



## Balancing and scaling techniques for Sakurai-Sugiura method in polynomial eigenvalue problem

著者	Chen Hongjia
year	2018
その他のタイトル	多項式固有値問題における櫻井-杉浦法に対するバランシング及びスケールリング手法の研究
学位授与大学	筑波大学 (University of Tsukuba)
学位授与年度	2017
報告番号	12102甲第8522号
URL	<a href="http://doi.org/10.15068/00152394">http://doi.org/10.15068/00152394</a>

Balancing and scaling techniques for Sakurai-Sugiura  
method in polynomial eigenvalue problem

March 2018

Chen Hongjia

Balancing and scaling techniques for Sakurai-Sugiura  
method in polynomial eigenvalue problem

Graduate School of Systems and Information Engineering  
University of Tsukuba

March 2018

Chen Hongjia

## Abstract

Eigenvalue problems lie in many fields of science and engineering, such as classical structural mechanics, molecular dynamics, gyroscopic systems, and MIMO systems in control theory. There are many types of eigenvalue problem, such as standard eigenvalue problem (SEP), generalized eigenvalue problem (GEP), and polynomial eigenvalue problem (PEP). QR and QZ methods are stable eigensolvers for solving the SEP and GEP.

Linearization form is a common way for solving the QEP or PEP. The idea of linearization form is to convert a QEP or PEP to a GEP, and compute eigenvalues of the GEP by QZ method. QZ method is a numerical stability method for computing eigenpairs in GEP. However, it can be not stability for computing eigenpairs in QEP and PEP. Moreover, in some applications, such as vibration analysis and mass-spring system, we are only interested in partial eigenvalues.

To avoid the difficulty for linearization form in solving QEP and PEP, we consider to use the Sakurai-Sugiura method with Rayleigh-Ritz projection (SS-RR method), which computes the eigenvalues inside a given curve using the contour integral. The aim of the SS-RR method is to reduce the dimension of original problem. The original PEP or QEP is converted into a small projected PEP or QEP using the Rayleigh-Ritz projection. However, the SS-RR method is not stable if the norms of matrices in projected matrix polynomial are separated widely.

The objectives of this thesis is to improve the backward stability of the SS-RR method for solving the PEP. To achieve this goal, we discuss two ideas for the SS-RR method.

The first one is the SS-RR method with scaling technique which combine the projected QEP with scaling technique. We give some assumptions and find the relation between backward error of original QEP and that of projected QEP. Based on these relations, we explain that the SS-RR method with scaling technique can reduce the backward error of computing eigenpairs in QEP.

Extending this idea, the second one is the SS-RR method with balancing technique which convert the projected PEP to SEP, then use the balancing technique in SEP. We investigate the reason that the SEP with balancing technique can improve the backward error of computing eigenpairs in PEP.

Finally, we show some numerical experiments that the SS-RR method with scaling and balancing techniques can improve the backward error of computing eigenpairs in PEP and QEP.

## Acknowledgements

At end of my Ph.D life, I would like to express many thanks for some persons who give me a lot of help

- Firstly, I would like to express my deepest appreciation to my supervisor Prof. T. Sakurai. I feel lucky that have this opportunity to do research under his guidance and support. He gives me a lot of helpful comments for my research and devote many time for discussing research with me. And also he gives me many chances to attend international conference, let me have opportunity to cooperate with other researchers. I would like to say thanks to him that give me a free environment for doing research. In past three years, under his guidance, I did well in my research field. Finally, Many thanks for him and his help.
- My second supervisor is Assistant Professor Imakura and I would like to thank him all of his help. He gives me a lot of useful comments for making paper. When I made my first paper, I did not have any experience. He help me modified my sentences in paper one by one. Thanks for his patient and discuss many times with my research.
- I would like to thank my thesis committee: Professor Kawabe, Professor Itoh, Associate Professor Sano and Assistant professor Imakura for carefully reading my thesis, and for the useful suggestions and comments.
- I would like to express thanks for Prof. F. Tisseur. I remembered that I met her on Sep. 2015. From that time, we discuss many times by email and also cooperate to make JSIAM paper. Thanks for her kind comments and give me a lot of comment in nonlinear eigenvalue problems.
- Last but not least, I would like to thank my family and friends. With my family unconditional support , I can finish my study and concentrate on research.



# Contents

<b>Abstract</b>	<b>i</b>
<b>Acknowledgements</b>	<b>iii</b>
<b>1 Introduction</b>	<b>2</b>
1.1 Background . . . . .	3
1.1.1 Standard eigenvalue problem . . . . .	3
1.1.2 Generalized eigenvalue problem . . . . .	3
1.2 Numerical methods for solving linear eigenvalue problems . . . . .	4
1.2.1 QR method . . . . .	4
1.2.2 QZ method . . . . .	6
1.3 Target problem . . . . .	7
1.4 Applications . . . . .	7
1.4.1 Vibration analysis . . . . .	8
1.4.2 Acoustic wave problem . . . . .	8
1.4.3 Mass-spring system . . . . .	9
1.5 Numerical method for solving PEP and QEP . . . . .	10



1.5.1	Linearization form . . . . .	11
1.5.2	Contour integral-based eigensolver . . . . .	12
1.6	Research objectives . . . . .	16
1.7	Overview of thesis . . . . .	17
<b>2</b>	<b>Backward error</b>	<b>19</b>
2.1	Introduction . . . . .	19
2.2	Normwise backward error . . . . .	19
2.3	Bounds of backward error of $P(\lambda)$ relative to that of $L(\lambda)$ . . . . .	21
2.4	Numerical experiments . . . . .	22
2.5	Conclusion . . . . .	23
<b>3</b>	<b>Scaling technique for Sakurai-Sugiura method in quadratic eigenvalue problem</b>	<b>26</b>
3.1	Introduction . . . . .	27
3.2	Scaling technique . . . . .	28
3.3	The proposed method . . . . .	29
3.4	Analysis of the backward errors of eigenpairs computed by the proposed method	31
3.5	Numerical experiments . . . . .	33
3.6	Conclusion . . . . .	35
<b>4</b>	<b>Balancing technique for Sakurai-Sugiura method in polynomial eigenvalue problem</b>	<b>38</b>
4.1	Introduction . . . . .	39

---

4.2	Balancing technique for the standard eigenvalue problem . . . . .	40
4.2.1	Osborne's balancing technique . . . . .	41
4.2.2	Parlett's balancing technique . . . . .	41
4.3	Eigensolver for the projected PEP using linearization . . . . .	42
4.4	The SS-RR method with the balancing technique for the PEP . . . . .	43
4.4.1	The proposed method . . . . .	44
4.4.2	Analysis of the backward error for the proposed method . . . . .	44
4.5	Numerical experiments . . . . .	51
4.5.1	Verification of the assumptions . . . . .	53
4.5.2	Evaluation of the backward error of $P(\lambda)$ . . . . .	53
4.6	Conclusion . . . . .	54
<b>5</b>	<b>A balancing technique for heavily damped quadratic eigenvalue problem</b>	<b>58</b>
5.1	Introduction . . . . .	58
5.2	Balancing technique . . . . .	59
5.2.1	Ward's balancing technique . . . . .	60
5.2.2	Lemonnier and Van Dooren's balancing technique . . . . .	60
5.2.3	Betcke's balancing technique . . . . .	60
5.3	A linearization form with balancing techniques for solving heavily damped QEP	61
5.3.1	Linearization form for heavily damped QEP . . . . .	61
5.3.2	Linearization form with Parlett's balancing technique . . . . .	62
5.3.3	Betcke's balancing technique for heavily damped QEP . . . . .	62

---

5.3.4	The proposed method . . . . .	63
5.4	Numerical experiments . . . . .	63
5.4.1	mod_wiresaw2 problem . . . . .	64
5.4.2	mod_hospital problem . . . . .	65
5.5	Conclusion . . . . .	66
<b>6</b>	<b>Conclusion and future work</b>	<b>69</b>
6.1	Conclusion . . . . .	69
6.2	Future work . . . . .	70
	<b>Bibliography</b>	<b>71</b>

# List of Tables

2.1	Polynomial eigenvalue problems [1]. . . . .	23
2.2	The maximum value of $\frac{\eta(P,\lambda,\mathbf{x})}{\eta(L,\lambda,\mathbf{z})}$ and value of $\rho$ . . . . .	23
3.1	List of test problems. . . . .	34
3.2	Parameters for the SS-RR method. . . . .	34
3.3	Norm of the coefficient matrices of $R(\lambda) = V^H Q(\lambda) V$ . . . . .	35
3.4	Minimum value of $\kappa$ in (3.8), type of scaling $\gamma$ used with <code>quadeig</code> and its value. . . . .	35
3.5	Largest backward errors of eigenpairs. . . . .	36
4.1	Polynomial eigenvalue problems [1]. . . . .	52
4.2	Parameters for the SS-RR method. . . . .	52
4.3	The value of $\varepsilon_1 := \max_{i=0:m} \frac{\ A_i\ _2}{\ V^H A_i V\ _2}$ . . . . .	53
4.4	The value of $\varepsilon_2 := \max_j B_U(\tilde{\lambda}_j, \tilde{\mathbf{y}}_j)$ . . . . .	54
4.5	The minimum value of the parameters $\delta$ , $\theta$ and $\tau$ for $P(\lambda)$ . . . . .	54
4.6	Maximum backward errors of the eigenpairs of $P(\lambda)$ . . . . .	55
5.1	Backward error ratio when $ \lambda $ close to $10^{-4}$ . . . . .	64

5.2	Backward error ratio when $ \lambda $ close to $10^6$ . . . . .	65
5.3	Backward error ratio when $ \lambda $ close to $10^{-4}$ . . . . .	66
5.4	Backward error ratio when $ \lambda $ close to $10^6$ . . . . .	67

# List of Figures

2.1	Backward error $\eta(L, \lambda, \mathbf{z})$ for the <code>sleeper</code> problem. . . . .	24
2.2	Backward error $\eta(P, \lambda, \mathbf{x})$ for the <code>sleeper</code> problem. . . . .	24
2.3	Backward error $\eta(L, \lambda, \mathbf{z})$ for the <code>power_plant</code> problem. . . . .	25
2.4	Backward error $\eta(P, \lambda, \mathbf{x})$ for the <code>power_plant</code> problem. . . . .	25
3.1	Backward errors for the <code>damped_beam</code> problem. . . . .	36
3.2	Backward errors for the <code>mod_sleeper</code> problem. . . . .	37
4.1	Backward error for the <code>damped_beam</code> problem. . . . .	56
4.2	Backward error for the <code>shaft</code> problem. . . . .	56
4.3	Backward error for the <code>plasma_drift</code> problem. . . . .	57
4.4	Backward error for the <code>orr_sommerfeld</code> problem. . . . .	57
5.1	Reduction of backward error of smallest eigenpairs of <code>mod_wiresaw2</code> using balancing. . . . .	65
5.2	Reduction of backward error of largest eigenpairs of <code>mod_wiresaw2</code> using balancing. . . . .	66
5.3	Reduction of backward error of smallest eigenpairs of <code>mod_hospital</code> using balancing. . . . .	67

5.4 Reduction of backward error of largest eigenpairs of `mod_hospital` using balancing. 68

6.1 A map of research in this thesis. . . . . 70

# Chapter 1

## Introduction

Many problems arise in science and engineering fields from the mathematical model of eigenvalue problems, such as finite elements analysis, vibration analysis of building, quantum physics and data analysis.

Matrix polynomial lies at a important position of eigenvalue problems, it has many applications in engineering areas, such as oscillation analysis of structural mechanics, and acoustic systems in electrical circuit simulation [2]. Based on matrix polynomial, there are several types of eigenvalue problems. Here, we give the definition of matrix polynomial  $P(\lambda)$ ,

$$P(\lambda) = \lambda^m A_m + \lambda^{m-1} A_{m-1} + \cdots + A_0 \quad (1.1)$$

where  $A_k \in \mathbb{C}^{n \times n} \setminus \{O\}$ ,  $k = 0, \dots, m$ .

In this chapter, we firstly introduce linear eigenvalue problems and numerical methods for solving linear eigenvalue problems in Section 1.1. Secondly, we define the target problem and applications in Sections 1.3 and 1.4. Then we review the state-of-the-art numerical methods, such as linearization form and contour integral-based eigensolvers for solving polynomial eigenvalue problem in Section 1.5. Finally, we show research objectives in Section 1.6 and give a overview of this research.



## 1.1 Background

In this section, firstly, we will introduce linear eigenvalue problems, such as standard eigenvalue problem and generalized eigenvalue problem. Then we will describe some methods for solving linear eigenvalue problem.

### 1.1.1 Standard eigenvalue problem

Based on (1.1), when  $A_1 = I_n$ ,  $I_n$  is identity matrix and  $m = 1$ , we have the standard eigenvalue problem (SEP)

$$A_0 \mathbf{x} = \lambda \mathbf{x}, \quad A_0 \in \mathbb{C}^{n \times n},$$

where  $\lambda$  are eigenvalues and  $\mathbf{x} \in \mathbb{C}^n \setminus \{\mathbf{0}\}$  are associated eigenvectors. QR method is a standard method for computing all eigenvalues in the SEP [3, 4].

### 1.1.2 Generalized eigenvalue problem

From (1.1), if  $m = 1$  and  $A_1$  is not an identity matrix, we have a generalized eigenvalue problem (GEP)

$$(\lambda A_1 + A_0) \mathbf{x} = 0,$$

where the matrices  $A_0, A_1 \in \mathbb{C}^{n \times n}$ ,  $\lambda$  are eigenvalues and  $\mathbf{x} \in \mathbb{C}^n \setminus \{\mathbf{0}\}$  are associated eigenvectors.

QZ method is a stable method for computing all eigenvalues in GEP. For large sparse GEP, eigenvalues are computed by a Krylov method [5].

## 1.2 Numerical methods for solving linear eigenvalue problems

In this section, we introduce some numerical methods based on unitary transformations for solving linear eigenvalue problem.

### 1.2.1 QR method

We review QR method for solving standard eigenvalue problems.

A matrix  $A \in \mathbb{C}^{n \times n}$  can be transformed into a Schur factorization

$$A = PUP^T,$$

where  $P^T P = I_n$ ,  $U$  is an upper triangular matrix. The diagonal elements of  $U$  are the eigenvalues of  $A$ .

The idea of QR method is to compute Schur factorization for target matrix using a similarity transformation. Let  $A_1 = A$  and compute QR decomposition

$$A_1 = Q_1 R_1,$$

where  $Q_1^T Q_1 = I$  and  $R_1$  is an upper triangular matrix. Using this relation and set

$$A_2 = Q_1^T A_1 Q_1 = R_1 Q_1.$$

Similarity, with  $k = 1, 2, \dots$ , we iterate

$$A_k = Q_k R_k, \quad A_{k+1} = R_k Q_k.$$

Then  $A_k$  is converged to an upper triangular matrix, and its diagonal elements are the eigen-

**Algorithm 1** QR iteration**Input:**A matrix  $A \in \mathbb{C}^{n \times n}$ .

- 1: Let  $A_0 = A$ .
- 2: **for**  $k = 1, \dots$ , **do**
- 3:    $A_k = Q_k R_k$  (QR decomposition).
- 4:   Compute  $A_{k+1} = R_k Q_k$ .
- 5: **end for**

values of  $A$ . The Algorithm 1 show the steps of QR iteration.

However, the basic QR method has two disadvantages in practice.

- Firstly, the computation costs of QR decomposition is relatively high  $O(n^3)$ .
- Secondly, the QR method includes many iteration steps for reaching convergence. To reach convergence, the computation cost for iteration steps is very high.

To reduce the computation cost of QR decomposition, we use the Householder method. The idea of Householder method is to transform the matrix  $A$  into a Hessenberg form by Householder transformation, such as

$$A = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix} \implies H = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \end{bmatrix},$$

then, the computation cost of QR decomposition for the Hessenberg matrix is reduced to  $O(n^2)$ .

To avoid the second disadvantage, we introduce QR method with a incorporating shifts. The main steps of the shifted QR method are presented in Algorithm 2.

---

**Algorithm 2** The shifted QR method
 

---

**Input:**

- A matrix  $A \in \mathbb{C}^{n \times n}$ .
- 1: Set  $A_0 = A$
  - 2: **for**  $k = 1, \dots$ , **do**
  - 3:   Compute QR-factorization  $Q_k A_k = A_{k-1} - \sigma_k I$ , where  $\sigma_k$  is a rough approximation to eigenvalue.
  - 4:   Compute  $A_k = R_k Q_k + \sigma_k I$ .
  - 5: **end for**
- 

### 1.2.2 QZ method

We now discuss numerical method for solving generalized eigenvalue problem (GEP). QZ method is a stable eigensolver for computing all eigenpairs in GEP. Moler and Stewart proposed QZ method for solving generalized eigenvalue problems [6]. After that, the QZ method is modified by [7, 8, 9, 10].

Let  $A, B \in \mathbb{C}^{n \times n}$ , the generalized eigenvalue problem is defined by

$$A\mathbf{x} = \lambda B\mathbf{x},$$

where  $\lambda$  are eigenvalues and  $\mathbf{x} \in \mathbb{C}^n \setminus \{\mathbf{0}\}$  are associated eigenvectors. We also define  $A - \lambda B$  is a matrix pencil with  $\lambda \in \mathbb{C}$ .

The idea of QZ method is to find two unitary matrices  $Q$  and  $Z$ , then convert  $(A, B)$  to  $(\tilde{A}, \tilde{B})$ ,

$$\tilde{A} = Q^H A Z, \quad \tilde{B} = Q^H B Z.$$

where  $\tilde{A}, \tilde{B}$  are upper triangular matrices. This transformation is called generalized Schur decomposition of a matrix pair  $(A, B)$ . We compute finite eigenvalues  $\lambda_i$  of  $A\mathbf{x} = \lambda B\mathbf{x}$  using  $\lambda = a_{ii}/b_{ii}$ , where  $a_{ii}$  and  $b_{ii}$  are the diagonal elements of  $\tilde{A}$  and  $\tilde{B}$  and  $b_{ii} \neq 0$ .

To compute generalized Schur decomposition of a matrix pair  $(A, B)$ , We transform the matrix pair  $(A, B)$  to a Hessenberg-Triangular matrix pair  $(H, T)$  by Householder transformation and the Givens rotations. When we convert the matrix pair  $(H, T)$  to  $HT^{-1}$ , the QZ method is

consider as the QR method [11].

### 1.3 Target problem

The target problem is polynomial eigenvalue problem (PEP). We defined the PEP by

$$P(\lambda)\mathbf{x} = \left( \sum_{i=0}^m \lambda^i A_i \right) \mathbf{x} = \mathbf{0}, \quad (1.2)$$

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

If the degree of (1.1)  $m = 2$ , we have a quadratic eigenvalue problem (QEP)

$$Q(\lambda)\mathbf{x} = (\lambda^2 A_2 + \lambda A_1 + A_0)\mathbf{x} = \mathbf{0}, \quad (1.3)$$

where the matrices  $A_0, A_1, A_2 \in \mathbb{C}^{n \times n}$ ,  $\lambda$  are eigenvalues and  $\mathbf{x} \in \mathbb{C}^n \setminus \{\mathbf{0}\}$  are associated eigenvectors.

QEP is a special type of PEP and it also can be converted to a linear eigenvalue problem, then compute all eigenvalues by QZ method.

In this thesis, we will study some numerical methods for computing eigenpairs  $(\lambda, \mathbf{x})$  of  $P(\lambda)$  and improve the backward stability of computing eigenpairs  $(\lambda, \mathbf{x})$  of  $P(\lambda)$ .

### 1.4 Applications

Polynomial eigenvalue problem arises in many fields of science and engineering. Here, we introduce several applications of PEP, such as vibration analysis of structures, acoustic wave problem and mass-spring system.

### 1.4.1 Vibration analysis

This quadratic eigenvalue problem arises from vibration analysis of structures. In vibration analysis,  $A_2$  is mass matrix,  $A_1$  is damping matrix and  $A_0$  is stiffness matrix. To reduce the damage of vibration in earthquake, a viscous damper has been designed in a piston [12], We consider this model of viscous damper as the solution of following equation,

$$A_2\ddot{u}(t) + A_1\dot{u}(t) + A_0u(t) = 0, \quad (1.4)$$

where  $A_2, A_1, A_0$  are mass, damping and stiffness matrices and  $\|A_1\|_2 \gg \sqrt{\|A_2\|_2\|A_0\|_2}$ . Based on (1.4), this problem can be solved by the heavily damped QEP

$$(\lambda^2 A_2 + \lambda A_1 + A_0)\mathbf{x} = \mathbf{0}, \quad \|A_1\|_2 \gg \sqrt{\|A_2\|_2\|A_0\|_2}. \quad (1.5)$$

### 1.4.2 Acoustic wave problem

This quadratic eigenvalue problem arises from acoustic wave problem [13]. The formulation of this problem is given by

$$\frac{1}{s^2} \frac{\partial^2 r}{\partial t^2} - \Delta r = p(\mathbf{x}, t). \quad (1.6)$$

Here, we define  $p(\mathbf{x}, t)$  and  $s(\mathbf{x})$  are acoustic pressure and sound speed functions with two independent variables  $\mathbf{x}$  and  $t$ . We also define  $\mathbf{x}$  are the coordinates variable,  $t$  is time variable.

When (1.6) is homogeneous, the problem is transformed into an eigenvalue problem

$$r(\mathbf{x}, t) = \tilde{r}(\mathbf{x})e^{\tilde{\lambda}t}. \quad (1.7)$$

(1.7) is satisfied boundary conditions which on  $\Omega_1, \Omega_2, \Omega_3$

$$\frac{\partial \tilde{r}}{\partial n} = 0, \quad (1.8)$$

$$\tilde{r} = 0, \quad (1.9)$$

$$\frac{\partial \tilde{r}}{\partial n} = -\frac{\tilde{\lambda}}{\xi} \tilde{r}. \quad (1.10)$$

Equation (1.7) and boundary conditions (1.8-1.10) is solved by a quadratic eigenvalue problem

$$(\lambda^2 M + \lambda D + K) \mathbf{r} = \mathbf{0}, \quad (1.11)$$

where  $M, D, K$  are square, sparse matrices and  $\lambda$  are eigenvalues and  $\mathbf{r}$  is associate eigenvectors.

### 1.4.3 Mass-spring system

This QEP arises in an  $n$  degree of freedom damped mass-spring system [14]. This problem is considered as a second-order differential equation

$$A_2 \frac{d^2}{dt^2} x + A_1 \frac{d}{dt} x + A_0 x = 0,$$

we denote the mass matrix that

$$A_2 = \begin{bmatrix} a_1 & & & \\ & a_2 & & \\ & & \ddots & \\ & & & a_n \end{bmatrix},$$

then the damping matrix  $A_1$  and stiffness matrix  $A_0$  are defined by

$$A_1 = D \begin{bmatrix} b_1 & & & & \\ & b_2 & & & \\ & & \ddots & & \\ & & & b_{n-1} & \\ & & & & 0 \end{bmatrix} D^T + \begin{bmatrix} \alpha_1 & & & & \\ & \alpha_2 & & & \\ & & \ddots & & \\ & & & \alpha_{n-1} & \\ & & & & \alpha_n \end{bmatrix},$$

$$A_0 = D \begin{bmatrix} c_1 & & & & \\ & c_2 & & & \\ & & \ddots & & \\ & & & c_{n-1} & \\ & & & & 0 \end{bmatrix} D^T + \begin{bmatrix} \beta_1 & & & & \\ & \beta_2 & & & \\ & & \ddots & & \\ & & & \beta_{n-1} & \\ & & & & \beta_n \end{bmatrix}.$$

Then the problem is transformed into a quadratic eigenvalue problem,

$$(\lambda^2 A_2 + \lambda A_1 + A_0)\mathbf{x} = \mathbf{0},$$

where  $\lambda$  are eigenvalues and  $\mathbf{x}$  are corresponding eigenvectors.

## 1.5 Numerical method for solving PEP and QEP

In this section, we discuss some numerical methods for computing eigenpairs for PEP. There are two types methods for computing eigenpairs in PEP. First one is linearization form for computing all eigenpairs in PEP. The second one computes partial eigenpairs in PEP by using a given contour.



### 1.5.1 Linearization form

Linearization form is a classic approach for computing all eigenpairs of  $P(\lambda)$ . The idea is to convert  $P(\lambda)$  to a linearization form  $L(\lambda)$

$$L(\lambda) = \lambda X + Y, \quad (1.12)$$

where  $X, Y \in \mathbb{C}^{mn \times mn}$ .  $L(\lambda)$  and  $P(\lambda)$  have the same eigenvalues if

$$H(\lambda)L(\lambda)M(\lambda) = \begin{bmatrix} P(\lambda) & O \\ O & I \end{bmatrix}$$

with  $H(\lambda)$  and  $M(\lambda)$  are unimodular matrix polynomial. Then we consider (1.12) as a generalized eigenvalue problem (GEP)

$$L(\lambda)\mathbf{z} = (\lambda X + Y)\mathbf{z} = \mathbf{0}. \quad (1.13)$$

We compute all eigenpairs  $(\lambda, \mathbf{z})$  of  $L(\lambda)$  with QZ method. Finally, we recover eigenpairs  $(\lambda, \mathbf{x})$  of  $P(\lambda)$  from  $(\lambda, \mathbf{z})$  of  $L(\lambda)$ .

There are several choice for linearization form  $L(\lambda)$ . Illustrating by example, assume the degree of matrix polynomial  $m = 2$ . We have the first and second companion linearization form

$$C_1(\lambda) = \lambda \begin{bmatrix} A_2 & O \\ O & I_n \end{bmatrix} + \begin{bmatrix} A_1 & A_0 \\ -I_n & O \end{bmatrix}, \quad C_2(\lambda) = \lambda \begin{bmatrix} A_2 & O \\ O & I_n \end{bmatrix} + \begin{bmatrix} A_1 & -I_n \\ A_0 & O \end{bmatrix},$$

where  $I_n \in \mathbb{R}^{n \times n}$  is the identity matrix. When  $A_2$  and  $A_0$  are nonsingular matrices, we have

$$L_1(\lambda) = \lambda \begin{bmatrix} A_2 & O \\ O & -A_0 \end{bmatrix} + \begin{bmatrix} A_1 & A_0 \\ A_0 & O \end{bmatrix}, \quad L_2(\lambda) = \lambda \begin{bmatrix} O & A_2 \\ A_2 & A_1 \end{bmatrix} + \begin{bmatrix} -A_2 & O \\ O & A_0 \end{bmatrix}.$$

Other linearization forms are introduced in [15, 16, 17].

However, there are some difficulty for linearization form in solving PEP.

- When the original matrix polynomial  $P(\lambda) \in \mathbb{C}^{n \times n}$  with degree  $m$  is converted to the larger  $mn \times mn$  linearization form  $L(\lambda)$ , if  $m \gg 2$ , the computation cost for eigenpairs of  $P(\lambda)$  is very large.
- A lot of PEP can not be solved by linearization form accurately when the norms of coefficient matrices vary widely.
- Linearization form is a classic way for computing all eigenpairs. However, in some applications, such as structural dynamics and structural-acoustic interaction, it is unnecessary to compute all eigenpairs, and partial eigenpairs  $(\lambda, \mathbf{x})$  which have physic property are sufficient.

### 1.5.2 Contour integral-based eigensolver

To prevent the inflation of matrix dimension of polynomial eigenvalue problem by using linearization form and only focus on partial eigenvalues, we introduce a contour integral-based eigensolver for computing partial eigenpairs in polynomial eigenvalue problem (PEP). This method is called Sakurai-Sugiura (SS) method [18].

The original SS method converted generalized eigenvalue problem (GEP) to a projected GEP based on a subspace corresponding the target eigenvalues and eigenvectors which are inside of a given curve. For solving generalized eigenvalue problem and nonlinear eigenvalue problem, the SS method has a number of extension types which are based on Hankel matrix pencil and Rayleigh-Ritz projection. When the target problem is generalized eigenvalue problem, SS-Hankel method [18, 19] transform the GEP to a small dimension Hankel matrix pencil. The SS-RR method projects the GEP to a projected GEP with small dimension by using Rayleigh-Ritz projection [20].

Extend to solve nonlinear eigenvalue problem, we also have SS-Hankel and SS-RR method [21, 22, 23] based on contour integral theorem.

## Hankel type of Sakurai-Sugiura method

The idea of SS-Hankel method is to convert the target problem to Hankel matrix pencil with a smaller dimension, then compute eigenvalues which inside of a given curve  $\Gamma$  using contour integral.

We denote  $U, V \in \mathbb{C}^{n \times L}$  that is input matrices and  $K, L \in \mathbb{N}^+$ . The block complex moment  $\Psi_k$  is defined by

$$\Psi_k = \frac{1}{2\pi i} \int_{\Gamma} z^k U^H P(z)^{-1} V dz, \quad k = 0, 1, \dots, 2K - 1 \quad (1.14)$$

where  $P(z)$  is a matrix polynomial and  $z \in \mathbb{C}$ . In numerical calculations, we approximate the contour integral (1.14) using trapezoidal rule,

$$\Psi_k \approx \widehat{\Psi}_k := \frac{1}{N} \sum_{j=1}^{N-1} \Theta_k U^H P(\omega_j)^{-1} V, \quad k = 0, 1, \dots, 2K - 1 \quad (1.15)$$

where  $\Theta_k = \left(\frac{\omega_j - \gamma}{\rho}\right)^{k+1}$ . We then construct the Hankel matrices  $H_{KL}$  and  $H_{KL}^<$ , such that

$$H_{KL} := \begin{bmatrix} \Psi_0 & \Psi_1 & \cdots & \Psi_{K-1} \\ \Psi_1 & \Psi_2 & \cdots & \Psi_K \\ \vdots & \vdots & \ddots & \vdots \\ \Psi_{K-1} & \Psi_K & \cdots & \Psi_{2K-2} \end{bmatrix} \quad H_{KL}^< := \begin{bmatrix} \Psi_1 & \Psi_2 & \cdots & \Psi_K \\ \Psi_2 & \Psi_3 & \cdots & \Psi_{K+1} \\ \vdots & \vdots & \ddots & \vdots \\ \Psi_K & \Psi_{K+1} & \cdots & \Psi_{2K-1} \end{bmatrix}$$

Then we reduce the dimension of Hankel matrices  $H_{KL}$  and  $H_{KL}^<$  by using the singular value decomposition of  $H_{KL} = V_{KL} \Sigma U_{KL}^H$  with discard small singular value  $\sigma_{\widehat{p}}$  in  $\Sigma$ ,

$$\Sigma = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \end{bmatrix}.$$

**Algorithm 3** The SS-Hankel method

**Input:**  $U, V \in \mathbb{C}^{n \times L}$ ,  $N, K, L$  are non-zero integers. A matrix polynomial  $P(\lambda)$  where  $P(\lambda) = \lambda^m A_m + \lambda^{m-1} A_{m-1} + \dots + A_0$ .

**Output:**  $\hat{\lambda}_i, \hat{\mathbf{x}}_i, i = 1, \dots, \hat{p}$ .

- 1: Compute  $\hat{S}_k$  by (1.16) and construct block moments  $\Psi_k$   $k = 0 : 2K - 1$ .
- 2: Construct block Hankel matrix  $H_{KL}$  and  $H_{KL}^<$ ,  $k = 0 : 2K - 1$ .
- 3: Make a singular value decomposition for  $H_{KL}$  and obtain smaller dimension Hankel matrices  $H_{\hat{p}} = H_{KL}(1 : \hat{p}, 1 : \hat{p})$ ,  $H_{\hat{p}}^< = H_{KL}^<(1 : \hat{p}, 1 : \hat{p})$
- 4: Compute eigenpairs  $(\phi_i, \mathbf{q}_i)$  for generalized eigenvalue problem  $(H_{\hat{p}}^< - \phi H_{\hat{p}})\mathbf{q} = \mathbf{0}$ .
- 5: Compute eigenpairs  $(\hat{\lambda}_i, \hat{\mathbf{x}}_i)$  of  $P(\lambda)$  using  $\hat{\lambda}_i = \gamma + \rho\phi_i$  and  $\hat{\mathbf{x}}_i = S\mathbf{q}_i$ .

Therefore, we have smaller dimension matrices

$$H_{\hat{p}} = H_{KL}(1 : \hat{p}, 1 : \hat{p}), \quad H_{\hat{p}}^< = H_{KL}^<(1 : \hat{p}, 1 : \hat{p})$$

with  $\hat{p} \leq KL$ . We compute the eigenpairs  $(\phi_i, \mathbf{q}_i)$  of  $H_{\hat{p}}^< - \lambda H_{\hat{p}}$  and extract the eigenvalues  $\lambda$  which inside  $\Gamma$ .

To compute the eigenvectors of  $P(\lambda)$ , we define

$$S_k := \frac{1}{2\pi i} \int_{\Gamma} z^k P(z)^{-1} V dz, \quad k = 0, 1, \dots, 2K - 1$$

and

$$S = \begin{bmatrix} S_0 & S_1 & \dots & S_{\hat{p}-1} \end{bmatrix}.$$

The approximation of  $S_k$  then is given by

$$S_k \approx \hat{S}_k := \frac{1}{N} \sum_{j=0}^{N-1} \Theta_k F(\omega_j)^{-1} V, \quad k = 0, 1, \dots, 2K - 1 \quad (1.16)$$

where  $\Theta_k = \left(\frac{\omega_j - \gamma}{\rho}\right)^{k+1}$ . Finally, the eigenvectors  $\mathbf{x}_i$  of  $P(\lambda)$  are computed by  $\mathbf{x}_i = S\mathbf{q}_i$  and eigenvalues  $\lambda_i$  of  $P(\lambda)$  are recovered from  $\lambda_i = \gamma + \rho\phi_i$ ,  $i = 1, \dots, \hat{p}$ .

The main steps of the SS-Hankel method are presented in Algorithm 3.

### Rayleigh-Ritz type of Sakurai-Sugiura method

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

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

and

$$S_k = \frac{1}{2\pi i} \int_{\Gamma} g_k(z) P(z)^{-1} U dz \in \mathbb{C}^{n \times L}, \quad (1.17)$$

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  $\mathcal{R}\{S\}$  [21].

We use a numerical quadrature to approximate the contour integral (1.17). The approximation of  $S_k$  is given by

$$S_k \approx \widehat{S}_k = \sum_{p=1}^N \omega_p g_k(z_p) P(z_p)^{-1} U, \quad (1.18)$$

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

We construct  $\widehat{S} = \begin{bmatrix} \widehat{S}_0, \dots, \widehat{S}_{K-1} \end{bmatrix}$  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  $V = \widehat{V}(:, 1 : \ell)$ , and  $\ell$  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  $\ell$ . Let the computed eigenpairs of  $R(\lambda)$  be denoted by  $(\widehat{\lambda}_j, \widehat{\mathbf{y}}_j)$ , where  $\widehat{\mathbf{y}}_j \in \mathbb{C}^\ell$ . Then, the eigenpairs  $(\widehat{\lambda}_j, \widehat{\mathbf{x}}_j)$  of  $P(\lambda)$  are approximated by

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

---

**Algorithm 4** 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  $\Gamma$ , and a matrix polynomial  $P(\lambda)$ .

**Output:**  $\hat{\lambda}_j, \hat{\mathbf{x}}_j, j = 1, \dots, n(\Gamma)$ , 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  $\hat{S}_k, k = 0, \dots, K - 1$  by (1.18).

3: Compute the singular value decomposition  $\hat{S} = \hat{V}\hat{\Sigma}\hat{W}^H$ , where  $\hat{S} = [\hat{S}_0, \dots, \hat{S}_{K-1}]$ .

4: Set  $V = \hat{V}(:, 1 : \ell)$ , where  $\ell$  is a numerical rank of  $\hat{S}$ .

5: Compute eigenpairs  $(\hat{\lambda}_j, \hat{\mathbf{y}}_j)$  of  $R(\lambda) = V^H P(\lambda) V, j = 1, \dots, n(\Gamma)$ .

6: Extract the  $n(\Gamma)$  eigenvalues  $\hat{\lambda}_j, j = 1, \dots, n(\Gamma)$  that are inside  $\Gamma$  and set  $\hat{\mathbf{x}}_j = V\hat{\mathbf{y}}_j, j = 1, \dots, n(\Gamma)$ .

---

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

The main steps of the SS-RR method are presented in Algorithm 4.

## 1.6 Research objectives

In this thesis, we compute partial eigenpairs of  $P(\lambda)$  using the SS-RR method.  $P(\lambda)$  is transformed into a matrix polynomial with a small dimension as

$$R(\lambda) = V^H P(\lambda) V, \quad (1.19)$$

where the matrix  $V \in \mathbb{C}^{n \times \ell}, \ell \ll n$ , has orthonormal columns consisting of basis vectors for the subspace constructed by the SS-RR method. However, the SS-RR method suffers from backward instability when the norms of the coefficient matrices of  $R(\lambda)$  vary widely. (Backward instability means the backward errors of computing eigenpairs are very large)

The objective of this research is that

- improve the backward stability of computing eigenpairs for the SS-RR method in PEP
- explain why the use of stable eigensolver in (1.19) improves the backward stability of computing eigenpairs for the SS-RR method.

To achieve this goal, we have two ideas for improving the backward stability of computing eigenpairs in the SS-RR method

- *The SS-RR method with scaling technique.* To solve the QEP (1.3), we combine (1.19) with the stable eigensolver `quadeig` (scaling technique has been implemented in `quadeig`) in the SS-RR method. To explain reason that scaling technique can improve the backward stability of the SS-RR method, we construct relation between the backward error of  $R(\lambda)$  and that of  $Q(\lambda)$ . We found the SS-RR with `quadeig` improve the backward stability of computing eigenpairs under some assumptions [24].
- *The SS-RR method with balancing technique.* We extend the idea in [24] to solve the PEP. One common way for solving (1.19) 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 [25, 26], that is a preprocessing 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 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 backward stability of computing eigenpairs under some assumptions [27].

## 1.7 Overview of thesis

In this section, we introduce the organization of this thesis.

In Chapter 1, we introduce the background of this research, target problem and numerical methods for solving polynomial eigenvalue problem. Finally, we give objectives of this thesis and how to achieve these goals. Finally, we give a overview of objectives of this thesis.

In Chapter 2, we introduce the definition and explicit expression for backward error of computing eigenpairs. Backward error is very important for analyzing the numerical stability of the SS-RR method. Then we show the bound of backward error of  $P(\lambda)$  relative to  $L(\lambda)$ . Some numerical experiments are shown in Chapter 2.

In Chapter 3, we propose a method based on the SS-RR method. We combined the SS-RR method with scaling technique for improving the numerical stability of the SS-RR method. Then we analyze the backward error of the proposed method and that of the SS-RR method. We analyze the backward stability of the proposed method and show, through numerical experiments, that it computes eigenpairs with backward errors that are smaller than those computed by the SS-RR method.

In Chapter 4, to improve the backward stability of the SS-RR method in polynomial eigenvalue problem, 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.

In Chapter 5, we discuss a new balancing technique for solving the heavily damped quadratic eigenvalue problem, we compare backward errors of linearization form with several types of balancing techniques. Numerical experiments show the proposed method can reduce the backward error for computing eigenpairs in heavily damped quadratic eigenvalue problem.

In Chapter 6, conclusions and suggestions for future studies are presented.



# Chapter 2

## Backward error

### 2.1 Introduction

Backward error is an important definition for analyzing the numerical stability of algorithms for solving standard eigenvalue problem, generalized eigenvalue problem, and polynomial eigenvalue problem.

In this section, we introduce the definition of backward error of eigenpairs in polynomial eigenvalue problems and generalized eigenvalue problems.

We also introduce some bounds of backward error of  $P(\lambda)$  relative to that of  $L(\lambda)$ . These bounds will be used in next chapters. Finally, we investigate bounds of backward error  $P(\lambda)$  relative to that of  $L(\lambda)$  in numerical experiments.

### 2.2 Normwise backward error

We usually use the normwise backward error for analyzing the numerical stability in polynomial eigenvalue problems.

**Definition 2.1** ([14]). Let  $P(\lambda)$  be the matrix polynomial,

$$P(\lambda_j) = \sum_{i=0}^m \lambda_j^i A_i.$$

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

$$\begin{aligned} \eta(P, \widehat{\lambda}_j, \widehat{\mathbf{x}}_j) &:= \min\{\epsilon : (P(\widehat{\lambda}_j) + \Delta P(\widehat{\lambda}_j))\widehat{\mathbf{x}}_j = \mathbf{0}, \\ &\|\Delta A_i\|_2 \leq \epsilon \|A_i\|_2, i = 0, \dots, m\}, \end{aligned}$$

where  $\Delta P(\lambda_j) = \sum_{i=0}^m \lambda_j^i \Delta A_i$ ,  $\Delta A_i$  is a perturbation matrix.

The definition of backward error  $\eta(L, \widehat{\lambda}_j, \widehat{\mathbf{z}}_j)$  for linearization form is similar to definition of backward error  $\eta(P, \widehat{\lambda}_j, \widehat{\mathbf{x}}_j)$ . We have the following definition.

**Definition 2.2** ([14]). Let  $L(\lambda)$  be a linearization form,

$$L(\lambda) = \lambda X + Y.$$

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

$$\begin{aligned} \eta(L, \widehat{\lambda}_j, \widehat{\mathbf{z}}_j) &:= \min\{\epsilon : (L(\widehat{\lambda}_j) + \Delta L(\widehat{\lambda}_j))\widehat{\mathbf{z}}_j = \mathbf{0}, \\ &\|\Delta X\|_2 \leq \epsilon \|X\|_2, \|\Delta Y\|_2 \leq \epsilon \|Y\|_2\}, \end{aligned}$$

where  $\Delta L(\lambda_j) = \lambda_j \Delta X + \Delta Y$ ,  $\Delta X$  and  $\Delta Y$  are perturbation matrices.

For computing the backward error numerically, explicit expressions for the backward error of  $\eta(P, \widehat{\lambda}_j, \widehat{\mathbf{x}}_j)$  and  $\eta(L, \widehat{\lambda}_j, \widehat{\mathbf{z}}_j)$  are given by the following formula [28]:

$$\eta(P, \widehat{\lambda}_j, \widehat{\mathbf{x}}_j) = \frac{\|P(\widehat{\lambda}_j)\widehat{\mathbf{x}}_j\|_2}{(\sum_{i=0}^m |\widehat{\lambda}_j|^i \|A_i\|_2) \|\widehat{\mathbf{x}}_j\|_2}, \quad (2.1)$$

$$\eta(L, \widehat{\lambda}_j, \widehat{\mathbf{z}}_j) = \frac{\|L(\widehat{\lambda}_j)\widehat{\mathbf{z}}_j\|_2}{(|\widehat{\lambda}_j|\|X\|_2 + \|Y\|_2)\|\widehat{\mathbf{z}}_j\|_2}. \quad (2.2)$$

## 2.3 Bounds of backward error of $P(\lambda)$ relative to that of $L(\lambda)$

To compute all eigenpairs of  $P(\lambda)$ , the classical approach is to convert  $P(\lambda)$  to a linearization form  $L(\lambda)$ . Based on this approach, we also discuss the backward errors of  $P(\lambda)$  and  $L(\lambda)$ . We introduce some relations between backward error of  $P(\lambda)$  and that of  $L(\lambda)$ . These relations are very useful for analyzing numerical stability of algorithms.

For finding a relation between backward error of  $P(\lambda)$  and that of  $L(\lambda)$ , we need to find a relation between  $P(\lambda)$  and  $Q(\lambda)$ . From [29], we have

$$G(\lambda)L(\lambda) = e_1^T \otimes P(\lambda),$$

where  $\otimes$  is the Kronecker product [30] and  $e_1$  is the 1st column of identity matrix.  $G(\lambda)$  is a  $n \times nm$  matrix polynomial. Based on this relation, we have

$$\|G(\lambda)L(\lambda)\mathbf{z}\|_2 = \|P(\lambda)\mathbf{x}\|_2 \leq \|G(\lambda)\|_2\|\mathbf{z}\|_2. \quad (2.3)$$

We have the following theorem

**Theorem 2.1** ([29]). *Let  $(\lambda, \mathbf{x})$  be approximate eigenpairs of  $P(\lambda)$  and  $(\lambda, \mathbf{z})$  be approximate right eigenpairs of  $L(\lambda)$ . The bound of  $\eta(P, \lambda, \mathbf{x})/\eta(L, \lambda, \mathbf{z})$  is given by*

$$\frac{\eta(P, \lambda, \mathbf{x})}{\eta(L, \lambda, \mathbf{z})} \leq \frac{|\lambda|\|X\|_2 + \|Y\|_2}{\sum_{i=0}^m \|A_i\|_2|\lambda|^i} \frac{\|G(\lambda)\|_2\|\mathbf{z}\|_2}{\|\mathbf{x}\|_2}. \quad (2.4)$$

*Proof.* Using (2.1) and (2.2), we have

$$\frac{\eta(P, \lambda, \mathbf{x})}{\eta(L, \lambda, \mathbf{z})} = \frac{\|P(\lambda)\mathbf{x}\|_2}{(\sum_{i=0}^m \|A_i\|_2|\lambda|^i)\|\mathbf{x}\|_2} \frac{(|\lambda|\|X\|_2 + \|Y\|_2)\|\mathbf{z}\|_2}{\|L(\lambda)\mathbf{z}\|_2}.$$

Based on (2.3), we have

$$\|P(\lambda)\mathbf{x}\|_2 \leq \|G(\lambda)\|_2 \|L(\lambda)\mathbf{z}\|_2.$$

Therefore, we have

$$\frac{\eta(P, \lambda, \mathbf{x})}{\eta(L, \lambda, \mathbf{z})} \leq \frac{|\lambda| \|X\|_2 + \|Y\|_2 \|G(\lambda)\|_2 \|\mathbf{z}\|_2}{\sum_{i=0}^m \|A_i\|_2 |\lambda|^i} \|\mathbf{x}\|_2.$$

□

To discuss the bound of  $\frac{\eta(P, \lambda, \mathbf{x})}{\eta(L, \lambda, \mathbf{z})}$  in detail, Higham [29] give a quantity instead of (2.4),

$$\rho = \frac{\max_i \|A_i\|_2}{\min(\|A_0\|_2, \|A_m\|_2)}. \quad (2.5)$$

QZ method can be stable for linear eigenvalue problems. However, it is unstable for PEP. This means that  $\eta(L, \lambda, \mathbf{z})$  is small but  $\eta(P, \lambda, \mathbf{x})$  may be very large. Therefore, Theorem 2.1 shows that if  $\frac{\eta(P, \lambda, \mathbf{x})}{\eta(L, \lambda, \mathbf{z})} \approx 1$  as long as  $\rho \approx 1$ , the backward error of  $P(\lambda)$  can be reduced.

The quantity  $\rho$  also give a predict that a eigensolver is stable or not for PEP before computing eigenpairs.

## 2.4 Numerical experiments

In this section, we choose the first companion form  $C_1$  and investigate the bound of  $\frac{\eta(P, \lambda, \mathbf{x})}{\eta(L, \lambda, \mathbf{z})}$ . We use the MATLAB command `eig` to compute all eigenpairs of  $L(\lambda)$ . All test problems are shown in Table 2.1 from nonlinear eigenvalue problems [1]. All the computations were performed using MATLAB 2016.

As shown in Table 2.2, the maximum value of  $\frac{\eta(P, \lambda, \mathbf{x})}{\eta(L, \lambda, \mathbf{z})}$  is not too large when  $\rho$  is not too larger than 1. Therefore,  $\rho$  can investigate the bound of backward error of  $\frac{\eta(P, \lambda, \mathbf{x})}{\eta(L, \lambda, \mathbf{z})}$ . The backward error  $\eta(P, \lambda, \mathbf{x})$  may be large when  $\rho$  is too large.

Table 2.1: Polynomial eigenvalue problems [1].

Problem	$n$	applications
sleeper	200	A Vibration analysis of a railtrack
spring	200	A damped mass-spring system
power_plant	8	A Nuclear power plant problem.
hospital	24	A building model

Table 2.2: The maximum value of  $\frac{\eta(P, \lambda, \mathbf{x})}{\eta(L, \lambda, \mathbf{z})}$  and value of  $\rho$ 

Problem	$\max \frac{\eta(P, \lambda, \mathbf{x})}{\eta(L, \lambda, \mathbf{z})}$	$\rho$
sleeper	10.7	17.0
spring	48.4	49.9
power_plant	$1 \times 10^{10}$	$1 \times 10^4$
hospital	$2.3 \times 10^3$	$8.0 \times 10^3$

Figures 2.1,2.2 show that the backward errors  $\eta(P, \lambda, \mathbf{x})$  close to backward errors  $\eta(L, \lambda, \mathbf{z})$  when  $\rho$  is not far from 1. Figures 2.3, 2.4 display the backward errors  $\eta(P, \lambda, \mathbf{x})$  are large when  $\rho \gg 1$ , we also find that the backward errors  $\eta(L, \lambda, \mathbf{z})$  are small but  $\eta(P, \lambda, \mathbf{x})$  are large in power\_plant.

## 2.5 Conclusion

In this Chapter, we introduce the definition of backward error of  $P(\lambda)$  and backward error of  $L(\lambda)$ . We also introduce the bound of backward error of  $P(\lambda)$  relative to that of  $L(\lambda)$  and a quantity  $\rho$ . We can use the quantity of  $\rho$  to investigate the backward error of  $P(\lambda)$ . We find that if  $\rho \gg 1$ , the eigensolver is unstable and the backward errors of  $P(\lambda)$  are large.

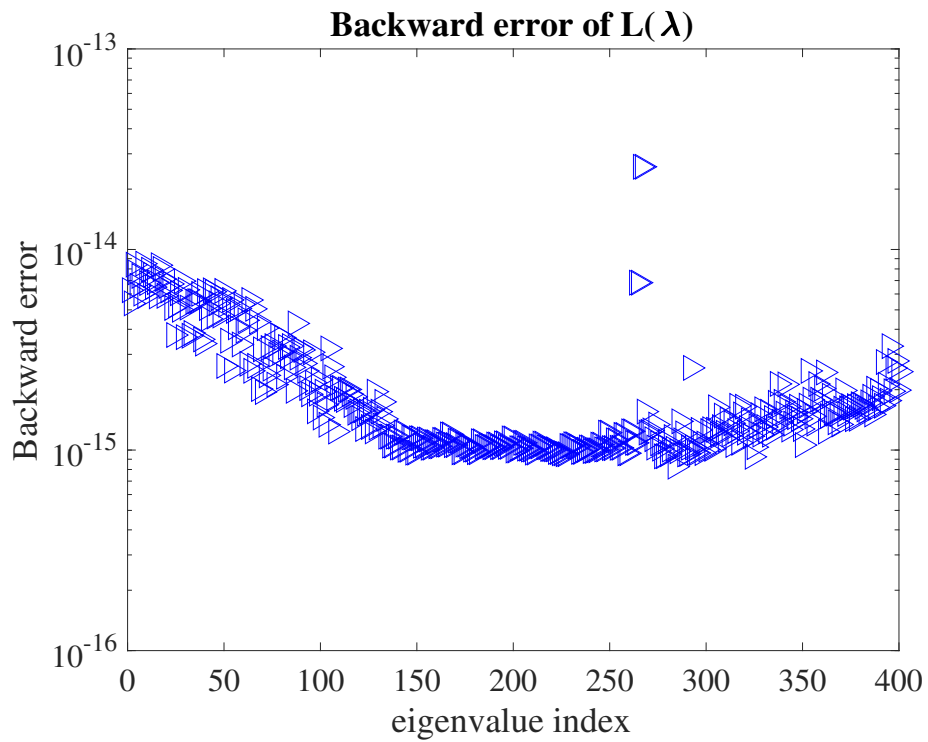


Figure 2.1: Backward error  $\eta(L, \lambda, \mathbf{z})$  for the sleeper problem.

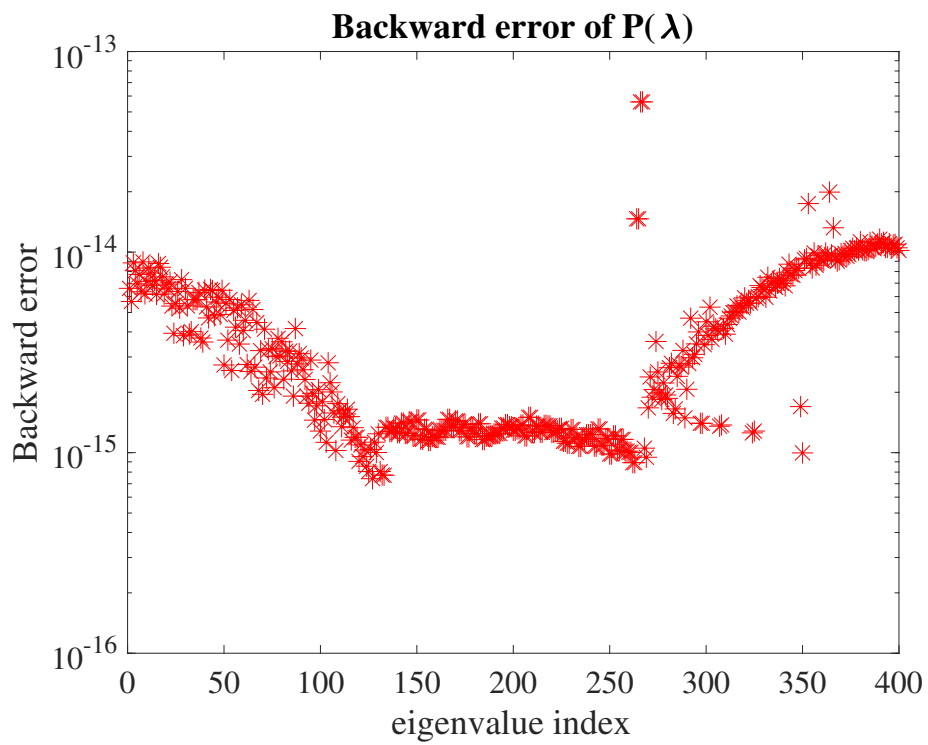


Figure 2.2: Backward error  $\eta(P, \lambda, \mathbf{x})$  for the sleeper problem.

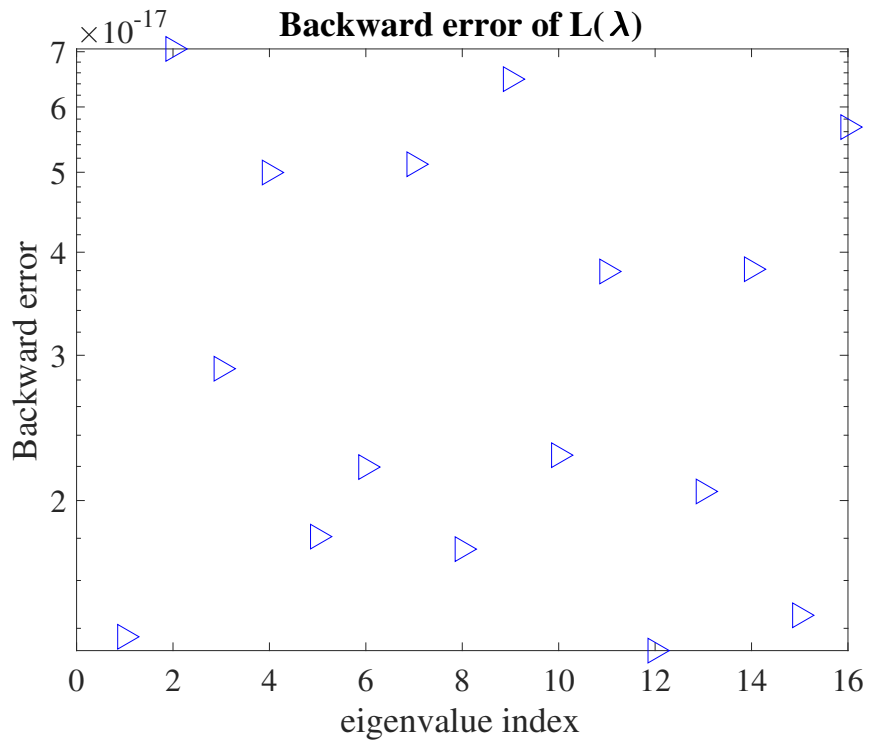


Figure 2.3: Backward error  $\eta(L, \lambda, \mathbf{z})$  for the `power_plant` problem.

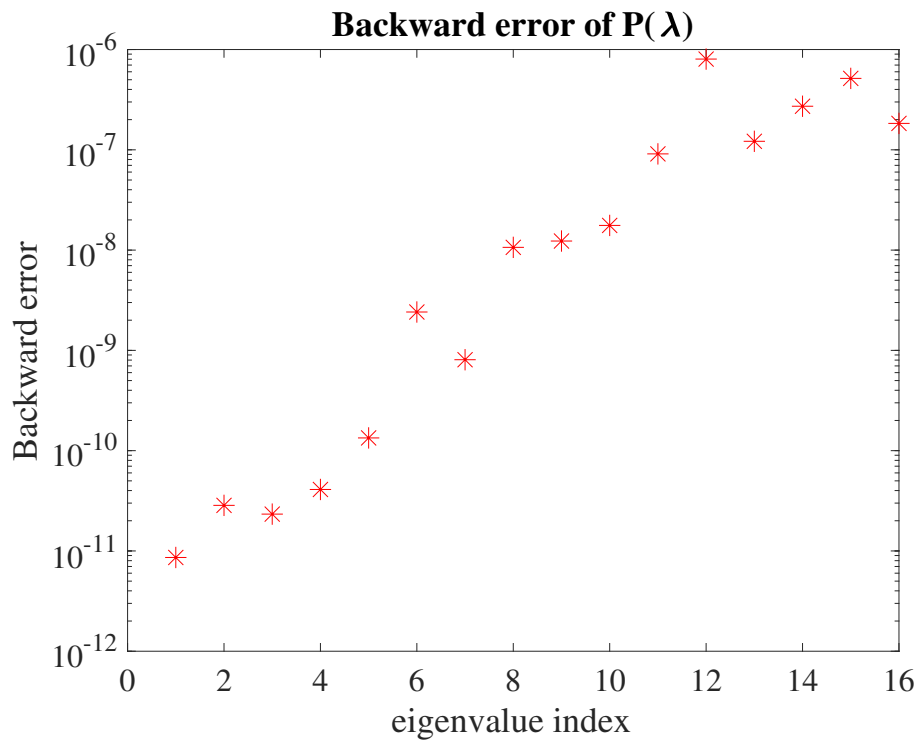


Figure 2.4: Backward error  $\eta(P, \lambda, \mathbf{x})$  for the `power_plant` problem.

## Chapter 3

# Scaling technique for Sakurai-Sugiura method in quadratic eigenvalue problem

In this chapter, we introduce Sakurai-Sugiura method with scaling technique for solving quadratic eigenvalue problem. The main content of this chapter is

- We combine the SS-RR method with scaling technique and reduce the backward error of the SS-RR method in QEP.
- We analyze the backward error of the SS-RR method in QEP and give some theorems to explain the reason that the SS-RR method with scaling technique can improve the backward error of computing eigenpairs in QEP.

This chapter is based on the paper *H. Chen, Y. Maeda, A. Imakura, T. Sakurai, F. Tisseur: Improving the numerical stability of the Sakurai-Sugiura method for quadratic eigenvalue problems. JSIAM Letters. 9 (2017), 17–20.*



### 3.1 Introduction

We define quadratic eigenvalue problem (QEP) as following

$$Q(\lambda)\mathbf{x} = (\lambda^2 A_2 + \lambda A_1 + A_0)\mathbf{x} = \mathbf{0}, \quad A_2, A_1, A_0 \in \mathbb{C}^{n \times n} \setminus \{O\}. \quad (3.1)$$

The QEP is to find  $\lambda \in \mathbb{C}$  and non-zero vectors  $\mathbf{x} \in \mathbb{C}^n \setminus \{\mathbf{0}\}$  that satisfy  $Q(\lambda)\mathbf{x} = \mathbf{0}$ . If  $(\lambda, \mathbf{x})$  are satisfied  $Q(\lambda)\mathbf{x} = \mathbf{0}$ ,  $\lambda$  are eigenvalues and  $\mathbf{x}$  are associated eigenvectors for  $Q(\lambda)\mathbf{x} = \mathbf{0}$ .

QEP appears in many models of structures [2], for example, the building model and earthquake vibration analysis. In some applications, it is unnecessary to compute all eigenpairs, and we only need partial eigenpairs  $(\lambda, \mathbf{x})$ .

To compute partial eigenvalues, we introduce a efficient method which called Sakurai-Sugiura method. Based on contour integral theorem, Sakurai-Sugiura method reduces the dimension of original coefficient matrices [18] and compute partial eigenpairs for target eigenvalue problems. Sakurai-Sugiura method has some types for solving quadratic eigenvalue problem. In this chapter, we consider Rayleigh-Ritz type of the Sakurai-Sugiura method [21] which called SS-RR method. This idea of the SS-RR method is to only compute target eigenvalues for locating in a given curve  $\Gamma$  based on contour integral theorem. Then we reduce the dimension of the target eigenvalue problem  $Q(\lambda)$  to a small dimension projected problem, such that

$$R(\lambda) = V^H Q(\lambda) V = \lambda^2 R_2 + \lambda R_1 + R_0 \quad (3.2)$$

Here, the non-square matrix  $V \in \mathbb{C}^{n \times m}$  with  $m \ll n$  which has orthonormal columns vectors and this matrix  $V$  is obtained from singular decomposition step in the SS-RR method.

We define  $(\widehat{\lambda}, V\widehat{\mathbf{y}})$  as an approximate eigenpair of  $Q(\lambda)$ . We also let  $(\widehat{\lambda}, \widehat{\mathbf{y}})$  be an approximate eigenpair of  $R(\lambda)$ .

However, this approach does not accurately compute the eigenpairs of  $Q(\lambda)$  when there is large variation in the magnitude of the coefficient matrices of  $R(\lambda)$ . To avoid this disadvantage, we

proposed a stable eigensolver for improving the accuracy of computing target eigenvalues in projected QEP (3.2) and hope to improve the numerical stability for the SS-RR method for solving the quadratic eigenvalue problem.

This chapter is organized as follows.

- **Section 3.2.** We introduce the definition of scaling technique and give a review for several types of scaling techniques.
- **Section 3.3.** We introduce the proposed method which combine the SS-RR method with a stable projected eigensolver, such as `quadeig`.
- **Section 3.4.** We give a theoretical analysis for the backward error of the SS-RR method.
- **Section 3.5.** Some numerical experiments are shown in this section. We investigate our theorem by numerical experiments.
- **Section 3.6.** We make a conclusion for this work and show some future works.

## 3.2 Scaling technique

Scaling technique is to find two parameters  $\gamma$  and  $\mu$  and convert  $Q(\lambda) = \lambda^2 A_2 + \lambda A_1 + A_0$  to

$$\tilde{Q}(\mu) = \mu^2 \tilde{A}_2 + \mu \tilde{A}_1 + \tilde{A}_0.$$

where  $\tilde{A}_2 = \delta\gamma^2 A_2$ ,  $\tilde{A}_1 = \delta\gamma A_1$ ,  $\tilde{A}_0 = \delta A_0$  and  $\lambda = \mu\gamma$ ,  $\gamma \neq 0$ .

We compute eigenpairs of  $\tilde{Q}(\mu)$  using a linearization form  $L(\lambda)$  with QZ method. The aim is to improve the backward error of  $L(\lambda)$  with scaling techniques and the backward error of  $Q(\lambda)$  obtain from  $L(\lambda)$  with scaling techniques.

### 3.3 The proposed method

In this section, we introduce how to combine the SS-RR method with scaling techniques.

We now discuss why the numerical solution of the QEP in step 5 requires special attention.

The standard way of solving small to medium size dense QEPs,

$$R(\lambda)\mathbf{y} = (\lambda^2 R_2 + \lambda R_1 + R_0)\mathbf{y} = \mathbf{0}, \quad (3.3)$$

is via linearization. We assume  $R_2, R_1, R_0 \in \mathbb{C}^{m \times m} \setminus \{O\}$ . This consists of rewriting (3.3) as a generalized eigenvalue problem (GEP)  $L(\lambda)\mathbf{v} = \mathbf{0}$  of twice the dimension, where for example,

$$L(\lambda) = \lambda \begin{bmatrix} R_2 & 0 \\ 0 & I \end{bmatrix} + \begin{bmatrix} R_1 & R_0 \\ -I & 0 \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} \lambda \mathbf{y} \\ \mathbf{y} \end{bmatrix},$$

then solve the GEP with the QZ algorithm, and finally recover the eigenvectors  $\mathbf{y}$  of  $R(\lambda)$  from the eigenvectors  $\mathbf{v}$  of  $L(\lambda)$ .

Despite the fact that the QZ algorithm is backward stable for GEPs, it can be backward unstable for QEPs, in particular when the norms of the coefficient matrices of  $R(\lambda)$  vary widely [14]. As a result, the computed eigenpairs of  $Q(\lambda)$  may not be the exact eigenpairs of a nearby quadratic. Scaling of the eigenvalue parameter (e.g.,  $\lambda = \gamma\mu$ ,  $\mu$  being the new eigenvalue) has been shown to improve the backward stability of the solution process [32, 29, 28]. Such scaling has been implemented in the eigensolver `quadeig` [33]. The latter offers three types of eigenvalue parameter scalings:

- Fan, Lin, and Van Dooren scaling:  $\gamma_{FLV} = \sqrt{\frac{\|R_0\|_2}{\|R_2\|_2}}$ ,
- tropical scaling with largest root,  $\gamma_{trop}^+ = \frac{\|R_1\|_2}{\|R_2\|_2}$ ,
- tropical scaling with smallest root  $\gamma_{trop}^- = \frac{\|R_0\|_2}{\|R_1\|_2}$ .

The Fan, Lin, and Van Dooren scaling [32] is employed by default in `quadeig` for QEPs that

are not too heavily damped, i.e., when

$$\|R_1\|_2 \leq \sqrt{\|R_2\|_2 \|R_0\|_2}. \quad (3.4)$$

In that case, an eigenpair  $(\widehat{\lambda}, \widehat{\mathbf{y}})$  computed by `quadeig` is guaranteed to have a small backward error. Recall from [14] that the backward error of an approximate eigenpair  $(\widehat{\lambda}, \widehat{\mathbf{y}})$  of  $R(\lambda)$  in (3.2) can be defined by

$$\begin{aligned} \eta(R, \widehat{\lambda}, \widehat{\mathbf{y}}) &:= \min\{\epsilon : (R(\widehat{\lambda}) + \Delta R(\widehat{\lambda}))\widehat{\mathbf{y}} = \mathbf{0}, \\ &\quad \|\Delta R_i\|_2 \leq \epsilon \|R_i\|_2, \quad i = 0, 1, 2\}, \end{aligned}$$

where  $\Delta R(\lambda) = \lambda^2 \Delta R_2 + \lambda \Delta R_1 + \Delta R_0$  is a perturbation of  $R(\lambda)$ .

For heavily damped QEPs, it follows from [31, Thm. 2] that when  $R_2$  and  $R_1$  are well conditioned, and  $\gamma_{trop}^-/\gamma_{trop}^+$  is small enough then there are precisely  $m$  eigenvalues of the  $m \times m$  quadratic matrix polynomial  $R(\lambda)$  with moduli of the order of  $\gamma_{trop}^+$ . Similarly, when  $R_1$  and  $R_0$  are both well conditioned, the moduli of the  $m$  smallest eigenvalues of  $R(\lambda)$  are close to the smallest tropical root  $\gamma_{trop}^-$ . Then `quadeig` with tropical scaling with largest root (respectively smallest root) guarantees to return computed eigenpairs  $(\widehat{\lambda}, \widehat{\mathbf{y}})$  with small backward errors for those eigenvalues  $\widehat{\lambda}$  of moduli close to  $\gamma_{trop}^+$  (respectively,  $\gamma_{trop}^-$ ).

Based on the above comments, we propose to use `quadeig` in step 5 of Algorithm 4 to solve the projected QEP  $R(\lambda)\mathbf{y} = \mathbf{0}$ . We use the Fan, Lin and Van Dooren scaling for not too heavily damped QEPs, i.e., when the matrix coefficients of  $R(\lambda)$  satisfy (3.4). For heavily damped QEPs, we use tropical scaling: we choose  $\gamma_{trop}^+$  if we are interested in the eigenvalues of large magnitude and  $\gamma_{trop}^-$  otherwise.

Note that we could have used the backward stable eigensolver for QEPs presented in [34], but the latter is not freely available unlike `quadeig`.

### 3.4 Analysis of the backward errors of eigenpairs computed by the proposed method

In this section, we investigate why the use of a backward stable eigensolver in step 5 of Algorithm 4 improves the backward stability of the SS-RR algorithm. We will make use of the explicit and computable expression for the backward error  $\eta(R, \hat{\lambda}, \hat{\mathbf{y}})$  given in [14]:

$$\eta(R, \hat{\lambda}, \hat{\mathbf{y}}) = \frac{\|R(\hat{\lambda})\hat{\mathbf{y}}\|_2}{(\sum_{i=0}^2 |\hat{\lambda}|^i \|R_i\|_2) \|\hat{\mathbf{y}}\|_2}. \quad (3.5)$$

Let  $(\tilde{\lambda}, \tilde{\mathbf{y}})$  and  $(\hat{\lambda}, \hat{\mathbf{y}})$  be approximations to the same eigenpair  $(\lambda, \mathbf{y})$  of  $R(\lambda)$  in (3.2). Assume that  $(\tilde{\lambda}, \tilde{\mathbf{y}})$  computed by a stable eigensolver and  $(\hat{\lambda}, \hat{\mathbf{y}})$  computed by an unstable eigensolver are such that

$$\eta(R, \tilde{\lambda}, \tilde{\mathbf{y}}) \leq \eta(R, \hat{\lambda}, \hat{\mathbf{y}}). \quad (3.6)$$

In what follows we identify a sufficient condition under which (3.6) implies that

$$\eta(Q, \tilde{\lambda}, V\tilde{\mathbf{y}}) \leq \eta(Q, \hat{\lambda}, V\hat{\mathbf{y}}). \quad (3.7)$$

We will need the following lemma.

**Lemma 3.1.** *Let  $(\hat{\lambda}, \hat{\mathbf{y}})$  be an approximate eigenpair of  $R(\lambda)$  in (3.2) with  $\hat{\mathbf{y}}$  normalized so that  $\|\hat{\mathbf{y}}\|_2 = 1$ . Then for the approximate eigenpair  $(\hat{\lambda}, V\hat{\mathbf{y}})$  of  $Q(\lambda)$  we have*

$$L_1(\hat{\lambda})\eta(R, \hat{\lambda}, \hat{\mathbf{y}}) \leq \eta(Q, \hat{\lambda}, V\hat{\mathbf{y}}) \leq L_2(\hat{\lambda}, \hat{\mathbf{y}})\eta(R, \hat{\lambda}, \hat{\mathbf{y}}),$$

where

$$L_1(\widehat{\lambda}) := \frac{(\sum_{i=0}^2 |\widehat{\lambda}|^i \|V^H A_i V\|_2)}{(\sum_{i=0}^2 |\widehat{\lambda}|^i \|A_i\|_2)},$$

$$L_2(\widehat{\lambda}, \widehat{\mathbf{y}}) := \frac{\|Q(\widehat{\lambda})V\widehat{\mathbf{y}}\|_2}{\|R(\widehat{\lambda})\widehat{\mathbf{y}}\|_2}.$$

*Proof.* This follows directly from (3.5) and  $\|V\widehat{\mathbf{y}}\|_2 = \|\widehat{\mathbf{y}}\|_2 = 1$ . Note that  $\|R(\widehat{\lambda})\widehat{\mathbf{y}}\|_2 \neq 0$  since  $(\widehat{\lambda}, \widehat{\mathbf{y}})$  is not an eigenpair of  $R(\lambda)$ .  $\square$

Based on Lemma 3.1, we have the following theorem.

**Theorem 3.1.** *Let  $(\widetilde{\lambda}, \widetilde{\mathbf{y}})$  and  $(\widehat{\lambda}, \widehat{\mathbf{y}})$  be approximations to an eigenpair  $(\lambda, \mathbf{y})$  of the QEP  $R(\lambda)$  in (3.2). Let  $\alpha \geq 1$  be such that  $\eta(R, \widehat{\lambda}, \widehat{\mathbf{y}}) = \alpha\eta(R, \widetilde{\lambda}, \widetilde{\mathbf{y}})$  and let the functions  $L_1, L_2$  be defined as in Lemma 3.1. If*

$$\kappa := \alpha L_1(\widehat{\lambda})/L_2(\widetilde{\lambda}, \widetilde{\mathbf{y}}) \geq 1 \quad (3.8)$$

*then the inequality (3.7) holds for the approximate eigenpairs  $(\widetilde{\lambda}, V\widetilde{\mathbf{y}})$  and  $(\widehat{\lambda}, V\widehat{\mathbf{y}})$  of  $Q(\lambda)$ .*

*Proof.* Based on Lemma 3.1, we have

$$\begin{aligned} \eta(Q, \widehat{\lambda}, V\widehat{\mathbf{y}}) &\geq L_1(\widehat{\lambda})\eta(R, \widehat{\lambda}, \widehat{\mathbf{y}}) \\ &= L_1(\widehat{\lambda}) \frac{\eta(R, \widehat{\lambda}, \widehat{\mathbf{y}})}{\eta(R, \widetilde{\lambda}, \widetilde{\mathbf{y}})} \eta(R, \widetilde{\lambda}, \widetilde{\mathbf{y}}) \\ &\geq \alpha \frac{L_1(\widehat{\lambda})}{L_2(\widetilde{\lambda}, \widetilde{\mathbf{y}})} \eta(Q, \widetilde{\lambda}, V\widetilde{\mathbf{y}}) \\ &= \kappa \eta(Q, \widetilde{\lambda}, V\widetilde{\mathbf{y}}). \end{aligned}$$

The inequality (3.7) holds because  $\kappa \geq 1$ .  $\square$

Let us examine the condition in (3.8). It is easy to show that  $L_1(\widehat{\lambda}) \leq 1$  and that  $L_2(\widetilde{\lambda}, \widetilde{\mathbf{y}}) \geq 1$  so that  $L_1(\widehat{\lambda})/L_2(\widetilde{\lambda}, \widetilde{\mathbf{y}}) \leq 1$ . But if the projection  $V$  does not change much the norms of the coefficient matrices of  $Q(\lambda)$ , that is,  $\|A_i\|_2 \approx \|V^H A_i V\|_2$  then  $L_1(\widehat{\lambda}) \approx 1$ . Also, if the norm

of the residual for the approximate eigenpair  $(\tilde{\lambda}, \tilde{\mathbf{y}})$  of  $R(\lambda)$  is small then we can expect the norm of the residual  $Q(\tilde{\lambda})V\tilde{\mathbf{y}}$  to be small as well so that  $L_2(\tilde{\lambda}, \tilde{\mathbf{y}}) \approx 1$ . Since  $\alpha \geq 1$  then (3.7) is likely to hold. So what Theorem 3.1 says is that if we can improve the backward error for the approximate eigenpairs of  $R(\lambda)$  then we can improve the backward error for the approximate eigenpairs of  $Q(\lambda)$ . This justifies the use of a numerically stable eigensolver in step 5 of Algorithm 4.

## 3.5 Numerical experiments

We now compare the numerical stability of the SS-RR method with either `quadeig` and the choice of scaling discussed in Section 3.3 or `polyeig` to perform step 5 of Algorithm 4. The MATLAB function `polyeig` solves polynomial eigenvalue problems of arbitrary degree and hence is more general than `quadeig`. However, it does not employ any scaling and can suffer from numerical instability. All the computations are performed using MATLAB 2015.

The test problems listed in Table 3.1 are QEPs belonging to the collection of nonlinear eigenvalue problems NLEVP [1] and are selected so as to have large variations in the norms of their coefficient matrices. The `mod_spring` and `mod_sleeper` problems correspond to the `spring` and `sleeper` problems in [1] but with the damping matrix  $A_1$  multiplied by  $10^2$ .

For each problem, the Jordan curve  $\Gamma$  is a circle of center  $c$  and radius  $r$ , whose values are given in Table 3.2. For the parameters  $N, K, L$  we use  $N = 32$ ,  $K = 8$  and  $L = 16$ . For the quadrature points and corresponding weights we use

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

As shown in Table 3.3, the norms of the coefficient matrices of the projected problems vary widely. The projected `damped_beam` and `wiresaw2` problems are not too heavily damped since  $\|V^H A_1 V\|_2^2 \leq \|V^H A_2 V\|_2 \|V^H A_0 V\|_2$ . Hence, for these two problems, `quadeig` is called with the Fan, Lin and Van Dooren scaling  $\gamma_{FLV}$ . The projected `mod_spring` and `mod_sleeper`

Table 3.1: List of test problems.

Problem	$n$	applications
damped_beam	400	vibration analysis
wiresaw2	500	vibration analysis of a wiresaw
mod_spring	200	damped mass-spring system
mod_sleeper	1000	A model vibration of a rail track

Table 3.2: Parameters for the SS-RR method.

Problem	center $c$	radius $r$	#eigs
damped_beam	$-2 + 2.6 \times 10^6 i$	$3 \times 10^5$	22
wiresaw2	$1.5 \times 10^3 i$	40	26
mod_spring	-5000	50	14
mod_sleeper	-1650	15	24

are overdamped. Since we are interested in computing eigenvalues of magnitude around  $10^3$  for these two problems (see Table 3.2), `quadeig` is called with tropical scaling with largest root  $\gamma_{trop}^+$  (Table 3.4 shows that  $\gamma_{trop}^+ \approx 10^3$  for both problems). With this choice of scalings, `quadeig` guarantees to return eigenpairs  $(\tilde{\lambda}, \tilde{\mathbf{y}})$  inside  $\Gamma$  with backward errors  $\eta(R, \tilde{\lambda}, \tilde{\mathbf{y}}) \approx n\mathbf{u}$ , where  $\mathbf{u}$  is the machine precision. For these problems, `polyeig` returns eigenpairs  $(\hat{\lambda}, \hat{\mathbf{y}})$  with  $\eta(R, \hat{\lambda}, \hat{\mathbf{y}}) \gg n\mathbf{u}$  so that (3.6) holds.

Table 3.4 displays the smallest value  $\kappa_{\min}$  of  $\kappa$  in (3.8), where  $\hat{\lambda}$  corresponds to eigenvalues computed by `polyeig` and  $(\tilde{\lambda}, \tilde{\mathbf{y}})$  are the corresponding eigenpairs computed by `quadeig`. Table 3.4 shows that the assumption  $\kappa > 1$  is satisfied for all the problems.

It then follows from Theorem 3.1 that the inequality (3.7) holds between eigenpairs from SS-RR with `quadeig` and SS-RR with `polyeig`. This is confirmed by the backward error results presented in Table 3.5, Fig. 4.1, and Fig. 4.2. They also show that the SS-RR method with `quadeig` and appropriate scaling computes eigenpairs  $(\tilde{\lambda}, V\tilde{\mathbf{y}})$  with backward errors  $\eta(Q, \tilde{\lambda}, V\tilde{\mathbf{y}}) \approx n\mathbf{u}$ , which is the best we can expect.



Table 3.3: Norm of the coefficient matrices of  $R(\lambda) = V^H Q(\lambda) V$ .

Problem	$\ V^H A_0 V\ _2$	$\ V^H A_1 V\ _2$	$\ V^H A_2 V\ _2$
damped_beam	$10^{10}$	$2 \times 10^{-1}$	$2 \times 10^{-3}$
wiresaw2	$10^6$	$2 \times 10^1$	$5 \times 10^{-1}$
mod_spring	25	$5 \times 10^3$	1
mod_sleeper	13	$2 \times 10^3$	1

Table 3.4: Minimum value of  $\kappa$  in (3.8), type of scaling  $\gamma$  used with `quadeig` and its value.

Problem	$\kappa_{\min}$	Parameter scaling	
		$\gamma$	value
damped_beam	1.4	$\gamma_{FLV}$	$3 \times 10^6$
wiresaw2	$1 \times 10^3$	$\gamma_{FLV}$	$2 \times 10^3$
mod_spring	$6 \times 10^2$	$\gamma_{trop}^+$	$5 \times 10^3$
mod_sleeper	$4 \times 10^3$	$\gamma_{trop}^+$	$2 \times 10^3$

## 3.6 Conclusion

We have shown that to improve the backward stability of the SS-RR method, it is crucial to combine it with a backward stable algorithm for the complete solution of the projected QEP. In future work, we plan to investigate the inclusion of scaling techniques in other types of Sakurai-Sugiura methods.

Table 3.5: Largest backward errors of eigenpairs.

Problem	SS-RR with		
	polyeig	quadeig	$nu$
damped_beam	$3 \times 10^{-9}$	$2 \times 10^{-13}$	$9 \times 10^{-14}$
wiresaw2	$6 \times 10^{-11}$	$8 \times 10^{-15}$	$1 \times 10^{-13}$
mod_spring	$4 \times 10^{-10}$	$2 \times 10^{-15}$	$4 \times 10^{-14}$
mod_sleeper	$2 \times 10^{-10}$	$5 \times 10^{-15}$	$2 \times 10^{-13}$

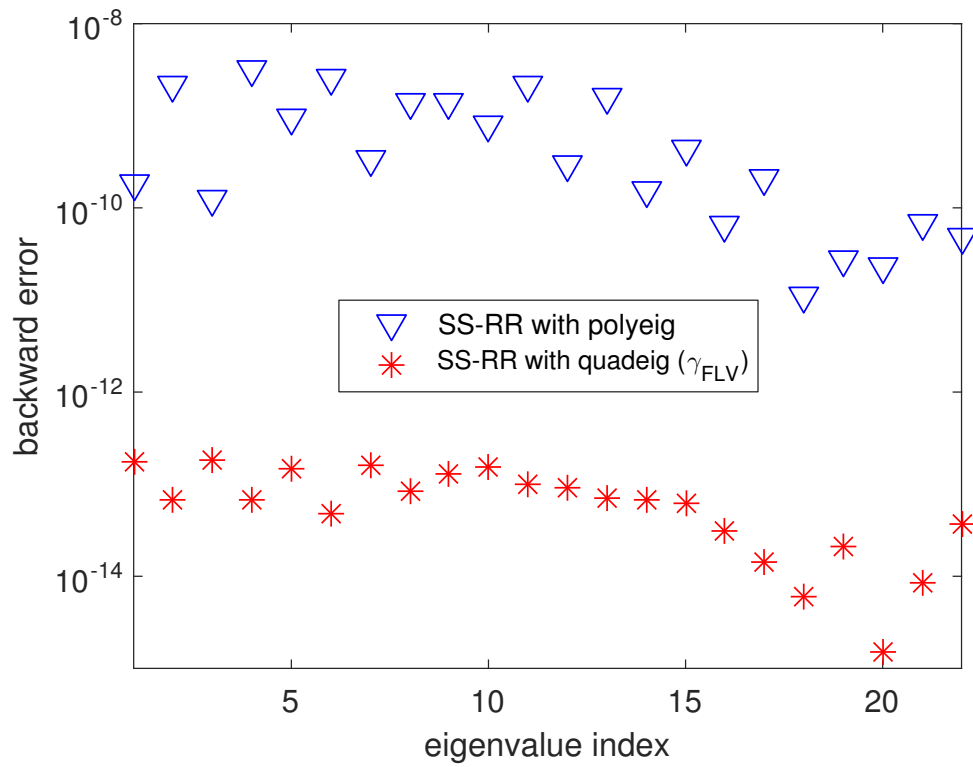


Figure 3.1: Backward errors for the damped\_beam problem.

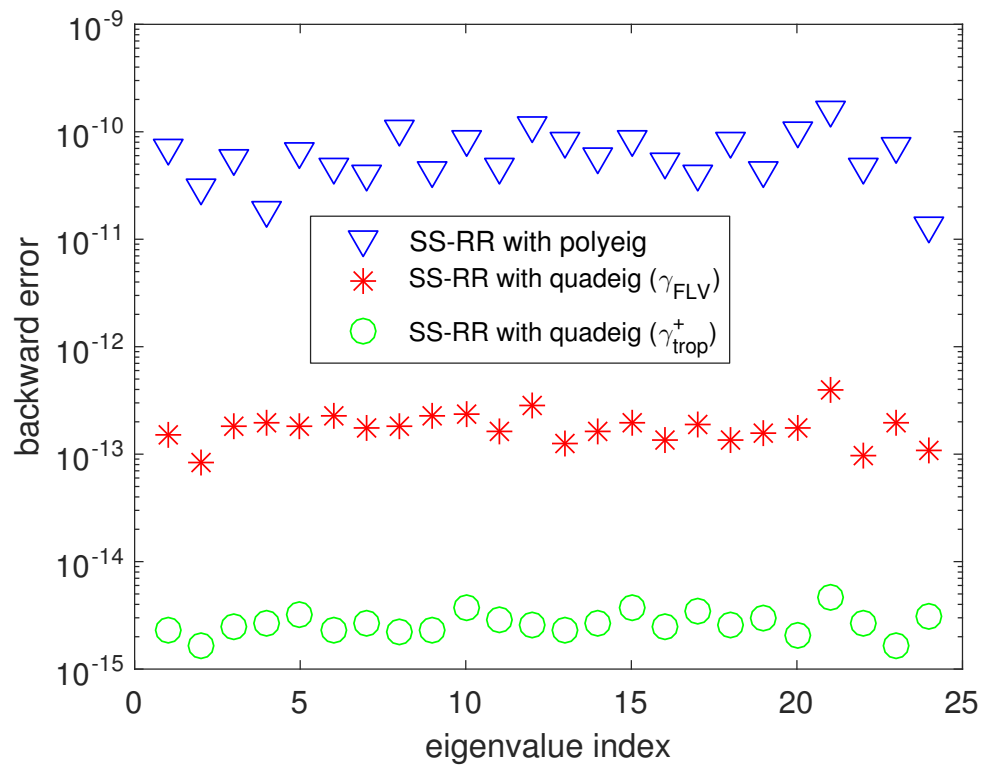


Figure 3.2: Backward errors for the mod\_sleeper problem.

## Chapter 4

# Balancing technique for Sakurai-Sugiura method in polynomial eigenvalue problem

In this chapter, we will introduce the Sakurai-Sugiura method with balancing technique for solving polynomial eigenvalue problem (PEP). The main content of this chapter is

- Introduce several types of balancing technique
- Explain the reason that the SS-RR method with balancing technique can reduce the backward error of computing eigenpairs of PEP.

This chapter is based on *H. Chen, A. Imakura and T. Sakurai, Improving backward stability of Sakurai-Sugiura method with balancing technique in polynomial eigenvalue problem, Applications of Mathematics. 62, 357–375, 2017.*

## 4.1 Introduction

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

$$P(\lambda)\mathbf{x} = \left( \sum_{i=0}^m \lambda^i A_i \right) \mathbf{x} = \mathbf{0}, \quad (4.1)$$

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.

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 [2].

In some applications, such as vibration analysis and some models of physics, it is unnecessary to compute all eigenpairs, and partial eigenpairs  $(\lambda, \mathbf{x})$  are sufficient.

The SS-RR method extracts only the eigenvalues within a Jordan curve  $\Gamma$ , using a subspace constructed with a contour integral. In the SS-RR method for the PEP [21],  $P(\lambda)$  is transformed into a projection of a matrix polynomial with a small dimension as

$$R(\lambda) = V^H P(\lambda) V, \quad (4.2)$$

where the matrix  $V \in \mathbb{C}^{n \times \ell}$ ,  $\ell \ll n$ , has orthonormal columns consisting of basis vectors for the subspace constructed by the SS-RR method. Then, the pair  $(\hat{\lambda}, V\hat{\mathbf{y}})$  is used as an approximate eigenpair for  $P(\lambda)$ , where  $(\hat{\lambda}, \hat{\mathbf{y}})$  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) [24]. Extending this idea, we improve the backward stability of the SS-RR method for the PEP using balancing technique.

One common way for solving (4.2) is to convert  $R(\lambda)$  into a GEP.  $R(\lambda)$  and the GEP have the same eigenvalues. For computing eigenvalues of  $R(\lambda)$ , we solve the GEP by some numerical stability methods. In this article, to improve the accuracy of computing eigenpairs, we consider using a balancing technique [25, 26], that is a preprocessing 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 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 Section 4.3, we introduce a linearization for solving the projected PEP. In Section 4.4, 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 4.5, 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 4.6.

## 4.2 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,

$$A\mathbf{v} = \lambda\mathbf{v}, \quad A \in \mathbb{C}^{n \times n}. \quad (4.3)$$

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

Here, we now discuss two types of balancing technique for solving SEP.

### 4.2.1 Osborne's balancing technique

Osborne proposed the use of a diagonal matrix  $D$  that minimizes the Frobenius-norm  $\|D^{-1}AD\|_F$  [25]. He showed that his technique also decreases the 2-norm, that is,  $\|A\|_2 \geq \|D^{-1}AD\|_2$  [25]. Let  $c_i$  and  $r_i$  be the  $p$ -norms of each column and row, which ignores the diagonal element of the matrix  $A$  defined by

$$col_i = \left( \sum_{j \neq i} |a_{j,i}|^p \right)^{1/p}, \quad row_i = \left( \sum_{j \neq i} |a_{i,j}|^p \right)^{1/p}. \quad (4.4)$$

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

$$g(f_i) = f_i^p col_i^p + \frac{row_i^p}{f_i^p},$$

with  $f_i = \sqrt{\frac{row_i}{col_i}}$ .

### 4.2.2 Parlett's balancing technique

Parlett and Reinsch extended Osborne's technique to any  $p$ -norm [26].

The purpose of Parlett-Reinsch algorithm is to equal the norms of columns and rows of  $A$ . The Parlett-Reinsch algorithm seeks  $f_i$  to minimize

$$g(f_i) = f_i^p col_i^p + \frac{row_i^p}{f_i^p},$$

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 5. The diagonal elements of  $D$  are obtained from the value  $f_i$  by Step 15 in Algorithm 5.

---

**Algorithm 5** Parlett-Reinsch algorithm (Balancing technique) [26, 35]

---

**Input:**A matrix  $A \in \mathbb{C}^{n \times n}$ ,  $\kappa = 2$ .**Output:**A balancing matrix  $D^{-1}AD$  and a diagonal matrix  $D$ .

- 1: Set  $D := I$ , where  $I$  is an identity matrix
  - 2: Set  $\zeta := 0$
  - 3: **while**  $\zeta = 0$  **do**
  - 4:   **for**  $i := 1, \dots, n$  **do**
  - 5:     Compute  $p$ -norm of each column and row that ignores diagonal elements of  $A$  by (4.4).
  - 6:     Set the quantity  $q_i := col_i^p + row_i^p$ ,  $f_i := 1$ .
  - 7:     **while**  $col_i < row_i/\kappa$  **do**
  - 8:        $col_i := col_i\kappa$ ,  $row_i := row_i/\kappa$ ,  $f_i := f_i \times \kappa$ .
  - 9:     **end while**
  - 10:    **while**  $col_i \geq row_i\kappa$  **do**
  - 11:      $col_i := col_i/\kappa$ ,  $row_i := row_i\kappa$ ,  $f_i := f_i/\kappa$ .
  - 12:    **end while**
  - 13:    **if**  $(col_i^p + row_i^p) < 0.95 \times q_i$  **then**
  - 14:      $\zeta := 0$ .
  - 15:     Construct the diagonal elements  $d_{ii}$  of matrix  $D$ , where  $d_{ii} := f_i \times d_{ii}$ .
  - 16:     Balance the  $i$ -th column and row of matrix  $A$  with  $A(:, i) := f_i \times A(:, i)$ ,  $A(:, i) := A(:, i)/f_i$ .
  - 17:    **end if**
  - 18:   **end for**
  - 19: **end while**
- 

### 4.3 Eigensolver for the projected PEP using linearization

We now discuss why the numerical solution of the PEP in Step 5 of Algorithm 4 requires special attention. In the SS-RR method, the standard way to solve small to medium size PEP,

$$R(\lambda)\mathbf{y} = \left( \sum_{i=0}^m \lambda^i R_i \right) \mathbf{y} = \mathbf{0}, \quad R_i = V^H A_i V \quad (4.5)$$

is via linearization. We assume  $R_i \in \mathbb{C}^{\ell \times \ell} \setminus \{O\}$ . We linearize (4.5) as follows:

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



where  $X, Y \in \mathbb{C}^{m\ell \times m\ell}$ .  $L(\lambda)$  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

$$L(\lambda) = \lambda \begin{bmatrix} R_1 & R_2 & \cdots & R_m \\ -I_\ell & O & \cdots & O \\ \vdots & \ddots & \ddots & \vdots \\ O & \cdots & -I_\ell & O \end{bmatrix} + \begin{bmatrix} R_0 & O & \cdots & O \\ O & I_\ell & \cdots & O \\ \vdots & \vdots & \ddots & \vdots \\ O & O & \cdots & I_\ell \end{bmatrix}, \quad \mathbf{z} = \begin{bmatrix} \mathbf{y} \\ \lambda \mathbf{y} \\ \vdots \\ \lambda^{m-1} \mathbf{y} \end{bmatrix}, \quad (4.6)$$

where  $I_\ell \in \mathbb{R}^{\ell \times \ell}$  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 [14].

## 4.4 The SS-RR method with the balancing technique for the PEP

As shown in Section 1.5.2, the SS-RR method extracts only eigenvalues within a Jordan curve  $\Gamma$ . However, the SS-RR method is not stable when the coefficient matrices of the projected PEP have widely varying norms.

From [28], it is clear that the backward error of  $L(\lambda)$  is reduced, then the backward error of quadratic matrix polynomial is also reduced. 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.

---

**Algorithm 6** 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  $\Gamma$ , 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: Construct  $R(\lambda) = V^H P(\lambda) V$  by step 1-5 in Algorithm 3.
  - 2: Convert the projected matrix polynomial  $R(\lambda)$  to  $L(\lambda)$ .
  - 3: Construct the SEP by (4.7) and compute eigenpairs  $(\tilde{\lambda}_j, \tilde{\mathbf{v}}_j)$  of (4.7).
  - 4: Compute eigenvalues  $\tilde{\lambda}_j$  and eigenvectors  $\tilde{\mathbf{z}}_j$  of  $L(\lambda)$  from (4.7) 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  $\tilde{\mathbf{y}}_j = \tilde{\mathbf{z}}_j(1 : \ell), j = 1, \dots, n(\Gamma)$ .
- 

#### 4.4.1 The proposed method

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

$$D^{-1}(-X^{-1}Y)D\mathbf{v} = \lambda\mathbf{v}. \quad (4.7)$$

Finally, we compute the eigenpairs of (4.7) with a backward stable method, such as the QR method. The eigenpairs of  $P(\lambda)$  are recovered from (4.7).

The SS-RR method with the balancing technique is presented in Algorithm 6.

#### 4.4.2 Analysis of the backward error for the proposed method

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 [24]. In this article, we extend the idea in [24] to solve the PEP. The analysis in [24] 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 6), 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  $(\widehat{\lambda}_j, \widehat{\mathbf{z}}_j)$  and  $(\widetilde{\lambda}_j, \widetilde{\mathbf{z}}_j)$  be the approximations of the same eigenpair  $(\lambda_j, \mathbf{z}_j)$  of  $L(\lambda)$ .  $(\widehat{\lambda}_j, \widehat{\mathbf{z}}_j)$  are computed by  $L(\lambda)$  without using the balancing technique and  $(\widetilde{\lambda}_j, \widetilde{\mathbf{z}}_j)$  are computed with  $D^{-1}(-X^{-1}Y)D\mathbf{v} = \lambda\mathbf{v}$ , where  $\widetilde{\mathbf{z}}_j = D\widetilde{\mathbf{v}}_j$ .

From (4.6), we also define  $\widehat{\mathbf{y}}_j$  and  $\widetilde{\mathbf{y}}_j$  as

$$\widehat{\mathbf{y}}_j = \widehat{\mathbf{z}}_j(1 : \ell), \quad \widetilde{\mathbf{y}}_j = \widetilde{\mathbf{z}}_j(1 : \ell).$$

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 with the SS-RR method with the balancing technique, we consider the backward error of the PEPs.

**Definition 4.1** ([14]). *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,*

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

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

The definition of backward error  $\eta(L, \widehat{\lambda}_j, \widehat{\mathbf{z}}_j)$  is similar to the definition of  $\eta(R, \widehat{\lambda}_j, \widehat{\mathbf{y}}_j)$ .

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 formula [28]:

$$\eta(R, \widehat{\lambda}_j, \widehat{\mathbf{y}}_j) = \frac{\|R(\widehat{\lambda}_j)\widehat{\mathbf{y}}_j\|_2}{(\sum_{i=0}^m |\widehat{\lambda}_j|^i \|V^H A_i V\|_2) \|\widehat{\mathbf{y}}_j\|_2}, \quad (4.8)$$

$$\eta(L, \widehat{\lambda}_j, \widehat{\mathbf{z}}_j) = \frac{\|L(\widehat{\lambda}_j)\widehat{\mathbf{z}}_j\|_2}{(|\widehat{\lambda}_j| \|X\|_2 + \|Y\|_2) \|\widehat{\mathbf{z}}_j\|_2}. \quad (4.9)$$

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

$$\eta(L, \widehat{\lambda}_j, \widehat{\mathbf{z}}_j) \geq \eta(L, \widetilde{\lambda}_j, \widetilde{\mathbf{z}}_j). \quad (4.10)$$

In the following steps, we try to identify the sufficient conditions under which (4.10) implies that

$$\eta(P, \widehat{\lambda}_j, V\widehat{\mathbf{y}}_j) \geq \eta(P, \widetilde{\lambda}_j, V\widetilde{\mathbf{y}}_j). \quad (4.11)$$

From [29], we have

$$\|G(\lambda)L(\lambda)\mathbf{z}\|_2 = \|R(\lambda)\mathbf{y}\|_2 \leq \|G(\lambda)\|_2 \|L(\lambda)\mathbf{y}\|_2.$$

If we use (4.6) to construct  $L(\lambda)$ , then  $G(\lambda)$  is given by

$$G(\lambda) = \begin{cases} \begin{bmatrix} I_\ell & -\lambda V^H A_2 V \end{bmatrix} & (m = 2) \\ \begin{bmatrix} I_\ell & -\lambda(V^H A_2 V + V^H A_3 V) & -\lambda V^H A_3 V \end{bmatrix} & (m = 3) \end{cases}$$

When  $m > 3$ , we can obtain  $G(\lambda)$  from [29]. Based on (4.8) and (4.9), to analyze the bounds for the backward error for  $R(\lambda)$  relative to  $L(\lambda)$ , we have the following theorem.

**Theorem 4.1** ([29]). *Let  $(\widehat{\lambda}_j, \widehat{\mathbf{z}}_j)$  be an approximation of the eigenpair of  $L(\lambda)$  and  $(\widehat{\lambda}_j, \widehat{\mathbf{y}}_j)$  be an approximation of the eigenpair of  $R(\lambda)$ , where  $\widehat{\mathbf{y}}_j$  is obtained from  $\widehat{\mathbf{z}}_j$  by (4.6) 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

$$\frac{\eta(R, \widehat{\lambda}_j, \widehat{\mathbf{y}}_j)}{\eta(L, \widehat{\lambda}_j, \widehat{\mathbf{z}}_j)} \leq C_U(\widehat{\lambda}_j, \widehat{\mathbf{z}}_j), \quad (4.12)$$

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  $\ell \times \ell m$  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** ([24]). *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 (4.8), we have

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

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

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

that proves Lemma 4.1. □

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

**Theorem 4.2** ([27]). *Let  $\theta_j$  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*

$$\delta_j = \theta_j \alpha_j \beta_j \geq 1, \tag{4.13}$$

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

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

From Lemma 4.1,

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

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. □

The computation of  $\delta_j$  may be complicated, because it requires  $\|A_i\|_2$ . To determine a more efficient way to compute  $\delta_j$ , we analyze (4.13) 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

$$B_L(\widehat{\lambda}_j) = \frac{(\sum_{i=0}^m |\widehat{\lambda}_j|^i \|V^H A_i V\|_2)}{(\sum_{i=0}^m |\widehat{\lambda}_j|^i \|A_i\|_2)} \geq \frac{1}{\varepsilon_1}. \quad (4.14)$$

In this case, the lower bound for  $\alpha_j$  is given by

$$\alpha_j = B_L(\widehat{\lambda}_j) \frac{\eta(R, \widehat{\lambda}_j, \widehat{\mathbf{y}}_j)}{\eta(L, \widehat{\lambda}_j, \widehat{\mathbf{z}}_j)} \geq \frac{1}{\varepsilon_1} \frac{\eta(R, \widehat{\lambda}_j, \widehat{\mathbf{y}}_j)}{\eta(L, \widehat{\lambda}_j, \widehat{\mathbf{z}}_j)}. \quad (4.15)$$

We also define

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

then  $\varepsilon_2 \geq B_U(\tilde{\lambda}_j, \tilde{\mathbf{y}}_j)$  and the lower bound of  $\beta_j$  is given by

$$\beta_j = \frac{1}{B_U(\tilde{\lambda}_j, \tilde{\mathbf{y}}_j)C_U(\tilde{\lambda}_j, \tilde{\mathbf{z}}_j)} \geq \frac{1}{\varepsilon_2 C_U(\tilde{\lambda}_j, \tilde{\mathbf{z}}_j)}. \quad (4.16)$$

Based on (4.10), (4.13), (4.15) and (4.16), the lower bound for  $\delta_j$  is given by

$$\delta_j = \theta_j \alpha_j \beta_j \geq \frac{1}{\varepsilon_1 \varepsilon_2} \frac{\eta(R, \hat{\lambda}_j, \hat{\mathbf{y}}_j)}{\eta(L, \hat{\lambda}_j, \hat{\mathbf{z}}_j)} \frac{1}{C_U(\tilde{\lambda}_j, \tilde{\mathbf{z}}_j)}. \quad (4.17)$$

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

$$\|A_i\|_2 \approx \|V^H A_i V\|_2. \quad (4.18)$$

We have

$$\varepsilon_1 \approx 1. \quad (4.19)$$

Next, we analyze the parameter  $\varepsilon_2$ . If  $\mathcal{R}(V)$  is an invariant subspace with respect to  $P(\tilde{\lambda}_j)$ , i.e., there is  $Q(\tilde{\lambda}_j)$  such that  $P(\tilde{\lambda}_j)V = VQ(\tilde{\lambda}_j)$ , then we have

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

and

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

Therefore,  $\|P(\tilde{\lambda}_j)V\tilde{\mathbf{y}}_j\|_2 = \|V^H P(\tilde{\lambda}_j)V\tilde{\mathbf{y}}_j\|_2$ . In the SS-RR method,  $V$  is constructed as an



approximation of invariant subspace with respect to the target eigenpairs. Based on this, we may assume

$$B_U(\tilde{\lambda}_j, \tilde{\mathbf{y}}_j) = \frac{\|P(\tilde{\lambda}_j)V\tilde{\mathbf{y}}_j\|_2}{\|V^H P(\tilde{\lambda}_j)V\tilde{\mathbf{y}}_j\|_2} \approx 1, \quad (4.20)$$

and thus  $\varepsilon_2$  is close to 1. Using these assumptions, the lower bound for  $\delta_j$  is given by

$$\delta_j = \theta_j \alpha_j \beta_j \gtrsim \frac{\eta(R, \hat{\lambda}_j, \hat{\mathbf{y}}_j)}{\eta(L, \hat{\lambda}_j, \hat{\mathbf{z}}_j) C_U(\tilde{\lambda}_j, \tilde{\mathbf{z}}_j)}. \quad (4.21)$$

Thus, if

$$\tau_j = \frac{\eta(R, \hat{\lambda}_j, \hat{\mathbf{y}}_j)}{\eta(L, \hat{\lambda}_j, \hat{\mathbf{z}}_j) C_U(\tilde{\lambda}_j, \tilde{\mathbf{z}}_j)} \geq 1, \quad (4.22)$$

we have

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

The parameter  $\tau_j$  in (4.22) can be computed with low cost, although it may sometimes happen that  $\delta_j \geq 1 > \tau_j$ .

## 4.5 Numerical experiments

In this section, we compare the backward error of the SS-RR method in combination with the balancing technique (Algorithm 6) with the backward error of the standard implementation of the SS-RR method (Algorithm 4). For Algorithms 4 and 6, we use (4.6) to construct the linearized form of  $R(\lambda)$ . We use the MATLAB command `balance` to balance the coefficient matrix of the SEP in step 4 of Algorithm 6. The MATLAB command `balance` implements Algorithm 5.

The test problems (shown in Table 4.1) are PEP of degree  $m = 2$  and higher-order PEP belonging to the collection of nonlinear eigenvalue problems NLEVP [1]. The problems `orr_sommerfeld` and `mod_butterfly` are higher-order PEP. Other problems are PEP of degree  $m = 2$ .

Table 4.1: Polynomial eigenvalue problems [1].

Problem	$n$	applications
damped_beam	400	A vibration model of damped beam
shaft	400	A vibration model of shaft
wiresaw1	400	A vibration model of a wiresaw
wiresaw2	400	A vibration model of wiresaw with viscous damping
sleeper	400	A vibration model of a railtrack with sleepers
spring	400	A finite element model of a mass-spring system with a damper
dirac	400	A model for Dirac operator
acoustic_wave_1d	400	A vibration model from acoustic wave problem
plasma_drift	128	A model from a cubic polynomial eigenvalue problem
orr_sommerfeld	400	A model from qrr-sommerfeld equation
mod_butterfly	400	A model of T-even structure

Table 4.2: Parameters for the SS-RR method.

Problem	center $\gamma$	radius $\rho$	#eigs
damped_beam	$-2 + 2.6 \times 10^6 i$	$3 \times 10^5$	22
shaft	$2 \times 10^5 i$	$9 \times 10^4$	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	$3.8 \times 10^{-4} i$	$0.4 \times 10^{-4}$	20

For each problem, the Jordan curve  $\Gamma$  is a circle with center  $\gamma$  and radius  $\rho$  whose values are given in Table 4.2. We set  $N = 32, K = 6$  and  $L = 12$  for the problem `plasma_drift`. We use  $N = 32, K = 8$  and  $L = 16$  for other problems. For the quadrature points and corresponding weights, we assign

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

All the computations were performed using MATLAB 2014.

Table 4.3: The value of  $\varepsilon_1 := \max_{i=0:m} \frac{\|A_i\|_2}{\|V^H A_i V\|_2}$ .

Problem	$\varepsilon_1$
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

### 4.5.1 Verification of the assumptions

Here, we verify the assumptions of (4.19), (4.20),  $\delta_j$  in (4.13) and  $\tau_j$  in (4.22) by using numerical experiments.

As shown in Table 4.3, 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 4.4 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 (4.20) is not satisfied are `damped_beam` and `orr_sommerfeld`.

Table 4.5 displays the assumption that  $\theta_j > 1$  is satisfied for all problems. Table 4.5 also shows that the assumption that  $\delta_j > 1$  is satisfied for all problems. The more practical approximation  $\tau_j$  for  $\delta_j$  is also larger than 1 for all problems except `orr_sommerfeld`, which confirms its wide applicability.

### 4.5.2 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 4.6 and Figures. 4.1-4.4, the backward errors of the SS-RR method with the balancing technique are smaller than those of the SS-RR

Table 4.4: The value of  $\varepsilon_2 := \max_j B_U(\tilde{\lambda}_j, \tilde{\mathbf{y}}_j)$ .

Problem	$\varepsilon_2$
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 4.5: The minimum value of the parameters  $\delta$ ,  $\theta$  and  $\tau$  for  $P(\lambda)$ .

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

method when  $\tau$  is larger than 1. The improvement in the backward error is significant even for `orr_sommerfeld` in spite of the bad estimate for  $\tau$ . We also find that there is almost no improvement in the `dirac` problem.

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

## 4.6 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

Table 4.6: Maximum backward errors of the eigenpairs of  $P(\lambda)$ .

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

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.

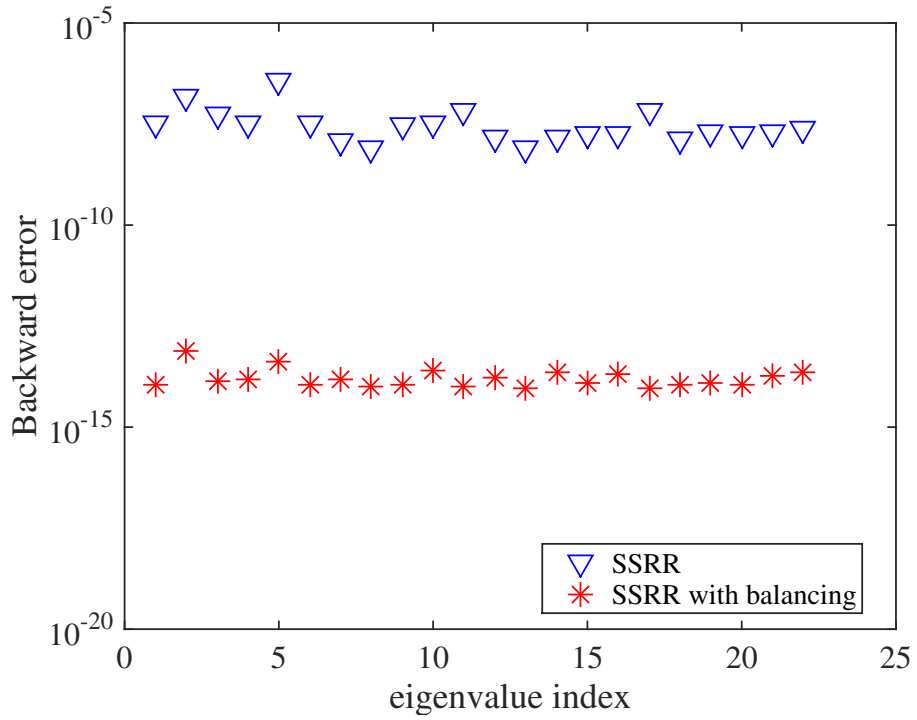


Figure 4.1: Backward error for the damped\_beam problem.

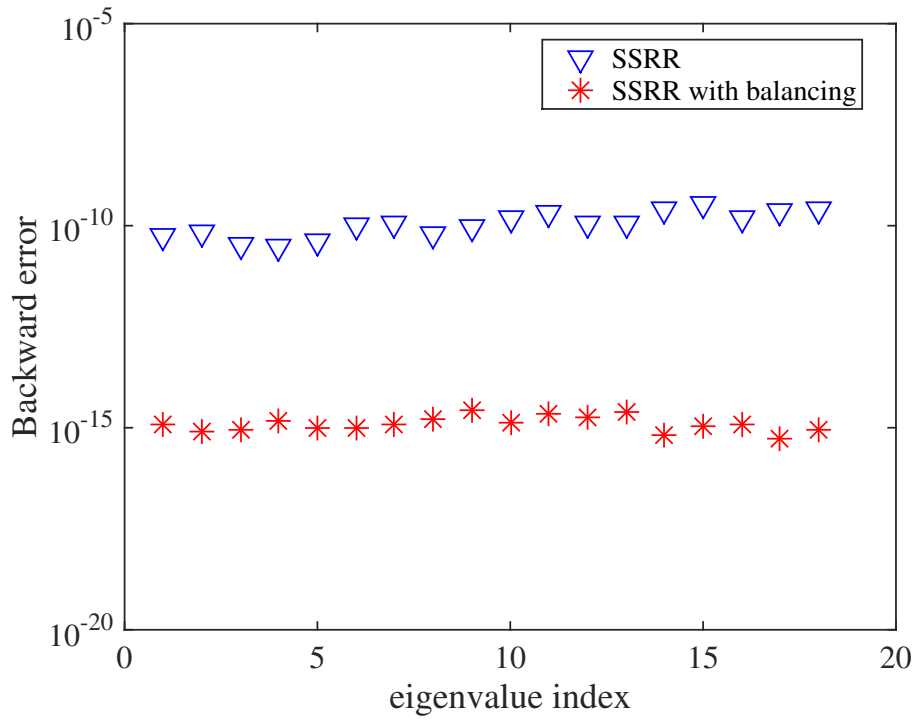
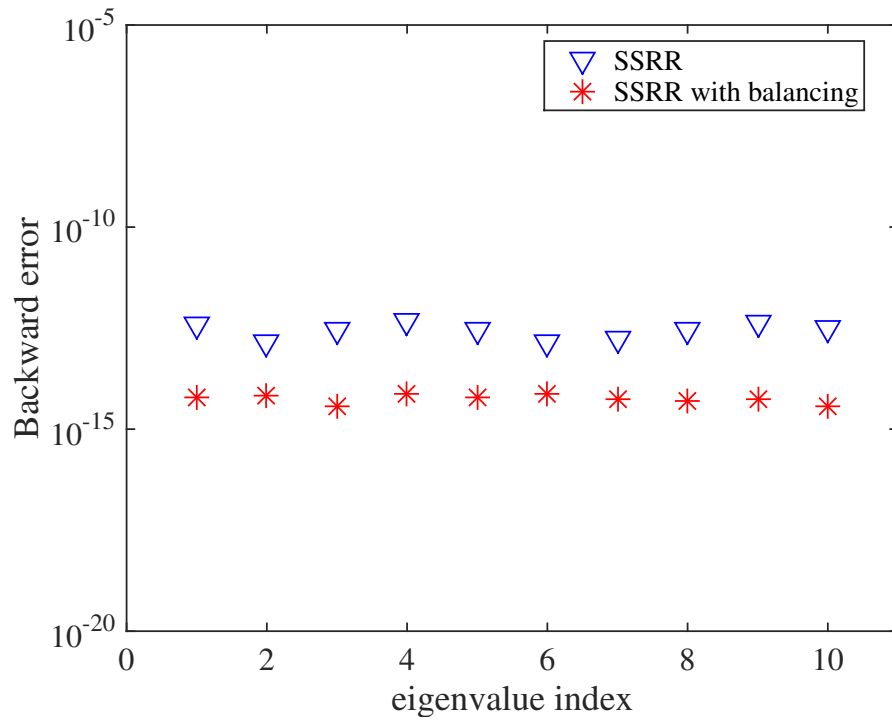
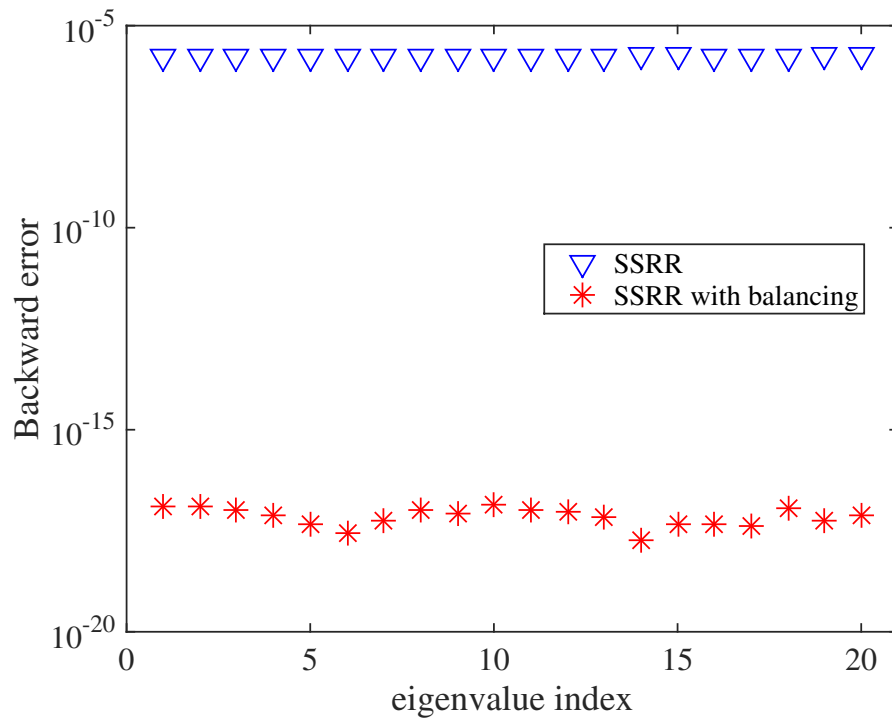


Figure 4.2: Backward error for the shaft problem.

Figure 4.3: Backward error for the `plasma_drift` problem.Figure 4.4: Backward error for the `orr_sommerfeld` problem.

# Chapter 5

## A balancing technique for heavily damped quadratic eigenvalue problem

In this chapter, we discuss a special class of quadratic eigenvalue problem which called heavily damped eigenvalue problem. The main content of this research is

- Investigate some numerical methods with different balancing techniques for solving heavily damped quadratic eigenvalue problem.
- Proposed an new approach with a balancing technique.
- Compare the backward error of the purposed method to other numerical methods by numerical experiments.

### 5.1 Introduction

We consider the heavily damped quadratic eigenvalue problem (QEP):

$$Q(\lambda)\mathbf{x} = (\lambda^2 A_2 + \lambda A_1 + A_0)\mathbf{x} = \mathbf{0} \quad (5.1)$$



where  $\|A_1\|_2 \gg \sqrt{\|A_0\|_2\|A_2\|_2}$ ,  $A_2, A_1, A_0 \in \mathbb{C}^{n \times n} \setminus \{O\}$ , and  $\lambda \in \mathbb{C}$  and  $\mathbf{x} \in \mathbb{C}^n \setminus \{\mathbf{0}\}$  are eigenvalues and their associated eigenvectors.

One common way for solving the QEP is to convert  $Q(\lambda)$  into a generalized eigenvalue problem  $L(\lambda)\mathbf{z} = \mathbf{0}$  (GEP).  $Q(\lambda)$  and  $L(\lambda)$  have the same eigenvalues [29]. We linearize (5.1) as follows:

$$L(\lambda)\mathbf{z} = (\lambda X + Y)\mathbf{z} = 0, \quad (5.2)$$

where  $X, Y \in \mathbb{C}^{2n \times 2n}$ . There are several choices for  $L(\lambda)$ . In practice, a common choice for  $L(\lambda)$  is its companion form, which is given by

$$L(\lambda) = \lambda \begin{bmatrix} A_2 & O \\ O & I_n \end{bmatrix} + \begin{bmatrix} A_1 & A_0 \\ -I_n & O \end{bmatrix}, \quad \mathbf{z} = \begin{bmatrix} \lambda \mathbf{x} \\ \mathbf{x} \end{bmatrix},$$

where  $I_n$  is an identity matrix. We compute all eigenpairs of  $L(\lambda)$  by using the QZ method. The QZ method is stable for GEP. However, it can be unstable for the QEP, especially when the norms of the coefficient matrices of  $Q(\lambda)$  vary widely [14].

The remainder of this paper is organized as follows. In Section 5.2, we introduce several balancing techniques. In Section, we show some numerical methods based on different balancing techniques and the proposed method. Finally, some numerical experiments are shown in Section and we compare several numerical methods to the proposed method.

## 5.2 Balancing technique

In Section 4.2, we review balancing techniques for solving the standard eigenvalue problems, such as Osborne and Parlett balancing techniques. Here, we will introduce another balancing techniques for solving generalized and polynomial eigenvalue problems.

### 5.2.1 Ward's balancing technique

Ward [8] proposed a balancing technique to improve accuracy of computing eigenvalues for generalized eigenvalue problem  $A\mathbf{x} = \lambda B\mathbf{x}$ . The idea of Ward is to transform two matrices  $A$  and  $B$  into  $D_1AD_2$  and  $D_1BD_2$  by finding two non-singular diagonal scaling matrices  $D_1$  and  $D_2$ . Then minimize the function

$$\min \sum_{i,j=1}^n (row_i + col_j + \log|A_{ij}|^2) + (row_i + col_j + \log|B_{ij}|^2),$$

where  $row_i$  and  $col_j$  are the the absolute values of the diagonal elements of  $D_1$  and  $D_2$ .

### 5.2.2 Lemonnier and Van Dooren's balancing technique

Lemonnier and Van Dooren's balancing technique [36] is to find two diagonal non-singular matrices  $D_1$  and  $D_2$  such that

$$\|D_1AD_2e_j\|_2^2 + \|D_1BD_2e_j\|_2^2 = \|e_i^*D_1AD_2\|_2^2 + \|e_i^*D_1BD_2\|_2^2 = 1, \quad i, j = 1, \dots, n.$$

Lemonnier and Van Dooren's balancing technique can reduce the condition number for GEP.

### 5.2.3 Betcke's balancing technique

Betcke's balancing technique [37] is to reduce the condition number for PEP by finding diagonal matrices  $D_1$  and  $D_2$

$$\sum_{k=0}^l w^{2k} \|D_1A_kD_2e_i\|_2^2 = 1, \quad \sum_{k=0}^l w^{2k} \|e_j^*D_1A_kD_2\|_2^2 = 1, \quad i, j = 1, \dots, n$$

where  $w$  close to absolute value of target eigenvalues.

## 5.3 A linearization form with balancing techniques for solving heavily damped QEP

In this section, we discuss a linearization form with balancing techniques for improving backward error in heavily damped QEP. Eigensolvers are not backward stable for solving the heavily damped QEPs when the norms of the coefficient matrices of  $Q(\lambda)$  vary widely. In order to improve the backward stability of heavily damped QEP, we proposed an new method based on Betcke's balancing technique. We also compare the backward error of the proposed method with that of a linearization form with Parlett's balancing techniques.

### 5.3.1 Linearization form for heavily damped QEP

Linearization form is a standard approach for solving QEP. Linearize  $Q(\lambda)$  as follows:

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

where  $X, Y \in \mathbb{C}^{2n \times 2n}$ ,  $L(\lambda)$  and  $Q(\lambda)$  have the same spectrum. In practice, a common choice for  $L(\lambda)$  is its companion form

$$L(\lambda)\mathbf{z} = \left( \lambda \begin{bmatrix} A_2 & O \\ O & I_n \end{bmatrix} + \begin{bmatrix} A_1 & A_0 \\ -I_n & O \end{bmatrix} \right) \mathbf{z} = \mathbf{0}, \quad \mathbf{z} = \begin{bmatrix} \lambda \mathbf{x} \\ \mathbf{x} \end{bmatrix}, \quad (5.3)$$

where  $I_n \in \mathbb{C}^{n \times n}$  is identity matrix. All eigenpairs of  $Q(\lambda)$  are computed by using the QZ method. Finally, recover the eigenvectors  $x$  of  $Q(\lambda)$  from the eigenvectors of  $L(\lambda)$ .

The main steps of this approach are shown in Algorithm 7.

---

**Algorithm 7** Linearization form for QEP

---

**Input:**

A matrix polynomial  $Q(\lambda) = \lambda^2 A_2 + \lambda A_1 + A_0$ .

**Output:**

All eigenpairs  $(\lambda, \mathbf{x})$  of  $Q(\lambda)$ .

- 1: Construct a linearization form 5.3 for  $Q(\lambda)$ .
  - 2: Compute eigenpairs  $(\lambda, \mathbf{z})$  of  $L(\lambda)$  using QZ method.
  - 3: Recover  $(\lambda, \mathbf{x})$  from  $(\lambda, \mathbf{z})$ , where  $\mathbf{x} = \mathbf{z}(1 : n)$ .
- 

---

**Algorithm 8** Linearization form with Parlett's balancing technique for QEP

---

**Input:**

A matrix polynomial  $Q(\lambda) = \lambda^2 A_2 + \lambda A_1 + A_0$ .

**Output:**

All eigenpairs  $(\lambda, \mathbf{x})$  of  $Q(\lambda)$ .

- 1: Construct a linearization form 5.3 for  $Q(\lambda)$ .
  - 2: Convert  $L(\lambda)$  to a standard eigenvalue problem.
  - 3: Balance the SEP with Betcke's balancing technique,  $D^{-1}(-X^{-1}Y)D\mathbf{v} = \lambda\mathbf{v}$ .
  - 4: Compute eigenpairs  $(\lambda, \mathbf{v})$  of the SEP with QR method and recover eigenpairs of  $(\lambda, \mathbf{z})$  from  $(\lambda, \mathbf{v})$ , where  $\mathbf{z} = D\mathbf{v}$ .
  - 5: Obtain eigenpairs of  $(\lambda, \mathbf{x})$  from  $(\lambda, \mathbf{z})$  with  $\mathbf{x} = \mathbf{z}(1 : n)$ .
- 

### 5.3.2 Linearization form with Parlett's balancing technique

Linearize  $Q(\lambda)$  to (5.3), then convert  $L(\lambda)$  to a SEP using Parlett's balancing technique

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

Compute all eigenpairs of the SEP with the QR method. Finally, recover the eigenvectors  $\mathbf{x}$  of  $Q(\lambda)$  from the eigenvectors of SEP.

### 5.3.3 Betcke's balancing technique for heavily damped QEP

Balance matrix polynomial by finding diagonal matrices  $D_1$  and  $D_2$

$$D_1(\lambda^2 A_2 + \lambda A_1 + A_0)D_2\mathbf{y} = \mathbf{0}. \quad (5.4)$$

Linearize (5.4) to a companion form

$$L(\lambda)\mathbf{z} = \left( \lambda \begin{bmatrix} D_1A_2D_2 & O \\ O & I_n \end{bmatrix} + \begin{bmatrix} D_1A_1D_2 & D_1A_0D_2 \\ -I_n & O \end{bmatrix} \right) \mathbf{z} = \mathbf{0},$$

Compute all eigenpairs of  $L(\lambda)$  using QZ method. Finally, recover the eigenvector  $\mathbf{x}$  of  $Q(\lambda)$  from the eigenvectors  $\mathbf{z}$  of  $L(\lambda)$ .

---

**Algorithm 9** Betcke's balancing technique for QEP

---

**Input:**

A matrix polynomial  $Q(\lambda) = \lambda^2A_2 + \lambda A_1 + A_0$ .

**Output:**

All eigenpairs  $(\lambda, \mathbf{x})$  of  $Q(\lambda)$ .

- 1: Balance matrix pair  $(A_2, A_1, A_0)$  to  $(D_1A_2D_2, D_1A_1D_2, D_1A_0D_2)$  with Betcke's balancing technique.
  - 2: Construct a linearization form (5.4) for  $Q(\lambda)$ .
  - 3: Compute eigenpairs  $(\lambda, \mathbf{z})$  of (5.4) using QZ method.
  - 4: Recover  $(\lambda, \mathbf{x})$  from  $(\lambda, \mathbf{z})$ , where  $\mathbf{x} = D_2\mathbf{z}(1:n)$ .
- 

### 5.3.4 The proposed method

We combine the linearization form with Betcke's balancing technique for solving heavily damped QEP. Linearize (5.1) to (5.3), then combine  $L(\lambda)$  with Betcke's balancing technique

$$D_1L(\lambda)D_2 = D_1 \left( \lambda \begin{bmatrix} A_2 & O \\ O & I_n \end{bmatrix} + \begin{bmatrix} A_1 & A_0 \\ -I_n & O \end{bmatrix} \right) D_2$$

Compute all eigenpairs of  $D_1L(\lambda)D_2$  using QZ method. Recover the eigenvectors  $\mathbf{x}$  of  $Q(\lambda)$  from the eigenvectors of  $D_1L(\lambda)D_2$ .

## 5.4 Numerical experiments

In this section, we illustrate the backward error of  $Q(\lambda)$  via Algs.9-12. The test problems are from the collection in **NLEVP** [1]. All the computations were performed using MATLAB 2016.

---

**Algorithm 10** The proposed method

---

**Input:**

A matrix polynomial  $Q(\lambda) = \lambda^2 A_2 + \lambda A_1 + A_0$ .

**Output:**

All eigenpairs  $(\lambda, \mathbf{x})$  of  $Q(\lambda)$ .

- 1: Construct a linearization form (5.3) for  $Q(\lambda)$ .
  - 2: Balance (5.3) with Betcke's balancing technique.
  - 3: Obtain eigenpairs of  $(\lambda, \mathbf{x})$  from  $(\lambda, \mathbf{z})$  with  $\mathbf{x} = D_2 \mathbf{z}(1 : n)$ .
- 

Table 5.1: Backward error ratio when  $|\lambda|$  close to  $10^{-4}$ .

Algorithms	$\max \frac{\eta(Q, \lambda, \mathbf{x})}{\eta(L, \lambda, \mathbf{z})}$	$\min \frac{\eta(Q, \lambda, \mathbf{x})}{\eta(L, \lambda, \mathbf{z})}$
Algorithm 7	$1.0 \times 10^{10}$	$9 \times 10^8$
Algorithm 8	$2.3 \times 10^8$	$1.8 \times 10^6$
Algorithm 9	$3.4 \times 10^7$	$2.8 \times 10^6$
Algorithm 10	$2.0 \times 10^0$	$3.6 \times 10^{-1}$

### 5.4.1 mod\_wiresaw2 problem

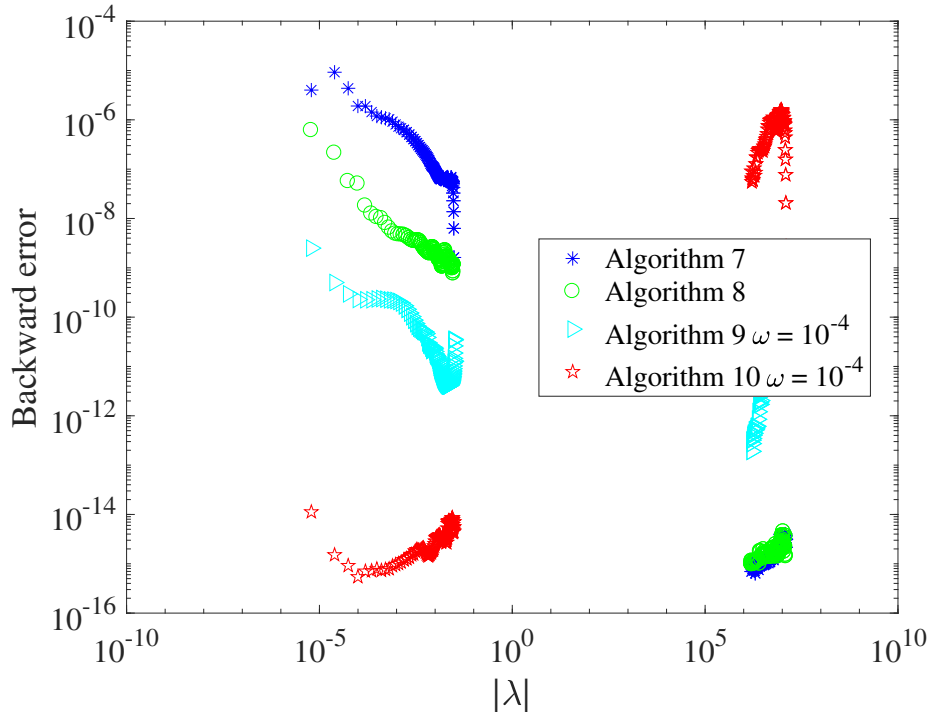
The first test problem is `mod_wiresaw2` problem from `wiresaw2` in NLEVP collection . The dimension of coefficient matrices  $A_2, A_1, A_0 \in \mathbb{C}^{200 \times 200}$ . Let  $A1 = 10^6 * A_1$  and

$$\tau = \|A_1\|_2 / \sqrt{\|A_0\|_2 \|A_2\|_2} = 2.0 \times 10^4.$$

Therefore,  $\|A_1\|_2 \gg \sqrt{\|A_0\|_2 \|A_2\|_2}$ . In Algorithms 9 and 10, we set  $w$  close to absolute value of target eigenvalues where  $w = 10^6$  and  $w = 10^{-4}$ . Fig. 5.1 show that the backward error of  $Q(\lambda)$  will be reduced by Algorithm 10 when we let  $w$  close to target eigenvalue. We would like to know what conditions need to be satisfied in order to obtain small backward errors in the solution of the QEP, we investigate the ratio  $\eta(Q, \lambda, \mathbf{x})/\eta(L, \lambda, \mathbf{z})$  are approximately equal to one . Table 5.1 and 5.2 show that Algorithm 10 reduces the ratios of the backward errors significantly. We also find that when we Algorithms 7-9 can reduce the backward error of  $Q(\lambda)$  when  $|\lambda|$  close to  $10^6$  and the ratio of  $\eta(Q, \lambda, \mathbf{x})/\eta(L, \lambda, \mathbf{z})$  are approximately equal one.

Table 5.2: Backward error ratio when  $|\lambda|$  close to  $10^6$ .

Algorithms	$\max \frac{\eta(Q,\lambda,x)}{\eta(L,\lambda,z)}$	$\min \frac{\eta(Q,\lambda,x)}{\eta(L,\lambda,z)}$
Algorithm 7	$1.4 \times 10^0$	$1.1 \times 10^0$
Algorithm 8	$1.5 \times 10^0$	$1.1 \times 10^0$
Algorithm 9	$3.1 \times 10^6$	$3.5 \times 10^5$
Algorithm 10	$3.4 \times 10^0$	$7.6 \times 10^{-1}$

Figure 5.1: Reduction of backward error of smallest eigenpairs of `mod_wiresaw2` using balancing.

### 5.4.2 `mod_hospital` problem

The `mod_hospital` problem is from `hospital` problem in the NLEVP collection. We set  $A_1 = 10^6 * A_1$  and  $\|A_1\|_2 \gg \sqrt{\|A_0\|_2 \|A_2\|_2}$  with  $A_2, A_1, A_0 \in \mathbb{C}^{200 \times 200}$ . In Algorithms 9 and 10, we set  $w$  close to absolute value of target eigenvalues where  $w = 10^6$  and  $w = 10^{-4}$ . As shown in Table 5.3, when  $|\lambda|$  close to  $10^{-4}$ , the backward error ratio of Algorithm 10 is close to 1. Backward error ratios of other algorithms are far from 1. The figure 5.3 show that the backward error of Algorithm 10 is smaller than other methods.

As shown in Table 5.4, when  $|\lambda|$  close to  $10^6$ , Algorithms 7, 8 and 10 have smaller backward ratio

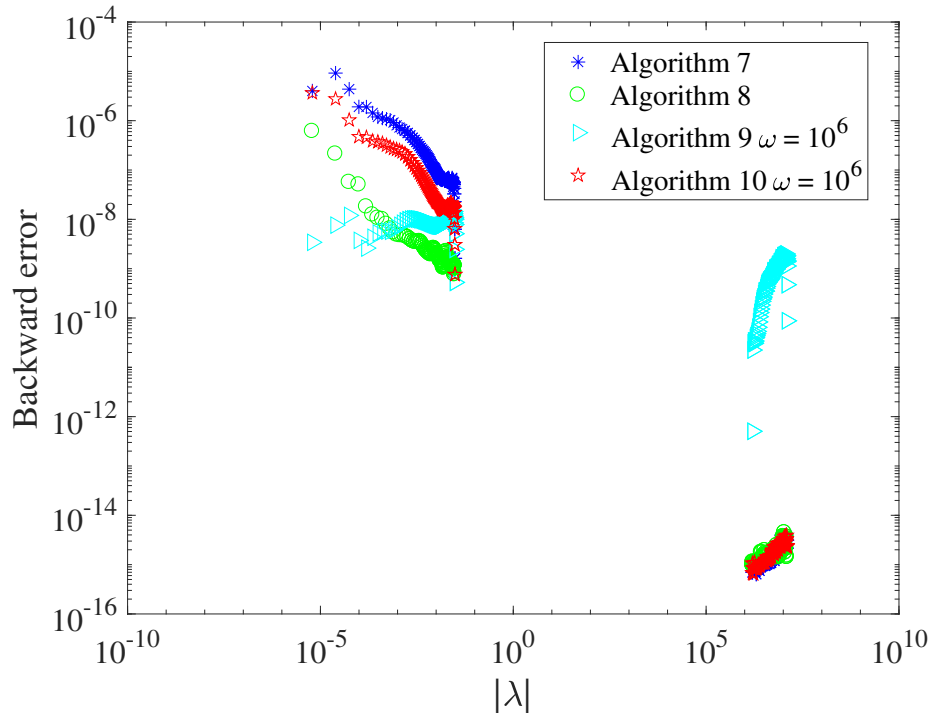


Figure 5.2: Reduction of backward error of largest eigenpairs of `mod_wiresaw2` using balancing.

Table 5.3: Backward error ratio when  $|\lambda|$  close to  $10^{-4}$ .

Algorithms	$\max \frac{\eta(Q,\lambda,x)}{\eta(L,\lambda,z)}$	$\min \frac{\eta(Q,\lambda,x)}{\eta(L,\lambda,z)}$
Algorithm 7	$4.4 \times 10^9$	$1.3 \times 10^8$
Algorithm 8	$5.7 \times 10^7$	$2.4 \times 10^6$
Algorithm 9	$1.0 \times 10^6$	$3.3 \times 10^4$
Algorithm 10	$5.2 \times 10^0$	$8.4 \times 10^{-1}$

which close to 1. However, the backward error ratio of Algorithm 9 is far from 1. The result suggests that Algorithm 10 reduce the backward error of computing eigenpairs and Algorithms 7 and 8 also improve the backward error of heavily damped QEP. Figure 5.4 satisfy this result.

## 5.5 Conclusion

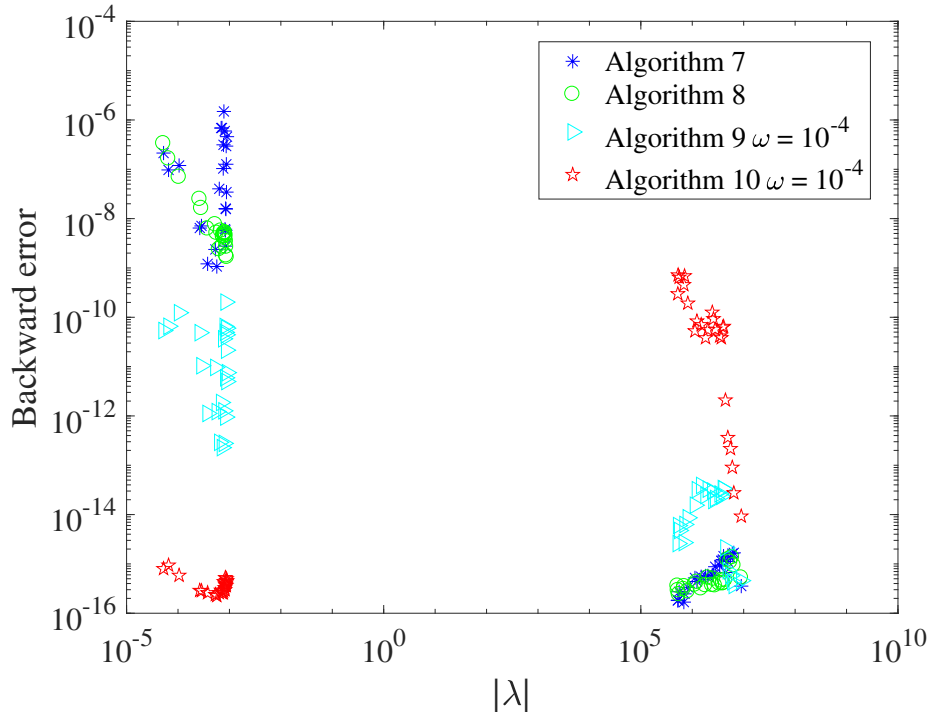
We have some conclusions as following

- The proposed method improve the backward error of target eigenpairs in heavily damped



Table 5.4: Backward error ratio when  $|\lambda|$  close to  $10^6$ .

Algorithms	$\max \frac{\eta(Q,\lambda,x)}{\eta(L,\lambda,z)}$	$\min \frac{\eta(Q,\lambda,x)}{\eta(L,\lambda,z)}$
Algorithm 7	$1.1 \times 10^0$	$9.5 \times 10^{-1}$
Algorithm 8	$1.0 \times 10^0$	$9.4 \times 10^{-1}$
Algorithm 9	$1.4 \times 10^6$	$8.2 \times 10^4$
Algorithm 10	$2.5 \times 10^0$	$1.1 \times 10^0$

Figure 5.3: Reduction of backward error of smallest eigenpairs of `mod_hospital` using balancing.

QEP.

- Only use linearization form and linearization form with Parlett's balancing technique improve backward error of some eigenpairs. However, these methods can not improve backward error of all eigenpairs in heavily damped QEP.
- In the future, we will analyze the relation of backward error of  $P(\lambda)$  relative to that of  $L(\lambda)$  and explain the reason that the proposed method can improve the backward error of heavily damped eigenvalue problems.

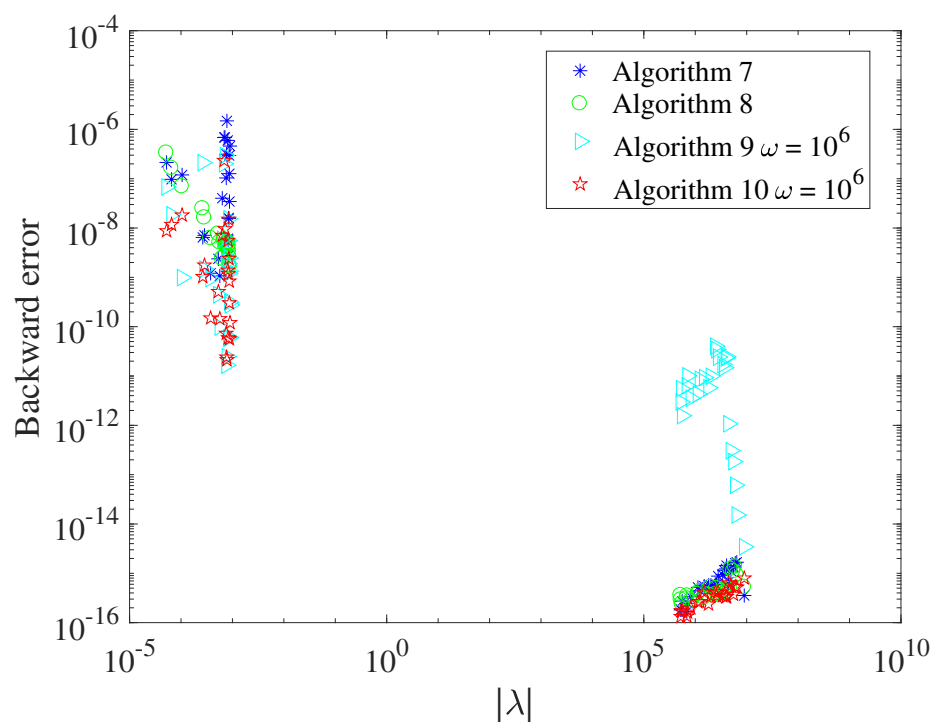


Figure 5.4: Reduction of backward error of largest eigenpairs of `mod_hospital` using balancing.

# Chapter 6

## Conclusion and future work

In this thesis, we improve the backward error of computing eigenpairs in polynomial eigenvalue problems. We give a overview of our contributions.

### 6.1 Conclusion

- In quadratic eigenvalue problem (QEP), we use scaling technique in SS-RR method for improving backward error of partial eignepairs in QEP. We explain the reason that the projected matrix polynomial with scaling technique can improve the backward error of the SS-RR method.
- In polynomial eigenvalue problem (PEP), we combine the SS-RR method with balancing technique. This method not only improve the backward error of eigenpairs in QEP, but also can improve the backward error in high-order degree PEP. We find some relation between backward error of  $P(\lambda)$  with backward error of SEP. Using these relation, we explain the reason that the backward error of the SS-RR method can be reduced by balancing technique.
- For heavily damped quadratic eigenvalue problem, we introduce some balancing techniques for GEP and PEP. Then we compare some methods with balancing techniques

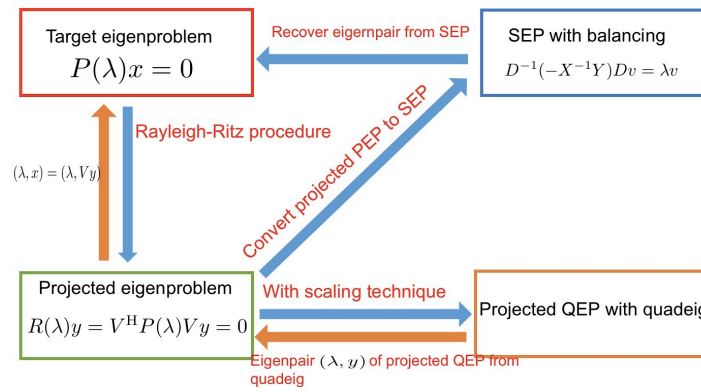


Figure 6.1: A map of research in this thesis.

for solving heavily damped QEP. In numerical experiments, we find that the proposed method can improve the backward error of heavily damped QEP.

A map of this research is shown in figure 6.1

## 6.2 Future work

- We will discuss other type of SS method with balancing techniques for solving PEP and QEP.
- We will analyze backward error of the proposed method for solving heavily damped QEP.
- For solving nonlinear eigenvalue problem, we will discuss nonlinear eigenvalue problem expressed non-monomial basis and analyze the backward error of the nonlinear eigenvalue problems.

# Bibliography

- [1] T. Betcke, N. J. Higham, V. Mehrmann, C. Schröder, and F. Tisseur. NLEVP: A collection of nonlinear eigenvalue problems. *ACM Trans. Math. Software*, 39, 7:1–7:28, 2013.
- [2] F. Tisseur and K. Meerbergen. The quadratic eigenvalue problem. *SIAM Rev.* 43, 235–286., 2001.
- [3] J. G. F. Francis. The qr transformation a unitary analogue to the lr transformation part 1. *Comput. J.*, 4, 265–271., 1961.
- [4] J. G. F. Francis. The qr transformation part 2. *Comput. J.*, 4, 332–345., 1962.
- [5] Z. Bai, J. W. Demmel, J. J. Dongarra, A. Ruhe, and H. A. van der Vorst. *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, Society for Industrial and Applied Mathematics. Philadelphia, PA, USA, 2000.
- [6] C. B. Moler and G. W. Stewart. An algorithm for generalized eigenvalue problems. *SIAM J. Numer. Anal.*, 10:241-256, 1973.
- [7] R. C. Ward. The combination shift qz algorithm. *SIAM J. Numer. Anal.*, 12(6):835-853, 1975.
- [8] R. C. Ward. Balancing the generalized eigenvalue problem. *SIAM J. Sci. Statist. Comput.*, 2(2):141-152, 1981.
- [9] L. Kaufman. Some thoughts on the qz algorithm for solving the generalized eigenvalue problem. *ACM Trans. Math. Software*, 3(1):65-75, 1977.

- [10] K. Dackland and B. Kagström. Blocked algorithms and software for reduction of a regular matrix pair to generalized schur form. *ACM Trans. Math. Software*, 25(4):425-454, 1999.
- [11] G. Golub and C. V. Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, 3rd edition,, 1996.
- [12] L. Taslamani. Strongly damped quadratic matrix polynomials. *SIAM J. MATRIX ANAL. APPL.*, Vol. 36, No. 2, 461-475, 2015.
- [13] F. Chaitin-Chatelin and M. B. Van Gijzen. Analysis of parameterized quadratic eigenvalue problems in computational acoustics with homotopic deviation theory. *Numer. Linear Algebra Appl.* 13, 487-512., 2006.
- [14] F. Tisseur. Backward error and condition of polynomial eigenvalue problems. *Linear Algebra Appl.* 309, 339-361., 2000.
- [15] S. D. Garvey, M. I. Friswell, and U. Prells. Co-ordinate transformations for second order systems. part i: General transformations. 258, 885-909., 2002.
- [16] P. Lancaster. *Lambda-Matrices and Vibrating Systems*. Pergamon Press, Oxford, 1966. Reprinted by Dover, New York, 2002.
- [17] D. S. Mackey, N. Mackey, C. Mehl, and V. Mehrmann. Vector spaces of linearizations for matrix polynomials. *SIAM J. Matrix Anal. Appl.*, 28 , 971-1004., 2006.
- [18] T. Sakurai and H. Sugiura. A projection method for generalized eigenvalue problems using numerical integration. *J. Comput. Appl. Math.* 159, 119-128, 2003.
- [19] T. Ikegami, T. Sakurai, and U. Nagashima. A filter diagonalization for generalized eigenvalue problems based on the sakurai-sugiura projection method. *J. Comput. Appl. Math.* 233, 1927-1936, 2010.
- [20] T. Ikegami and T. Sakurai. Contour integral eigensolver for non-hermitian systems: a rayleigh-ritz-type approach. *Taiwan. J. Math.* 14, 825-837., 2010.

- [21] S. Yokota and T. Sakurai. A projection method for nonlinear eigenvalue problems using contour integrals. *JSIAM Letters*. 5, 41–44, 2013.
- [22] J. Asakura, T. Sakurai, H. Tadano, T. Ikegami, and K. Kimura. A numerical method for nonlinear eigenvalue problems using contour integrals. *JSIAM Letters*. 1, 52–55, 2009.
- [23] J. Asakura, T. Sakurai, H. Tadano, T. Ikegami, and K. Kimura. A numerical method for polynomial eigenvalue problems using contour integral. *Japan J. Indust. Appl. Math.* 27, 73–90, 2010.
- [24] H. Chen, Y. Maeda, A. Imakura, T. Sakurai, and F. Tisseur. Improving the numerical stability of the sakurai-sugiura method for quadratic eigenvalue problems. *JSIAM Letters*. 9, 17–20, 2017.
- [25] E. E. Osborne. On preconditioning of matrices. *Journal of the ACM*. 7, 338–345, 1960.
- [26] B. Parlett and C. Reinsch. Balancing a matrix for calculation of eigenvalues and eigenvectors. *Numer. Math.* 13, 293–304, 1969.
- [27] H. Chen, A. Imakura, and T. Sakurai. Improving backward stability of sakurai-sugiura method with balancing technique in polynomial eigenvalue problem. *Applications of Mathematics*. 62, 357–375, 2017.
- [28] N. J. Higham, D. S. Mackey, F. Tisseur, and S. D. Garvey. Scaling, sensitivity and stability in the numerical solution of quadratic eigenvalue problems. *Internat. J. Numer. Methods Eng.* 73 (2008), 344–360, 2008.
- [29] N. J. Higham, R.-C. Li, and F. Tisseur. Backward error of polynomial eigenproblems solved by linearization. *SIAM J. Matrix Anal. Appl.*, 29, 1218–1241, 2007.
- [30] P. Lancaster and M. Tismenetsky. *The Theory of Matrices*. 2nd ed., Academic Press, London, 1985.
- [31] S. Gaubert and M. Sharify. Tropical scaling of polynomial matrices. *Lecture Notes in Control and Information Sciences*, 389, 291–303, 2009.

- 
- [32] H. Y. Fan, W. W. Lin, and P. Van Dooren. Normwise scaling of second order polynomial matrices. *SIAM J. Matrix Anal. Appl.*, 26, 252–256, 2004.
- [33] S. Hammarling, C. J. Munro, and F. Tisseur. An algorithm for the complete solution of quadratic eigenvalue problems. *ACM Trans. Math. Software*, 39, 18:1–18:9, 2013.
- [34] L. Z. Hui and Y. F. Su. A backward stable algorithm for quadratic eigenvalue problems. *SIAM. J. Matrix Anal. Appl.*, 35, 499–516, 2014.
- [35] Rodney James, Julien Langou, and Bradley R. Lowery. On matrix balancing and eigenvector computation. *arXiv:1401.5766, Numerical Analysis (math.NA)*, 2014.
- [36] T. Betcke. Balancing regular matrix pencils. *SIAM J. Matrix Anal. Appl.*, 28, 253–263, 2006.
- [37] T. Betcke. Optimal scaling of generalized and polynomial eigenvalue problems. *SIAM J. Matrix Anal. Appl.*, 30, 1320–1338, 2008.



# Publications

## Journals:

- H. Chen, A. Imakura, T. Sakurai, Improving backward stability of Sakurai-Sugiura method with balancing technique in polynomial eigenvalue problem, *Applications of Mathematics*, 62 (4), 357–375 (2017).
- H. Chen, Y. Maeda, A. Imakura, T. Sakurai, F. Tisseur, Improving the numerical stability of the Sakurai-Sugiura method for quadratic eigenvalue problems, *JSIAM Letters*, Vol. 9, pp. 17– 20 (2017).

## International conferences:

- H. Chen, T. Sakurai, Backward error of nonlinear eigenvalue problem expressed in non-monomial basis, *The 27th Biennial Numerical Analysis Conference*, Glasgow, United Kingdom, June 27 -30, 2017.
- H. Chen, A. Imakura, T. Sakurai, Balancing technique for Rayleigh-Ritz type of Sakurai-Sugiura method in quadratic eigenvalue problems, *5th IMA Conference on Numerical Linear Algebra and Optimization*, University of Birmingham, United Kingdom, September 7-9, 2016.

## Japanese conferences:

- H. Chen, A. Imakura, T. Sakurai, A balancing technique for improving backward error of heavily damped quadratic eigenvalue problems. the 2017 conference of the Japan Society

for Industrial and Applied Mathematics, Musashino University, Japan. September 6-8, 2017.

- H. Chen, Y. Maeda, T. Sakurai, Scaling technique for Sakurai-Sugiura method in quadratic eigenvalue problems, the 2015 conference of the Japan Society for Industrial and Applied Mathematics, Kanazawa University, Japan. September 9-11, 2015.