Journal of Computational and Applied Mathematics 236 (2012) 2545–2552

Contents lists available at SciVerse ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Numerical enclosure for each eigenvalue in generalized eigenvalue problem

Shinya Miyajima

Faculty of Engineering, Gifu University, 501-1193, Japan

ARTICLE INFO

Article history:
Received 25 September 2010
Received in revised form 30 November 2011

MSC:
65F15
65G20
65G50

Keywords:
Generalized eigenvalue problem
Numerical enclosure
Non-Hermitian matrices

ABSTRACT

An algorithm for enclosing all eigenvalues in generalized eigenvalue problem $Ax = \lambda Bx$ is proposed. This algorithm is applicable even if $A \in \mathbb{C}^{n\times n}$ is not Hermitian and/or $B \in \mathbb{C}^{n\times n}$ is not Hermitian positive definite, and supplies *n error bounds* while the algorithm previously developed by the author supplies *a single error bound*. It is proved that the error bounds obtained by the proposed algorithm are equal or smaller than that by the previous algorithm. Computational cost for the proposed algorithm is similar to that for the previous algorithm. Numerical results show the property of the proposed algorithm.

1. Introduction

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

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

where λ is an eigenvalue and $x \neq 0$ is an eigenvector corresponding to λ. Assume that B is nonsingular. The problem (1) arises in many applications of scientific computations, e.g., stationary analysis of circuits, image processing, structure analysis and so forth [1,2].

There are several algorithms for enclosing eigenvalues in (1), e.g., [3–11]. On enclosing eigenvalues in the case when A is Hermitian and B is Hermitian positive definite; see [3–5,7,10,11]. The case when A is Hermitian and B is Hermitian positive definite is important (see e.g., [12]). On the other hand, the cases when A is not Hermitian and/or B is not Hermitian positive definite are also important. For instance, these cases arise in the finite element analysis of Maxwell's equation and forward kinematics for the Stewart platform of robotics [13]. *A few specified eigenvalues and eigenvectors* can be enclosed by applying the algorithms in [8,9] even to the non-Hermitian cases.

Assume as a result of numerical computation, we have an $n \times n$ complex diagonal matrix $\tilde{D}$ and an $n \times n$ complex matrix $\tilde{X}$ such that $A\tilde{X} \approx B\tilde{X}\tilde{D}$ follows approximately. Maruyama et al. [5] have developed an algorithm for enclosing *all eigenvalues* in the Hermitian case. Their algorithm can be expanded to the non-Hermitian cases, and is based on the following theorem:

Theorem 1 (*Maruyama et al. [5]*). *Suppose $\tilde{X}$ is nonsingular. Let $\mathbf{D}$ be an $n \times n$ interval matrix including $\tilde{X}^{-1}B^{-1}A\tilde{X}$, $\tilde{\mu}_i,\ i = 1, \ldots, n$ be the center of $\mathbf{D}_{ii}$, where $\mathbf{D}_{ij}$ is the (i, j) element of $\mathbf{D}$, $\Delta \in \mathbb{C}^{n\times n}$ be diagonal with $\Delta_{ii} = \tilde{\mu}_i$, and $\mathbf{Q} := \mathbf{D} - \Delta$. Then all*

E-mail address: miyajima@gifu-u.ac.jp.

0377-0427/$ – see front matter
doi:10.1016/j.cam.2011.12.013

eigenvalues in (1) are included in the set

$$\bigcup_{i=1}^{n} \{z \in \mathbb{C} : |z - \tilde{\mu}_i| \le \zeta_i\}, \quad \zeta_i := \sum_{j=1}^{n} |\mathbf{Q}_{ij}|.$$

Their algorithm, hereafter we name this Algorithm 1, supplies approximate eigenvalues $\tilde{\mu}_1, \ldots, \tilde{\mu}_n$ and n *error bounds* $\zeta_1, \ldots, \zeta_n$ by computing $\mathbf{D}$ and $\mathbf{Q}$ directly. The computational cost for Algorithm 1 is $148n^3$ flops.

Alternatively the author [6] has proposed a fast algorithm for enclosing *all eigenvalues*, which is applicable even to the non-Hermitian cases. This algorithm is based on the following theorem:

Theorem 2 (*Miyajima [6]*). *Let $\tilde{\lambda}_i := \tilde{D}_{ii}$, $i = 1, \ldots, n$, and I be the $n \times n$ identity matrix. For an arbitrary $n \times n$ complex matrix Y, let $n \times n$ complex matrices R and S be defined as follows:*

$$R := Y(A\tilde{X} - B\tilde{X}\tilde{D}), \qquad S := YB\tilde{X} - I.$$

If $\|S\|_\infty < 1$, then $B, \tilde{X}$ and Y are nonsingular, and all eigenvalues in (1) are included in the set

$$\bigcup_{i=1}^{n} \left\{ z \in \mathbb{C} : |z - \tilde{\lambda}_i| \le \varepsilon \right\}, \quad \varepsilon := \frac{\|R\|_\infty}{1 - \|S\|_\infty}.$$

This algorithm, hereafter we call this Algorithm 2, computes *a single error bound* ε and requires $66n^3$ flops. In the practical execution of Algorithm 2, Y is computed such that $Y \approx (B\tilde{X})^{-1}$ (see [6] for detail). The nonsingularity of B can be verified during Algorithm 2.

In this paper, we present a theorem for enclosing *all eigenvalues*, but now in a componentwise sense. This theorem is applicable even to the non-Hermitian cases, and supplies n *error bounds* $r_1, \ldots, r_n$ such that the all eigenvalues are included in the set

$$\bigcup_{i=1}^{n} \left\{ z \in \mathbb{C} : |z - \tilde{\lambda}_i| \le r_i \right\},$$

when $\tilde{D}$ and $\tilde{X}$ are given. We present a theorem showing $\max_{1 \le i \le n} r_i \le \varepsilon$, and propose an algorithm for enclosing the all eigenvalues based on the first theorem. This algorithm takes into account the presence of underflow in floating point arithmetic. Computational cost for this algorithm is similar to that for Algorithm 2.

This paper is organized as follows: In Section 2, theory for computing $r_1, \ldots, r_n$ is presented and $\max_{1 \le i \le n} r_i \le \varepsilon$ is proved. In Section 3, an algorithm for enclosing all eigenvalues in (1) is proposed. In Section 4, numerical results are reported to show the properties of the proposed algorithm. Finally Section 5 summarizes the results in this paper, and highlights possible extensions and future work.

2. Enclosure theory

In this section, we establish theory for enclosing all eigenvalues in (1). Throughout this paper, assume as a result of numerical computation, we have the matrices $\tilde{D}$ and $\tilde{X}$ in Section 1. Let $\tilde{\lambda}_i$ be defined as in Theorem 2. Then $A\tilde{x}^{(i)} \approx \tilde{\lambda}_i B\tilde{x}^{(i)}$, $i = 1, \ldots, n$ holds approximately, where $\tilde{x}^{(i)}$ denotes the i-th column of $\tilde{X}$. Let I be defined as in Theorem 2 and $e := (1, \ldots, 1)^T \in \mathbb{R}^n$. For $M \in \mathbb{C}^{n \times n}$, M_{ij}, $|M|$ and M^T are defined as the (i, j) element of M, $|M| := \{|M_{ij}|\}$ and $M^T := \{M_{ji}\}$, respectively. For $v \in \mathbb{R}^n$, v_i denotes the i-th component of v. For a complex number η and a nonnegative real number δ, $\mathcal{C}(\eta, \delta)$ is defined as follows:

$$\mathcal{C}(\eta, \delta) := \{z \in \mathbb{C} : |z - \eta| \le \delta\}.$$

We present Theorem 3 which supplies n *error bounds* for $\tilde{\lambda}_1, \ldots, \tilde{\lambda}_n$.

Theorem 3. *Let Y, R and S be defined as in Theorem 2. If $\|S\|_\infty < 1$, then $B, \tilde{X}$ and Y are nonsingular, and all eigenvalues in (1) are included in the set*

$$\bigcup_{i=1}^{n} \mathcal{C}\left(\tilde{\lambda}_i, r_i\right),$$

where r_i is the i-th component of a real n-vector r defined as follows:

$$r := |R|e + \frac{\|R\|_\infty}{1 - \|S\|_\infty} s, \quad s := \left(\sum_{j=1}^{n} |S_{1j}|, \ldots, \sum_{j=1}^{n} |S_{nj}| \right)^T.$$

Proof. Similarly to [6, Proof of Theorem 1], $\|S\|_\infty < 1$ implies the nonsingularities of B, $\tilde{X}$ and Y. Therefore (1) is equivalent to the following standard eigenvalue problem:

$$(\tilde{D} + Q)y = \lambda y, \quad Q := (YB\tilde{X})^{-1}R,\ y := \tilde{X}^{-1}x.$$

Hence Gershgorin circle theorem (e.g., [14, Theorem 7.2.1]) gives that the all eigenvalues are included in the set

$$\bigcup_{i=1}^{n} \mathcal{C}\left(\tilde{D}_{ii} + Q_{ii}, \sum_{j=1, j\neq i}^{n} |Q_{ij}|\right). \tag{2}$$

We obtain

$$\bigcup_{i=1}^{n} \mathcal{C}\left(\tilde{D}_{ii} + Q_{ii}, \sum_{j=1, j\neq i}^{n} |Q_{ij}|\right) \subseteq \bigcup_{i=1}^{n} \mathcal{C}\left(\tilde{D}_{ii}, \sum_{j=1}^{n} |Q_{ij}|\right) = \bigcup_{i=1}^{n} \mathcal{C}\left(\tilde{\lambda}_i, (|Q|e)_i\right). \tag{3}$$

Neumann series (e.g., [15, Chapter 7]) and $\|S\|_\infty < 1$ yield

$$\begin{aligned}
|Q|e &\leq |(YB\tilde{X})^{-1}||R|e = |(I - (-S))^{-1}||R|e = |I + (-S) + (-S)^2 + \cdots||R|e \\
&\leq (I + |S| + |S|^2 + \cdots)|R|e = |R|e + |S|(|R|e) + |S|(|S||R|e) + |S|(|S|^2|R|e) + \cdots \\
&\leq |R|e + \||R|e\|_\infty s + \||S||R|e\|_\infty s + \||S|^2|R|e\|_\infty s + \cdots \\
&= |R|e + (\||R|e\|_\infty + \||S||R|e\|_\infty + \||S|^2|R|e\|_\infty + \cdots)s \\
&\leq |R|e + (\||R|e\|_\infty + \|S\|_\infty\||R|e\|_\infty + \|S\|_\infty^2\||R|e\|_\infty + \cdots)s \\
&= |R|e + (\|R\|_\infty + \|S\|_\infty\|R\|_\infty + \|S\|_\infty^2\|R\|_\infty + \cdots)s = |R|e + \|R\|_\infty(1 + \|S\|_\infty + \|S\|_\infty^2 + \cdots)s \\
&= |R|e + \frac{\|R\|_\infty}{1 - \|S\|_\infty}s = r.
\end{aligned} \tag{4}$$

The result follows from (2), (3) and (4). □

Remark 1. Let R, S and ε be defined as in Theorem 2, and Q be defined as in the proof of Theorem 3. We have

$$\bigcup_{i=1}^{n} \mathcal{C}\left(\tilde{D}_{ii}, \sum_{j=1}^{n} |Q_{ij}|\right) \subseteq \bigcup_{i=1}^{n} \mathcal{C}\left(\tilde{D}_{ii}, \max_{1\leq i\leq n}\left(\sum_{j=1}^{n} |Q_{ij}|\right)\right) = \bigcup_{i=1}^{n} \mathcal{C}\left(\tilde{\lambda}_i, \|Q\|_\infty\right).$$

If $\|S\|_\infty < 1$, it holds that

$$\begin{aligned}
\|Q\|_\infty &\leq \|(YB\tilde{X})^{-1}\|_\infty\|R\|_\infty = \|(I - (-S))^{-1}\|_\infty\|R\|_\infty = \|I + (-S) + (-S)^2 + \cdots\|_\infty\|R\|_\infty \\
&\leq (1 + \|S\|_\infty + \|S\|_\infty^2 + \cdots)\|R\|_\infty = \frac{\|R\|_\infty}{1 - \|S\|_\infty} = \varepsilon.
\end{aligned}$$

Hence all eigenvalues in (1) are included in the set $\bigcup_{i=1}^{n} \mathcal{C}(\tilde{\lambda}_i, \varepsilon)$. This is an alternative proof of Theorem 2 given by Michael Plum during private communication in 2008.

We present Theorem 4 for clarifying the relation between ε and r in Theorems 2 and 3, respectively.

Theorem 4. *Let ε and r be defined as in Theorems 2 and 3, respectively. Then $\max_{1\leq i\leq n} r_i \leq \varepsilon$ holds.*

Proof. We obtain

$$\begin{aligned}
\max_{1\leq i\leq n} r_i &= \|r\|_\infty \leq \||R|e\|_\infty + \frac{\|R\|_\infty}{1 - \|S\|_\infty}\|s\|_\infty = \|R\|_\infty + \frac{\|R\|_\infty\|S\|_\infty}{1 - \|S\|_\infty} \\
&= \frac{\|R\|_\infty(1 - \|S\|_\infty) + \|R\|_\infty\|S\|_\infty}{1 - \|S\|_\infty} = \frac{\|R\|_\infty}{1 - \|S\|_\infty} = \varepsilon. \quad \square
\end{aligned}$$

3. Proposed algorithm

In this section, we propose an algorithm for enclosing all eigenvalues in (1). This algorithm computes r in Theorem 3 considering rounding errors. Hereafter $\mathrm{fl}(\cdot)$ denotes a 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 $\mathrm{fl}_\triangledown(\cdot)$ in rounding-downward. Denote a relative rounding error unit and an underflow constant by $\mathbf{u}$ and $\underline{\mathbf{u}}$, respectively. For IEEE754 double precision, we have $\mathbf{u} = 2^{-53}$ and $\underline{\mathbf{u}} = 2^{-1074}$. Then define γ_n and γ'_n as $\gamma_n := n\mathbf{u}/(1 - n\mathbf{u})$ and $\gamma'_n := \sqrt{5}\mathbf{u} + \gamma_n(1 + \sqrt{5}\mathbf{u})$, respectively. For $F_c \in \mathbb{C}^{n\times n}$ and $F_r \in \mathbb{R}^{n\times n}$, where all elements in F_r are nonnegative, the notation $\langle F_c, F_r\rangle$ denotes a matrix interval whose center and radius are F_c and F_r, respectively.

Let Y and S be defined as in Theorem 2, and s be defined as in Theorem 3. Denote the real and the imaginary part of Y by Y_r and Y_i, respectively. Let all elements in $Z_r \in \mathbb{R}^{n\times n}$ be nonnegative, and $Z_c \in \mathbb{C}^{n\times n}$ and Z_r satisfy $B\tilde{X} \in \langle Z_c, Z_r\rangle$. It can be shown from [6, Section 3] that

$$s = |S|e \le |\mathrm{fl}_\square(YZ_c - I)|e + (|Y_r| + |Y_i|)|Z_r|e + \gamma'_{n-1}|Y||Z_c|e + \mathbf{u}(|\mathrm{fl}_\square(YZ_c)| + I)e + 4n^2\underline{\mathbf{u}}(1 + \gamma_{n-1})e, \tag{5}$$

also in the presence of underflow, if $\mathbf{u} \le 2^{-5}$. Thus a rigorous upper bound for s can be computed with $\mathcal{O}(n^2)$ flops if Y, Z_c, Z_r and $\mathrm{fl}_\square(YZ_c)$ have already been obtained.

Based on this discussion, we present steps of the proposed algorithm.

Algorithm 3. Let Y, S and R be defined as in Theorem 2, s be defined as in Theorem 3, and Z_c and Z_r be as in the above discussion. This algorithm computes a real n-vector $r = (r_1, \ldots, r_n)^T$ such that $r_i \ge 0$ for $i = 1, \ldots, n$ and all eigenvalues in (1) are included in the set

$$\bigcup_{i=1}^{n} \mathcal{C}\left(\tilde{\lambda}_i, r_i\right)$$

on the assumptions that $\tilde{D}$ and $\tilde{X}$ are given, and $\mathbf{u} \le 2^{-5}$.

Step 1 Compute Z_c and Z_r by applying [6, Algorithms 2 and 3].
Step 2 Compute Y such that $Y \approx Z_c^{-1}$.
Step 3 Compute $\mathrm{fl}_\square(YZ_c)$.
Step 4 Compute σ, a rigorous upper bound for $\|S\|_\infty$, based on [6, Corollary 1] using Y, Z_c, Z_r and $\mathrm{fl}_\square(YZ_c)$.
Step 5 If $\sigma \ge 1$, then terminate with failure. Otherwise go to Step 6.
Step 6 Compute $\bar{s}$, a rigorous upper bound for s, based on (5) reusing Y, Z_c, Z_r and $\mathrm{fl}_\square(YZ_c)$.
Step 7 Compute $[R]$, a rigorous enclosure for R, by applying [6, Procedure 2] reusing Y, Z_c and Z_r.
Step 8 Compute r such that $r = \mathrm{fl}_\triangle\left(|[R]|e + \frac{\|[R]\|_\infty}{\mathrm{fl}_\triangledown(1-\sigma)}\bar{s}\right)$.

Similarly to Algorithm 2, nonsingularity of B can be verified during Algorithm 3.

We see from [6] that Steps 1 and 7 require $16n^3$ and $34n^3$ flops, respectively. Step 3 requires $8n^3$ flops. Computational costs for Steps 4, 6 and 8 are $\mathcal{O}(n^2)$ flops. Hence Algorithm 3 requires $58n^3$ flops except the computation of $Y \approx Z_c^{-1}$. Algorithm 2 also requires $58n^3$ flops except it. Therefore computational cost for Algorithm 3 is similar to that for Algorithm 2 if Y is computed similarly between two algorithms.

4. Numerical results

In this section, we report some numerical results to show the property of Algorithm 3 and performance of our implementation. We used a computer with Intel Xeon 2.66 GHz Dual CPU, 4.00 GB RAM and MATLAB 7.5 with ATLAS and IEEE 754 double precision.

We applied the MATLAB functions `eig` and `inv` to obtain $\tilde{D}$ and $\tilde{X}$, and to compute $Y \approx Z_c^{-1}$ in Algorithm 3, respectively. We computed Y similarly to Algorithm 2. Consequently computational cost for Algorithm 3 is similar to that for Algorithm 2 and $66n^3$ flops.

Let $\tilde{\mu}_i$ and ζ_i be defined as in Theorem 1, and ε and r be as in Theorems 2 and 3, respectively. For nonnegative real numbers $q_1, \ldots, q_n$, mean q_i denotes

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

to see the mean values of error bounds. Let t_λ, $t_{\lambda x}$, t_ζ, t_ε and t_r be the computing times (s) for calculating $\tilde{D}$, calculating $\tilde{D}$ and $\tilde{X}$, Algorithms 1, 2, and 3, respectively. For $M \in \mathbb{C}^{n\times n}$, define the condition number $\kappa(M) := \|M\|_2\|M^{-1}\|_2$ if M is nonsingular, and $\kappa(M) := \infty$ if M is singular.

Table 1
Obtained error bounds in Example 1.

n	$\max \zeta_i$	$\text{mean } \zeta_i$	$\min \zeta_i$	ε	$\max r_i$	$\text{mean } r_i$	$\min r_i$
500	6.45e−06	2.01e−06	1.02e−06	7.25e−08	7.25e−08	3.46e−09	7.90e−10
1000	5.81e−05	2.08e−05	9.62e−06	3.99e−07	3.99e−07	1.87e−08	4.13e−09
1500	3.78e−04	9.60e−05	3.50e−05	1.41e−06	1.41e−06	5.65e−08	9.35e−09
2000	5.25e−04	1.92e−04	7.82e−05	1.45e−06	1.45e−06	1.02e−07	2.35e−08

Table 2
Computing times (s) in Example 1.

n	t_λ	$t_{\lambda x}$	t_ζ	t_ε	t_r
500	7.05	12.7	8.47	4.03	4.01
1000	59.4	100	63.5	29.9	29.8
1500	200	339	215	100	100
2000	513	869	508	236	236

Table 3
Obtained error bounds in Example 2.

cndA	$\max \zeta_i$	$\text{mean } \zeta_i$	$\min \zeta_i$	ε	$\max r_i$	$\text{mean } r_i$	$\min r_i$
1e+04.	3.54e−04	1.05e−04	2.43e−05	1.94e−06	1.94e−06	2.26e−07	1.81e−08
1e+06.	1.32e−02	5.57e−03	1.28e−03	7.78e−05	7.78e−05	1.80e−05	1.47e−06
1e+08.	1.23e+00	6.00e−01	1.10e−01	1.53e−02	1.53e−02	1.77e−03	1.18e−04
1e+10.	2.88e+02	9.27e+01	1.60e+01	1.30e+00	1.30e+00	1.73e−01	1.03e−02
1e+12.	1.29e+04	7.20e+03	1.34e+03	1.37e+02	1.37e+02	1.54e+01	1.06e+00

Algorithms 2 and 3 verified the nonsingularity of B for examples in which these algorithms succeeded. Tendencies regarding to computing times in Sections 4.2 and 4.3 were similar to that in Section 4.1.

4.1. Example 1

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

```
A = randn(n) + i*randn(n);
B = randn(n) + i*randn(n);
```

on MATLAB. Then the real and the imaginary parts of entries in A and B are pseudo random numbers uniformly distributed in $[-1, 1]$. Table 1 displays $\max_{1\le i\le n} \zeta_i$, $\text{mean } \zeta_i$, $\min_{1\le i\le n} \zeta_i$, ε, $\max_{1\le i\le n} r_i$, $\text{mean } r_i$ and $\min_{1\le i\le n} r_i$ for various n. Table 2 shows t_λ, $t_{\lambda x}$, t_ζ, t_ε and t_r.

We can confirm from Table 1 that $\max_{1\le i\le n} r_i$ is similar to ε, and $\text{mean } r_i$ and $\min_{1\le i\le n} r_i$ are smaller than ε. This relation coincides Theorem 4. Moreover $\max_{1\le i\le n} r_i$ and ε are smaller than $\min_{1\le i\le n} \zeta_i$ in this example. It can be seen from Table 2 that t_r is approximately equal to t_ε and a half of t_ζ. This relation coincides the fact that the computational costs for Algorithm 1 is $148n^3$ flops and those for Algorithms 2 and 3 are $66n^3$ flops. Additionally t_ε and t_r are smaller than t_λ in this example.

4.2. Example 2

In this example, we observe how the magnitudes of error bounds change when $\kappa(A)$ increases. Consider the case when 500×500 complex matrices A and B are generated by the following MATLAB code:

```
cndA10 = log10(cndA); % cndA: anticipated condition number of A
D = diag(logspace(0,cndA10,500));
[U,S,V] = svd(randn(500) + i*randn(500));
A = U*D*V';
B = randn(500) + i*randn(500);
```

If U and V are *strictly* unitary, and A is computed without rounding errors, then $\kappa(A) =$ cndA holds. In the practical generation, however, U and V are *approximately* unitary and the rounding errors are included in the computation. Hence, strictly speaking, $\kappa(A)$ and cndA are not equal but approximately equal. Table 3 displays similar quantities to Table 1 for various cndA.

It can be seen from Table 3 that error bounds increase as cndA increases. We can confirm the similar relations to Table 1 between $\min_{1\le i\le n} \zeta_i$, ε, $\max_{1\le i\le n} r_i$, $\text{mean } r_i$ and $\min_{1\le i\le n} r_i$.

Table 4
Obtained error bounds in Example 3.

cndB	max ζ_i	mean ζ_i	min ζ_i	ε	max r_i	mean r_i	min r_i
1e+04.	5.72e−05	1.08e−05	3.70e−07	8.69e−07	8.69e−07	1.05e−07	4.22e−10
1e+06.	2.45e−03	3.30e−04	3.47e−07	2.82e−05	2.82e−05	2.47e−06	2.08e−10
1e+08.	1.07e−01	1.38e−02	2.78e−07	1.09e−03	1.09e−03	9.27e−05	1.48e−10
1e+10.	9.38e+00	7.01e−01	3.58e−07	6.05e−02	6.05e−02	4.44e−03	1.25e−10
1e+12.	5.95e+02	4.71e+01	3.99e−07	–	–	–	–

When cndA = 1e+12, we applied the INTLAB [16] function `verifyeig` to $(\tilde{\lambda}_j, \tilde{x}^{(j)})$, $j \in \{1, 2, 3, 498, 499, 500\}$. Then enclosure succeeded for $\tilde{\lambda}_1$, $\tilde{\lambda}_2$ and $\tilde{\lambda}_3$, and failed for $\tilde{\lambda}_{498}$, $\tilde{\lambda}_{499}$ and $\tilde{\lambda}_{500}$. The obtained error bounds for $\tilde{\lambda}_1$, $\tilde{\lambda}_2$ and $\tilde{\lambda}_3$ were approximately 10^2 times smaller than mean r_i. Computing time for `verifyeig` was 19.7 s for all j. On the other hand, Algorithm 3 required 4.03 s for enclosing *all eigenvalues*.

4.3. Example 3

In this example, we observe how the magnitudes of error bounds change when $\kappa(B)$ increases. Consider the case when 500×500 complex matrices A and B are generated by the following MATLAB code:

```
A = randn(500) + i*randn(500);
cndB10 = log10(cndB); % cndB: anticipated condition number of B
D = diag(logspace(0,cndB10,500));
[U,S,V] = svd(randn(500) + i*randn(500));
B = U*D*V';
```

From the similar reason in Section 4.2, $\kappa(B) \approx$ cndB holds approximately. Table 4 displays the similar quantities to Table 1 for various cndB. In Table 4, the notation "–" means that the upper bound for $\|S\|_\infty$ became larger than 1 so that Algorithms 2 and 3 failed, where S is defined as in Theorem 2.

We can confirm from Table 4 that $\min_{1\le i\le n} \zeta_i$ and $\min_{1\le i\le n} r_i$ do not increase even though cndB increase. Consequently $\min_{1\le i\le n} r_i$ is about 10^8 times smaller than ε when cndB = 1e+10. Moreover $\max_{1\le i\le n} r_i$, mean r_i and $\min_{1\le i\le n} r_i$ are smaller than $\max_{1\le i\le n} \zeta_i$, mean ζ_i and $\min_{1\le i\le n} \zeta_i$, respectively, except the case when cndB = 1e+12. Algorithms 2 and 3 failed when cndB = 1e+12, although Algorithm 1 succeeded. From this, it can be seen that Algorithm 1 is robuster than Algorithms 2 and 3 when $\kappa(B)$ is large.

4.4. Example 4

In this example, we observe the magnitudes of error bounds and computing times for matrices in Matrix Market [13]. Table 5 shows names, n, $\kappa(A)$ and $\kappa(B)$ of the matrices being used. In Table 5, $\kappa(A)$ and $\kappa(B)$ are approximations obtained by the MATLAB function `cond` except $\kappa(A)$ in DGW961.[1] For various matrices, Tables 6 and 7 display the similar quantities to Tables 1 and 2, respectively.

From Table 6, we see the similar relations to Table 1 between ε, $\max_{1\le i\le n} r_i$, mean r_i and $\min_{1\le i\le n} r_i$. Similarly to Table 4, $\max_{1\le i\le n} r_i$, mean r_i and $\min_{1\le i\le n} r_i$ are smaller than $\max_{1\le i\le n} \zeta_i$, mean ζ_i and $\min_{1\le i\le n} \zeta_i$, respectively, except the case when the matrices are BFW782. Table 7 shows the similar relations to Table 2 between t_ζ, t_ε and t_r.

4.5. Example 5

In this example, we observe the property of Algorithms 1, 2, and 3 when there exist multiple eigenvalues. Consider the case when A and B are defined as follows:

$$A := \begin{pmatrix} -30 & 6 & 9 \\ -30 & 6 & 9 \\ -170 & 34 & 51 \end{pmatrix}, \qquad B := \begin{pmatrix} 2 & -1 & 5 \\ 1 & 0 & 2 \\ 1 & 5 & -4 \end{pmatrix}.$$

This example was discussed in [6]. In this case, $\lambda \in \{0, 1\}$ and both of algebraic and geometric multiplicities for $\lambda = 0$ are two. Table 8 displays $\tilde{\mu}_i$, ζ_i, $\tilde{\lambda}_i$, ε and r_i for $i = 1, 2, 3$.

We can confirm from Table 8 that Algorithm 3 could enclose all eigenvalues even if there exist multiple eigenvalues, although Algorithm 3 cannot check whether there exist multiple eigenvalues or closely clustered eigenvalues. It can be seen that r_i is smaller than ζ_i for all i in this example.

[1] The matrix A in DGW961 is singular since its 706, . . . , 961-th column vectors are zero vectors.

Table 5
Properties of matrices.

Name	n	$\kappa(A)$	$\kappa(B)$
BFW62	62	5.53e+02	1.72e+01
BFW398	398	2.99e+03	2.11e+01
BFW782	782	1.74e+03	1.81e+01
DGW961	961	∞	1.12e+07
LUND	147	2.80e+06	3.00e+04
RBS480	480	1.04e+04	1.20e+04

Table 6
Obtained error bounds in Example 4.

Matrices	max ζ_i	mean ζ_i	min ζ_i	ε	max r_i	mean r_i	min r_i
BFW62	1.06e−07	3.90e−08	1.02e−08	5.49e−08	5.49e−08	7.84e−09	1.03e−09
BFW398	3.83e−05	7.19e−06	8.77e−07	1.43e−05	1.43e−05	5.91e−07	4.05e−08
BFW782	3.50e−04	4.18e−05	4.13e−06	9.75e−04	9.75e−04	4.71e−06	8.15e−08
DGW961	2.97e−02	2.73e−03	1.58e−06	1.45e−02	1.45e−02	2.68e−04	1.45e−08
LUND	1.22e−05	4.57e−06	3.40e−07	3.53e−07	3.53e−07	6.58e−08	2.71e−08
RBS480	9.57e−07	2.37e−07	1.39e−08	3.89e−09	3.89e−09	1.78e−10	6.49e−12

Table 7
Computing times (s) in Example 4.

Matrices	t_λ	$t_{\lambda x}$	t_ζ	t_ε	t_r
BFW62	0.00813	0.0108	0.0456	0.0289	0.0282
BFW398	1.15	1.94	4.08	2.10	2.09
BFW782	9.38	17.4	28.5	14.5	14.4
DGW961	36.9	55.3	58.0	27.2	27.1
LUND	0.0103	0.0298	0.214	0.106	0.103
RBS480	2.36	4.35	7.03	3.64	3.60

Table 8
Approximate eigenvalues and obtained error bounds in Example 5.

$\tilde{\mu}_1$	$\tilde{\mu}_2$	$\tilde{\mu}_3$	ζ_1	ζ_2	ζ_3	$\tilde{\lambda}_1$	$\tilde{\lambda}_2$	$\tilde{\lambda}_3$	ε	r_1	r_2	r_3
4.42e−13	1.00e−00	−1.59e−14	2.39e−11	1.64e−11	9.01e−12	−2.08e−13	1.00e−00	6.38e−14	8.32e−12	6.16e−12	8.32e−12	2.07e−12

5. Conclusion

In this paper, we presented Theorem 3 for enclosing all eigenvalues in (1). Theorem 3 is applicable even if A is not Hermitian and/or B is not Hermitian positive definite, and supplies n *error bounds* $r_1, \ldots, r_n$ while Theorem 2 supplies *a single error bound* ε. We presented Theorem 4 showing $\max_{1\le i\le n} r_i \le \varepsilon$, proposed Algorithm 3 based on Theorem 3, and reported numerical results to show the property of Algorithm 3.

Similarly to Algorithm 2, Algorithm 3 can be applied to enclose all eigenvalues in polynomial eigenvalue problem (see [6] for some details). By modifying Algorithm 3 slightly, moreover, enclosure for all eigenvalues in (1) where A and/or B are complex interval matrices is also possible. Our future work will be to construct an algorithm for enclosing eigenvalues in (1) which is applicable even when B is singular.

Acknowledgments

This research was partially supported by Grants-in-Aid for Young Scientists (B) (19760055, 2007–2010) and Scientific Research (C) (23560066, 2011–2015) from the Ministry of Education, Science, Sports and Culture of Japan.

References

[1] F. Chatelin, Eigenvalues of Matrices, John Wiley & Sons, 1993.
[2] W. Kerner, Large-scale complex eigenvalue problems, J. Comput. Phys. 85 (1989) 1–85.
[3] H. Behnke, Inclusion of eigenvalues of general eigenvalue problems for matrices, in: U. Kulisch, H.J. Stetter (Eds.), Scientific Computation with Automatic Result Verification, Computing 6 (Suppl.) (1988), 69–78.
[4] H. Behnke, The calculation of guaranteed bounds for eigenvalues using complementary variational principles, Computing 47 (1991) 11–27.
[5] K. Maruyama, T. Ogita, Y. Nakaya, S. Oishi, Numerical inclusion method for all eigenvalues of real symmetric definite generalized eigenvalue problem, IEICE Trans. J87-A (8) (2004) 1111–1119 (in Japanese).
[6] S. Miyajima, Fast enclosure for all eigenvalues in generalized eigenvalue problems, J. Comput. Appl. Math. 233 (2010) 2994–3004.

[7] S. Miyajima, T. Ogita, S.M. Rump, S. Oishi, Fast verification for all eigenpairs in symmetric positive definite generalized eigenvalue problems, Reliab. Comput. 14 (2010) 24–45.
[8] S.M. Rump, Guaranteed inclusions for the complex generalized eigenproblem, Computing 42 (1989) 225–238.
[9] S.M. Rump, Computational error bounds for multiple or nearly multiple eigenvalues, Linear Algebra Appl. 324 (2001) 209–226.
[10] Y. Watanabe, N. Yamamoto, M.T. Nakao, Verification methods of generalized eigenvalue problems and its applications, Trans. JSIAM 9 (3) (1999) 137–150 (in Japanese).
[11] N. Yamamoto, A simple method for error bounds of eigenvalues of symmetric matrices, Linear Algebra Appl. 324 (2001) 227–234.
[12] B.N. Parlett, The symmetric eigenvalue problem, in: Classics in Applied Mathematics, vol. 20, SIAM Publications, Philadelphia, 1997.
[13] Matrix Market. http://math.nist.gov/MatrixMarket/ (last change: 10.05.07).
[14] G.H. Golub, C.F. Van Loan, Matrix Computations, third ed., The Johns Hopkins University Press, Baltimore, London, 1996.
[15] C.D. Meyer, Matrix Analysis and Applied Linear Algebra, SIAM Publications, Philadelphia, 2000.
[16] S.M. Rump, Intlab—interval laboratory, in: T. Csendes (Ed.), Developments in Reliable Computing, Kluwer Academic Publishers, Dordrecht, 1999, pp. 77–107.