

IMPROVING BACKWARD STABILITY OF
SAKURAI-SUGIURA METHOD WITH BALANCING TECHNIQUE
IN POLYNOMIAL EIGENVALUE PROBLEM

HONGJIA CHEN, AKIRA IMAKURA, TETSUYA SAKURAI, Tsukuba

Received January 29, 2017. First published July 5, 2017.

Abstract. One of the most efficient methods for solving the polynomial eigenvalue problem (PEP) is the Sakurai-Sugiura method with Rayleigh-Ritz projection (SS-RR), which finds the eigenvalues contained in a certain domain using the contour integral. The SS-RR method converts the original PEP to a small projected PEP using the Rayleigh-Ritz projection. However, the SS-RR method suffers from backward instability when the norms of the coefficient matrices of the projected PEP vary widely. To improve the backward stability of the SS-RR method, we combine it with a balancing technique for solving a small projected PEP. We then analyze the backward stability of the SS-RR method. Several numerical examples demonstrate that the SS-RR method with the balancing technique reduces the backward error of eigenpairs of PEP.

Keywords: SS-RR method; polynomial eigenvalue problem; balancing technique

MSC 2010: 65F15, 15A18

1. INTRODUCTION

In this paper, we consider the polynomial eigenvalue problem (PEP)

$$(1.1) \quad P(\lambda)\mathbf{x} = \left(\sum_{i=0}^m \lambda^i A_i \right) \mathbf{x} = \mathbf{0},$$

where $A_i \in \mathbb{C}^{n \times n} \setminus \{O\}$, $\lambda \in \mathbb{C}$ and $\mathbf{x} \in \mathbb{C}^n \setminus \{\mathbf{0}\}$ are eigenvalues and their associated eigenvectors.

This research was supported partly by JST/ACT-I (No. JPMJPR16U6), JST/CREST, KAKENHI (No. 17K12690) and University of Tsukuba Basic Research Support Program Type A.

A number of problems that arise in science and engineering involve the PEP, such as oscillation analysis of structural mechanics, and acoustic systems in electrical circuit simulation [13].

In some applications, such as structural dynamics and structural-acoustic interaction, it is unnecessary to compute all eigenpairs, and partial eigenpairs $(\lambda, \boldsymbol{x})$ are sufficient.

The Sakurai-Sugiura (SS) method has been proposed to efficiently compute partial eigenpairs for the generalized eigenvalue problem (GEP) [11]. The SS method projects the matrix pencil onto a subspace associated with the eigenvalues that are located in a domain via numerical integration. A number of extensions of the SS method are available for solving the GEP, including the SS-Hankel method [11], [8] and the SS method with the Rayleigh-Ritz projection (SS-RR) [7]. The SS-Hankel method transforms the original GEP to a small GEP with Hankel matrices. The SS-RR method, which is based on the Rayleigh-Ritz projection, projects the GEP onto a projected matrix pencil. The SS methods [14], [1], [2] have also been extended to the nonlinear eigenvalue problem (NEP).

The SS-RR method extracts only the eigenvalues within a Jordan curve Γ , using a subspace constructed with a contour integral. In the SS-RR method for the PEP [14], $P(\lambda)$ is transformed into a projection of a matrix polynomial with a small dimension as

$$(1.2) \quad R(\lambda) = V^H P(\lambda) V,$$

where the matrix $V \in \mathbb{C}^{n \times l}$, $l \ll n$, has orthonormal columns consisting of basis vectors for the subspace constructed by the SS-RR method. Then, the pair $(\widehat{\lambda}, V\widehat{\boldsymbol{y}})$ is used as an approximate eigenpair for $P(\lambda)$, where $(\widehat{\lambda}, \widehat{\boldsymbol{y}})$ is an approximate eigenpair for $R(\lambda)$. However, the SS-RR method can suffer from backward instability when the coefficient matrices of $R(\lambda)$ vary widely in their norm. Recently, we have improved the backward stability of the SS-RR method for the quadratic eigenvalue problem (QEP) [4]. Extending this idea, we improve the backward stability of the SS-RR method for the PEP using balancing technique.

One common way for solving (1.2) is to convert $R(\lambda)$ into a GEP with the same spectrum as $R(\lambda)$ and solve the GEP. In this article, to improve the accuracy of computing eigenpairs, we consider using a balancing technique [9], [10], that is a pre-processing technique for improving accuracy of computing eigenpairs in the standard eigenvalue problem (SEP). To allow the use of the balancing technique, we transform the GEP into the SEP. We also explain why the use of a stable eigensolver for the SEP, such as the QR method with a balancing technique, can improve the backward stability of the SS-RR method. To achieve this goal, we need to find relations

between the backward error of the SS-RR method and that of the SS-RR method with the balancing technique. We found that the SS-RR method with the balancing technique improves the accuracy of computing eigenpairs under some assumptions.

The remainder of this paper is organized as follows. In the next section, we review the SS-RR method for the PEP. In Section 3, we introduce a linearization for solving the projected PEP. In Section 4, we provide a brief description of the balancing technique and present an algorithm for the SS-RR method with the balancing technique. Then we investigate why the SS-RR method with the balancing technique improves the accuracy for computing eigenpairs. In Section 5, we present numerical experiments that confirm the accuracy of the SS-RR method with the balancing technique. Finally, conclusions and suggestions for future studies are presented in Section 6.

The following notation is used in this paper: $V = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_L] \in \mathbb{C}^{n \times L}$ and $\mathcal{R}(V) := \text{span}\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_L\}$, where $\mathcal{R}(V)$ is the range space of the matrix V .

2. THE SS-RR METHOD FOR THE PEP

The SS-RR method computes the eigenvalues that are located inside a Jordan curve Γ . Let $K, L \in \mathbb{N}^+$ be input parameters and $U \in \mathbb{C}^{n \times L}$ the input matrix with $KL < n$. We define

$$S = [S_0, \dots, S_{K-1}] \in \mathbb{C}^{n \times KL}$$

and

$$(2.1) \quad S_k = \frac{1}{2\pi i} \int_{\Gamma} g_k(z) P(z)^{-1} U \, dz \in \mathbb{C}^{n \times L},$$

where g_k is a k -th degree polynomial function. Since the target eigenvectors are in $\mathcal{R}\{S\}$, the target eigenpairs can be computed using the Rayleigh-Ritz procedure with $\mathcal{R}\{S\}$, see [14].

In numerical calculations, we use a numerical quadrature to approximate the contour integral (2.1). The approximation of S_k is given by

$$(2.2) \quad S_k \approx \widehat{S}_k = \sum_{p=1}^N \omega_p g_k(z_p) P(z_p)^{-1} U,$$

where z_p and ω_p , $p = 1, \dots, N$, are the integral points and their associated weights.

We construct $\widehat{S} = [\widehat{S}_0, \dots, \widehat{S}_{K-1}]$ and compute a low-rank approximation of \widehat{S} by singular value decomposition as

$$\widehat{S} = \widehat{V} \widehat{\Sigma} \widehat{W}^H \approx V \Sigma W^H,$$

where $V = \widehat{V}(:, 1 : l)$, and l is the numerical rank of \widehat{S} . Then we convert the original problem $P(\lambda)$ to $R(\lambda) = V^H P(\lambda) V$ and compute all eigenpairs of $R(\lambda)$, where the dimension of $R(\lambda)$ is l . Let the computed eigenpairs of $R(\lambda)$ be denoted by $(\widehat{\lambda}_j, \widehat{\mathbf{y}}_j)$, where $\widehat{\mathbf{y}}_j \in \mathbb{C}^l$. Then the eigenpairs $(\widehat{\lambda}_j, \widehat{\mathbf{x}}_j)$ of $P(\lambda)$ are approximated by

$$(\widehat{\lambda}_j, \widehat{\mathbf{x}}_j) = (\widehat{\lambda}_j, V \widehat{\mathbf{y}}_j), \quad j = 1, \dots, n(\Gamma),$$

where $n(\Gamma) \leq l$ is the number of approximate eigenvalues in the target region Ω .

The main steps of the SS-RR method are presented in Algorithm 1.

Algorithm 1. The SS-RR method

Input: $N, K, L \in \mathbb{N}^+$, $U \in \mathbb{C}^{n \times L}$, $z_p, \omega_p, p = 1, \dots, N$. A Jordan curve Γ and a matrix polynomial $P(\lambda)$.

Output: $\widehat{\lambda}_j, \widehat{\mathbf{x}}_j, j = 1, \dots, n(\Gamma)$, where $n(\Gamma)$ is the number of eigenvalues inside the Jordan curve.

- 1: Compute $P(z_p)^{-1}U, p = 1, \dots, N$.
 - 2: Compute $\widehat{S}_k, k = 0, \dots, K - 1$ by (2.2).
 - 3: Construct $\widehat{S} = [\widehat{S}_0, \dots, \widehat{S}_{K-1}]$ and perform the singular value decomposition of \widehat{S} , where $\widehat{S} = \widehat{V} \widehat{\Sigma} \widehat{W}^H$.
 - 4: Construct the orthonormal basis V from $\widehat{V}(:, 1 : l)$, where l is the numerical rank.
 - 5: Form $R(\lambda) = V^H P(\lambda) V$.
 - 6: Compute eigenvalues $\widehat{\lambda}_j$ and eigenvectors $\widehat{\mathbf{y}}_j$ of $R(\lambda), j = 1, \dots, n(\Gamma)$.
 - 7: Set $\widehat{\mathbf{x}}_j = V \widehat{\mathbf{y}}_j, j = 1, \dots, n(\Gamma)$.
-

3. EIGENSOLVER FOR THE PROJECTED PEP USING LINEARIZATION

We now discuss why the numerical solution of the PEP in Step 6 of Algorithm 1 requires special attention. In the SS-RR method, the standard way to solve small to medium size PEP,

$$(3.1) \quad R(\lambda) \mathbf{y} = \left(\sum_{i=0}^m \lambda^i R_i \right) \mathbf{y} = \mathbf{0}, \quad R_i = V^H A_i V,$$

is via linearization. We assume $R_i \in \mathbb{C}^{l \times l} \setminus \{O\}$. We linearize (3.1) as follows:

$$L(\lambda) \mathbf{z} = (\lambda X + Y) \mathbf{z} = \mathbf{0},$$

where $X, Y \in \mathbb{C}^{ml \times ml}$, $L(\lambda)$ and $R(\lambda)$ have the same spectrum. There are several choices for $L(\lambda)$. In practice, a common choice for $L(\lambda)$ is its companion form, which

is given by

$$(3.2) \quad L(\lambda) = \lambda \begin{bmatrix} R_1 & R_2 & \dots & R_m \\ -I_l & O & \dots & O \\ \vdots & \ddots & \ddots & \vdots \\ O & \dots & -I_l & O \end{bmatrix} + \begin{bmatrix} R_0 & O & \dots & O \\ O & I_l & \dots & O \\ \vdots & \vdots & \ddots & \vdots \\ O & O & \dots & I_l \end{bmatrix}, \quad \mathbf{z} = \begin{bmatrix} \mathbf{y} \\ \lambda \mathbf{y} \\ \vdots \\ \lambda^{m-1} \mathbf{y} \end{bmatrix},$$

where $I_l \in \mathbb{R}^{l \times l}$ is the identity matrix. We compute all eigenpairs of $L(\lambda)$ by using the QZ algorithm. Finally, we recover the eigenvectors $V\mathbf{y}$ of $P(\lambda)$ from the eigenvectors \mathbf{z} of $L(\lambda)$.

The QZ algorithm is backward stable for the GEP; however, it can be backward unstable for the PEP, especially when the norms of the coefficient matrices of $R(\lambda)$ vary widely [12].

4. THE SS-RR METHOD WITH THE BALANCING TECHNIQUE FOR THE PEP

As shown in Section 2, the SS-RR method extracts only the eigenvalues within a Jordan curve Γ . However, the SS-RR method is not stable when the coefficient matrices of the projected PEP have widely varying norms.

From [6], it is clear that if the backward error of $L(\lambda)$ is reduced, then the backward error of quadratic matrix polynomial is reduced as well. To reduce the backward error of $L(\lambda)$, in this section we convert the GEP $L(\lambda)$ to an SEP and solve it using the QR method with the balancing technique that improves the backward stability of the SEP.

4.1. Balancing technique for the standard eigenvalue problem. In this section, we review the balancing techniques.

The balancing technique is a preprocessing step to improve the accuracy for solving the SEP,

$$(4.1) \quad A\mathbf{v} = \lambda\mathbf{v}, \quad A \in \mathbb{C}^{n \times n}.$$

The main idea of the balancing technique is to minimize the norm of $D^{-1}AD$ with a similarity transformation using a diagonal matrix D .

Osborne proposed the use of a diagonal matrix D that minimizes the Frobenius norm $\|D^{-1}AD\|_F$, see [9]. He showed that his technique also decreases the 2-norm, that is, $\|A\|_2 \geq \|D^{-1}AD\|_2$, see [9]. Parlett and Reinsch extended Osborne's technique to any p -norm [10].

The Parlett-Reinsch algorithm operates on columns and rows of A in a cyclic fashion. Let c_i and r_i be the p -norms of each column and row, which ignore the diagonal element of the matrix A defined by

$$c_i = \left(\sum_{j \neq i} |a_{j,i}|^p \right)^{1/p}, \quad r_i = \left(\sum_{j \neq i} |a_{i,j}|^p \right)^{1/p}.$$

The norm of $D^{-1}AD$ can be reduced when the norms of the columns and rows are equal. The Parlett-Reinsch algorithm seeks f_i to minimize

$$g(f_i) = f_i^p c_i^p + \frac{r_i^p}{f_i^p},$$

and finds an approximation of the exact value f_i that minimizes $g(f_i)$.

The main steps of the Parlett-Reinsch algorithm are summarized in Algorithm 2. The diagonal elements of the matrix D are obtained from the value f_i by Step 15 in Algorithm 2.

Algorithm 2. Parlett-Reinsch algorithm (Balancing technique) [10]

Input: A matrix $A \in \mathbb{C}^{n \times n}$, κ is a radix base.

Output: A diagonal matrix D and \tilde{A} that is overwritten by $D^{-1}AD$.

```

1:  $D \leftarrow I$ 
2:  $converged \leftarrow 0$ 
3: while  $converged = 0$  do
4:    $converged = 1$ ;
5:   for  $i \leftarrow 1, \dots, n$  do
6:      $c \leftarrow \left( \sum_{j \neq i} |a_{j,i}|^p \right)^{1/p}$ ,  $r \leftarrow \left( \sum_{j \neq i} |a_{i,j}|^p \right)^{1/p}$ 
7:      $s \leftarrow c^p + r^p$ ,  $f \leftarrow 1$ 
8:     while  $c < r/\kappa$  do
9:        $c \leftarrow c\kappa$ ,  $r = r/\kappa$ ,  $f \leftarrow f \times \kappa$ 
10:    end while
11:    while  $c \geq r\kappa$  do
12:       $c \leftarrow c/\kappa$ ,  $r = r\kappa$ ,  $f \leftarrow f/\kappa$ 
13:    end while
14:    if  $(c^p + r^p) < 0.95 \times s$  then
15:       $converged \leftarrow 0$ ,  $d_{ii} \leftarrow f \times d_{ii}$ 
16:       $A(:, i) \leftarrow f \times A(:, i)$ ,  $A(:, i) \leftarrow A(:, i)/f$ 
17:    end if
18:  end for
19: end while

```

4.2. The SS-RR method with the balancing technique. In the SS-RR method, we transform $L(\lambda)$ to an SEP and apply the balancing technique to the SEP with the nonsingular diagonal matrix D such that

$$(4.2) \quad D^{-1}(-X^{-1}Y)D\mathbf{v} = \lambda\mathbf{v}.$$

Finally, we compute the eigenpairs of (4.2) with a backward stable method, such as the QR method. The eigenpairs of $P(\lambda)$ are recovered from (4.2).

The SS-RR method with the balancing technique is presented in Algorithm 3.

Algorithm 3. The SS-RR method with the balancing technique

Input: $N, K, L \in \mathbb{N}^+, U \in \mathbb{C}^{n \times L}, z_p, \omega_p, p = 1, \dots, N$. A Jordan curve Γ , and a matrix polynomial $P(\lambda)$.

Output: $\tilde{\lambda}_j, \tilde{\mathbf{x}}_j, j = 1, \dots, n(\Gamma)$, where $n(\Gamma)$ is the number of eigenvalues inside the Jordan curve.

- 1: Form $R(\lambda) = V^H P(\lambda) V$ by step 1-5 in Algorithm 1.
 - 2: Convert the projected matrix polynomial $R(\lambda)$ to $L(\lambda)$.
 - 3: Construct the SEP by (4.2) and compute eigenpairs $(\tilde{\lambda}_j, \tilde{\mathbf{v}}_j)$ of (4.2).
 - 4: Compute eigenvalues $\tilde{\lambda}_j$ and eigenvectors $\tilde{\mathbf{z}}_j$ of $L(\lambda)$ from (4.2) by setting $\tilde{\mathbf{z}}_j = D\tilde{\mathbf{v}}_j, j = 1, \dots, n(\Gamma)$.
 - 5: Compute eigenvalues $\tilde{\lambda}_j$ and eigenvectors $\tilde{\mathbf{x}}_j$ of $P(\lambda)$ by setting $\tilde{\mathbf{x}}_j = V\tilde{\mathbf{y}}_j$, where $\tilde{\mathbf{y}}_j = \tilde{\mathbf{z}}_j(1:l), j = 1, \dots, n(\Gamma)$.
-

4.3. Analysis of the backward error for the SS-RR method with the balancing technique. For solving QEP, an improvement of the backward error of the SS-RR method using a backward stable QEP eigensolver has been proposed and analyzed in [4]. In this article, we extend the idea in [4] to solve the PEP. The analysis in [4] is only based on the relationship between backward errors of the original QEP and projected QEP. Instead, to analyze the backward stability of the proposed method (Algorithm 3), we additionally need to analyze the relationship between the backward error of the projected PEP and the linearized eigenvalue problems. In what follows, we analyze these relationships and provide a theory to explain why the use of a stable eigensolver for the SEP improves the backward stability of the SS-RR method.

In the SS-RR method, let $(\hat{\lambda}_j, \hat{\mathbf{z}}_j)$ and $(\tilde{\lambda}_j, \tilde{\mathbf{z}}_j)$ be the approximations of the same eigenpair $(\lambda_j, \mathbf{z}_j)$ of $L(\lambda)$, where $(\hat{\lambda}_j, \hat{\mathbf{z}}_j)$ are computed by $L(\lambda)$ without using the balancing technique and $(\tilde{\lambda}_j, \tilde{\mathbf{z}}_j)$ are computed with $D^{-1}(-X^{-1}Y)D\mathbf{v} = \lambda\mathbf{v}$, where $\tilde{\mathbf{z}}_j = D\tilde{\mathbf{v}}_j$.

From (3.2), we also define $\hat{\mathbf{y}}_j$ and $\tilde{\mathbf{y}}_j$ as

$$\hat{\mathbf{y}}_j = \hat{\mathbf{z}}_j(1:l), \quad \tilde{\mathbf{y}}_j = \tilde{\mathbf{z}}_j(1:l).$$

Here, we also assume that $\widehat{\mathbf{y}}_j, \widetilde{\mathbf{y}}_j$ are normalized, that is, $\|\widehat{\mathbf{y}}_j\|_2 = \|\widetilde{\mathbf{y}}_j\|_2 = 1$. Then $(\widehat{\lambda}_j, \widehat{\mathbf{y}}_j)$, $(\widetilde{\lambda}_j, \widetilde{\mathbf{y}}_j)$ are approximate eigenpairs of $R(\lambda)$ and $(\widehat{\lambda}_j, V\widehat{\mathbf{y}}_j)$, $(\widetilde{\lambda}_j, V\widetilde{\mathbf{y}}_j)$ are approximate eigenpairs of $P(\lambda)$.

To analyze the accuracy of the eigenpairs obtained by the SS-RR method with the balancing technique, we consider the backward error of the PEPs.

Definition 4.1 ([12]). Let $R(\lambda)$ be the matrix polynomial,

$$R(\lambda) = \sum_{i=0}^m \lambda^i R_i.$$

The backward error of the approximated eigenpairs $(\widehat{\lambda}_j, \widehat{\mathbf{y}}_j)$ of $R(\lambda)$ is given by

$$\eta(R, \widehat{\lambda}_j, \widehat{\mathbf{y}}_j) := \min\{\varepsilon : (R(\widehat{\lambda}_j) + \Delta R(\widehat{\lambda}_j))\widehat{\mathbf{y}}_j = \mathbf{0}, \|\Delta R_i\|_2 \leq \varepsilon \|R_i\|_2, i = 0, \dots, m\},$$

where $\Delta R(\lambda) = \sum_{i=0}^m \lambda^i \Delta R_i$, ΔR_i is a perturbation matrix.

An analogous definition holds for the backward error $\eta(L, \widehat{\lambda}_j, \widehat{\mathbf{z}}_j)$ of an approximate eigenpair $(\widehat{\lambda}_j, \widehat{\mathbf{z}}_j)$.

For computing the backward error numerically, explicit expressions for the backward error of $\eta(R, \widehat{\lambda}_j, \widehat{\mathbf{y}}_j)$ and $\eta(L, \widehat{\lambda}_j, \widehat{\mathbf{z}}_j)$ are given by the following formulas [6]:

$$(4.3) \quad \eta(R, \widehat{\lambda}_j, \widehat{\mathbf{y}}_j) = \frac{\|R(\widehat{\lambda}_j)\widehat{\mathbf{y}}_j\|_2}{(\sum_{i=0}^m |\widehat{\lambda}_j|^i \|V^H A_i V\|_2) \|\widehat{\mathbf{y}}_j\|_2},$$

$$(4.4) \quad \eta(L, \widehat{\lambda}_j, \widehat{\mathbf{z}}_j) = \frac{\|L(\widehat{\lambda}_j)\widehat{\mathbf{z}}_j\|_2}{(|\widehat{\lambda}_j| \|X\|_2 + \|Y\|_2) \|\widehat{\mathbf{z}}_j\|_2}.$$

The approximate eigenpair $(\widetilde{\lambda}_j, \widetilde{\mathbf{z}}_j)$ is computed with the balancing technique, therefore, we assume

$$(4.5) \quad \eta(L, \widehat{\lambda}_j, \widehat{\mathbf{z}}_j) \geq \eta(L, \widetilde{\lambda}_j, \widetilde{\mathbf{z}}_j).$$

In the following steps, we try to identify the sufficient conditions under which (4.5) implies that

$$(4.6) \quad \eta(P, \widehat{\lambda}_j, V\widehat{\mathbf{y}}_j) \geq \eta(P, \widetilde{\lambda}_j, V\widetilde{\mathbf{y}}_j).$$

By virtue of the relation of the backward errors for $R(\lambda)$ and $L(\lambda)$, $L(\lambda)$ satisfies a left-side factorization such that for some non-zero $\mathbf{g} \in \mathbb{C}^l$

$$G(\lambda)L(\lambda) = \mathbf{g}^T \otimes R(\lambda),$$

where \otimes denotes the Kronecker product [5]. If we use (3.2) to construct $L(\lambda)$, then $G(\lambda)$ is given by

$$G(\lambda) = \begin{cases} [I_l & -\lambda V^H A_2 V] & (m = 2) \\ [I_l & -\lambda(V^H A_2 V + V^H A_3 V) & -\lambda V^H A_3 V] & (m = 3). \end{cases}$$

When $m > 3$, we can obtain $G(\lambda)$ from [5]. Based on (4.3) and (4.4), to analyze the bounds for the backward error for $R(\lambda)$ relative to $L(\lambda)$, we have the following theorem.

Theorem 4.1 ([5]). *Let $(\widehat{\lambda}_j, \widehat{\mathbf{z}}_j)$ be an approximation of the eigenpair of $L(\lambda)$ and $(\widehat{\lambda}_j, \widehat{\mathbf{y}}_j)$ an approximation of the eigenpair of $R(\lambda)$, where $\widehat{\mathbf{y}}_j$ is obtained from $\widehat{\mathbf{z}}_j$ by (3.2) and is normalized so that $\|\widehat{\mathbf{y}}_j\|_2 = 1$. Then the bound for the backward error of $R(\lambda)$ relative to $L(\lambda)$ is*

$$(4.7) \quad \frac{\eta(R, \widehat{\lambda}_j, \widehat{\mathbf{y}}_j)}{\eta(L, \widehat{\lambda}_j, \widehat{\mathbf{z}}_j)} \leq C_U(\widehat{\lambda}_j, \widehat{\mathbf{z}}_j),$$

where

$$C_U(\widehat{\lambda}_j, \widehat{\mathbf{z}}_j) = \frac{(\|\widehat{\lambda}_j\| \|X\|_2 + \|Y\|_2) \|G(\widehat{\lambda}_j)\|_2}{\sum_{i=0}^m |\widehat{\lambda}_j|^i \|V^H A_i V\|_2} \|\widehat{\mathbf{z}}_j\|_2,$$

and $G(\widehat{\lambda}_j)$ is an $l \times lm$ matrix polynomial.

To analyze the bounds of the backward error of $P(\lambda)$ relative to $R(\lambda)$, we introduce the following lemma.

Lemma 4.1 ([4]). *Let $(\widehat{\lambda}_j, \widehat{\mathbf{y}}_j)$ be the approximate eigenpairs of $R(\lambda)$, where $R(\lambda) := V^H P(\lambda) V$, and V is an orthogonal matrix, $V^H V = I$. Let $(\widehat{\lambda}_j, V\widehat{\mathbf{y}}_j)$ be the approximated eigenpairs of $P(\lambda)$, $\|V\widehat{\mathbf{y}}_j\|_2 = 1$. Then we have*

$$B_L(\widehat{\lambda}_j) \leq \frac{\eta(P, \widehat{\lambda}_j, V\widehat{\mathbf{y}}_j)}{\eta(R, \widehat{\lambda}_j, \widehat{\mathbf{y}}_j)} \leq B_U(\widehat{\lambda}_j, \widehat{\mathbf{y}}_j),$$

where

$$B_L(\widehat{\lambda}_j) = \frac{\sum_{i=0}^m |\widehat{\lambda}_j|^i \|V^H A_i V\|_2}{\sum_{i=0}^m |\widehat{\lambda}_j|^i \|A_i\|_2}, \quad B_U(\widehat{\lambda}_j, \widehat{\mathbf{y}}_j) = \frac{\|P(\widehat{\lambda}_j) V\widehat{\mathbf{y}}_j\|_2}{\|V^H P(\widehat{\lambda}_j) V\widehat{\mathbf{y}}_j\|_2}$$

are functions that depend on the eigenpairs of the problem.

Proof. Based on (4.3), we have

$$\frac{\eta(P, \widehat{\lambda}_j, V\widehat{\mathbf{y}}_j)}{\eta(R, \widehat{\lambda}_j, \widehat{\mathbf{y}}_j)} = \frac{\|P(\widehat{\lambda}_j)V\widehat{\mathbf{y}}_j\|_2}{\left(\sum_{i=0}^m |\widehat{\lambda}_j|^i \|A_i\|_2\right) \|V\widehat{\mathbf{y}}_j\|_2} \frac{\left(\sum_{i=0}^m |\widehat{\lambda}_j|^i \|V^H A_i V\|_2\right) \|\widehat{\mathbf{y}}_j\|_2}{\|V^H P(\widehat{\lambda}_j)V\widehat{\mathbf{y}}_j\|_2}.$$

Because of $\|V^H P(\widehat{\lambda}_j)V\widehat{\mathbf{y}}_j\|_2 \leq \|V^H\|_2 \|P(\widehat{\lambda}_j)V\widehat{\mathbf{y}}_j\|_2$, we have

$$\frac{\eta(P, \widehat{\lambda}_j, V\widehat{\mathbf{y}}_j)}{\eta(R, \widehat{\lambda}_j, \widehat{\mathbf{y}}_j)} \geq \frac{\sum_{i=0}^m |\widehat{\lambda}_j|^i \|V^H A_i V\|_2}{\sum_{i=0}^m |\widehat{\lambda}_j|^i \|A_i\|_2} = B_L(\widehat{\lambda}_j),$$

and based on $\|V^H A_i V\|_2 \leq \|A_i\|_2$, we also have

$$\frac{\eta(P, \widehat{\lambda}_j, V\widehat{\mathbf{y}}_j)}{\eta(R, \widehat{\lambda}_j, \widehat{\mathbf{y}}_j)} \leq \frac{\|P(\widehat{\lambda}_j)V\widehat{\mathbf{y}}_j\|_2}{\|V^H P(\widehat{\lambda}_j)V\widehat{\mathbf{y}}_j\|_2} = B_U(\widehat{\lambda}_j, \widehat{\mathbf{y}}_j),$$

which proves Lemma 4.1. \square

Using Theorem 4.1 and Lemma 4.1, we have the following theorem.

Theorem 4.2. Let θ_j be a scalar value satisfying $\eta(L, \widehat{\lambda}_j, \widehat{\mathbf{z}}_j) = \theta_j \eta(L, \widetilde{\lambda}_j, \widetilde{\mathbf{z}}_j)$. Assume

$$(4.8) \quad \delta_j = \theta_j \alpha_j \beta_j \geq 1,$$

where

$$\alpha_j = B_L(\widehat{\lambda}_j) \frac{\eta(R, \widehat{\lambda}_j, \widehat{\mathbf{y}}_j)}{\eta(L, \widehat{\lambda}_j, \widehat{\mathbf{z}}_j)}, \quad \beta_j = \frac{1}{B_U(\widetilde{\lambda}_j, \widetilde{\mathbf{y}}_j) C_U(\widetilde{\lambda}_j, \widetilde{\mathbf{z}}_j)}.$$

Then we have

$$\eta(P, \widehat{\lambda}_j, V\widehat{\mathbf{y}}_j) \geq \eta(P, \widetilde{\lambda}_j, V\widetilde{\mathbf{y}}_j).$$

Proof. Based on Theorem 4.1, we have

$$\begin{aligned} \eta(R, \widehat{\lambda}_j, \widehat{\mathbf{y}}_j) &= \frac{\eta(R, \widehat{\lambda}_j, \widehat{\mathbf{y}}_j)}{\eta(L, \widehat{\lambda}_j, \widehat{\mathbf{z}}_j)} \eta(L, \widehat{\lambda}_j, \widehat{\mathbf{z}}_j) = \frac{\eta(R, \widehat{\lambda}_j, \widehat{\mathbf{y}}_j)}{\eta(L, \widehat{\lambda}_j, \widehat{\mathbf{z}}_j)} \theta_j \eta(L, \widetilde{\lambda}_j, \widetilde{\mathbf{z}}_j) \\ &\geq \frac{\eta(R, \widehat{\lambda}_j, \widehat{\mathbf{y}}_j)}{\eta(L, \widehat{\lambda}_j, \widehat{\mathbf{z}}_j)} \theta_j \frac{\eta(R, \widetilde{\lambda}_j, \widetilde{\mathbf{y}}_j)}{C_U(\widetilde{\lambda}_j, \widetilde{\mathbf{z}}_j)}. \end{aligned}$$

Lemma 4.1 yields

$$\begin{aligned}
\eta(P, \widehat{\lambda}_j, V\widehat{\mathbf{y}}_j) &\geq B_L(\widehat{\lambda}_j)\eta(R, \widehat{\lambda}_j, \widehat{\mathbf{y}}_j) \\
&\geq B_L(\widehat{\lambda}_j)\frac{\eta(R, \widehat{\lambda}_j, \widehat{\mathbf{y}}_j)}{\eta(L, \widehat{\lambda}_j, \widehat{\mathbf{z}}_j)}\frac{\theta_j}{C_U(\widetilde{\lambda}_j, \widetilde{\mathbf{z}}_j)}\eta(R, \widetilde{\lambda}_j, \widetilde{\mathbf{y}}_j) \\
&\geq B_L(\widehat{\lambda}_j)\frac{\eta(R, \widehat{\lambda}_j, \widehat{\mathbf{y}}_j)}{\eta(L, \widehat{\lambda}_j, \widehat{\mathbf{z}}_j)}\frac{\theta_j}{C_U(\widetilde{\lambda}_j, \widetilde{\mathbf{z}}_j)}\frac{\eta(P, \widetilde{\lambda}_j, V\widetilde{\mathbf{y}}_j)}{B_U(\widetilde{\lambda}_j, \widetilde{\mathbf{y}}_j)} \\
&= \theta_j\left(B_L(\widehat{\lambda}_j)\frac{\eta(R, \widehat{\lambda}_j, \widehat{\mathbf{y}}_j)}{\eta(L, \widehat{\lambda}_j, \widehat{\mathbf{z}}_j)}\right)\frac{1}{C_U(\widetilde{\lambda}_j, \widetilde{\mathbf{z}}_j)B_U(\widetilde{\lambda}_j, \widetilde{\mathbf{y}}_j)}\eta(P, \widetilde{\lambda}_j, V\widetilde{\mathbf{y}}_j) \\
&= \delta_j\eta(P, \widetilde{\lambda}_j, V\widetilde{\mathbf{y}}_j).
\end{aligned}$$

Therefore, from the assumption $\delta_j \geq 1$ we have

$$\eta(P, \widehat{\lambda}_j, V\widehat{\mathbf{y}}_j) \geq \eta(P, \widetilde{\lambda}_j, V\widetilde{\mathbf{y}}_j),$$

thus proving Theorem 4.2. \square

The computation of δ_j may be complicated, because it requires $\|A_i\|_2$. To determine a more efficient way of computing δ_j , we analyze (4.8) in detail.

Defining

$$\varepsilon_1 := \max_{i=0:m} \frac{\|A_i\|_2}{\|V^H A_i V\|_2},$$

we have $\|A_i\|_2 \leq \varepsilon_1 \|V^H A_i V\|_2$. Therefore, the lower bound for $B_L(\widehat{\lambda}_j)$ is given by

$$(4.9) \quad B_L(\widehat{\lambda}_j) = \frac{\sum_{i=0}^m |\widehat{\lambda}_j|^i \|V^H A_i V\|_2}{\sum_{i=0}^m |\widehat{\lambda}_j|^i \|A_i\|_2} \geq \frac{1}{\varepsilon_1}.$$

In this case, the lower bound for α_j is given by

$$(4.10) \quad \alpha_j = B_L(\widehat{\lambda}_j)\frac{\eta(R, \widehat{\lambda}_j, \widehat{\mathbf{y}}_j)}{\eta(L, \widehat{\lambda}_j, \widehat{\mathbf{z}}_j)} \geq \frac{1}{\varepsilon_1} \frac{\eta(R, \widehat{\lambda}_j, \widehat{\mathbf{y}}_j)}{\eta(L, \widehat{\lambda}_j, \widehat{\mathbf{z}}_j)}.$$

We also define

$$\varepsilon_2 := \max_j \frac{\|P(\widetilde{\lambda}_j)V\widetilde{\mathbf{y}}_j\|_2}{\|V^H P(\widetilde{\lambda}_j)V\widetilde{\mathbf{y}}_j\|_2} = \max_j B_U(\widetilde{\lambda}_j, \widetilde{\mathbf{y}}_j).$$

Then $\varepsilon_2 \geq B_U(\widetilde{\lambda}_j, \widetilde{\mathbf{y}}_j)$ and the lower bound of β_j is given by

$$(4.11) \quad \beta_j = \frac{1}{B_U(\widetilde{\lambda}_j, \widetilde{\mathbf{y}}_j)C_U(\widetilde{\lambda}_j, \widetilde{\mathbf{z}}_j)} \geq \frac{1}{\varepsilon_2 C_U(\widetilde{\lambda}_j, \widetilde{\mathbf{z}}_j)}.$$

Based on (4.5), (4.8), (4.10), and (4.11), the lower bound for δ_j is given by

$$(4.12) \quad \delta_j = \theta_j \alpha_j \beta_j \geq \frac{1}{\varepsilon_1 \varepsilon_2} \frac{\eta(R, \widehat{\lambda}_j, \widehat{\mathbf{y}}_j)}{\eta(L, \widehat{\lambda}_j, \widehat{\mathbf{z}}_j)} \frac{1}{C_U(\widetilde{\lambda}_j, \widetilde{\mathbf{z}}_j)}.$$

If the projection V does not significantly change the norms of the coefficient matrices of $P(\lambda)$, that is,

$$(4.13) \quad \|A_i\|_2 \approx \|V^H A_i V\|_2,$$

we have

$$(4.14) \quad \varepsilon_1 \approx 1.$$

Next, we analyze the parameter ε_2 . If $\mathcal{R}(V)$ is an invariant subspace with respect to $P(\widetilde{\lambda}_j)$, i.e., there is $Q(\widetilde{\lambda}_j)$ such that $P(\widetilde{\lambda}_j)V = VQ(\widetilde{\lambda}_j)$, then we have

$$\|P(\widetilde{\lambda}_j)V\widetilde{\mathbf{y}}_j\|_2 = \|VQ(\widetilde{\lambda}_j)\widetilde{\mathbf{y}}_j\|_2 = \|Q(\widetilde{\lambda}_j)\widetilde{\mathbf{y}}_j\|_2,$$

and

$$\|V^H P(\widetilde{\lambda}_j)V\widetilde{\mathbf{y}}_j\|_2 = \|V^H VQ(\widetilde{\lambda}_j)\widetilde{\mathbf{y}}_j\|_2 = \|Q(\widetilde{\lambda}_j)\widetilde{\mathbf{y}}_j\|_2.$$

Therefore, $\|P(\widetilde{\lambda}_j)V\widetilde{\mathbf{y}}_j\|_2 = \|V^H P(\widetilde{\lambda}_j)V\widetilde{\mathbf{y}}_j\|_2$. In the SS-RR method, V is constructed as an approximation of the invariant subspace with respect to the target eigenpairs. Based on this, we may assume

$$(4.15) \quad B_U(\widetilde{\lambda}_j, \widetilde{\mathbf{y}}_j) = \frac{\|P(\widetilde{\lambda}_j)V\widetilde{\mathbf{y}}_j\|_2}{\|V^H P(\widetilde{\lambda}_j)V\widetilde{\mathbf{y}}_j\|_2} \approx 1,$$

and thus ε_2 is close to 1. Using these assumptions, the lower bound for δ_j is given by

$$(4.16) \quad \delta_j = \theta_j \alpha_j \beta_j \gtrsim \frac{\eta(R, \widehat{\lambda}_j, \widehat{\mathbf{y}}_j)}{\eta(L, \widehat{\lambda}_j, \widehat{\mathbf{z}}_j) C_U(\widetilde{\lambda}_j, \widetilde{\mathbf{z}}_j)}.$$

Thus, if

$$(4.17) \quad \tau_j = \frac{\eta(R, \widehat{\lambda}_j, \widehat{\mathbf{y}}_j)}{\eta(L, \widehat{\lambda}_j, \widehat{\mathbf{z}}_j) C_U(\widetilde{\lambda}_j, \widetilde{\mathbf{z}}_j)} \geq 1,$$

we have

$$\eta(P, \widehat{\lambda}_j, V\widehat{\mathbf{y}}_j) \gtrsim \eta(P, \widetilde{\lambda}_j, V\widetilde{\mathbf{y}}_j).$$

The parameter τ_j in (4.17) can be computed with low cost, although it may sometimes happen that $\delta_j \geq 1 > \tau_j$.

5. NUMERICAL EXPERIMENTS

In this section, we compare the backward error of the SS-RR method in combination with the balancing technique (Algorithm 3) with the backward error of the standard implementation of the SS-RR method (Algorithm 1). For Algorithms 1 and 3, we use (3.2) to construct the linearized form of $R(\lambda)$. We use the MATLAB command `balance` to balance the coefficient matrix of the SEP in step 7 of Algorithm 3. The MATLAB command `balance` implements Algorithm 2.

The test problems (shown in Table 1) are PEP of degree $m = 2$ and higher-order PEP belonging to the collection of nonlinear eigenvalue problems NLEVP, see [3]. The problems `orr_sommerfeld` and `mod_butterfly` are higher-order PEP. Other problems are PEP of degree $m = 2$.

Problem	n	applications
<code>damped_beam</code>	400	A simple supported beam damped in the middle
<code>shaft</code>	400	The model of a shaft on bearing supports with a damper
<code>wiresaw1</code>	400	Vibration analysis of a wiresaw
<code>wiresaw2</code>	400	Vibration analysis of wiresaw with viscous damping
<code>sleeper</code>	400	Model of a railtrack resting on sleepers
<code>spring</code>	400	A finite element model of a damped mass-spring system
<code>dirac</code>	400	The Dirac operator
<code>acoustic_wave_1d</code>	400	Acoustic wave problem in 1 dimension
<code>plasma_drift</code>	128	Cubic PEP arising in tokamak reactor design
<code>orr_sommerfeld</code>	400	Arising from Orr-Sommerfeld equation
<code>mod_butterfly</code>	400	Quartic matrix polynomial with T-even structure

Table 1. Polynomial eigenvalue problems [3].

For each problem, the Jordan curve Γ is a circle with center γ and radius ϱ whose values are given in Table 2. We set $N = 32$, $K = 6$ and $L = 12$ for the problem `plasma_drift`. We use $N = 32$, $K = 8$, and $L = 16$ for the other problems. For the quadrature points and the corresponding weights, we assign

$$z_p = \gamma + \varrho \exp\left(\frac{2\pi i(p-1/2)}{N}\right), \quad \omega_p = \frac{z_p - \gamma}{N}, \quad p = 1, \dots, N.$$

All the computations were performed using MATLAB 2014.

5.1. Verification of the assumptions. Here we verify the assumptions of (4.14), (4.15), δ_j in (4.8) and τ_j in (4.17) by using numerical experiments.

Problem	center γ	radius ρ	#eigs
damped_beam	$-2 + 2.6 \times 10^6 i$	3×10^5	22
shaft	$2 \times 10^5 i$	9×10^4	18
wiresaw1	$-180i$	40	26
wiresaw2	$140i$	40	26
sleeper	-16	0.2	29
spring	-12	1	26
dirac	-5	0.7	24
acoustic_wave_1d	$-126 + 0.03i$	1	30
plasma_drift	10	1	10
mod_butterfly	$70i$	10	18
orr_sommerfeld	$3.8 \times 10^{-4} i$	0.4×10^{-4}	20

Table 2. Parameters for the SS-RR method.

Problem	ε_1
damped_beam	16.0
shaft	1.3
wiresaw1	4.0
wiresaw2	3.8
sleeper	1.0
spring	1.1
dirac	1.4
acoustic_wave_1d	17.0
plasma_drift	1.0
mod_butterfly	1.2
orr_sommerfeld	182.0

Table 3. The value of $\varepsilon_1 := \max_{i=0:m} \|A_i\|_2 / \|V^H A_i V\|_2$.

Problem	ε_2
damped_beam	27.0
shaft	1.3
wiresaw1	2.4
wiresaw2	1.1
sleeper	1.2
spring	1.6
dirac	5.0
acoustic_wave_1d	1.1
plasma_drift	1.0
mod_butterfly	1.3
orr_sommerfeld	64.0

Table 4. The value of $\varepsilon_2 := \max_j B_U(\tilde{\lambda}_j, \tilde{\mathbf{y}}_j)$.

As shown in Table 3, the norms of the coefficient matrices of $R(\lambda)$ are similar to that of $P(\lambda)$ for all problems except `damped_beam`, `acoustic_wave_1d` and `orr_sommerfeld`.

Table 4 shows the maximum values of $B_U(\tilde{\lambda}_j, \tilde{\mathbf{y}}_j)$. The value of $B_U(\tilde{\lambda}_j, \tilde{\mathbf{y}}_j)$ is not much larger than 1 in most problems. The only exceptions where the assumption (4.15) is not satisfied are `damped_beam` and `orr_sommerfeld`.

Table 5 displays the assumption that $\theta_j > 1$ is satisfied for all problems. Table 5 also shows that the assumption that $\delta_j > 1$ is satisfied for all problems. The

Problem	$\min \theta$	$\min \delta$	$\min \tau$
damped beam	7.1×10^4	6.9×10^5	2.4×10^2
shaft	7.2×10^3	5.7×10^7	6.6×10^3
wiresaw1	2.3×10^1	2.4×10^2	4.6×10^1
wiresaw2	2.0×10^1	3.1×10^2	3.9×10^1
sleeper	1.8×10^0	5.8×10^0	3.0×10^0
spring	1.5×10^0	2.9×10^0	1.3×10^0
dirac	4.1×10^0	2.0×10^0	1.6×10^0
acoustic_wave_1d	1.0×10^1	3.5×10^2	3.0×10^1
plasma_drift	1.0×10^0	4.1×10^0	3.9×10^0
mod_butterfly	2.1×10^0	9.9×10^1	6.3×10^1
orr_sommerfeld	3.6×10^8	3.6×10^2	1.3×10^{-4}

Table 5. The minimum value of the parameters δ , θ and τ for $P(\lambda)$.

more practical approximation τ_j for δ_j is also larger than 1 for all problems except `orr_sommerfeld`, which confirms its wide applicability.

5.2. Evaluation of the backward error of $P(\lambda)$. In this section we evaluate the backward errors of $P(\lambda)$ for the SS-RR method and the SS-RR method with the balancing technique. As shown in Table 6 and Figures 1–4, the backward errors of the SS-RR method with the balancing technique are smaller than those of the SS-RR method when τ is larger than 1. The improvement in the backward error is significant even for `orr_sommerfeld` in spite of the bad estimate for τ . We also find that there is almost no improvement in the `dirac` problem.

Problems	SS-RR	SS-RR method with balancing
damped_beam	3.7×10^{-7}	7.7×10^{-14}
shaft	3.4×10^{-10}	2.6×10^{-15}
wiresaw1	8.4×10^{-13}	6.0×10^{-15}
wiresaw2	4.4×10^{-13}	1.3×10^{-15}
sleeper	1.8×10^{-13}	5.3×10^{-15}
spring	7.1×10^{-14}	1.6×10^{-15}
dirac	3.4×10^{-15}	4.7×10^{-16}
acoustic_wave_1d	4.1×10^{-13}	7.7×10^{-15}
plasma_drift	4.7×10^{-13}	7.8×10^{-15}
mod_butterfly	1.1×10^{-9}	4.2×10^{-11}
orr_sommerfeld	1.8×10^{-6}	1.4×10^{-17}

Table 6. Maximum backward errors of the eigenpairs of $P(\lambda)$.

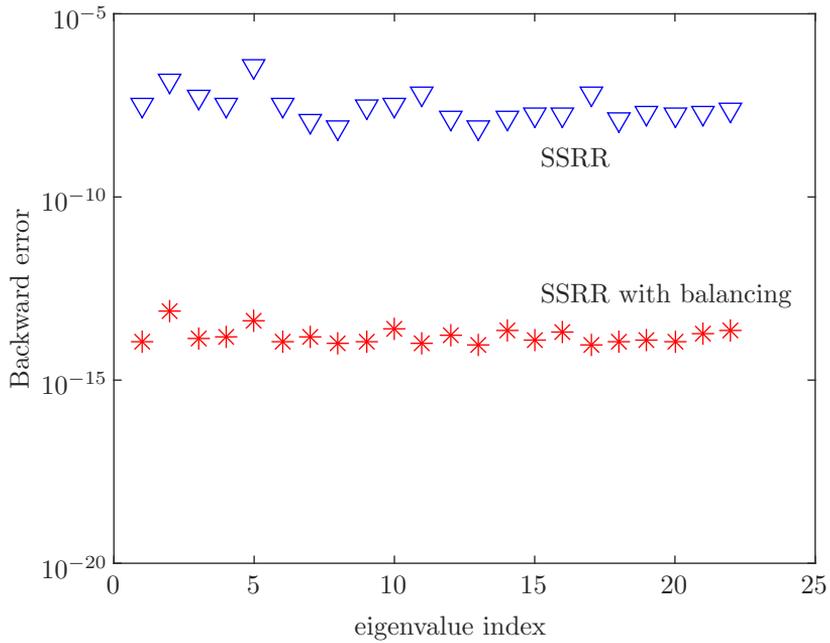


Figure 1. Backward error for the `damped_beam` problem.

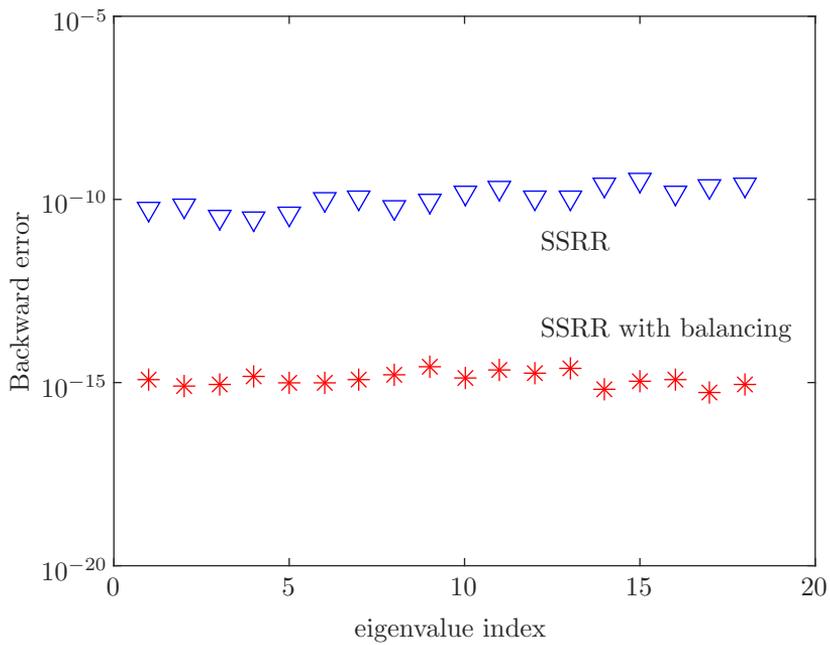


Figure 2. Backward error for the `shaft` problem.

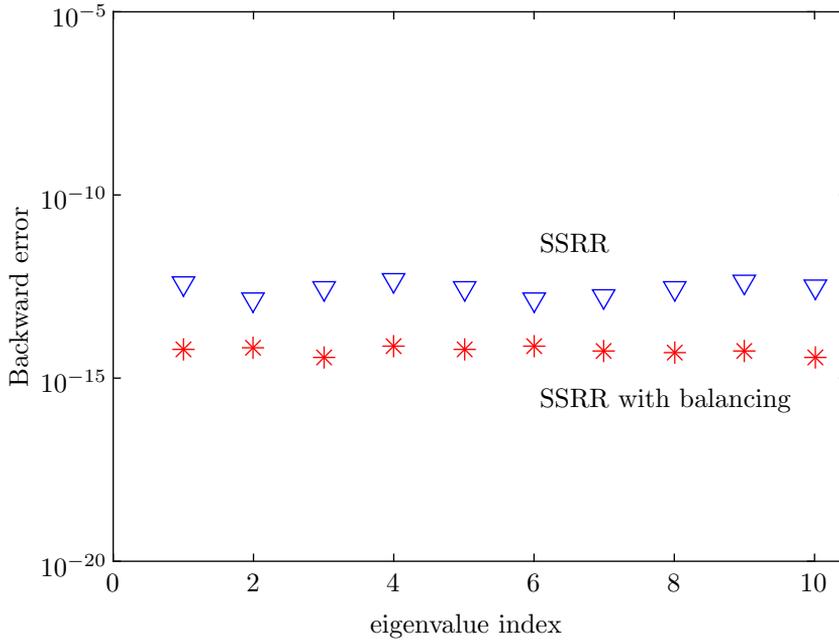


Figure 3. Backward error for the `plasma_drift` problem.

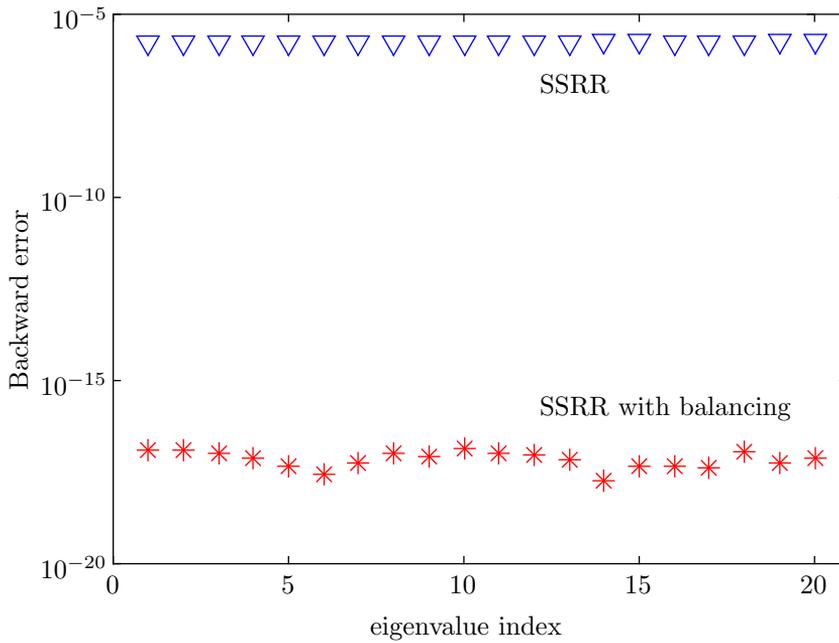


Figure 4. Backward error for the `orr_sommerfeld` problem.

Based on the experimental results, we find that the SS-RR method with the balancing technique can reduce the backward error of $P(\lambda)$.

6. CONCLUSION

We have proposed an approach for accurately computing the eigenpairs of the PEP using the SS-RR method with the balancing technique. In this paper we discussed why the SS-RR method with the balancing technique can improve the accuracy of computing eigenpairs and we found a relation between the backward error of the SS-RR method and that of the SS-RR method with the balancing technique. The analysis suggests that the SS-RR method with the balancing technique can reduce the backward error of the SS-RR method under certain conditions. In the numerical experiments, we found that these conditions are satisfied in most practical problems and the SS-RR method with the balancing technique is more accurate than the original SS-RR method. In our future investigations, we propose to study the results of combining the balancing technique with other types of SS methods.

References

- [1] *J. Asakura, T. Sakurai, H. Tadano, T. Ikegami, K. Kimura*: A numerical method for non-linear eigenvalue problems using contour integrals. *JSIAM Lett.* *1* (2009), 52–55. [zbl](#) [MR](#) [doi](#)
- [2] *J. Asakura, T. Sakurai, H. Tadano, T. Ikegami, K. Kimura*: A numerical method for polynomial eigenvalue problems using contour integral. *Japan J. Ind. Appl. Math.* *27* (2010), 73–90. [zbl](#) [MR](#) [doi](#)
- [3] *T. Betcke, N. J. Higham, V. Mehrmann, C. Schröder, F. Tisseur*: NLEVP: A collection of nonlinear eigenvalue problems. *ACM Trans. Math. Softw.* *39* (2013), Paper No. 7, 28 pages. [zbl](#) [MR](#) [doi](#)
- [4] *H. Chen, Y. Maeda, A. Imakura, T. Sakurai, F. Tisseur*: Improving the numerical stability of the Sakurai-Sugiura method for quadratic eigenvalue problems. *JSIAM Lett.* *9* (2017), 17–20. [doi](#)
- [5] *N. J. Higham, R. Li, F. Tisseur*: Backward error of polynomial eigenproblems solved by linearization. *SIAM J. Matrix Anal. Appl.* *29* (2007), 1218–1241. [zbl](#) [MR](#) [doi](#)
- [6] *N. J. Higham, D. S. Mackey, F. Tisseur, S. D. Garvey*: Scaling, sensitivity and stability in the numerical solution of quadratic eigenvalue problems. *Int. J. Numer. Methods Eng.* *73* (2008), 344–360. [zbl](#) [MR](#) [doi](#)
- [7] *T. Ikegami, T. Sakurai*: Contour integral eigensolver for non-Hermitian systems: a Rayleigh-Ritz-type approach. *Taiwanese J. Math.* *14* (2010), 825–837. [zbl](#) [MR](#)
- [8] *T. Ikegami, T. Sakurai, U. Nagashima*: A filter diagonalization for generalized eigenvalue problems based on the Sakurai-Sugiura projection method. *J. Comput. Appl. Math.* *233* (2010), 1927–1936. [zbl](#) [MR](#) [doi](#)
- [9] *E. E. Osborne*: On preconditioning of matrices. *J. Assoc. Comput. Math.* *7* (1960), 338–345. [zbl](#) [MR](#) [doi](#)
- [10] *B. Parlett, C. Reinsch*: Balancing a matrix for calculation of eigenvalues and eigenvectors. *Numer. Math.* *13* (1969), 293–304. [zbl](#) [MR](#) [doi](#)
- [11] *T. Sakurai, H. Sugiura*: A projection method for generalized eigenvalue problems using numerical integration. *J. Comput. Appl. Math.* *159* (2003), 119–128. [zbl](#) [MR](#) [doi](#)

- [12] *F. Tisseur*: Backward error and condition of polynomial eigenvalue problems. *Linear Algebra Appl.* 309 (2000), 339–361. [zbl](#) [MR](#) [doi](#)
- [13] *F. Tisseur, K. Meerbergen*: The quadratic eigenvalue problem. *SIAM Rev.* 43 (2001), 235–286. [zbl](#) [MR](#) [doi](#)
- [14] *S. Yokota, T. Sakurai*: A projection method for nonlinear eigenvalue problems using contour integrals. *JSIAM Lett.* 5 (2013), 41–44. [MR](#) [doi](#)

Authors' address: Hongjia Chen, Akira Imakura, Tetsuya Sakurai, Department of Computer Science, University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Ibaraki 305-8573, Japan, e-mail: chen@mma.cs.tsukuba.ac.jp.