Solution of Nonlinear Eigenvalue Problems Arising from Constrained Rayleigh Quotient Optimization and Resonant Modes Computation of Accelerator Cavity

By

YUNSHEN ZHOU

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Applied Mathematics

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Zhaojun Bai, Chair

Robert Guy

Naoki Saito

Committee in charge

2020

Contents

List of Figures

List of Tables

Solution of Nonlinear Eigenvalue Problems Arising from Constrained Rayleigh Quotient Optimization and Resonant Modes Computation of Accelerator Cavity

Abstract

This dissertation consists of two parts: constrained Rayleigh quotient optimization problems and its application in image segmentation, and Padé approximation linearization algorithm to solve nonlinear eigenvalue problems arising from resonant modes computation of accelerator cavity.

In the first part, we consider the following constrained Rayleigh quotient optimization problem (CRQopt)

$$\min_{x\in\mathbb{R}^n} x^{\mathrm{T}}Ax \text{ subject to } x^{\mathrm{T}}x = 1 \text{ and } C^{\mathrm{T}}x = b,$$

where A is an $n \times n$ real symmetric matrix and C is an $n \times m$ real matrix. Usually, $m \ll n$. The problem is also known as the constrained eigenvalue problem in the literature because it becomes an eigenvalue problem if the linear constraint $C^{\mathrm{T}}x = b$ is removed. We start by equivalently transforming CRQopt into an optimization problem, called LGopt, of minimizing the Lagrangian multiplier of CRQopt, and then an eigenvalue problem, called QEPmin, of finding the smallest eigenvalue of a quadratic eigenvalue problem. Although such equivalences has been discussed in the literature, it appears to be the first time that these equivalences are rigorously justified. Then we propose to numerically solve LGopt and QEPmin by the Krylov subspace projection method via the Lanczos process. The basic idea, as the Lanczos method for the symmetric eigenvalue problem, is to first reduce LGopt and QEPmin by projecting them onto Krylov subspaces to yield problems of the same types but of much smaller sizes, and then solve the reduced problems by some direct methods, which is either a secular equation solver (in the case of LGopt) or an eigensolver (in the case of QEPmin). The resulting algorithm is called the Lanczos algorithm. We perform convergence analysis for the proposed method and obtain error bounds. The sharpness of the error bound is demonstrated by artificial examples, although in applications the method often converges much faster than the bounds suggest. Finally, we apply the Lanczos algorithm to semi-supervised learning in the context of constrained clustering.

In the second part, we propose a method to solve nonlinear eigenvalue problems with

low-rank nonlinear terms. We first apply rational approximants to the nonlinear functions and transform the nonlinear eigenvalue problem to a rational eigenvalue problem. Then we transform the rational eigenvalue problem to a linear eigenvalue problem by trimmed linearization. For solving the linear eigenvalue problem, we provide a method to compute the matrix-vector multiplication for the shift-invert Arnoldi method, which is suitable for any shift. Moreover, the method of matrix-vector multiplication is designed to make the arithmetic as real as possible. We show the effectiveness of our Padé approximate linearization (PAL) method by comparing it with the fully rational Krylov method for nonlinear eigenvalue problems (NLEIGS) and compact rational Krylov (CORK) methods to solve nonlinear eigenvalue problems arising from finite element electromagnetic simulations in accelerator modeling. For these problems, we provide a method to choose the expansion point of rational approximants to reduce the residual for the eigenpairs by up to three digits. Numerical examples show that our PAL algorithm runs 48% to 87% faster than NLEIGS and CORK algorithm with comparable results.

Acknowledgments

My advisor, Professor Zhaojun Bai, has mentored me for more than three years during my graduate studies. His knowledge, intuition, and patience are the biggest supporters of my Ph.D. degree. I would like to thank Professor Naoki Saito and Professor Robert Guy for being my dissertation committee, Professor Roland Freund and Professor Javier Arsuaga for attending my qualifying exam.

In constrained Rayleigh quotient optimization problem, I really appreciate Professor Ren-Cang Li for his suggestion about the theory and algorithm, Ning Wan for his early work, Yanwen Luo about the idea of the proof for Lemma 3.2.8, Chengming Jiang for part of implementation for constrained image segmentation problems, Chao-Ping Lin for the discussion about the implementation of the software CRQPACK, and Michael Ragone for suggestions of writing.

In nonlinear eigenvalue problems arising from resonant modes computation of accelerator cavity, Jacob Johnson provided me with his early work, Ding Lu discussed with me about the ideas behind Example 2.3.3, Roel Van Beeumen provided me with some examples about his CORK software, Osni Marques provided me with implementation of the PAL algorithm in C, and Colin Hagemeyer revised my writing. I appreciate their help.

Thank Bohan Zhou, Ji Chen, and Yiqun Shao for being my officemate for three years. They gave me many career suggestions. Finally, thank my father Jian Zhou and my mother Xiaobang Chen for supporting my decisions during my graduate studies.

The work in this dissertation was supported in part by NSF grants DMS-1522697 and DMS-1913364.

Chapter 1

Introduction

1.1 Motivations and related work

Nonlinear eigenvalue problems are widely used in many applications. Today there are many existing methods to solve different kinds of nonlinear eigenvalue problems. However, methods to solve some nonlinear eigenvalue problems with special structures are still under development. In this dissertation, we discuss two problems related to nonlinear eigenvalue problems. The first problem is the constrained Rayleigh quotient optimization problem, which can be transformed into a structured quadratic eigenvalue problem. The second problem is the nonlinear eigenvalue problem with low-rank nonlinear terms, which arises from resonant modes computation of accelerator cavity.

The first part of the dissertation studies constrained Rayleigh quotient optimization problem (CRQopt). It is a constrained optimization problem where the objective is to minimize a quadratic function and there is one norm constraint and one linear constraint. The optimization problem is also known as *the constrained eigenvalue problem*, a term coined in [17] in 1989. However, it had appeared in the literature much earlier than that [22]. In that sense, it is a classical problem. However, past studies are fragmented with some claims, although often true, not rigorously justified or that needed conditions to hold. In this dissertation, our goal is to provide a thorough investigation into this classical problem, including rigorous justifications of statements previously taken for granted in the literature and addressing the theoretical subtleties that were not paid attention to. We also present a quantitative convergence analysis for the Krylov type subspace projection method, which we will also call the Lanczos algorithm, for solving large scale

optimization problems. The optimization problem has found a wide range of applications, such as ridge regression [11, 19], trust-region subproblem [45, 54], constrained least square problem [16], spectral image segmentation [14, 58], transductive learning [34], and community detection [46].

The second part of the dissertation is about nonlinear eigenvalue problems with low-rank nonlinear terms. There are many exising methods to solve general nonlinear eigenvalue problems [26]. In this disseration, our goal is to develop an efficient algorithm for this specific eigenvalue problem. The problem has applications in particle in a canyon [27], delay problems [32] with low rank and cavity design of a linear accelerator [63].

1.1.1 Early work of CRQopt

The first systematic study of CRQopt perhaps belongs to Gander, Golub and von Matt [17]. Using the full QR and eigen-decompositions, they first reformulated CRQopt as an optimization problem of finding the minimal Lagrangian multiplier via solving a secular equation (in a way that is different from our secular equation solver in Section 2.2.2). Alternatively, they also turned CRQopt into an optimization problem of finding the smallest real eigenvalue of a quadratic eigenvalue problem (QEP). However, the equivalence between the QEP optimization and the Lagrangian multiplier problem was not rigorously justified there.

Numerical algorithms proposed in [17] are not suitable for large scale CRQopt because they requires a full eigen-decomposition as a dense matrix. Later in [21], Golub, Zhang and Zha considered large and sparse CRQopt but only with the homogeneous constraint. In this special case, CRQopt is equivalent to computing the smallest eigenvalue of the matrix in the objective function restricted to the null space of the matrix in the linear constraint. An inner-outer iterative Lanczos method was proposed to solve the homogeneous CRQopt. In [68], Xu, Li and Schuurmans proposed a projected power method for solving CRQopt. The projected power method is an iterative method only involving matrix-vector products, and thus it is suitable for large and sparse CRQopt. However, its convergence is linear at best and often too slow. In [14], Eriksson, Olsson and Kahl reformulated CRQopt into an eigenvalue optimization problem (see Appendix B for details). An algorithm based on the line search was used to find the optimal solution. This algorithm is suitable for CRQopt with a large and sparse matrix A, but it is too costly because the smallest eigenvalue has to be computed multiple times during each line search action.

1.1.2 Brief review of existing nonlinear eigensolvers

Article [26] summarized existing algorithms for solving NEPs. Roughly speaking, existing algorithms for solving NEPs can be characterized into three main classes: Newton-based techniques, contour integration method, and methods based on approximation of matrix-valued functions.

In this dissertation, we are particularly interested in methods based on approximation. For methods based on approximation, matrix-valued functions $\mathcal{T}(\lambda)$ are approximated by polynomials or rational functions. Then the polynomial or rational eigenvalue problems can be solved by standard methods (the rational linearization method [61]).

Examples of methods based on approximation include

1. Chebyshev interpolation method [12], In the Chebyshev interpolation method, $\mathcal{T}(\lambda)$ is first approximated by an interpolating matrix polynomial, and the resulting polynomial eigenvalue problem is solved via linearization.

2. the Taylor-Arnoldi method [32], In the Taylor-Arnoldi method, the degree of the polynomial approximation is increased in every iteration and yielding a better approximation of $\mathcal{T}(\lambda)$ near a given shift.

3. the infinite Arnoldi method [62].

4. the Newton rational Krylov method [64].

5. the fully rational Krylov method [27], The fully rational Krylov method for nonlinear eigenvalue problems (NLEIGS) utilizes a dynamically constructed rational interpolant of the nonlinear functions and a new companion-type linearization for obtaining a generalized eigenvalue problem with particular structures.

6. The generic class of CORK methods [65]. The compact rational Krylov (CORK) methods exploit the structure of the linearization pencils by using a generalization of the compact Arnoldi decomposition to save the memory and the orthogonalization cost.

However, the reliability of these methods heavily depends on the quality of the approximation of the nonlinear functions. The structure exploiting implementation in the CORK framework is generally more efficient than methods based on contour integration. However, the computation

for CORK requires multiple LU factorizations and high degree rational approximations in some cases.

1.2 Contributions

In the first part of the dissertation, our study begins with the standard approach of Lagrangian multipliers, as was taken in [17], which leads to an optimization problem of minimizing the Lagrangian multiplier of CRQopt, called LGopt (Section 3.2.2), and then an optimization problem of finding the smallest real eigenvalue of a quadratic eigenvalue problem (QEP), called QEPmin (Section 3.2.3). We summarize our major contributions as follows:

1. Although transforming CRQopt into LGopt and QEPmin is not really new, our formulations of LGopt and QEPmin set them up onto a natural path for use in Krylov subspace type projection methods that only requires matrix-vector products. Therefore, the formulations are suitable for large scale CRQopt. We rigorously proved the equivalences among the three problems while they were only loosely argued previously as, e.g., in [17]. As far as subtle technicalities are concerned, we prove that the leftmost eigenvalue in the complex plane is real, which has a significant implication when it comes to numerical computations.

2. We devise a Lanczos algorithm to solve the induced optimization problems: LGopt and QEPmin. This algorithm is made possible, as we argued moments ago, by our different formulations from what is in the literature. Along the way, we also propose an efficient numerical algorithm for the type of secular equations arising from solving each projected LGopt.

3. We establish a quantitative convergence analysis for the Lanczos algorithm and obtain error bounds on approximations generated by the algorithm. These error bounds are in general sharp in the worst case as demonstrated by artificially designed numerical examples.

4. We apply our algorithm to the large scale CRQopt from the constrained clustering that arises from the standard spectral algorithm with linear constraints to encode prior knowledge labels. During our tests, we observed that our algorithm was 2 to 23 times faster than FAST-GE-2.0 [33] for constrained image segmentation, depending on given image data.

In the first part of the dissertation, our study begins with introducing the Padé approximation linearization (PAL) algorithm (Chapter 5) and then applying our PAL algorithm to resonant modes computation of accelerator cavity (Chapter 6). We summarize our major contributions as follows:

1. We proposed a PAL algorithm to nonlinear eigenvalue problems with low-rank nonlinear terms. For computing resonant modes of accelerator cavities, our PAL algorithm is faster than existing NLEIGS and CORK algorithms.

2. We designed an efficient matrix-vector multiplication method to avoid forming the matrix explicitly. The multiplication method can be used to compute the eigenvalue of the linear eigenvalue problem near an arbitrary shift.

3. For computing resonant modes of accelerator cavities, we discussed a method to choose the expansion point of rational approximants, which reduces the residual of the computed eigenpairs by two to three digits.

1.3 Organization and notations

The dissertation is organized as follows. In Chapter 2, we review Lanczos method, trust-region subproblems, Padé approximation and realization of rational functions. In Chapter 3, we discuss the theory, algorithm and convergence analysis of constrained Rayleigh quotient optimization problems. In Chapter 4, we discuss the applications of our algorithm for the optimization problem in image segmentation problems. In Chapter 5, we show our PAL algorithm to nonlinear eigenvalue problems and the application in accelerator cavity design is discussed in Chapter 6.

Throughout the dissertation, $\mathbb{R}$, $\mathbb{R}^n$ and $\mathbb{R}^{m\times n}$ are the set of real numbers, columns vectors of dimension n, and $m\times n$ matrices, respectively. $\mathbb{C}$, $\mathbb{C}^n$ and $\mathbb{C}^{m\times n}$ are the set of complex numbers, columns vectors of dimension n, and $m\times n$ matrices, respectively. We use MATLAB-like notation $X_{(i:j,k:l)}$ to denote the submatrix of X, consisting of the intersections of rows i to j and columns k to l, and when $i:j$ is replaced by :, it means all rows, similarly for columns. For a vector $v\in\mathbb{C}$, $v_{(k)}$ refers the kth entry of v and $v_{(i:j)}$ is the subvector of v consisting of the ith to jth entries inclusive. The $n\times n$ identity matrix is I_n or simply I if its size is clear from the context, and e_j is

the jth column of an identity matrix whose size is determined by the context. $\mathrm{diag}(c_1, c_2, \ldots, c_n)$ is an $n \times n$ diagonal matrix with diagonal elements $c_1, c_2, \ldots, c_n$. The imaginary unit is $\mathtt{i} = \sqrt{-1}$.

For $X \in \mathbb{C}^{m\times n}$, X^{T}, $\mathcal{R}(X)$, $\mathcal{N}(X)$ denote its transpose (without conjugate), range and null space, respectively. For a real symmetric matrix H, $\mathrm{eig}(H)$ stands for the set of all eigenvalues of H, and $\lambda_{\min}(H)$ and $\lambda_{\max}(H)$ denote the smallest and largest eigenvalue of H, respectively. $\|\cdot\|_p$ $(1 \le p \le \infty)$ is the ℓ_p-vector or ℓ_p-operator norm, respectively, depending on the argument. As a special case, $\|\cdot\|_2$ or $\|\cdot\|$ is either the Euclidean norm of vector or the spectral norm of a matrix.

Chapter 2

Preliminaries

We review Lanczos method in Section 2.1, trust-region subproblems in Section 2.2, Padé approximation and expression of rational functions in realization form in Section 2.3.

2.1 Lanczos process

We review the standard symmetric Lanczos process [10, 20, 51, 55]. Given a real symmetric matrix $M \in \mathbb{R}^{n\times n}$ and a starting vector $r_0 \in \mathbb{R}^n$, the Lanczos process partially computes the decomposition $MQ = QT$, where $T \in \mathbb{R}^{n\times n}$ is symmetric and tridiagonal, $Q \in \mathbb{R}^{n\times n}$ is orthogonal and the first column of Q is parallel to r_0.

Specifically, let $Q = [q_1, q_2, \ldots, q_n]$ and denote by α_i for $1 \le i \le n$ the diagonal entries of T, and by β_i for $2 \le i \le n$ the sub-diagonal and super-diagonal entries of T. The Lanczos process goes as follows: set $q_1 = r_0/\|r_0\|$, and equate the first column of both sides of the equation $MQ = QT$ to get

$$Mq_1 = q_1\alpha_1 + q_2\beta_2. \tag{2.1}$$

Pre-multiply both sides of the equation (2.1) by q_1^{T} to get $\alpha_1 = q_1^{\mathrm{T}} M q_1$, and then let

$$\widehat{q}_2 = Mq_1 - q_1\alpha_1,\ \ \beta_2 = \|\widehat{q}_2\|.$$

Now if $\beta_2 > 0$, we set $q_2 = \widehat{q}_2/\beta_2$; otherwise the process breaks down. In general for $j \ge 2$, equating the jth column of both sides of the equation $MQ = QT$ leads to

$$Mq_j = q_{j-1}\beta_j + q_j\alpha_j + q_{j+1}\beta_{j+1}. \tag{2.2}$$

Up to this point, q_i for $1 \le i \le j$, α_i for $1 \le i \le j-1$, and β_i for $2 \le i \le j$ have already been determined. Pre-multiply both sides of the equation (2.2) by q_j^{T} to get $\alpha_j = q_j^{\mathrm{T}} M q_j$, and then let

$$\widehat{q}_{j+1} = M q_j - q_{j-1}\beta_j - q_j \alpha_j, \ \beta_{j+1} = \|\widehat{q}_{j+1}\|.$$

Now if $\beta_{j+1} > 0$, we set $q_{j+1} = \widehat{q}_{j+1}/\beta_{j+1}$; otherwise the process breaks down. The process can be compactly expressed by[1]

$$M Q_k = Q_k T_k + \beta_{k+1} q_{k+1} e_k^{\mathrm{T}} \tag{2.3}$$

assuming the process encounters no breakdown for the first k steps, i.e., no $\beta_i = 0$ for $2 \le i \le k$, where

$$Q_k = [q_1, q_2, \ldots, q_k], \ T_k = Q_k^{\mathrm{T}} M Q_k = \begin{bmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{k-1} & \alpha_{k-1} & \beta_k \\ & & & \beta_k & \alpha_k \end{bmatrix}.$$

Furthermore, the column space $\mathcal{R}(Q_k)$ is the same as the kth Krylov subspace

$$\mathcal{K}_k(M, r_0) := \mathrm{span}(r_0, M r_0, \cdots, M^{k-1} r_0).$$

In the case of a breakdown with $\beta_{k+1} = 0$, $M Q_k = Q_k T_k$ and $\mathcal{R}(Q_k)$ is an invariant subspace of M.

2.2 Trust-region subproblems

2.2.1 Problem statement

In this section we discuss the following trust region subproblems [45, 47]:

$$\min_{\|y\| \le \gamma} y^T H y + 2 g_0^T y, \tag{2.4}$$

where $H \in \mathbb{R}^{n \times n}$, $H = H^T$, $g_0 \in \mathbb{R}^n$ and $\gamma > 0$ is the trust region radius. Note that if there exists a solution y of (2.4) such that $\|y\| < \gamma$, which means y lies in the interior of the trust region, then H is positive semidefinite and y is the global minimizer of the function $y^T H y + 2 g_0^T y$. Therefore,

[1] We sacrifice slightly mathematical rigor in writing (2.3) in exchange for simplicity and convenience, since q_{k+1} cannot be determined unless also $\beta_{k+1} > 0$.

it can be solved as an unconstrained optimization problem. In the rest of our dissertation we only consider the optimization problem with equality constraint

$$\min_{\|y\|=\gamma} y^T H y + 2g_0^T y. \tag{2.5}$$

The solution of (2.5) can be characterized by the following conditions [28, Lemma 2.1]:

Lemma 2.2.1. *y is a solution of (2.5) if and only if there exists λ_* such that $(H - \lambda_* I)y = -g_0$, $\|y\| = \gamma$ and $H - \lambda_* I$ is positive semidefinite.*

Let $\theta_1 = \theta_2 = \cdots = \theta_d < \theta_{d+1} \leq \cdots \leq \theta_n$ be the eigenvalues of H, $y_1, y_2, \cdots, y_n$ be the corresponding eigenvectors, $\xi_i = g_0^T y_i$ for $i = 1, 2, \cdots, n$ and $\mathcal{E}_1$ be the linear space spanned by $y_1, y_2, \cdots, y_d$, then the solution of (2.5) can be characterized as:

1. Hard case [47, Sec.4.3] (or degenerate case [28, Lemma 2.2]): $g_0 \perp \mathcal{E}_1$ and $\|(H - \theta_1 I)^\dagger g_0\| \leq \gamma$, then $\lambda_* = \theta_1$ and the general solution of (2.5) is

   $$y = x + \sqrt{\gamma^2 - \|x\|^2} z$$

   where $x = -(H - \theta_1 I)^\dagger g_0$ and z is any unit vector in $\mathcal{E}_1$.

2. Easy case [47, Sec.4.3] (or nondegenerate case [28, Lemma 2.2]): if the hard case (or degenerate case) does not hold, then $\lambda_* < \theta_1$ and the unique solution of (2.5) is

   $$y = -(H - \lambda_* I)^{-1} g_0.$$

2.2.2 Solving the secular equation

For the easy case, λ_* can be obtained by solving the smallest zero of a secular function. We are interested in computing the smallest zero λ_* of the secular function

$$\chi(\lambda) := \sum_{i=1}^{n} \frac{\xi_i^2}{(\lambda - \theta_i)^2} - \gamma^2, \tag{2.6}$$

where it is assumed

$$\gamma > 0,\ \theta_1 \leq \theta_2 \leq \cdots \leq \theta_n,\ \text{and,}$$

$$\text{either } \xi_1 \neq 0, \text{ or } \xi_1 = 0 \text{ but } \lim_{\lambda \to \theta_1^-} \chi(\lambda) > 0.$$

Those assumptions guarantee that $\chi(\lambda)$ has a unique zero λ_* in $(-\infty, \theta_1)$. This is because

$$\lim_{\lambda\to-\infty}\chi(\lambda) = -\gamma^2 < 0, \ \lim_{\lambda\to\theta_1^-}\chi(\lambda) > 0, \text{ and } \chi'(\lambda) = -2\sum_{i=1}^{n}\frac{\xi_i^2}{(\lambda-\theta_i)^3} > 0 \text{ for } \lambda < \theta_1.$$

First, we find an initial lower bound $\alpha^{(0)}$ of λ_*, i.e., $\alpha^{(0)} < \theta_1$ such that $\chi(\alpha^{(0)}) < 0$. Note

$$\chi(\lambda) \le \sum_{i=1}^{n}\frac{\xi_i^2}{(\lambda-\theta_1)^2} - \gamma^2 \quad \text{for } \lambda < \theta_1.$$

One such $\alpha^{(0)}$ can be found by solving

$$\sum_{i=1}^{n}\frac{\xi_i^2}{(\alpha^{(0)}-\theta_1)^2} - \gamma^2 = 0 \quad \Rightarrow \quad \alpha^{(0)} = \theta_1 - \delta_0 \text{ with } \delta_0 = \frac{1}{\gamma}\sqrt{\sum_{i=1}^{n}\xi_i^2}\,.$$

We conclude that $\lambda_* \in [\alpha^{(0)}, \beta^{(0)}]$, where $\beta^{(0)} = \theta_1$. Quantities $\alpha^{(k)}$ and $\beta^{(k)}$ will be determined during our iterative process to be described such that $\lambda_* \in [\alpha^{(k)}, \beta^{(k)}]$.

Without loss of generality, we may assume that

$$\text{if } \theta_1 = \cdots = \theta_d < \theta_{d+1}, \text{ then } \xi_2 = \cdots = \xi_d = 0.$$

Let

$$j_0 = \min\{i \,:\, \xi_i \neq 0\}. \tag{2.7}$$

To find the initial guess of the root, we solve

$$\frac{\xi_{j_0}^2}{(\lambda-\theta_{j_0})^2} + \underbrace{\sum_{i=j_0+1}^{n}\frac{\xi_i^2}{([\theta_{j_0}-\delta_0]-\theta_i)^2} - \gamma^2}_{=:-\eta} = 0$$

for λ to get

$$\lambda^{(0)} = \begin{cases} \theta_{j_0} - |\xi_{j_0}|/\sqrt{\eta}, & \text{if } \eta > 0, \\ \theta_{j_0} - \delta_0/2, & \text{if } \eta \le 0, \end{cases}$$

where the second case is based on bisection.

For the iterative scheme, suppose we have an approximation $\lambda^{(k)} \approx \lambda_*$. First, the interval $(\alpha^{(k)}, \beta^{(k)})$ will be updated as

$$\alpha^{(k+1)} \leftarrow \lambda^{(k)} \text{ and } \beta^{(k+1)} \leftarrow \beta^{(k)} \text{ if } \chi(\lambda^{(k)}) < 0$$

$$\beta^{(k+1)} \leftarrow \lambda^{(k)} \text{ and } \alpha^{(k+1)} \leftarrow \alpha^{(k)} \text{ if } \chi(\lambda^{(k)}) > 0.$$

Then we find the next approximation $\lambda^{(k+1)}$. For that purpose, we seek to approximate χ, in the neighborhood of $\lambda^{(k)}$, by

$$g(\lambda) := -b + \frac{a}{(\lambda - \theta_{j_0})^2} \approx \chi(\lambda),$$

such that

$$\begin{aligned} g(\lambda^{(k)}) &\equiv -b + \frac{a}{(\lambda^{(k)} - \theta_{j_0})^2} = \chi(\lambda^{(k)}) = \sum_{i=1}^{n} \frac{\xi_i^2}{(\lambda^{(k)} - \theta_i)^2} - \gamma^2, \\ g'(\lambda^{(k)}) &\equiv -2\frac{a}{(\lambda^{(k)} - \theta_{j_0})^3} = \chi'(\lambda^{(k)}) = -2\sum_{i=1}^{n} \frac{\xi_i^2}{(\lambda^{(k)} - \theta_i)^3}, \end{aligned}$$

yielding

$$\begin{aligned} a &= -\frac{1}{2}(\lambda^{(k)} - \theta_{j_0})^3 \chi'(\lambda^{(k)}) = (\lambda^{(k)} - \theta_{j_0})^3 \sum_{i=1}^{n} \frac{\xi_i^2}{(\lambda^{(k)} - \theta_i)^3} > 0, \\ b &= \frac{a}{(\lambda^{(k)} - \theta_{j_0})^2} - \chi(\lambda^{(k)}) = (\lambda^{(k)} - \theta_{j_0}) \sum_{i=1}^{n} \frac{\xi_i^2}{(\lambda^{(k)} - \theta_i)^3} - \chi(\lambda^{(k)}). \end{aligned}$$

Ideally, $b > 0$ so that $g(\lambda) = 0$ has a solution in $(-\infty, \theta_{j_0})$. Assuming $b > 0$, we find the next approximation $\lambda^{(k+1)} \approx \lambda_*$ is given by

$$\lambda^{(k+1)} = \theta_1 - \sqrt{a/b}. \tag{2.8}$$

Now if $b \le 0$ (then $\lambda^{(k+1)}$ as in (2.8) is undefined) or if $\lambda^{(k+1)} \notin (\alpha, \beta)$, we let $\lambda^{(k+1)}$ be $(\alpha^{(k+1)} + \beta^{(k+1)})/2$ according to bisection method.

2.3 Padé approximation and realization of rational functions

One of the most well-known rational approximations to a complex-valued function is Padé approximation, an extension of Taylor polynomial approximation. According to Baker's definition [3, Sec.1.4], if polynomials $p_n(z)$ and $q_m(z)$ of degrees n and m respectively, the rational function

$$r_{[n,m]}(z) = \frac{p_n(z)}{q_m(z)} = f(z) + O(z^{m+n+1}) \quad \text{with} \quad q_m(0) = 1$$

is called an order-(n, m) Padé approximant of $f(z)$ near zero. For example,

$$r_{[2,2]}(z) = \frac{1 + \frac{1}{2}z + \frac{1}{12}z^2}{1 - \frac{1}{2}z + \frac{1}{12}z^2}$$

is an order-$(2,2)$ Padé approximant of $f(z) = e^z$ and

$$r_{[2,2]}(z) = \frac{1 + \frac{4}{5}z + \frac{5}{16}z^2}{1 + \frac{3}{4}z + \frac{1}{16}z^2}$$

is an order-$(2,2)$ Padé approximant of $f(z) = \sqrt{z+1}$. For the construction of Padé approximation, besides the classical book by Baker in 1996 [3, Sec.1.4], more recent study include [23, 5] and references therein.

Let $p_{m-1}(z)$ be a polynomial of degree no more than $m-1$, $q_m(z)$ is a polynomial of degree m with the coefficients of z^m be 1 and $r_{[m-1,m]}(z)$ be a rational function such that

$$r_{[m-1,m]}(z) = \frac{p_{m-1}(z)}{q_m(z)} = \frac{g_0 + g_1 z + \cdots + g_{m-1}z^{m-1}}{h_0 + h_1 z + \cdots + h_{m-1}z^{m-1} + z^m}.$$

Then a realization of rational function $r_{[m-1,m]}(z)$ is given by

$$r_{[m-1,m]}(z) = -a_m^{\mathrm{T}}(I_m z - D_m)^{-1} b_m \tag{2.9}$$

where

$$D_m = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -h_0 & -h_1 & -h_2 & \cdots & -h_{m-1} \end{bmatrix} \in \mathbb{C}^{m\times m}$$

and

$$a_m = -[g_0, g_1, \cdots, g_{m-1}]^{\mathrm{T}} \in \mathbb{C}^m, \quad b_m = [0, 0, \cdots, 0, 1]^{\mathrm{T}} \in \mathbb{R}^m.$$

Note that for the complex case, a_m^{T} is transpose without conjugate [1, Sec.4.4.2].

When a partial fraction form of a proper rational function $r_{[m-1,m]}(z)$ is available, namely,

$$r_{[m-1,m]}(z) = -\sum_{j=1}^{m} \frac{b_j}{c_j - d_j z}, \tag{2.10}$$

where, without loss of generality, we assume that $b_j \in \mathbb{R}$, $b_j \geq 0$ and $c_j, d_j \in \mathbb{C}$ for $j = 1, 2, \cdots m$. A realization of $r_{[m-1,m]}(z)$ is then given by

$$r_{[m-1,m]}(z) = -a_m^{\mathrm{T}}(C_m - D_m z)^{-1} a_m \tag{2.11}$$

where $a_m = [\sqrt{b_1}, \sqrt{b_2}, \cdots, \sqrt{b_m}]^{\mathrm{T}}$, $C_m = \mathrm{diag}(c_1, c_2, \ldots, c_m)$ and $D_m = \mathrm{diag}(d_1, d_2, \ldots, d_m)$. Note that a_m is a real vector and C_m, D_m are complex matrices.

In general, let $r_{[n,m]}(z) = \frac{p_n(z)}{q_m(z)}$ be a rational function where the degree of the polynomial on the numerator is no less than the degree of the polynomial on the denominator, i.e., $n \geq m$. By the polynomial long division, rational function $r_{[n,m]}(z)$ can be written as the sum of a polynomial and a proper rational function [61]

$$r_{[n,m]}(z) = s_{n-m}(z) + \frac{p_{m-1}(z)}{q_m(z)} \equiv s_{n-m}(z) + r_{[m-1,m]}(z), \tag{2.12}$$

where $s_{n-m}(z) = s_0 + s_1 z + \cdots + s_{n-m} z^{n-m}$ is a polynomial of degree no more than $n - m$ [61]. Subsequently we can obtain a realization of $r_{[m-1,m]}(z)$ as (2.9) or (2.11). .

Here we present three examples, which are going to be used extensively in computing resonant modes of accelerator cavities to be discussed in Sections 6.1 and 6.2.

Example 2.3.1 (Padé approximation of $\sqrt{z+1}$)**.** By [4], an order-(m, m) Padé approximation of $\sqrt{z+1}$ in the partial fraction form is given by

$$r_{[m,m]}(z) = 1 + \sum_{j=1}^{m} \frac{\eta_j z}{1 + \xi_j z} = 1 + \sum_{j=1}^{m} \left(\frac{\eta_j}{\xi_j} - \frac{\eta_j/\xi_j}{1 + \xi_j z} \right) = \gamma_m - \sum_{j=1}^{m} \frac{\frac{\eta_j}{\xi_j}}{1 + \xi_j z}, \tag{2.13}$$

where

$$\eta_j = \frac{2}{2m+1} \sin^2 \frac{j\pi}{2m+1}, \quad \xi_j = \cos^2 \frac{j\pi}{2m+1}, \quad \gamma_m = 2m+1$$

and the last equality of (2.13) use the fact[2] that

$$\sum_{j=1}^{m} \frac{\eta_j}{\xi_j} = \frac{2}{2m+1} \sum_{j=1}^{m} \tan^2 \frac{j\pi}{2m+1} = \frac{2(2m^2+m)}{2m+1} = 2m.$$

The poles of rational function $r_{[m]}(z)$ are $-1/\xi_j$ for $j = 1, 2, \ldots, m$.

In a realization form, $r_{[m]}(z)$ can be written as

$$r_{[m,m]}(z) = -a_m^{\mathrm{T}} (I_m - z D_m)^{-1} a_m + \gamma_m, \tag{2.14}$$

where

$$a_m = \left[(\eta_1/\xi_1)^{1/2}, \ldots, , (\eta_m/\xi_m)^{1/2} \right], \quad D_m = -\operatorname{diag}(\xi_1, \xi_2, \ldots, \xi_m), \quad \gamma_m = 2m+1. \tag{2.15}$$

Let $e_m(z) = \sqrt{z+1} - r_{[m]}(z)$ be the error of approximation. Then it is shown [43] that for $z \in \mathbb{R}$ and $z > -1$,

$$e_m(z) = 2\sqrt{z+1} \frac{\delta^{2m+1}}{1 + \delta^{2m+1}} \tag{2.16}$$

[2] http://functions.wolfram.com/01.08.23.0007.01/, accessed April 2018.

where $\delta = (\sqrt{z+1}-1)/(\sqrt{z+1}+1)$. We note that the error formula is also true for complex z since the proof did not use the assumption about $z \in \mathbb{R}$ and works for complex z with $\mathrm{Re}(z) > -1$ [43, 42]. The error $e_m(z)$ for real $z \in (-1, +\infty)$ has the following properties:

1. $e_m(0) = 0$.

2. $e_m(z)$ is increasing in $(-1, +\infty)$, which can be verified by taking the derivative of $e_m(z)$.

3. $e_m(z) < 0$ for $z \in (-1, 0)$ and $e_m(z) > 0$ for $z \in (0, +\infty)$. This is due to the fact that $-1 < \delta^{2m+1} < 0$ for $z \in (-1, 0)$, $0 < \delta^{2m+1} < 1$ for $z > 0$ and $\sqrt{z+1}, 1 + \delta^{2m+1} > 0$ for all $z > -1$.

4. From the previous properties 2, 3, we have $|e_m(z)|$ is decreasing for $z \in (-1, 0)$ and increasing for $z \in (0, +\infty)$.

5. For any real $z > -1$, $\lim_{m\to\infty} e_m(z) = 0$ since for any $z > -1$, $-1 < \delta < 1$ and $\lim_{m\to\infty} \delta^{2m+1} = 0$.

Example 2.3.2 (Padé-type approximations of $\frac{1}{\sqrt{z+1}}$)**.** We first show that the function

$$u_{[m,m]}(z) = \frac{1}{2m+1} + \frac{2}{2m+1} \sum_{j=1}^{m} \frac{1}{1 + z\left(\sin^2 \frac{j\pi}{2m+1}\right)} \tag{2.17}$$

is an order-(m, m) Padé approximation of $\frac{1}{\sqrt{z+1}}$, i.e.,

$$\frac{1}{\sqrt{z+1}} - u_{[m,m]}(z) = O(z^{2m+1}). \tag{2.18}$$

Let us start with Taylor's expansion of $\frac{1}{\sqrt{z+1}}$:

$$\frac{1}{\sqrt{z+1}} = 1 + \sum_{j=1}^{2m} (-1)^j 2^{-j} \frac{(2j-1)!!}{j!} z^j + O(z^{2m+1}), \tag{2.19}$$

where $(2j-1)!! = 1 \cdot 3 \cdot 5 \cdots (2j-1)$. On the other hand, the Taylor expansion of $u_{[m,m]}(z)$ is given

by given by

$$u_{[m,m]}(z) = 1 + \frac{2}{2m+1}\sum_{j=1}^{2m}(-1)^j\left(\sum_{i=1}^{m}\sin^2\frac{i\pi}{2m+1}\right)^j z^j + O(z^{2m+1}) \tag{2.20}$$

$$= 1 + \frac{1}{2m+1}\sum_{j=1}^{2m}(-1)^j\left(\sum_{i=1}^{2m+1}\sin^{2j}\frac{i\pi}{2m+1}\right) z^j + O(z^{2m+1}) \tag{2.21}$$

$$= 1 + \frac{1}{2(2m+1)}\sum_{j=1}^{2m}(-1)^j\left(\sum_{i=1}^{2(2m+1)}\sin^{2j}\frac{i\pi}{2m+1}\right) z^j + O(z^{2m+1}) \tag{2.22}$$

$$= 1 + \frac{1}{2(2m+1)}\sum_{j=1}^{2m}(-1)^j\left(\sum_{i=1}^{2(2m+1)}\cos^{2j}(\frac{2i\pi}{2(2m+1)}+\frac{\pi}{2})\right) z^j + O(z^{2m+1}) \tag{2.23}$$

$$= 1 + \sum_{j=1}^{2m}(-1)^j 2^{-j}\frac{(2j-1)!!}{j!}z^j + O(z^{2m+1}) \tag{2.24}$$

where the equality (2.21) used the fact that $\sin(\pi - x) = \sin(x)$ and $\sin(\pi) = 0$, the equality (2.22) used the fact that $\sin^2(\pi + x) = \sin^2(x)$ and the equality (2.24) used the fact[3] that

$$\sum_{i=1}^{2(2m+1)}\cos^{2j}\left(\frac{2i\pi}{2(2m+1)}+\frac{\pi}{2}\right) = 2(2m+1)2^{-j}\frac{(2j-1)!!}{j!}, j = 1, 2, \cdots, 2m.$$

Consequently, the order of approximation (2.18) can be devised immediately from (2.19) and (2.24).

In fact, we can show that

$$u_{[m,m]}(z) = 1/r_{[m,m]}(z) \quad \text{for any } z \in \mathbb{C},\ \mathrm{Re}(z) > -1, \tag{2.25}$$

where $r_{[m,m]}(z)$ is defined in (2.13). Note that the degrees of the polynomials on the numerators and denominators of $u_{[m,m]}(z)$ and $r_{[m,m]}(z)$, are both m and the constants on the denominators are nonzero. Without loss of generality, let $r_{[m,m]}(z) = p_m(z)/q_m(z)$ and $u_{[m,m]}(z) = s_m(z)/t_m(z)$, where $p_m(z)$, $q_m(z)$, $s_m(z)$ and $t_m(z)$ are polynomials of degrees m, then

$$\sqrt{z+1} = \frac{p_m(z)}{q_m(z)} + O(z^{2m+1}) \tag{2.26}$$

and

$$\frac{1}{\sqrt{z+1}} = \frac{s_m(z)}{t_m(z)} + O(z^{2m+1}). \tag{2.27}$$

[3] From the following idenity available at http://functions.wolfram.com/ElementaryFunctions/Cos/23/01/0009/: $\sum_{k=1}^{n}\cos^{2q}(\frac{2k\pi p}{n}+\alpha) = \frac{n2^{-q}(2q-1)!!}{q!}; p, q \in \mathbb{N}^+, 2pq < n$

Since the poles of $\frac{p_m(z)}{q_m(z)}$ and $\frac{s_m(z)}{t_m(z)}$ satisfy $\mathrm{Re}(z) < -1$, $\frac{p_m(z)}{q_m(z)}$ and $\frac{s_m(z)}{t_m(z)}$ have no singularities at $z = 0$. Therefore, multiplying equations (2.26) and (2.27), we can get

$$1 = \frac{p_m(z)}{q_m(z)} \cdot \frac{s_m(z)}{t_m(z)} + O(z^{2m+1}). \tag{2.28}$$

Furthermore, $q_m(z)$ and $t_m(z)$ are nonzero for any $z \in \mathbb{C}$ with $\mathrm{Re}(z) > -1$. Therefore, multiplying (2.28) by $q_m(z)t_m(z)$ we can get

$$q_m(z)t_m(z) = p_m(z)s_m(z) + O(z^{2m+1}). \tag{2.29}$$

Note that the degrees for polynomials $q_m(z)t_m(z)$ and $p_m(z)s_m(z)$ are at most $2m$, so in order to make (2.29) satisfied, it is only possible that

$$q_m(z)t_m(z) = p_m(z)s_m(z),$$

which proves the identity (2.25).

By the identity (2.25), the error function is given by

$$\begin{aligned} \frac{1}{\sqrt{z+1}} - u_{[m]}(z) &= \frac{1}{\sqrt{z+1}} - \frac{1}{r_{[m]}(z)} = \frac{r_{[m]}(z) - \sqrt{z+1}}{r_{[m]}(z)\sqrt{z+1}} \\ &= -\frac{e_m(z)}{\sqrt{z+1}(\sqrt{z+1} - e_m(z))} = -\frac{2}{\sqrt{z+1}} \frac{\delta^{2m+1}}{1-\delta^{2m+1}}, \end{aligned} \tag{2.30}$$

where $\delta = (\sqrt{z+1} - 1)/(\sqrt{z+1} + 1)$.

Example 2.3.3 (Padé-type approximations of $\frac{z+\beta}{\sqrt{z+1}}$)**.** For a rational approximation of $g(z) = \frac{z+\beta}{\sqrt{z+1}}$, multiply $u_{[m]}(z)$ defined in Example 2.3.2 by $z + \beta$, we can get an order-$(m+1, m)$ rational approximation of $g(z)$. It is given by

$$\begin{aligned} h_{[m+1,m]}(z) &= (z+\beta)u_{[m]}(z) \\ &= \frac{2}{2m+1} \sum_{j=1}^{m} \frac{z+\beta}{1 + z\sin^2 \frac{j\pi}{2m+1}} + \frac{z+\beta}{2m+1} \\ &= \frac{2}{2m+1} \sum_{j=1}^{m} \frac{\beta - \csc^2 \frac{j\pi}{2m+1}}{1 + z\sin^2 \frac{j\pi}{2m+1}} + \frac{2}{2m+1} \sum_{j=1}^{m} \csc^2 \frac{j\pi}{2m+1} + \frac{\beta}{2m+1} + \frac{z}{2m+1} \\ &= \sum_{j=1}^{m} \frac{\frac{2}{2m+1}(\beta - \csc^2 \frac{j\pi}{2m+1})}{1 + z\sin^2 \frac{j\pi}{2m+1}} + \frac{3\beta + 4m(m+1)}{3(2m+1)} + \frac{z}{2m+1} \qquad (*) \\ &\equiv -\sum_{j=1}^{m} \frac{\tau_j}{1+\zeta_j z} + \kappa_m + \nu_m z \end{aligned}$$

where

$$\tau_j = \frac{2}{2m+1}\csc^2\frac{j\pi}{2m+1} - \frac{2\beta}{2m+1}, \quad \zeta_j = \sin^2\frac{j\pi}{2m+1}, \quad j = 1, \ldots, m,$$
$$\kappa_m = \frac{3\beta + 4m(m+1)}{3(2m+1)}, \quad \nu_m = \frac{1}{2m+1}$$

and the equality in $(*)$ is based on the fact[4]

$$\sum_{j=1}^{m} \csc^2\frac{j\pi}{2m+1} = \frac{2}{3}m(m+1).$$

By (2.18), we have

$$g(z) - h_{[m+1,m]}(z) = (z+\beta)\left(\frac{1}{\sqrt{z+1}} - u_{[m,m]}(z)\right) = O(z^{2m+1}).$$

A realization form of $h_{[m+1,m]}(z)$ can be written as follows

$$h_{[m+1,m]}(z) = -b_m^{\mathrm{T}}(I_m - zC_m)^{-1}b_m + \kappa_m + \nu_m z \tag{2.31}$$

where

$$b_m = [\tau_1^{1/2}, \ldots, \tau_m^{1/2}], \quad C_m = -\operatorname{diag}(\zeta_1, \ldots, \zeta_m), \quad \kappa_m = \frac{3\beta + 4m(m+1)}{3(2m+1)}, \quad \nu_m = \frac{1}{2m+1}. \tag{2.32}$$

The poles of the approximant $h_{[m+1,m]}(z)$ are $-1/\zeta_j$ for $j = 1, \ldots, m$.

By multiplying (2.30) by $z+\beta$, the error function is then given by

$$d_m(z,\beta) = \frac{z+\beta}{\sqrt{z+1}} - h_{[m+1,m]}(z) = -2\frac{z+\beta}{\sqrt{z+1}}\frac{\delta^{2m+1}}{1-\delta^{2m+1}} \tag{2.33}$$

for any $z \in \mathbb{C},\ \mathrm{Re}(z) > -1$.

Example 2.3.4 (Padé approximations of $e^{-\tau\lambda}$)**.** The nonlinear function in the delay eigenvalue problems [32] is of the form $e^{-\tau\lambda}$. Explicit form of Padé approximation of $e^{-\tau\lambda}$ of order-(m,m) is [3, Sec.1.2]

$$g_{[m,m]}(\lambda) = \frac{\gamma_0 + \cdots + \gamma_{m-1}(-\tau\lambda)^{m-1} + \gamma_m(-\tau\lambda)^m}{\xi_0 + \cdots + \xi_{m-1}(-\tau\lambda)^{m-1} + \xi_m(-\tau\lambda)^m} \tag{2.34}$$

where

$$\gamma_j = \frac{(2m-j)!m!}{(2m)!j!(m-j)!}, \quad \xi_j = \frac{(-1)^j(2m-j)!m!}{(2m)!j!(m-j)!}$$

[4] Available at http://functions.wolfram.com/ElementaryFunctions/Csc/23/01/0003/

for $j = 1, \ldots, m$. A realization of rational function $g_{[m]}(\lambda)$ as

$$g_{[m,m]}(\lambda) = -a_m^{\mathrm{T}}[(-\tau I_m)\lambda - D_m]^{-1} b_m + d_m \tag{2.35}$$

where

$$D_m = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -h_0 & -h_1 & -h_2 & \cdots & -h_{m-1} \end{bmatrix} \in \mathbb{C}^{m \times m}$$

$$a_m = -[g_0, g_1, \cdots, g_{m-1}]^{\mathrm{T}} \in \mathbb{C}^m,\ b_m = [0, 0, \cdots, 0, 1]^{\mathrm{T}} \in \mathbb{R}^m$$

and

$$g_j = (\gamma_j - (-1)^m \xi_j)/\xi_m,\ h_j = \xi_j/\xi_m,\ d_m = \gamma_m/\xi_m = (-1)^m.$$

Chapter 3

Constrained Rayleigh quotient optimization

In Section 3.1, we discuss the constrained Rayleigh quotient optimization. In Section 3.2, we discuss the existence of the solution, provide a way to transform the optimization problem to a Lagrange multiplier problem then to a quadratic eigenvalue problem (QEP) and show a rigorous proof of the equivalency between these problems. An algorithm for solving the optimization problem is shown in 3.3 and the convergence result is provided in 3.4. Numerical examples are shown in Section 3.5. We summarize our work about the optimization problem in Section 3.6.

3.1 Problem statement

We are concerned with the following *linear constrained Rayleigh quotient* (CRQ) optimization:

$$\text{CRQopt:} \quad \begin{cases} \min\ v^{\mathrm{T}}Av, & \text{(3.1a)} \\ \text{s.t. } v^{\mathrm{T}}v = 1, & \text{(3.1b)} \\ \quad C^{\mathrm{T}}v = b, & \text{(3.1c)} \end{cases}$$

where $A \in \mathbb{R}^{n\times n}$ is symmetric, i.e., $A = A^{\mathrm{T}}$, $C \in \mathbb{R}^{n\times m}$ has full column rank, and $b \in \mathbb{R}^m$. Necessarily $m < n$ but often $m \ll n$. We are particularly interested in the case where A is large and sparse and $b \neq 0$.

3.2 Theory

3.2.1 Feasible set and solution existence

In CRQopt (3.1), we assumed $\operatorname{rank}(C) = m$. Let

$$n_0 = (C^{\mathrm{T}})^{\dagger} b, \tag{3.2}$$

i.e., n_0 is the unique minimal norm solution of $C^{\mathrm{T}} v = b$, where $C^{\dagger}$ is the Moore-Penrose inverse of C. Because of the assumption $\operatorname{rank}(C) = m$, we have [7, 10, 60]

$$C^{\dagger} = (C^{\mathrm{T}} C)^{-1} C^{\mathrm{T}}, \ (C^{\mathrm{T}})^{\dagger} = (C^{\dagger})^{\mathrm{T}} = C (C^{\mathrm{T}} C)^{-1}.$$

The most important orthogonal projection throughout this dissertation is

$$P = I - C C^{\dagger} \tag{3.3}$$

which orthogonally projects any vector onto $\mathcal{N}(C^{\mathrm{T}})$, the null space of C^{T} [60]. Any $v \in \mathbb{R}^n$ that satisfies $C^{\mathrm{T}} v = b$ can be orthogonally decomposed as

$$v = (I - P)v + Pv = n_0 + Pv \in n_0 + \mathcal{N}(C^{\mathrm{T}}). \tag{3.4}$$

Evidently $\|v\|^2 = \|n_0\|^2 + \|Pv\|^2$, which, together with the unit length constraint (3.1b), lead to the following immediate conclusions about the solvability of CRQopt (3.1):

- If $\|n_0\| > 1$, then there is no unit vector v satisfying $C^{\mathrm{T}} v = b$. This is because for any v satisfying $C^{\mathrm{T}} v = b$ has norm no smaller than $\|n_0\|$. Thus CRQopt (3.1) has no minimizer.
- If $\|n_0\| = 1$, then $v = n_0$ is the only unit vector that satisfies $C^{\mathrm{T}} v = b$. Thus CRQopt (3.1) has a unique minimizer $v = n_0$.
- If $\|n_0\| < 1$, then there are infinitely many feasible vectors v that satisfy $C^{\mathrm{T}} v = b$.

Therefore only the case $\|n_0\| < 1$ needs further investigation. Consequently, throughout the rest of the dissertation, we will assume $\|n_0\| < 1$.

3.2.2 Equivalent LGopt

Using the orthogonal decomposition (3.4), we have

$$v^{\mathrm{T}}Av = v^{\mathrm{T}}PAPv + 2v^{\mathrm{T}}PAn_0 + n_0^{\mathrm{T}}An_0, \tag{3.5a}$$

$$v^{\mathrm{T}}v = \|n_0\|^2 + \|Pv\|^2. \tag{3.5b}$$

Since $n_0^{\mathrm{T}}An_0$ and $\|n_0\|$ are constants, CRQopt (3.1) is equivalent to the following constrained quadratic minimization problem

$$\text{CQopt:} \quad \begin{cases} \min\ v^{\mathrm{T}}PAPv + 2v^{\mathrm{T}}b_0, & \text{(3.6a)} \\ \text{s.t. } \|Pv\| = \gamma, & \text{(3.6b)} \\ \qquad v \in n_0 + \mathcal{N}(C^{\mathrm{T}}), & \text{(3.6c)} \end{cases}$$

where

$$b_0 = PAn_0 \in \mathcal{N}(C^{\mathrm{T}}), \quad \gamma := \sqrt{1 - \|n_0\|^2} > 0. \tag{3.7}$$

Necessarily, $0 < \gamma < 1$. However, in the rest of our development, unless we refer back to CRQopt (3.1), $\gamma < 1$ can be removed, i.e., γ can be any positive number.

Theorem 3.2.1. *v_* is a minimizer of* CRQopt (3.1) *if and only if v_* is a minimizer of* CQopt (3.6).

One way to solve CQopt (3.6) is the method of the Lagrangian multipliers. It seeks the stationary points of the Lagrangian function

$$\mathscr{L}(v, \lambda) = v^{\mathrm{T}}PAPv + 2v^{\mathrm{T}}b_0 - \lambda(v^{\mathrm{T}}Pv - \gamma^2). \tag{3.8}$$

Differentiating $\mathscr{L}$ with respect to v and λ, we get

$$(PA - \lambda I)Pv = -b_0, \tag{3.9a}$$

$$\|Pv\| = \gamma. \tag{3.9b}$$

Let $u = Pv \in \mathcal{N}(C^{\mathrm{T}})$. Then $u = Pu$ and $v = n_0 + u$. The Lagrangian equations in (3.9) are equivalent to the following equations:

$$(PAP - \lambda I)u = -b_0, \tag{3.10a}$$

$$\|u\| = \gamma, \tag{3.10b}$$

$$u \in \mathcal{N}(C^{\mathrm{T}}). \tag{3.10c}$$

In fact, any solution (λ, v) of (3.9) gives rise to a solution (λ, u) with $u = Pv$ of (3.10), and conversely any solution (λ, u) of (3.10) leads to a solution (λ, v) with $v = n_0 + u$ of (3.9).

The system of equations (3.10) has more than one solution pairs (λ, u). We seek a pair (λ, u) among them that minimizes the objective function of (3.6) for $v \in \mathbb{R}^n$. Note that

$$\begin{aligned} f(v) :&= v^{\mathrm{T}} PAPv + 2v^{\mathrm{T}} b_0 \\ &= v^{\mathrm{T}} PAPv + 2v^{\mathrm{T}} PAn_0 \\ &\overset{u=Pv}{=} u^{\mathrm{T}} Au + 2u^{\mathrm{T}} An_0 \\ &\overset{u=Pu}{=} u^{\mathrm{T}} PAPu + 2u^{\mathrm{T}} PAn_0 \\ &= u^{\mathrm{T}} PAPu + 2u^{\mathrm{T}} b_0 \\ &= f(u), \end{aligned} \tag{3.11}$$

i.e., $f(v) = f(u)$ for $v \in \mathbb{R}^n$ and $u = Pv$. Therefore minimizing $f(v)$ over $v \in \mathbb{R}^n$ is equivalent to minimizing $f(u)$ over $u \in \mathcal{N}(C^{\mathrm{T}})$. The following lemma compares the value of f at different solution pairs (λ, u) of the system (3.10). The proof of the lemma is inspired by Gander [16] on solving a least squares problem with a quadratic constraint,

Lemma 3.2.1. *For two solution pairs (λ_i, u_i) for $i = 1, 2$ of the Lagrangian system of equations (3.10), $\lambda_1 < \lambda_2$ if and only if $f(u_1) < f(u_2)$.*

Proof. The proof relies on the following three facts:

1. For any solution pair (λ, u) of (3.10), we have

$$\lambda u = PAPu + b_0 \quad \Rightarrow \quad \lambda = \frac{1}{u^{\mathrm{T}} u} u^{\mathrm{T}} (PAPu + b_0) = \frac{1}{\gamma^2} u^{\mathrm{T}} (PAPu + b_0). \tag{3.12}$$

2. Given (λ_i, u_i) for $i = 1, 2$, satisfying (3.10), we have

$$\begin{aligned} f(u_1) &= u_1^{\mathrm{T}} PAPu_1 + 2u_1^{\mathrm{T}} b_0 \\ &\overset{(3.10a)}{=} -b_0^{\mathrm{T}} u_1 + \lambda_1 u_1^{\mathrm{T}} u_1 + 2u_1^{\mathrm{T}} b_0 \\ &\overset{(3.10b)}{=} u_1^{\mathrm{T}} b_0 + \lambda_1 \gamma^2 \\ &\overset{(3.10a)}{=} -u_2^{\mathrm{T}} (PAP - \lambda_2 I) u_1 + \lambda_1 \gamma^2. \end{aligned}$$

Similarly, we have $f(u_2) = -u_1^{\mathrm{T}} (PAP - \lambda_1 I) u_2 + \lambda_2 \gamma^2$. Therefore

$$f(u_1) - f(u_2) = (\lambda_1 - \lambda_2)(\gamma^2 - u_1^{\mathrm{T}} u_2). \tag{3.13}$$

3. For u_i of norm γ, by the Cauchy-Schwartz inequality, $u_1^{\mathrm{T}}u_2 \le \|u_1\|\,\|u_2\| = \gamma^2$, and $u_1^{\mathrm{T}}u_2 = \|u_1\|\,\|u_2\| = \gamma^2$ if and only if $u_1 = u_2$. Hence if $u_1 \ne u_2$, then $\gamma^2 - u_1^{\mathrm{T}}u_2 > 0$.

Now we are ready to prove the claim of the lemma. If $\lambda_1 < \lambda_2$, then $u_1 \ne u_2$ otherwise (3.12) would imply $\lambda_1 = \lambda_2$, and thus $f(u_1) < f(u_2)$ by (3.13). On the other hand, if $f(u_1) < f(u_2)$, then $\gamma^2 - u_1^{\mathrm{T}}u_2 > 0$ because $\gamma^2 - u_1^{\mathrm{T}}u_2 \ge 0$ always and it cannot be 0 by (3.13), and thus $\lambda_1 - \lambda_2 < 0$ again by (3.13). □

As a consequence of Lemma 3.2.1, we find that solving CQopt (3.6) is equivalent to solving the smallest Lagrangian multiplier λ of (3.8), i.e., those λ that satisfy (3.10). Specifically, solving CQopt (3.6) is equivalent to solving the following Lagrangian minimization problem:

$$\text{LGopt:}\quad \begin{cases} \min\ \lambda & \text{(3.14a)} \\ \text{s.t. } (PAP - \lambda I)u = -b_0, & \text{(3.14b)} \\ \quad \|u\| = \gamma, & \text{(3.14c)} \\ \quad u \in \mathcal{N}(C^{\mathrm{T}}). & \text{(3.14d)} \end{cases}$$

Theorem 3.2.2. *If v_* is a minimizer of* CQopt (3.6)*, then (λ_*, u_*) with*

$$u_* = Pv_*,\ \lambda_* = \frac{1}{\gamma^2}u_*^{\mathrm{T}}(PAPu_* + b_0)$$

is a minimizer of LGopt (3.14)*. Conversely if (λ_*, u_*) is a minimizer of* LGopt (3.14)*, then $v_* = n_0 + u_*$ is a minimizer of* CQopt (3.6)*.*

The case $b_0 = PAn_0 = 0$, which includes but is not equivalent to the homogeneous CRQopt (3.1) (i.e., $b = 0$) [22, 21], can be dealt with as follows. Suppose $b_0 = 0$ and let θ_1 be the smallest eigenvalue of PAP. Keep in mind that PAP always has an eigenvalue 0 with multiplicity m associated with the subspace $\mathcal{N}(C^{\mathrm{T}})^{\perp} = \mathcal{R}(C)$, the column space of C. There are the following two subcases:

- **Subcase $\theta_1 \ne 0$:** Then[1] $\theta_1 < 0$. Let z_1 be a corresponding eigenvector of PAP. Then $z_1 = PAPz_1/\theta_1 \in \mathcal{N}(C^{\mathrm{T}})$. So (θ_1, z_1) is a minimizer of LGopt (3.14) and therefore z_1 is a minimizer of CQopt (3.6), which in turn implies that $v_* = n_0 + \gamma z_1/\|z_1\|$ is a minimizer of CRQopt (3.1).

[1] This cannot happen if A is positive semidefinite.

- **Subcase $\theta_1 = 0$:** If there exists a corresponding eigenvector $z_1 \in \mathcal{N}(C^{\mathrm{T}})$, i.e., $Pz_1 \neq 0$, then (θ_1, Pz_1) is a minimizer of LGopt (3.14) and therefore Pz_1 is a minimizer of CQopt (3.6), which in turn implies that $v_* = n_0 + \gamma Pz_1/\|Pz_1\|$ is a minimizer of CRQopt (3.1). Otherwise there exists no corresponding eigenvector z_1 such that $Pz_1 \neq 0$. Let θ_2 be the second smallest eigenvalue of PAP, which is nonzero, and z_2 a corresponding eigenvector. Then $z_2 = PAPz_2/\theta_2 \in \mathcal{N}(C^{\mathrm{T}})$, and (θ_2, z_2) is a minimizer of LGopt (3.14) and therefore z_2 is a minimizer of CQopt (3.6), which in turn implies that $v_* = n_0 + \gamma z_2/\|z_2\|$ is a minimizer of CRQopt (3.1).

In view of such a quick resolution for the case $b_0 = 0$, in the rest of this dissertation, we will assume

$$b_0 = PAn_0 \neq 0. \tag{3.15}$$

3.2.3 Equivalent QEPmin

Let (λ, u) be a feasible pair of LGopt (3.14) and $\lambda \notin \mathrm{eig}(PAP)$. We can write $u = -(PAP - \lambda I)^{-1}b_0$, and then

$$\gamma^2 = u^{\mathrm{T}}u = b_0^{\mathrm{T}}(PAP - \lambda I)^{-2}b_0 = b_0^{\mathrm{T}}z, \tag{3.16}$$

where $z = (PAP - \lambda I)^{-2}b_0$, or equivalently, $(PAP - \lambda I)^2 z = b_0$. Therefore $b_0^{\mathrm{T}}z/\gamma^2 = 1$ by (3.16), and thus the pair (λ, z) satisfies the quadratic eigenvalue problem (QEP):

$$(PAP - \lambda I)^2 z = b_0 = b_0 \cdot 1 = b_0\left(b_0^{\mathrm{T}}z/\gamma^2\right) = \frac{1}{\gamma^2}b_0 b_0^{\mathrm{T}}z. \tag{3.17}$$

We claim that any z satisfying (3.17) is in $\mathcal{N}(C^{\mathrm{T}})$. To see this, we expand $(PAP - \lambda I)^2 z$ and extract $\lambda^2 z$ from $(PAP - \lambda I)^2 z = b_0$ to get

$$z = \frac{1}{\lambda^2}\left[-(PAP)^2 z + 2\lambda \cdot PAPz + b_0\right] \in \mathcal{N}(C^{\mathrm{T}}),$$

where we have used the assumption $\lambda \notin \mathrm{eig}(PAP)$ to conclude $\lambda \neq 0$, and $b_0 = PAn_0 \in \mathcal{N}(C^{\mathrm{T}})$. Therefore we have shown that under the assumption that LGopt (3.14) has no feasible pair (λ, u) with $\lambda \in \mathrm{eig}(PAP)$, any feasible pair (λ, u) of LGopt (3.14) satisfies QEP (3.17) with $z \in \mathcal{N}(C^{\mathrm{T}})$.

Next, we prove that any pair (λ, z) satisfying

$$0 \neq z \in \mathcal{N}(C^{\mathrm{T}}),\ \lambda \notin \mathrm{eig}(PAP) \text{ and QEP (3.17)}, \tag{3.18}$$

leads to a feasible pair of the Lagrange equations (3.14). First we note that $b_0^{\mathrm{T}} z \neq 0$; otherwise we would have $(PAP - \lambda I)^2 z = 0$ by (3.17), implying $z = 0$ since $\lambda \notin \mathrm{eig}(PAP)$, a contradiction. Let (λ, z) be a scalar-vector pair that satisfying (3.18). Define $u := -(PAP - \lambda I)^{-1} b_0$. Then $(PAP - \lambda I)u = -b_0$, i.e., (3.14b) holds, and also

$$\lambda u = PAPu + b_0 \quad \Rightarrow \quad u = \frac{1}{\lambda}(PAPu + b_0) \in \mathcal{N}(C^{\mathrm{T}}),$$

i.e., (3.14d) holds. Without loss of generality, we may scale z such that $b_0^{\mathrm{T}} z = \gamma^2$. It follows from (3.17) that

$$(PAP - \lambda I)^2 z = b_0 \quad \Rightarrow \quad z = (PAP - \lambda I)^{-2} b_0,$$

implying

$$1 = \frac{1}{\gamma^2} b_0^{\mathrm{T}} z = \frac{1}{\gamma^2} b_0^{\mathrm{T}} (PAP - \lambda I)^{-2} b_0 = \frac{1}{\gamma^2} u^{\mathrm{T}} u \quad \Rightarrow \quad \|u\| = \gamma,$$

i.e., (3.14c) holds. Lemma 3.2.2 summarizes what we have just proved.

Lemma 3.2.2. *Suppose the constraints of* LGopt (3.14) *has no feasible pair* (λ, u) *with* $\lambda \in \mathrm{eig}(PAP)$*, and suppose that* QEP (3.17) *has no solution pair* (λ, z) *with* $0 \neq z \in \mathcal{N}(C^{\mathrm{T}})$ *and* $\lambda \in \mathrm{eig}(PAP)$*. Then any pair* (λ, u) *satisfying the constraints of* LGopt (3.14) *gives rise to a pair* (λ, z) *with* $z = (PAP - \lambda I)^{-2} b_0$ *that satisfies* QEP (3.17)*. Conversely, any pair* (λ, z) *with* $z \neq 0$ *satisfying* QEP (3.17) *leads to a pair* (λ, u) *with* $u := -(PAP - \lambda I)^{-1} b_0$ *that satisfies the constraints of* LGopt (3.14)*.*

As a corollary of Lemma 3.2.2, we conclude that LGopt (3.14) is equivalent to

$$\text{QEPmin:} \quad \begin{cases} \min \ \lambda & \text{(3.19a)} \\ \text{s.t. } (PAP - \lambda I)^2 z = \gamma^{-2} b_0 b_0^{\mathrm{T}} z, & \text{(3.19b)} \\ \qquad \lambda \in \mathbb{R}, 0 \neq z \in \mathcal{N}(C^{\mathrm{T}}), & \text{(3.19c)} \end{cases}$$

under the assumptions of Lemma 3.2.2. Soon we show that LGopt (3.14) and QEPmin (3.19) are still equivalent even without the assumptions.

We name the minimization problem (3.19) QEPmin because the constraint (3.19b) is a quadratic eigenvalue problem (QEP). Although this QEP generally may have complex eigenvalues λ, the "min" in (3.19a) implicitly restricts the consideration only to the real eigenvalues λ of QEP (3.19b) in the context of QEPmin (3.19). In this sense, there is no need to specify $\lambda \in \mathbb{R}$ in

(3.19c), but we are doing it anyway to emphasize the implication. This comment applies to two other minimization problems pQEPmin (3.28) and rQEPmin (3.62) later that involve a QEP as a constraint as well.

In the rest of this section, we prove the equivalence between LGopt (3.14) and QEPmin (3.19) without the assumptions of Lemma 3.2.2. The key idea is is to remove the null space conditions $u, z \in \mathcal{N}(C^{\mathrm{T}})$ by projecting equations (3.14b), (3.14c) in LGopt and (3.19b) in QEPmin onto an appropriate subspace.

3.2.4 pLGopt

Let $S = [S_1,\ S_2] \in \mathbb{R}^{n\times n}$ be an orthogonal matrix with

$$\mathcal{R}(S_1) = \mathcal{N}(C^{\mathrm{T}}),\ \mathcal{R}(S_2) = \mathcal{N}(C^{\mathrm{T}})^{\perp}. \tag{3.20}$$

Since $\mathrm{rank}(C) = m$, we know $S_1 \in \mathbb{R}^{n\times(n-m)}$ and $S_2 \in \mathbb{R}^{n\times m}$. It can be verified that the projection matrix $P = I - CC^{\dagger}$ in (3.3) can be written as

$$P = S_1 S_1^{\mathrm{T}} = I - S_2 S_2^{\mathrm{T}}, \tag{3.21}$$

and we have

$$PS_1 = S_1,\ PS_2 = 0. \tag{3.22}$$

Set

$$g_0 = S_1^{\mathrm{T}} b_0,\ H = S_1^{\mathrm{T}} PAPS_1 = S_1^{\mathrm{T}} AS_1 \in \mathbb{R}^{(n-m)\times(n-m)}, \tag{3.23}$$

we have

$$S^{\mathrm{T}} PAPS = \begin{bmatrix} S_1^{\mathrm{T}} PAPS_1 & S_1^{\mathrm{T}} PAPS_2 \\ S_2^{\mathrm{T}} PAPS_1 & S_2^{\mathrm{T}} PAPS_2 \end{bmatrix} = \begin{matrix} {\scriptstyle n-m} \\ {\scriptstyle m} \end{matrix} \overset{\begin{matrix} {\scriptstyle n-m} & {\scriptstyle m} \end{matrix}}{\begin{bmatrix} H & 0 \\ 0 & 0 \end{bmatrix}}, \tag{3.24a}$$

$$S^{\mathrm{T}} b_0 = \begin{bmatrix} S_1^{\mathrm{T}} b_0 \\ S_2^{\mathrm{T}} b_0 \end{bmatrix} = \begin{matrix} {\scriptstyle n-m} \\ {\scriptstyle m} \end{matrix} \begin{bmatrix} g_0 \\ 0 \end{bmatrix}. \tag{3.24b}$$

Immediately from the decomposition (3.24a), we conclude the following lemma:

Lemma 3.2.3. *The eigenvalues of PAP consist of those of H and 0 with multiplicities m, i.e.,* $\mathrm{eig}(PAP) = \mathrm{eig}(H) \cup \{0, 0, \ldots, 0\}$. *If $0 \neq \lambda \in \mathrm{eig}(PAP)$, then $\lambda \in \mathrm{eig}(H)$ and its associated eigenvector must be in $\mathcal{N}(C^{\mathrm{T}})$. The matrix PAP has more than m eigenvalues 0 if and only if H is singular. For each eigenvalue 0 of PAP coming from* $\mathrm{eig}(H)$, *there is an eigenvector z of PAP such that $Pz \neq 0$ (in fact, Pz is an eigenvector for that particular eigenvalue 0 as well).*

To explicitly eliminate the constraint $u \in \mathcal{N}(C^{\mathrm{T}})$ in LGopt (3.14), we project LGopt (3.14) onto $\mathcal{R}(S_1)$ and introduce the following projected minimization problem

$$\text{pLGopt:} \quad \begin{cases} \min \ \lambda & \text{(3.25a)} \\ \text{s.t. } (H - \lambda I)y = -g_0, & \text{(3.25b)} \\ \quad \|y\| = \gamma. & \text{(3.25c)} \end{cases}$$

The next theorem establishes the equivalence between LGopt (3.14) and pLGopt (3.25).

Theorem 3.2.3. *The pair (λ_*, y_*) is a minimizer of* pLGopt (3.25) *if and only if (λ_*, u_*) with $u_* = S_1 y_*$ is a minimizer of* LGopt (3.14).

Proof. We begin by showing the equivalence between the constraints of LGopt (3.14) and those of pLGopt (3.25). Note that any $0 \neq u \in \mathcal{N}(C^{\mathrm{T}})$ can be expressed by $u = S_1 y$ for some $0 \neq y \in \mathbb{R}^{n-m}$ and vice versa. Making use of (3.24), we have

$$\begin{aligned} S^{\mathrm{T}}[(PAP - \lambda I)u + b_0] &= S^{\mathrm{T}}(PAP - \lambda I)SS^{\mathrm{T}}u + S^{\mathrm{T}}b_0 \\ &= \begin{bmatrix} H - \lambda I & 0 \\ 0 & -\lambda I \end{bmatrix} \begin{bmatrix} y \\ 0 \end{bmatrix} + \begin{bmatrix} g_0 \\ 0 \end{bmatrix}, \end{aligned} \tag{3.26}$$

and

$$u^{\mathrm{T}}u = y^{\mathrm{T}}S_1^{\mathrm{T}}S_1 y = y^{\mathrm{T}}y. \tag{3.27}$$

Now if (λ, u) satisfies the constraints of LGopt (3.14), then $S^{\mathrm{T}}[(PAP - \lambda I)u + b_0] = 0$ because of (3.14b), $u = S_1 y$ for some y because of (3.14d), and $\|y\| = \gamma$ because of (3.14c) and (3.27). It follows from (3.26) that $(H - \lambda I)y + g_0 = 0$. Thus (λ, y) satisfies the constraints of pLGopt (3.25).

On the other hand, suppose (λ, y) satisfies the constraints of pLGopt (3.25). Let $u = S_1 y \in \mathcal{N}(C^{\mathrm{T}})$. Both (3.26) and (3.27) remain valid. Then $S^{\mathrm{T}}[(PAP - \lambda I)u + b_0] = 0$ which implies

$(PAP-\lambda I)u+b_0 = 0$ because S^{T} is an orthogonal matrix. Also $\|u\| = \gamma$ by (3.27). This completes the proof of that (λ, u) satisfies the constraints of LGopt (3.14).

Therefore, LGopt (3.14) and pLGopt (3.25) have the same optimal value λ_*. More than that, if (λ_*, u_*) is a minimizer of LGopt (3.14), then there exists y_* such that $u_* = S_1 y_*$ and that (λ_*, y_*) is a minimizer of pLGopt (3.25), and vice versa. □

We note that for a modest-sized CRQopt (3.1), say n up to 2000, we may as well perform the reduction to form pLGopt (3.25) explicitly. Due to its modest size, pLGopt (3.25) can be solved as a dense matrix computational problem. The detail is buried later in the proof of Lemma 3.2.4.

3.2.5 pQEPmin

For the same purpose as we projected the Lagrange equations, we introduce the following projected minimization problem as the counterpart of QEPmin (3.19):

$$\text{pQEPmin:} \quad \begin{cases} \min \ \lambda & \text{(3.28a)} \\ \text{s.t. } (H-\lambda I)^2 w = \gamma^{-2} g_0 g_0^{\mathrm{T}} w, & \text{(3.28b)} \\ \qquad \lambda \in \mathbb{R}, w \neq 0. & \text{(3.28c)} \end{cases}$$

The equation in (3.28b) has an appearance of a QEP. As stated, the optimal value of pQEPmin (3.28) is the smallest real eigenvalue of QEP (3.28b). The next theorem establishes the equivalence between QEPmin (3.19) and pQEPmin (3.28).

Theorem 3.2.4. *The pair* (λ_*, w_*) *is a minimizer of* pQEPmin (3.28) *if and only if* (λ_*, z_*) *with* $z_* = S_1 w_*$ *is a minimizer of* QEPmin (3.19).

Proof. Similarly, we begin by showing the equivalence between the constraints of QEPmin (3.19) and those of pQEPmin (3.28). Keeping (3.24) in mind, we have for any $z = S_1 w$

$$\begin{aligned} & S^{\mathrm{T}} \left[(PAP-\lambda I)^2 z - \gamma^{-2}\, b_0 b_0^{\mathrm{T}} z \right] \\ & \quad = S^{\mathrm{T}} (PAP-\lambda I) S S^{\mathrm{T}} (PAP-\lambda I) S S^{\mathrm{T}} z - \gamma^{-2}\, S^{\mathrm{T}} b_0 b_0^{\mathrm{T}} S S^{\mathrm{T}} z \\ & \quad = \begin{bmatrix} (H-\lambda I)^2 & 0 \\ 0 & \lambda^2 I \end{bmatrix} \begin{bmatrix} w \\ 0 \end{bmatrix} - \begin{bmatrix} \gamma^{-2}\, g_0 g_0^{\mathrm{T}} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} w \\ 0 \end{bmatrix}. \end{aligned} \tag{3.29}$$

Now if (λ, z) satisfies the constraints of QEPmin (3.19), then $0 \neq z \in \mathcal{N}(C^{\mathrm{T}})$ and thus $z = S_1 w$ for some $0 \neq w \in \mathbb{R}^{n-m}$. Therefore, by (3.29), (λ, w) satisfies (3.28b).

On the other hand, suppose (λ, w) satisfies (3.28b) and (3.28c). Let $z = S_1 w \in \mathcal{N}(C^{\mathrm{T}})$. Then $z \neq 0$ and by (3.29), $S^{\mathrm{T}}[(PAP - \lambda I)^2 z - \gamma^{-2} b_0 b_0^{\mathrm{T}} z] = 0$. Since S^{T} is orthogonal, we get (3.19b). This proves that (λ, z) satisfies the constraints of QEPmin (3.19).

Therefore, QEPmin (3.19) and pQEPmin (3.28) have the same optimal value λ_*. More than that, if (λ_*, z_*) is a minimizer of QEPmin (3.19), then there exists $w_* \neq 0$ such that $z_* = S_1 w_*$ and that (λ_*, w_*) is a minimizer of pQEPmin (3.28), and vice versa. □

3.2.6 pLGopt and pQEPmin are equivalent

Although, in leading to pLGopt (3.25) and pQEPmin (3.28), the matrix H and the vector g_0 are derived from reducing A, C, and b in the original CRQopt (3.1), the developments in this section does not require that. Given this, in the rest of this section, we consider general pLGopt (3.25) and pQEPmin (3.28) with[2]

$$H \in \mathbb{R}^{\ell \times \ell},\ H^{\mathrm{T}} = H,\ 0 \neq g_0 \in \mathbb{R}^{\ell},\ \text{and } \gamma > 0.$$

To set up the stage for the rest of this subsection, we let $H = Y \Theta Y^{\mathrm{T}}$ be the eigen-decomposition of H:

$$H = Y \Theta Y^{\mathrm{T}} \text{ with } \Theta = \mathrm{diag}(\theta_1, \theta_2, \ldots, \theta_\ell),\ Y = [y_1, y_2, \ldots, y_\ell],\ Y^{\mathrm{T}} Y = I_\ell. \tag{3.30}$$

Without loss of generality, we arrange θ_i in the ascending order, i.e.,

$$\theta_1 = \theta_2 = \cdots = \theta_d < \theta_{d+1} \leq \cdots \leq \theta_\ell,$$

so $\lambda_{\min}(H) = \theta_1$. Define the secular function

$$\chi(\lambda) := g_0^{\mathrm{T}} (H - \lambda I)^{-2} g_0 - \gamma^2 = (Y^{\mathrm{T}} g_0)^{\mathrm{T}} (\Theta - \lambda I)^{-2} (Y^{\mathrm{T}} g_0) - \gamma^2 = \sum_{i=1}^{l} \frac{\xi_i^2}{(\lambda - \theta_i)^2} - \gamma^2, \tag{3.31}$$

where $\xi_i = g_0^{\mathrm{T}} y_i$ for $i = 1, 2, \cdots, n$, and let

$$j_0 = \min\{i \,:\, \xi_i \neq 0\}. \tag{3.32}$$

[2]Unlike before, there is no need to assume $\gamma < 1$. In addition, the size of square matrix H and vector g_0 can be arbitrary, not necessarily equal to $n - m$.

Lemma 3.2.4. *Let (λ_*, y_*) be a minimizer of* pLGopt (3.25). *The following statements hold.*

(a) $\lambda_* \le \lambda_{\min}(H)$.

(b) $\lambda_* = \lambda_{\min}(H)$ *if and only if*

$$g_0 \perp \mathcal{U} \text{ and } \|(H - \lambda_{\min}(H)I)^\dagger g_0\|_2 \le \gamma,$$

where $\mathcal{U}$ is the eigenspace of H associated with its eigenvalue $\lambda_{\min}(H)$.

(c) *If g_0 is not perpendicular to $\mathcal{U}$, then $\lambda_* < \lambda_{\min}(H)$ and λ_* is the smallest root of the secular function $\chi(\lambda)$, and $y_* = -(H - \lambda_* I)^{-1} g_0$.*

Proof. The secular function $\chi(\lambda)$ in (3.31) is continuous on $(-\infty, \theta_1)$ and $\lim_{\lambda \to -\infty} \chi(\lambda) = -\gamma^2 < 0$. Since

$$\chi'(\lambda) = -2 \sum_{i=1}^{\ell} \frac{\xi_i^2}{(\lambda - \theta_i)^3} > 0 \quad \text{for } \lambda < \theta_1,$$

$\chi(\lambda)$ is strictly increasing in $(-\infty, \theta_1)$. We have the following situations to deal with:

(1) If g_0 is not perpendicular to $\mathcal{U}$, then $\sum_{i=1}^{d} \xi_i^2 > 0$, i.e., $j_0 \le d$, then $\lim_{\lambda \to \theta_1^-} \chi(\lambda) = +\infty > 0$. There exists a unique $\lambda_* \in (-\infty, \theta_1)$ such that $\chi(\lambda_*) = 0$. Let $y_* = -(H - \lambda_* I)^{-1} g_0$. We have

$$(H - \lambda_* I) y_* = -g_0, \quad y_*^{\mathrm{T}} y_* = g_0^{\mathrm{T}} (H - \lambda_* I)^{-2} g_0 = \chi(\lambda_*) + \gamma^2 = \gamma^2.$$

Therefore, (λ_*, y_*) satisfies the constraints of pLGopt (3.25).

(2) Suppose that $g_0 \perp \mathcal{U}$, then $\sum_{i=1}^{d} \xi_i^2 = 0$, i.e., $j_0 > d$. Let

$$w = -(H - \theta_1 I)^\dagger g_0 = -\sum_{i=d+1}^{\ell} \frac{\xi_i}{\theta_i - \theta_1} y_i.$$

Then $(H - \theta_1 I) w = -g_0$ and $\lim_{\lambda \to \theta_1^-} \chi(\lambda) = w^{\mathrm{T}} w - \gamma^2$. There are three subcases to consider.

(i) If $\|w\| > \gamma$, then there exists a unique $\lambda_* \in (-\infty, \theta_1)$ such that $\chi(\lambda_*) = 0$. Moreover (λ_*, y_*) with $y_* = -(H - \lambda_* I)^{-1} g_0$ satisfies the constraints of pLGopt (3.25).

(ii) If $\|w\| = \gamma$, then (λ_*, y_*) with $\lambda_* = \theta_1$ and $y_* = w$ satisfies the constraints of pLGopt (3.25).

(iii) If $\|w\| < \gamma$, then (λ_*, y_*) with $\lambda_* = \theta_1$ and $y_* = w + \sqrt{\gamma^2 - \|w\|^2}\, y_1$ satisfies the constraints of pLGopt (3.25).

So far we have proved that (λ_*, y_*) satisfies the constraints of pLGopt (3.25) for all situations. Now we prove λ_* is the smallest Lagrange multiplier of pLGopt (3.25). Suppose there exists $\widehat{\lambda} < \lambda_*$ such that $(\widehat{\lambda}, \widehat{y})$ satisfies the constraints of pLGopt (3.25), then $\widehat{\lambda} < \lambda_* \le \theta_1$, so $\widehat{\lambda} \notin \mathrm{eig}(H)$. Therefore, in order to make $(\widehat{\lambda}, \widehat{y})$ satisfies (3.25b), we have $\widehat{y} = -(H - \widehat{\lambda} I)^{-1} g_0$. Note that $\lim_{\lambda \to \lambda_*^-} \chi(\lambda) \le 0$ for all cases and $\chi(\lambda)$ is strictly increasing in $(-\infty, \lambda_*)$, so $\chi(\widehat{\lambda}) = \widehat{y}^{\mathrm{T}} \widehat{y} - \gamma^2 < 0$, which is contradictory to (3.25c) that $\|\widehat{y}\| = \gamma$. Therefore, λ_* is the smallest Lagrangian multiplier, and thus (λ_*, y_*) is a minimizer of pLGopt (3.25).

For all situations, the smallest Lagrangian multiplier λ_* of pLGopt (3.25) satisfies $\lambda_* \le \lambda_{\min}(H)$, as expected. Also $\lambda_* = \theta_1$ can only happen in the subcase (ii) or (iii). □

Buried in the proof above is a viable numerical algorithm to solve pLGopt (3.25), provided λ_* in the case (a) and the subcase (i) of the case (b) can be efficiently solved. In both cases, it is the unique root of secular equation $\chi(\lambda) = 0$ in $(-\infty, \theta_1)$ in which $\chi(\lambda)$ monotonically increasing. A default method is Newton's method which applies the tangent line approximation, since both $\chi(\lambda)$ and its derivative $\chi'(\lambda)$ is rather straightforward to evaluate. However, this secular equation $\chi(\lambda) = 0$ has a special rational form. Previous ideas in solving secular equations of similar types [8, 17, 36, 70] can be adopted to devise a much fast method than Newton's method. Details are presented in Section 2.2.2.

Lemma 3.2.5. *If (λ, y) satisfies the constraints of* pLGopt (3.25)*, then there exists a vector $w \in \mathbb{R}^\ell$ such that (λ, w) satisfies the constraints of* pQEPmin (3.28)*. Specifically,*

$$w = \begin{cases} (H - \lambda I)^{-1} y, & \text{if } \lambda \notin \mathrm{eig}(H), \\ \text{the corresponding eigenvector of } H, & \text{if } \lambda \in \mathrm{eig}(H). \end{cases}$$

In particular, the optimal value of pQEPmin (3.28) *is less than or equal to the optimal value of* pLGopt (3.25)*.*

Proof. There are two cases to consider.

- Case $\lambda \in \mathrm{eig}(H)$: Let w be an eigenvector of H corresponding to eigenvalue λ, i.e., $Hw = \lambda w$. By (3.25b), $g_0 = -(H - \lambda I) y$, and thus

$$\gamma^{-2} g_0 g_0^{\mathrm{T}} w = -\gamma^{-2} g_0 y^{\mathrm{T}} (H - \lambda I) w = 0.$$

Evidently, $(H-\lambda I)^2 w = 0$. Hence (λ, w) satisfies (3.28b).

- Case $\lambda \notin \mathrm{eig}(H)$: Let $w = (H-\lambda I)^{-1} y$. Using (3.25b), we have

$$(H-\lambda I)^2 w = (H-\lambda I) y = -g_0,$$
$$\gamma^{-2} g_0 g_0^{\mathrm{T}} w = \gamma^{-2} g_0 g_0^{\mathrm{T}} (H-\lambda I)^{-1} y = -\gamma^{-2} g_0 y^{\mathrm{T}} y = -g_0.$$

Again (λ, w) satisfies (3.28b).

This proves that (λ, w) satisfies the constraints of pQEPmin (3.28). As a corollary, the optimal value of pQEPmin (3.28) is less than or equal to the optimal value of pLGopt (3.25). □

The next lemma claims a stronger conclusion than the last statement in the previous lemma.

Lemma 3.2.6. *The optimal value of* pLGopt (3.25) *is equal to the optimal value of* pQEPmin (3.28).

Proof. Let (λ_*, y_*) be a minimizer of pLGopt (3.25), and let $\widehat{\lambda}$ be the optimal value of pQEPmin (3.28). By Lemma 3.2.5, we have $\widehat{\lambda} \le \lambda_*$. It suffices to show that $\widehat{\lambda} < \lambda_*$ cannot happen. Assume, to the contrary, that $\widehat{\lambda} < \lambda_*$. By Lemma 3.2.4, we have $\widehat{\lambda} < \lambda_{\min}(H)$. In particular, $\widehat{\lambda} \notin \mathrm{eig}(H)$. Let $(\widehat{\lambda}, \widehat{w})$ be a minimizer of pQEPmin (3.28). By (3.28b), we have

$$\frac{1}{\gamma^2}(\widehat{w}^{\mathrm{T}} g_0)^2 = \widehat{w}^{\mathrm{T}} \frac{1}{\gamma^2} g_0 g_0^{\mathrm{T}} \widehat{w} = \widehat{w}^{\mathrm{T}} (H-\widehat{\lambda} I)^2 \widehat{w} > 0,$$

implying $g_0^{\mathrm{T}} \widehat{w} \neq 0$. Let $\widehat{y} = -(\gamma^2 / g_0^{\mathrm{T}} \widehat{w})\,(H-\widehat{\lambda} I)\widehat{w}$, and observe that

$$(H-\widehat{\lambda} I)\widehat{y} = -\frac{\gamma^2}{g_0^{\mathrm{T}} \widehat{w}} \cdot (H-\widehat{\lambda} I)^2 \widehat{w} = -\frac{\gamma^2}{g_0^{\mathrm{T}} \widehat{w}} \cdot \gamma^{-2} g_0 g_0^{\mathrm{T}} \widehat{w} = -g_0, \tag{3.33a}$$
$$\widehat{y}^{\mathrm{T}} \widehat{y} = \left(\frac{\gamma^2}{g_0^{\mathrm{T}} \widehat{w}}\right)^2 \widehat{w}^{\mathrm{T}} (H-\widehat{\lambda} I)^2 \widehat{w} = \left(\frac{\gamma^2}{g_0^{\mathrm{T}} \widehat{w}}\right)^2 \frac{\widehat{w}^{\mathrm{T}} g_0 g_0^{\mathrm{T}} \widehat{w}}{\gamma^2} = \gamma^2, \tag{3.33b}$$

i.e., $(\widehat{\lambda}, \widehat{y})$ satisfies the constraints of pLGopt (3.25). This implies $\lambda_* \le \widehat{\lambda}$, contradicting the assumption $\widehat{\lambda} < \lambda_*$. Therefore, $\widehat{\lambda} = \lambda_*$, as expected. □

We are ready to establish the equivalence between pLGopt (3.25) and pQEPmin (3.28).

Theorem 3.2.5 (pLGopt (3.25) and pQEPmin (3.28) are equivalent).

(1) *Let* (λ_*, y_*) *be a minimizer of* pLGopt (3.25). *Then either* $\lambda_* < \lambda_{\min}(H)$ *or* $\lambda_* = \lambda_{\min}(H)$, *and there exists* w_* *such that* (λ_*, w_*) *is a minimizer of* pQEPmin (3.28). *Specifically,*

$$w_* = \begin{cases} (H - \lambda_* I)^{-1} y_*, & \text{if } \lambda_* < \lambda_{\min}(H), \\ \text{the corresponding eigenvector of } H, & \text{if } \lambda_* = \lambda_{\min}(H). \end{cases}$$

(2) *Conversely, if* (λ_*, w_*) *is a minimizer of* pQEPmin (3.28), *then there exists* y_* *such that* (λ_*, y_*) *is a minimizer of* pLGopt (3.25). *Specifically,*

$$y_* = \begin{cases} -(\gamma^2 / g_0^{\mathrm{T}} w_*)\,(H - \lambda_* I) w_*, & \text{if } g_0^{\mathrm{T}} w_* \neq 0, \\ x_* + \sqrt{\gamma^2 - \|x_*\|^2}\,(w_* / \|w_*\|), & \text{if } g_0^{\mathrm{T}} w_* = 0, \end{cases}$$

where $x_* = -(H - \lambda_* I)^{\dagger} g_0$ *in the case* $g_0^{\mathrm{T}} w_* = 0$, *and it is guaranteed that* $\|x_*\| \leq \gamma$.

Proof. Item (1) is a consequence of Lemmas 3.2.5 and 3.2.6.

Consider item (2). Suppose (λ_*, w_*) is a minimizer of pQEPmin (3.28). By Lemma 3.2.6, it suffices to show that there exists y_* such that (λ_*, y_*) satisfies the constraints of pLGopt (3.25).

- Case $g_0^{\mathrm{T}} w_* \neq 0$: The equations in (3.33) hold with substitutions

  $$\widehat{\lambda} \to \lambda_*, \quad \widehat{y} \to y_* = -(\gamma^2 / g_0^{\mathrm{T}} w_*)\,(H - \lambda_* I) w_*.$$

  So (λ_*, y_*) satisfies the constraints of pLGopt (3.25).

- Case $g_0^{\mathrm{T}} w_* = 0$: By (3.28b), we find that $(H - \lambda_* I)^2 w_* = 0$, implying $(H - \lambda_* I) w_* = 0$ since $H - \lambda_* I$ is real symmetric. Hence $\lambda_* \in \mathrm{eig}(H)$ and w_* is an associated eigenvector. Let x_* be the minimum norm solution of $(H - \lambda_* I) x_* = -g_0$. Note that we already know λ_* is the optimal value of pLGopt (3.25), which means there exists y such that (λ_*, y) satisfies (3.25b) and $\|y\| = \gamma$. On the other hand, x is minimal norm solution of (3.25b), so $\|x\| \leq \|y\| = \gamma$. Then it can be verified that (λ_*, y_*) with $y_* = x_* + \sqrt{\gamma^2 - \|x_*\|^2}\,(w_* / \|w_*\|)$ satisfies the constraints of pLGopt (3.25).

This proves that (λ_*, y_*) satisfies the constraints of pLGopt (3.25). In addition, by Lemma 3.2.6, λ_* is the optimal value of pLGopt (3.25), which proves the result. □

The following theorem is about the uniqueness of the solution for pLGopt (3.25).

Theorem 3.2.6 (Uniqueness of the minimizer for pLGopt (3.25)). *Let (λ_*, w_*) be a minimizer of* pQEPmin (3.28).

(1) *If $g_0^{\mathrm{T}} w_* \neq 0$ for all possible minimizers for* pQEPmin (3.28)*, then $\lambda_* < \lambda_{\min}(H)$ and the minimizer of* pLGopt (3.25) *is unique.*

(2) *If there exists a minimizer for* pQEPmin (3.28) *such that $g_0^{\mathrm{T}} w_* = 0$, then $\lambda_* = \lambda_{\min}(H)$ and the minimizer of* pLGopt (3.25) *is unique if and only if $\|x_*\| = \gamma$, where $x_* = -(H - \lambda_* I)^{\dagger} g_0$.*

Proof. (1) First we prove $\lambda_* < \lambda_{\min}(H)$. Suppose it is not true, i.e., $\lambda_* = \lambda_{\min}(H)$, let w_* be an eigenvector of H corresponding with eigenvalue $\lambda_{\min}(H)$, then by Theorem 3.2.5, (λ_*, w_*) is a minimizer of pQEPmin (3.28). Since QEP (3.28b) leads to $\gamma^{-2} g_0 g_0^{\mathrm{T}} w_* = (H - \lambda_* I)^2 w_* = 0$ and $w_* \neq 0$, we have $g_0^{\mathrm{T}} w_* = 0$, which is contradictory to our assumption that $g_0^{\mathrm{T}} w_* \neq 0$ for all possible minimizers (λ_*, w_*) of pQEPmin (3.28). Therefore, $\lambda_* < \lambda_{\min}(H)$.

In this case $(\lambda_*, x_* = -(H - \lambda_* I)^{-1} g_0)$ is the unique minimizer of pLGopt (3.25) since the $H - \lambda_* I$ is nonsingular and x_* is the unique solution of (3.25b).

(2) Making use of (3.28b), we have

$$(H - \lambda_* I)^2 w_* = \gamma^{-2} g_0 g_0^{\mathrm{T}} w_* = 0 \quad \Rightarrow \quad (H - \lambda_* I) w_* = 0$$

because $H - \lambda_* I$ is real symmetric. Therefore $\lambda_* \in \mathrm{eig}(H)$, which yields $\lambda_* = \lambda_{\min}(H)$. Note that x_* is unique and w_* can be chosen arbitrarily in the eigenspance of H corresponding with eigenvalue $\lambda_{\min}(H)$, so w_* is not unique. Therefore, $y_* = x_* + \sqrt{\gamma^2 - \|x_*\|^2}\,(w_*/\|w_*\|)$ is unique if and only if $\|x_*\| = \gamma$.

□

Remark 3.2.1. In [17], the authors investigate the relationship between the problems

$$\text{pLG:} \quad (H - \lambda I) y = -g_0, \ \|y\| = \gamma, \tag{3.34}$$

$$\text{pQEP:} \quad (H - \lambda I)^2 w = \gamma^{-2} g_0 g_0^{\mathrm{T}} w, \ \lambda \in \mathbb{R}, \ w \neq 0. \tag{3.35}$$

They differ from pLGopt and pQEPmin, respectively, just without taking the min over λ. The following results were obtained there:

1. If (λ, y) is a solution of pLG (3.34), then there exists w such that (λ, w) is a solution of pQEP (3.35).

2. Suppose that (λ, w) is a solution of pQEP (3.35).

   - If $\lambda \notin \text{eig}(H)$, then there exists y such that (λ, y) is a solution of pLG (3.34).
   - If $\lambda \in \text{eig}(H)$, then there exists y such that (λ, y) is a solution of pLG (3.34) if and only if $\|(H - \lambda I)^{\dagger} g_0\| \leq \gamma$.

Consequently, these results provide no guarantee that for any solution (λ, w) of pQEP (3.35), there exists a corresponding solution (λ, y) of pLG (3.34). Nonetheless, the authors stated without any proof that for the solution (λ_*, w_*) of pQEP (3.35) with λ_* being the smallest eigenvalue of pQEP (3.35), there does exist a solution (λ_*, y_*) of pLGopt (3.25), a conclusion that doesn't look like a straightforward one to us. Because of that, in Theorem 3.2.5 we rigorously proved that for any minimizer (λ_*, w_*) of pQEPmin (3.28), there exists y_* such that (λ_*, y_*) is a minimizer of pLGopt (3.25). □

Next we will establish an important result in Theorem 3.2.7 below that says the leftmost eigenvalue of QEP (3.28b) is real. We begin by establishing a close relationship in Lemma 3.2.7 between the zeros of the secular function $\chi(\lambda)$ in (3.31) and the eigenvalues of QEP (3.28b), and then using the relation to expose an eigenvalue distribution property of QEP (3.28b) in Lemmas 3.2.8 and 3.2.9, in preparing for proving our main result in Theorem 3.2.7.

Lemma 3.2.7. *Suppose $\lambda \notin \text{eig}(H)$, λ (possibly complex) is an eigenvalue of* QEP (3.28b) *if and only if $\chi(\lambda) = 0$, where $\chi(\lambda)$ is defined in* (3.31).

Proof. Let $\chi(\lambda) = 0$ and $\lambda \notin \text{eig}(H)$. Define $z = (H - \lambda I)^{-2} g_0$. Then we have $(H - \lambda I)^2 z = g_0$ and

$$g_0^{\mathrm{T}} z = \sum_{i=1}^{\ell} \frac{\xi_i^2}{(\theta_i - \lambda)^2} = \gamma^2 \text{ and thus } (H - \lambda I)^2 z = g = \gamma^{-2} g g_0^{\mathrm{T}} z,$$

i.e., (λ, z) is an eigenpair of QEP (3.28b).

On the other hand, suppose λ is an eigenvalue of QEP (3.28b) and $\lambda \notin \text{eig}(H)$. Pre-multiply (3.28b) by $g_0^{\mathrm{T}}(H - \lambda I)^{-2}$ to get

$$g_0^{\mathrm{T}} z = \gamma^{-2} g_0^{\mathrm{T}} (H - \lambda I)^{-2} g_0 g_0^{\mathrm{T}} z. \tag{3.36}$$

We claim that $g_0^{\mathrm{T}} z \neq 0$. Otherwise, $(H-\lambda I)^2 z = 0$ by (3.28b), which implies $(H-\lambda I)z = 0$, i.e., $\lambda \in \mathrm{eig}(H)$, a contradiction. So $g_0^{\mathrm{T}} z \neq 0$ and thus it follows from (3.36) that

$$\gamma^{-2} g_0^{\mathrm{T}} (H-\lambda I)^{-2} g_0 = 1,$$

i.e., λ is a zero of $\chi(\lambda)$, as was to be shown. □

Lemma 3.2.8. QEP (3.28b) *has no eigenvalue* $\lambda = \alpha + \mathtt{i}\beta$ *with* $\alpha < \theta_{j_0}$ *and* $\beta \neq 0$, *where* α, $\beta \in \mathbb{R}$, $\mathtt{i}$ *is the imaginary unit, and* j_0 *is defined in* (3.32).

Proof. Suppose, to the contrary, that QEP (3.28b) has an eigenvalue $\lambda = \alpha + \mathtt{i}\beta$ with $\alpha < \theta_{j_0}$ and $\beta \neq 0$. Evidently $\lambda = \alpha + \mathtt{i}\beta \notin \mathrm{eig}(H)$ because all eigenvalues of H are real. By Lemma 3.2.7, $\alpha + \mathtt{i}\beta$ must be a zero of the secular function $\chi(\lambda)$ in (3.31), i.e.,

$$\begin{aligned} 0 = \chi(\alpha+\mathtt{i}\beta) &= \sum_{i=1}^{\ell} \frac{\xi_i^2}{(\alpha-\theta_i+\mathtt{i}\beta)^2} - \gamma^2 \\ &= \sum_{i=1}^{\ell} \frac{\xi_i^2}{(\alpha-\theta_i)^2 - \beta^2 + 2\mathtt{i}(\alpha-\theta_i)\beta} - \gamma^2 \\ &= \sum_{i=1}^{\ell} \frac{\xi_i^2[(\alpha-\theta_i)^2 - \beta^2 - 2\mathtt{i}(\alpha-\theta_i)\beta]}{[(\alpha-\theta_i)^2-\beta^2]^2 + 4\beta^2(\alpha-\theta_i)^2} - \gamma^2. \end{aligned}$$

In particular, the imaginary part of $\chi(\alpha+\mathtt{i}\beta)$ is zero, i.e.,

$$\sum_{i=1}^{\ell} \frac{-2(\alpha-\theta_i)\beta\xi_i^2}{[(\alpha-\theta_i)^2-\beta^2]^2 + 4\beta^2(\alpha-\theta_i)^2} = \beta \left(\sum_{i=j_0}^{\ell} \frac{-2(\alpha-\theta_i)\xi_i^2}{[(\alpha-\theta_i)^2-\beta^2]^2 + 4\beta^2(\alpha-\theta_i)^2} \right) = 0. \tag{3.37}$$

Since $\alpha < \theta_i$ for all $i \geq j_0$, $\xi_{j_0}^2 > 0$ and $\xi_i^2 \geq 0$ for all $i > j_0$, we know

$$\sum_{i=j_0}^{\ell} \frac{-2(\alpha-\theta_i)\xi_i^2}{[(\alpha-\theta_i)^2-\beta^2]^2 + 4\beta^2(\alpha-\theta_i)^2} > 0.$$

Therefore, by (3.37), we conclude $\beta = 0$, a contradiction. □

Lemma 3.2.9. QEP (3.28b) *has an eigenvalue* $\widetilde{\lambda} < \theta_{j_0}$ *(necessarily* $\widetilde{\lambda} \in \mathbb{R}$*), where* j_0 *is defined in* (3.32).

Proof. There are two possible cases:

- Case $\theta_{j_0} = \theta_1$: Without loss of generality, let $\xi_1 \neq 0$. Since $\chi(\lambda)$ is continuous and strictly increasing in $(-\infty, \theta_1)$, and

$$\lim_{\lambda \to -\infty} \chi(\lambda) = -\gamma^2 < 0, \quad \lim_{\lambda \to \theta_1^-} \chi(\lambda) \geq \lim_{\lambda \to \theta_1^-} \frac{\xi_1^2}{(\lambda - \theta_1)^2} - \gamma^2 = +\infty > 0,$$

there exists a zero $\widetilde{\lambda} \in (-\infty, \theta_1)$ of $\chi(\lambda)$. Evidently $\widetilde{\lambda} \notin \mathrm{eig}(H)$, and then by Lemma 3.2.7, $\widetilde{\lambda}$ must be an eigenvalue of QEP (3.28b).

- Case $\theta_{j_0} > \theta_1$: Let $\widetilde{\lambda} = \theta_1$ and $z = y_1$. We have $(H - \widetilde{\lambda} I)^2 z = (H - \widetilde{\lambda} I)^2 y_1 = 0$. Furthermore, $g_0^{\mathrm{T}} z = g_0^{\mathrm{T}} y_1 = \xi_1 = 0$. Therefore $(\widetilde{\lambda}, z)$ satisfies (3.28b), impliying $\widetilde{\lambda}$ is an eigenvalue of QEP (3.28b) and $\tilde{\lambda} = \theta_1 < \theta_{j_0}$.

The proof is completed. □

With the three lemmas above, now we are ready to prove our main result on the leftmost eigenvalue of QEP (3.28b).

Theorem 3.2.7. *The leftmost eigenvalue, by which we mean the one with the smallest real part, of* QEP (3.28b) *is real. As a consequence, the optimal value of* pQEPmin (3.28) λ_* *is the leftmost eigenvalue of* QEP (3.28b).

Proof. Let $\lambda_* = \alpha_* + \mathtt{i}\beta_*$ be the leftmost eigenvalue. By Lemma 3.2.9, QEP (3.28b) has a real eigenvalue $\widetilde{\lambda}$ with $\widetilde{\lambda} < \theta_{j_0}$. Hence $\alpha_* \leq \widetilde{\lambda} < \theta_{j_0}$, which together with Lemma 3.2.8 tell us that $\beta_* = 0$ and thus $\lambda_* \in \mathbb{R}$. □

Remark 3.2.2. In [59], the authors stated without proof that the rightmost eigenvalue of the QEP

$$((W + \lambda I)^2 - \delta^{-2} h h^{\mathrm{T}}) x = 0 \tag{3.38}$$

is real and positive, where W is a real symmetric matrix, h is a vector, and $\delta > 0$ is a scalar. It was pointed out in [35] that the rightmost eigenvalue of (3.38) may not always be positive and the authors proved in [35, Theorem 4.1] that the largest real eigenvalue of (3.38) is the rightmost eigenvalue. The authors applied a maximin principle for nonlinear eigenproblems for the proof. In Theorem 3.2.7 we have proved the leftmost eigenvalue λ_* of (3.28b) is real, i.e., there is no complex eigenvalue of QEP (3.28b) with real part equal to λ_* and nonzero complex part. This result cannot be obtained by the approach used in [35]. □

3.2.7 LGopt and QEPmin are equivalent

Theorem 3.2.5 says that pLGopt (3.25) and pQEPmin (3.28) are equivalent. Previously in Lemma 3.2.2, we showed that LGopt (3.14) and QEPmin (3.19) are also equivalent under the assumptions stated there. Our goal in this subsection is to have the assumptions of Lemma 3.2.2 removed.

For convenience, we restate LGopt (3.14) and QEPmin (3.19) as follows:

$$\text{LGopt:} \quad \begin{cases} \min \ \lambda & (3.14a) \\ \text{s.t. } (PAP - \lambda I)u = -b_0, & (3.14b) \\ \quad \|u\| = \gamma, & (3.14c) \\ \quad u \in \mathcal{N}(C^{\mathrm{T}}); & (3.14d) \end{cases}$$

$$\text{QEPmin:} \quad \begin{cases} \min \ \lambda & (3.19a) \\ \text{s.t. } (PAP - \lambda I)^2 z = \gamma^{-2} b_0 b_0^{\mathrm{T}} z, & (3.19b) \\ \quad \lambda \in \mathbb{R}, \ 0 \neq z \in \mathcal{N}(C^{\mathrm{T}}). & (3.19c) \end{cases}$$

Recall S_1 and S_2 as defined in (3.20) and H and g as defined in (3.23). Before stating our main result in this subsection, we need two lemmas. The first one is about an eigen-relationship between PAP and H and the second one is on the relationships among $PAP - \lambda I$, $H - \lambda I$, $(PAP - \lambda I)^\dagger$ and $(H - \lambda I)^\dagger$.

Lemma 3.2.10. *(λ, s) is an eigenpair of H if and only if $(\lambda, S_1 s)$ is an eigenpair of PAP with $S_1 s \in \mathcal{N}(C^{\mathrm{T}})$.*

Proof. This is a consequence of the decomposition (3.24a). □

Lemma 3.2.11. *For any $\lambda \in \mathbb{R}$, $(PAP - \lambda I)S_1 = S_1(H - \lambda I)$ and $(PAP - \lambda I)^\dagger S_1 = S_1(H - \lambda I)^\dagger$.*

Proof. Let $H = Y \Theta Y^{\mathrm{T}}$ be the eigen-decomposition of H, where $Y \in \mathbb{R}^{(n-m)\times(n-m)}$ is orthogonal and Θ is a diagonal matrix. Then the eigen-decomposition of PAP is given by

$$PAP = [S_1 \ S_2] \begin{bmatrix} Y & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \Theta & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} Y^{\mathrm{T}} & 0 \\ 0 & I \end{bmatrix} [S_1 \ S_2]^{\mathrm{T}}. \tag{3.39}$$

Therefore $(PAP - \lambda I)S_1 = S_1 Y(\Theta - \lambda I)Y^{\mathrm{T}} = S_1(H - \lambda I)$. On the other hand, for $\lambda \neq 0$,

$$(PAP - \lambda I)^{\dagger} = [S_1 \ S_2] \begin{bmatrix} Y & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} (\Theta - \lambda I)^{\dagger} & 0 \\ 0 & -\frac{1}{\lambda} I \end{bmatrix} \begin{bmatrix} Y^{\mathrm{T}} & 0 \\ 0 & I \end{bmatrix} [S_1 \ S_2]^{\mathrm{T}},$$

and for $\lambda = 0$,

$$(PAP)^{\dagger} = [S_1 \ S_2] \begin{bmatrix} Y & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \Theta^{\dagger} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} Y^{\mathrm{T}} & 0 \\ 0 & I \end{bmatrix} [S_1 \ S_2]^{\mathrm{T}}.$$

Hence $(PAP - \lambda I)^{\dagger} S_1 = S_1 Y(\Theta - \lambda I)^{\dagger} Y^{\mathrm{T}} = S_1(H - \lambda I)^{\dagger}$, as was to be shown. □

Now we are ready to state the main result of the subsection.

Theorem 3.2.8 (LGopt (3.14) and QEPmin (3.19) are equivalent).

(1) *Let* (λ_*, u_*) *be a minimizer of* LGopt (3.14). *Then there exists* z_* *such that* (λ_*, z_*) *is a minimizer of* QEPmin (3.19). *Specifically,*

$$z_* = \begin{cases} (PAP - \lambda_* I)^{\dagger} u_*, & \text{if } \lambda_* \notin \mathrm{eig}(PAP) \text{ or } \lambda_* \in \mathrm{eig}(PAP) \text{ but there is no} \\ & \text{corresponding eigenvector entirely in } \mathcal{N}(C^{\mathrm{T}}), \\ s, & \text{if } \lambda_* \in \mathrm{eig}(PAP) \text{ and there is a corresponding eigen-} \\ & \text{vector } s \in \mathcal{N}(C^{\mathrm{T}}). \end{cases}$$

(2) *Let* (λ_*, z_*) *be a minimizer of* QEPmin (3.19). *Then there exists* $u_* \in \mathbb{R}^n$ *such that* (λ_*, u_*) *is a minimizer of* LGopt (3.14). *Specifically,*

$$u_* = \begin{cases} -(\gamma^2 / b_0^{\mathrm{T}} z_*)(PAP - \lambda_* I) z_*, & \text{if } b_0^{\mathrm{T}} z_* \neq 0, \\ x_* + \sqrt{\gamma^2 - \|x_*\|^2}\, (z_* / \|z_*\|), & \text{if } b_0^{\mathrm{T}} z_* = 0, \end{cases}$$

where $x_* = -(PAP - \lambda_* I)^{\dagger} b_0$ *in the case* $b_0^{\mathrm{T}} z_* = 0$ *and it is guaranteed that* $\|x_*\| \leq \gamma$.

Proof. We prove item (1) first. By Theorem 3.2.3, (λ_*, y_*) with $y_* = S_1^{\mathrm{T}} u_*$ is a minimizer of pLGopt (3.25). We have two cases to consider.

(a) If $\lambda_* \notin \mathrm{eig}(PAP)$ or $\lambda_* \in \mathrm{eig}(PAP)$ but there is no corresponding eigenvector $s \in \mathcal{N}(C^{\mathrm{T}})$, then $\lambda \notin \mathrm{eig}(H)$ by Lemma 3.2.10. Using Theorem 3.2.5, we conclude that (λ_*, w_*) with

$$w_* = (H - \lambda_* I)^{-1} y_* = (H - \lambda_* I)^{\dagger} y_*$$

is a minimizer of pQEPmin (3.28). Now use Theorem 3.2.4 to conclude that (λ_*, z_*) with $z_* = S_1(H - \lambda_* I)^\dagger y_*$ is a minimizer of QEPmin (3.19). By Lemma 3.2.11,

$$z_* = S_1(H - \lambda_* I)^\dagger w_* = (PAP - \lambda_* I)^\dagger S_1 w_* = (PAP - \lambda_* I)^\dagger u_*.$$

(b) Suppose that $\lambda_* \in \mathrm{eig}(PAP)$ and there is a corresponding eigenvector $s \in \mathcal{N}(C^{\mathrm{T}})$. Then $s = S_1 r$ for some $0 \neq r \in \mathbb{R}^{n-m}$. By Lemma 3.2.10, r is an eigenvector of H corresponding to the eigenvalue λ_*. Use Theorem 3.2.5 to conclude that (λ_*, w_*) with $w_* = r$ is a minimizer of pQEPmin (3.28), which in turn, by Theorem 3.2.4, yields that (λ_*, z_*) with $z_* = s = S_1 r$ is a minimizer of QEPmin (3.19).

Next we consider item (2). By Theorem 3.2.4, (λ_*, w_*) with $w_* = S_1^{\mathrm{T}} z_*$ is a minimizer of pQEPmin (3.28). Since b_0, $z_* \in \mathcal{N}(C^{\mathrm{T}})$, we have $z_* = S_1 w_*$ and $b_0^{\mathrm{T}} z_* = g_0^{\mathrm{T}} S_1^{\mathrm{T}} S_1 w_* = g_0^{\mathrm{T}} w_*$.

- Case $b_0^{\mathrm{T}} z_* \neq 0$: We have $g_0^{\mathrm{T}} w_* \neq 0$. By Theorem 3.2.5, (λ_*, y_*) with $y_* = -(\gamma^2 / g_0^{\mathrm{T}} w_*)\,(H - \lambda_* I) w_*$ solves pLGopt (3.25). By Theorem 3.2.3, (λ_*, u_*) with $u_* = -(\gamma^2 / g_0^{\mathrm{T}} w_*)\, S_1 (H - \lambda_* I) w_*$ solves LGopt (3.14). Furthermore, by Lemma 3.2.11, $(PAP - \lambda_* I) z_* = (PAP - \lambda_* I) S_1 w_* = S_1 (H - \lambda_* I) w_*$. Therefore $u_* = -(\gamma^2 / g_0^{\mathrm{T}} w_*)\, S_1 (H - \lambda_* I) w_* = -(\gamma^2 / b_0^{\mathrm{T}} z)\,(PAP - \lambda_* I) z_*$.

- Case $b_0^{\mathrm{T}} z_* = 0$: We have $g_0^{\mathrm{T}} w_* = 0$ and z_* is an eigenvector of PAP corresponding to its eigenvalue λ_*. By Lemma 3.2.10, $y_* = S_1^{\mathrm{T}} z_*$ is an eigenvector of H corresponding to its eigenvalue λ_*. Let $s = -(H - \lambda_* I)^\dagger g$, according to Theorem 3.2.5, $\|s\| \leq \gamma$ and (λ_*, w_*) with $w_* = s + \sqrt{\gamma^2 - \|s\|^2}\,(y_* / \|y_*\|)$ solves pLGopt (3.25). By Theorem 3.2.4, (λ_*, u_*) with $u_* = S_1 w_*$ is a minimizer of LGopt (3.14). Now set

  $$x_* = S_1 s = -S_1 (H - \lambda_* I)^\dagger g = -(PAP - \lambda_* I)^\dagger b_0,$$

  and thus

  $$u_* = S_1 w_* = S_1 s + \sqrt{\gamma^2 - \|S_1 s\|^2}\, \frac{S_1 y_*}{\|S_1 y_*\|} = x_* + \sqrt{\gamma^2 - \|x_*\|^2}\, \frac{z_*}{\|z_*\|},$$

  as expected.

This completes the proof. □

We note that proving the equivalence between LGopt (3.14) and QEPmin (3.19) is of theoretical interest. The proof in [17] is incomplete since in Remark 3.2.1 we mentioned that they did not prove that pLGopt (3.25) and pQEPmin (3.28) are equivalent. Here we provided a complete proof in Theorem 3.2.8.

Returning to the original CRQopt (3.1), we observe that if (λ_*, u_*) solves LGopt (3.14), then $n_0 + u_*$ solves CRQopt (3.1). Therefore immediately we obtain the following theorem.

Theorem 3.2.9. *Suppose* (λ_*, z_*) *is a minimizer of* QEPmin (3.19). *Then a minimizer* v_* *of* CRQopt (3.1) *is given by*

$$v_* = \begin{cases} n_0 - (\gamma^2/b_0^{\mathrm{T}} z_*)\,(PAP - \lambda_* I) z_*, & \text{if } b_0^{\mathrm{T}} z_* \neq 0, \\ n_0 + x_* + \sqrt{\gamma^2 - \|x_*\|^2}\,(z_*/\|z_*\|), & \text{if } b_0^{\mathrm{T}} z_* = 0, \end{cases}$$

where $x_* = -(PAP - \lambda_* I)^{\dagger} b_0$ *in the case of* $b_0^{\mathrm{T}} z_* = 0$ *and it is guaranteed that* $\|x_*\| \leq \gamma$.

What the next theorem says is that solving QEPmin (3.19) is equivalent to calculating the leftmost eigenvalue of QEP (3.19b) among those having eigenvectors[3] in $\mathcal{N}(C^{\mathrm{T}})$. This result paves the way for the use of a Krylov subspace method to calculate the minimizer of QEPmin (3.19) in Section 3.3 ahead.

Theorem 3.2.10. *If* (λ_*, z_*) *is a minimizer of* QEPmin (3.19), *then* λ_* *is the leftmost eigenvalue of* QEP (3.19b) *among those having eigenvectors in* $\mathcal{N}(C^{\mathrm{T}})$.

Proof. Following the argument in the proof of Theorem 3.2.4, we find that the set of eigenvalues of QEP (3.19b) that have eigenvectors in $\in \mathcal{N}(C^{\mathrm{T}})$ and the set of eigenvalues of QEP (3.28b) are the same. The conclusion is an immediate consequence of Theorems 3.2.4 and 3.2.7. □

3.2.8 Summary

Starting with CRQopt (3.1), we have introduced five equivalent optimization problems. Figure 3.1 summarizes the relationships of these problems. The edge "$\longleftrightarrow$" in Figure 3.1 connecting two optimization problems indicates that we have an equivalent relationship in the previous subsections. We note that CRQopt (3.1) and CQopt (3.6) share the same minimizers v_*, while correspondingly the minimizer for LGopt (3.14) is $u_* = Pv_*$. Slightly more efforts are needed to describe

[3] This does not exclude the possibility that they may have eigenvectors not in $\mathcal{N}(C^{\mathrm{T}})$.

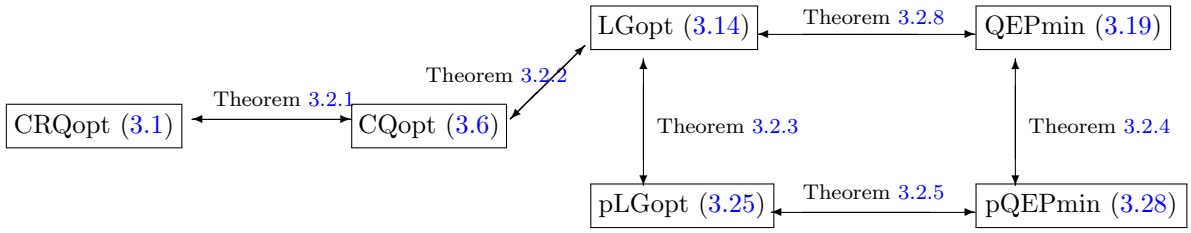

Figure 3.1: Equivalence of optimization problems

corresponding minimizers for other equivalent optimization problems as shown in Figure 3.1. The optimal values for the objective functions of LGopt (3.14), pLGopt (3.25), QEPmin (3.19), and pQEPmin (3.28) are all the same. The proof of Theorem 3.2.8 relies on Theorems 3.2.3, 3.2.4, and 3.2.5.

3.2.9 Easy and hard cases

Motivated by the treatments of the trust-region subproblem [45, 70], QEPmin (3.19) can be classified into two categories: the *easy* case and the *hard* case, defined as follows.

Definition 3.2.1. QEPmin (3.19) is in the *hard* case if it has a minimizer (λ_*, z_*) with $b_0^{\mathrm{T}} z_* = 0$. Otherwise, QEPmin (3.19) is in the *easy* case. Furthermore, any one of the equivalent optimization problems as shown in Figure 3.1 is said to be in the *hard* or *easy* case if the corresponding QEPmin is.

This notion of hardness and easiness exists has its historical reason in dealing with the trust-region subproblem. The hard case is not really hard as its name suggests when it comes to numerical computation. It is just a degenerate and rare case that needs special attention. The easy case is a generic one. Consider the hard case, let $\mathcal{V}$ be the maximal eigenspace of PAP corresponding to eigenvalue λ_*, then $b_0 \perp \mathcal{V}$ by Theorem 3.2.11. This creates difficulties to our later Lanczos method to solve QEPmin (3.19) in that the Krylov subspace $\mathcal{K}_k(PAP, b_0) \subset \mathcal{V}^{\perp}$ for any k. So in theory there is no vector in $\mathcal{K}_k(PAP, b_0)$ can approximate any eigenvector $z \in \mathcal{V}$ well.

In Theorems 3.2.11 and 3.2.12 below, we present a number of characterizations about the *hard* case.

Lemma 3.2.12. QEPmin (3.19) *is in the* hard *case if and only if* pQEPmin (3.28) *has a minimizer* (λ_*, w_*) *satisfying* $g_0^{\mathrm{T}} w_* = 0$.

Proof. To see this, we let (λ_*, z_*) be a minimizer QEPmin (3.19) satisfying $b_0^{\mathrm{T}} z_* = 0$. By Theorem 3.2.4, we know that z_* and w_* are related by $z_* = S_1 w_*$. Since also $b_0 = S_1 g_0$, $b_0^{\mathrm{T}} z_* = g_0^{\mathrm{T}} w_*$. □

Theorem 3.2.11. *Suppose that* QEPmin (3.19) *is in the* hard *case, and let* (λ_*, z_*) *be a minimizer such that* $b_0^{\mathrm{T}} z_* = 0$. *Then we have the following statements:*

(1) $\lambda_* = \lambda_{\min}(H)$, *the smallest eigenvalue of* H;

(2) $g_0 \perp \mathcal{U}$, *where* $\mathcal{U}$ *is the eigenspace of* H *associated with its eigenvalue* $\lambda_{\min}(H)$;

(3) $b_0 \perp \mathcal{V}$, *where* $\mathcal{V}$ *is the eigenspace of* PAP *associated with its eigenvalue* $\lambda_{\min}(H) \in \mathrm{eig}(PAP)$.

Proof. By Lemma 3.2.12, pQEPmin (3.28) has a minimizer (λ_*, w_*) satisfying $g_0^{\mathrm{T}} w_* = 0$. Theorem 3.2.6 immediately leads to item (1). Item (2) is a corollary of Lemma 3.2.4.

For item (3), it follows from Lemma 3.2.3 that if $\lambda_{\min}(H) \neq 0$, then $\mathcal{V} = S_1 \mathcal{U}$. Since $b_0 = S_1 g_0$ and $g_0 \perp \mathcal{U}$ by item (2), we conclude that $b_0 \perp S_1 \mathcal{U}$. If, however, $\lambda_{\min}(H) = 0$, then $\mathcal{V} = S_1 \mathcal{U} + \mathcal{R}(S_2)$. Since again $g_0 \perp \mathcal{U}$ by item (2) and also $b_0 \perp \mathcal{R}(S_2)$, we still have $b_0 \perp \mathcal{V}$. □

Theorem 3.2.12. QEPmin (3.19) *is in the* hard *case if and only if*

$$g_0 \perp \mathcal{U} \text{ and } \|[H - \lambda_{\min}(H) I]^{\dagger} g_0\|_2 \leq \gamma, \tag{3.40}$$

where $\mathcal{U}$ *is as defined in* Theorem 3.2.11.

Proof. If QEPmin (3.19) is in the *hard* case, then its optimal value (which is also the one of LGopt (3.14)) $\lambda_* = \lambda_{\min}(H)$. This can only happen when (3.40) holds. On the other hand, if (3.40) holds, then $\lambda_* = \lambda_{\min}(H)$ by Lemma 3.2.4. By Theorem 3.2.5, pQEPmin (3.28) has a minimizer (λ_*, w_*), where $H w_* = \lambda_* w_*$. Thus $g_0^{\mathrm{T}} w_* = 0$ because $g_0 \perp \mathcal{U}$ and $w_* \in \mathcal{U}$. Hence QEPmin (3.19) is in the *hard* case by Lemma 3.2.12. □

When QEPmin (3.19) is in the easy case, the situation is much simpler to characterize.

Theorem 3.2.13. CRQopt (3.1) *has a unique minimizer when* QEPmin (3.19) *is in the easy case.*

Proof. Suppose that QEPmin (3.19) is in the easy case. By Definition 3.2.1, all minimizers (λ_*, w_*) of pQEPmin (3.28) satisfy $g_0^{\mathrm{T}} w_* \neq 0$. Theorem 3.2.6 guarantees that pLGopt (3.25) has a unique minimizer. Consequently, the minimizer of LGopt (3.14) is unique by Theorem 3.2.3 and so is the minimizer of CRQopt (3.1). □

We use the remaining part of this subsection to explain how CRQopt (3.1) and the well-known trust-region subproblem (TRS) are related.

We have already proved in Theorem 3.2.1 that CRQopt (3.1) is equivalent to CQopt (3.6). Set $u = Pv$. Solving CQopt (3.6) is equivalent to solving

$$\begin{cases} \min \ u^{\mathrm{T}} PAPu + 2u^{\mathrm{T}} b_0, & (3.41a) \\ \text{s.t. } \|u\| = \gamma, & (3.41b) \\ \quad u \in \mathcal{N}(C^{\mathrm{T}}). & (3.41c) \end{cases}$$

Let H and g_0 be defined in (3.23) and S_1 be defined in (3.20). Then u is a minimizer of optimization problem (3.41) if and only if $y = S_1^{\mathrm{T}} u$ is a minimizer of the following equality constrained optimization problem

$$\begin{cases} \min \ y^{\mathrm{T}} H y + 2y^{\mathrm{T}} g_0, & (3.42a) \\ \text{s.t. } \|y\| = \gamma. & (3.42b) \end{cases}$$

The Lagrange equations for (3.42) is exactly the same as pLGopt (3.25). The problem (3.42) is similar to TRS

$$\begin{cases} \min \ y^{\mathrm{T}} H y + 2y^{\mathrm{T}} g_0, & (3.43a) \\ \text{s.t. } \|y\| \leq \gamma, & (3.43b) \end{cases}$$

except that its constraint is an equality instead of an inequality. When H is not positive definite, solution of (3.42) and TRS (3.43) are exactly the same. But when H is positive definite, we need to check whether $\|H^{-1} g_0\| < \gamma$. If so, $H^{-1} g_0$, instead of the minimizer of (3.42), is the minimizer of TRS (3.43). If, however, $\|H^{-1} g_0\| \geq \gamma$, then the minimizer of TRS (3.43) is the same as that of (3.42).

Lemma 2.1 in [28] shows that y is the (3.43) of (3.42) if and only if there exists $\widehat{\lambda} \in \mathbb{R}$ such that $(\widehat{\lambda}, y)$ satisfies the constraints of pLGopt (3.25) and $H - \widehat{\lambda} I$ is positive semi-definite. According to Lemma 3.2.4, the optimal value of pLGopt (3.25) satisfies $\lambda_* \leq \lambda_{\min}(H)$, which indicates that $H - \lambda_* I$ is positive semi-definite. Therefore, solving the equality constrained problem (3.42) is equivalent to solving pLGopt (3.25).

As we have mentioned, the terms "*easy*" and "*hard*" were adopted from the treatments of the trust-region subproblem [45, 70], where the term "easy" means the associated case is easy

to explain, not implying the case is easy to solve, however. A more detailed connection with TRS (3.43) is as follows.

1. In the easy case of QEPmin (3.19), $b_0^{\mathrm{T}} z_* \neq 0$ for all mimimizers (λ_*, z_*). By Theorem 3.2.4, $z_* = S_1 w_*$ for some $w_* \in \mathbb{R}^{n-m}$ and thus $g_0^{\mathrm{T}} w_* = b_0^{\mathrm{T}} S_1 w_* = b_0^{\mathrm{T}} z_* \neq 0$. By Theorem 3.2.6, $\lambda_* < \lambda_{\min}(H)$, and thus (λ_*, y_*) with $y_* = (H - \lambda_* I)^{-1} g_0$ is the unique minimizer of pLGopt (3.25). Hence y_* is the unique minimizer of (3.42), which is related to the easy case of TRS (3.43).

2. In the hard case of QEPmin (3.19), there exists a minimizer (λ_*, z_*) such that $b_0^{\mathrm{T}} z_* = 0$. Again by Theorem 3.2.4, $z_* = S_1 w_*$ for some $w_* \in \mathbb{R}^{n-m}$ and $g_0^{\mathrm{T}} w_* = 0$. By Theorem 3.2.5, a minimizer of pLGopt (3.25) is given by

$$(\lambda_*, y_*) \quad \text{with } y_* = x_* + \sqrt{\gamma^2 - \|x_*\|^2}\, \frac{w_*}{\|w_*\|},$$

where $x_* = -(H - \lambda_* I)^{\dagger} g_0$ and it is guaranteed that $\|x_*\| \leq \gamma$. Therefore, in general a minimizer of (3.42) can be expressed by $y_* = x_* + \sqrt{\gamma^2 - \|x_*\|^2}\,(w_*/\|w_*\|)$, which is related to hard case of TRS (3.43).

It is known that the generalized Lanczos method does not work for TRS (3.43) in the hard case [70, Theorem 4.6]. A restarting strategy was proposed to overcome the difficulty, but it was commented that the strategy computationally is very expensive for large scale problems [25, Theorem 5.8].

In the next section, we present that the Lanczos algorithms for CRQopt (3.1), which resemble the generalized Lanczos method for TRS and are suitable for handling the easy case. However, with some additional effort, the hard case can be detected. In the rest of this dissertation, we mostly focus only on the easy case.

3.3 Lanczos algorithm

As was shown in Section 3.2, solving CRQopt (3.1) is equivalent to solving LGopt (3.14) or QEPmin (3.19). In this section we present algorithms to solve CRQopt (3.1) by solving LGopt (3.14) and QEPmin (3.19). We first review the Lanczos procedure in section 2.1, then we apply the procedure to reduce LGopt (3.14) and QEPmin (3.19), and finally solve the reduced LGopt

and QEPmin to yield approximations to the minimizer of the original CRQopt (3.1). Besides, we prove the finite step stopping property of the proposed algorithms and comment on how to detect the hard case.

3.3.1 Solving LGopt

In this subsection, we first use (2.3) obtained by the Lanczos process with $M = PAP$ to reduce LGopt (3.14), and then solve the reduced LGopt via an approach based on a secular equation solver.

For the dimensional reduction of LGopt (3.14), we restate the Lagrange equations (3.14b) and (3.14b) here

$$(PAP - \lambda I)u = -b_0, \quad \|u\| = \gamma, \quad Pu = u, \tag{3.44}$$

where we include the constraint $Pu = u$ since we are only interested in those vectors $u \in \mathcal{N}(C^{\mathrm{T}})$.

Apply the Lanczos process with $M = PAP$ and the starting vector $r_0 = b_0$ to get (2.3) with $M = PAP$. It then follows that for any scalar λ

$$Q_k^{\mathrm{T}}(PAP - \lambda I)Q_k = T_k - \lambda I \quad \text{and} \quad Q_k^{\mathrm{T}} b_0 = \|b_0\| e_1.$$

Consequently, we arrive at the reduced LGopt (3.14)

$$\text{rLGopt:} \quad \begin{cases} \min \ \lambda & \text{(3.45a)} \\ \text{s.t. } (T_k - \lambda I)x = -\|b_0\| e_1, & \text{(3.45b)} \\ \quad \|x\| = \gamma. & \text{(3.45c)} \end{cases}$$

A couple of comments are for the efficiency of the Lanczos process with $M = PAP$. In the process, we have to calculate matrix-vector products $Mx = P(A(Pq_j))$ efficiently. For that purpose, it suffices for us to be able to calculate the product Pc efficiently for any given $c \in \mathbb{R}^n$. In fact

$$Pc = q_j - CC^{\dagger}c = q_j - Cy,$$

where $y = C^{\dagger}c$ is the minimum-norm solution of the least squares problem

$$y = \arg\min_{z \in \mathbb{R}^m} \|Cz - c\|_2, \tag{3.46}$$

which can be computed by using the QR decomposition of $C \in \mathbb{R}^{n\times m}$ or an iterative method such as LSQR [15, 48, 57]. Another cost-saving observation due to [21] is that for the matrix-vector product $Mq_j = P(A(Pq_j))$, the first application of P in Pq_j can be skipped due to the fact that if the initial vector $b_0 \in \mathcal{N}(C^{\mathrm{T}})$, then $Pq_j = q_j$ for all $1 \le j \le k+1$.

We end this subsection by pointing out rLGopt (3.45) cannot fall into the hard case. The same phenomenon happens to the tridiagonal TRS generated by the generalized Lanczos method [25, Theorem 5.3] as well. Let the eigen-decomposition of T_k be

$$T_k = Y\Theta Y^{\mathrm{T}},\ Y^{\mathrm{T}}Y = I_k,\ \Theta = \mathrm{diag}(\vartheta_1, \vartheta_2, \ldots, \vartheta_k), \tag{3.47}$$

where we suppress the dependency of Y, Θ, and ϑ_j on k for notational convenience. Further, we arrange ϑ_j in nondecreasing order, i.e., $\vartheta_1 \le \vartheta_2 \le \cdots \le \vartheta_k$ and $Y = [y_1, y_2, \cdots, y_k]$. Let $\mu^{(k)}$ be the optimal value of rLGopt (3.45).

Theorem 3.3.1. *Suppose that $\beta_j \neq 0$ for $j = 2, 3, \ldots, k$ in the Lanczos process. Then $\mu^{(k)} < \vartheta_1 \equiv \lambda_{\min}(T_k)$, and* rLGopt *cannot fall into the hard case.*

Proof. It is well-known that the first components of all eigenvectors y_i of irreducible T_k are nonzero [51, p.140]. In particular, $e_1^{\mathrm{T}} y_1 \neq 0$. Lemma 3.2.4 immediately leads to $\mu^{(k)} < \vartheta_1$.

Since $\mu^{(k)} < \lambda_{\min}(T_k)$ by Theorem 3.2.11(1), we conclude that rLGopt cannot fall into the hard case. □

Now we explain how to solve rLGopt (3.45). Suppose that $\beta_j \neq 0$ for $j = 2, 3, \ldots, k$, and let the eigen-decomposition of T_k be given by (3.47).

Theorem 3.3.2. *The optimal value $\mu^{(k)}$ of* rLGopt (3.45) *is the smallest root of the secular function*

$$\widehat{\chi}(\lambda) = \|b_0\|^2 e_1^{\mathrm{T}}(T_k - \lambda I)^{-2} e_1 - \gamma^2 = \sum_{i=1}^{k} \frac{\zeta_i^2}{(\lambda - \vartheta_i)^2} - \gamma^2, \tag{3.48}$$

where $\zeta_i = \|b_0\| e_1^{\mathrm{T}} y_i$ for $i = 1, 2, \cdots, k$. Furthermore,

$$(\mu^{(k)}, x^{(k)}) = (\mu^{(k)},\ -\|b_0\|(T_k - \mu^{(k)} I)^{-1} e_1) \tag{3.49}$$

is a minimizer of rLGopt (3.45).

Proof. rLGopt (3.45) takes the same form as pLGopt (3.25). By Theorem 3.3.1, $\mu^{(k)} < \lambda_{\min}(T_k)$. The conclusions of the lemma are now consequences of Lemma 3.2.4. □

Theorem 3.3.2 naturally leads to a method for solving rLGopt (3.45) through calculating the smallest root of the secular function $\widehat{\chi}(\lambda)$. Algorithm 3.45 outlines the method, based on an efficient secular equation solver in Section 2.2.2.

Algorithm 1 Solving rLGopt (3.45)

Input: $T_k \in \mathbb{R}^{k\times k}$, $\|b_0\|$, $\gamma > 0$, and tolerance ϵ;

Output: $(\mu^{(k)}, x^{(k)})$, approximate minimizer of rLGopt (3.45);

Compute the eigenvalues $\theta_1 \le \theta_2 \le \cdots \le \theta_k$ of T_k and the corresponding eigenvectors $y_1, \cdots, y_k$;
$\xi_i \leftarrow \|b_0\| e_1^{\mathrm{T}} y_i$ for $i = 1, 2, \cdots, k$;
$\delta_0 \leftarrow \frac{1}{\gamma}\sqrt{\sum_{i=1}^{k} \xi_i^2}$, $\alpha^{(0)} \leftarrow \theta_1 - \delta_0$, $\beta^{(0)} \leftarrow \theta_1$ and $\eta \leftarrow \gamma^2 - \sum_{i=2}^{k} \frac{\xi_i^2}{([\theta_1 - \delta_0] - \theta_i)^2}$;
if $\eta > 0$ **then** $\lambda^{(0)} \leftarrow \theta_1 - |\xi_1|/\sqrt{\eta}$ **else** $\lambda^{(0)} \leftarrow \theta_1 - \delta_0/2$;
for $j = 0, 1, 2, \ldots$ **do**
    $\chi \leftarrow \sum_{i=1}^{k} \frac{\xi_i^2}{(\lambda^{(j)} - \theta_i)^2} - \gamma^2$;
    if $\chi > 0$ **then** $\alpha^{(j+1)} \leftarrow \alpha^{(j)}$, $\beta^{(j+1)} \leftarrow \lambda^{(j)}$ **else** $\alpha^{(j+1)} \leftarrow \lambda^{(j)}$, $\beta^{(j+1)} \leftarrow \beta^{(j)}$;
    $a \leftarrow (\lambda^{(j)} - \theta_1)^3 \sum_{i=1}^{n} \frac{\xi_i^2}{(\lambda^{(j)} - \theta_i)^3}$, $b \leftarrow (\lambda^{(j)} - \theta_1) \sum_{i=1}^{n} \frac{\xi_i^2}{(\lambda^{(j)} - \theta_i)^3} - \chi$;
    if $b > 0$ **then**
        $\lambda_1 \leftarrow \theta_1 - \sqrt{a/b}$;
        if $\lambda_1 \in (\alpha^{(j+1)}, \beta^{(j+1)})$ **then** $\lambda^{(j+1)} \leftarrow \lambda_1$ **else** $\lambda^{(j+1)} \leftarrow (\alpha^{(j+1)} + \beta^{(j+1)})/2$;
    else
        $\lambda^{(j+1)} \leftarrow (\alpha^{(j+1)} + \beta^{(j+1)})/2$;
    end if
    if $|\lambda^{(j+1)} - \lambda^{(j)}| < \epsilon$ **then stop**;
end for
return $(\mu^{(k)}, x^{(k)}) = (\lambda^{(j+1)}, -(T_k - \mu^{(k)} I)^{-1} \|b_0\| e_1)$ as a solution of rLGopt (3.45).

Although Theorem 3.3.2 assures us that the hard case cannot happen for rLGopt (3.45), cases where $|e_1^{\mathrm{T}} y_1|$ is very tiny are possible. Such a nearly hard case has to be treated with care, a subject of further future study.

Remark 3.3.1. Let us discuss the relationship between solving rLGopt (3.45) and solving TRS by a generalized Lanczos (GLTRS) method proposed in [25]. GLTRS projects a similar problem to (3.41a) and (3.41b) by a Krylov subspace to yield a small-size problem. Ignoring (3.41c) for

the moment, we run the Lanczos process with $M = PAP$ and the starting vector be $r_0 = b_0$ to generate the orthonormal basis matrix Q_k and the tridiagonal matrix T_k. Since $b_0 \in \mathcal{N}(C^{\mathrm{T}})$, it can be verified that $\mathcal{R}(Q_k) \subset \mathcal{N}(C^{\mathrm{T}})$, which means that (3.41c) is automatically taken care of. Project (3.41a) and (3.41b) onto the column space of Q_k and we arrive at the following equality constrained optimization problem:

$$\begin{cases} \min \ x^{\mathrm{T}} T_k x + 2x^{\mathrm{T}} \|g_0\| e_1, & \text{(3.50a)} \\ \text{s.t. } \|x\| = \gamma. & \text{(3.50b)} \end{cases}$$

Problem (3.50) is similar to the tridiagonal TRS generated by GLTRS except that the constraint here is equality instead of inequality. Solving (3.50) by the method of the Lagrangian multipliers leads to exactly rLGopt (3.45). □

After computing $(\mu^{(k)}, x^{(k)})$, the minimizer of rLGopt (3.45), we deduce an approximate minimizer of LGopt (3.14):

$$(\mu^{(k)}, u^{(k)}) = (\mu^{(k)}, Q_k x^{(k)}) \tag{3.51}$$

It can be verified that

$$\|u^{(k)}\| = \|x^{(k)}\| = \gamma, \ u^{(k)} \in \mathcal{R}(Q_k) \subset \mathcal{N}(C^{\mathrm{T}}). \tag{3.52}$$

That is the pair in (3.51) satisfies the constraints (3.14c) and (3.14d).

The accuracy of this approximate minimizer $(\mu^{(k)}, u^{(k)})$ can be measured by the residual vector

$$r_k^{\mathrm{LGopt}} = (PAP - \mu^{(k)} I) u^{(k)} + b_0. \tag{3.53}$$

For simplicity, we may assume that $(\mu^{(k)}, x^{(k)})$ satisfies the constraint of rLGopt (3.45) exactly, in particular $(T_k - \mu^{(k)} I) x^{(k)} = -\|b_0\| e_1$, since it is reasonable to assume that the error in $(\mu^{(k)}, u^{(k)})$ as an approximate minimizer of LGopt (3.14) is much larger than the error in $(\mu^{(k)}, x^{(k)})$ as the computed minimizer of rLGopt (3.45). Subsequently, we have the following expression for the residual vector r_k^{LGopt}, similar to the one on the generalized Lanczos method for TRS [25].

Proposition 3.3.1. *Suppose that the approximate minimizer* $(\mu^{(k)}, x^{(k)})$ *of* rLGopt (3.45) *satisfies the constraints of* rLGopt (3.45) *exactly. We have*

$$r_k^{\mathrm{LGopt}} = \beta_{k+1} q_{k+1} e_k^{\mathrm{T}} x^{(k)}. \tag{3.54}$$

Proof. We have by (2.3)

$$\begin{aligned} r_k^{\mathrm{LGopt}} &= (PAP - \mu^{(k)} I) Q_k x^{(k)} + b_0 \\ &= [Q_k (T_k - \mu^{(k)} I) + \beta_{k+1} q_{k+1} e_k^{\mathrm{T}}] x^{(k)} + b_0 \\ &= -Q_k \|b_0\| e_1 + \beta_{k+1} q_{k+1} e_k^{\mathrm{T}} x^{(k)} + b_0 \\ &= \beta_{k+1} q_{k+1} e_k^{\mathrm{T}} x^{(k)}, \end{aligned}$$

as was to be shown. □

In deciding if r_k^{LGopt} is sufficiently small, a sensible way is to check some kind of normalized residual. In view of (3.53), a reasonable one is

$$\mathrm{NRes}_k^{\mathrm{LGopt}} := \frac{\|r_k^{\mathrm{LGopt}}\|}{(\|A\| + |\mu^{(k)}|)\|x^{(k)}\| + \|b_0\|} = \frac{|\beta_{k+1}|\,|e_k^{\mathrm{T}} x^{(k)}|}{(\|A\| + |\mu^{(k)}|)\|x^{(k)}\| + \|b_0\|} =: \delta_k^{\mathrm{LGopt}}. \tag{3.55}$$

The Lanczos process is stopped if $\delta_k^{\mathrm{LGopt}} \le \epsilon$, a prescribed tolerance. In summary, the Lanczos algorithm for solving LGopt (3.14) is given in Algorithm 2.

Algorithm 2 Solving LGopt (3.14)

Input: $A \in \mathbb{R}^{n\times n}$, $C \in \mathbb{R}^{n\times m}$, $b_0 \in \mathbb{R}^n$, $\gamma > 0$, and tolerance ϵ;

Output: $(\mu^{(k)}, u^{(k)})$, approximate minimizer of LGopt (3.14);

$\beta_1 \leftarrow \|b_0\|$;
if $\beta_1 = 0$ **then stop**;
$q_1 \leftarrow r_0/\beta_1$, $q_0 \leftarrow 0$;
for $k = 1, 2, \ldots$ **do**
    $\hat{q} \leftarrow Aq_k$, $\hat{q} \leftarrow P\hat{q}$, $\hat{q} \leftarrow \hat{q} - \beta_k q_{k-1}$;
    $\alpha_k \leftarrow q_k^{\mathrm{T}} \hat{q}$, $\hat{q} \leftarrow \hat{q} - \alpha_k q_k$, $\beta_{k+1} \leftarrow \|\hat{q}\|$;
    compute the minimizer $(\mu^{(k)}, x^{(k)})$ of rLGopt (3.45) by Algorithm 1;
    if $\delta_k^{\mathrm{LGopt}} \le \epsilon$ **then stop**;
    $q_{k+1} \leftarrow \hat{q}/\beta_{k+1}$;
end for
$Q_k = [q_1, q_2, \cdots, q_k]$;
return $(\mu^{(k)}, u^{(k)})$ with $u^{(k)} = Q_k x^{(k)}$ as an approximate minimizer of LGopt (3.14).

3.3.2 Solving QEPmin

In this section, we propose our Lanczos algorithm for the numerical solution of QEPmin (3.19). It follows the same idea as the previous subsection. First, we reduce QEPmin (3.19) to a smaller problem by projection, and then solve the reduced QEPmin by an eigensolver. One immediate advantage of doing so is the availability of mature eigensolvers for use to solve the underlying QEP. Independently, QEPmin (3.19) is of interest of its own, e.g., it plays a role in solving the total least square problems [35, 59].

The Lanczos process is natural as a method to solve QEP (3.19b) for its leftmost eigenvalue and the corresponding eigenvector. For convenience, we restate QEP (3.19b) here:

$$(PAP - \lambda I)^2 z = \gamma^{-2} b_0 b_0^{\mathrm{T}} z, \quad Pz = z. \tag{3.56}$$

Note that we have added the constraint $Pz = z$ since we are only interested in those eigenvectors $z \in \mathcal{N}(C^{\mathrm{T}})$.

Now we discuss how to perform the dimensional reduction of the QEP (3.56) via the projection onto the Krylov subspace generated by the Lanczos process described in Section 2.1. Let Q_k be the orthogonal matrix and T_k be the tridiagonal matrix generated by k steps of the Lanczos process with the matrix $M = PAP$ and the starting vector b_0. We will again have (2.3), i.e.,

$$PAPQ_k = Q_k T_k + \beta_{k+1} q_{k+1} e_k^{\mathrm{T}} \quad \text{and} \quad Q_k^{\mathrm{T}} b_0 b_0^{\mathrm{T}} Q_k = \|b_0\|^2 e_1 e_1^{\mathrm{T}}. \tag{3.57}$$

By a straightforward calculation, we have

$$\begin{aligned}
(PAP - \lambda I)^2 Q_k &= (PAP - \lambda I)\big[Q_k(T_k - \lambda I) + \beta_{k+1} q_{k+1} e_k^{\mathrm{T}}\big] \\
&= \big[Q_k(T_k - \lambda I) + \beta_{k+1} q_{k+1} e_k^{\mathrm{T}}\big](T_k - \lambda I) + (PAP - \lambda I)\beta_{k+1} q_{k+1} e_k^{\mathrm{T}} \\
&= Q_k(T_k - \lambda I)^2 + \beta_{k+1} q_{k+1} e_k^{\mathrm{T}}(T_k - \lambda I) + \beta_{k+1}(PAP - \lambda I) q_{k+1} e_k^{\mathrm{T}}
\end{aligned} \tag{3.58}$$

and

$$\begin{aligned}
Q_k^{\mathrm{T}}(PAP - \lambda I)^2 Q_k &= (T_k - \lambda I)^2 + 0 + \beta_{k+1} Q_k^{\mathrm{T}}(PAP - \lambda I) q_{k+1} e_k^{\mathrm{T}} \\
&= (T_k - \lambda I)^2 + \beta_{k+1}\big[Q_k(T_k - \lambda I) + \beta_{k+1} q_{k+1} e_k^{\mathrm{T}}\big]^{\mathrm{T}} q_{k+1} e_k^{\mathrm{T}} \\
&= (T_k - \lambda I)^2 + \beta_{k+1}^2 e_k e_k^{\mathrm{T}}.
\end{aligned} \tag{3.59}$$

By (3.57) and (3.59), naturally one would like to take the reduced QEP (3.56) to be

$$\left[(T_k - \lambda I)^2 + \beta_{k+1}^2 e_k e_k^{\mathrm{T}}\right] w = \gamma^{-2}\|b_0\|^2 e_1 e_1^{\mathrm{T}} w. \tag{3.60}$$

Unfortunately, this reduced QEP may not have any real eigenvalue, not to mention that the leftmost eigenvalue is guaranteed to be real, as demonstrated by Example 3.3.1 below. To overcome it, we propose to drop the term $\beta_{k+1}^2 e_k e_k^{\mathrm{T}}$ in (3.59) and use the following reduced QEP

$$(T_k - \lambda I)^2 w = \gamma^{-2}\|b_0\|^2 e_1 e_1^{\mathrm{T}} w. \tag{3.61}$$

Since it has the same form as the QEP in pQEPmin (3.28b), the leftmost eigenvalue of the reduced QEP (3.61) is guaranteed to be real by Theorem 3.2.7.

It can be seen that the corresponding reduced QEPmin (3.19) to QEP (3.61) is given by

$$\text{rQEPmin:} \quad \begin{cases} \min\ \lambda & \text{(3.62a)} \\ \text{s.t. } (T_k - \lambda I)^2 w = \gamma^{-2}\|b_0\|^2 e_1 e_1^{\mathrm{T}} w, & \text{(3.62b)} \\ \qquad \lambda \in \mathbb{R}, w \neq 0. & \text{(3.62c)} \end{cases}$$

We note that the Lanczos process of PAP on b_0 is the same as, upon a linear transformation by S_1^{T}, that of H on g_0 in pQEPmin (3.28). Therefore, rQEPmin (3.62) can be viewed as a reduced-form of pQEPmin (3.28).

Example 3.3.1. Let $A = \mathrm{diag}(1, 2, 3, 4, 5)$, $C = [0.65, 1, 0.68, 1.13, -0.23]^{\mathrm{T}}$ and $b = [1]$. The eigenvalues of QEP (3.19b) and (3.19c) in QEPmin, computed by MATLAB, are

$$\texttt{0.8333},\ \texttt{1.6493},\ \texttt{2.0000},\ \texttt{2.9916} \pm \texttt{0.2369i},\ \texttt{3.8786},\ \texttt{4.8236},\ \texttt{5.1196}.$$

We see the leftmost eigenvalue $\texttt{0.8333} \in \mathbb{R}$. Apply the Lanczos process with $k = 2$ leads to a 2×2 QEP (3.60) whose eigenvalues are computed to be

$$\texttt{1.8124} \pm \texttt{0.4172i},\ \texttt{3.3714} \pm \texttt{0.2547i},$$

both are genuine complex numbers! In contrast, the eigenvalues of QEP (3.61) are

$$\texttt{1.1429},\ \texttt{2.2661},\ \texttt{2.8915},\ \texttt{4.0672},$$

all of which are real.

To solve rQEPmin (3.61), we first linearize it into a linear eigenvalue problem (LEP). The reader is referred to [18, Chapter 1] for many different ways to linearize a general polynomial eigenvalue problem. Our rQEPmin (3.61) takes a rather particular form, and we use similar ideas but slightly different linearization. Specifically, we let $y = (T_k - \lambda I)w$ and $s = \begin{bmatrix} y \\ w \end{bmatrix}$. Then QEP (3.62b) can be converted to the following LEP:

$$\begin{bmatrix} T_k & -\gamma^{-2}\|b_0\|^2 e_1 e_1^{\mathrm{T}} \\ -I & T_k \end{bmatrix} s = \lambda s. \tag{3.63}$$

At this point, one can use a standard eigensolver to find the leftmost real eigenvalue $\mu^{(k)}$ of LEP (3.63) and its corresponding eigenvector $s^{(k)} = \begin{bmatrix} y^{(k)} \\ w^{(k)} \end{bmatrix}$. Subsequently, an approximate optimizer of rQEPmin (3.62) is given by $(\mu^{(k)}, w^{(k)})$.

The minimizer $(\mu^{(k)}, w^{(k)})$ of rQEPmin (3.62) yields an approximate minimizer of QEPmin (3.19) as

$$(\mu^{(k)}, z^{(k)}) = (\mu^{(k)}, Q_k w^{(k)}). \tag{3.64}$$

The accuracy of this pair $(\mu^{(k)}, z^{(k)})$ as an approximate minimizer can be measured by the norm of the following the residual vector

$$r_k^{\text{QEPmin}} = \left(PAP - \mu^{(k)} I\right)^2 z^{(k)} - \gamma^{-2} b_0 b_0^{\mathrm{T}} z^{(k)}. \tag{3.65}$$

The following proposition shows that this residual vector can be efficiently obtained during computation.

Proposition 3.3.2. *Suppose that* $(\mu^{(k)}, w^{(k)})$ *is an exact minimizer of* rQEPmin (3.62) *and* $y^{(k)} = (T_k - \mu^{(k)} I)w^{(k)}$. *Then*

$$r_k^{\text{QEPmin}} = \beta_{k+1} q_{k+1} e_k^{\mathrm{T}} y^{(k)} + \beta_{k+1}(PAP - \mu^{(k)} I) q_{k+1} e_k^{\mathrm{T}} w^{(k)}. \tag{3.66}$$

Proof. Keeping (3.58) in mind, we find that

$$\begin{aligned} r_k^{\text{QEPmin}} &= \left(PAP - \mu^{(k)} I\right)^2 Q_k w^{(k)} - \gamma^{-2} b_0 b_0^{\mathrm{T}} Q_k w^{(k)} \\ &\overset{(3.58)}{=} Q_k (T_k - \mu^{(k)} I)^2 w^{(k)} + \beta_{k+1} q_{k+1} e_k^{\mathrm{T}} (T_k - \mu^{(k)} I) w^{(k)} \\ &\quad + \beta_{k+1} (PAP - \mu^{(k)} I) q_{k+1} e_k^{\mathrm{T}} w^{(k)} - Q_k \frac{\|b_0\|^2}{\gamma^2} e_1 e_1^{\mathrm{T}} w^{(k)} \\ &\overset{(3.62b)}{=} \beta_{k+1} q_{k+1} e_k^{\mathrm{T}} (T_k - \mu^{(k)} I) w^{(k)} + \beta_{k+1} (PAP - \mu^{(k)} I) q_{k+1} e_k^{\mathrm{T}} w^{(k)} \\ &= \beta_{k+1} q_{k+1} e_k^{\mathrm{T}} y^{(k)} + \beta_{k+1} (PAP - \mu^{(k)} I) q_{k+1} e_k^{\mathrm{T}} w^{(k)}, \end{aligned}$$

as expected. □

We note that if the $(k+1)$st step are carried out in the Lanczos process (2.3), then the term $(PAP - \mu^{(k)} I) q_{k+1}$ in (3.66) can be expressed as a linear combination of q_k, q_{k+1}, and q_{k+2}. We propose to use the following normalized residual norm as a stopping criterion for the Lanczos process:

$$\text{NRes}_k^{\text{QEPmin}} := \frac{\|r_k^{\text{QEPmin}}\|}{\left[(\|A\| + |\mu^{(k)}|)^2 + \gamma^{-2}\|b_0\|^2\right]\|w^{(k)}\|_2} \tag{3.67a}$$

$$\leq \frac{|\beta_{k+1}|\left[|e_k^{\mathrm{T}} y^{(k)}| + (\|A\| + |\mu^{(k)}|)\,|e_k^{\mathrm{T}} w^{(k)}|\right]}{\left[(\|A\| + |\mu^{(k)}|)^2 + \gamma^{-2}\|b_0\|^2\right]\|w^{(k)}\|_2} =: \delta_k^{\text{QEPmin}}. \tag{3.67b}$$

The Lanczos algorithm for solving QEPmin (3.19) is summarized in Algorithm 3.

It remains to explain why $(\mu^{(k)}, u^{(k)})$ at Line 14 of Algorithm 3 is an approximated minimizer of LGopt (3.14). Let $(\mu^{(k)}, \begin{bmatrix} y^{(k)} \\ w^{(k)} \end{bmatrix})$ be the leftmost eigenpair of LEP (3.63). By Theorem 3.3.2, $\mu^{(k)} \notin \text{eig}(T_k)$, and so $(T_k - \mu^{(k)} I)^2 w^{(k)} \neq 0$ and $e_1^{\mathrm{T}} w^{(k)} \neq 0$. Through a straightforward application of Theorem 3.2.5 to rLGopt (3.45) and rQEPmin (3.62), we find that $(\mu^{(k)}, x^{(k)})$ is the minimizer of rLGopt (3.45) where

$$x^{(k)} = -\frac{\gamma^2}{\|b_0\| e_1^{\mathrm{T}} w^{(k)}} (T_k - \mu^{(k)} I) w^{(k)} = -\frac{\gamma^2}{\|b_0\| e_1^{\mathrm{T}} w^{(k)}} y^{(k)}. \tag{3.68}$$

Therefore, as a by-product, an approximate minimizer of LGopt (3.14) is given by

$$(\mu^{(k)}, u^{(k)}) = \left(\mu^{(k)}, \; -\frac{\gamma^2}{\|b_0\| e_1^{\mathrm{T}} w^{(k)}} Q_k y^{(k)}\right). \tag{3.69}$$

Algorithm 3 Solving QEPmin (3.19)

Input: $A \in \mathbb{R}^{n\times n}$, $C \in \mathbb{R}^{n\times m}$, $b_0 \in \mathbb{R}^n$, $\gamma > 0$, and tolerance ϵ;

Output: $(\mu^{(k)}, z^{(k)})$, approximate minimizer of QEPmin (3.19)

$\beta_1 \leftarrow \|b_0\|$;

if $\beta_1 = 0$ **then stop**;

$q_1 \leftarrow r_0/\beta_1$, $q_0 \leftarrow 0$;

for $k = 1, 2, \ldots$ **do**

  $\hat{q} \leftarrow Aq_k$, $\hat{q} \leftarrow P\hat{q}$, $\hat{q} \leftarrow \hat{q} - \beta_k q_{k-1}$;

  $\alpha_k \leftarrow q_k^{\mathrm{T}}\hat{q}$, $\hat{q} \leftarrow \hat{q} - \alpha_k q_k$, $\beta_{k+1} \leftarrow \|\hat{q}\|$;

  compute the leftmost eigenpair $(\mu^{(k)}, s)$ of LEP (3.63);

  $y^{(k)} \leftarrow s_{(1:k)}$, $w^{(k)} \leftarrow s_{(k+1:2k)}$;

  if $\delta_k^{\mathrm{QEPmin}} \le \epsilon$ **then stop**;

  $q_{k+1} \leftarrow \hat{q}/\beta_{k+1}$;

end for

$Q_k = [q_1, q_2, \cdots, q_k]$;

$z^{(k)} = Q_k w^{(k)}$ and $u^{(k)} = -\frac{\gamma^2}{\|b_0\| e_1^{\mathrm{T}} w^{(k)}} Q_k y^{(k)}$;

return $(\mu^{(k)}, z^{(k)})$ as an approximated minimizer of QEPmin (3.19) and, as a by-product, $(\mu^{(k)}, u^{(k)})$ as an approximated minimizer of LGopt (3.14).

3.3.3 Lanczos algorithm for CRQopt

Having obtained approximate minimizers of LGopt (3.14) and QEPmin (3.19), by Theorem 3.2.2 we can recover an approximate minimizer of CRQopt (3.1) as

$$v^{(k)} = n_0 + u^{(k)}. \tag{3.70}$$

where $u^{(k)}$ is given by (3.51) if via solving LGopt (3.14) or by (3.69) if via solving QEPmin (3.19). The overall algorithm called *the Lanczos Method*, is outlined in Algorithm 4.

3.3.4 Finite step stopping property

As in many Lanczos type methods for numerical linear algebra problems [10, 20, 51, 55], Algorithm 4 also enjoys a finite-step-stopping property in the exact arithmetic, i.e., it will deliver

Algorithm 4 Solving CRQopt (3.1)

Input: $A \in \mathbb{R}^{n\times n}$, $C \in \mathbb{R}^{n\times m}$ with full column rank, $b \in \mathbb{R}^m$, tolerance ϵ;

Output: approximate minimizer v of CRQopt (3.1);

1: $n_0 \leftarrow (C^{\mathrm{T}})^{\dagger} b$ (by, e.g., the QR decomposition of C);

2: **if** $\|n_0\| > 1$ **then output** ***no solution***;

3: **if** $\|n_0\| = 1$ **then** $v \leftarrow n_0$ **and output** v;

4: **if** $\|n_0\| < 1$ **then**

5: $\quad \gamma \leftarrow \sqrt{1 - \|n_0\|^2}$, $q \leftarrow An_0$, $b_0 \leftarrow (I - CC^{\dagger})q$;

6: $\quad$ compute an approximate solution of LGopt (3.14) $(\mu^{(k)}, u^{(k)})$ by Algorithm 2 or 3

7: $\quad$ **return** $v^{(k)} = n_0 + u^{(k)}$, approximate minimizer of CRQopt (3.1);

8: **end if**

an exact solution in at most n steps. It is an excellent theoretic property but of little or no practical significance for large scale problems. We often expect that the Lanczos process would stop much sooner before the nth step for otherwise the method would be deemed too expensive to be practical.

We will show the property using LGopt (3.14) as an example, which, for convenience, is restated here.

$$\text{LGopt:} \quad \begin{cases} \min\ \lambda & \text{(3.14a)} \\ \text{s.t. } (PAP - \lambda I)u = -b_0, & \text{(3.14b)} \\ \quad \|u\| = \gamma, & \text{(3.14c)} \\ \quad u \in \mathcal{N}(C^{\mathrm{T}}). & \text{(3.14d)} \end{cases}$$

Let (λ_*, u_*) be the minimizer of LGopt (3.14) and $k_{\max}$ be the smallest k such that $\beta_{k+1} = 0$ in the Lancozs process, namely the Lanczos process breaks down at step $k = k_{\max}$. We will prove that $\mu^{(k_{\max})} = \lambda_*$ and $u^{(k_{\max})} = u_*$.

We have already shown in (3.52) that the second and third constraints of LGopt (3.14) are satisfied by $u^{(k_{\max})}$. Besides, since $\beta_{k_{\max}+1} = 0$, $r_{k_{\max}}^{\mathrm{LGopt}} = 0$ by Proposition 3.3.1, i.e., the first constraint of LGopt (3.14) holds. It remains to show that $\mu^{(k_{\max})} = \lambda_*$.

Lemma 3.3.1. *$\mu^{(k_{\max})}$ is the smallest root of*

$$\widetilde{\chi}(\lambda) := g^{\mathrm{T}}[(H - \lambda I)^{\dagger}]^2 g^{\mathrm{T}} - \gamma^2. \tag{3.71}$$

In addition, if LGopt (3.14) *is in the easy case, then* $\mu^{(k_{\max})} = \lambda_*$, *where* (λ_*, z_*) *is the minimizer of* LGopt (3.14).

Proof. Let $\vartheta_1 \le \vartheta_2 \le \cdots \le \vartheta_{k_{\max}}$ be the eigenvalues of $T_{k_{\max}}$ and let $y_1, y_2, \cdots, y_{k_{\max}}$ be the corresponding orthonormal eigenvectors. Expand $\|b_0\|e_1 = \sum_{i=1}^{k_{\max}} \zeta_i y_i$ and define the secular function

$$\widehat{\chi}(\lambda) = \|b_0\|^2 e_1^{\mathrm{T}}(T_{k_{\max}} - \lambda I)^{-2} e_1 - \gamma^2 = \sum_{i=1}^{k_{\max}} \frac{\zeta_i^2}{(\lambda - \vartheta_i)^2} - \gamma^2. \tag{3.72}$$

By Theorem 3.3.2, $\mu^{(k_{\max})} < \vartheta_1$. Apply Lemma 3.2.7 with $H = T_{k_{\max}}$ and $g = \|b_0\|e_1$ to conclude that $\mu^{(k_{\max})}$ is a root of the secular function (3.72). Since $\widehat{\chi}(\lambda)$ is strictly increasing in $(-\infty, \mu^{(k_{\max})})$, $\mu^{(k_{\max})}$ is the smallest root of $\widehat{\chi}(\lambda)$.

Expand $Q_{k_{\max}}$ to form an the orthogonal matrix $\widehat{Q} := [Q_{k_{\max}},\ Q_\perp] \in \mathbb{R}^{n\times n}$ and let $T = \widehat{Q}^{\mathrm{T}} PAP \widehat{Q}$. Since the column space of $Q_{k_{\max}}$ is an invariant subspace of PAP, we have

$$T = \begin{bmatrix} T_{k_{\max}} & \\ & T_\perp \end{bmatrix}.$$

Let $S = [S_1, S_2]$ be defined in (3.20), and let $H = S_1^{\mathrm{T}} PAP S_1$ and $g_0 = S_1^{\mathrm{T}} b_0$. For any $\lambda < \vartheta_1$, we have

$$\begin{aligned}
\widehat{\chi}(\lambda) &= \|b_0\| e_1^{\mathrm{T}} [(T_{k_{\max}} - \lambda I)^{-1}]^2 \|b_0\| e_1 - \gamma^2 \\
&= \|b_0\| e_1^{\mathrm{T}} [(T - \lambda I)^{\dagger}]^2 \|b_0\| e_1 - \gamma^2 \\
&= b_0^{\mathrm{T}} \widehat{Q}\widehat{Q}^{\mathrm{T}} [(PAP - \lambda I)^{\dagger}]^2 \widehat{Q}\widehat{Q}^{\mathrm{T}} b_0 - \gamma^2 \\
&= b_0^{\mathrm{T}} [(PAP - \lambda I)^{\dagger}]^2 b_0 - \gamma^2 \\
&= b_0^{\mathrm{T}} SS^{\mathrm{T}} [(PAP - \lambda I)^{\dagger}]^2 SS^{\mathrm{T}} b_0 - \gamma^2 \\
&= [g_0^{\mathrm{T}}\ 0] \begin{bmatrix} [(H - \lambda I)^{\dagger}]^2 & 0 \\ 0 & [(-\lambda I)^{\dagger}]^2 \end{bmatrix} [g_0^{\mathrm{T}}\ 0]^{\mathrm{T}} - \gamma^2 \\
&= g_0^{\mathrm{T}} [(H - \lambda I)^{\dagger}]^2 g_0 - \gamma^2 =: \widetilde{\chi}(\lambda).
\end{aligned}$$

Therefore, $\widetilde{\chi}(\lambda) = 0$ and $\widetilde{\chi}(\lambda) < 0$ for $\lambda < \mu^{(k_{\max})}$, implying $\mu^{(k_{\max})}$ is the smallest root of $\widetilde{\chi}(\lambda)$.

On the other hand, by the definition of the easy case, $b_0^{\mathrm{T}} z_* \neq 0$ for all possible minimizers (λ_*, z_*) of QEPmin (3.19). Theorem 3.2.4 says that $z_* = S_1 w_*$ for some $w_* \in \mathbb{R}^{n-m}$ and thus $g^{\mathrm{T}} w_* = b_0^{\mathrm{T}} S_1 w_* = b_0^{\mathrm{T}} z_* \neq 0$. By Theorem 3.2.6, $\lambda_* < \lambda_{\min}(H)$. Therefore, it is related to case (1)

or subcase (i) in case (2) of the proof in Lemma 3.2.4, for which λ_* is the smallest root of $\widetilde{\chi}(\lambda)$, and thus $\lambda_* = \mu^{(k_{\max})}$. □

Theorem 3.2.13 guarantees that the minimizer of CRQopt (3.1) is unique if QEPmin (3.19) is in the easy case. We also have established a finite step stopping property for Algorithm 4 as detailed in the following theorem, since $k_{\max} \leq n$.

Corollary 3.3.1. *Suppose* QEPmin (3.19) *is in the easy case, and let* $(\mu^{(k)}, w^{(k)})$ *be the minimizer of* rQEPmin (3.62). *Define* $u^{(k)}$ *as in* (3.51) *and* $k_{\max}$ *is the smallest* k *such that* $\beta_{k+1} = 0$. *Then* $\left(\mu^{(k_{\max})}, u^{(k_{\max})}\right)$ *solves* LGopt (3.14), *and* $v^{(k_{\max})} = u^{(k_{\max})} + n_0$ *is the unique minimizer of* CRQopt (3.1).

3.3.5 Hard case

The hard case is characterized by Theorem 3.2.12 and we translate $g_0 \perp \mathcal{U}$ into $b_0 \perp \mathcal{V}$, where $\mathcal{V}$ is the eigenspace of PAP associated with its eigenvalue $\lambda_{\min}(H)$. For this reason, $\mathcal{K}_k(PAP, b_0)$ will contain no eigen-information of PAP associated with $\lambda_{\min}(H)$. Nonetheless, rLGopt (3.45) and rQEPmin (3.62) can be still formed and solved to yield approximations to the original CRQopt (3.1) with suitable stoping criteria satisfied. But the approximations will be utterly wrong if it is indeed in the hard case. Hence in practice it is important to detect when the hard case occurs.

Denote by (λ_*, z_*) the minimizer of LGopt (3.14). In the easy case, the smallest root of $\widetilde{\chi}(\lambda)$ is λ_* and $\lambda_* < \lambda_{\min}(H)$, while in the hard case, $\lambda_* = \lambda_{\min}(H)$ and the smallest root of $\widetilde{\chi}(\lambda)$ defined in (3.71) is greater than or equal to $\lambda_{\min}(H)$. Since $\mu^{(k)}$ converges to $\mu^{(k_{\max})}$, eventually whether $\mu^{(k)} < \lambda_{\min}(H)$ provide a reasonably good test to see if it is the easy case. Therefore, we propose to detect hard case as follows:

1. Solve rLGopt (3.45) or rQEPmin (3.62).
2. Run the Lanczos process with $M = PAP$ with $r_0 = Pc$, where $c \in \mathbb{R}^n$ is random to compute $\lambda_{\min}(H)$ of PAP and its associated eigenvector $\tilde{z}$;
3. Check if the optimal value of rLGopt (3.45) or rQEPmin (3.62) is greater than or equal to $\lambda_{\min}(H)$ within a prescribed accuracy.

4. If it is, then QEPmin (3.19) is in the hard case; Compute an approximation $\tilde{x}$ of $x_* = -(PAP - \lambda_* I)^\dagger b_0$ as follows:

$$\tilde{y} = \arg\min_{y \in \mathbb{R}^k} \left\| \begin{bmatrix} T_k \\ \beta_{k+1} e_k^{\mathrm{T}} \end{bmatrix} y + \|b_0\| e_1 \right\|, \ \tilde{x} = Q_k \tilde{y}.$$

Finally an approximate minimizer of LGopt (3.14) is given by $\tilde{x} + \sqrt{\gamma^2 - \|\tilde{x}\|^2}\,(\tilde{z}/\|\tilde{z}\|)$.

A remark is in order for item 2 above. Because of the randomness in c, with probability 1, $r_0 = Pc$ will have a significant component in $S_1\mathcal{U}$, where $\mathcal{U}$ is as defined in Theorem 3.2.11. Thus $\lambda_{\min}(H)$ will get computed.

3.4 Convergence analysis of the Lanczos algorithm

In this section, we present a convergence analysis of the Lanczos algorithm (Algorithm 4) for solving CRQopt (3.1) in the easy case. Let $h(v) = v^{\mathrm{T}} A v$ be the objective function of CRQopt (3.1), v_* be the unique solution of CRQopt (3.1) and (λ_*, u_*) be the solution of LGopt (3.14). Our main results are upper bounds on the errors $h(v^{(k)}) - h(v_*)$, $\|v^{(k)} - v_*\|$ and $|\mu^{(k)} - \lambda_*|$, where $v^{(k)}$, defined in (3.70), is the kth approximation by Algorithm 4 and $(\mu^{(k)}, x^{(k)})$ is the solution of rLGopt (3.45). Our technique is analogous to that in [70].

We start by establishing an optimality property of $v^{(k)}$, as an approximation of v_*, that minimizes $h(v)$ over $n_0 + \mathcal{K}_k(PAP, b_0)$.

Theorem 3.4.1. *Let $v^{(k)}$ be defined in (3.70). Then it holds that*

$$h(v^{(k)}) = \min_{v \in n_0 + \mathcal{K}_k(PAP, b_0), \|v\| = 1} h(v). \tag{3.73}$$

Proof. Recall that $(\mu^{(k)}, x^{(k)})$ solves rLGopt (3.45). Consider the optimization problem

$$\begin{cases} \min \ \ell(x) := x^{\mathrm{T}} T_k x + 2\|b_0\| e_1^{\mathrm{T}} x, & \text{(3.74a)} \\ \text{s.t. } \|x\| = \gamma. & \text{(3.74b)} \end{cases}$$

By the theory of Lagrangian multipliers, we find the Lagrangian equations for (3.74) are

$$(T_k - \lambda I)x = -\|b_0\| e_1, \quad \|x\| = \gamma. \tag{3.75}$$

Following the same argument as we did to prove Lemma 3.2.1, we can reach the same conclusion that $\ell(x)$ is strictly increasing with respect to λ in the solution pair (λ, x) of (3.75). Therefore, in order to minimize $\ell(x)$, we need to find the smallest Lagrangian multiplier satisfying (3.75). Hence, solving (3.74) is equivalent to solving rLGopt (3.45) for which $(\mu^{(k)}, x^{(k)})$ is a minimizer and thus $x^{(k)}$ solves (3.74), where $x^{(k)}$ is defined in (3.68).

By definition, $u^{(k)} = Q_k x^{(k)}$ and $v^{(k)} = u^{(k)} + n_0$. For any $v \in n_0 + \mathcal{K}_k(PAP, b_0)$ with $\|v\| = 1$, let

$$u = v - n_0 \in \mathcal{K}_k(PAP, b_0) \subset \mathcal{N}(C^{\mathrm{T}}). \tag{3.76}$$

Hence $Pu = u$, $\|u\| = \gamma$, and $u = Q_k \widetilde{u}$ for some $\widetilde{u} \in \mathbb{R}^k$. We have $v = u + n_0 = Pu + n_0$ and

$$\begin{aligned}
h(v) &= (Pu + n_0)^{\mathrm{T}} A (Pu + n_0) \\
&= u^{\mathrm{T}} PAPu + 2b_0^{\mathrm{T}} u + n_0^{\mathrm{T}} A n_0 \\
&= \widetilde{u}^{\mathrm{T}} Q_k^{\mathrm{T}} PAPQ_k \widetilde{u} + 2b_0^{\mathrm{T}} Q_k \widetilde{u} + n_0^{\mathrm{T}} A n_0 \\
&= \widetilde{u}^{\mathrm{T}} T_k \widetilde{u} + 2\|b_0\| e_1^{\mathrm{T}} \widetilde{u} + n_0^{\mathrm{T}} A n_0 \\
&\geq [x^{(k)}]^{\mathrm{T}} T_k x^{(k)} + 2\|b_0\| e_1^{\mathrm{T}} x^{(k)} + n_0^{\mathrm{T}} A n_0 \qquad \text{(since } x^{(k)} \text{ solves (3.74))} \\
&= [x^{(k)}]^{\mathrm{T}} Q_k^{\mathrm{T}} PAPQ_k x^{(k)} + 2b_0^{\mathrm{T}} Q_k x^{(k)} + n_0^{\mathrm{T}} A n_0 \\
&= [u^{(k)}]^{\mathrm{T}} PAPu^{(k)} + 2b_0^{\mathrm{T}} u^{(k)} + n_0^{\mathrm{T}} A n_0 \\
&= (u^{(k)} + n_0)^{\mathrm{T}} A (u^{(k)} + n_0) \\
&= h(v^{(k)}).
\end{aligned}$$

Since $v \in n_0 + \mathcal{K}_k(PAP, b_0)$ with $\|v\| = 1$ but otherwise is arbitrary, (3.73) holds. □

Recall that H and g_0 are defined in (3.23) and S_1, S_2 in (3.20). Let $\theta_{\min}$ and $\theta_{\max}$ be the smallest and the largest eigenvalue of H, respectively, v_* be the minimizer of CRQopt (3.1), and λ_* be the optimal objective value of LGopt (3.14). Then

$$(\lambda_*, u_*) \quad \text{with } u_* = Pv_* = v_* - n_0$$

is a minimizer of LGopt (3.14). Set

$$\kappa \equiv \kappa(H - \lambda_* I) := \frac{\theta_{\max} - \lambda_*}{\theta_{\min} - \lambda_*}.$$

To estimate $h(v^{(k)})-h(v_*)$, $\|v^{(k)}-v_*\|$ and $|\mu^{(k)}-\lambda_*|$, we first establish a lemma that provides a way to bound $h(v^{(k)})-h(v_*)$, $\|v^{(k)}-v_*\|$ and $|\mu^{(k)}-\lambda_*|$ in terms of any nonzero $v \in n_0+\mathcal{K}_k(PAP, b_0)$.

Lemma 3.4.1. *For any nonzero $v \in n_0+\mathcal{K}_k(PAP, b_0)$, we have*

$$0 \le h(v^{(k)}) - h(v_*) \le 4\|H-\lambda_* I\|_2 \cdot \|v-v_*\|_2^2, \tag{3.77a}$$

$$\|v^{(k)}-v_*\| \le 2\sqrt{\kappa}\,\|v-v_*\|_2, \tag{3.77b}$$

$$|\mu^{(k)}-\lambda_*| \le \frac{1}{\gamma^2}\left[4\|H-\lambda_* I\|_2 \cdot \|v-v_*\|_2^2 + 2\sqrt{\kappa}\,\|b_0\|_2 \cdot \|v-v_*\|_2\right]. \tag{3.77c}$$

Proof. For $v \in n_0+\mathcal{K}_k(PAP, b_0)$, let

$$u = v-n_0 \in \mathcal{K}_k(PAP, b_0),\ \widetilde{u} = \gamma u/\|u\|,\ \widetilde{v} = n_0+\widetilde{u} \in n_0+\mathcal{K}_k(PAP, b_0). \tag{3.78}$$

First, we have $|\|u\|-\gamma| = |\|u\|-\|u_*\|| \le \|u-u_*\| = \|v-v_*\|$, which leads to

$$\left|1-\frac{\gamma}{\|u\|}\right| \le \frac{\|v-v_*\|}{\|u\|}. \tag{3.79}$$

Let $r = \widetilde{v}-v_*$. We have

$$\begin{aligned}
\|r\| = \|v_*-\widetilde{v}\| &\le \|v_*-v\| + \|v-\widetilde{v}\| \\
&\le \|v_*-v\| + \|u-\widetilde{u}\| \\
&= \|v_*-v\| + \left\|u-\frac{\gamma u}{\|u\|}\right\| \\
&= \|v_*-v\| + \|u\| \times \left|1-\frac{\gamma}{\|u\|}\right| \\
&\le 2\|v_*-v\|,
\end{aligned} \tag{3.80}$$

where we have used (3.79) to infer the last inequality.

The first inequality in (3.77a) holds because

$$h(v^{(k)}) = \min_{v\in n_0+\mathcal{K}_k(PAP,b_0),\,\|v\|=1} h(v) \ge \min_{v\in n_0+\mathcal{N}(C^{\mathrm{T}}),\,\|v\|=1} h(v) = h(v_*).$$

Let $f(u) = u^{\mathrm{T}}Au + 2u^{\mathrm{T}}b_0$, it can be verified that $h(v) = h(u+n_0) = f(u) + n_0^{\mathrm{T}}An_0$. Therefore,

$$\widetilde{u}-u_* = \widetilde{v}-v_* = r,\ h(\widetilde{v})-h(v_*) = f(\widetilde{u})-f(u_*). \tag{3.81}$$

Set $s = S_1^{\mathrm{T}} r$. It follows from $r \in \mathcal{N}(C^{\mathrm{T}})$ that $r = S_1 s$ and $\|s\| = \|r\|$. Noting that $\widetilde{v}$ satisfies the constraint of CRQopt (3.1) and that $\widetilde{u} = u_* + r$, we have

$$
\begin{aligned}
0 \le h(v^{(k)}) - h(v_*) &\le h(\widetilde{v}) - h(v_*) && (3.82)\\
&\overset{(3.81)}{=} f(\widetilde{u}) - f(u_*) = f(u_* + r) - f(u_*)\\
&= r^{\mathrm{T}} PAPr + 2r^{\mathrm{T}}(PAPu_* + b_0)\\
&= r^{\mathrm{T}} PAPr + 2\lambda_* r^{\mathrm{T}} u_* && (3.83)\\
&= r^{\mathrm{T}}(PAP - \lambda_* I)r && (3.84)\\
&= s^{\mathrm{T}} S_1^{\mathrm{T}}(PAP - \lambda_* I) S_1 s\\
&= s^{\mathrm{T}}(H - \lambda_* I)s\\
&\le \|H - \lambda_* I\| \|s\|^2 = \|H - \lambda_* I\| \|r\|^2\\
&\overset{(3.80)}{\le} 4\|H - \lambda_* I\| \|v_* - v\|^2, && (3.85)
\end{aligned}
$$

yielding the second inequality in (3.77a), where we have used $(PAP - \lambda_* I)u_* = -b_0$ to get (3.83) and

$$
\|r\|^2 + 2r^{\mathrm{T}} u_* = \|u_* + r\|^2 - \|u_*\|^2 = \|\widetilde{u}\|^2 - \|u_*\|^2 = 0
$$

to obtain $2r^{\mathrm{T}} u_* = -r^{\mathrm{T}} r$ and then (3.84).

Next we prove (3.77b). Define

$$
\widetilde{f}(u) := f(u) - \lambda_* u^{\mathrm{T}} u = u^{\mathrm{T}}(PAP - \lambda_* I)u + 2u^{\mathrm{T}} b_0.
$$

Noticing $(PAP - \lambda_* I)u_* + b_0 = 0$ by (3.14b), let $u^{(k)} = v^{(k)} - n_0$, we have

$$
\widetilde{f}(u^{(k)}) = \widetilde{f}(u_*) + (u^{(k)} - u_*)^{\mathrm{T}}(PAP - \lambda_* I)(u^{(k)} - u_*).
$$

Therefore

$$
\widetilde{f}(u^{(k)}) - \widetilde{f}(u_*) \ge (\theta_{\min} - \lambda_*)\|u^{(k)} - u_*\|^2 = (\theta_{\min} - \lambda_*)\|v^{(k)} - v_*\|^2.
$$

On the other hand,

$$
\widetilde{f}(u^{(k)}) - \widetilde{f}(u_*) = [f(u^{(k)}) + \lambda_* \|u^{(k)}\|^2] - [f(u_*) + \lambda_* \|u_*\|^2] = f(u^{(k)}) - f(u_*) = h(v^{(k)}) - h(v_*),
$$

yielding

$$
(\theta_{\max} - \lambda_*)\|v^{(k)} - v_*\|^2 \le h(v^{(k)}) - h(v_*) \le 4\|H - \lambda_* I\| \|v - v_*\|^2, \qquad (3.86)
$$

which leads to (3.77b).

To prove (3.77c), we pre-multiply $(PAP - \lambda_* I)u_* = -b_0$ by u_*^{T} and use $u_*^{\mathrm{T}} u_* = \gamma^2$ to get

$$\gamma^2 \lambda_* = u_*^{\mathrm{T}} PAP u_* + u_*^{\mathrm{T}} b_0 = v_*^{\mathrm{T}} PAP v_* + v_*^{\mathrm{T}} b_0, \tag{3.87}$$

since $Pv_* = u_*$ and $Pb_0 = b_0$. By (3.5a), we have $h(v_*) = v_*^{\mathrm{T}} PAP v_* + 2v_*^{\mathrm{T}} b_0 + n_0^{\mathrm{T}} A n_0$ and thus

$$\gamma^2 \lambda_* = h(v_*) - v_*^{\mathrm{T}} b_0 - n_0^{\mathrm{T}} A n_0.$$

On the other hand, it follows from rLGopt (3.45) that $[x^{(k)}]^{\mathrm{T}} T_k x^{(k)} + \|b_0\|_2 [x^{(k)}]^{\mathrm{T}} e_1 = \gamma^2 \mu^{(k)}$. Plug in

$$T_k = Q_k^{\mathrm{T}} PAP Q_k, \;\; u^{(k)} = Q_k x^{(k)}, \;\; Q_k^{\mathrm{T}} b_0 = \|b_0\|_2 e_1, \;\; v^{(k)} = u^{(k)} + n_0$$

to get

$$\gamma^2 \mu^{(k)} = h(u^{(k)}) - [u^{(k)}]^{\mathrm{T}} b_0 = h(v^{(k)}) - [v^{(k)}]^{\mathrm{T}} b_0 - n_0^{\mathrm{T}} A n_0. \tag{3.88}$$

It follows from (3.87) and (3.88) that

$$\left| \mu^{(k)} - \lambda_* \right| = \frac{1}{\gamma^2} \left| h(v^{(k)}) - h(v_*) - b_0^{\mathrm{T}} (v^{(k)} - v_*) \right| \le \frac{1}{\gamma^2} \left[|h(v^{(k)}) - h(v_*)| + \|b_0\|_2 \|v^{(k)} - v_*\|_2 \right], \tag{3.89}$$

which combined with (3.77a) and (3.77b) yield (3.77c). □

The inequalities in (3.77) hold for any $v \in n_0 + \mathcal{K}_k(PAP, b_0)$ which, in general can be expressed as

$$v = n_0 + \phi_{k-1}(PAP) b_0,$$

where $\phi_{k-1}(\,\cdot\,)$ is a polynomial of degree $k-1$. By judicially picking certain ϕ_{k-1}, meaningful upper bounds on $h(v^{(k)}) - h(v_*)$, $\|v^{(k)} - v_*\|$ and $|\mu^{(k)} - \lambda_*|$ are readily obtained. These upper bounds expose the convergence behavior of $v^{(k)}$. The next theorem contains our main results of the section.

Theorem 3.4.2. *Suppose* CRQopt (3.1) *is in the easy case, and let v_* be its minimizer. Let (λ_*, u_*) be the minimizer of the corresponding* LGopt (3.14)*, and, for its corresponding* pLGopt (3.25)*, let $\theta_{\min}$ and $\theta_{\max}$ be the smallest and largest eigenvalue of H, respectively, and set*

$$\kappa = \kappa(H - \lambda_* I) := \frac{\theta_{\max} - \lambda_*}{\theta_{\min} - \lambda_*}.$$

Then the following statements hold:

(a) *The sequence* $\{h(v^{(k)})\}$ *is nonincreasing;*

(b) *For* $k \leq k_{\max}$*, the smallest* k *such that* $\beta_{k+1} = 0$*,*

$$0 \leq h(v^{(k)}) - h(v_*) \leq 16\gamma^2 \|H - \lambda_* I\|_2 \left[\Gamma_\kappa^k + \Gamma_\kappa^{-k}\right]^{-2}, \tag{3.90a}$$

$$\|v^{(k)} - v_*\|_2 \leq 4\gamma\sqrt{\kappa} \left[\Gamma_\kappa^k + \Gamma_\kappa^{-k}\right]^{-1}, \tag{3.90b}$$

$$|\mu^{(k)} - \lambda_*| \leq 16\|H - \lambda_* I\|_2 \left[\Gamma_\kappa^k + \Gamma_\kappa^{-k}\right]^{-2} + \frac{4}{\gamma}\|b_0\|_2\sqrt{\kappa} \left[\Gamma_\kappa^k + \Gamma_\kappa^{-k}\right]^{-1}, \tag{3.90c}$$

where

$$\Gamma_\kappa := \frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}. \tag{3.91}$$

Proof. Item (a) holds because for any $0 \leq k \leq k_{\max}$,

$$h(v^{(k)}) = \min_{v \in n_0 + \mathcal{K}_k(PAP, b_0),\, \|v\|=1} h(v) \geq \min_{v \in n_0 + \mathcal{K}_{k+1}(PAP, b_0),\, \|v\|=1} h(v) = h(v^{(k+1)}).$$

Before we prove item (b), we note that $(\lambda_*, S_1^{\mathrm{T}} v_*)$ solves pLGopt (3.25). In particular, since pLGopt (3.25) is in the easy case,

$$S_1^{\mathrm{T}} v_* = -(H - \lambda_* I)^{-1} g_0. \tag{3.92}$$

Consider now $v \in n_0 + \mathcal{K}_k(PAP, b_0)$. Then $S_1^{\mathrm{T}} v \in \mathcal{K}_k(H, g_0) = \mathcal{K}_k(H - \lambda_* I, g_0)$. Therefore by (3.92)

$$\begin{aligned} S_1^{\mathrm{T}} v - S_1^{\mathrm{T}} v_* &= \phi_{k-1}(H - \lambda_* I) g + (H - \lambda_* I)^{-1} g_0 \\ &= [\phi_{k-1}(H - \lambda_* I)\,(H - \lambda_* I) + I](H - \lambda_* I)^{-1} g_0 \\ &= -\psi_k(H - \lambda_* I) S_1^{\mathrm{T}} v_*, \end{aligned} \tag{3.93}$$

where ϕ_{k-1} is a polynomial of degree $k-1$, and $\psi_k(t) = 1 + t\phi_{k-1}(t)$, a polynomial of degree k, that satisfies $\psi_k(0) = 1$. Note that $\psi_k(0) = 1$ but otherwise ψ_k is an arbitrary polynomial of degree k, offering the freedom that we will take advantage of in a moment.

Given that v_* solves CRQopt (3.1), we have

$$\gamma = \|P v_*\| = \|S_1 S_1^{\mathrm{T}} v_*\| = \|S_1^{\mathrm{T}} v_*\|.$$

Thus

$$\begin{aligned}\min_{v\in n_0+\mathcal{K}_k(PAP,b_0)}\|v-v_*\| &= \min_{v\in n_0+\mathcal{K}_k(PAP,b_0)}\|S_1^{\mathrm{T}}v-S_1^{\mathrm{T}}v_*\| \qquad \text{(use (3.93))}\\ &\le \gamma \min_{\psi_k(0)=1}\|\psi_k(H-\lambda_* I)\|\\ &\le \gamma \min_{\psi_k(0)=1}\max_{1\le i\le n-m}|\psi_k(\theta_i-\lambda_*)| \qquad (3.94)\\ &\le \gamma \min_{\psi_k(0)=1}\max_{t\in[\theta_{\min}-\lambda_*,\theta_{\max}-\lambda_*]}|\psi_k(t)|. \qquad (3.95)\end{aligned}$$

The inequality (3.95) holds for any polynomial ψ_k of degree k such that $\psi_k(0)=1$. For the purpose of establishing upper bounds, we will pick one that is defined through the kth Chebyshev polynomial of the first kind:

$$\begin{aligned}\mathscr{T}_k(t) &= \cos(k\arccos t) && \text{for } |t|\le 1, && (3.96a)\\ &= \frac{1}{2}\left[\left(t+\sqrt{t^2-1}\right)^k+\left(t+\sqrt{t^2-1}\right)^{-k}\right] && \text{for } |t|\ge 1. && (3.96b)\end{aligned}$$

Specifically, we take

$$\psi_k(t)=\mathscr{T}_k\left(\frac{2t-(\alpha+\beta)}{\beta-\alpha}\right)\Big/\mathscr{T}_k\left(\frac{-(\alpha+\beta)}{\beta-\alpha}\right), \qquad (3.97)$$

where $\alpha=\theta_{\min}-\lambda_*$ and $\beta=\theta_{\max}-\lambda_*$. Evidently, $\psi_k(0)=1$, and for $t\in[\theta_{\min}-\lambda_*,\theta_{\max}-\lambda_*]=[\alpha,\beta]$, we have

$$|2t-(\alpha+\beta)| = \big||t+\lambda_*-\theta_{\min}|-|t+\lambda_*-\theta_{\max}|\big| \le |\theta_{\max}-\theta_{\min}| = \beta-\alpha.$$

Therefore, $[2t-(\alpha+\beta)]/(\beta-\alpha)\in[-1,1]$, and thus for $t\in[\alpha,\beta]$ [38]

$$|\psi_k(t)| \le \left|\mathscr{T}_k\left(\frac{-(\alpha+\beta)}{\beta-\alpha}\right)\right|^{-1} = \left|\mathscr{T}_k\left(\frac{\kappa+1}{\kappa-1}\right)\right|^{-1} = 2\left[\Gamma_\kappa^k+\Gamma_\kappa^{-k}\right]^{-1}. \qquad (3.98)$$

Minimize the right-most quantities in (3.77) over $v\in n_0+\mathcal{K}_k(PAP,b_0)$, utilize (3.95) and (3.98) to get the inequalities in (3.90). □

We end this section with remarks regarding the results in Theorem 3.4.2.

Remark 3.4.1. The rate of convergence for our Lanczos algorithm depends on κ. Recall that $\kappa=\frac{\theta_{\max}-\lambda_*}{\theta_{\min}-\lambda_*}$. When λ_* is far away from $\theta_{\min}$, we may regard that CRQopt (3.1) is far from hard case. In this case, κ moves towards 1, and we expect faster convergence of our Lanczos algorithm. However, when CRQopt (3.1) is near hard case, i.e., $\theta_{\min}\approx\lambda_*$, κ is large, and Theorem 3.4.2

suggests slow convergence. These conclusions derived from Theorem 3.4.2 are consistent with the numerical observations in [28] that "a Lanczos type process seems to be very effective when the problem is far from the hard case". We provide an example in example 3.5.2 later to illustrate the relationship between the rate of convergence and κ. □

Remark 3.4.2. For most examples, the bounds suggested in (3.90a) and (3.90b) are sharp. However, there are some cases where the bounds suggested in (3.90a) and (3.90b) are pessimistic. This occurs for near-hard situations where $\lambda_* \approx \theta_{\min}$ and sometimes the Lanczos method can still enjoy fast convergence, even though the bounds in (3.90a) and (3.90b) do not suggest so. One of such situations is when

$$\kappa_+ := \frac{\theta_{\max} - \lambda_*}{\theta_2 - \lambda_*}$$

is small, even though $\theta_{\min} \approx \lambda_*$ and thus κ is huge, where θ_2 is the second smallest eigenvalue of H. This suggests that the bounds by (3.90a) and (3.90b) have room for improvement. In fact, instead of (3.97), we may choose

$$\psi_k(t) = \frac{t-\alpha}{-\alpha} \cdot \mathscr{T}_{k-1}\left(\frac{2t-(\alpha_+ + \beta)}{\beta - \alpha_+}\right) \Big/ \mathscr{T}_{k-1}\left(\frac{-(\alpha_+ + \beta)}{\beta - \alpha_+}\right), \tag{3.99}$$

where α and β are as before, and $\alpha_+ = \theta_2 - \lambda_*$. Evidently, again $\psi_k(0) = 1$, but now $\psi_k(\theta_1 - \lambda_*) = 0$. We have

$$\begin{aligned} \max_{1\le i\le n-m} |\psi_k(\theta_i - \lambda_*)| &= \max_{2\le i\le n-m} |\psi_k(\theta_i - \lambda_*)| \le \max_{t\in[\alpha_+,\beta]} |\psi_k(t)| \\ &\le \max_{t\in[\alpha_+,\beta]} \left|\frac{t-\alpha}{-\alpha}\right| \cdot 2\left[\Gamma_{\kappa_+}^{(k-1)} + \Gamma_{\kappa_+}^{-(k-1)}\right]^{-1} \\ &= \frac{2(\theta_{\max} - \theta_{\min})}{\theta_{\min} - \lambda_*}\left[\Gamma_{\kappa_+}^{(k-1)} + \Gamma_{\kappa_+}^{-(k-1)}\right]^{-1}. \end{aligned} \tag{3.100}$$

It combined with (3.94) will lead to bounds

$$h(v^{(k)}) - h(v_*) \le \frac{16\gamma^2\|H - \lambda_* I\|_2(\theta_{\max} - \theta_{\min})}{(\theta_{\min} - \lambda_*)}\left[\Gamma_{\kappa_+}^{(k-1)} + \Gamma_{\kappa_+}^{-(k-1)}\right]^{-2}, \tag{3.101a}$$

$$\|v^{(k)} - v_*\|_2 \le 4\gamma\sqrt{\kappa}\frac{\theta_{\max} - \theta_{\min}}{\theta_{\min} - \lambda_*}\left[\Gamma_{\kappa_+}^{(k-1)} + \Gamma_{\kappa_+}^{-(k-1)}\right]^{-1}, \tag{3.101b}$$

$$\begin{aligned} |\mu^{(k)} - \lambda_*| \le \frac{\theta_{\max} - \theta_{\min}}{\theta_{\min} - \lambda_*}\Bigg[16\|H - \lambda_* I\|_2\left[\Gamma_{\kappa_+}^{(k-1)} + \Gamma_{\kappa_+}^{-(k-1)}\right]^{-2} \\ + \frac{4}{\gamma}\|b_0\|_2\sqrt{\kappa}\left[\Gamma_{\kappa_+}^{(k-1)} + \Gamma_{\kappa_+}^{-(k-1)}\right]^{-1}\Bigg]. \end{aligned} \tag{3.101c}$$

which can be much sharper than the ones by (3.90a) and (3.90b) and they will be sharper if $\theta_{\min} \approx \lambda_*$ and there is a reasonably gap between $\theta_{\min}$ and θ_2. We show such an example later in example 3.5.3. □

Remark 3.4.3. In our numerical experiments, we observed that the bound (3.90c) often decays much slower than $|\mu^{(k)} - \lambda_*|$. Recall that in obtaining (3.90c), we used

$$\left|b_0^{\mathrm{T}}(v^{(k)} - v_*)\right| \leq \|b_0\| \left\|v^{(k)} - v_*\right\| \tag{3.102}$$

in (3.89). It turns out that $\|b_0\| \left\|v^{(k)} - v_*\right\|$ decays much slower than $\left|b_0^{\mathrm{T}}(v^{(k)} - v_*)\right|$, as evidenced by our numerical tests. While at this point we don't know how to estimate $\left|b_0^{\mathrm{T}}(v^{(k)} - v_*)\right|$ much more than accurately than via the inequality (3.102), we offer a plausible explanation as follows. Let $u^{(k)} = v^{(k)} - n_0$ and $u_* = v_* - n_0$. Since $u_*^{\mathrm{T}} u_* = [u^{(k)}]^{\mathrm{T}} u^{(k)} = \gamma^2$, we have

$$\begin{aligned} \left|u_*^{\mathrm{T}}(v^{(k)} - v_*)\right| &= \left|u_*^{\mathrm{T}} u^{(k)} - u_*^{\mathrm{T}} u_*\right| = \frac{1}{2}\left|2u_*^{\mathrm{T}} u^{(k)} - u_*^{\mathrm{T}} u_* - [u^{(k)}]^{\mathrm{T}} u^{(k)}\right| \\ &= \frac{1}{2}\left\|u^{(k)} - u_*\right\|_2^2 = \frac{1}{2}\left\|v^{(k)} - v_*\right\|_2^2. \end{aligned} \tag{3.103}$$

By (3.90b), $\left\|v^{(k)} - v_*\right\|_2^2$ is of order $O\left(\left[\Gamma_\kappa^k + \Gamma_\kappa^{-k}\right]^{-2}\right)$, and thus $\left|u_*^{\mathrm{T}}(v^{(k)} - v_*)\right|$ is also of order $O\left(\left[\Gamma_\kappa^k + \Gamma_\kappa^{-k}\right]^{-2}\right)$ as (3.103) suggests. Let $\theta_1 \leq \theta_2 \leq \cdots \leq \theta_{n-m}$ be the eigenvalues of PAP restricted to the subspace $\mathcal{R}(P)$, $y_1, y_2, \cdots, y_{n-m}$ be the corresponding orthonormal eigenvectors in $\mathcal{R}(P)$, $u_* = \sum_{i=1}^{n-m} \xi_i y_i$, and $v^{(k)} - v_* = u^{(k)} - u_* = \sum_{i=1}^{n-m} \epsilon_i y_i$. Then we have

$$\left|u_*^{\mathrm{T}}(v^{(k)} - v_*)\right| = \left|\sum_{i=1}^{n-m} \xi_i \epsilon_i\right|.$$

On the other hand $b_0 = -(PAP - \lambda_* I)u_* = -\sum_{i=1}^{n-m}(\theta_i - \lambda_*)\xi_i y_i$ and thus

$$\left|b_0^{\mathrm{T}}(v^{(k)} - v_*)\right| = \left|\sum_{i=1}^{n}(\theta_i - \lambda_*)\xi_i \epsilon_i\right|.$$

Note that sequence $\{\theta_i - \lambda_*\}$ is positive and increasing for the easy case and sequence $\{\xi_i y_i\}$ oscillates for most cases in practice. Therefore, when $\kappa(PAP - \lambda_* I) = \frac{\theta_{n-m} - \lambda_*}{\theta_1 - \lambda_*}$ is modest, i.e., the difference between $\theta_i - \lambda_*$ for different i is modest, we expect that the difference between $\left|b_0^{\mathrm{T}}(v^{(k)} - v_*)\right| = \left|\sum_{i=1}^{n-m}(\theta_i - \lambda_*)\xi_i \epsilon_i\right|$ and $\left|u_*^{\mathrm{T}}(v^{(k)} - v_*)\right| = \left|\sum_{i=1}^{n-m} \xi_i \epsilon_i\right|$ is small. Therefore, the convergence rate of $\left|b_0^{\mathrm{T}}(v^{(k)} - v_*)\right|$ can be similar to the convergence rate of $\left|u_*^{\mathrm{T}}(v^{(k)} - v_*)\right|$, which is $O\left(\left[\Gamma_\kappa^k + \Gamma_\kappa^{-k}\right]^{-2}\right)$. Plausibly, we have explained why the bound (3.90c) decays much slower than the actual $|\mu^{(k)} - \lambda_*|$. □

3.5 Numerical examples – sharpness of error bounds

In this section, we demonstrate the sharpness of our convergence error bounds in Theorem 3.4.2 for the Lanczos algorithm (Algorithm 4) for solving CRQopt (3.1). For that purpose,

we first test examples that are hard for the Lanczos algorithm. The basic idea is similar to that in [39]. Also shown are the history of the normalized residual $\mathrm{NRes}_k^{\mathrm{QEPmin}}$ and its upper bound $\delta_k^{\mathrm{QEPmin}}$ in (3.67b). All numerical examples were carried out in MATLAB.

3.5.1 Construction of difficult CRQopt problems

The convergence analysis of the Lanczos algorithm (Algorithm 4) for solving CRQopt (3.1) presented in Theorem 3.4.2 indicates that the convergence behavior is determined by the spectral distribution of the matrix H defined in pLGopt (3.25) and the optimal value λ_* of LGopt (3.14). Therefore, we construct matrices A, C and vector b through constructing matrices H and g_0 of pLGopt (3.25).

H and g_0. It is not a secret that approximations by the Lanczos procedure converge most slowly when the eigenvalues of these matrices distribute like the zeros or the extreme nodes of Chebyshev polynomials of the first kind [38, 37, 39, 70]. In what follows, we describe one set of test matrix-vector pair (H, g_0) using the extreme nodes of Chebyshev polynomials of the first kind.

The ℓth Chebyshev polynomials of the first kind $\mathscr{T}_\ell(t)$ has $\ell+1$ extreme points in $[-1, 1]$, defined by

$$\tau_{j\ell} = \cos\vartheta_{j\ell}, \quad \text{with} \quad \vartheta_{jl} = \frac{j}{\ell}\pi \quad \text{for} \quad j = 0, 1, \ldots, \ell. \tag{3.104}$$

At these extreme points, $|\mathscr{T}_\ell(\tau_{j\ell})| = 1$. Given scalars α and β such that $\alpha < \beta$, set

$$\omega = \frac{\beta - \alpha}{2}, \quad \tau = -\frac{\alpha + \beta}{\beta - \alpha}. \tag{3.105}$$

The so-called *the ℓth translated Chebyshev extreme nodes* on $[\alpha, \beta]$ are given by [38, 37]

$$\tau_{j\ell}^{\mathrm{trans}} = \omega(\tau_{j\ell} - \tau) \quad \text{for} \quad j = 0, 1, \ldots, \ell. \tag{3.106}$$

It can be verified that $\tau_{0\ell}^{\mathrm{trans}} = \beta$ and $\tau_{\ell\ell}^{\mathrm{trans}} = \alpha$.

Given integers n and m with $m < n$, and the interval $[\alpha, \beta]$, we take

$$H = \mathrm{diag}\left(\tau_{0\,n-m-1}^{\mathrm{trans}}, \tau_{1\,n-m-1}^{\mathrm{trans}}, \ldots, \tau_{n-m-1\,n-m-1}^{\mathrm{trans}}\right). \tag{3.107}$$

Now we construct $g_0 = [g_1, g_2, \cdots, g_{n-m}]^{\mathrm{T}} \in \mathbb{R}^{n-m}$. Recall that the eigenvector of H corresponding to the smallest eigenvalue is e_{n-m}. In order to make pLGopt (3.25) in the easy case, we need to

make g_0 not perpendicular to that eigenvector e_{n-m}, i.e., $g_{n-m} \neq 0$. As a simple choice, we take

$$g_0 = [1, 1, \cdots, 1]^{\mathrm{T}} \in \mathbb{R}^{n-m}. \tag{3.108}$$

A, C and b. With H and g_0 set, we construct matrices A, C and vector b in the following way:

1. Pick $0 < \zeta < 1$, and $a \in \mathbb{R}^m$ with $\|a\| = 1/\zeta$;

2. Pick a random $C \in \mathbb{R}^{n\times m}$ and compute its QR decomposition

$$C = \begin{array}{c} \overset{m}{} \quad \overset{n-m}{} \\ \left[\begin{array}{cc} S_2 & S_1 \end{array}\right] \end{array} \times \begin{array}{c} \overset{m}{} \\ \left[\begin{array}{c} R \\ 0 \end{array}\right] \\ {\scriptstyle n-m} \end{array} \equiv S_2 R. \tag{3.109}$$

3. Let $b = \zeta^2 R^{\mathrm{T}} a$.

4. Take $A_{12} = g_0 a^{\mathrm{T}}$, $A_{22} = \eta I_m$ with $\eta = (g_0^{\mathrm{T}} H^{-1} g_0)/\zeta^2$.

5. Set $A = S \begin{bmatrix} H & A_{12} \\ A_{12}^{\mathrm{T}} & A_{22} \end{bmatrix} S^{\mathrm{T}}$, where $S = [S_1, S_2]$.

Note that by the construction, the matrix A is positive semidefinite when H is positive definite. This is because the Schur complement of H in the matrix $\begin{bmatrix} H & A_{12} \\ A_{12}^{\mathrm{T}} & A_{22} \end{bmatrix}$:

$$\begin{aligned} A_{22} - A_{12}^{\mathrm{T}} H^{-1} A_{12} &= A_{22} - a g_0^{\mathrm{T}} H^{-1} g_0 a^{\mathrm{T}} = A_{22} - (g_0^{\mathrm{T}} H^{-1} g_0) a a^{\mathrm{T}} \\ &= \eta I - (g_0^{\mathrm{T}} H^{-1} g_0) a a^{\mathrm{T}} = (g_0^{\mathrm{T}} H^{-1} g_0)(\|a\|^2 I - a a^{\mathrm{T}}) \end{aligned}$$

is positive semidefinite since H is positive definite and $g_0^{\mathrm{T}} H^{-1} g_0 > 0$.

Verification. Now we verify that CRQopt (3.1) with A, C, b constructed from the process above will yield pLGopt (3.25) with matrices H and g_0 and scalar $\gamma = \sqrt{1-\zeta^2}$, as desired.

Recall the definitions in (3.23):

$$g_0 = S_1^{\mathrm{T}} b_0, \; H = S_1^{\mathrm{T}} P A P S_1 = S_1^{\mathrm{T}} A S_1 \in \mathbb{R}^{(n-m)\times(n-m)}. \tag{3.110}$$

By the construction of A, $S_1^{\mathrm{T}} A S_1 = H$, which is consistent with H defined in (3.110). Further recall that P is a projection matrix onto $\mathcal{N}(C^{\mathrm{T}})$ and the columns of S_1 form an orthonormal basis

of $\mathcal{N}(C^{\mathrm{T}})$. So $P = S_1 S_1^{\mathrm{T}}$. In addition, by the QR factorization (3.109), $(C^{\mathrm{T}})^{\dagger} = S_2 R^{-\mathrm{T}}$, and so $n_0 = (C^{\mathrm{T}})^{\dagger} b = S_2 R^{-\mathrm{T}} b$. By the definition of matrix A, $S_1^{\mathrm{T}} A S_2 = A_{12}$, we have

$$S_1^{\mathrm{T}} b_0 = S_1^{\mathrm{T}} P A n_0 = S_1^{\mathrm{T}} S_1 S_1^{\mathrm{T}} A S_2 R^{-\mathrm{T}} b = S_1^{\mathrm{T}} A S_2 R^{-\mathrm{T}} b = \zeta^2 A_{12} a = \zeta^2 g_0 a^{\mathrm{T}} a = g_0. \tag{3.111}$$

which is consistent with g_0 defined in (3.110). Finally,

$$\gamma = \sqrt{1 - \|n_0\|^2} = \sqrt{1 - \|S_2 R^{-\mathrm{T}} b\|^2} = \sqrt{1 - \|R^{-\mathrm{T}} b\|^2} = \sqrt{1 - \|\zeta^2 a\|^2} = \sqrt{1 - \zeta^2}.$$

3.5.2 Numerical results

For testing purpose, we compute a solution v_* by the direct method in [17] as a reference (exact) solution; otherwise it is generally unknown. We also compute $\kappa = \frac{\lambda_{\max}(H) - \lambda_*}{\lambda_{\min}(H) - \lambda_*}$ to examine our error bounds in Theorem 3.4.2.

The Lanczos algorithm (Algorithm 4) is applied to solve CRQopt (3.1) via QEPmin (3.19) and via LGopt (3.14). For each computed $v^{(k)}$, the kth iteration, we compute relative errors

$$\mathrm{err}_1 = \frac{|(v^{(k)})^{\mathrm{T}} A v^{(k)} - v_*^{\mathrm{T}} A v_*|}{|v_*^{\mathrm{T}} A v_*|}, \quad \mathrm{err}_2 = \|v^{(k)} - v_*\|, \quad \text{and} \quad \mathrm{err}_3 = \frac{|\mu^{(k)} - \lambda_*|}{|\lambda_*|}.$$

Since $\|v_*\| = 1$, the absolute error err_2 is also relative. The stoping criterion for solving QEPmin (3.19) is either $\delta_k^{\mathrm{QEPmin}} < 10^{-15}$ or the number of Lanczos steps reaches $\texttt{maxit} = 200$, where $\delta_k^{\mathrm{QEPmin}}$ is defined in (3.67). The stopping criterion for solving LGopt (3.14) is either $\mathrm{NRes}_k^{\mathrm{LGopt}} < 10^{-15}$ or the number of Lanczos steps reaches $\texttt{maxit} = 200$.

Example 3.5.1. In this example, we test the correctness and convergence behavior of the Lanczos algorithm to solve CRQopt (3.1). Let $n = 1100$, $m = 100$, $\alpha = 1$, $\beta = 100$ or 1000, and construct H as in (3.107) and g_0 as in (3.108). For (A, C, b), let $\zeta = 0.9$ and a be random vector normalized to have norm $1/\zeta$ and then the rest follows Section 3.5.1 in constructing A, C and b.

The convergence histories for err_1, err_2 and err_3 are plotted in Figure 3.2. It can be seen that all converge to the machine precision. Also err_1, err_2 and err_3 are the same, respectively, at every iteration whether CRQopt (3.1) is solved via QEPmin (3.19) or LGopt (3.14), which is consistent with our theory that solving rLGopt (3.45) is equivalent to solving rQEPmin (3.62).

Example 3.5.2. We illustrate the sharpness of the error bounds (3.90) in Theorem 3.4.2 and the relationship between the convergence rate of our Lanczos algorithm and κ.

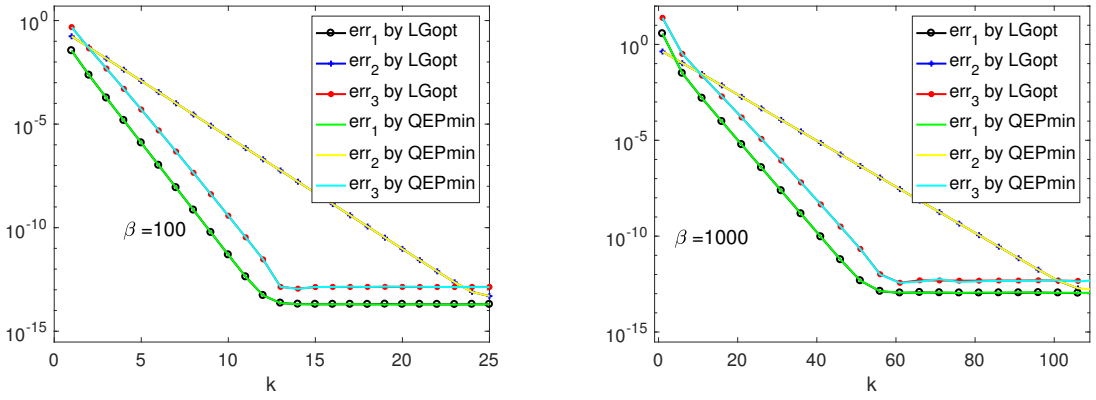

Figure 3.2: Example 3.5.1: history of err_1, err_2 and err_3 for the cases where $\beta = 100$ (left) and $\beta = 1000$ (right).

The same test matrices as in Example 3.5.1, with $\beta = 100$ and 1000 are used. We solve CRQopt (3.1) by solving QEPmin (3.19) and choose the same parameters as in Example 3.5.1. For $\alpha = 1$ and $\beta = 100$, We calculate

$$(\lambda_*, \kappa) = \begin{cases} (-42.6007, 3.2706), & \text{for } (\alpha, \beta) = (1, 100); \\ (-18.2629, 52.8613), & \text{for } (\alpha, \beta) = (1, 1000). \end{cases}$$

Judging from the corresponding κ, we expect our Lanczos algorithm will converge faster for the case $\beta = 100$ than the case $\beta = 1000$. We plot in Figure 3.3 the convergence histories for

err_1 and its upper bound $\frac{16\gamma^2\|H-\lambda_* I\|}{v_*^{\mathrm{T}} A v_*} \left[\Gamma_\kappa^k + \Gamma_\kappa^{-k}\right]^{-2}$ by (3.90a),

err_2 and its upper bound $4\gamma\sqrt{\kappa}\left[\Gamma_\kappa^k + \Gamma_\kappa^{-k}\right]^{-1}$ by (3.90b),

err_3 and its upper bound $\frac{16}{|\lambda_*|}\|H - \lambda_* I\| \left[\Gamma_\kappa^k + \Gamma_\kappa^{-k}\right]^{-2} + \frac{4}{\gamma|\lambda_*|}\sqrt{\kappa}\left[\Gamma_\kappa^k + \Gamma_\kappa^{-k}\right]^{-1}$ by (3.90c).

The bounds for err_1 and err_2 by (3.90a) and (3.90b) for both $\beta = 100$ and $\beta = 1000$ appear sharp. However, the bound for err_3 by (3.90c) is pessimistic. In the plots, err_3 goes to 0 at about a similar rate of err_1, but the bounds by (3.90b) and (3.90c) for err_3 progress at the same rate as the bound by (3.90a) for err_2. We unsuccessfully tried to establish a better bound for err_3 to reflect what we just witnessed, but we offered a plausible explanation in Remark 3.4.3.

As expected, err_1, err_2 and err_3 go to 0 faster for the case $\beta = 100$ than the case $\beta = 1000$. It is consistent with our convergence results in Theorem 3.4.2 that our Lanczos algorithm for CRQopt (3.1) converges faster when κ is smaller.

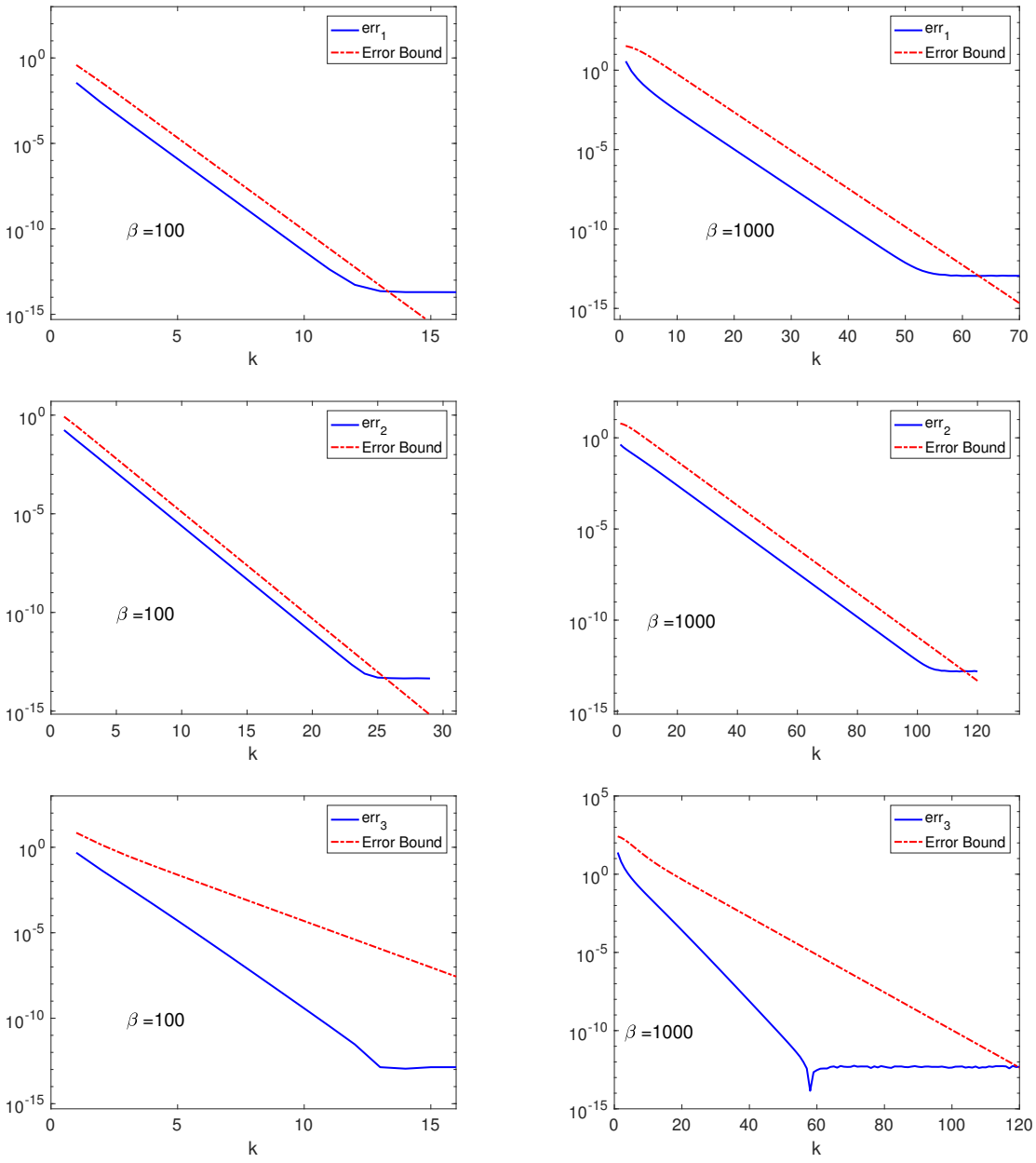

Figure 3.3: Example 3.5.2: histories for err_1 (first row), err_2 (second row), err_3 (third row) and their upper bounds for $\beta = 100$ (left column) and $\beta = 1000$ (right column).

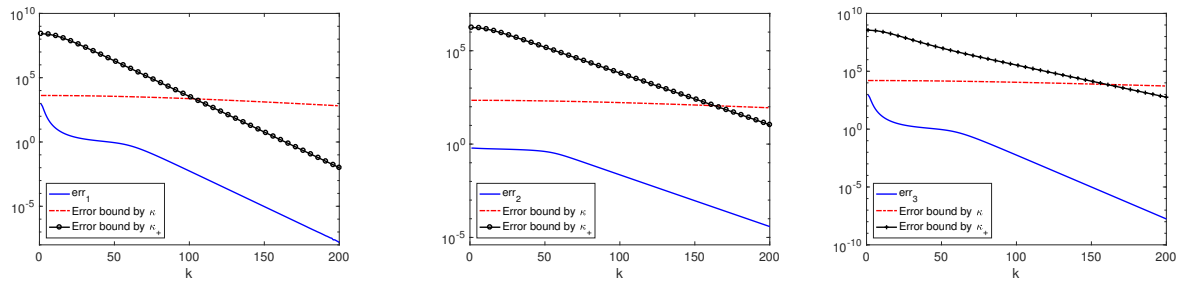

Figure 3.4: Example 3.5.3: histories of err_1, err_2, err_3 and their upper bounds. "Error bound by κ" and "Error bound by κ_+" means upper bounds in (3.90) and (3.101), respectively.

Example 3.5.3. We consider an example where the error bounds in Theorem 3.4.2 are pessimistic, while those by (3.101) can correctly reveal the speed of convergence. This occurs when CRQopt is a "*nearly hard case*", i.e., where the optimal value of the corresponding pLGopt (3.25) $\lambda_* \approx \lambda_{\min}(H)$. Specifically, we choose $n = 1100$, $m = 100$, $\zeta = 0.9$, a a random vector with the norm $1/\zeta$, and

$$H = \mathrm{diag}(\tau^{\mathrm{trans}}_{0\,n-m-2}, \tau^{\mathrm{trans}}_{1\,n-m-2}, \ldots, \tau^{\mathrm{trans}}_{n-m-2\,n-m-2}, 1)$$

with $(\alpha, \beta) = (2, 1000)$ in (3.105) and (3.106), and

$$g_0 = \left[e^{\eta}, e^{2\eta}, \cdots, e^{(n-m)\eta}\right]^{\mathrm{T}}$$

where $\eta = -5 \times 10^{-3}$. In this case, $\lambda_{\min}(H) = 1$ and $\lambda_* = 0.9845$, so $\lambda_{\min}(H) \approx \lambda_*$ and thus it is a nearly hard case. It is computed that

$$\kappa = \frac{\lambda_{\max}(H) - \lambda_*}{\lambda_{\min}(H) - \lambda_*} \approx 6.4466 \times 10^4$$

which is big. We solve the associated CRQopt (3.1) via QEPmin (3.19). In Figure 3.4, we plot the

convergence history:

err_1, its upper bounds $\frac{16\gamma^2\|H-\lambda_* I\|}{v_*^{\mathrm{T}}Av_*}\left[\Gamma_\kappa^k+\Gamma_\kappa^{-k}\right]^{-2}$ by (3.90a), and
$\frac{16\gamma^2\|H-\lambda_* I\|(\theta_{\max}-\theta_{\min})}{(\theta_{\min}-\lambda_*)v_*^{\mathrm{T}}Av_*}\left[\Gamma_{\kappa_+}^{(k-1)}+\Gamma_{\kappa_+}^{-(k-1)}\right]^{-2}$ by (3.101a),

err_2, its upper bounds $4\gamma\sqrt{\kappa}\left[\Gamma_\kappa^k+\Gamma_\kappa^{-k}\right]^{-1}$ by (3.90b), and
$4\gamma\sqrt{\kappa}\frac{\theta_{\max}-\theta_{\min}}{\theta_{\min}-\lambda_*}\left[\Gamma_{\kappa_+}^{(k-1)}+\Gamma_{\kappa_+}^{-(k-1)}\right]^{-1}$ by (3.101b),

err_3, its upper bounds $\frac{16}{|\lambda_*|}\|H-\lambda_* I\|\left[\Gamma_\kappa^k+\Gamma_\kappa^{-k}\right]^{-2}+\frac{4\|b_0\|}{\gamma|\lambda_*|}\sqrt{\kappa}\left[\Gamma_\kappa^k+\Gamma_\kappa^{-k}\right]^{-1}$ by (3.90c), and
$\frac{\theta_{\max}-\theta_{\min}}{|\lambda_*|(\theta_{\min}-\lambda_*)}\left[16\|H-\lambda_* I\|\left[\Gamma_{\kappa_+}^{(k-1)}+\Gamma_{\kappa_+}^{-(k-1)}\right]^{-2}+\frac{4}{\gamma}\|b_0\|\sqrt{\kappa}\left[\Gamma_{\kappa_+}^{(k-1)}+\Gamma_{\kappa_+}^{-(k-1)}\right]^{-1}\right]$
by (3.101c).

It can be observed that The error bounds by Theorem 3.4.2 decay much slower than err_1, err_2 and err_3 in this "near hard case". This is an example for which κ is large but κ_+ is small:

$$\kappa_+ := \frac{\theta_{\max}-\lambda_*}{\theta_2-\lambda_*} \approx 983.7702,$$

As commented in Remark 3.4.2, sharper bounds like ones by (3.101) should be used. They are also included in Figure 3.4. We can see that the bounds (3.101) correctly reflect the speed of convergence, but they are bigger than the corresponding errors by several orders of magnitudes.

Example 3.5.4. In this example, we test the effectiveness of the residual bound $\delta_k^{\mathrm{QEPmin}}$ in (3.67). We use the same test problem as in Example 3.5.1 for both $\beta = 100$ and $\beta = 1000$. We run our Lanczos algorithm for QEPmin (3.19) and record the residual $\mathrm{NRes}_k^{\mathrm{QEPmin}}$ and its bound $\delta_k^{\mathrm{QEPmin}}$ defined in (3.67) for every Lanczos step. They are plotted in Figure 3.5. We observe that both $\mathrm{NRes}_k^{\mathrm{QEPmin}}$ and $\delta_k^{\mathrm{QEPmin}}$ in (3.67) converge to 0 at the same rate, suggesting $\delta_k^{\mathrm{QEPmin}}$ is an very effective upper bound of the residual $\mathrm{NRes}_k^{\mathrm{QEPmin}}$.

3.6 Summary

According to our theory, solving CRQopt (3.1) is equivalent to solving LGopt (3.14) and QEPmin (3.19), and Lanczos algorithm is suitable for solving LGopt (3.14) and QEPmin (3.19). We give a convergence analysis of the Lanczos algorithm. Numerical examples show the correctness of the Lanczos algorithm and the sharpness of the bound.

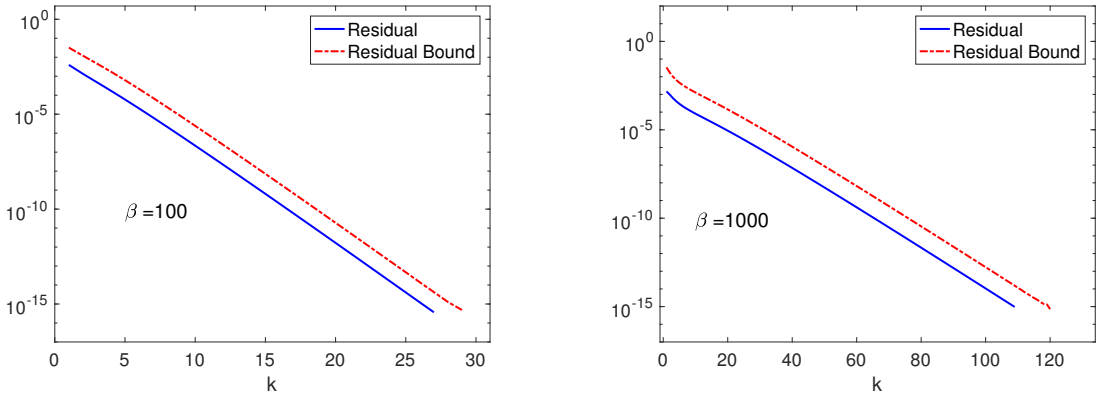

Figure 3.5: Example 3.5.4: relative residual of QEP $\mathrm{NRes}_k^{\mathrm{QEPmin}}$ and the bound of the relative residual $\delta_k^{\mathrm{QEPmin}}$ for the case where $\beta = 100$ (left) and $\beta = 1000$ (right).

Chapter 4

Application in constrained clustering

In this section, we use semi-supervised learning for clustering as an application of CRQopt (3.1). We first discuss unconstrained clustering in Section 4.1 and then discuss a new model for constrained clustering in Section 4.2. We show the experimental settings in Section 4.3 and numerical experiments are shown in Section 4.3.

4.1 Unconstrained clustering

Clustering is an important technique for data analysis and is widely used in machine learning [30, Chapter 14.5.3], bioinformatics [53], social science [44] and image analysis [58]. Clustering uses some similarity metric to group data into different categories. In this section, we discuss the normalized cut, a spectral clustering method that are popular for image segmentation [58, 66].

Given an undirected graph $G = (\mathcal{V}, \mathcal{E})$ whose edge weights are represented by an affinity matrix $W = [w_{ij}]$, we define the *cut* of a partition on its vertices $\mathcal{V}$ into two disjoint sets $\mathcal{A}$ and $\mathcal{B}$, i.e., $\mathcal{A} \cup \mathcal{B} = \mathcal{V}$, $\mathcal{A} \cap \mathcal{B} = \emptyset$ as

$$\text{cut}(\mathcal{A}, \mathcal{B}) = \sum_{i \in \mathcal{A}, j \in \mathcal{B}} w_{ij}. \tag{4.1}$$

Intuitively one would minimize the cut to achieve an optimal bipartition of the graph G, but it often results in a partition $(\mathcal{A}, \mathcal{B})$ with one of them containing only a few isolated vertices in the graph while the other containing the rest. Such a bipartition is not balanced and not useful in practice. To avoid such an unnatural bias that leads to small sets of isolated vertices, the following

normalized cut [58] is introduced:

$$\mathrm{Ncut}(\mathcal{A},\mathcal{B}) = \frac{\mathrm{cut}(\mathcal{A},\mathcal{B})}{\mathrm{vol}(\mathcal{A})} + \frac{\mathrm{cut}(\mathcal{A},\mathcal{B})}{\mathrm{vol}(\mathcal{B})}, \tag{4.2}$$

where

$$\mathrm{vol}(\mathcal{A}) = \sum_{i\in\mathcal{A}, j\in\mathcal{V}} w_{ij} \quad \text{and} \quad \mathrm{vol}(\mathcal{B}) = \sum_{i\in\mathcal{B}, j\in\mathcal{V}} w_{ij}.$$

It turns out that minimizing $\mathrm{Ncut}(\mathcal{A},\mathcal{B})$ usually yields a more balanced bipartition. Let

$$c_+ = \sqrt{\frac{\mathrm{vol}(\mathcal{B})}{\mathrm{vol}(\mathcal{A})\cdot\mathrm{vol}(\mathcal{V})}} \quad \text{and} \quad c_- = -\sqrt{\frac{\mathrm{vol}(\mathcal{A})}{\mathrm{vol}(\mathcal{B})\cdot\mathrm{vol}(\mathcal{V})}},$$

and $x \in \mathbb{R}^n$ ($n = |\mathcal{V}|$, the cardinality of $\mathcal{V}$) be the indicator vector for bipartition $(\mathcal{A},\mathcal{B})$, i.e.,

$$x_{(i)} = \begin{cases} c_+, & i \in \mathcal{A}, \\ c_-, & i \in \mathcal{B}, \end{cases} \tag{4.3}$$

and D be a diagonal matrix with the row sums of W on the diagonal, i.e., $D = \mathrm{diag}(W\mathbf{1})$. Then it can be verified that

$$\mathrm{Ncut}(\mathcal{A},\mathcal{B}) = x^{\mathrm{T}}(D-W)x, \quad x^{\mathrm{T}}Dx = 1, \quad (Dx)^{\mathrm{T}}\mathbf{1} = 0,$$

where $\mathbf{1}$ is a vector of ones. Therefore in order to minimize $\mathrm{Ncut}(\mathcal{A},\mathcal{B})$, we will solve the following combinatorial optimization problem

$$\begin{cases} \min\ x^{\mathrm{T}}(D-W)x, & \text{(4.4a)} \\ \text{s.t. } x_{(i)} \in \{c_+, c_-\}, & \text{(4.4b)} \\ \quad x^{\mathrm{T}}Dx = 1, & \text{(4.4c)} \\ \quad (Dx)^{\mathrm{T}}\mathbf{1} = 0. & \text{(4.4d)} \end{cases}$$

However, the problem (4.4) is a discrete optimization problem and known to be NP-complete. A common practice to make it numerical feasible is to relax x to a real vector and solve instead the following optimization problem

$$\begin{cases} \min\ x^{\mathrm{T}}(D-W)x, & \text{(4.5a)} \\ \text{s.t. } x^{\mathrm{T}}Dx = 1, & \text{(4.5b)} \\ \quad (Dx)^{\mathrm{T}}\mathbf{1} = 0, & \text{(4.5c)} \\ \quad x \in \mathbb{R}^n. & \text{(4.5d)} \end{cases}$$

Under the assumption that D is positive definite, by the Courant-Fisher variational principle [20, Sec 8.1.1], solving (4.5) is equivalent to finding the eigenvector x corresponding to the second smallest eigenvalue of the generalized symmetric definite eigenproblem

$$(D - W)x = \lambda Dx.$$

Note that the setting here is different from the one in [58], where the indicator vector $x_{(i)} \in \{1, -b\}$ and $b = \frac{\mathrm{vol}(\mathcal{A})}{\mathrm{vol}(\mathcal{B})}$. Instead of minimizing a quotient of two quadratic functions in [58], we use the constraint that $x^{\mathrm{T}}Dx = 1$. The model (4.4) is similar to the one in [66, section 5.1], where they use the number of vertices in the sets $\mathcal{A}$ and $\mathcal{B}$ instead of the volumes. The model (4.4) is derived in a similar way to the derivation in [66, section 5.1].

4.2 Constrained clustering

When partial grouping information is known in advance, we can use partial grouping information to set up different models for better clustering. These models are known as constrained clustering. Existing methods for constrained spectral clustering includes implicitly incorporating the constraints into Laplacians [9, 33] and imposing the constraints in linear forms [14, 68, 69] or bilinear forms [67].

We encode the partial grouping information into a linear constraint, which can be either homogeneous [69] or nonhomogeneous [14, 68]. In [14], the authors set up a model where the objective function is the quotient of two quadratic functions and used hard coding for the known associations of pixels to specific classes in terms of linear constraints. In [68], the authors used a model for which the objective function is quadratic and encoded known labels by linear constraints. This is an approach that we take to set up the model.

Let $\mathcal{I} = \{i_1, \cdots, i_\ell\}$ be the index set for which we have the prior information such as $\mathcal{I} \subseteq \mathcal{A}$. According to (4.3), we set $x_{(i)} = c_+$ for $i \in \mathcal{I}$. Similarly, let $\mathcal{J} = \{j_1, \cdots, j_k\}$ be the index set for which we have the prior information that $\mathcal{J} \subseteq \mathcal{B}$, and we set $x_{(j)} = c_-$ for $j \in \mathcal{J}$. This leads

to the following discrete constrained normalized cut problem

$$\begin{cases} \min\ x^{\mathrm{T}}(D-W)x, & (4.6a)\\ \text{s.t. } x_{(i)} \in \{c_+,\, c_-\}\,, & (4.6b)\\ \quad x^{\mathrm{T}}Dx = 1, & (4.6c)\\ \quad (Dx)^{\mathrm{T}}\mathbf{1} = 0, & (4.6d)\\ \quad x_{(i)} = c_+ \quad \text{for} \quad i \in \mathcal{I}, & (4.6e)\\ \quad x_{(i)} = c_- \quad \text{for} \quad i \in \mathcal{J}. & (4.6f) \end{cases}$$

However, there are two imminent issues associated with the model (4.6):

1. the combinatorial optimization (4.6) is NP-hard;
2. the model is incomplete because to calculate c_+ and c_- we need to know $\mathrm{vol}(\mathcal{A})$ and $\mathrm{vol}(\mathcal{B})$, which are unknown before the clustering.

Common workarounds, which we use, are as follows. For the first issue, we relax the model (4.6) by allowing x to be a real vector, i.e., $x \in \mathbb{R}^n$. For the second issue, we use $\frac{\mathrm{vol}(\mathcal{J})}{\mathrm{vol}(\mathcal{I})}$ as an estimate of $\frac{\mathrm{vol}(\mathcal{B})}{\mathrm{vol}(\mathcal{A})}$ to get

$$c_+ \approx \widehat{c}_+ = \sqrt{\frac{\mathrm{vol}(\mathcal{J})}{\mathrm{vol}(\mathcal{I})\cdot\mathrm{vol}(\mathcal{V})}},\ c_- \approx \widehat{c}_- = -\sqrt{\frac{\mathrm{vol}(\mathcal{I})}{\mathrm{vol}(\mathcal{J})\cdot\mathrm{vol}(\mathcal{V})}}.$$

By these relaxation, we reach a computational feasible model:

$$\begin{cases} \min\ x^{\mathrm{T}}(D-W)x, & (4.7a)\\ \text{s.t. } x^{\mathrm{T}}Dx = 1, & (4.7b)\\ \quad (Dx)^{\mathrm{T}}\mathbf{1} = 0, & (4.7c)\\ \quad x_{(i)} = \widehat{c}_+, \quad i \in \mathcal{I}, & (4.7d)\\ \quad x_{(i)} = \widehat{c}_-. \quad i \in \mathcal{J}, & (4.7e) \end{cases}$$

The last three equations are linear constraints and can be collectively written as a linear system of equations:

$$N^{\mathrm{T}}x = b.$$

Let $v = D^{1/2}x$, and define

$$A = D^{-1/2}(D - W)D^{-1/2} \quad \text{and} \quad C = D^{1/2}N.$$

Then the optimization problem (4.7) is turned into CRQopt (3.1) with matrices A, C and b just defined.

4.3 Numerical results

Experimental setting. For a grayscale image, we can construct a weighted graph $G = (\mathcal{V}, \mathcal{E})$ by taking each pixel as a node and connecting each pair (i, j) of pixel i and j by an edge with a weight given by

$$w_{ij} = e^{-\frac{\|F(i)-F(j)\|_2^2}{\delta_F}} \times \begin{cases} 1 & \text{if } \|X(i) - X(j)\|_\infty < r, \\ 0 & \text{otherwise,} \end{cases} \tag{4.8}$$

where δ_F and r are chosen parameters, F is the brightness value and X is the location of a pixel [58].[1] In our experiment, we take

$$\delta_F = \delta \max_{i,j} \|F(i) - F(j)\|_2^2$$

for some parameter δ to be specified.

The definition of weight in (4.8) ensures that every pixel is connected with an edge to at most $(2r + 1)^2$ other pixels. As shown in Table 4.1, in our experiments, r is taken either 5 or 10, and thus the weight matrix W is sparse, which in turn makes the matrix A in CRQopt (3.1) sparse, too. Note that for the example Crab, the contrast between the upper right of the object and the background is not significant. Therefore, we choose r to be twice as much as other examples to ensure the weight matrix correctly reflect the connectivity of the graph. In addition, in our experiments δ is around 0.1, to be consistent with the statement in [58] that "δ_F is typically set to 10 to 20 percent of the total range of the feature distance function". Besides, size m of linear constraints is relatively small compared with the number of pixels n, yielding CRQopt (3.1) with $m \ll n$.

[1] In a 2-D image, pixel i may naturally be represented by (i_x, i_y) where i_x and i_y are two integers.

Table 4.1: The number of pixels n, parameters δ and r and size m of linear constraints.

Image	Number of pixels n	δ	r	m
Flower	30,000	0.1	5	24
Road	50,268	0.1	5	46
Crab	143,000	0.1	10	32
Camel	240,057	0.08	5	24
Dog	395,520	0.1	5	33
Face1	562,500	0.1	5	31
Face2	922,560	0.1	5	19
Daisy	1,024,000	0.08	5	29
Daisy2	1,024,000	0.08	5	59

All experiments were conducted on a PC with Intel Core i7-4770K CPU@3.5GHz and 16-GB RAM. CRQopt (3.1) is solved via solving QEPmin (3.19). In our tests, we choose the maximum Lanczos steps $\texttt{maxit} = 300$ and use $\delta_k^{\text{QEPmin}} < 8\times 10^{-5}$ as the stopping criterion. Besides, we choose the minimum Lanczos steps $\texttt{minit} = 120$ and check the stopping conditions every 5 Lanczos steps to reduce the cost of checking the stopping conditions.

Quality of the model. We apply the model (4.7) and Lanczos algorithm for CRQopt (3.1) on different kinds of images and show the results for segmentation and the computed eigenvector in Figure 4.1. We can see that the image cut results of the model (4.7) indeed agree with our natural visual separation of the object and the background. Daisy and Daisy2 are the same image but with two different ways of prior partial labeling. For both ways of prior partial labeling, the computed image cuts look equally well. Table 4.2 displays the wall-clock runtime and the numbers of Lanczos steps used for the images.

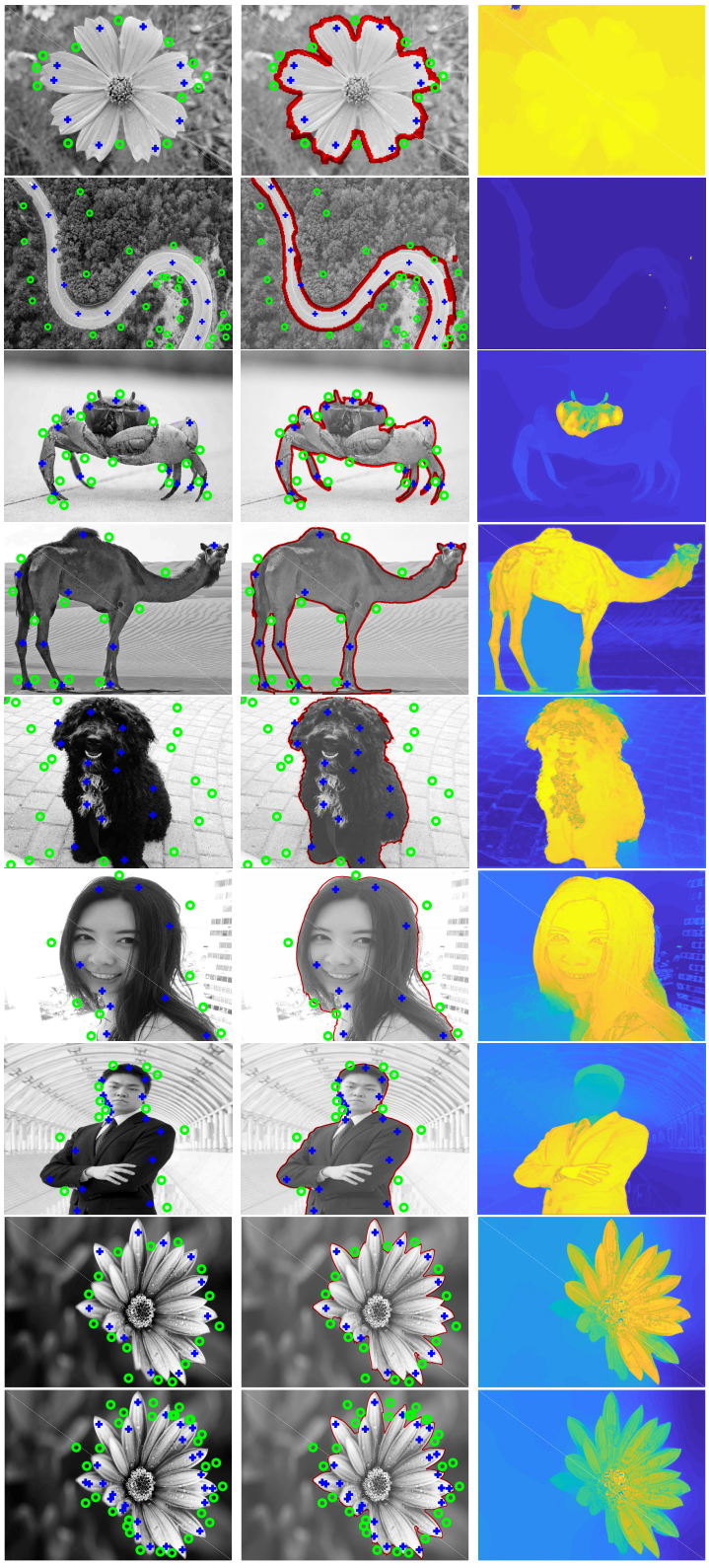

Figure 4.1: The left, middle and right columns are labels, results of image cut and the heat maps of the solutions by the Lanczos algorithm for CRQopt, respectively. Images from top to bottom are Flower, Road, Crab, Camel, Dog, Face1, Face2, Daisy and Daisy2, respectively.

Table 4.2: Runtime (in seconds) and number of Lanczos steps

Image	Run Time	Lanczos steps
Flower	4.61	210
Road	14.92	200
Crab	21.58	135
Camel	31.12	300
Dog	22.33	135
Face1	67.46	215
Face2	35.54	165
Daisy	84.09	235
Daisy2	105.80	245

The purpose of experiments on Daisy and Daisy2 which are the same images but with two different ways of labeling is to observe how the size m of the linear constraints may affect running time. Daisy has 29 linear constraints while Daisy2 has 59. As shown in Table 4.2, the Lanczos algorithm took 84.09 seconds for Daisy and 105.80 seconds for Daisy2, suggesting the larger m is, the more times the Lanczos algorithm needs, as expected, to solve the associated CQRopt. This is because matrix-vector product Px does more work as m increases.

In Table 4.3, we show the running time for Fast-GE-2.0 [33], projected power method [68], and the Lanczos algorithm for a few examples. For comparable segmentation quality, the runtime of the Lanczos algorithm for CRQopt (3.1) is significantly less than the existing methods, including Fast-GE-2.0 and the projected power method. For example, with the same prior labeling on the image Crab, Fast-GE-2.0 and the projected power method take 47.13 seconds and 446.76 seconds, respectively, while our Lanczos algorithm only takes 21.18 seconds. Again, with the same labeling on Daisy and Daisy2, Fast-GE-2.0 takes 1572.81 and seconds 1319.58 seconds, respectively, the projected power method fails to converge in three hours, while the Lanczos algorithm only takes 84.09 seconds and 105.80 seconds, respectively.

Table 4.3: Runtime for Fast-GE-2.0, projected power method and the Lanczos algorithm

Image	Fast-GE-2.0	Projected Power Method	Lanczos algorithm
Crab	47.13 s	446.76 s	21.58 s
Daisy	1572.81 s	3+ hours	84.09 s
Daisy2	1319.58 s	3+ hours	105.80 s

Chapter 5

Padé approximate linearization algorithm

In Section 5.1, we define a nonlinear eigenvalue problem. We show the spectral transformation, rational approximation of the nonlinear eigenvalue, trimmed linearization in Sections 5.2, 5.3 and 5.4, respectively. Then we show our Padé approximate linearization (PAL) algorithm in Section 5.5. Implementation issues are shown in Section 5.6.

5.1 Problem statement

We consider the nonlinear eigenvalue problem (NEP) given by

$$\mathcal{T}(\lambda)x \equiv \left(K - \lambda M + \sum_{j=1}^{q} f_j(\lambda) W_j \right) x = 0 \tag{5.1}$$

where $K, M, W_j \in \mathbb{R}^{n\times n}$ for all j and each $f_j : \mathbb{C} \to \mathbb{C}$ is assumed to be analytic. We are interested in computing eigenvalues in a disk $\Omega(\theta, \gamma) = \{z \in \mathbb{C} \,|\, |z - \theta| \leq \gamma\}$ with given center θ and γ in complex plane. We assume that

1. Matrices K and M are large and sparse and the pencil $K - \lambda M$ can be easily factorized into an LU decomposition. In practice, it is often that K and M are symmetric, and M is positive definite, the LDL^{T}-decomposition of the pencil $K - \lambda M$ is available.

2. The matrices W_j are of rank $r_j \ll n$, and the rank-revealing factorizations $W_j = E_j F_j^{\mathrm{T}}$ are readily available, where $E_j, F_j \in \mathbb{R}^{n \times r_j}$.

Applications. Examples for the NEP (5.1) include rational eigenvalue problems (REP) [61] where $f_j(\lambda)$ are rational functions, particle in a canyon [27] where $K = I$ and $f_j(\lambda) = e^{\mathtt{i}\sqrt{m(\lambda - a_j)}}$, the delay eigenvalue problem [32] with low rank where $M = -I$, $q = 1$ and $f_1(\lambda) = e^{-\tau\lambda}$ and the gun problem [40] where $q = 2$ and $f_j(\lambda) = \mathtt{i}\sqrt{\lambda - \sigma_j^2}$.

5.2 Spectral transformation

Let

$$\mu = \lambda - \alpha, \tag{5.2}$$

where α is a chosen expansion point, then (λ, x) is an eigenpair of the NEP (5.1) if and only if (μ, x) is an eigenpair of

$$\widehat{\mathcal{T}}(\mu)x = \left(K_\alpha - \mu M + \sum_{j=1}^{q} \widehat{f}_j(\mu) W_j \right) x = 0, \tag{5.3}$$

where $K_\alpha = K - \alpha M$ and $\widehat{f}_j(\mu) = f_j(\mu + \alpha)$. Consequently, finding eigenvalues of $\mathcal{T}(\lambda)$ in the disk $\Omega(\theta, r)$ in complex plane is equivalent to finding eigenvalues of $\widehat{\mathcal{T}}(\mu)$ in the disk $\Omega(s, r)$, where $s = \theta - \alpha$.

By the spectral transformation (5.2), applying rational approximants of $f_j(\lambda)$ at expansion point α is equivalent to applying rational approximants of $\widehat{f}_j(\mu)$ at $\mu = 0$. Let a rational approximant to the functions $\widehat{f}_j(\mu)$ be $r_{[n_j,m_j]}(\mu)$ and the error of approximation be $e_{m_j}(\mu) = \widehat{f}_j(\mu) - r_{[n_j,m_j]}(\mu)$, In Sections 6.1 and 6.2, we discuss how to choose α to reduce the error of rational approximations on $\Omega(s, r)$.

5.3 Rational approximation

Applying the approximants $r_{[n_j,m_j]}(\mu)$ to the functions $\widehat{f}_j(\mu)$ in $\widehat{\mathcal{T}}(\mu)$, it is then turned into the following rational eigenvalue problem (REP):

$$\mathcal{R}(\mu)x \equiv \left(K_\alpha - \mu M + \sum_{j=1}^{q} r_{[n_j,m_j]}(\mu) W_j \right) x = 0. \tag{5.4}$$

In general, a realization form of $r_{[n_j,m_j]}(\mu)$ is

$$r_{[n_j,m_j]}(\mu) = s_j(\mu) - a_{m_j}^{\mathrm{T}}(C_{m_j} - \mu D_{m_j})^{-1} b_{m_j}, \tag{5.5}$$

where $s_j(\mu)$ is a polynomial, and $a_{m_j}, b_{m_j} \in \mathbb{C}^{m_j}$, $C_{m_j}, D_{m_j} \in \mathbb{C}^{m_j \times m_j}$. For simplicity, in the rest of the discussion we only consider the case where $s_j(\mu)$ is linear with complex coefficients, i.e.,

$$r_{[m_j+1,m_j]}(\mu) = \gamma_{m_j} + \omega_{m_j}\mu - a_{m_j}^{\mathrm{T}}(C_{m_j} - \mu D_{m_j})^{-1} b_{m_j}, \tag{5.6}$$

where $\gamma_{m_j}, \omega_{m_j} \in \mathbb{C}$. The treatment for linearization of REP (5.5) with higher degree polynomial of $s_j(\mu)$ can be found in [61].

By the rational approximation (5.6) and the rank-revealing factorization $W_j = E_j F_j^{\mathrm{T}}$, the jth rational term in (5.4) can be written as

$$\begin{aligned}
r_{[m_j+1,m_j]}(\mu) W_j &= -a_{m_j}^{\mathrm{T}}(C_{m_j} - \mu D_{m_j})^{-1} b_{m_j} W_j + \gamma_{m_j} W_j + \omega_{m_j}\mu W_j \\
&= -a_{m_j}^{\mathrm{T}}(C_{m_j} - \mu D_{m_j})^{-1} b_{m_j} E_j F_j^{\mathrm{T}} + \gamma_{m_j} E_j F_j^{\mathrm{T}} + \omega_{m_j}\mu E_j F_j^{\mathrm{T}} \\
&= -E_j \cdot a_{m_j}^{\mathrm{T}}(C_{m_j} - \mu D_{m_j})^{-1} b_{m_j} I_{r_j} \cdot F_j^{\mathrm{T}} + E_j \cdot \gamma_{m_j} I_{r_j} \cdot F_j^{\mathrm{T}} + \mu E_j \cdot \omega_{m_j} I_{r_j} \cdot F_j^{\mathrm{T}} \\
&= -E_j (I_{r_j} \otimes a_{m_j})^{\mathrm{T}} (I_{r_j} \otimes C_{m_j} - \mu I_{r_j} \otimes D_{m_j})^{-1} (I_{r_j} \otimes b_{m_j}) F_j^{\mathrm{T}} + \\
&\quad E_j (\gamma_{m_j} I_{r_j}) F_j^{\mathrm{T}} + \mu E_j (\omega_{m_j} I_{r_j}) F_j^{\mathrm{T}},
\end{aligned}$$

where $\otimes$ is the Kronecker product.

Define

$$\begin{aligned}
E &= \begin{bmatrix} E_1 & E_2 & \dots & E_q \end{bmatrix} \in \mathbb{R}^{n \times r}, \\
\widehat{E} &= E \begin{bmatrix} I_{r_1} \otimes a_{m_1}^{\mathrm{T}} & I_{r_2} \otimes a_{m_2}^{\mathrm{T}} & \dots & I_{r_q} \otimes a_{m_q}^{\mathrm{T}} \end{bmatrix} \in \mathbb{C}^{n \times p}, \\
\widehat{C} &= \mathrm{diag}(I_{r_1} \otimes C_{m_1},\ I_{r_2} \otimes C_{m_2},\ \dots,\ I_{r_q} \otimes C_{m_q}) \in \mathbb{C}^{p \times p}, \\
\widehat{D} &= \mathrm{diag}(I_{r_1} \otimes D_{m_1},\ I_{r_2} \otimes D_{m_2},\ \dots,\ I_{r_q} \otimes D_{m_q}) \in \mathbb{C}^{p \times p}, \\
F &= \begin{bmatrix} F_1 & F_2 & \dots & F_q \end{bmatrix} \in \mathbb{R}^{n \times r}, \\
\widehat{F} &= F \begin{bmatrix} I_{r_1} \otimes b_{m_1}^{\mathrm{T}} & I_{r_2} \otimes b_{m_2}^{\mathrm{T}} & \dots & I_{r_q} \otimes b_{m_q}^{\mathrm{T}} \end{bmatrix} \in \mathbb{C}^{n \times p},
\end{aligned} \tag{5.7}$$

where $r = r_1 + r_2 + \cdots + r_q$ and $p = r_1 m_1 + r_2 m_2 + \cdots + r_q m_q$. Then the REP (5.4) can be written in the compact form

$$\mathcal{R}(\mu)x = \left(\widehat{K} - \mu\widehat{M} - \widehat{E}(\widehat{C} - \mu\widehat{D})^{-1}\widehat{F}^{\mathrm{T}}\right) x = 0, \tag{5.8}$$

where

$$\widehat{K}_\alpha = K_\alpha + E\Gamma F^{\mathrm{T}}, \quad \widehat{M} = M - E\Omega F^{\mathrm{T}},$$

and

$$\Gamma = \mathrm{diag}(\gamma_{m_1} I_{r_1}, \ldots, \gamma_{m_q} I_{r_q}) \quad \Omega = \mathrm{diag}(\omega_{m_1} I_{r_1}, \ldots, \omega_{m_q} I_{r_q}).$$

5.4 Trimmed linearization and LEP

Applying the trimmed linearization technique presented in [61], we can convert the REP (5.8) to the following linear eigenvalue problem (LEP) of dimension $N_L = n + p$:

$$\mathcal{L}(\mu)\mathbf{v} \equiv (\mathcal{A} - \mu\mathcal{B})v = 0 \tag{5.9}$$

where

$$\mathcal{A} = \begin{array}{c} \\ n \\ p \end{array} \begin{array}{c} \begin{array}{cc} n & p \end{array} \\ \begin{bmatrix} \widehat{K}_\alpha & \widehat{E} \\ \widehat{F}^{\mathrm{T}} & \widehat{C} \end{bmatrix} \end{array}, \quad \mathcal{B} = \begin{array}{c} \\ n \\ p \end{array} \begin{array}{c} \begin{array}{cc} n & p \end{array} \\ \begin{bmatrix} \widehat{M} & \\ & \widehat{D} \end{bmatrix} \end{array} \quad \text{and} \quad v = \begin{array}{c} \\ n \\ p \end{array} \begin{array}{c} \begin{array}{c} 1 \end{array} \\ \begin{bmatrix} x \\ -(\widehat{C} - \mu\widehat{D})^{-1}\widehat{F}^{\mathrm{T}}x \end{bmatrix} \end{array}. \tag{5.10}$$

The relationship between the eigenpairs of the REP (5.4) and those of the LEP (5.9) is given in the following theorem.

Theorem 5.4.1 ([61])**.**

(a) If μ is an eigenvalue of the REP (5.8) then it is also an eigenvalue of the LEP (5.9).

(b) If (μ, v) is an eigenpair of of the LEP (5.9), μ is not a pole of the REP (5.8), and $v_{(1:n)} \neq 0$, then $(\mu, v_{(1:n)})$ is an eigenpair of the REP (5.8).

By Theorem 5.4.1, finding eigenvalues of the REP (5.8) in $\Omega(s, r)$ is equivalent to finding eigenvalues of the LEP (5.9) in $\Omega(s, r)$ and these eigenvalues are not poles of rational functions $r_{[m_j+1,m_j]}(\mu)$. Several remarks are in order:

1. In practice, we first find a sufficiently large number of eigenvalues of the LEP (5.9) nearest to s in module and remove the eigenvalues outside $\Omega(s, r)$ or near poles of $r_{[m_j+1,m_j]}(\mu)$.

2. For computing eigenvalues of the LEP (5.9) nearest to s, we apply a shift-invert Arnoldi method [56, Sec.8.1.3]. Alternative methods for computing eigenvalues of large matrices nearest to s include polynomial filtering [13] and Jacobi-Davidson method [2, Sec.7.12.3].

3. When $\widehat{K}$, $\widehat{M}$ and C, D are real symmetric matrices and $\widehat{E} = \widehat{F}$, $\mathcal{A}$ and $\mathcal{B}$ are real symmetric matrices. Specifically, if W_j are symmetric positive semi-definite, we can compute decomposition $W_j = E_j F_j^{\mathrm{T}}$ such that $E_j = F_j$. In this case $\widehat{E} = \widehat{F}$ when $a_{m_j} = b_{m_j}$. When C, D are real symmetric matrices, $\mathcal{A}$, $\mathcal{B}$ are real and symmetric.

5.5 PAL algorithm

Algorithm 5 gives a description of the Padé Approximate Linearization (PAL) algorithm for finding eigenpairs of NEPs of the form (5.1) in $\Omega(\theta, r)$.

Algorithm 5 PAL algorithm

Input: Matrices K, M, W_j for $j = 1, 2, \cdots, q$, the center θ and the radius r of the target region Ω of NEP (5.1), the number of desired eigenpairs `neig`, the expansion point α and orders m_j of the Padé approximants $r_{[m_j+1,m_j]}(\mu)$ of $\widehat{f}_j(\mu) = f_j(\mu + \alpha)$;

Output: The computed eigenvalues, eigenvectors and the relative residual errors;

1: Compute the rank-revealing factorizations $W_j = E_j F_j^{\mathrm{T}}$;

2: Compute the LU factorization of $K - \theta M$;

3: Compute N eigenvalues nearest to $s = \theta - \alpha$ and corresponding eigenvectors (μ, v) of the LEP (5.9), where $N \geq$ `neig`.

4: Remove any values μ which fall near the poles of any $r_{[m_j+1,m_j]}(\mu)$ or outside $\Omega(s, r)$.

5: Compute the approximate eigenpairs $(\lambda, x) = (\mu + \alpha, v_{(1:n)})$ of the NEP (5.1) and the relative residual errors.

5.6 Implementation issues

5.6.1 Matrix-vector multiplications

Since we have transformed the problem of finding a few eigenvalues of NEP (5.1) in $\Omega(\theta, r)$ to finding a few eigenvalues of LEP (5.9) nearest to $s = \theta - \alpha$ in module. We apply the shift-invert Arnoldi method [56, Sec.8.1.3].

For applying the shift-invert Arnoldi method, it is necessary to provide a method for computing the matrix-vector product

$$v = (\mathcal{A} - s\mathcal{B})^{-1}\mathcal{B}u. \tag{5.11}$$

We now describe a way to efficiently compute the vector v by exploiting the matrix structure. First, by the factorization,

$$\mathcal{A} - s\mathcal{B} = \begin{bmatrix} \widehat{K}_\alpha - s\widehat{M} & \widehat{E} \\ \widehat{F}^{\mathrm{T}} & G \end{bmatrix} = \begin{bmatrix} I_n & \widehat{E} \\ & G \end{bmatrix} \begin{bmatrix} H & \\ & G^{-1} \end{bmatrix} \begin{bmatrix} I_n & \\ \widehat{F}^{\mathrm{T}} & G \end{bmatrix},$$

the inverse of $\mathcal{A} - s\mathcal{B}$ is given by

$$(\mathcal{A} - s\mathcal{B})^{-1} = \begin{bmatrix} I_n & \\ -G^{-1}\widehat{F}^{\mathrm{T}} & G^{-1} \end{bmatrix} \begin{bmatrix} H^{-1} & \\ & G \end{bmatrix} \begin{bmatrix} I_n & -\widehat{E}G^{-1} \\ & G^{-1} \end{bmatrix},$$

where

$$G = \widehat{C} - s\widehat{D} \quad \text{and} \quad H = \widehat{K}_\alpha - s\widehat{M} - \widehat{E}G^{-1}\widehat{F}^{\mathrm{T}}.$$

Let $v = [v_1^{\mathrm{T}}\ v_2^{\mathrm{T}}]^{\mathrm{T}}$ and $u = [u_1^{\mathrm{T}}\ u_2^{\mathrm{T}}]^{\mathrm{T}}$, where $u_1, v_1 \in \mathbb{R}^n$ and $u_2, v_2 \in \mathbb{R}^p$, then the matrix-vector product (5.11) can be expressed as follows:

$$\begin{aligned} v_1 &= H^{-1}\left(\widehat{M}u_1 - \widehat{E}G^{-1}(\widehat{D}u_2)\right), \\ v_2 &= G^{-1}(\widehat{D}u_2) - G^{-1}(\widehat{F}^{\mathrm{T}}v_1). \end{aligned} \tag{5.12}$$

now let us discuss how to efficiently compute the submatrix-vector multiplications in (5.12) by exploiting the rich structure of the submatrices $\widehat{M}$, $\widehat{E}$, $\widehat{F}$, G and H.

- The matrix-vector product $\widehat{M}x$ can be computed by exploiting the sparse plus low-rank structure of $\widehat{M}$ defined in (5.8) as follows:

$$\widehat{M}x = (M - E\Omega F^{\mathrm{T}})x = Mx - \sum_{j=1}^{q} \omega_{m_j} E_j (F_j^{\mathrm{T}} x). \tag{5.13}$$

- By exploiting the Kronecker product of $\widehat{E}$ defined in (5.7), the matrix-vector multiplication $\widehat{E}x$ can be computed by the following expression:

$$\begin{aligned}\widehat{E}x &= [E_1, E_2, \cdots, E_q]([I_{r_1} \otimes a_{m_1}^{\mathrm{T}}, I_{r_2} \otimes a_{m_2}^{\mathrm{T}}, \cdots, I_{r_q} \otimes a_{m_q}^{\mathrm{T}}]x) \\ &= [E_1, E_2, \cdots, E_q][a_{m_1}^{\mathrm{T}} X_1, \cdots, a_{m_q}^{\mathrm{T}} X_q]^{\mathrm{T}} = \sum_{j=1}^{q} E_j (X_j^{\mathrm{T}} a_{m_j})\end{aligned} \tag{5.14}$$

  where $m_j \times r_j$ matrix X_j and is a reshaped matrix from $1 + \sum_{k=1}^{j-1} m_k r_k$ to $\sum_{k=1}^{j} m_k r_k$ entries of the vector x.

- By the property of the Kronecker product, the matrix-vector multiplication $\widehat{F}^{\mathrm{T}}x$ can be computed as follows:

$$\widehat{F}^{\mathrm{T}}x = \begin{bmatrix} (I_{r_1} \otimes b_{m_1})F_1^{\mathrm{T}} \\ (I_{r_2} \otimes b_{m_2})F_2^{\mathrm{T}} \\ \vdots \\ (I_{r_q} \otimes b_{m_q})F_q^{\mathrm{T}} \end{bmatrix} x = \begin{bmatrix} (I_{r_1} \otimes b_{m_1})F_1^{\mathrm{T}}x \\ (I_{r_2} \otimes b_{m_2})F_2^{\mathrm{T}}x \\ \vdots \\ (I_{r_q} \otimes b_{m_q})F_q^{\mathrm{T}}x \end{bmatrix} = \begin{bmatrix} F_1^{\mathrm{T}}x \otimes b_{m_1} \\ F_2^{\mathrm{T}}x \otimes b_{m_2} \\ \vdots \\ F_q^{\mathrm{T}}x \otimes b_{m_q} \end{bmatrix}. \tag{5.15}$$

- For computing $G^{-1}x = (\widehat{C} - s\widehat{D})^{-1}x$, we can pre-generate the LU decomposition of $\widehat{C} - s\widehat{D}$ and then reuse the factorization with different vector x. In the case of diagonal matrices $\widehat{C}$ and $\widehat{D}$, $G^{-1}x = (\widehat{C} - s\widehat{D})^{-1}x$ can be computed componentwisely.

- Recall that one of main assumptions is that only the matrix-vector $(K - \lambda M)^{-1}x$ is available via a factorization of $K - \lambda M$, such as $\mathrm{LDL}^{\mathrm{T}}$ factorization. Therefore, we need to reformulate the matrix H in order to apply this computational kernel for forming $H^{-1}x$. To do so, first note that

$$\begin{aligned} H &= \widehat{K}_\alpha - s\widehat{M} - \widehat{E}G^{-1}\widehat{F}^{\mathrm{T}} \\ &= K - (\alpha + s)M + \sum_{j=1}^{q} \left(\gamma_{m_j} + s\omega_{m_j} - a_{m_j}^{\mathrm{T}}(C_{m_j} - sD_{m_j})^{-1}b_{m_j}\right) E_j F_j^{\mathrm{T}} \\ &= K - \theta M + \sum_{j=1}^{q} r_{[m_j+1,m_j]}(s) E_j F_j^{\mathrm{T}} \\ &= K - \theta M + \sum_{j=1}^{q} E_j \cdot r_{[m_j+1,m_j]}(s) I_{r_j} \cdot F_j^{\mathrm{T}} \\ &= K - \theta M + [E_1, \ldots, E_q] \cdot \mathrm{diag}\left(r_{[m_1+1,m_1]}(s)I_{r_1}, \ldots, r_{[m_q+1,m_q]}(s)I_{r_q}\right) \cdot [F_1, \ldots, F_q]^{\mathrm{T}} \\ &\equiv K - \theta M + E\Delta(s)F^{\mathrm{T}} \end{aligned}$$

where $\Delta(s) = \text{diag}\left(r_{[m_1+1,m_1]}(s)I_{r_1}, \ldots, r_{[m_q+1,m_q]}(s)I_{r_q}\right)$.

Now by applying Sherman-Morrison-Woodbury formula [31][1], we have

$$H^{-1}x = y - V\left(\Delta^{-1}(s) + F^{\mathrm{T}}V\right)^{-1}F^{\mathrm{T}}y, \tag{5.16}$$

where $y = (K-\theta M)^{-1}x$ and $V = (K-\theta M)^{-1}E$. We note that $n\times r$ matrix $V = (K-\theta M)^{-1}E$. and a factorization form, say LU of the $r\times r$ matrix $\Delta^{-1}(s)+F^{\mathrm{T}}V$ can be computed in advance. There are tradeoffs in computational efficiency whether we should exploit the sparsity of E for computing V, see numerical examples in Section 6.3.

5.6.2 Real and complex arithmetic

For some applications, the vectors and matrices in the matrix-vector of rational approximation are all real such as the nonlinear function of the delay eigenvalue problem in Example 2.3.4, then all the arithmetic in the matrix-vector multiplications $(\mathcal{A} - s\mathcal{B})^{-1}\mathcal{B}u$ are mutiplying vectors by real matrices, where $\mathcal{A}$ and $\mathcal{B}$ are defined in (5.10).

However, for computing resonant modes of accelerator cavities to be discussed in Section 6, the realization forms of rational approximants are given in (6.3) and (6.11). Comparing with the realization form of $r_{[m+1,m]}(\mu)$ defined in (5.6), the vectors $a_{m_j}, b_{m_j} \in \mathbb{R}^{m_j}$ are real and the matrices $C_{m_j}, D_{m_j} \in \mathbb{C}^{m_j}$ are matrices with pure imaginary elements. In this case, $K - \theta M$, E, F and $V = (K - \theta M)^{-1}E$ in (5.16) are real matrices. Therefore, generating V only involves multiplying real matrices by real vectors. For the matrix-vector multiplication $(\mathcal{A}-s\mathcal{B})^{-1}\mathcal{B}u$, the most expensive part is $H^{-1}x$ defined in (5.16). When computing $H^{-1}x$ after V is generated, matrices F and V are real. But the typically small matrix $\left(\Delta^{-1}(s) + F^{\mathrm{T}}V\right)^{-1}$ is complex. Therefore, the method for the matrix-vector multiplication $(\mathcal{A} - s\mathcal{B})^{-1}\mathcal{B}u$ makes arithmetic as real as possible.

5.6.3 Rank-revealing factorization

When W_j is extremely sparse, we use the following two-step approach for a rank-revealing factorization. For the simplicity of notation, we use a general matrix $A \in \mathbb{R}^{m\times n}$ to replace W_j.

- **Step 1.** Extracting rows and columns with nonzero elements.

[1] $(A + UCV^{\mathrm{T}})^{-1} = A^{-1} - A^{-1}U(C^{-1} + V^{\mathrm{T}}A^{-1}U)^{-1}V^{\mathrm{T}}A^{-1}$

In many applications, most rows and columns of A are all zeros. In the factorization we only have to consider the rows and columns with nonzero elements. Let $I = \{i \,|\, \exists\, j \ s.t.\ A_{ij} \neq 0\}$ be the row index set such that there is at least one nonzero element in that row and $J = \{j \,|\, \exists\, i \ s.t.\ A_{ij} \neq 0\}$ be the column index set such that there is at least one nonzero element in that column, then let

$$A_1 = A_{(I,J)} \in \mathbb{R}^{k \times l}$$

be the submatrix of A with rows in index set I and columns in index set J. Then we can compute the factorization on A_1 instead of A to save computing time. Suppose $A_1 = E_1 F_1^{\mathrm{T}}$ is a rank-revealing factorization of A_1, where $E_1 \in \mathbb{R}^{k \times r}$ and $F_1 \in \mathbb{R}^{l \times r}$, then let $E \in \mathbb{R}^{m \times r}$, $F \in \mathbb{R}^{n \times r}$ such that $E_{(I,:)} = E_1$ and $F_{(J,:)} = F_1$, $A = EF^{\mathrm{T}}$ is an rank-revealing factorization of A.

- **Step 2.** Rank-revealing factorization of A_1 by SVD.

  Let the SVD of A_1 be $A_1 = UDV^{\mathrm{T}}$, where $D = \mathrm{diag}(\sigma_1, \sigma_2, \cdots, \sigma_r)$ with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$, and $U \in \mathbb{R}^{k \times r}$, $V \in \mathbb{R}^{l \times r}$, then the rank of A_1 is r. Let $E_1 = UD^{1/2}$ and $F_1 = VD^{1/2}$, then $A_1 = E_1 F_1^{\mathrm{T}}$ is rank-revealing factorization. When A_1 is symmetric positive semi-definite, then we have $E_1 = F_1$. Of course, we can use the spectral decomposition to replace the SVD.

If A_1 is too expensive for computing the SVD, computational cheaper but less reliable alternative methods for rank-revealing factorization are rank-revealing QR (RRQR) factorization [6, 50], [10, Sec.3.6], rank-revealing LU (RRLU) factorization [49], even randomized algorithms [29].

Chapter 6

Application in resonant modes computation of accelerator cavity

We provide examples on the application of the NEP stated in Chapter 5 in resonant modes computation of accelerator cavity. We show eigenvalue problems with TE modes only in Section 6.1 and eigenvalue problems with TE and TM modes in Section 6.2. Then we show experiments in Section 6.3.

6.1 Eigenvalue problems with TE modes only

When there are only so-called transverse electric (TE) modes but no transverse magnetic (TM) modes onto the waveguide boundaries accelerator cavities, the finite element discretization leads to the following NEP

$$\mathcal{T}(\lambda)v \equiv \left(K - \lambda M + \sum_{j=1}^{q} \mathtt{i}\sqrt{\lambda - \sigma_j^2}\, W_j \right) x = 0 \tag{6.1}$$

where $K, M, W_j \in \mathbb{R}^{n\times n}$ are symmetric matrices, $\mathtt{i} = \sqrt{-1}$, and the cutoff values $\sigma_j \in \mathbb{R}$. Furthermore each W_j is symmetric and low rank with the rank-revealing factorization $W_j = E_j E_j^{\mathrm{T}}$, $E_j \in \mathbb{R}^{n\times r_j}$ and $r_j \ll n$. The complex square root $\sqrt{\cdot}$ corresponds to the principal branch. We are interested in finding eigenvalues in the upper half of a disk $\Omega(\theta, r)$. We now follow the PAL algorithm developed in Section 5 to solve the NEP (6.1).

Spectral transformation of the NEP (6.1). Let $\alpha \in \mathbb{R} \cap \Omega(\theta, r)$ be an expansion point for Padé approximation of functions $\sqrt{\lambda - \sigma_j^2}$ and $\mu = \lambda - \alpha$, it yields the shifted NEP

$$\widehat{\mathcal{T}}(\mu)x \equiv \left(K_\alpha - \mu M + \sum_{j=1}^{q} \mathtt{i}\widehat{\sigma}_j^{1/2} \sqrt{\frac{\mu}{\widehat{\sigma}_j} + 1}\ W_j \right) x = 0 \tag{6.2}$$

where $K_\alpha = K - \alpha M$ and $\widehat{\sigma}_j = \alpha - \sigma_j^2$. We are supposed to find N eigenvalues of (6.2) in the disk $\Omega(s, r)$, where $s = \theta - \alpha$.

Rational approximation of NEP (6.2). Recall that the realization form of order-(m, m) Padé approximant of $\sqrt{z+1}$ is $r_{[m,m]}(z)$ defined in (2.14).

Let $z = \frac{\mu}{\widehat{\sigma}_j}$, then an order (m_j, m_j) Padé approximation of $\mathtt{i}\widehat{\sigma}_j^{1/2}\sqrt{\frac{\mu}{\widehat{\sigma}_j} + 1}$ is given by

$$\mathtt{i}\widehat{\sigma}_j^{1/2} r_{[m_j,m_j]}\left(\frac{\mu}{\widehat{\sigma}_j}\right) = -a_{m_j}^{\mathrm{T}} \left(-\frac{\mathtt{i}}{\widehat{\sigma}_j^{1/2}} I_{m_j} + \mathtt{i}\frac{\mu}{\widehat{\sigma}_j^{3/2}} D_{m_j} \right)^{-1} a_{m_j} + \mathtt{i}\widehat{\sigma}_j^{1/2}\gamma_{m_j}. \tag{6.3}$$

where a_{m_j}, D_{m_j} and $\gamma_{m_j} = 2m_j + 1$ are defined in (2.15). Note that D_{m_j} are diagonal matrices. and $\mathtt{i}^{-1} = -\mathtt{i}$.

By following the derivation in Section 5.3, an REP of the NEP (6.2) in a compact form is given by

$$\mathcal{R}(\lambda)x \equiv \left(K_\alpha - \mu M + \sum_{j=1}^{q} \mathtt{i}\widehat{\sigma}_j^{1/2}\, r_{[m_j,m_j]}\left(\frac{\mu}{\widehat{\sigma}_j}\right) W_j \right) x \tag{6.4}$$

$$= \left(\widehat{K}_\alpha - \mu M - \widehat{E}(\widehat{C} - \mu\widehat{D})^{-1}\widehat{E}^{\mathrm{T}} \right) x = 0, \tag{6.5}$$

where $\widehat{K}_\alpha = K_\alpha + E\Gamma E^{\mathrm{T}} = K - \alpha M + E\Gamma E^{\mathrm{T}}$ and

$$\begin{aligned}
\Gamma &= \mathrm{diag}(\mathtt{i}\widehat{\sigma}_1^{1/2}\gamma_{m_1} I_{r_1}, \mathtt{i}\widehat{\sigma}_2^{1/2}\gamma_{m_2} I_{r_2}, \ldots, \mathtt{i}\widehat{\sigma}_q^{1/2}\gamma_{m_q} I_{r_q}) \in \mathbb{C}^{r\times r}, \\
E &= \begin{bmatrix} E_1 & E_2 & \ldots & E_q \end{bmatrix} \in \mathbb{R}^{n\times r}, \\
\widehat{E} &= E \begin{bmatrix} I_{r_1} \otimes a_{m_1}^{\mathrm{T}} & I_{r_2} \otimes a_{m_2}^{\mathrm{T}} & \ldots & I_{r_q} \otimes a_{m_q}^{\mathrm{T}} \end{bmatrix} \in \mathbb{R}^{n\times p}, \\
\widehat{C} &= \mathrm{diag}(-\mathtt{i}I_{r_1 m_1}/\widehat{\sigma}_1^{1/2},\ -\mathtt{i}I_{r_2 m_2}/\widehat{\sigma}_2^{1/2},\ \ldots,\ -\mathtt{i}I_{r_q m_q}/\widehat{\sigma}_q^{1/2}) \in \mathbb{C}^{p\times p}, \\
\widehat{D} &= \mathrm{diag}(-\mathtt{i}I_{r_1} \otimes D_{m_1}, /\widehat{\sigma}_1^{3/2}\ -\mathtt{i}I_{r_2} \otimes D_{m_2}/\widehat{\sigma}_2^{3/2},\ \ldots,\ -\mathtt{i}I_{r_q} \otimes D_{m_q}/\widehat{\sigma}_q^{3/2}) \in \mathbb{C}^{p\times p},
\end{aligned}$$

and $r = r_1 + r_2 + \cdots + r_q$ and $p = r_1 m_1 + r_2 m_2 + \cdots + r_q m_q$.

LEP. So far we have transformed NEP (6.1) to REP (6.5). We can follow the method stated in Section 5.4 to linearize REP (6.5) and find N eigenvalues of the corresponding LEP (5.9) nearest to s.

Note that E and $\widehat{E}$ are real matrices and Γ, $\widehat{C}$ and $\widehat{D}$ are pure imaginary matrices. Therefore, as is shown in Section 5.6.2, all matrices involving matrix-vector multiplication $H^{-1}x$ (5.16) except the matrix $\Delta^{-1}(s) + E^{\mathrm{T}}V$ are real. Thus we can keep mostly real arithmetic for matrix-vector multiplications.

Preprocessing: choice of the expansion point α for the spectral transformation. Recall that in Section 5.2, we stated that we choose α to make the error of rational approximation in $\Omega(s,r)$ as small as possible, where $\Omega(s,r)$ is the target region of NEP (6.2). Here we provide a method to choose the expansion point α to reduce the error of rational approximations in $\Omega(s,r)$ based on the error formulas of rational approximations of $\sqrt{z+1}$ stated in Example 2.3.1.

Recall that in (6.4) $\sqrt{\frac{\mu}{\widehat{\sigma}_j}+1}$ is approximated by $r_{[m_j,m_j]}\left(\frac{\mu}{\widehat{\sigma}_j}\right)$. Let the error of approximations be

$$e_{m_j}\left(\frac{\mu}{\widehat{\sigma}_j}\right) = \sqrt{\frac{\mu}{\widehat{\sigma}_j}+1} - r_{[m_j,m_j]}\left(\frac{\mu}{\widehat{\sigma}_j}\right),$$

where the explicit formula for $e_{m_j}(z)$ is defined in (2.16). Different expansion points α cause different errors of approximations in $\Omega(s,r)$. A natural choice of expansion point α is to choose the center of target region, i.e., $\alpha = \theta$. However, this may cause large error on one side of $\Omega(s,r)$. We show some numerical examples in Section 6.3. Note that the left and the right endpoints of $\Omega(s,r)$ are $\theta - r - \alpha$ and $\theta + r - \alpha$, respectively, we want to find $\alpha \in (\theta - r, \theta + r)$ to balance $e(\mu)$ on both endpoints of $\Omega(s,r)$ by solving equation

$$e(\theta + r - \alpha) - e(\theta - r - \alpha) = 0, \tag{6.6}$$

where

$$e(\mu) = \sum_{j=1}^{q} \left| e_{m_j}\left(\frac{\mu}{\widehat{\sigma}_j}\right)\right| = \sum_{j=1}^{q} \left| e_{m_j}\left(\frac{\mu}{\alpha - \sigma_j^2}\right)\right| \tag{6.7}$$

is the total error of rational approximation.

We can show the existence and uniqueness of the root of (6.6) and the root of (6.6) minimizes the maximum error on two sides of $\Omega(s,r)$. Before showing that theorem, we first prove a lemma for the monotonicity of $e(\theta \pm r - \alpha)$.

Lemma 6.1.1. *Let the target region of NEP (6.1) be $\Omega(\theta, r)$, then when all real numbers $\sigma_j^2 \notin \Omega(\theta, r)$, $e(\theta + r - \alpha)$ is strictly decreasing with respect to α and $e(\theta - r - \alpha)$ is strictly increasing with respect to α for $\alpha \in (\theta - r, \theta + r)$.*

Proof. It is sufficient to prove that for any j, $\left|e_{m_j}\left(\frac{\theta+r-\alpha}{\alpha-\sigma_j^2}\right)\right|$ is strictly decreasing with respect to α and $\left|e_{m_j}\left(\frac{\theta-r-\alpha}{\alpha-\sigma_j^2}\right)\right|$ is strictly increasing with respect to α. There are following two cases.

Case $\sigma_j^2 < \theta - r < \alpha < \theta + r$: In this case

$$\frac{\theta + r - \alpha}{\alpha - \sigma_j^2} = -1 + \frac{\theta + r - \sigma_j^2}{\alpha - \sigma_j^2} > -1 + \frac{\alpha - \sigma_j^2}{\alpha - \sigma_j^2} = 0$$

and is strictly decreasing with respect to α. Since $|e_{m_j}(z)|$ is strictly increasing with respect to z for $z > 0$, $\left|e_{m_j}\left(\frac{\theta+r-\alpha}{\alpha-\sigma_j^2}\right)\right|$ is strictly decreasing with respect to α. On the other hand,

$$\frac{\theta - r - \alpha}{\alpha - \sigma_j^2} = -1 + \frac{\theta - r - \sigma_j^2}{\alpha - \sigma_j^2}$$

and

$$-1 < -1 + \frac{\theta - r - \sigma_j^2}{\alpha - \sigma_j^2} < -1 + \frac{\alpha - \sigma_j^2}{\alpha - \sigma_j^2} = 0.$$

Besides, $\frac{\theta-r-\alpha}{\alpha-\sigma_j^2}$ is strictly decreasing with respect to α. Since $|e_{m_j}(z)|$ is strictly decreasing with respect to z for $-1 < z < 0$, $\left|e_{m_j}\left(\frac{\theta-r-\alpha}{\alpha-\sigma_j^2}\right)\right|$ is strictly increasing with respect to α.

Case $\theta - r < \alpha < \theta + r < \sigma_j^2$: In this case

$$\frac{\theta + r - \alpha}{\alpha - \sigma_j^2} = -1 + \frac{\sigma_j^2 - (\theta + r)}{\sigma_j^2 - \alpha}$$

and

$$-1 < -1 + \frac{\sigma_j^2 - (\theta + r)}{\sigma_j^2 - \alpha} < -1 + \frac{\sigma_j^2 - \alpha}{\sigma_j^2 - \alpha} = 0.$$

Besides, $\frac{\theta+r-\alpha}{\alpha-\sigma_j^2}$ is strictly increasing with respect to α. Since $|e_{m_j}(z)|$ is strictly decreasing with respect to z for $-1 < z < 0$, $\left|e_{m_j}\left(\frac{\theta+r-\alpha}{\alpha-\sigma_j^2}\right)\right|$ is strictly decreasing with respect to α. On the other hand,

$$\frac{\theta - r - \alpha}{\alpha - \sigma_j^2} = -1 + \frac{\sigma_j^2 - (\theta - r)}{\sigma_j^2 - \alpha} > -1 + \frac{\sigma_j^2 - \alpha}{\sigma_j^2 - \alpha} = 0$$

and is strictly increasing with respect to α. Since $|e_{m_j}(z)|$ is strictly increasing with respect to z for $z > 0$, $\left|e_{m_j}\left(\frac{\theta-r-\alpha}{\alpha-\sigma_j^2}\right)\right|$ is strictly increasing with respect to α. □

Now we show the theorem.

Theorem 6.1.1. *Let the target region of NEP (6.1) be $\Omega(\theta, r)$, and real numbers $\sigma_j^2 \notin \Omega(\theta, r)$ for all $j = 1, 2, \cdots, p$, then*

1. *Equation (6.6) has a unique root α_0 in the interval $(\theta - r, \theta + r)$.*

2. *Let $\widehat{e}(\alpha) = \max\{e(\theta - r - \alpha), e(\theta + r - \alpha)\}$, then*

$$\alpha_0 = \arg \min_{\alpha \in (\theta - r, \theta + r)} \widehat{e}(\alpha). \tag{6.8}$$

Proof. Note that

$$\lim_{\alpha \to (\theta + r)^-} e(\theta + r - \alpha) = 0 \quad \text{and} \quad \lim_{\alpha \to (\theta + r)^-} e(\theta - r - \alpha) > 0.$$

Therefore

$$\lim_{\alpha \to (\theta + r)^-} e(\theta + r - \alpha) - e(\theta - r - \alpha) < 0.$$

Similarly

$$\lim_{\alpha \to (\theta - r)^+} e(\theta + r - \alpha) - e(\theta - r - \alpha) > 0.$$

Besides, $e(\theta + r - \alpha) - e(\theta - r - \alpha)$ is continuous and strictly decreasing for $\alpha \in (\theta - r, \theta + r)$ by Lemma 6.1.1, so there exists a unique solution of (6.6) for $\alpha \in (\theta - r, \theta + r)$.

Suppose α_0 satisfies (6.6), then let $\widehat{e}(\alpha) = \max\{e(\theta - r - \alpha), e(\theta + r - \alpha)\}$, then

$$\widehat{e}(\alpha) = \begin{cases} e(\theta + r - \alpha), \alpha < \alpha_0 \\ e(\theta - r - \alpha), \alpha \geq \alpha_0 \end{cases}$$

By the definition above and Lemma 6.1.1, $\widehat{e}(\alpha)$ is strictly decreasing in $(\theta - r, \alpha_0)$ and is strictly increasing in $(\alpha_0, \theta + r)$. Therefore, α_0 is the minimizer of $\widehat{e}(\alpha)$. □

When $\sigma_1^2 < \theta - r < \theta + r < \sigma_2^2$ and $\theta - \sigma_1^2 = \sigma_2^2 - \theta$, we have a so-called symmetric case. In this case, σ_1^2 and σ_2^2 are symmetric with respect to the center θ of $\Omega(\theta, r)$. $\alpha_0 = (\sigma_1^2 + \sigma_2^2)/2$ is the root of (6.6). Therefore, α_0 is an optimal expansion point. In general, by the monotonicity of $e(\alpha, \theta + r) - e(\alpha, \theta - r)$ for $\alpha \in (\theta - r, \theta + r)$, we can apply the bisection method to find the root α_0 of (6.6).

Post-processing: poles of rational functions. So far we transformed solving the NEP (6.1) to solving the LEP (5.9). Recall that Theorem 5.4.1 indicates that when the eigenvalue μ of LEP (5.9) is not a pole of REP (6.5), then μ is an eigenvalue of REP (6.5). Therefore, after the eigenvalues are computed, we need to remove the eigenvalues of LEP (5.9) near poles of $r_{[m_j,m_j]}\left(\frac{\mu}{\widehat{\sigma}_j}\right)$. The following theorem shows that if all cutoff frequencies σ_j^2 are not in the target region $\Omega(\theta,r)$ of NEP (6.1) and the expansion point $\alpha \in \mathbb{R}\cap\Omega(\theta,r)$, then the poles of $r_{[m_j,m_j]}\left(\frac{\mu}{\widehat{\sigma}_j}\right)$ are not in target region $\Omega(s,r)$ of REP (6.5) for all $j=1,2,\cdots,q$. In this case it is not required to remove the eigenvalues of LEP (5.9) near the poles of $r_{[m_j,m_j]}\left(\frac{\mu}{\widehat{\sigma}_j}\right)$.

Theorem 6.1.2. *Let the target region of NEP (6.1) be $\Omega(\theta,r)$ and target region of REP (6.5) be $\Omega(s,r)$ with $s=\theta-\alpha$. When real numbers $\sigma_j^2\notin\Omega(\theta,r)$ for all $j=1,2,\cdots,q$ and $\alpha\in\mathbb{R}\cap\Omega(\theta,r)=(\theta-r,\theta+r)$, the poles of rational functions $r_{[m_j,m_j]}\left(\frac{\mu}{\widehat{\sigma}_j}\right)$ are not in $\Omega(s,r)$ for all $j=1,2,\cdots,q$.*

Proof. By (2.13), the poles for $r_{[m_j]}(z)$ are $z_k=-\frac{1}{\xi_k}$, $\xi_k=\cos^2\frac{k\pi}{2m_j+1}$, for $k=1,2,\cdots,m_j$. The poles satisfy $z_k\in\mathbb{R}$ and $z_k<-1$. Then the poles for $r_{[m_j]}\left(\frac{\mu}{\widehat{\sigma}_j}\right)$ are $\mu_k=-\frac{\widehat{\sigma}_j}{\xi_k}=(-\frac{1}{\xi_k})(\alpha-\sigma_j^2), k=1,2,\cdots,m_j$.

If $\alpha>\sigma_j^2$, then $\mu_k<\sigma_j^2-\alpha$ for all $k=1,2,\cdots,m_j$ since $-\frac{1}{\xi_k}<-1$. Similarly, when $\alpha<\sigma_j^2$, then $\mu_k>\sigma_j^2-\alpha$ for all $k=1,2,\cdots,m_j$.

When $\theta-r>\sigma_j^2$, which means the target region of NEP (6.1) is to the right of σ_j^2, then when we choose $\alpha>\theta-r>\sigma_j^2$, the poles satisfy $\mu_k<\sigma_j^2-\alpha<\theta-r-\alpha$ for all $k=1,2,\cdots,m_j$. Therefore, μ_k are to the left of $\Omega(s,r)$. Similarly, when $\theta+r<\sigma_j^2$, then when we choose $\alpha<\theta+r<\sigma_j^2$, the poles satisfy $\mu_k>\sigma_j^2-\alpha>\theta+r-\alpha$ for all $k=1,2,\cdots,m_j$. Therefore, μ_k are to the right of $\Omega(s,r)$.

Overall, if $\sigma_j^2\notin\Omega(\theta,r)$ for all $j=1,2,\cdots,q$, then when we choose any $\alpha\in\mathbb{R}\cap\Omega(\theta,r)$, all the poles are not in $\Omega(s,r)$. □

6.2 Eigenvalue problems with both TE and TM modes

When there are both transverse electric (TE) modes and transverse magnetic (TM) modes onto the waveguide boundaries accelerator cavities, the finite element discretization leads to the

following NEP

$$\mathcal{T}(\lambda)x = \left(K - \lambda M + \sum_{j=1}^{q_1} \mathtt{i}\sqrt{\lambda - \sigma_j^2} W_j + \sum_{j=q_1+1}^{q} \mathtt{i}\frac{\lambda}{\sqrt{\lambda - \rho_j^2}} W_j \right) x = 0, \tag{6.9}$$

where $K, M, W_j \in \mathbb{R}^{n\times n}$ are constant matrices and $\sigma_j, \rho_j \in \mathbb{R}$, W_j have rank-revealing factorizations $W_j = E_j E_j^{\mathrm{T}}$ with $E_j \in \mathbb{R}^{n\times r_j}$. We are interested in finding N eigenvalues in the upper half of the disk $\Omega(\theta, r)$ in complex plane.

Spectral transformation. Let $\mu = \lambda - \alpha$, where $\alpha \in \mathbb{R} \cap \Omega(\theta, r)$ is a chosen expansion point, then it yields the shifted NEP

$$\widehat{\mathcal{T}}(\mu)x = \left(K_\alpha - \mu M + \sum_{j=1}^{q_1} \mathtt{i}\widehat{\sigma}_j^{1/2} \sqrt{\frac{\mu}{\widehat{\sigma}_j} + 1} W_j + \sum_{j=q_1+1}^{q} \mathtt{i}\widehat{\rho}_j^{1/2} \frac{\frac{\mu}{\widehat{\rho}_j} + \frac{\alpha}{\widehat{\rho}_j}}{\sqrt{\frac{\mu}{\widehat{\rho}_j} + 1}} W_j \right) x = 0 \tag{6.10}$$

where $K_\alpha = K - \alpha M$, $\widehat{\sigma}_j = \alpha - \sigma_j^2$, and $\widehat{\rho}_j = \alpha - \rho_j^2$. We are supposed to find the eigenvalues in the upper half of the disk $\Omega(s, r)$, where $s = \theta - \alpha$.

Rational approximation of NEP (6.10). For the TE modes term, as treatmented in Section 6.1, we apply the Padé approximants of $\mathtt{i}\widehat{\sigma}_j^{1/2}\sqrt{\frac{\mu}{\widehat{\sigma}_j} + 1}$ by $\mathtt{i}\widehat{\sigma}_j^{1/2} r_{[m_j,m_j]}\left(\frac{\mu}{\widehat{\sigma}_j}\right)$, where $\mathtt{i}\widehat{\sigma}_j^{1/2} r_{[m_j]}\left(\frac{\mu}{\widehat{\sigma}_j}\right)$ is defined in (6.3).

For the TM modes, Example 2.3.3 shows that the formula for a Padé-type approximation of $\frac{z+\beta}{\sqrt{z+1}}$ is $h_{[m_j+1,m_j]}(z)$. Let $z = \frac{\mu}{\widehat{\rho}_j}$, $\beta = \frac{\alpha}{\widehat{\rho}_j}$, multiply (2.32) by $\mathtt{i}\widehat{\rho}_j^{1/2}$ and note that $\mathtt{i}^{-1} = -\mathtt{i}$, the rational approximation of $\mathtt{i}\widehat{\rho}_j^{1/2} \frac{\frac{\mu}{\widehat{\rho}_j} + \frac{\alpha}{\widehat{\rho}_j}}{\sqrt{\frac{\mu}{\widehat{\rho}_j}+1}}$ is

$$\mathtt{i}\widehat{\rho}_j^{1/2} h_{[m_j+1,m_j]}\left(\frac{\mu}{\widehat{\rho}_j}\right) = -b_{m_j}^{\mathrm{T}} \left(-\frac{\mathtt{i}}{\widehat{\rho}_j^{1/2}} I_{m_j} + \mathtt{i}\frac{\mu}{\widehat{\rho}_j^{3/2}} C_{m_j} \right)^{-1} b_{m_j} + \mathtt{i}\widehat{\rho}_j^{1/2} \kappa_{m_j} + \mathtt{i}\nu_{m_j} \frac{\mu}{\widehat{\rho}_j^{1/2}}, \tag{6.11}$$

where b_{m_j}, C_{m_j}, κ_{m_j} and ν_{m_j} are defined in (2.32).

Applying the rational approximants yields the following REP as an approximation of the NEP (6.10) in a compact form given by

$$\begin{aligned} \mathcal{R}_\alpha(\mu)x &\equiv \left(K_\alpha - \mu M + \sum_{j=1}^{q_1} \mathtt{i}\widehat{\sigma}_j^{1/2} r_{[m_j,m_j]}\left(\frac{\mu}{\widehat{\sigma}_j}\right) W_j + \sum_{j=q_1+1}^{q} \mathtt{i}\widehat{\rho}_j^{1/2} h_{[m_j+1,m_j]}\left(\frac{\mu}{\widehat{\rho}_j}\right) W_j \right) x \\ &= \left(\widehat{K}_\alpha - \mu\widehat{M} - \widehat{E}(\widehat{C} - \mu\widehat{D})^{-1}\widehat{E}^{\mathrm{T}} \right) x = 0 \end{aligned} \tag{6.12}$$

where $\widehat{K}_\alpha = K_\alpha + E\Gamma E^{\mathrm{T}} = K - \alpha M + E\Gamma E^{\mathrm{T}}$, $\widehat{M} = M - E\Omega E^{\mathrm{T}}$ and

$$\begin{aligned}
\Gamma &= \mathrm{diag}(\mathtt{i}\widehat{\sigma}_1^{1/2}\gamma_{m_1} I_{r_1}, \ldots, \mathtt{i}\widehat{\sigma}_{q_1}^{1/2}\gamma_{m_{q_1}} I_{r_{q_1}}, \mathtt{i}\widehat{\rho}_{q_1+1}^{1/2}\kappa_{m_{q_1+1}} I_{r_{q_1+1}}, \ldots, \mathtt{i}\widehat{\rho}_q^{1/2}\kappa_{m_q} I_{r_q}) \in \mathbb{C}^{r\times r},\\
\Omega &= \mathrm{diag}(O_{p_1\times p_1}, \mathtt{i}\nu_{m_{q_1+1}}/\widehat{\rho}_{q_1+1}^{1/2} I_{r_{q_1+1}}, \ldots, \mathtt{i}\nu_{m_q}/\widehat{\rho}_q^{1/2} I_{r_q}) \in \mathbb{C}^{r\times r},\\
E &= \begin{bmatrix} E_1 & E_2 & \ldots & E_q \end{bmatrix} \in \mathbb{R}^{n\times r},\\
\widehat{E} &= E \begin{bmatrix} I_{r_1}\otimes a_{m_1}^{\mathrm{T}} & \ldots & I_{r_{q_1}}\otimes a_{m_{q_1}}^{\mathrm{T}} & I_{r_{q_1+1}}\otimes b_{m_{q_1+1}}^{\mathrm{T}} & \ldots & I_{r_q}\otimes b_{m_q}^{\mathrm{T}} \end{bmatrix} \in \mathbb{R}^{n\times p},\\
\widehat{C} &= \mathrm{diag}(-\mathtt{i}I_{r_1 m_1}/\widehat{\sigma}_1^{1/2}, \ \ldots, \ -\mathtt{i}I_{r_{q_1} m_{q_1}}/\widehat{\sigma}_{q_1}^{1/2}, -\mathtt{i}I_{r_{q_1+1} m_{q_1}}/\widehat{\rho}_{q_1+1}^{1/2}, \ \ldots, \ -\mathtt{i}I_{r_q m_q}/\widehat{\rho}_q^{1/2}) \in \mathbb{C}^{p\times p},\\
\widehat{D} &= \mathrm{diag}\Big(-\mathtt{i}I_{r_1}\otimes D_{m_1}/\widehat{\sigma}_1^{3/2}, \ \ldots, \ -\mathtt{i}I_{r_{q_1}}\otimes D_{m_{q_1}}/\widehat{\sigma}_{q_1}^{3/2},\\
&\qquad -\mathtt{i}I_{r_{q_1+1}}\otimes C_{m_{q_1+1}}/\widehat{\rho}_{q_1+1}^{3/2}, \ \ldots, \ -\mathtt{i}I_{r_q}\otimes C_{m_q}/\widehat{\rho}_q^{3/2}\Big) \in \mathbb{C}^{p\times p},
\end{aligned}$$

where $r = r_1 + r_2 + \cdots + r_q$, $p = r_1 m_1 + r_2 m_2 + \cdots + r_q m_q$, $p_1 = r_1 m_1 + r_2 m_2 + \cdots + r_{q_1} m_{q_1}$ and $O_{p_1\times p_1}$ stands for a zero matrix with size $p_1 \times p_1$.

LEP. Similar to eigenvalue problems with no TM modes onto the waveguide boundaries, E and $\widehat{E}$ are real matrices and Γ, Ω, $\widehat{C}$ and $\widehat{D}$ are pure imaginary matrices. Therefore, all matrices involving matrix-vector multiplication for $H^{-1}x$ (5.16) are real matrices $E^{\mathrm{T}}V + \Delta^{-1}(s)$. Therefore, the arithmetic for matrix-vector multiplication can be as real as possible.

Pre-processing: proper choice of expansion point. In this section we discuss a method for choosing α based on the error formulas in Examples 2.3.1 and 2.3.3.

Recall that for the NEP (6.9), we approximated $\sqrt{\frac{\mu}{\widehat{\sigma}_j}+1}$ by $r_{[m_j]}\left(\frac{\mu}{\widehat{\sigma}_j}\right)$ defined in (6.3) for $j = 1, 2, \cdots, q$ and approximated $\frac{\frac{\mu}{\widehat{\rho}_j}+\frac{\alpha}{\widehat{\rho}_j}}{\sqrt{\frac{\mu}{\widehat{\rho}_j}+1}}$ by $h_{[m_j+1,m_j]}\left(\frac{\mu}{\widehat{\rho}_j}\right)$ defined in (6.11) for $j = q_1, q_1+1, \cdots, q$. The errors of approximations are

$$e_{m_j}\left(\frac{\mu}{\widehat{\sigma}_j}\right) = \sqrt{\frac{\mu}{\widehat{\sigma}_j}+1} - r_{[m_j]}\left(\frac{\mu}{\widehat{\sigma}_j}\right)$$

for $j = 1, 2, \cdots, q$ and

$$d_{m_j}\left(\frac{\mu}{\widehat{\rho}_j}, \beta_j\right) = \frac{\frac{\lambda}{\widehat{\rho}_j}+\frac{\alpha}{\widehat{\rho}_j}}{\sqrt{\frac{\mu}{\widehat{\rho}_j}+1}} - h_{[m_j+1,m_j]}\left(\frac{\mu}{\widehat{\rho}_j}\right)$$

with $\beta_j = \frac{\alpha}{\widehat{\rho}_j} = \frac{\alpha}{\alpha-\rho_j^2}$ for $j = q_1, q_1+1, \cdots, q$, where the explicit formula for $e_{m_j}(z)$ and $d_{m_j}(z,\beta)$

are defined in (2.16) and (2.33), respectively. Let the total error of approximation be

$$\begin{aligned} e(\mu) &= \sum_{j=1}^{q_1} \left| e_{m_j}\left(\frac{\mu}{\widehat{\sigma}_j}\right)\right| + \sum_{j=q_1+1}^{q} \left| d_{m_j}\left(\frac{\mu}{\widehat{\rho}_j}, \beta_j\right)\right| \\ &= \sum_{j=1}^{q_1} \left| e_{m_j}\left(\frac{\mu}{\alpha-\sigma_j^2}\right)\right| + \sum_{j=q_1+1}^{q} \left| d_{m_j}\left(\frac{\mu}{\alpha-\rho_j^2}, \frac{\alpha}{\alpha-\rho_j^2}\right)\right| \end{aligned} \tag{6.13}$$

Similar to eigenvalue problems with TE modes only, we select $\alpha \in (\theta - r, \theta + r)$ to balance the errors at the left and right ends of the target region $\Omega(s, r)$ by solving the equation

$$e(\theta + r - \alpha) = e(\theta - r - \alpha). \tag{6.14}$$

Again, we apply the bisection method to find the root.

Post-processing: poles of rational functions. We have already discussed the poles of $r_{[m_j,m_j]}\left(\frac{\mu}{\widehat{\sigma}_j}\right)$ for $j = 1, 2, \cdots, q_1$ in Section 6.1. In this section we discuss the poles of $h_{[m_j+1,m_j]}\left(\frac{\mu}{\widehat{\rho}_j}\right)$ for $j = q_1 + 1, q_1 + 2, \cdots, q$ by showing the following theorem.

Theorem 6.2.1. *Let the target region of NEP (6.9) be $\Omega(\theta, r)$ and target region of REP (6.12) be $\Omega(s, r)$ with $s = \theta - \alpha$. When $\rho_j^2 \notin \Omega(\theta, r)$ for $j = q_1 + 1, \cdots, q$ and $\alpha \in \mathbb{R} \cap \Omega(\theta, r) = (\theta - r, \theta + r)$, then the poles of rational functions $h_{[m_j+1,m_j]}\left(\frac{\mu}{\widehat{\rho}_j}\right)$ are not in $\Omega(s, r)$ for $j = q_1 + 1, \cdots, q$.*

Proof. By (2.31), the poles for $h_{[m_j+1,m_j]}(z)$ are $z_k = -\frac{1}{\zeta_k}$, where $\zeta_k = \sin^2 \frac{k\pi}{2m_j+1}$ for $k = 1, 2, \cdots, m_j$. The poles satisfy $z_k \in \mathbb{R}$ and $z_k < -1$. Following the method of proof for Theorem 6.1.2 we immediately get the results. □

Therefore, by Theorems 6.1.2 and 6.2.1, when $\sigma_j^2 \notin \Omega(\theta, r)$ for all $j = 1, 2, \cdots, q$, $\rho_j^2 \notin \Omega(\theta, r)$ for $j = q_1 + 1, \cdots, q$ and $\alpha \in \mathbb{R} \cap \Omega(\theta, r) = (\theta - r, \theta + r)$, then after computing the eigenvalues of LEP (5.9) the region $\Omega(s, r)$, we do not have to remove any eigenvalues near poles.

6.3 Numerical examples

In this section, we present numerical examples of eigenvalue problems discussed in the previous two subsections. In our MATLAB implementation of PAL algorithm (Algorithm 5), we use MATLAB's function `eigs` to solve LEP (5.9), and choose default convergence tolerance 10^{-14} and the maximum number of algorithm iterations 100. For computing the LU factorization of

$K-\theta M$, we apply MATLAB built-in function `LU` and choose the pivoting thresholds 0.1. For rank revealing factorization, we apply the SVD method.

We compare the PAL algorithm with NLEIGS (variant S) [27][1] and CORK [65][2]. For using NLEIGS and CORK, the target region is discretized by equally spaced points on the boundary of the region. For rational interpolations, we choose Leja-Bagby interpolation nodes and poles and set tolerance and maximum degree for rational interpolation 10^{-11} and 100, respectively. For rational Krylov subspace iterations, the tolerance for residual of eigenpairs was set to be 10^{-10} and the maximum number of iterations 130.

The accuracy of a computed eigenpair $(\widehat{\lambda},\widehat{x})$, is measured by the normalized residual norm

$$\mathrm{Res}(\widehat{\lambda},\widehat{x})=\frac{\|\mathcal{T}(\widehat{\lambda})\widehat{x}\|_2/\|\widehat{x}\|_2}{\|K\|_1+|\widehat{\lambda}|\|M\|_1+\sum_{j=1}^{q}\sqrt{|\widehat{\lambda}-\sigma_j^2|}\|W_j\|_1}.$$

for the eigenvalue problems with TE modes only, and and by the normalized residual norm

$$\mathrm{Res}(\widehat{\lambda},\widehat{x})=\frac{\|\mathcal{T}(\widehat{\lambda})\widehat{x}\|_2/\|\widehat{x}\|_2}{\|K\|_1+|\widehat{\lambda}|\|M\|_1+\sum_{j=1}^{q_1}\sqrt{|\widehat{\lambda}-\sigma_j^2|}\|W_j\|_1+\sum_{j=q_1+1}^{q}\frac{|\widehat{\lambda}|}{\sqrt{|\widehat{\lambda}-{\rho_j}^2|}}\|W_j\|_1}.$$

for the eigenvalue problems with both TM and TE modes.

All numerical experiments where conducted on an Intel(R) Core i7-4770K CPU@3.5GHz and 16GB RAM.

[1] http://twr.cs.kuleuven.be/research/software/nleps/nleigs.php, version 0.5 dated April 5, 2016
[2] https://bitbucket.org/roelvb/cork/src/master/, version 0.3 dated October 14, 2018.

Example 6.3.1 (Pillbox110658)**.** This is an example of NEP (6.1) with TE modes only:

$$\mathcal{T}(\lambda) \equiv K - \lambda M + \mathtt{i}\sqrt{\lambda - \sigma_1^2}W_1 + \mathtt{i}\sqrt{\lambda - \sigma_2^2}W_2 \tag{6.15}$$

where $\sigma_1 = 19.0400$, $\sigma_2 = 39.7633$, $\mathrm{rank}(W_1) = \mathrm{rank}(W_2) = 1$, and $n = 110,658$. It is called the pillbox waveguide problem

We report the numerical results of experiments with two different target domains. The first target domain is the upper half of the disk $\Omega_1(\theta, r) = \Omega(65^2, 65^2 - 41^2)$. To use the PAL algorithm, we set Padé orders $m_1 = m_2 = 9$. The LEP is of size $N_L = 110,658 + 9 + 9 = 110,676$. Since both σ_1^2 and σ_2^2 are outside of $\Omega_1(\theta, r)$, by Theorem 6.1.2, the poles for the rational functions $r_{[m_j,m_j]}\left(\frac{\mu}{\widehat{\sigma}_j}\right)$ are outside the target region of the corresponding REP (6.5). Therefore, the post-processing for the removal of the poles is not required. We use two different expansion points α: $\alpha = \theta = 65^2$ and $\alpha_{\mathrm{opt}} \approx 2393$, which is a root of of (6.6). Figure 6.1 shows the heat map of $\log_{10}[\widehat{e}(\sqrt{\lambda})]$, where the error function $\widehat{e}(\lambda) = e(\mu + \alpha)$ and $e(\mu)$ is defined in (6.7).

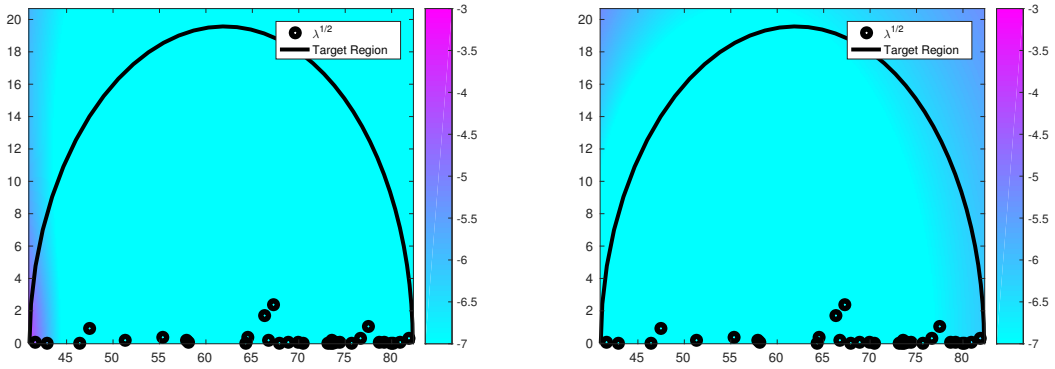

Figure 6.1: Square roots of computed eigenvalue (black dots) and the heat map of approximation errors $\log_{10}[\widehat{e}(\sqrt{\lambda})]$ with $\alpha = \theta$ (left) and α_{opt} (right) in the upper half of the disk $\Omega_1(\theta, r) = \Omega(65^2, 65^2 - 41^2)$ (Example 6.3.1).

Table 6.1 lists $\sqrt{\lambda}$ of computed 32 eigenvalues λ found in the domain (shown as the black-dots in Figure 6.1) and the corresponding residual norms. As we can see, with the choice of α_{opt}, the accuracy of computed eigenvalues near the left end of the region have been improved significantly. Furthermore, it is more accurate than the one computed by NLEIGS and CORK shown in the the left two columns of Table 6.1. For the rational interpolations of NLEIGS and CORK, we choose the singularity set $(-\infty, \sigma_2^2)$ and discretize the singularity set by logarithmically spaced

points $(\sigma_2^2 - 10^{-8+16k/10000})$ where $k = 0, 1, \cdots, 10000$. The minimum number of rational Krylov iterations is set to be 60 with the shifts: $\theta + 2r/3,\ \theta + (1+\mathtt{i})r/3,\ \theta,\ \theta + (-1+\mathtt{i})r/3,\ \theta - 2r/3$.

Table 6.1: Example 6.3.1, Square roots of 32 computed eigenvalues in the upper half of the disk $\Omega_1(\theta, r) = \Omega(65^2, 65^2 - 41^2)$.

Re$\sqrt{\lambda}$	Im$\sqrt{\lambda}$	PAL ($\alpha = \theta$)	PAL (α_{opt})	NLEIGS	CORK
4.1615e+01	3.7550e-02	4.0852e-09	1.0513e-12	3.8103e-11	7.0927e-12
4.2892e+01	1.1689e-02	1.3047e-10	3.2660e-14	3.7486e-13	1.1287e-16
4.6429e+01	9.3926e-03	1.5196e-13	2.6615e-14	8.8567e-14	7.0823e-17
4.7504e+01	9.3611e-01	1.2677e-13	1.2182e-13	6.5351e-15	3.6431e-17
5.1325e+01	2.0366e-01	5.4929e-14	3.8098e-14	7.7337e-16	2.7156e-17
5.5375e+01	3.6068e-01	4.8441e-14	4.0793e-14	1.6161e-14	7.2195e-17
5.7922e+01	1.8751e-01	3.3376e-14	2.1814e-14	6.0865e-14	5.7634e-17
5.8148e+01	4.2810e-02	1.5193e-14	1.7091e-14	5.6731e-14	1.2251e-16
6.4342e+01	4.3495e-03	2.7361e-15	3.0835e-15	1.4297e-15	2.4312e-17
6.4523e+01	3.3946e-01	4.4038e-15	1.2636e-14	2.2788e-16	2.5063e-17
6.6288e+01	1.7308e+00	2.9796e-14	2.7639e-14	9.2643e-16	2.5230e-17
6.6739e+01	1.7466e-01	1.1911e-14	4.1975e-14	8.5054e-15	3.9432e-17
6.7320e+01	2.3904e+00	4.9679e-14	4.6260e-14	8.0070e-16	3.0474e-17
6.7919e+01	4.2020e-03	9.3912e-15	1.7129e-14	4.8084e-14	4.7479e-17
6.8925e+01	4.6398e-02	1.4943e-14	8.5857e-14	1.7712e-13	1.0331e-16
6.9986e+01	3.9285e-02	1.2085e-14	1.6614e-14	3.0536e-13	1.0163e-16
7.0462e+01	3.2844e-03	1.2354e-14	4.9312e-14	2.6049e-13	1.2303e-16
7.3114e+01	1.2937e-03	1.3389e-14	9.0627e-15	2.4854e-13	8.4140e-17
7.3424e+01	9.6643e-03	1.8603e-14	2.0060e-14	1.6917e-13	7.3958e-17
7.3550e+01	1.7643e-01	3.7404e-14	1.6716e-12	7.8972e-14	6.7703e-17
7.3687e+01	5.7134e-03	8.1700e-15	1.0831e-14	4.8922e-14	5.4237e-17
7.3959e+01	3.7369e-02	1.5287e-14	1.9519e-14	2.3373e-13	5.5441e-17
7.4372e+01	4.4237e-02	2.4134e-14	1.2288e-12	2.3052e-13	5.4135e-17
7.5638e+01	6.0937e-05	1.1535e-14	7.5072e-14	6.7943e-15	3.4297e-17
7.6643e+01	3.1409e-01	6.2145e-14	7.8769e-12	9.8965e-17	3.0434e-17
7.7545e+01	1.0153e+00	1.1422e-13	9.5836e-14	5.7643e-16	4.5591e-17
7.8646e+01	4.6664e-02	2.7112e-14	2.6820e-14	4.2299e-14	5.6667e-17
7.9159e+01	3.4645e-02	2.4065e-14	2.3928e-14	1.9304e-13	2.9494e-17
7.9979e+01	1.8317e-03	1.2536e-14	1.7454e-14	2.8562e-13	3.9308e-17
8.0143e+01	2.4255e-06	1.4596e-14	1.4373e-14	1.2749e-13	3.3963e-17
8.0885e+01	3.0397e-02	3.4548e-14	1.0053e-11	3.2434e-13	9.6994e-17
8.1890e+01	3.1475e-01	8.0231e-14	7.4538e-14	1.7856e-13	1.2929e-16

In terms of wall-clock timing, PAL with α_{opt} took 40.50 seconds with about 11.73 second on the LDL factorization of $K - sM$ and 28.36 seconds on solving LEP. In contrast, NLEIGS takes 333.69 seconds and CORK 326.57 seconds.

The second target domain is the upper half of the disk $\Omega_2(\theta, r) = \Omega((\sigma_1^2 + \sigma_2^2)/2, 0.99(\sigma_2^2 - \theta))$. This is the so-called symmetric case discussed in Section 6.1. In this case, the optimal expansion

point $\alpha_{\text{opt}} = \theta$. Note that since σ_1^2 and σ_2^2 are not in Ω_2, by Theorem 6.1.2, the poles for the rational functions $r_{[m_j,m_j]}\left(\frac{\mu}{\widehat{\sigma}_j}\right)$ are not in the target region of REP (6.5). Therefore, removal of the poles is not necessary.

We use Padé orders $m_1 = m_2 = 9$. Figure 6.2 is the plot of the errors $\log_{10}[\widehat{e}(\sqrt{\lambda})]$ and the computed eigenvalues in the region, where $\widehat{e}(\lambda) = e(\mu + \alpha)$ and $e(\mu)$ is defined in (6.7). The error of rational approximation at λ is of order 10^{-10}.

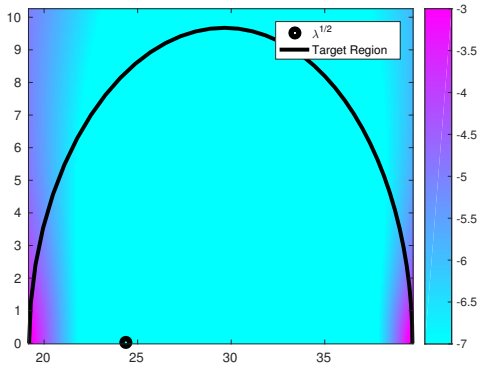

Figure 6.2: Square root of computed eigenvalues and heat map of the errors $\log_{10}[\widehat{e}(\sqrt{\lambda})]$ for Padé approximations for α_{opt} for Example 6.3.1.

The PAL algorithm found one eigenvalue in the target domain. The same number of eigenvalues are also found by NLEIGS and CORK as shown in Table 6.2.

Table 6.2: Example 6.3.1, Square root of computed eigenvalues in the upper half of the disk Ω_2.

$\text{Re}\sqrt{\lambda}$	$\text{Im}\sqrt{\lambda}$	PAL (α_{opt})	NLEIGS	CORK
2.4355e+01	3.4978e-02	1.3145e-15	6.8358e-13	2.7126e-13

For rational interpolations of NLEIGS and CORK, we choose the singularity set $(-\infty, \sigma_1^2) \cup (\sigma_2^2, +\infty)$ and discretize it by logarithmically spaced points $(\sigma_1^2 - 10^{-8+16k/10000})$ and $(\sigma_2^2 + 10^{-8+16k/10000})$ where $k = 0, 1, \cdots, 10000$. We set the minimum number of rational Krylov iterations 40 and use a single shift θ for rational Krylov steps. We note that when running NLEIGS for this problem, we replace function `ratnewtoncoeffsm` in NLEIGS v0.5 by function `ratnewtoncoeffsm` in CORK v0.3. Otherwise, it reports "`Linearization not converged after 100 iterations`".

In terms of wall-clock timing, PAL with α_{opt} took 21.11 seconds with about 11.62 seconds on the LDL factorization of $K - sM$ and 9.08 seconds on solving LEP. In contrast, NLEIGS takes

73.33 seconds and CORK 62.48 seconds.

Example 6.3.2 (TE10142)**.** The NEP of this example is of the same form of Example 6.3.1 with $\sigma_1 = 188.4956$, $\sigma_2 = 110.2352$, $\mathrm{rank}(W_1) = 355$, $\mathrm{rank}(W_2) = 200$, and $n = 10142$. This is an example where the ranks of W_1 and W_2 are large. The target domain is the upper half of the disk $\Omega(\theta, r) = \Omega(260^2, 0.99(260^2 - \sigma_1^2))$.

For the PAL algorithm, we choose the order of Padé approximation $m_1 = 11$, $m_2 = 6$. The LEP is of size $10,142 + 355 \times 11 + 200 \times 6 = 15,247$. We use two expansion points $\alpha = \theta = 260^2$ $(s = 0)$ and $\alpha_{\mathrm{opt}} \approx 41067$, an approximate root of (6.6) $(s = \theta - \alpha_{\mathrm{opt}} \approx 26533)$. Since both σ_1^2 and σ_2^2 are outside of $\Omega(\theta, r)$, by Theorem 6.1.2, the poles for the rational functions $r_{[m_j,m_j]}\left(\frac{\mu}{\widehat{\sigma}_j}\right)$ are outside of $\Omega(s, r)$ of the corresponding REP (6.5). Therefore, the post-processing for the removal of the poles is not required. Figure 6.3 is the plot the errors $\log_{10}[\widehat{e}(\sqrt{\lambda})]$ and computed eigenvalues in $\Omega(\theta, r)$, where $\widehat{e}(\lambda) = e(\mu + \alpha)$ and $e(\mu)$ is defined in (6.7).

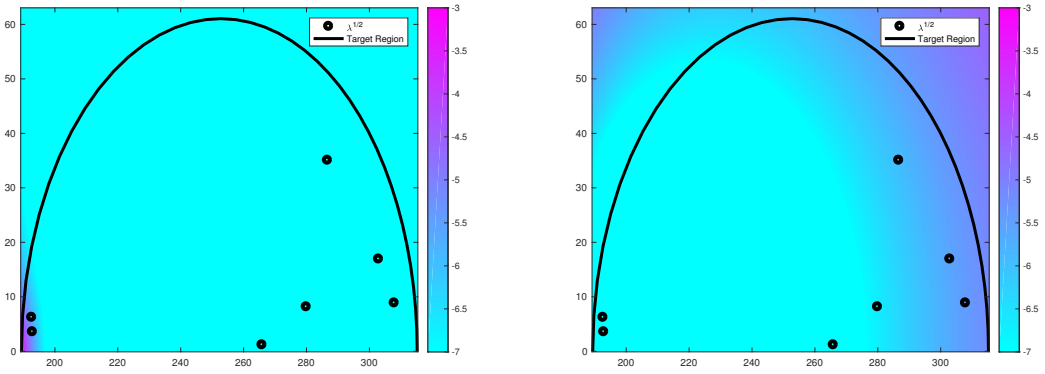

Figure 6.3: Square root of computed eigenvalues and heat map of the errors $\log_{10}[\widehat{e}(\sqrt{\lambda})]$ for Padé approximations for $\alpha = \theta$ (left) and α_{opt} (right) for Example 6.3.2.

We show the computed eigenvalues and residuals of PAL, NLEIGS and CORK in Table 6.3. For rational interpolations in NLEIGS and CORK, we choose the singularity set $(-\infty, \sigma_1^2)$ and discretize it by logarithmically spaced points $(\sigma_1^2 - 10^{-8+16k/10000})$ where $k = 0, 1, \cdots, 10000$. We set the minimum number of rational Krylov iterations 60 and choose five shifts $\theta + \frac{2r}{3}$, $\theta + \frac{(1+1\mathtt{i})r}{3}$, θ, $\theta + \frac{(-1+1\mathtt{i})r}{3}$, $\theta - \frac{2r}{3}$ in the rational Krylov steps. As we observe that by choosing α_{opt}, we can significantly improve the accuracy of he leftmost two eigenvalues.

Table 6.3: Square root of 7 computed eigenvalues and residuals for PAL with $\alpha = \theta$ and α_{opt}, NLEIGS and CORK for Example 6.3.2.

Re$\sqrt{\lambda}$	Im$\sqrt{\lambda}$	PAL ($\alpha = \theta$)	PAL (α_{opt})	NLEIGS	CORK
1.9260e+02	6.2441e+00	3.3720e-10	3.7457e-15	2.6624e-11	8.6956e-12
1.9283e+02	3.5934e+00	4.1736e-10	2.6772e-15	8.5848e-11	2.5824e-11
2.6582e+02	1.1983e+00	8.4819e-17	2.1234e-13	1.2230e-17	4.4950e-18
2.7990e+02	8.1778e+00	1.3499e-15	4.9687e-12	1.7144e-17	4.4884e-18
2.8664e+02	3.5082e+01	2.0375e-15	5.8645e-11	1.4606e-17	6.9619e-18
3.0288e+02	1.6948e+01	2.3045e-15	1.5286e-11	2.3443e-17	7.6373e-18
3.0787e+02	8.8885e+00	2.4566e-15	8.6016e-11	1.7816e-17	6.0367e-18

In terms of wall-clock timing, PAL with α_{opt} took 21.11 seconds with about 0.16 seconds on the LDL factorization of $K - sM$ and 1.88 seconds on solving LEP. In contrast, NLEIGS takes 12.38 seconds and CORK 7.34 seconds. The improvement of PAL in timing is less significant since we observed that due the relatively large rank of W_1 and W_2, the cost for computing the matrix V in the matrix-vector $H^{-1}x$ (in fact, it took 1.96 seconds, which is higher than solving the LEP). Furthermore, we used the full matrix E to generate V. If we use sparse matrix E, the running time for generating V and eigs are increased to 3.1315 s and 2.3411s, respectively. Therefore using the full matrix to represent E and V is more efficient than using the sparse matrix.

Example 6.3.3 (TETM5384)**.** This is an NEP of the form (6.9) with the presence of both TE and TM modes:

$$\mathcal{T}(\lambda)x \equiv \left(K - \lambda M + \mathtt{i}\sqrt{\lambda - \sigma_1^2}W_1 + \mathtt{i}\frac{\lambda}{\sqrt{\lambda - \rho_2^2}}W_2 \right) x = 0$$

where $\sigma_1 = 110.2353$, $\rho_2 = 258.9862$, $\text{rank}(W_1) = \text{rank}(W_2) = 1$, and $n = 5,384$. We are interested in finding eigenvalues in the upper half of the disk $\Omega(\theta, r) = \Omega(320^2, 320^2 - 264^2)$.

For the PAL algorithm, the orders of rational approximations $m_1 = 8$ and $m_2 = 10$. The LEP is of size $N_L = 5,384 + 8 + 10 = 5,402$. We choose two expansion points $\alpha = \theta = 320^2$ ($s = 0$) and $\alpha_{\text{opt}} \approx 79884$ ($s = \theta - \alpha_{\text{opt}} \approx 22516$) by the root of (6.14). Since both σ_1^2 and ρ_2^2 are not in $\Omega(\theta, r)$, by Theorem 6.2.1, the poles for the rational functions $r_{[m_j,m_j]}\left(\frac{\mu}{\widehat{\sigma}_j}\right)$ and $h_{[m_j+1,m_j]}\left(\frac{\mu}{\widehat{\rho}_j}\right)$ are not in $\Omega(s, r)$ of REP (6.12). Therefore, removal of the poles is not required.

Figure 6.4 is the plot of the square root of the computed eigenvalues and $\log_{10}[\widehat{e}(\sqrt{\lambda})]$ in $\Omega(\theta, r)$, where $\widehat{e}(\lambda) = e(\mu + \alpha)$ and $e(\mu)$ is defined in (6.13).

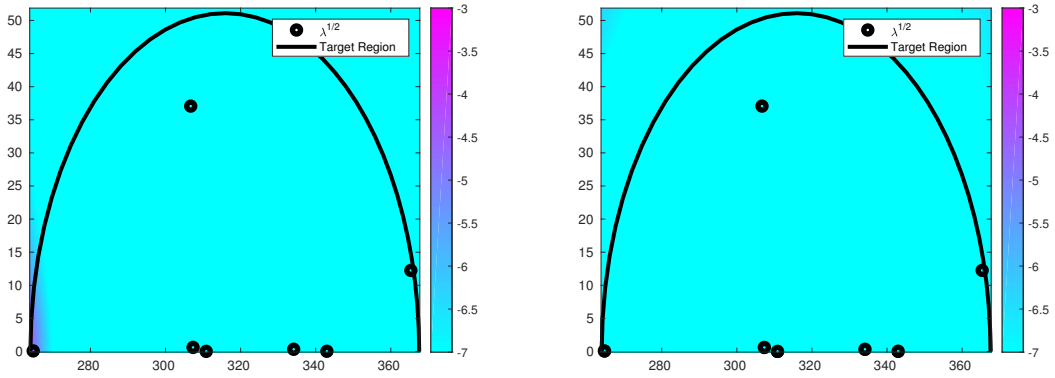

Figure 6.4: Square root of computed eigenvalues and heat map of $\log_{10}[\widehat{e}(\sqrt{\lambda})]$ for the error of rational approximations, $\alpha = \theta$ (left) and α_{opt} (right) for Example 6.3.3.

Table 6.4 shows the computed eigenvalues and the residual norms of PAL, NLEIGS and CORK. PAL took only 0.44 seconds to find these eigenvalues. For rational interpolations of NLEIGS and CORK, we choose the singularity set $(-\infty, \rho_2^2)$ and discretize it by logarithmically spaced points $(\rho_2^2 - 10^{-8+16k/10000})$ where $k = 0, 1, \cdots, 10000$. The minimum number of rational Krylov iterations is set to be 60 and five shifts are $\theta + \frac{2r}{3}$, $\theta + \frac{(1+1\mathtt{i})r}{3}$, θ, $\theta + \frac{(-1+1\mathtt{i})r}{3}$, $\theta - \frac{2r}{3}$ in the rational Krylov steps. As we observe that the expansion point α_{opt} significantly improves the accuracy of leftmost eigenvalue.

Table 6.4: Square root of 7 computed eigenvalues and residuals for PAL with $\alpha = \theta$ and α_{opt}, NLEIGS and CORK for Example 6.3.3.

Re$\sqrt{\lambda}$	Im$\sqrt{\lambda}$	PAL ($\alpha = \theta$)	PAL (α_{opt})	NLEIGS	CORK
2.6475e+02	8.9603e-02	8.0123e-10	5.3338e-14	2.2543e-12	1.3870e-12
3.0670e+02	3.7063e+01	5.7919e-15	4.4156e-14	5.8011e-17	3.5688e-17
3.0732e+02	6.1827e-01	1.0250e-15	7.9785e-16	4.0277e-17	2.7932e-17
3.1084e+02	1.9913e-02	4.3425e-16	5.1676e-16	4.0742e-17	2.8468e-17
3.3413e+02	3.3599e-01	4.7402e-16	1.6233e-15	3.4742e-17	2.7728e-17
3.4297e+02	3.7082e-02	1.1856e-15	8.3504e-15	4.9409e-17	2.9157e-17
3.6534e+02	1.2249e+01	9.0218e-15	1.5386e-13	8.3441e-17	3.1062e-17

We note that the dataset for this example was shared with us by Rich Lee around 2009.[3] Back to then, with much effort, he used a Picard-type iteration to find the leftmost eigenvalue.

[3]Private communication, Spring 2009

Here is a list of the eigenvalue found by Rich Lee, PAL and NLEIGS:

$$2.647526611365192 \times 10^2 + 8.960277296484288 \times 10^{-2}\mathtt{i} \quad \text{(Rich Lee)}$$

$$2.647526611422919 \times 10^2 + 8.960277540772293 \times 10^{-2}\mathtt{i} \quad \text{(PAL)}$$

$$2.647526611399669 \times 10^2 + 8.960277933397420 \times 10^{-2}\mathtt{i} \quad \text{(NLEIGS)}$$

Example 6.3.4 (TETM170562)**.** This is an NEP of the form (6.9) with the presence of three TE modes and two TM modes:

$$\mathcal{T}(\lambda)x \equiv \left(K - \lambda M + \sum_{j=1}^{3} \mathtt{i}\sqrt{\lambda - \sigma_j^2}W_j + \sum_{j=4}^{5} \mathtt{i}\frac{\lambda}{\sqrt{\lambda - \rho_j^2}}W_j \right) x = 0$$

where $\sigma_1 = 19.0400$, $\sigma_2 = 27.7658$, $\sigma_3 = \rho_5 = 39.7619$, $\rho_4 = 21.8621$, $\text{rank}(W_1) = \text{rank}(W_2) = \text{rank}(W_3) = \text{rank}(W_4) = \text{rank}(W_5) = 1$, and $n = 170,562$.

The first target domain is the upper half of the disk $\Omega_1(\theta, r) = \Omega(45^2, 0.99(45^2 - \sigma_3^2))$. For the PAL algorithm, we set the orders of rational approximations $m_1 = m_2 = 5$, $m_3 = 15$, $m_4 = 10$ and $m_5 = 20$, and use two expansion points: $\alpha = \theta = 45^2$ $(s = 0)$ $\alpha_{\text{opt}} \approx 1650$ $(s = \theta - \alpha_{\text{opt}} \approx 375)$ by solving (6.14). Since σ_1^2, σ_2^2, σ_3^2, ρ_4^2 and ρ_5^2 are not in $\Omega(\theta, r)$, by Theorem 6.2.1, the poles for the rational functions $r_{[m_j,m_j]}\left(\frac{\mu}{\sigma_j}\right)$ and $h_{[m_j+1,m_j]}\left(\frac{\mu}{\rho_j}\right)$ are not in the target region of REP (6.12). Therefore, removal of the poles is not required.

Figure 6.5 is the plot of the square root of the computed eigenvalues and the $\log_{10}[\widehat{e}(\sqrt{\lambda})]$, where $\widehat{e}(\lambda) = e(\mu + \alpha)$ and $e(\mu)$ is defined in (6.13).

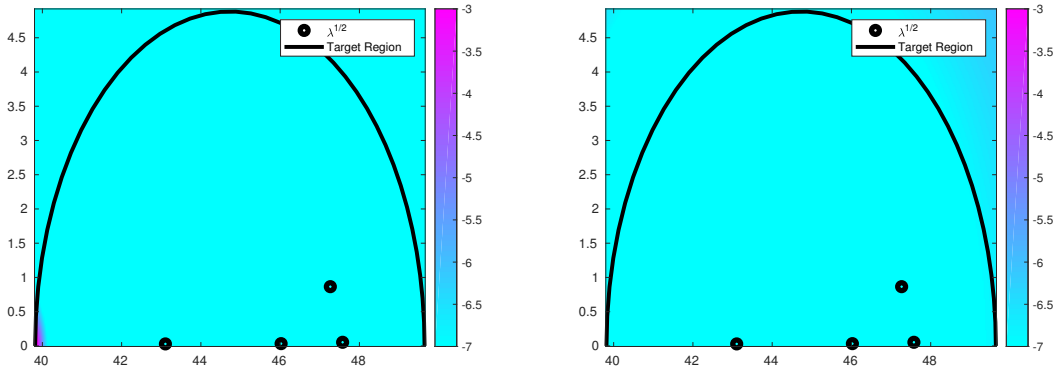

Figure 6.5: Square root of computed eigenvalues and heat map of the errors $\log_{10}[\widehat{e}(\sqrt{\lambda})]$ for rational approximations for $\alpha = \theta$ (left) and α_{opt} (right) Example 6.3.4 with target domain Ω_1.

Table 6.5 shows the computed eigenvalues and the residual norms of PAL and CORK. For rational interpolations in CORK, we choose the singularity set $(-\infty, \sigma_3^2)$ and discretize it by logarithmically spaced points $(\sigma_3^2 - 10^{-8+16k/10000})$ where $k = 0, 1, \cdots, 10000$. We set the minimum number of rational Krylov iterations 60 and a single shift θ in the rational Krylov steps. PAL took a total of 42.03 second to find 4 eigenvalues in Ω_1, among it 24.20 seconds for the factorization of $K - \lambda M$ and 15.93 for solving the LEP. CORK takes a total of 87.29 seconds to find the same number of eigenvalues in Ω_1.

Table 6.5: Square root of 4 computed eigenvalues and residual norm by PAL and CORK in target domain Ω_1 of Example 6.3.4.

Re$\sqrt{\lambda}$	Im$\sqrt{\lambda}$	PAL ($\alpha = \theta$)	PAL (α_{opt})	CORK
4.3105e+01	2.9150e-02	3.2443e-15	3.0370e-15	1.2374e-15
4.6024e+01	3.3226e-02	7.2697e-16	2.6691e-15	4.3598e-17
4.7266e+01	8.6521e-01	2.2715e-15	5.3040e-15	1.2301e-13
4.7576e+01	5.2981e-02	1.5438e-15	1.3437e-14	3.5899e-13

The second target domain is the upper half of the disk $\Omega_2(\theta, r) = \Omega((\rho_4^2 + \sigma_2^2)/2, 0.99(\sigma_2^2 - \theta))$. The PAL algorithm set the orders of rational approximations $m_1 = m_3 = 5$, $m_2 = m_4 = 15$ and $m_5 = 10$. The expansion point $\alpha_{\text{opt}} \approx 522$ ($s = \theta - \alpha_{\text{opt}} \approx 102$) is from the solution of (6.14). Figure 6.6 is the plot of the square root of the computed eigenvalues and the $\log_{10}[\widehat{e}(\sqrt{\lambda})]$ in Ω_2, where $\widehat{e}(\lambda) = e(\mu + \alpha)$ and $e(\mu)$ is defined in (6.13).

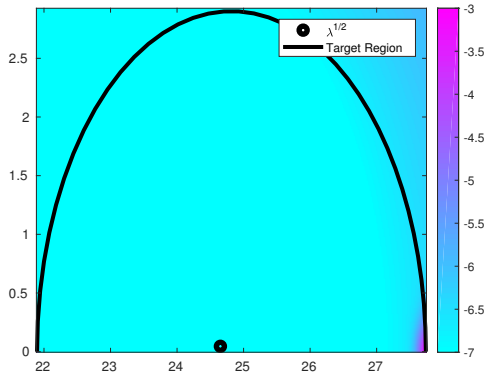

Figure 6.6: Square root of computed eigenvalues and heat map of the errors $\log_{10}[\widehat{e}(\sqrt{\lambda})]$ for target domain Ω_2 of Example 6.3.4.

Table 6.6 shows the computed eigenvalues and the residual norms of PAL and CORK. For rational interpolations of CORK, we choose the singularity set $(-\infty, \rho_4^2) \cup (\sigma_2^2, +\infty)$ and discretize

it by logarithmically spaced points $(\rho_4^2 - 10^{-8+16k/10000})$ and $(\sigma_2^2 + 10^{-8+16k/10000})$ where $k = 0, 1, \cdots, 10000$. We set the minimum number of rational Krylov iterations 20 and use a single shift θ for rational Krylov steps. PAL took a total of 32.33 seconds to find the eigenvalues in Ω_2, among it 23.56 seconds for the factorization of $K - \lambda M$ and 8.93 for solving the LEP. CORK takes a total of 65.49 seconds to find the number of eigenvalues in Ω_2.

Table 6.6: Square root of the computed eigenvalues and residual norm by PAL and CORK in target domain Ω_2 of Example 6.3.4.

$\mathrm{Re}\sqrt{\lambda}$	$\mathrm{Im}\sqrt{\lambda}$	PAL (α_{opt})	CORK
2.4651e+01	4.5545e-02	1.0877e-15	1.6621e-17

In accelerator design, the waveguides are used to introduce damping into the cavity to suppress the so-called *Higher-Order Modes* (HOMs). Those resonant modes are forming the wake-fields that can disrupt the stability of the beam accelerator and transport. The so-called external quality factors measure the effectiveness of the damping. Let $\kappa = \sqrt{\lambda}$, where λ is an eigenvalue in the region of interest, the resonant frequency ν and the corresponding external Q_e of the cavity are defined by

$$\nu(\kappa) = \frac{c}{2\pi} \cdot \mathrm{Re}(\kappa) \quad \text{and} \quad Q_e(\kappa) = \frac{1}{2} \cdot \frac{\mathrm{Re}(\kappa)}{\mathrm{Im}(\kappa)}, \tag{6.16}$$

where c is the speed of the light. The quantity Q_e measures the electromagnetic coupling between the cavity and waveguide. It characterizes the energy loss through the waveguide. With a given resonant frequency $\nu_0 > 0$, the cavity designers would like to seek frequencies ν that are close to ν_0 and $Q_e > 1$.

For the following three eigenvalues computed in the target domains:

$$\sqrt{\lambda_1} = 2.4651 \times 10^1 + 4.5545 \times 10^{-2}\mathtt{i} \quad (\text{in } \Omega_2)$$
$$\sqrt{\lambda_2} = 4.3105 \times 10^1 + 2.9150 \times 10^{-2}\mathtt{i} \quad (\text{in } \Omega_1)$$
$$\sqrt{\lambda_3} = 4.6024 \times 10^1 + 3.3226 \times 10^{-2}\mathtt{i} \quad (\text{in } \Omega_2)$$

the correpsonding frequencies are

$$\nu(\sqrt{\lambda_i}) = \frac{c}{2\pi} \cdot \mathrm{Re}\sqrt{\lambda_i} = 1.1762,\ 2.1960,\ 2.0567\mathrm{GHz}$$

and damping qualities are

$$Q_e(\sqrt{\lambda_i}) = \frac{1}{2}\frac{\mathrm{Re}\sqrt{\lambda_i}}{\mathrm{Im}\sqrt{\lambda_i}} = 270.6255,\ 739.3598,\ 692.5900,$$

respectively. These data are consistent with data reported in Table 1 of [63]. Residual norms of these computed eigenvalues by PAL algorithm and CORK are of order $O(10^{-14})$ to $O(10^{-15})$, which is of the same orders as reported in Table 1 of [63].

Chapter 7

Concluding remarks

In this dissertation, we discussed two problems related to nonlinear eigenvalue problems. The first problem is the constrained Rayleigh quotient optimization problem (CRQopt) and the second problem is the nonlinear eigenvalue problem with low rank nonlinear terms. We applied CRQopt for constrained image segmentation problem. For the nonlinear eigenvalue problem with low rank nonlinear terms, we applied it for resonant modes computation of accelerator cavity.

In the first part of the dissertation, we discussed an algorithm for CRQopt based on the reduction of Lagrange equations and QEP. Numerical examples and applications on constrained image segmentation problems showed the correctness and efficiency of our algorithm. For future work, our goal is to solve rLGopt (3.45) for the nearly hard case and to apply our algorithms on more machine learning problems such as outlier removal [41], semi-supervised kernel PCA [52], and transductive learning [34].

In the second part of the dissertation, we discussed an algorithm to solve nonlinear eigenvalue problems with low-rank nonlinear terms and applied our algorithm in resonant modes computation of accelerator cavity. For resonant modes computation of accelerator cavity, we proposed a method to choose the proper expansion point to reduce the approximation error on the target region and derived a method for matrix-vector multiplications to make arithmetic as real as possible. Numerical examples showed the efficiency and accuracy of our algorithm compared with existing algorithms. Our future work is application of our algorithm on more examples such as particle in a canyon [27] and delay problem [32] with low rank, and use multi-Padé approximation [24] for

nonlinear functions.

Appendices

Appendix A

Proof of the equivalence between CRQopt and the eigenvalue optimization problem

Suppose $U \in \mathbb{R}^{n\times(n-m)}$ has full column rank and that $\mathcal{R}(U) = \mathcal{N}(C^{\mathrm{T}})$ and let $u \in \mathbb{R}^n$ satisfies $C^{\mathrm{T}}u = \sqrt{n}\,b$. Define

$$\widehat{C} = [C^{\mathrm{T}},\ -\sqrt{n}b],\ N = \begin{array}{c} \\ n \\ 1 \end{array}\!\!\begin{array}{c} \begin{array}{cc} n-m & 1 \end{array} \\ \begin{bmatrix} U & u \\ 0 & 1 \end{bmatrix} \end{array}. \tag{A.1}$$

and

$$L = N^{\mathrm{T}}\begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix}N,\ E = N^{\mathrm{T}}\begin{bmatrix} -\frac{I}{n+1} & 0 \\ 0 & 1-\frac{1}{n+1} \end{bmatrix}N,\ M = N^{\mathrm{T}}\begin{bmatrix} I_n & 0 \\ 0 & 0 \end{bmatrix}N.$$

Note that it is easy to see that $\mathcal{R}(N) = \mathcal{N}(\widehat{C})$.

In this appendix we prove that CRQopt (3.1) is equivalent to the following eigenvalue optimization problem

$$\max_{t\in\mathbb{R}} \lambda_{\min}(L + tE, M), \tag{A.2}$$

where $\lambda_{\min}(L + tE, M)$ is the smallest eigenvalue of $(L + tE)x = \lambda Mx$. This equivalency was initiately established by Eriksson, Olsson and Kahl [14]. However, the statements presented here

are stronger than the related ones in [14]. For examples, we will prove M is positive definite, and we can use 'max' in (A.2) instead of 'sup' in [14].

Let $\widetilde{v} = \sqrt{n}v$, $\widehat{v} = \begin{bmatrix} \widetilde{v} \\ 1 \end{bmatrix}$, $\widehat{A} = \begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix}$, $\widehat{B} = \begin{bmatrix} I_n & 0 \\ 0 & 0 \end{bmatrix}$. Then v is a minimizer of CRQopt (3.1) if and only if $\widehat{v}$ is a minimizer of

$$\min \frac{\widehat{v}^{\mathrm{T}}\widehat{A}\widehat{v}}{\widehat{v}^{\mathrm{T}}\widehat{B}\widehat{v}}, \quad \text{s.t. } \widehat{v}_{(n+1)}^2 = 1,\ \widehat{v}^{\mathrm{T}}\widehat{v} = n+1,\ \widehat{C}\widehat{v} = 0. \tag{A.3}$$

Since $\mathcal{R}(N) = \mathcal{N}(\widehat{C})$, for any $\widehat{v}$ satisfying $\widehat{C}\widehat{v} = 0$, there exists $\widehat{y} \in \mathbb{R}^{n-m+1}$ such that $\widehat{v} = N\widehat{y}$, N is defined in (A.1). By the matrix structure in (A.1), we know that $\widehat{v}_{(n+1)}^2 = 1$ if and only if $\widehat{y}_{(n-m+1)}^2 = 1$. Therefore, solving (A.3) is equivalent to solving

$$\min \frac{\widehat{y}^{\mathrm{T}}L\widehat{y}}{\widehat{y}M\widehat{y}}, \quad \text{s.t. } \widehat{y}_{(n-m+1)}^2 - 1 = 0,\ \widehat{y}^{\mathrm{T}}N^{\mathrm{T}}N\widehat{y} = n+1. \tag{A.4}$$

To prove (A.4) is equivalent to its dual problem, we use the following result on the duality of the quadratic constrained optimization problems.

Lemma A.0.1 ([14, Corollary 1]). *Let $y^{\mathrm{T}}A_2y + 2b_2^{\mathrm{T}}y + c_2$ be a positive semidefinite quadratic form. If there exists y such that $y^{\mathrm{T}}A_3y + 2b_3^{\mathrm{T}}y + c_3 < 0$ and if A_3 is positive semidefinite, then the primal problem*

$$\inf_y \frac{y^{\mathrm{T}}A_1y + 2b_1^{\mathrm{T}}y + c_1}{y^{\mathrm{T}}A_2y + 2b_2^{\mathrm{T}}y + c_2}, \quad \text{s.t. } y^{\mathrm{T}}A_3y + 2b_3^{\mathrm{T}}y + c_3 = 0$$

and the dual problem

$$\sup_\lambda \inf_y \frac{y^{\mathrm{T}}(A_1 + \lambda A_3)y + 2(b_1 + \lambda b_3)^{\mathrm{T}}y + (c_1 + \lambda c_3)}{y^{\mathrm{T}}A_2y + 2b_2^{\mathrm{T}}y + c_2}$$

has no duality gap.

Proof. See [14, Corollary 1]. □

With the help of Lemma A.0.1, we have the following theorem to show that that there is no duality gap between the optimization problem (A.4) and its dual problem.

Theorem A.0.1 ([14, Theorem 1]). *Let $\widehat{A}_i = \begin{bmatrix} A_i & b_i \\ b_i^{\mathrm{T}} & c_i \end{bmatrix}$ for $i = 1, 2, 3$. If $\widehat{A}_2$ and A_3 are positive semidefinite and if there exists $\widehat{y}$ such that $\widehat{y}^{\mathrm{T}}\widehat{A}_3\widehat{y} < n+1$ and $\widehat{y}_{n+1}^2 = 1$, then the primal problem*

$$\inf_{y^{\mathrm{T}}A_3y + 2b_3^{\mathrm{T}}y + c_3 = n+1} \frac{y^{\mathrm{T}}A_1y + 2b_1^{\mathrm{T}}y + c_1}{y^{\mathrm{T}}A_2y + 2b_2^{\mathrm{T}}y + c_2} = \inf_{\widehat{y}^{\mathrm{T}}\widehat{A}_3\widehat{y} = n+1, \widehat{y}_{n+1}^2 = 1} \frac{\widehat{y}^{\mathrm{T}}\widehat{A}_1\widehat{y}}{\widehat{y}^{\mathrm{T}}\widehat{A}_2\widehat{y}} \tag{A.5}$$

and its dual

$$\sup_t \inf_{\widehat{y}^{\mathrm{T}}\widehat{A}_3\widehat{y}=n+1} \frac{\widehat{y}^{\mathrm{T}}\widehat{A}_1\widehat{y} - t\widehat{y}_{n+1}^2 - t}{\widehat{y}^{\mathrm{T}}\widehat{A}_2\widehat{y}}$$

has no duality gap.

Proof. Let γ_* be the optimal value of (A.5), then

$$\begin{aligned}
\gamma_* &= \inf_{\widehat{y}^{\mathrm{T}}\widehat{A}_3\widehat{y}=n+1, \widehat{y}_{n+1}^2=1} \frac{\widehat{y}^{\mathrm{T}}\widehat{A}_1\widehat{y}}{\widehat{y}^{\mathrm{T}}\widehat{A}_2\widehat{y}} \\
&= \sup_t \inf_{\widehat{y}^{\mathrm{T}}\widehat{A}_3\widehat{y}=n+1, \widehat{y}_{n+1}^2=1} \frac{\widehat{y}^{\mathrm{T}}\widehat{A}_1\widehat{y} + t\widehat{y}_{n+1}^2 - t}{\widehat{y}^{\mathrm{T}}\widehat{A}_2\widehat{y}} \\
&\geq \sup_t \inf_{\widehat{y}^{\mathrm{T}}\widehat{A}_3\widehat{y}=n+1} \frac{\widehat{y}^{\mathrm{T}}\widehat{A}_1\widehat{y} + t\widehat{y}_{n+1}^2 - t}{\widehat{y}^{\mathrm{T}}\widehat{A}_2\widehat{y}} \\
&\geq \sup_{t,\lambda} \inf_{\widehat{y}} \frac{\widehat{y}^{\mathrm{T}}\widehat{A}_1\widehat{y} + t\widehat{y}_{n+1}^2 - t + \lambda(\widehat{y}^{\mathrm{T}}\widehat{A}_3\widehat{y} - (n+1))}{\widehat{y}^{\mathrm{T}}\widehat{A}_2\widehat{y}} \\
&= \sup_{t,\lambda} \inf_{\widehat{y}} \frac{y^{\mathrm{T}}A_1y + 2b_1^{\mathrm{T}}y + c_1 + t\widehat{y}_{n+1}^2 - t + \lambda(y^{\mathrm{T}}A_3y + 2b_3^{\mathrm{T}}y + c_3 - (n+1))}{y^{\mathrm{T}}A_2y + 2b_2^{\mathrm{T}}y + c_2} \\
&= \sup_{t,\lambda} \inf_{\widehat{y}_{n+1}^2=1} \frac{y^{\mathrm{T}}A_1y + 2b_1^{\mathrm{T}}y + c_1 + \lambda(y^{\mathrm{T}}A_3y + 2b_3^{\mathrm{T}}y + c_3 - (n+1))}{y^{\mathrm{T}}A_2y + 2b_2^{\mathrm{T}}y + c_2} \qquad \text{(A.6)} \\
&= \inf_{y^{\mathrm{T}}A_3y + 2b_3^{\mathrm{T}}y + c_3 = n+1} \frac{y^{\mathrm{T}}A_1y + 2b_1^{\mathrm{T}}y + c_1}{y^{\mathrm{T}}A_2y + 2b_2^{\mathrm{T}}y + c_2} = \gamma_*, \qquad \text{(A.7)}
\end{aligned}$$

where (A.6) and (A.7) apply Lemma A.0.1. □

Remark A.0.1. One of the conditions in [14, Theorem 1] is "$\widehat{A}_3$ is positive semidefinite". However, the proof of Theorem A.0.1 applies Lemma A.0.1, which requires A_3 to be positive semidefinite and there exists $\widehat{y}$ such that $\widehat{y}^{\mathrm{T}}\widehat{A}_3\widehat{y} < n+1$ and $\widehat{y}_{n+1}^2 = 1$. Therefore, the condition "$\widehat{A}_3$ is positive semidefinite" is not necessary. In addition, in the statement of [14, Theorem 1], one of the constraints is $y_{n+1}^2 = 1$. However, in (A.5), the size of the matrix A_i and $\widehat{A_i}$ is $n \times n$ and $(n+1) \times (n+1)$ for $i = 1, 2, 3$, respectively. Therefore, we consider $y \in \mathbb{R}^n$ and $\widehat{y} \in \mathbb{R}^{n+1}$. Therefore, we change the constraint $y_{n+1}^2 = 1$ to $\widehat{y}_{n+1}^2 = 1$.

We now prove that the conditions of Theorem A.0.1 are staisfied for the constrained Rayleigh quotient optimization problem (A.4).

Lemma A.0.2. *Suppose* $\|v_0\| < 1$, *where* $v_0 = (C^{\mathrm{T}})^{\dagger}b$. *Then there exists* $\widehat{y}$ *such that* $\|\widehat{y}\|_N^2 = \widehat{y}^{\mathrm{T}}N^{\mathrm{T}}N\widehat{y} < n+1$ *and* $\widehat{y}_{(n-m+1)} = 1$.

Proof. Note that $v_0 = (C^{\mathrm{T}})^{\dagger} b$ is the minimum norm solution of $C^{\mathrm{T}} v = b$. Let $\widehat{v} = [\sqrt{n} v_0^{\mathrm{T}}, 1]^{\mathrm{T}}$. Then $\widehat{v} \in \mathcal{N}(\widehat{C})$ and thus there exists $\widehat{y}$ such that $\widehat{v} = N\widehat{y}$ for which we have $\|\widehat{y}\|_N = \|\widehat{v}\|_2 < \sqrt{n+1}$, and, at the same time, $\widehat{y}_{(n-m+1)} = \widehat{v}_{(n+1)} = 1$. □

By Lemma A.0.2 and Theorem A.0.1, the optimization problem (A.4) is equivalent to its dual problem

$$\sup_{t} \inf_{\widehat{y}^{\mathrm{T}} N^{\mathrm{T}} N \widehat{y} = n+1} \frac{\widehat{y}^{\mathrm{T}} L \widehat{y} + t \widehat{y}_{n-m+1}^2 - t}{\widehat{y}^{\mathrm{T}} M \widehat{y}}. \tag{A.8}$$

Since

$$t\widehat{y}_{n-m+1}^2 - t = t\widehat{y}_{n-m+1}^2 - t\frac{\widehat{y}^{\mathrm{T}} N^{\mathrm{T}} N \widehat{y}}{n+1} = \widehat{y}^{\mathrm{T}} E \widehat{y},$$

(A.8) is equivalent to

$$\sup_{t} \inf_{\widehat{y}^{\mathrm{T}} N^{\mathrm{T}} N \widehat{y} = n+1} \frac{\widehat{y}^{\mathrm{T}} (L + tE) \widehat{y}}{\widehat{y}^{\mathrm{T}} M \widehat{y}}. \tag{A.9}$$

To transform the dual problem (A.9) to an eigenvalue problem, we first prove that M is positive definite.

Lemma A.0.3. *Let b be as defined in (3.1c) and $b \neq 0$. N has full column rank, then M is positive definite.*

Proof. It is clear that M is positive semi-definite. We claim that M is nonsingular. Suppose, to the contrary, that M is singular. Then there exists a nonzero x such that $Mx = 0$.

We claim that $x_{(n-m+1)} \neq 0$; otherwise suppose $x_{(n-m+1)} = 0$ and write $x = \begin{bmatrix} x_1 \\ 0 \end{bmatrix}$. It follows from $Mx = 0$ that $U^{\mathrm{T}} U x = 0$, implying $x_1 = 0$ because U has full column rank. Thus $x = 0$, a contradiction.

Without loss of generality, we may normalize $x_{(n-m+1)}$ to 1, i.e., $x = \begin{bmatrix} x_1 \\ 1 \end{bmatrix}$. Note that $M = N^{\mathrm{T}} N - e_{n-m+1} e_{n-m+1}^{\mathrm{T}}$. $Mx = 0$ implies $N^{\mathrm{T}} N x = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$. $N^{\mathrm{T}} N$ is invertible. We now express $(N^{\mathrm{T}} N)^{-1}_{(n-m+1,n-m+1)}$ in two different ways. $N^{\mathrm{T}} N x = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ yields $x = (N^{\mathrm{T}} N)^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ and thus

$$1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}^{\mathrm{T}} x = \begin{bmatrix} 0 \\ 1 \end{bmatrix}^{\mathrm{T}} (N^{\mathrm{T}} N)^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = (N^{\mathrm{T}} N)^{-1}_{(n-m+1,n-m+1)}.$$

On the other hand,

$$N^{\mathrm{T}}N = \begin{bmatrix} U^{\mathrm{T}}U & U^{\mathrm{T}}u \\ u^{\mathrm{T}}U & u^{\mathrm{T}}u + 1 \end{bmatrix}.$$

By the assumption that U has full column rank, $U^{\mathrm{T}}U$ is invertible. With help of a formula [1] of the determinant of block matrices, we have

$$\det(N^{\mathrm{T}}N) = \det(U^{\mathrm{T}}U)\det[(1 + u^{\mathrm{T}}u - u^{\mathrm{T}}U(U^{\mathrm{T}}U)^{-1}U^{\mathrm{T}}u].$$

According to the relationship between the inverse and the adjoint of a matrix, we find

$$\begin{aligned}(N^{\mathrm{T}}N)^{-1}_{(n-m+1,n-m+1)} &= (-1)^{n-m+1+n-m+1}\frac{\det(U^{\mathrm{T}}U)}{\det(N^{\mathrm{T}}N)} \\ &= \frac{\det(U^{\mathrm{T}}U)}{\det(U^{\mathrm{T}}U)\det[(1 + u^{\mathrm{T}}u - u^{\mathrm{T}}U(U^{\mathrm{T}}U)^{-1}U^{\mathrm{T}}u]} \\ &= \frac{\det(U^{\mathrm{T}}U)}{\det(U^{\mathrm{T}}U)[1 + u^{\mathrm{T}}(I - P_U)u]},\end{aligned}$$

where P_U is the orthogonal projection onto $\mathcal{R}(U)$. Therefore, $(N^{\mathrm{T}}N)^{-1}_{(n-m+1,n-m+1)} = 1$ if and only if $u^{\mathrm{T}}(I - P_U)u = 0$ implying that u is in the column space of U. Without loss of generality, we may assume the first column of U is u. Now subtract the first column of N from its last column to conclude that e_{n+1} is in the null space of $\widehat{C}$, which contradicts that $b \neq 0$. □

By Lemma A.0.3 and Courant-Fisher minimax theorem [20, Theorem 8.1.2], finding

$$\inf_{\widehat{y}^{\mathrm{T}}N^{\mathrm{T}}N\widehat{y}=n+1} \frac{\widehat{y}^{\mathrm{T}}(L + tE)\widehat{y}}{\widehat{y}^{\mathrm{T}}M\widehat{y}}$$

is equivalent to finding the smallest eigenvalue of $K^{-1}(L + tE)K^{-\mathrm{T}}x = \lambda x$, where $M = KK^{\mathrm{T}}$ is the Cholesky factorization of M. Therefore, (A.9) is equivalent to

$$\sup_t \lambda_{\min}(L + tE, M). \tag{A.10}$$

Finally, we prove that the maximum value can be obtained, i.e., 'sup' in (A.10) can be replaced by 'max'.

Lemma A.0.4. *Let $f(t) = \lambda_{\min}(L + tE, M)$. There exits $t_0 \in \mathbb{R}$ such that $f(t_0) = \sup_{t\in\mathbb{R}} f(t)$.*

[1] $\det\left(\begin{bmatrix} A & B \\ C & D \end{bmatrix}\right) = \det(A)\det(D - CA^{-1}B)$ when A is invertable.

Proof. We prove the claim by showing that

$$\lim_{t\to+\infty} f(t) = \lim_{t\to-\infty} f(t) = -\infty.$$

First, let $v_1 \in \mathcal{R}(N)$ with the last component being zero, and set $y_1 = N^{\mathrm{T}} v_1$. We have $y_1^{\mathrm{T}} E y_1 = -\frac{\|v_1\|_2^2}{n+1} < 0$ and $y_1^{\mathrm{T}} M y_1 > 0$ since M is positive definite. Hence

$$\lim_{t\to+\infty} f(t) = \lim_{t\to+\infty} \inf_{\widehat{y}} \frac{\widehat{y}^{\mathrm{T}}(L+tE)\widehat{y}}{\widehat{y}^{\mathrm{T}} M \widehat{y}} \le \lim_{t\to+\infty} \frac{y_1^{\mathrm{T}}(L+tE)y_1}{y_1^{\mathrm{T}} M y_1} \le \lim_{t\to+\infty} t\frac{y_1^{\mathrm{T}} E y_1}{y_1^{\mathrm{T}} M y_1} + \lambda_{\max}(L,M) = -\infty.$$

Recall $v_0 = (C^{\mathrm{T}})^{\dagger} b$ and the assumption that $\|v_0\| < 1$. Let $v_2 = [\sqrt{n} v_0^{\mathrm{T}},\ 1]^{\mathrm{T}}$. Clearly $v_2 \in \mathcal{R}(N)$ and let $y_2 = N^{\mathrm{T}} v_2$. We have $y_2^{\mathrm{T}} E y_2 = -\frac{\|v_0\|_2^2}{n+1} + 1 - \frac{1}{n+1} > 0$ since $\|v_0\| < 1$ and $y_2^{\mathrm{T}} M y_2 > 0$ since M is positive definite. Hence

$$\lim_{t\to-\infty} f(t) = \lim_{t\to-\infty} \inf_{\widehat{y}} \frac{\widehat{y}^{\mathrm{T}}(L+tE)\widehat{y}}{\widehat{y}^{\mathrm{T}} M \widehat{y}} \le \lim_{t\to-\infty} \frac{y_2^{\mathrm{T}}(L+tE)y_2}{y_2^{\mathrm{T}} M y_2} \le \lim_{t\to-\infty} t\frac{y_2^{\mathrm{T}} E y_2}{y_2^{\mathrm{T}} M y_2} + \lambda_{\max}(L,M) = -\infty.$$

Therefore, there exits $t_1 < 0$ such that $f(t) < f(0)$ for $t < t_1$ and there exits $t_2 > 0$ such that $f(t) < f(0)$ for when $t > t_2$. Therefore

$$\sup_{t\in\mathbb{R}} f(t) = \sup_{t\in[t_1,t_2]} f(t).$$

Because $f(t) = \lambda_{\min}(L + tE, M)$ is a continuous function [60], there exists $t_0 \in [t_1, t_2]$ such that $f(t_0) = \sup_{t\in\mathbb{R}} f(t)$. □

In conclusion, we have shown that CRQopt (3.1) is equivalent to the eigenvalue optimization problem (A.2).

Appendix B

Sofrware package CRQPACK

The Lanczos algorithm for solving CRQopt (3.1) described in this paper has been implemented in MATLAB. In the spirit of reproducible research, MATLAB scripts of the implementation of the Lanczos algorithm and the data that used to generate numerical results presented in this paper are packed in a software called package called CRQPACK. CRQPACK can be obtained from

https://www.math.ucdavis.edu/~yszhou/CRQPACK.zip.

CRQPACK consists of three folders:

- `src`: the source code for solving CRQopt (3.1).

  It consists of four functions `CRQ_Lanczos`, `QEPmin`, `LGopt` and `rLGopt`. `CRQ_Lanczos` is the driver and calls `QEPmin` and `LGopt`. `LGopt` is dependent on `rLGopt`.

  In addition, we also provide two other drivers for solving CRQopt (3.1), namely `CRQ_explicit` for the direct method [17] and `CRQ_ppm` for the projected power method [68].

- `synthetic`: the drivers for numerical examples in section 3.5.

  `correct.m` and `QEPres.m` are for the examples in Sections 3.5.1 and 3.5.4, respectively. `CRQsharp.m` is used to generate the plots for Example 3.5.2 on error bounds in (3.90a) and (3.90b), while `CRQnotsharp.m` on the error bounds (3.90a) and (3.90b).

- `imagecut`: the code for constrained image segmentation.

It has three subfolders: `examples` contains the drivers, `data` contains image data including prior labeling information, and `auxiliary` contains program to generate the matrices A, C, and vector b of CRQopt (3.1).

The syntax of calling the driver `CRQ_Lanczos` is as follows:

```
[v,info] = CRQopt(A,C,b,opts)
```

where

- `A`: the matrix A in CRQopt (3.1)
- `C`: the matrix C in CRQopt (3.1)
- `b`: the vector b in CRQopt (3.1)
- `opts`: option parameters:
  - `opts.maxit`: maximum number of Lanczos iteraions
  - `opts.minit`: minimum number of Lanczos iteraions
  - `opts.tol`: tolerance of relative residual
  - `opts.method`: method to solve the optimization problem

    1: solve CRQopt via LGopt (default)

    2: solve CRQopt via QEPmin
  - `opts.checkstep`: the number of Lanczos steps between solving two rLGopt or two rQEPmin and checking the residuals
  - `opts.resopt`: option for computing the residual (only valid when `opts.method=2`)

    0 : using residual bound (3.67b) to estimate residual (default)

    1: using residual (3.67a)
  - `opts.returnQ`: indicator that whether the algorithm returns Q_k in structure `info`
- `v`: computed solution of CRQopt (3.1)
- `info`: information for some internal data:

– `info.n0`: vector n_0

– `info.b0`: vector b_0

– `info.gamma2`: the square of parameter γ

– `info.k`: the number of Lanczos steps

– `info.T`: tridiagonal matrix T_k

– `info.mu`: computed eigenvalue or Lagrange multipliers in each iteration

– `info.res`: norms of relative residual of Lagrange equations/QEP in each iteration

– `info.Q`: the matrix Q_k. This field is valid only when `opts.returnQ=1`

– `info.x`: a cell, whose elements are the solutions of all rLGopt (3.45) solved. This field is valid only when`opts.method=1`.

– `info.s`: a cell, whose element are the eigenvectors of all LEP (3.63) corresponding to the desired eigenvalue. This field is valid only when `opts.method=2`.

Appendix C

Software package PALPACK

We implemented PAL algorithm in MATLAB. The package is called PALAPCK and PALPACK can be obtained from

https://www.math.ucdavis.edu/~yszhou/PALPACK.zip.

Our software consists of three folders:

- `src`: source code for the solver of NEP (5.1) by PAL algorithm
  - `pal`: solves NEP (5.1);
  - `rrd`: rank revealing decompoisition;
  - `invmat`: matrix-vector multiplication (5.11).
- `data`: data matrices
- `examples`: the driver routines of the example of eigenvalue problems arising from computing resonant modes of accelerator cavities by PAL algorithm
  - `SLAC-I-Pillbox110658.m`: run Example 6.3.1
  - `SLAC-I-10142.m`: run Example 6.3.2
  - `SLAC-II-5384.m`: run Example 6.3.3
  - `SLAC-II-170562.m`: run Example 6.3.4
- `src_nleigs`: source code for the solver of NEP (5.1) by NLEIGS algorithm

- `examples_nleigs`: the driver routines of the example of eigenvalue problems arising from computing resonant modes of accelerator cavities by NLEIGS algorithm
- `src_cork`: source code for the solver of NEP (5.1) by CORK algorithm
- `examples_cork`: the driver routines of the example of eigenvalue problems arising from computing resonant modes of accelerator cavities by CORK algorithm

The syntax and the description of function `pal` is as follows:

```
[Lam, V, info] = pal(K, M, W, padefun, neig, shift, opts)
```

Input

- `K`: matrix K in NEP
- `M`: matrix C in NEP
- `padefun`: A structure representing the Pade approximation of each nonlinear function
  - `padefun.a`: cell array containing vectors a_{m_j}
  - `padefun.b`: cell array containing vectors b_{m_j}
  - `padefun.C`: matrix $\widehat{C} = \mathrm{diag}(I_{r_1} \otimes C_{m_1},\ I_{r_2} \otimes C_{m_2},\ \ldots,\ I_{r_q} \otimes C_{m_q}) \in \mathbb{C}^{p \times p}$
  - `padefun.D`: matrix $\widehat{D} = \mathrm{diag}(I_{r_1} \otimes D_{m_1},\ I_{r_2} \otimes D_{m_2},\ \ldots,\ I_{r_q} \otimes D_{m_q}) \in \mathbb{C}^{p \times p}$
  - `padefun.gamma`: vector containing scalars γ_j
  - `padefun.omega`: vector containing scalars ω_j
  - `padefun.poles`: cell array containing poles for each rational function
- `neig`: number of desired eigenvalues
- `shift`: the shift s
- `opts`: specify options
  - `opts.E`, `opts.F`: cell arrays containing rank revealing decompositions $W_j = E_j F_j^{\mathrm{T}}$
  - `opts.dodisp`: indicator of whether display timing or not

- `opts.rrdmethod`: indicator of the method to compute rank revealing decomposition. options: 'SVD'(default), 'LU', 'QR'.
- `opts.returnEF`: indicator of whether return the results for rank-revealing factorization $W_j = E_j F_j^{\mathrm{T}}$.

Output

- `Lam`: computed eigenvalues
- `V`: computed eigenvectors
- `info`: info structure with
  - `info.neig`: number of converged eigenvalues
  - `info.rank`: rank of W_j
  - `info.E` and `info.F`: rank revealing factorization $W_j = E_j F_j^{\mathrm{T}}$, valid when `opts.returnEF=1`
  - `info.poles`: poles of REP
  - `info.poles_Lam`, `info.poles_Lam`: eigenvalues removed as potential poles, with the corresponding eigenvectors

References

[1] A. C. Antoulas. *Approximation of Large-scale Dynamical Systems*. SIAM, Philadelphia, 2005.

[2] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst. *Templates for the Solution of Algebraic Eigenvalue Problems: a Practical Guide*. SIAM, Philadelphia, 2000.

[3] G. A. Baker and P. R. Graves-Morris. *Padé Approximants, 2nd edition*. Cambridge University Press, Cambridge, UK, 1996.

[4] A. Bamberger, B. Engquist, L. Halpern, and P. Joly. Higher order paraxial wave equation approximations in heterogeneous media. *SIAM Journal on Applied Mathematics*, 48(1):129–154, 1988.

[5] B. Beckermann and A. C. Matos. Algebraic properties of robust Padé approximants. *Journal of Approximation Theory*, 190:91–115, 2015.

[6] C. H. Bischof and G. Quintana-Ortí. Computing rank-revealing QR factorizations of dense matrices. *ACM Transactions on Mathematical Software*, 24(2):226–253, 1998.

[7] Å. Björck. *Numerical Methods for Least Squares Problems*. SIAM, Philadelphia, 1996.

[8] J. R. Bunch, C. P. Nielsen, and D. C. Sorensen. Rank-one modification of the symmetric eigenproblem. *Numerische Mathematik*, 31:31–48, 1978.

[9] S. E. Chew and N. D. Cahill. Semi-supervised normalized cuts for image segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1716–1723, 2015.

[10] J. W. Demmel. *Applied Numerical Linear Algebra*. SIAM, Philadelphia, 1997.

[11] N. R. Draper. "Ridge analysis" of response surfaces. *Technometrics*, 5(4):469–479, 1963.

[12] C. Effenberger and D. Kressner. Chebyshev interpolation for nonlinear eigenvalue problems. *BIT Numerical Mathematics*, 52(4):933–951, 2012.

[13] M. Embree, J. A. Loe, and R. B. Morgan. Polynomial preconditioned Arnoldi. *arXiv preprint arXiv:1806.08020*, 2018.

[14] A. Eriksson, C. Olsson, and F. Kahl. Normalized cuts revisited: A reformulation for segmentation with linear grouping constraints. *Journal of Mathematical Imaging and Vision*, 39(1):45–61, 2011.

[15] D. Fong and M. Saunders. LSMR: An iterative algorithm for sparse least-squares problems. *SIAM Journal on Scientific Computing*, 33(5):2950–2971, 2011.

[16] W. Gander. Least squares with a quadratic constraint. *Numerische Mathematik*, 36:291–307, 1981.

[17] W. Gander, G. H. Golub, and U. von Matt. A constrained eigenvalue problem. *Linear Algebra and its Applications*, 114-115:815–839, 1989.

[18] I. Gohberg, P. Lancaster, and L. Rodman. *Matrix Polynomials*. Academic Press, New York, 1982.

[19] G. H. Golub, M. Heath, and G. Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223, 1979.

[20] G. H. Golub and C. F. van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, 4th edition, 2013.

[21] G. H. Golub, Z. Zhang, and H. Zha. Large sparse symmetric eigenvalue problems with homogeneous linear constraints: the Lanczos process with inner-outer iterations. *Linear Algebra and its Applications*, 309(1):289–306, 2000.

[22] G. Golubr. Some modified matrix eigenvalue problems. *SIAM Review*, 15:318–334, 1973.

[23] P. Gonnet, S. Guttel, and L. N. Trefethen. Robust Padé approximation via SVD. *SIAM review*, 55(1):101–117, 2013.

[24] P. González-Vera and M. J. Páiz. Multipoint Padé-type approximation: an algebraic approach. *The Rocky Mountain Journal of Mathematics*, pages 531–558, 1999.

[25] N. I. M. Gould, S. Lucidi, M. Roma, and P. L. Toint. Solving the trust-region subproblem using the Lanczos method. *SIAM Journal on Optimization*, 9(2):504–525, 1999.

[26] S. Güttel and F. Tisseur. The nonlinear eigenvalue problem. *Acta Numerica*, 26:1–94, 2017.

[27] S. Guttel, R. Van Beeumen, K. Meerbergen, and W. Michiels. NLEIGS: A class of fully rational Krylov methods for nonlinear eigenvalue problems. *SIAM Journal on Scientific Computing*, 36(6):A2842–A2864, 2014.

[28] W. W. Hager. Minimizing a quadratic over a sphere. *SIAM Journal on Optimization*, 12(1):188–208, 2001.

[29] N. Halko, P.-G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

[30] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, 2001.

[31] N. J. Higham. *Accuracy and Stability of Numerical Algorithms*. SIAM, Philadelphia, 2002.

[32] E. Jarlebring, K. Meerbergen, and W. Michiels. A Krylov method for the delay eigenvalue problem. *SIAM Journal on Scientific Computing*, 32(6):3278–3300, 2010.

[33] C. Jiang, H. Xie, and Z. Bai. Robust and efficient computation of eigenvectors in a generalized spectral method for constrained clustering. In *Artificial Intelligence and Statistics*, pages 757–766, 2017.

[34] T. Joachims. Transductive learning via spectral graph partitioning. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, pages 290–297, 2003.

[35] J. Lampe and H. Voss. On a quadratic eigenproblem occurring in regularized total least squares. *Computational Statistics & Data Analysis*, 52(2):1090–1102, 2007.

[36] R.-C. Li. Solving secular equations stably and efficiently. Technical Report UCB//CSD-94-851, Computer Science Division, Department of EECS, University of California at Berkeley, 1993.

[37] R.-C. Li. Vandermonde matrices with Chebyshev nodes. *Linear Algebra and its Applications*, 428:1803–1832, 2007.

[38] R.-C. Li. On Meinardus' examples for the conjugate gradient method. *Mathematics of Computation*, 77(261):335–352, 2008.

[39] R.-C. Li. Sharpness in rates of convergence for symmetric Lanczos method. *Mathematics of Computation*, 79(269):419–435, 2010.

[40] B.-S. Liao, Z. Bai, L.-Q. Lee, and K. Ko. Nonlinear Rayleigh-Ritz iterative method for solving large scale nonlinear eigenvalue problems. *Taiwanese Journal of Mathematics*, 14(3A):869–883, 2010.

[41] W. Liu, G. Hua, and J. R. Smith. Unsupervised one-class learning for automatic outlier removal. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3826–3833, 2014.

[42] D. Lu, X. Huang, Z. Bai, and Y. Su. A Padé approximate linearization algorithm for solving the quadratic eigenvalue problem with low-rank damping. *International Journal for Numerical Methods in Engineering*, 103(11):840–858, 2015.

[43] Y. Y. Lu. A Padé approximation method for square roots of symmetric positive definite matrices. *SIAM Journal on Matrix Analysis and Applications*, 19(3):833–845, 1998.

[44] N. Mishra, R. Schreiber, I. Stanton, and R. E. Tarjan. Clustering social networks. In *International Workshop on Algorithms and Models for the Web-Graph*, pages 56–67, 2007.

[45] J. Moré and D. Sorensen. Computing a trust region step. *SIAM Journal on Scientific and Statistical Computing*, 4(3):553–572, 1983.

[46] M. E. J. Newman. Spectral methods for community detection and graph partitioning. *Physical Review E*, 88:042822, 2013.

[47] J Nocedal and S. J. Wright. *Numerical Optimization, 2nd edition*. Springer series in operations research and financial engineering. Springer, New York, 2006.

[48] C. C. Paige and M. A. Saunders. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software*, 8(1):43–71, 1982.

[49] C.-T. Pan. On the existence and computation of rank-revealing LU factorizations. *Linear Algebra and its Applications*, 316(1-3):199–222, 2000.

[50] C.-T. Pan and P. T. P. Tang. Bounds on singular values revealed by QR factorizations. *BIT. Numerical Mathematics*, 39(4):740–756, 1999.

[51] B. N. Parlett. *The Symmetric Eigenvalue Problem*. SIAM, Philadelphia, 1998.

[52] D. Paurat, D. Oglic, and T. Gärtner. Supervised PCA for interactive data analysis. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS) 2nd Workshop on Spectral Learning*, 2013.

[53] W. Pentney and M. Meila. Spectral clustering of biological sequence data. In *Association for the Advancement of Artificial Intelligence*, pages 845–850, 2005.

[54] F. Rendl and H. Wolkowicz. A semidefinite framework for trust region subproblems with applications to large scale minimization. *Mathematical Programming, Series A*, 77(1):273–299, 1997.

[55] Y. Saad. *Numerical Methods for Large Eigenvalue Problems*. Manchester University Press, Manchester, UK, 1992.

[56] Y. Saad. *Numerical methods for Large Eigenvalue Problems: Revised Edition*. SIAM, Philadelphia, 2011.

[57] M. A. Saunders. Solution of sparse rectangular systems using LSQR and CRAIG. *BIT Numerical Mathematics*, 35(4):588–604, 1995.

[58] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[59] D. M. Sima, S. Van Huffel, and G. H. Golub. Regularized total least squares based on quadratic eigenvalue problem solvers. *BIT Numerical Mathematics*, 44(4):793–812, 2004.

[60] G. W. Stewart and J.-G. Sun. *Matrix Perturbation Theory*. Academic Press, Boston, 1990.

[61] Y. Su and Z. Bai. Solving rational eigenvalue problems via linearization. *SIAM Journal on Matrix Analysis and Applications*, 32(1):201–216, 2011.

[62] R. Van Beeumen, E. Jarlebring, and W. Michiels. A rank-exploiting infinite Arnoldi algorithm for nonlinear eigenvalue problems. *Numerical Linear Algebra with Applications*, 23(4):607–628, 2016.

[63] R. Van Beeumen, O. Marques, E. G. Ng, C. Yang, Z. Bai, L. Ge, O. Kononenko, Z. Li, C.-K. Ng, and L. Xiao. Computing resonant modes of accelerator cavities by solving nonlinear eigenvalue problems via rational approximation. *Journal of Computational Physics*, 374:1031–1043, 2018.

[64] R. Van Beeumen, K. Meerbergen, and W. Michiels. A rational Krylov method based on hermite interpolation for nonlinear eigenvalue problems. *SIAM Journal on Scientific Computing*, 35(1):A327–A350, 2013.

[65] R. Van Beeumen, K. Meerbergen, and W. Michiels. Compact rational Krylov methods for nonlinear eigenvalue problems. *SIAM Journal on Matrix Analysis and Applications*, 36(2):820–838, 2015.

[66] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[67] X. Wang, B. Qian, and I. Davidson. On constrained spectral clustering and its applications. *Data Mining and Knowledge Discovery*, 28(1):1–30, 2014.

[68] L. Xu, W. Li, and D. Schuurmans. Fast normalized cut with linear constraints. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2866–2873, 2009.

[69] S. X. Yu and J. Shi. Segmentation given partial grouping constraints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):173–183, 2004.

[70] L.-H. Zhang, C. Shen, and R.-C. Li. On the generalized Lanczos trust-region method. *SIAM Journal on Optimization*, 27(3):2110–2142, 2017.