Fast Verification for All Eigenpairs in Symmetric Positive Definite Generalized Eigenvalue Problems*

Shinya Miyajima
Faculty of Engineering, Gifu University, Gifu 501-1193 Japan
miyajima@gifu-u.ac.jp

Takeshi Ogita
Japan Science and Technology Agency, Faculty of Science and Engineering, Waseda University, Tokyo 169-0072 Japan
ogita@waseda.jp

Siegfried M. Rump
Institute for Reliable Computing, Hamburg University of Technology, Hamburg 21071 Germany
rump@tu-harburg.de

Shin'ichi Oishi
Faculty of Science and Engineering, Waseda University, Tokyo 169-0072 Japan
oishi@waseda.jp

Abstract

A fast method for enclosing all eigenpairs in symmetric positive definite generalized eigenvalue problems is proposed. Firstly theorems on verifying all eigenvalues are presented. Next a theorem on verifying all eigenvectors is presented. The proposed method is developed based on these theorems. Numerical results are presented showing the efficiency of the proposed method. As an application of the proposed method, an efficient method of enclosing all eigenpairs in the quadratic eigenvalue problem is also sketched.

Keywords: numerical verification, eigenpairs, generalized eigenvalue problem
AMS subject classifications: 65G20, 65G50

*Submitted: July 26, 2007; Accepted: December 2,2007.

1 Introduction

In this paper, we are concerned with the accuracy of computed eigenpairs in the generalized eigenvalue problem

$$Ax = \lambda Bx, \quad A, B \in \mathbb{R}^{n\times n},\ \lambda \in \mathbb{R},\ x \in \mathbb{R}^n \tag{1}$$

where A is symmetric and B is symmetric positive definite. Here an eigenpair (λ, x) denotes a pair of an eigenvalue λ and its corresponding eigenvector x. The problem (1) arises in many applications of scientific computations, e.g. stationary analysis of circuits, image processing, structure analysis and so forth.

Since B is symmetric positive definite, there exists a Cholesky factorization of B such that

$$B = LL^T, \quad L \in \mathbb{R}^{n\times n} \tag{2}$$

where L is nonsingular lower triangular. Substituting (2) into (1), we obtain

$$L^{-1}AL^{-T}L^Tx = \lambda L^Tx. \tag{3}$$

Therefore (1) is equivalent to the standard eigenvalue problem (3) with eigenpairs $(\lambda,\ L^Tx)$, where $L^{-1}AL^{-T}$ is a real symmetric matrix. In the Cholesky-QR method (e.g. [4, p. 463]), the QR algorithm is applied to (3) and consequently all eigenpairs in (1) are computed. In the MATLAB function `eig`, this method is adopted for computing all eigenpairs in (1).

There are several methods for calculating guaranteed error bounds for approximate eigenvalues and eigenvectors, e.g. [1, 2, 6, 16, 17, 19, 24, 26]. On enclosing *a few specified eigenvalues*, see [1, 2, 24, 26]. On *a few specified eigenpairs*, see [16, 17]. On *all eigenvalues*, see [6]. In [19] methods are presented for computing inclusions of multiple eigenvalues and a basis for a corresponding invariant subspace. Moreover in [20] it is shown how to compute an inclusion of an individual eigenvector to a multiple eigenvalue of geometric multiplicity one. Excellent overviews on perturbation theory for matrix eigenvalues can be found in [3, 15, 23].

In this paper, we propose a fast method of enclosing *all eigenpairs* for the generalized eigenvalue problem, which is the expansion of the verification method for the standard eigenvalue problem [10]. The proposed method supplies error bounds for each approximate eigenpair in (1). In [20] it is also shown that we cannot expect to be able to compute an inclusion in floating-point of an individual eigenvector to a multiple eigenvalue which is not of geometric multiplicity one. Since this is also shown for normal, so especially for Hermitian or symmetric matrices, it limits the following considerations to matrices with only simple eigenvalues. Moreover some techniques for accelerating the proposed method are suggested. The proposed method, where these techniques are used, allows the presence of underflow in floating-point arithmetic. This paper also includes some numerical examples to show the performance and properties of the proposed method.

As an application of the proposed method, we also sketch an efficient method of enclosing all eigenpairs (λ, x) in the quadratic eigenvalue problem

$$(\lambda^2A + \lambda B + C)x = 0, \quad A, B, C \in \mathbb{R}^{n\times n},\ \lambda \in \mathbb{R},\ x \in \mathbb{R}^n$$

where A is symmetric negative definite, B is symmetric and C is symmetric positive definite. This problem arises in, e.g. the dynamic analysis of rotating structures.

2 Utilized Theorems

In this section, we introduce some theorems which are used in Section 3.

2.1 Theorems on Verifying All Eigenpairs in the Standard Eigenvalue Problem

In this section, we refer the theorems on enclosing all eigenpairs in the standard eigenvalue problem: For $i = 1, \dots, n$

$$Ax^{(i)} = \lambda_i x^{(i)}, \quad A \in \mathbb{R}^{n\times n},\ \lambda_i \in \mathbb{R},\ x^{(i)} \in \mathbb{R}^n \tag{4}$$

where A is symmetric, λ_i is an eigenvalue of A and $x^{(i)}$ is an eigenvector corresponding to λ_i.

In this section, we assume that approximate eigenvalues $\tilde{\lambda}_i$ and eigenvectors $\tilde{x}^{(i)}$ for all i in (4) are given. Let $\tilde{D}$ and $\tilde{X}$ be an $n \times n$ diagonal matrix and an $n \times n$ approximately orthogonal matrix defined as

$$\tilde{D} := \mathrm{diag}(\tilde{\lambda}_1, \dots, \tilde{\lambda}_n) \quad \text{and} \quad \tilde{X} := (\tilde{x}^{(1)}, \dots, \tilde{x}^{(n)}),$$

respectively, so that $A\tilde{X} \approx \tilde{X}\tilde{D}$ and $I \approx \tilde{X}^T\tilde{X}$ where I denotes the $n \times n$ identity matrix. Additionally we define $n \times n$ residual matrices R_s and G_s as

$$R_s := A\tilde{X} - \tilde{X}\tilde{D} \quad \text{and} \quad G_s := I - \tilde{X}^T\tilde{X}. \tag{5}$$

First we cite Theorems 1 and 2 on verifying $\tilde{\lambda}_i$.

Theorem 1 (Rump) *Let A be a real symmetric $n \times n$ matrix. Let λ_i and $\tilde{\lambda}_i$ for $i = 1, \dots, n$ be the true eigenvalues in (4) and their approximations such that*

$$\lambda_1 \le \cdots \le \lambda_n \quad \textit{and} \quad \tilde{\lambda}_1 \le \cdots \le \tilde{\lambda}_n,$$

respectively. Let R_s and G_s be defined as in (5). If $\|G_s\|_2 < 1$, it holds for all i that

$$|\lambda_i - \tilde{\lambda}_i| \le \frac{\|R_s\|_2}{1 - \|G_s\|_2}. \tag{6}$$

The proof of Theorem 1 is due to the third author.

Proof The result follows from the more general theorem by Cao, Xie, and Li [3], see for example Theorem 11.10.1 in [15], which states

$$|\lambda_i - \tilde{\lambda}_i| \le \frac{\|R_s\|_2}{\sigma_1(\tilde{X})}, \tag{7}$$

where $\sigma_1(\tilde{X}), \dots, \sigma_n(\tilde{X})$ denote the singular values of $\tilde{X}$ such that $\sigma_1(\tilde{X}) \le \cdots \le \sigma_n(\tilde{X})$. To show that (7) implies (6) we have to show

$$\frac{1}{\sigma_1(\tilde{X})} \le \frac{1}{1 - \|G_s\|_2}$$

if $\|G_s\|_2 < 1$, which is equivalent to

$$1 - \sigma_1(\tilde{X}) \le \|G_s\|_2.$$

For $\sigma_1(\tilde{X}) \geq 1$, this is obvious. Otherwise we have

$$\begin{aligned}\|G_s\|_2 &= \rho(G_s) = \max_{1\leq i\leq n} |1 - \lambda_i(\tilde{X}^T\tilde{X})| = \max_{1\leq i\leq n} |1 - \sigma_i(\tilde{X})^2| \\ &\geq 1 - \sigma_1(\tilde{X})^2 \geq 1 - \sigma_1(\tilde{X}),\end{aligned}$$

where $\rho(G_s)$ and $\lambda_1(\tilde{X}^T\tilde{X}), \dots, \lambda_n(\tilde{X}^T\tilde{X})$ denote the spectral radius of G_s and the eigenvalues of $\tilde{X}^T\tilde{X}$, respectively. Thus the result follows.

□

The advantage of (6) is that G_s is symmetric, so $\|G_s\|_2 = \rho(G_s)$ is easily estimated by Perron-Frobenius Theory. Therefore, the error bound (6) is easily and effectively computable.

Theorem 2 (Wilkinson [25]) *Let A, $\tilde{\lambda}_i$ and λ_j be defined as in Theorem 1. Let $\tilde{x}^{(i)}$ be an approximate eigenvector in (4) corresponding to $\tilde{\lambda}_i$. Then it holds that*

$$\min_{1\leq j\leq n} |\lambda_j - \tilde{\lambda}_i| \leq \varepsilon_i, \quad \varepsilon_i := \frac{\|r^{(i)}\|_2}{\|\tilde{x}^{(i)}\|_2}, \tag{8}$$

where

$$r^{(i)} := A\tilde{x}^{(i)} - \tilde{\lambda}_i\tilde{x}^{(i)}. \tag{9}$$

Next we present Theorem 3 on verifying $\tilde{x}^{(i)}$.

Theorem 3 (Miyajima et al. [10]) *Let λ_i and $\tilde{\lambda}_i$ for some i be defined as in Theorem 1. Assume that $|\lambda_i - \tilde{\lambda}_i| \leq \eta_i$ for each i. Let $r^{(i)}$ be defined as in (9). Moreover let ρ_i and ξ_i be defined as follows:*

$$\rho_i := \begin{cases} \tilde{\lambda}_2 - \tilde{\lambda}_1 - \eta_2 & (i = 1) \\ \min(\tilde{\lambda}_i - \tilde{\lambda}_{i-1} - \eta_{i-1}, \tilde{\lambda}_{i+1} - \tilde{\lambda}_i - \eta_{i+1}) & (2 \leq i \leq n-1) \\ \tilde{\lambda}_n - \tilde{\lambda}_{n-1} - \eta_{n-1} & (i = n) \end{cases} \tag{10}$$

$$\xi_i := \frac{\|r^{(i)}\|_2}{\rho_i}. \tag{11}$$

If $\rho_i > 0$ and $\xi_i < \|\tilde{x}^{(i)}\|_2$ hold, then there exists an eigenvector $\hat{x}^{(i)}$ corresponding to λ_i such that

$$\left\|\hat{x}^{(i)} - \tilde{x}^{(i)}\right\|_2 \leq \xi_i. \tag{12}$$

2.2 Miscellaneous Lemmas

In this section, we cite miscellaneous lemmas.

First we introduce Lemma 1 about a matrix norm.

Lemma 1 (Miyajima et al. [9]) *Let $\tilde{Q}$ be a real $n \times n$ matrix. Let I and e_i be the $n \times n$ identity matrix and the i-th column of I for $i = 1, \dots, n$, respectively. Then it holds that*

$$1 - \|I - \tilde{Q}^T\tilde{Q}\|_\infty \leq \|\tilde{Q}e_i\|_2. \tag{13}$$

Let λ_i and $\tilde{\lambda}_i$ be defined as in Theorem 1. Let λ_j be the true eigenvalue nearest to $\tilde{\lambda}_i$. Theorem 2 supplies an upper bound of distance between $\tilde{\lambda}_i$ and λ_j. Hence it does not necessarily follow that $\min_{1\le j\le n}|\lambda_j - \tilde{\lambda}_i| = |\lambda_i - \tilde{\lambda}_i|$ i.e. $j = i$. In the verification for all eigenvalues, it is required that $j = i$ holds strictly to utilize the upper bound of $\min_{1\le j\le n}|\lambda_j - \tilde{\lambda}_i|$ as the error bound for $\tilde{\lambda}_i$. Therefore we cite Lemmas 2 and 3 with respect to checking whether $j = i$ holds.

Lemma 2 (Miyajima et al. [9]) *Let λ_i and $\tilde{\lambda}_i$ for $i = 1, \dots, n$ be sequences of real numbers such that*

$$\lambda_1 \le \cdots \le \lambda_n \quad \textit{and} \quad \tilde{\lambda}_1 \le \cdots \le \tilde{\lambda}_n,$$

respectively. Assume that $|\lambda_i - \tilde{\lambda}_i| \le \delta$ for all i. Suppose

$$\begin{cases} \tilde{\lambda}_{i+1} - \tilde{\lambda}_i > 2\delta & (i = 1) \\ \tilde{\lambda}_i - \tilde{\lambda}_{i-1} > 2\delta \quad \wedge \quad \tilde{\lambda}_{i+1} - \tilde{\lambda}_i > 2\delta & (2 \le i \le n-1) \\ \tilde{\lambda}_i - \tilde{\lambda}_{i-1} > 2\delta & (i = n) \end{cases} \tag{14}$$

holds for some i. Then

$$\min_{1\le j\le n}|\lambda_j - \tilde{\lambda}_i| = |\lambda_i - \tilde{\lambda}_i| \quad \textit{for some } i.$$

Figure 1 illustrates the case that $\tilde{\lambda}_i - \tilde{\lambda}_{i-1} > 2\delta \ \wedge \ \tilde{\lambda}_{i+1} - \tilde{\lambda}_i > 2\delta$ holds.

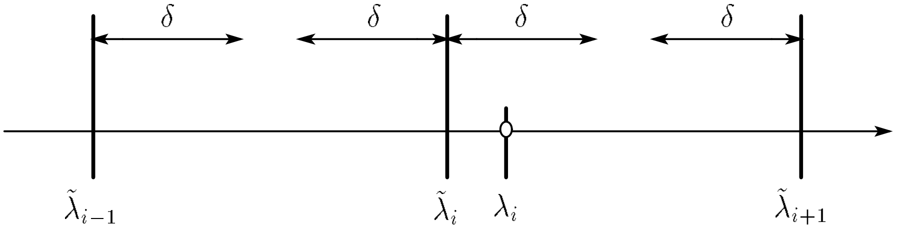

Figure 1: The case that $\tilde{\lambda}_i - \tilde{\lambda}_{i-1} > 2\delta \ \wedge \ \tilde{\lambda}_{i+1} - \tilde{\lambda}_i > 2\delta$ holds.

Lemma 3 (Miyajima et al. [9]) *Let λ_i, $\tilde{\lambda}_i$ for $i = 1, \dots, n$ and δ be defined as in Lemma 2. Assume that $\min_{1\le j\le n}|\lambda_j - \tilde{\lambda}_i| \le \varepsilon_i$ for each i. Suppose that 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*

$$\tilde{\lambda}_{\underline{k}} - \tilde{\lambda}_{\underline{k}-1} > 2\delta \ \wedge \ \tilde{\lambda}_{\overline{k}+1} - \tilde{\lambda}_{\overline{k}} > 2\delta \ \wedge \ \tilde{\lambda}_{k+1} - \tilde{\lambda}_k \le 2\delta \tag{15}$$

for all $k = \underline{k}, \dots, \overline{k} - 1$. If it holds for all $k = \underline{k}, \dots, \overline{k} - 1$ that

$$\varepsilon_k + \varepsilon_{k+1} < \tilde{\lambda}_{k+1} - \tilde{\lambda}_k, \tag{16}$$

then

$$\min_{1\le j\le n}|\lambda_j - \tilde{\lambda}_k| = |\lambda_k - \tilde{\lambda}_k| \quad \textit{for all } k = \underline{k}, \dots, \overline{k}.$$

Figure 2 illustrates the case that (15) and (16) hold.

Remark 1 *Note that Lemmas 2 and 3 hold for any sequences λ_i and $\tilde{\lambda}_i$.*

In [9], Lemmas 2 and 3 are applied for the standard eigenvalue problem (4). We stress that these lemmas are also applicable for the generalized eigenvalue problem (1). In this paper, we utilize these lemmas for (1).

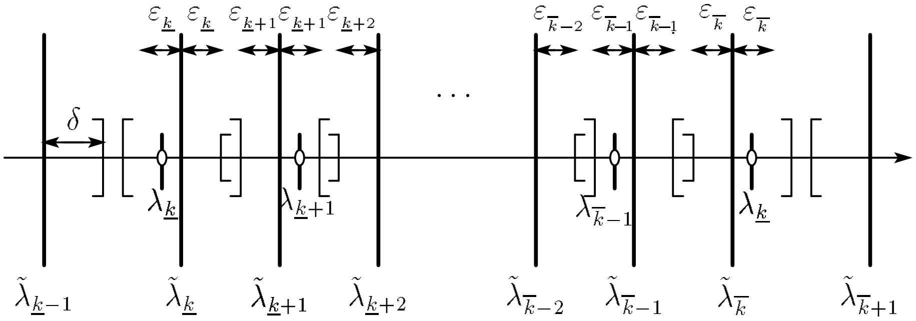
$\varepsilon_{\underline{k}}$
$\varepsilon_{\underline{k}}$
$\varepsilon_{\underline{k}+1}$
$\varepsilon_{\underline{k}+1}$
$\varepsilon_{\underline{k}+2}$
$\varepsilon_{\overline{k}-2}$
$\varepsilon_{\overline{k}-1}$
$\varepsilon_{\overline{k}-1}$
$\varepsilon_{\overline{k}}$
$\varepsilon_{\overline{k}}$
δ
$\cdots$
$\lambda_{\underline{k}}$
$\lambda_{\underline{k}+1}$
$\lambda_{\overline{k}-1}$
$\lambda_{\overline{k}}$
$\tilde\lambda_{\underline{k}-1}$
$\tilde\lambda_{\underline{k}}$
$\tilde\lambda_{\underline{k}+1}$
$\tilde\lambda_{\underline{k}+2}$
$\tilde\lambda_{\overline{k}-2}$
$\tilde\lambda_{\overline{k}-1}$
$\tilde\lambda_{\overline{k}}$
$\tilde\lambda_{\overline{k}+1}$

Figure 2: The case that (15) and (16) hold.

3 Proposed Method for Verifying All Eigenpairs in the Generalized Eigenvalue problem

In this section, we propose a fast method of enclosing all eigenpairs in the generalized eigenvalue problem: For $i = 1, \dots, n$

$$Ax^{(i)} = \lambda_i Bx^{(i)}, \quad A, B \in \mathbb{R}^{n\times n},\ \lambda_i \in \mathbb{R},\ x^{(i)} \in \mathbb{R}^n \tag{17}$$

where A is symmetric, B is symmetric positive definite, λ_i is an eigenvalue and $x^{(i)}$ is an eigenvector corresponding to λ_i.

In this section, we assume that approximate eigenvalues $\tilde\lambda_i$ and eigenvectors $\tilde x^{(i)}$ for all i in (17) are given. Let $\tilde D$ and $\tilde X$ be defined similar to those in (4) for $\tilde\lambda_i$ and $\tilde x^{(i)}$ in (17), respectively, so that $A\tilde X \approx B\tilde X\tilde D$ and $\tilde X^T B\tilde X \approx I$ where I denotes the $n \times n$ identity matrix. Additionally we define $n \times n$ matrices R_g and G_g as follows:

$$\begin{aligned} R_g &:= A\tilde X - B\tilde X\tilde D && (18)\\ G_g &:= \tilde X^T B\tilde X. && (19)\end{aligned}$$

3.1 Theorems on Verifying All Eigenvalues

In this section, we present theorems on verifying $\tilde\lambda_i$ for all i in (17).

At first we present Theorems 4 and 5 on verifying $\tilde\lambda_i$.

Theorem 4 *Let A and B be real symmetric $n \times n$ matrices. Let λ_i and $\tilde\lambda_i$ for $i = 1, \dots, n$ be the true eigenvalues in (17) and their approximations such that*

$$\lambda_1 \le \cdots \le \lambda_n \quad \textit{and} \quad \tilde\lambda_1 \le \cdots \le \tilde\lambda_n,$$

respectively. Let R_g and G_g be defined as in (18) and (19), respectively. If $\|I - G_g\|_2 < 1$ holds, then B is positive definite and it holds for all i that

$$|\lambda_i - \tilde\lambda_i| \le \hat\delta, \quad \hat\delta := \frac{\beta\|R_g\|_2}{1 - \|I - G_g\|_2},$$

where

$$\beta \ge \sqrt{\|B^{-1}\|_2}. \tag{20}$$

Proof Let P and Q be real $n \times n$ matrices. It is well known (e.g. [4, 13]) that P is nonsingular if $\|I - QP\|_{\hat{p}} < 1,\ 1 \leq \hat{p} \leq \infty$. From this and $\|I - G_g\|_2 < 1$, $\tilde{X}$ is nonsingular. So it can be shown (e.g. [4, Theorem 8.1.17]) that B and G_g have the same inertia. Therefore if G_g is positive definite, then B is also positive definite. Accordingly we will prove that G_g is positive definite.

Since G_g is symmetric, the eigen decomposition of G_g can be written as $G_g = VTV^T$ in which V is orthogonal and $T = \mathrm{diag}(\lambda_1(G_g), \dots, \lambda_n(G_g))$ where $\lambda_1(G_g), \dots, \lambda_n(G_g)$ denote the eigenvalues of G_g such that $\lambda_1(G_g) \leq \cdots \leq \lambda_n(G_g)$. Then it follows that

$$\begin{aligned} \|I - G_g\|_2 &= \|I - VTV^T\|_2 = \|V(I-T)V^T\|_2 = \|I - T\|_2 \\ &= \max_{1 \leq i \leq n} |1 - \lambda_i(G_g)| \geq |1 - \lambda_1(G_g)| \geq 1 - \lambda_1(G_g). \end{aligned}$$

Therefore if $\|I - G_g\|_2 < 1$, then $1 - \lambda_1(G_g) < 1$, which implies that G_g is positive definite. Thus B is also positive definite.

Since B is positive definite, there exists the Cholesky factorization (2). Applying Theorem 1 to (3) yields

$$|\lambda_i - \tilde{\lambda}_i| \leq \frac{\|L^{-1}AL^{-T}L^T\tilde{X} - L^T\tilde{X}\tilde{D}\|_2}{1 - \|I - (L^T\tilde{X})^T(L^T\tilde{X})\|_2}. \tag{21}$$

Utilizing (2) and (18), we have

$$\begin{aligned} \|L^{-1}AL^{-T}L^T\tilde{X} - L^T\tilde{X}\tilde{D}\|_2 &= \|L^{-1}(A\tilde{X} - LL^T\tilde{X}\tilde{D})\|_2 = \|L^{-1}R_g\|_2 \\ &\leq \|L^{-1}\|_2\|R_g\|_2. \end{aligned} \tag{22}$$

It is well known (e.g. [5, p.108]) that $\|C^TC\|_2 = \|C\|_2^2$ holds for any matrix C, so that

$$\|L^{-1}\|_2 = \sqrt{\|B^{-1}\|_2}. \tag{23}$$

From (20), (22) and (23), it holds that

$$\|L^{-1}AL^{-T}L^T\tilde{X} - L^T\tilde{X}\tilde{D}\|_2 \leq \beta\|R_g\|_2. \tag{24}$$

Moreover utilizing (2) and (19), we obtain

$$\begin{aligned} 1 - \|I - (L^T\tilde{X})^T(L^T\tilde{X})\|_2 &= 1 - \|I - \tilde{X}^TLL^T\tilde{X}\|_2 \\ &= 1 - \|I - \tilde{X}^TB\tilde{X}\|_2 \\ &= 1 - \|I - G_g\|_2. \end{aligned} \tag{25}$$

Substituting (24) and (25) into (21) proves the theorem. □

Remark 2 *There are several methods (e.g. [11, 13, 14, 17, 21, 22]) to compute a rigorous upper bound of $\|B^{-1}\|_2$. By applying one of these methods, we can compute β. For example, a fast method using Cholesky factorization of $B - \sigma I$ for appropriate $\sigma \in \mathbb{R}$ can be utilized [17]. On the other hand, in Section 3.5, we present a new faster method of calculating β in the case that a computed Cholesky factor of B and its approximate inverse are given.*

Remark 3 *Comparing to matrix 1-norm and ∞-norm, it is disadvantageous in computational cost to compute matrix 2-norm with guaranteed accuracy. For a square matrix P, it is known that $\|P\|_2 \leq \sqrt{\|P\|_1\|P\|_\infty}$. Moreover, if P is symmetric in particular, then it follows that $\|P\|_2 \leq \|P\|_\infty$. Thus, we obtain*

$$\hat{\delta} \leq \delta, \quad \delta := \frac{\beta\sqrt{\|R_g\|_1\|R_g\|_\infty}}{1 - \|I - G_g\|_\infty}. \tag{26}$$

In the proposed method, δ is computed instead of $\hat{\delta}$ to obtain the error bound of $\tilde{\lambda}_i$ based on Theorem 4.

Theorem 5 *Let $\tilde{\lambda}_i$, λ_j and β be defined as in Theorem 4. Let $\tilde{x}^{(i)}$ be an approximate eigenvector in (17) corresponding to $\tilde{\lambda}_i$. Then it holds that*

$$\min_{1\leq j\leq n}|\lambda_j - \tilde{\lambda}_i| \leq \varepsilon_i, \quad \varepsilon_i := \frac{\beta\|r^{(i)}\|_2}{\sqrt{g_i}} \tag{27}$$

where

$$\begin{aligned} r^{(i)} &:= A\tilde{x}^{(i)} - \tilde{\lambda}_i B\tilde{x}^{(i)} &(28)\\ g_i &:= \tilde{x}^{(i)T}B\tilde{x}^{(i)}. &(29)\end{aligned}$$

Proof Applying Theorem 2 to (3) yields

$$\min_{1\leq j\leq n}|\lambda_j - \tilde{\lambda}_i| \leq \frac{\|L^{-1}AL^{-T}L^T\tilde{x}^{(i)} - \tilde{\lambda}_i L^T\tilde{x}^{(i)}\|_2}{\|L^T\tilde{x}^{(i)}\|_2}. \tag{30}$$

Utilizing (2), (20), (23) and (28), we have

$$\begin{aligned} \|L^{-1}AL^{-T}L^T\tilde{x}^{(i)} - \tilde{\lambda}_i L^T\tilde{x}^{(i)}\|_2 &= \|L^{-1}(A\tilde{x}^{(i)} - \tilde{\lambda}_i LL^T\tilde{x}^{(i)})\|_2 \\ &= \|L^{-1}r^{(i)}\|_2 \\ &\leq \|L^{-1}\|_2\|r^{(i)}\|_2 = \sqrt{\|B^{-1}\|_2}\|r^{(i)}\|_2 \\ &\leq \beta\|r^{(i)}\|_2. &(31)\end{aligned}$$

On the other hand, it holds that $g_i = (L^T\tilde{x}^{(i)})^T(L^T\tilde{x}^{(i)})$. Accordingly

$$\|L^T\tilde{x}^{(i)}\|_2 = \sqrt{g_i}. \tag{32}$$

Substituting (31) and (32) into (30), we obtain the desired result. □

Next we explain how Theorems 4 and 5 are related and used. For this purpose, we present Theorem 6.

Theorem 6 *Let δ and ε_i for $i = 1, \ldots, n$ be defined as in (26) and (27), respectively. Then it holds for all i that*

$$\varepsilon_i \leq \delta. \tag{33}$$

Proof Let R_g and $r^{(i)}$ be defined as in (18) and (28), respectively. Then $r^{(i)}$ is identical to the i-th column of R_g. Therefore we have

$$\|r^{(i)}\|_2 \leq \|R_g\|_2 \leq \sqrt{\|R_g\|_1\|R_g\|_\infty}. \tag{34}$$

On the other hand, let Y be defined as

$$Y := L^T \tilde{X} \tag{35}$$

where L is defined as in (2). Let e_i be defined as in Lemma 1. Then it holds that

$$\begin{aligned} 1 - \|I - G_g\|_\infty &= 1 - \|I - Y^T Y\|_\infty && (36) \\ \sqrt{g_i} &= \|Y e_i\|_2. && (37) \end{aligned}$$

Applying Lemma 1 to the right hand sides of (36) and (37), we obtain

$$1 - \|I - G_g\|_\infty \le \sqrt{g_i}. \tag{38}$$

Combining (34) and (38) proves the theorem. □

Let δ and ε_i for $i = 1, \dots, n$ be defined as in (26) and (27), respectively. Based on Theorem 6, we design the proposed method to supply error bounds η_i which satisfies $|\lambda_i - \tilde{\lambda}_i| \le \eta_i$ such that

$$\eta_i = \begin{cases} \varepsilon_i & (\text{if it is proven that } \min_{1 \le j \le n} |\lambda_j - \tilde{\lambda}_i| = |\lambda_i - \tilde{\lambda}_i|) \\ \delta & (\text{otherwise}) \end{cases} . \tag{39}$$

Therefore it is guaranteed that the proposed method can give the error bounds such that $\eta_i \le \delta$ for all i. To check whether $\min_{1 \le j \le n} |\lambda_j - \tilde{\lambda}_i| = |\lambda_i - \tilde{\lambda}_i|$ holds, we can use Lemmas 2 and 3.

3.2 A Theorem on Verifying All Eigenvectors

In this section, we present Theorem 7 on verifying $\tilde{x}^{(i)}$ for all i in (17).

Theorem 7 *Let λ_i and $\tilde{\lambda}_i$ be defined as in Theorem 4. Assume $|\lambda_i - \tilde{\lambda}_i| \le \eta_i$ for each i. Let $\overline{\rho}_i$ for each i be defined similar to ρ_i in (10). Let β be defined as in Theorem 4. Let $r^{(i)}$ and $g^{(i)}$ be defined as in (28) and (29), respectively. Let $\hat{\xi}_i$ be defined as*

$$\hat{\xi}_i := \frac{\beta \|r^{(i)}\|_2}{\overline{\rho}_i}. \tag{40}$$

If $\overline{\rho}_i > 0$ and $\hat{\xi}_i < \sqrt{g_i}$ hold, then there exists an eigenvector $\hat{x}^{(i)}$ corresponding to λ_i such that

$$\|\hat{x}^{(i)} - \tilde{x}^{(i)}\|_2 \le \xi_i, \quad \xi_i := \beta \hat{\xi}_i. \tag{41}$$

Proof As mentioned in Section 1, the generalized eigenvalue problem (17) is equivalent to the standard eigenvalue problem

$$L^{-1} A L^{-T} L^T x^{(i)} = \lambda_i L^T x^{(i)} \tag{42}$$

where $LL^T = B$. Let ξ_i^* be defined as

$$\xi_i^* := \frac{\|L^{-1} A L^{-T} L^T \tilde{x}^{(i)} - \tilde{\lambda}_i L^T \tilde{x}^{(i)}\|_2}{\overline{\rho}_i}. \tag{43}$$

From (31), (40) and (43) it holds that

$$\xi_i^* \le \hat{\xi}_i. \tag{44}$$

Therefore if $\hat{\xi}_i < \sqrt{g_i}$ holds, then $\xi_i^* \leq \hat{\xi} < \sqrt{g_i}$ holds. Utilizing this and (44), and applying Theorem 3 to (42), if $\overline{\rho}_i > 0$ and $\hat{\xi}_i < \sqrt{g_i}$ hold, then there exists an eigenvector $L^T\hat{x}^{(i)}$ corresponding to λ_i such that

$$\|L^T\hat{x}^{(i)} - L^T\tilde{x}^{(i)}\|_2 \leq \hat{\xi}_i. \tag{45}$$

From (20), (23), (41) and (45) it follows that

$$\begin{aligned}
\|\hat{x}^{(i)} - \tilde{x}^{(i)}\|_2 &= \|L^{-T}L^T(\hat{x}^{(i)} - \tilde{x}^{(i)})\|_2 \\
&\leq \|L^{-T}\|_2\|L^T(\hat{x}^{(i)} - \tilde{x}^{(i)})\|_2 \\
&\leq \|L^{-T}\|_2\hat{\xi}_i = \sqrt{\|B^{-T}\|_2}\hat{\xi}_i \\
&\leq \xi_i, \qquad (46)
\end{aligned}$$

which proves Theorem 7. □

3.3 Concrete Step

Based on Sections 3.1 and 3.2, we present concrete steps of the proposed method in Algorithm 1.

Algorithm 1 *Let β be an upper bound of $\sqrt{\|B^{-1}\|_2}$. Let δ and ε_i be defined as in (26) and (27), respectively. This algorithm computes error bounds $\eta := (\eta_1, \ldots, \eta_n)^T$ and $\xi := (\xi_1, \ldots, \xi_n)^T$ for*

$$|\lambda_i - \tilde{\lambda}_i| \leq \eta_i \quad \text{and} \quad \|x^{(i)} - \tilde{x}^{(i)}\|_2 \leq \xi_i$$

on the assumption that $\tilde{D}$ and $\tilde{X}$ have already been obtained.

Step 1: *Compute β (see Remark 2).*

Step 2: *Compute δ and $\varepsilon := (\varepsilon_1, \ldots, \varepsilon_n)^T$.*

Step 3: *Determine η using δ, ε, Lemmas 2 and 3, and (39).*

Step 4: *Compute ξ.*

By changing rounding modes (e.g. [12, 13]) we can compute η and ξ involving rounding errors. Note that positive definiteness of B is also verified in the process of computing δ at Step 2. Moreover by modifying this algorithm, verifications of a few specified eigenvectors are also possible if Step 3 is completed.

3.4 Techniques on Accelerating the Verification

In this section, we explain the techniques to accelerate Algorithm 1. Let $s := (1, \ldots, 1)^T \in \mathbb{R}^n$. Let $\mathbf{u} \in \mathbb{R}$ and $\underline{\mathbf{u}} \in \mathbb{R}$ be defined as unit roundoff and underflow unit (especially, $\mathbf{u} = 2^{-53}$ and $\underline{\mathbf{u}} = 2^{-1074}$ in IEEE 754 double precision), respectively. Moreover let γ_n be defined as

$$\gamma_n := \frac{n\mathbf{u}}{1 - n\mathbf{u}}. \tag{47}$$

Throughout this paper, fl$(\cdot)$ denotes the result of floating point computations, where all operations inside parentheses are executed by ordinary floating point arithmetic fulfilling rounding mode instruction, especially $\mathrm{fl}_{\square}(\cdot)$ in rounding-to-nearest, $\mathrm{fl}_{\triangle}(\cdot)$ in rounding-upward and $\mathrm{fl}_{\bigtriangledown}(\cdot)$ in rounding-downward.

At Step 2 in Algorithm 1, we can use Techniques 1, 2, and 3. At Step 4, we can use Technique 4.

Technique 1
To compute δ, we need to compute rigorous enclosures of R_g and $I - G_g$, to get a rigorous enclosure of $B\tilde{X}$. Therefore in the proposed method, $\overline{Z} := \mathrm{fl}_{\triangle}(B\tilde{X})$ and $\underline{Z} := \mathrm{fl}_{\bigtriangledown}(B\tilde{X})$ are computed in the process of calculating the enclosure of R_g and reused for calculating the enclosure of $I - G_g$. By these reuses, computational cost for computing δ can be reduced.

Technique 2
Utilizing an a priori error estimation (e.g. [5, 8, 14]), it holds that

$$\begin{aligned} \|I - G_g\|_\infty \quad &\le \quad \|\mathrm{fl}_{\square}(I - \tilde{X}^T Z_c)\|_\infty + \||\tilde{X}^T||Z_r|s\|_\infty \\ &\quad\; + \gamma_n(\||\tilde{X}^T||Z_c|s\|_\infty + 1) + n\underline{\mathbf{u}} \qquad (48) \end{aligned}$$

where

$$Z_c \;:=\; \mathrm{fl}_{\triangle}(\underline{Z} + \frac{1}{2}(\overline{Z} - \underline{Z})) \qquad (49)$$

$$Z_r \;:=\; \mathrm{fl}_{\triangle}(Z_c - \underline{Z}). \qquad (50)$$

Note that (48) holds also in the presence of underflow. From (48) we need to execute matrix multiplication only once in rounding-to-nearest for calculating the rigorous upper bound of $\|I - G_g\|_\infty$, if $\overline{Z}$ and $\underline{Z}$ have already been obtained. Thus the computational cost for $\|\mathrm{fl}_{\square}(I - \tilde{X}^T Z_c)\|_\infty$ is $2n^3$ flops. The computational cost for the other parts in (48) is $\mathcal{O}(n^2)$ flops.

Technique 3
Let $r^{(i)}$ and g_i for $i = 1, \ldots, n$ be defined as in (28) and (29), respectively. To obtain ε_i, we need to compute $r^{(i)}$ and g_i. Here, $r^{(i)}$ and g_i are identical to the i-th column of R_g and the (i, i) element of G_g, respectively. Therefore if R_g, $\overline{Z}$ and $\underline{Z}$ have already been obtained in the process of calculating δ, we can reuse them for calculating ε_i. By these reuses, the computational cost of ε_i for all i becomes $\mathcal{O}(n^2)$ flops.

Technique 4
Let τ_i and μ_i be defined as $\tau_i := \beta\|r^{(i)}\|_2$ and $\mu_i := \sqrt{g_i}$, respectively. To verify $\tilde{x}^{(i)}$, we need to compute τ_i and μ_i. Therefore if τ_i and μ_i have already been obtained in the process of enclosing all eigenvalues, we can reuse them for verifying $\tilde{x}^{(i)}$. By these reuses, the computational cost for verifying $\tilde{x}^{(i)}$ becomes $\mathcal{O}(n)$ flops.

3.5 A Method to obtain β

Let β be defined as in Theorem 4. In this section, we propose a method to compute β, which is applied in Section 4.

Let γ_n and s be defined as in Section 3.4. Let e_i be defined as in Lemma 1. For preliminaries we present Theorems 8 and 9.

Theorem 8 (Oishi and Rump [13]) *Let a nonsingular triangular $n \times n$ matrix L be given. Suppose the columns $X_L e_i$ of an approximate inverse X_L are computed by* *substitution, in any order, of* n *linear systems* $L(X_Le_i) = e_i$*. Then including possible underflow,*

$$|X_LL - I| \le \gamma_n|X_L||L| + \frac{\mathbf{u}}{1 - n\mathbf{u}}(ns + \mathrm{diag}(|L|))s^T.$$

Theorem 9 (e.g. Higham [5]) *If floating point Cholesky factorization applied to a symmetric positive definite matrix* $B \in \mathbb{R}^{n\times n}$ *runs to completion, then the computed Cholesky factor* $\tilde{L}$ *satisfies*

$$\begin{aligned} \tilde{L}\tilde{L}^T &= B + \Delta B \\ |\Delta B| &\le \gamma_n|\tilde{L}||\tilde{L}^T| + \frac{\mathbf{u}}{1-(n-1)\mathbf{u}}((n-1)s + \mathrm{diag}(|L|))s^T \end{aligned} \tag{51}$$

also in the presence of underflow.

Remark 4 *The second term in the right hand side of (51) is devised by the authors. By adding this term Theorem 9 holds also in the presence of underflow.*

Utilizing Theorems 8 and 9, we present Theorem 10.

Theorem 10 *Let* B *and* $\tilde{L}$ *be defined as in Theorem 9. Let* X_L *be an approximate inverse of* $\tilde{L}$ *computed similarly to Theorem 8. Let* ζ_p, α_p *and* α_C *for* $p \in \{1, \infty\}$ *be defined as*

$$\begin{aligned} \zeta_p &:= \gamma_n\||X_L||\tilde{L}|s\|_p + \frac{n\mathbf{u}}{1-n\mathbf{u}}\|ns + \mathrm{diag}(|\tilde{L}|)\|_p, \\ \alpha_p &:= \frac{\|X_L\|_p}{1-\zeta_p}, \\ \alpha_C &:= \gamma_n\||\tilde{L}||\tilde{L}^T|s\|_\infty + \frac{n\mathbf{u}}{1-(n-1)\mathbf{u}}\|(n-1)s + \mathrm{diag}(|\tilde{L}|)\|_\infty. \end{aligned}$$

If $\alpha_1\alpha_\infty\alpha_C < 1$*, it holds that*

$$\|B^{-1}\|_2 \le \frac{\alpha_1\alpha_\infty}{1 - \alpha_1\alpha_\infty\alpha_C}.$$

Proof Let P and Q be real $n \times n$ matrices with P being nonsingular. It is well known (e.g. [4, 13]) that

$$\|P^{-1}\|_{\hat{p}} \le \frac{\|Q\|_{\hat{p}}}{1 - \|QP - I\|_{\hat{p}}}, \quad 1 \le \hat{p} \le \infty \tag{52}$$

if $\|QP - I\|_{\hat{p}} < 1$. Utilizing the symmetry of B^{-1}, substituting $P = B$ and $Q = \tilde{L}^{-T}\tilde{L}^{-1}$ into (52), and putting $\hat{p} = \infty$, we have

$$\|B^{-1}\|_2 \le \|B^{-1}\|_\infty \le \frac{\|\tilde{L}^{-T}\tilde{L}^{-1}\|_\infty}{1 - \|\tilde{L}^{-T}\tilde{L}^{-1}B - I\|_\infty}. \tag{53}$$

Thus if $\|\tilde{L}^{-T}\tilde{L}^{-1}\|_\infty \le \alpha_1\alpha_\infty$ and $\|\tilde{L}^{-T}\tilde{L}^{-1}B - I\|_\infty \le \alpha_1\alpha_\infty\alpha_C$, we obtain Theorem 10. Therefore we will prove them.

Substituting $P = \tilde{L}$ and $Q = X_L$ into (52) yields

$$\|\tilde{L}^{-1}\|_{\hat{p}} \le \frac{\|X_L\|_{\hat{p}}}{1 - \|X_L\tilde{L} - I\|_{\hat{p}}}. \tag{54}$$

Let $\zeta_{\hat{p}}$ and $\alpha_{\hat{p}}$ be defined similarly to ζ_p and α_p, respectively. From Theorem 8, we have

$$\|X_L\tilde{L} - I\|_{\hat{p}} \le \zeta_{\hat{p}}. \tag{55}$$

Substituting (55) into (54) yields

$$\|\tilde{L}^{-1}\|_{\hat{p}} \le \alpha_{\hat{p}}. \tag{56}$$

Therefore it holds that

$$\|\tilde{L}^{-T}\tilde{L}^{-1}\|_\infty \le \|\tilde{L}^{-T}\|_\infty\|\tilde{L}^{-1}\|_\infty = \|\tilde{L}^{-1}\|_1\|\tilde{L}^{-1}\|_\infty \le \alpha_1\alpha_\infty. \tag{57}$$

On the other hand, from Theorem 9 and (56), it follows that

$$\begin{aligned} \|\tilde{L}^{-T}\tilde{L}^{-1}B - I\|_\infty &= \|\tilde{L}^{-T}\tilde{L}^{-1}(B - \tilde{L}\tilde{L}^T)\|_\infty \\ &\le \|\tilde{L}^{-1}\|_1\|\tilde{L}^{-1}\|_\infty\|B - \tilde{L}\tilde{L}^T\|_\infty \\ &\le \alpha_1\alpha_\infty\alpha_C. \end{aligned} \tag{58}$$

Thus (53), (57) and (58) prove the theorem. □

In the next section, we will compute β based on Theorem 10 involving rounding errors. Note that if $\tilde{L}$ and X_L have already been obtained in the process of computing all approximate eigenpairs, then computing β requires only $\mathcal{O}(n^2)$ flops.

4 Numerical Examples

In this section, we report some numerical results to show the property of Algorithm 1 and performance of our implementation. We use a computer with a Pentium IV 3.4GHz CPU and MATLAB 7.0 with ATLAS and IEEE 754 double precision for all computations.

We use the MATLAB function `eig` to obtain all approximate eigenpairs. As mentioned in Section 1, this function adopts the Cholesky-QR method. Accordingly $\tilde{L}$ and X_L in Theorem 10 can also be obtained in the process of computing all approximate eigenpairs. By reusing them, Step 1 in Algorithm 1 requires only $\mathcal{O}(n^2)$ flops. Additionally we use Techniques 1, 2, 3, and 4 described in Section 3.4. Then in Step 2, the computational parts which require $\mathcal{O}(n^3)$ flops are matrix multiplications $\mathrm{fl}_\triangle(A\tilde{X})$, $\mathrm{fl}_\triangledown(A\tilde{X})$, $\mathrm{fl}_\triangle(B\tilde{X})$, $\mathrm{fl}_\triangledown(B\tilde{X})$ and $\mathrm{fl}_\square(\tilde{X}^TZ_c)$. The computational cost of the other parts in Step 2 is $\mathcal{O}(n^2)$ flops. Moreover both Steps 3 and 4 require $\mathcal{O}(n)$ flops. From these, the total computational cost of Algorithm 1 becomes $10n^3$ flops.

Let η_i and ξ_i be defined as in Algorithm 1. Let $\eta_i^{(M)}$ be the error bound of $\tilde{\lambda}_i$ obtained by the method in [6]. Moreover we define $\bar{\eta}_i$, $\bar{\xi}_i$ and $\bar{\eta}_i^{(M)}$ in terms of relative error bound as

$$\bar{\eta}_i := \mathrm{fl}_\triangle\left(\frac{\eta_i}{\mathrm{fl}_\triangledown(|\tilde{\lambda}_i| - \eta_i)}\right), \quad \bar{\xi}_i := \mathrm{fl}_\triangle\left(\frac{\xi_i}{\mathrm{fl}_\triangledown(\|\tilde{x}^{(i)}\|_2 - \xi_i)}\right)$$
$$\text{and} \quad \bar{\eta}_i^{(M)} := \mathrm{fl}_\triangle\left(\frac{\eta_i^{(M)}}{\mathrm{fl}_\triangledown(|\tilde{\lambda}_i| - \eta_i^{(M)})}\right),$$

respectively. If $\bar{\eta}_i$, $\bar{\xi}_i$ and $\bar{\eta}_i^{(M)}$ are nonnegative, it holds that

$$\frac{|\lambda_i - \tilde{\lambda}_i|}{|\lambda_i|} \le \bar{\eta}_i, \quad \frac{\|\hat{x}^{(i)} - \tilde{x}^{(i)}\|_2}{\|\hat{x}^{(i)}\|_2} \le \bar{\xi}_i, \quad \text{and} \quad \frac{|\lambda_i - \tilde{\lambda}_i|}{|\lambda_i|} \le \bar{\eta}_i^{(M)},$$

where $\hat{x}^{(i)}$ is defined as in Theorem 7. Let n_M be the number of nonnegative error bounds in $\bar{\eta}_1^{(M)}, \ldots, \bar{\eta}_n^{(M)}$. For nonnegative real numbers $q_1, \ldots, q_n$, $\operatorname{mean} q$ denotes

$$\operatorname{mean} q := \mathrm{fl}_\square \left(\sum_{i=1}^{n} q_i \Big/ n \right)$$

to see the mean values of the relative error bounds.

Let t_a, t_v and t_M be the computing time (sec) for calculating all approximate eigenpairs, Algorithm 1 and the method in [6], respectively. Note that t_M is the computing time for enclosing *eigenvalues* only and t_v is that for enclosing *eigenpairs*. Moreover define $\kappa(Q) := \|Q\|_2 \|Q^{-1}\|_2$ for a nonsingular matrix Q.

4.1 Example 1

In this example, we observe how the sizes of error bounds change when $\kappa(B)$ increases. Consider the case

$$A := \begin{pmatrix} 5 & -4 & 1 & & & & \\ -4 & 6 & -4 & 1 & & & \\ 1 & -4 & 6 & -4 & 1 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & 1 & -4 & 6 & -4 & 1 \\ & & & 1 & -4 & 6 & -4 \\ & & & & 1 & -4 & 5 \end{pmatrix} \text{ and } B(i,j) := \frac{232792560}{i+j-1}$$

where $A, B \in \mathbb{R}^{n \times n}$. This example was discussed in [2]. Note that the numerator of $B(i,j)$ is chosen so that the matrix B has integer entries for $n \leq 10$. For various n Tables 1 and 2 display $\kappa(B)$, $\max_{1 \leq i \leq n} \bar{\eta}_i$, $\min_{1 \leq i \leq n} \bar{\eta}_i$, $\operatorname{mean} \bar{\eta}$, $\max_{1 \leq i \leq n} \bar{\eta}_i^{(M)}$, $\min_{1 \leq i \leq n} \bar{\eta}_i^{(M)}$, $\operatorname{mean} \bar{\eta}^{(M)}$ and n_M, and $\max_{1 \leq i \leq n} \bar{\xi}_i$, $\min_{1 \leq i \leq n} \bar{\xi}_i$ and $\operatorname{mean} \bar{\xi}$, respectively. In Tables 1, 6 and 9, the notation "–" means that some of the error bounds became negative so that neither $\max \bar{\eta}_i^{(M)}$ nor $\operatorname{mean} \bar{\eta}^{(M)}$ are available. In this case $\min \bar{\eta}_i^{(M)}$ denotes the minimum value among available relative error bounds.

Table 1: $\kappa(B)$, n_M and obtained error bounds for eigenvalues in Example 1

n	$\kappa(B)$	$\max \bar{\eta}_i$	$\min \bar{\eta}_i$	$\operatorname{mean} \bar{\eta}$	$\max \bar{\eta}_i^{(M)}$	$\min \bar{\eta}_i^{(M)}$	$\operatorname{mean} \bar{\eta}^{(M)}$	n_M
5	4.8e+05	1.99e-09	4.58e-12	4.22e-10	6.73e-01	1.83e-11	1.35e-01	5
6	1.5e+07	6.25e-08	2.70e-11	1.11e-08	–	4.71e-10	–	4
7	4.8e+08	1.39e-06	3.53e-10	2.12e-07	–	1.31e-08	–	3
8	1.5e+10	4.72e-05	3.08e-09	6.35e-06	–	4.41e-07	–	3
9	4.9e+11	1.33e-03	4.39e-08	1.58e-04	–	8.03e-06	–	2
10	1.6e+13	3.46e-02	3.18e-07	3.73e-03	–	3.06e-04	–	2

From Table 1 we can confirm that Algorithm 1 supplies smaller error bounds for approximate eigenvalues than those by the method in [6] in this example. Moreover it can be seen that in Algorithm 1, verifications of all eigenvalues succeeded in all cases of this example although in the method in [6], verifications of some eigenvalues failed in some cases. Additionally Tables 1 and 2 show that error bounds increase as $\kappa(B)$ increases.

Table 2: Obtained error bounds for eigenvectors in Example 1

n	$\max \xi_i$	$\min \xi_i$	$\text{mean}\, \xi$
5	3.17e-12	2.07e-13	1.29e-12
6	5.61e-10	6.50e-12	1.90e-10
7	7.29e-08	4.98e-10	2.21e-08
8	1.47e-05	2.46e-08	4.03e-06
9	2.30e-03	1.95e-06	5.32e-04
10	3.46e-01	8.24e-05	7.99e-02

4.2 Example 2

In this example, we observe the sizes of error bounds and computing times for large n when $\kappa(A)$ and $\kappa(B)$ are small. Consider the case that A and B are generated by

```
A=randn(n);
A=(A+A')/2;
B=randn(n);
B=n*eye(n)+(B+B')/2; % eye(n): the n-by-n identity matrix
```

on MATLAB. Here the function `randn` generates a random matrix whose elements are uniformly distributed in $[-1, 1]$. Algorithm 1 verified that B is positive definite. For various n Tables 3 and 4 display the similar quantities as Tables 1 and 2 except for $\kappa(B)$ and n_M, respectively. Table 5 displays t_a, t_v and t_M for various n.

Table 3: Obtained error bounds for eigenvalues in Example 2

n	$\max \bar{\eta}_i$	$\min \bar{\eta}_i$	$\text{mean}\, \bar{\eta}$	$\max \bar{\eta}_i^{(M)}$	$\min \bar{\eta}_i^{(M)}$	$\text{mean}\, \bar{\eta}^{(M)}$
100	1.09e-12	1.62e-14	4.28e-14	1.47e-10	6.24e-13	4.84e-12
250	3.46e-12	3.61e-14	8.62e-14	1.52e-09	3.54e-12	2.79e-11
500	8.69e-12	7.24e-14	1.64e-13	9.43e-09	1.38e-11	1.25e-10
1000	3.12e-11	1.39e-13	3.41e-13	3.96e-11	5.29e-13	4.09e-12
2000	9.71e-11	2.74e-13	6.68e-13	4.56e-07	2.15e-10	2.24e-09
2500	2.44e-10	3.43e-13	8.51e-13	1.47e-06	3.35e-10	3.65e-09

Table 4: Obtained error bounds for eigenvectors in Example 2

n	$\max \xi_i$	$\min \xi_i$	$\text{mean}\, \xi$
100	1.18e-12	3.01e-14	1.85e-13
250	4.25e-12	8.98e-14	5.44e-13
500	2.47e-11	1.80e-13	1.40e-12
1000	3.96e-11	5.29e-13	4.09e-12
2000	6.13e-10	1.19e-12	1.14e-11
2500	1.46e-09	1.14e-12	1.59e-11

From Table 3 we can confirm that Algorithm 1 supplies smaller error bounds for approximate eigenvalues than those by the method in [6] also in this example. It can

Table 5: Computing times (sec) in Example 2

n	t_a	t_v	t_M
100	0.03	0.03	0.06
250	0.19	0.17	0.45
500	1.36	1.06	3.14
1000	10.1	6.61	23.0
2000	77.4	47.2	179
2500	146	90.2	353

be seen from Table 4 that Algorithm 1 supplies sufficiently small error bounds for approximate eigenvectors in this example. Moreover we can confirm from Table 5 that t_v are smaller than t_M even though t_M is the computing time for enclosing eigenvalues only. Accordingly Algorithm 1 was from twice to four times faster than the method in [6] in this example. This identifies the fact that computational costs of Algorithm 1 and the method in [6] are $10n^3$ flops and $44n^3$ flops, respectively. We can also confirm that t_v are smaller than t_a for large n. Namely Algorithm 1 was faster than the computation of all approximate eigenpairs in almost all of the cases in this example.

One may be interested in enclosing *a few specified* eigenpairs. For instance, consider the case of $n = 1000$ and verifying $(\tilde{\lambda}_j, \tilde{x}^{(j)})$ for $j \in \{1, 2, 3, 998, 999, 1000\}$ in this example. In this case, we can apply an INTLAB [18] function `VerifyEig`. When we applied `VerifyEig` to this case, the obtained error bounds of $(\tilde{\lambda}_j, \tilde{x}^{(j)})$ for each j were approximately ten times as small as that by Algorithm 1. Note that `VerifyEig` does not necessarily compute the inclusions of λ_j and $x^{(j)}$ when eigenvalues are clustered near λ_j. The computing time for `VerifyEig` was approximately 35 sec for all j. Note that `VerifyEig` is designed to include one eigenpair, and can be significantly accelerated if more than one eigenvalue is to be included. As shown in Table 5, Algorithm 1 requires 6.61 sec to verify *all* approximate eigenpairs. From these it can be seen that Algorithm 1 is faster than `VerifyEig` although `VerifyEig` supplies smaller error bounds than those by Algorithm 1 in this case.

4.3 Example 3

In this example, we observe the sizes of error bounds and computing times when $\kappa(B)$ is moderately large. Consider the case that A and B are generated by

```
A=randn(n);
A=(A+A')/2;
B=gallery('randsvd',n,-1e6);
```

on MATLAB. We use the Higham's test matrix `randsvd` [5]. Then B is mostly symmetric positive definite with $\kappa(B) \approx$ 1e+6. Algorithm 1 verified that B is positive definite rigorously. Table 6 displays n_M/n and the similar quantities to Table 3. Tables 7 and 8 display the similar quantities to Tables 4 and 5, respectively.

From Tables 6 and 7 we can confirm the similar tendency to Tables 1 and 2 with respect to the error bounds, respectively. Moreover it can be seen that error bounds became larger comparing to that in Example 2. The tendencies about the computing time were similar to Table 5.

Table 6: Obtained error bounds for eigenvalues and n_M/n in Example 3

n	$\max \bar{\eta}_i$	$\min \bar{\eta}_i$	mean $\bar{\eta}$	$\max \bar{\eta}_i^{(M)}$	$\min \bar{\eta}_i^{(M)}$	mean $\bar{\eta}^{(M)}$	n_M/n
100	5.93e-09	3.74e-11	4.83e-10	–	8.59e-09	–	0.98
250	4.37e-08	1.43e-10	1.49e-09	–	5.42e-08	–	0.87
500	3.80e-08	3.50e-10	3.71e-09	–	3.07e-07	–	0.78
1000	5.69e-07	8.62e-10	1.03e-08	–	1.70e-06	–	0.69
2000	3.32e-07	2.28e-09	2.56e-08	–	9.03e-06	–	0.60
2500	1.85e-06	3.13e-09	3.69e-08	–	1.57e-05	–	0.57

Table 7: Obtained error bounds for eigenvectors in Example 3

n	$\max \xi_i$	$\min \xi_i$	mean ξ
100	5.86e-05	5.69e-07	1.07e-05
250	1.48e-03	5.11e-06	1.45e-04
500	1.09e-02	3.42e-05	1.04e-03
1000	1.03e-01	2.54e-04	7.45e-03
2000	3.04e-00	1.26e-03	5.74e-02
2500	4.06e-00	2.57e-03	1.20e-01

In the case of $n = 1000$, we applied `VerifyEig` to $(\tilde{\lambda}_j, \tilde{x}^{(j)})$ for $j \in \{1, 2, 3, 998, 999, 1000\}$. Then the verification of $(\tilde{\lambda}_1, \tilde{x}^{(1)})$ succeeded and the verification of the other eigenpairs failed. The error bound for $(\tilde{\lambda}_1, \tilde{x}^{(1)})$ was approximately ten times as small as that by Algorithm 1. The computing time for `VerifyEig` was approximately 60 sec for all j. As shown in Table 8, Algorithm 1 requires 6.63 sec to verify *all* approximate eigenpairs. From these it can be seen that Algorithm 1 is robuster than `VerifyEig` although `VerifyEig` supplies smaller error bound for $(\tilde{\lambda}_1, \tilde{x}^{(1)})$ than that by Algorithm 1 in this case.

4.4 Example 4

In this example, we observe how the sizes of error bounds change when $\kappa(A)$ increases. Consider the case that 1000 × 1000 matrices A and B are generated by the following MATLAB code:

```
cond10 = log10(cond); % cond: anticipated condition number
s = sign(randn(1,1000));
D = diag(s .* logspace(0,cond10,1000));
X = randorth(1000);
A = X*D*X';
A = (A+A')/2;
B=randn(1000);
B=n*eye(1000)+(B+B')/2;
```

We use the INTLAB function `randorth` for generating a random (approximately) orthogonal matrix. Then A is symmetric with $\kappa(A) \approx$ `cond`. Algorithm 1 verified that B is positive definite. For various `cond` Tables 9 and 10 display the similar quantities to Tables 6 and 7, respectively.

From Table 9, we can confirm the similar tendency to Table 1 with respect to the relation between $\bar{\eta}_i$ and $\bar{\eta}^{(M)}$. Moreover Tables 9 and 10 show that error bounds increase as $\kappa(A)$ increases.

Table 8: Computing times (sec) in Example 3

n	t_a	t_v	t_M
100	0.03	0.03	0.06
250	0.25	0.17	0.49
500	1.13	1.06	3.25
1000	8.11	6.63	22.5
2000	62.4	46.1	177
2500	118	88.7	346

Table 9: Obtained error bounds for eigenvalues and n_M/n in Example 4

`cond`	$\max\bar{\eta}_i$	$\min\bar{\eta}_i$	mean $\bar{\eta}$	$\max\bar{\eta}_i^{(M)}$	$\min\bar{\eta}_i^{(M)}$	mean $\bar{\eta}^{(M)}$	n_M/n
1e+02	1.69e-12	1.33e-13	4.22e-13	3.05e-09	3.01e-11	6.52e-10	1.00
1e+04	1.16e-10	1.32e-13	1.22e-11	1.80e-07	1.76e-11	1.94e-08	1.00
1e+06	9.28e-09	1.34e-13	6.60e-10	1.31e-05	1.34e-11	9.58e-07	1.00
1e+08	8.37e-07	1.33e-13	4.29e-08	1.06e-03	1.06e-11	5.74e-05	1.00
1e+10	7.46e-05	1.31e-13	3.11e-06	1.01e-01	9.43e-12	4.25e-03	1.00
1e+12	6.75e-03	1.34e-13	2.31e-04	–	7.94e-12	–	0.93

In the case of `cond` = 1e+12, we applied `VerifyEig` to $(\tilde{\lambda}_j, \tilde{x}^{(j)})$ for $j \in \{1, 2, 3, 998, 999, 1000\}$. The obtained error bounds of $(\tilde{\lambda}_j, \tilde{x}^{(j)})$ for each j were approximately 10^{10} times as small as that by Algorithm 1. From this it can be seen that `VerifyEig` is very useful for verifying a few specified eigenpairs in the case that $\kappa(A)$ is large and small error bounds are required.

4.5 Example 5

In this example, we observe the property of Algorithm 1 and the method in [6] when some eigenvalues are near from zero and closely clustered. Consider the case that A and B are defined as

$$A = \begin{pmatrix} 6 & 3a & -6 & 3a \\ 3a & 2a^2 & -3a & a^2 \\ -6 & -3a & 6 & -3a \\ 3a & a^2 & -3a & 2a^2 \end{pmatrix}, \quad B = \begin{pmatrix} 156 & 22a & 54 & -13a \\ 22a & 4a^2 & 13a & -3a^2 \\ 54 & 13a & 156 & -22a \\ -13a & -3a^2 & -22a & 4a^2 \end{pmatrix}$$

where a is a parameter [7]. This example relates vibration analysis. We consider the case that $a = 2$. In this case, $\kappa(B) \approx$ 2.9e+2. Algorithm 1 verified that B is positive definite. Moreover we obtained $\tilde{\lambda}_1 = -$1.958e-16, $\tilde{\lambda}_2 =$7.608e-17, $\tilde{\lambda}_3 =$0.8572 and $\tilde{\lambda}_4 =$10.001 by `eig`. Thus $\tilde{\lambda}_1$ and $\tilde{\lambda}_2$ are near from zero and closely clustered.

Table 11 displays η_i, ξ_i and $\eta_i^{(M)}$ for $i = 1, \ldots, 4$. In Table 11 the notation "–" means that the corresponding error bound became negative so that verification for $\tilde{\lambda}_i$ failed in the sense of relative error bound. Moreover the notation "$*$" means that $\bar{\rho}_i \leq 0$, where $\bar{\rho}_i$ is defined as in Theorem 7, so that verification for $\tilde{x}^{(i)}$ failed in the sense of absolute error bound.

From Table 11 we can confirm that both of Algorithm 1 and the method in [6] failed in the verifications of $\tilde{\lambda}_1$ and $\tilde{\lambda}_2$ in the sense of relative error bound. The reason is that $\tilde{\lambda}_1$ and $\tilde{\lambda}_2$ are too near from zero compared with $\tilde{\lambda}_4$. Moreover it can be seen that Algorithm 1 also failed in the verifications of $\tilde{x}^{(1)}$ and $\tilde{x}^{(2)}$ in the sense of absolute error bound. The reason is that $\tilde{\lambda}_1$ and $\tilde{\lambda}_2$ are closely clustered.

Table 10: Obtained error bounds for eigenvectors in Example 4

`cond`	$\max \xi_i$	$\min \xi_i$	mean ξ
1e+02	3.02e-11	1.76e-13	4.59e-12
1e+04	8.24e-10	1.65e-13	5.90e-11
1e+06	3.63e-08	1.14e-13	2.11e-09
1e+08	2.32e-06	6.96e-14	9.66e-08
1e+10	1.70e-04	2.44e-14	5.98e-06
1e+12	6.75e-03	1.34e-13	2.31e-04

Table 11: Numerical Results in Example 5

i	$\bar{\eta}_i$	$\bar{\eta}_i^{(M)}$	$\bar{\xi}_i$
1	–	–	$*$
2	–	–	$*$
3	2.49e-14	1.73e-13	3.46e-14
4	3.34e-14	8.53e-15	5.08e-14

We applied `VerifyEig` to $(\tilde{\lambda}_i, \tilde{x}^{(i)})$ for all i. Then `VerifyEig` also failed in the verification of $(\tilde{\lambda}_1, \tilde{x}^{(1)})$ and $(\tilde{\lambda}_2, \tilde{x}^{(2)})$. Alternatively `VerifyEig` succeeded in the verification of $(\tilde{\lambda}_3, \tilde{x}^{(3)})$ and $(\tilde{\lambda}_4, \tilde{x}^{(4)})$. The obtained error bounds for $(\tilde{\lambda}_3, \tilde{x}^{(3)})$ and $(\tilde{\lambda}_4, \tilde{x}^{(4)})$ were approximately equal to those by Algorithm 1. From these we can confirm that Algorithm 1 supplied the comparable results with `VerifyEig` in this example.

5 Application to Quadratic Eigenvalue Problem

As an application of the proposed method, in this section, we sketch an efficient method of enclosing all eigenpairs in the quadratic eigenvalue problem

$$\begin{aligned}&(\lambda_i^2 A + \lambda_i B + C)x^{(i)} = 0\\ &A, B, C \in \mathbb{R}^{n\times n},\ \lambda_i \in \mathbb{R},\ x^{(i)} \in \mathbb{R}^n,\ i = 1, \dots, 2n\end{aligned} \tag{59}$$

where A is symmetric negative definite, B is symmetric and C is symmetric positive definite. Then (59) is equivalent to the following generalized eigenvalue problem

$$\begin{aligned}&\acute{A}\acute{x}^{(i)} = \lambda_i \acute{B}\acute{x}^{(i)}\\ &\acute{A}, \acute{B} \in \mathbb{R}^{2n\times 2n},\ \lambda_i \in \mathbb{R},\ \acute{x}^{(i)} \in \mathbb{R}^{2n},\ i = 1, \dots, 2n\end{aligned} \tag{60}$$

where

$$\acute{A} := \begin{pmatrix} B & C \\ C & 0 \end{pmatrix}, \quad \acute{x}^{(i)} := \begin{pmatrix} \lambda_i x^{(i)} \\ x^{(i)} \end{pmatrix}, \quad \acute{B} := \begin{pmatrix} -A & 0 \\ 0 & C \end{pmatrix}.$$

$\acute{A}$ is symmetric and $\acute{B}$ is symmetric positive definite. Therefore all eigenpairs in (59) can be enclosed by applying the proposed method to (60). With respect to approximate eigenpairs $(\tilde{\lambda}_i, \tilde{x}^{(i)})$ for all i, it can be expected that $\acute{A}\tilde{X} \approx \acute{B}\tilde{X}\tilde{D}$ and $I_{2n} \approx \tilde{X}^T \acute{B}\tilde{X}$

where I_m denotes the $m \times m$ identity matrix and

$$\begin{aligned} \tilde{X} &:= \begin{pmatrix} \tilde{X}_1\tilde{D}_1 & \tilde{X}_2\tilde{D}_2 \\ \tilde{X}_1 & \tilde{X}_2 \end{pmatrix}, \quad \tilde{D} := \begin{pmatrix} \tilde{D}_1 & 0 \\ 0 & \tilde{D}_2 \end{pmatrix} \\ \tilde{X}_1 &:= (\tilde{x}^{(1)}, \dots, \tilde{x}^{(n)}), \quad \tilde{X}_2 := (\tilde{x}^{(n+1)}, \dots, \tilde{x}^{(2n)}) \\ \tilde{D}_1 &:= \mathrm{diag}(\tilde{\lambda}_1, \dots, \tilde{\lambda}_n), \quad \tilde{D}_2 := \mathrm{diag}(\tilde{\lambda}_{n+1}, \dots, \tilde{\lambda}_{2n}). \end{aligned}$$

Accordingly as regards to the parts in the proposed method whose computational costs are $\mathcal{O}(n^3)$ flops, we obtain

$$\begin{aligned} \acute{A}\tilde{X} - \acute{B}\tilde{X}\tilde{D} &= \begin{pmatrix} E_1 & E_2 \\ 0 & 0 \end{pmatrix} \\ I_{2n} - \tilde{X}^T\acute{B}\tilde{X} &= \begin{pmatrix} I_n - F_{11} & -F_{12} \\ -F_{21} & I_n - F_{22} \end{pmatrix} \\ E_j &:= A\tilde{X}_j\tilde{D}_j^2 + B\tilde{X}_j\tilde{D}_j + C\tilde{X}_j \\ F_{jk} &:= -\tilde{X}_j\tilde{D}_jA\tilde{X}_k\tilde{D}_k + \tilde{X}_jC\tilde{X}_k. \end{aligned}$$

We can reuse the results of matrix multiplications $A\tilde{X}_j$ and $C\tilde{X}_j$ for $j \in \{1, 2\}$.

6 Conclusion

In this paper, we proposed a fast method of enclosing all eigenpairs for the generalized eigenvalue problem (1) where A is symmetric and B is symmetric positive definite. Some numerical results were reported to show the performance of the proposed method.

As an application of the proposed method, we also sketched an efficient method of enclosing all eigenpairs in the quadratic eigenvalue problem (59) where A is symmetric negative definite, B is symmetric and C is symmetric positive definite.

By modifying the proposed method slightly, enclosing all eigenpairs in (1) where A is Hermitian and B is Hermitian positive definite is possible.

Acknowledgements

This research was partially supported by Grant-in-Aid for Young Scientists (B) (19760055, 2007–2010) from the Ministry of Education, Science, Sports and Culture of Japan.

References

[1] Behnke, H.: Inclusion of Eigenvalues of General Eigenvalue Problems for Matrices, in: U. Kulisch, H. J. Stetter (Eds.), Scientific Computation with Automatic Result Verification, *Computing Supplement* **6** (1988), pp. 69–78.

[2] Behnke, H.: The Calculation of Guaranteed Bounds for Eigenvalues Using Complementary Variational Principles, *Computing* **47** (1991), pp. 11–27.

[3] Cao, Z. H., Xie, J. J., and Li, R. C.: A sharp version of Kahan's theorem on clustered eigenvalues, *Linear Algebra and its Applications* **245** (1996), pp. 147–155.

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

[5] Higham, N. J.: *Accuracy and Stability of Numerical Algorithms*, Second ed., SIAM Publications, Philadelphia, 2002.

[6] Maruyama, K., Ogita, T., Nakaya, Y., and Oishi, S.: Numerical Inclusion Method for all Eigenvalues of Real Symmetric Definite Generalized Eigenvalue Problem, *IEICE Trans.* **J87–A(8)** (2004), pp. 1111–1119 [in Japanese].

[7] Meirovitch, L.: *Elements of Vibration Analysis*, Second ed., McGraw-Hill, New York, 1986.

[8] Miyajima, S., Ogita, T., Ozaki, K., and Oishi, S.: Fast Error Estimation for Eigenvalues of Symmetric Matrix without Directed Rounding, *Proceedings of 2004 International Symposium on Nonlinear Theory and its Applications*, Fukuoka, Japan (2004), pp. 167–170.

[9] Miyajima, S., Ogita, T., and Oishi, S.: Fast Verification for Respective Eigenvalues of Symmetric Matrix, *Lecture Notes in Computer Science* **3718** (2005), pp. 306–317.

[10] Miyajima, S., Ogita, T., and Oishi, S.: Numerical Verification for Each Eigenpair of Symmetric Matrix, *Trans. JSIAM* **16(4)** (2006), pp. 535–552 [in Japanese].

[11] Ogita, T. and Oishi, S.: Fast Verification for Large-scale Systems of Linear Equations, *IPSJ Trans.* **46:SIG10 (TOM12)** (2005), pp. 10–18 [in Japanese].

[12] Oishi, S.: Fast Enclosure of Matrix Eigenvalues and Singular Values via Rounding Mode Controlled Computation, *Linear Algebra and its Applications* **324** (2001), pp. 133–146.

[13] Oishi, S. and Rump, S. M.: Fast Verification of Solutions of Matrix Equations, *Numer. Math.* **90(4)** (2002), pp. 755–773.

[14] Ozaki, K., Ogita, T., Miyajima, S., Oishi, S., and Rump, S. M.: A Method of Obtaining Verified Solutions for Linear Systems suited for Java, *Journal of Computational and Applied Mathematics* **199(2)** (2007), pp. 337–344.

[15] Parlett, B. N.: *The Symmetric Eigenvalue Problem*, Classics in Applied Mathematics, 20, SIAM Publications, Philadelphia, 1997.

[16] Rump, S. M.: Guaranteed Inclusions for the Complex Generalized Eigenproblem, *Computing* **42** (1989), pp. 225–238.

[17] Rump, S. M.: Verification Methods for Dense and Sparse Systems of Equations, *Topics in Validated Computations*, Elsevier Science (1994), pp. 63–135.

[18] Rump, S. M.: INTLAB - INTerval LABoratory, *Developments in Reliable Computing* (T. Csendes ed.), Kluwer Academic Publishers, Dordrecht (1999), pp. 77–107.

[19] Rump, S. M.: Computational Error Bounds for Multiple or Nearly Multiple Eigenvalues, *Linear Algebra and its Applications* **324** (2001), pp. 209–226.

[20] Rump, S. M. and Zemke, J.: On Eigenvector Bounds, *BIT Numerical Mathematics* **43** (2004), pp. 823–837.

[21] Rump, S. M.: Verification of Positive Definiteness, *BIT Numerical Mathematics* **46** (2006), pp. 433–452.

[22] Rump, S. M. and Ogita, T.: Super–fast Validated Solution of Linear Systems, *Journal of Computational and Applied Mathematics* **199(2)** (2007), pp. 199–206.

[23] Stewart, G. W. and Sun, J. G.: *Matrix Perturbation Theory*, Academic Press, New York, 1990.

[24] Watanabe, Y., Yamamoto, N., and Nakao, M. T.: Verification Methods of Generalized Eigenvalue Problems and its Applications, *Trans. JSIAM* **9(3)** (1999), pp. 137–150 [in Japanese].

[25] Wilkinson, J. H.: Rigorous Error Bounds for Computed Eigensystem, *Computer J.* **4** (1961), pp. 230–24.

[26] Yamamoto, N.: A Simple Method for Error Bounds of Eigenvalues of Symmetric Matrices, *Linear Algebra and its Applications* **324** (2001), pp. 227–234.