, and R.W. Heath. Finite-step algorithms for constructing optimal CDMA signature sequences. *IEEE Trans. Inf. Theory*, 50:2916–2921, 2004.

[30] J.H. Wilkinson. Rigorous error bounds for computed eigensystem. *Computer J.*, 4:230–241, 1961.