EIGENVALUE OPTIMIZATION FOR HERMITIAN FUNCTIONS - THEORY, APPLICATIONS AND ALGORITHMS

by
Mustafa Kılıç

A Thesis Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of
Master of Science

in
Mathematics
Koç University
August 2012

Koç University
Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by
Mustafa Kılıç

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Thesis Committee Members:

Assist. Prof. Emre Mengi (Advisor)	.....
Assoc. Prof. Emre Alper Yıldırım	.....
Prof. Varga Kalantarov	.....

Date: 27 August 2012

ABSTRACT

In this thesis we describe an algorithm to find the globally minimal value of a specified eigenvalue of a Hermitian matrix function depending on its parameters analytically. The algorithm exploits the boundedness of the second derivatives of the eigenvalues, and is globally convergent. It is based on the determination of the globally minimal value of a piece-wise quadratic under-estimator for the eigenvalue function repeatedly, and can be considered as an extension of an algorithm due to Breiman and Cutler. In the multi-variate case determining this globally minimal value can be posed as a quadratic program. The derivatives of the eigenvalue functions are used to construct quadratic models yielding rapid global convergence as compared to traditional global optimization algorithms. We also provide surveys on **(i)** the analytical properties of eigenvalues of Hermitian matrix functions, **(ii)** applications of the eigenvalue optimization, and **(iii)** existing numerical algorithms. Finally, we illustrate the asymptotic convergence behavior of the algorithm on numerical examples related to the distance to instability and distance to a nearest defective matrix from a given matrix as well as the Crawford number.

ÖZET

Bu tezde parametrelerine analitik olarak bağlı Hermit bir matris fonksiyonunun belirtilen bir özdeğerinin en ufak değerinin bulunması üzerine yoğunlaşıyoruz. Bu problemin global olarak en iyi çözümünü bulmak için Breiman ve Cutler algoritmasının bir uzantısını sunuyoruz. Algoritma Hermit matris fonsiyonunun özdeğerlerinin ikinci türevlerinin sınırlı olmasını kullanmakta ve global olarak en iyi çözüme yakınsıyor. Özdeğer fonksiyonunun altında yatan parçalı sürekli kuadratik fonksiyonların global minimumunu tekrar tekrar bulma fikri üzerine kurulu. Çok boyutlu durumda ise bu parçalı-kuadratik fonksiyonun global minimumu bir kuadratik optimizasyon probleminin çözümü olarak ifade edilebilir. Kuadratik fonksiyonları oluşturmak için kullanılan özdeğer fonksiyonlarının türevleri, bu algoritmanın geleneksel global optimizasyon algoritmalarına göre daha hızlı yakınsamasını sağlıyor. Ayrıca bu tezde **(i)** Hermit matris fonksiyonlarının özdeğerlerinin analitik özellikleri, **(ii)** özdeğer optimizasyon problemlerinin uygulamaları, ve **(iii)** varolan algoritmalar ile ilgili literatür taramalarına yer verilmekte. Son olarak algoritmanın asimtotik yakınsama özellikleri nümerik olarak dengesizlik mesafesi, Crawford numarası ve en yakındaki kusurlu bir matrise uzaklık problemleri üzerinde gösterilmiştir.

To my family

ACKNOWLEDGEMENTS

I have an eternal gratitude to my advisor, Assist. Prof. Emre Mengi, for his guidance, order and enthusiasm. At each step of my studies, I knew that he would be ready for help with accurate insight and without him, my efforts would carry little value.

I thank Assoc. Prof. E. Alper Yıldırım and Prof. Varga Kalantarov for their effort, participating in my thesis committee.

Additionally I want to thank TÜBİTAK, for its financial support during my graduate studies.

My homemates, Murat Tekşahin and Cemal Erdem have been great sources of joint wisdom and happiness during my studies, I can never repay them. I am also grateful to my friends in the graduate office of mathematics students for the fun and healthy environment they created. I know all of them will be both successful and as entertaining as ever.

Last, but not the least, I want to thank my parents Mehmet Kılıç & Emine Kılıç, my sister Hatice Nur Kılıç, my brother Gazi Kılıç and my prospective lifelong partner Işıl Taştan. I know that any speck of courage, dedication, responsibility and love I may have, is because of them. They are my warmth and laughter.

LIST OF SYMBOLS & ABBREVIATIONS

$\mathbb{R}$	real numbers
$\mathbb{C}$	complex number
$\mathbb{C}^n$	set of complex vectors of size n
$\mathbb{R}^n$	set of real vectors of size n
$\mathbb{C}^{n\times m}$	set of complex $n \times m$ matrices
$\mathbb{R}^{n\times m}$	set of real $n \times m$ matrices
A^*	adjoint (complex conjugate transpose) of the matrix A
A^T	transpose of the matrix A
$\det(A)$	determinant of the matrix A
$\|A\|_2$	2-norm of the matrix A
$\|A\|_F$	Frobenius norm of the matrix A
$\|A\|_\infty$	∞-norm of the matrix A
$\Lambda(A)$	spectrum of the matrix A
$F(A)$	field of values of the matrix A
$\sigma(A)$	set of the singular values
$\lambda_{\max}$	largest eigenvalue
$\lambda_{\min}$	smallest eigenvalue
λ_j	jth largest eigenvalue
$\sigma_{\min}$	smallest singular value
σ_j	jth largest singular value
$\beta(A)$	distance to instability of the matrix A
$\mu(A, B)$	distance to uncontrollability of the matrix pair (A, B)
$r(A)$	numerical radius of the matrix A
$\gamma(A)$	Crawford number of the matrix A
$\zeta(A)$	inner numerical radius of the matrix A
$\alpha_F(A)$	numerical abscissa of the matrix A
$\Re(z)$	real part of the complex number z
$\Im(z)$	imaginary part of the complex number z
$\|z\|$	modulus of the complex number z
$\bar{z}$	complex conjugate of the complex number z
$A \otimes B$	Kronecker product of the matrices A and B

List of Figures

List of Tables

Contents

Chapter 1

Introduction

This thesis concerns a Hermitian matrix function $\mathcal{A}(\epsilon) : \mathbb{R}^d \to \mathbb{C}^{n\times n}$ depending on its parameters analytically. We consider the *global* optimization of a prescribed eigenvalue $\lambda(\epsilon)$ of $\mathcal{A}(\epsilon)$ over $\epsilon \in \mathbb{R}^d$ numerically. Typically prescribed refers to the jth largest eigenvalue $\lambda(\epsilon) := \lambda_j(\mathcal{A}(\epsilon))$. But, it might as well as refer to a particular eigenvalue after ordering the eigenvalues with respect to a different criterion provided the analyticity properties discussed below hold.

We will focus on the univariate case and the multivariate case separately. In the univariate case, the eigenvalue functions of $\mathcal{A}(\epsilon) : \mathbb{R} \to \mathbb{C}^{n\times n}$ can be arranged so that each of its eigenvalue functions is analytic over $\mathbb{R}$. This analyticity property remains valid even if some of the eigenvalues repeat. On the other hand this analyticity property does not hold for non-Hermitian matrix functions, e.g., the eigenvalues of the matrix

$$\begin{bmatrix} 0 & -1 \\ \epsilon & 0 \end{bmatrix}$$

are not analytic functions with respect to $\epsilon \in \mathbb{R}$. From an application point of view the eigenvalues often need to be arranged from largest to smallest. With this arrangement, the eigenvalues of $\mathcal{A}(\epsilon)$ are still continuous, but only piece-wise analytic.

In the multivariate case, the analyticity of the eigenvalue functions does not hold anymore. But along any line in $\mathbb{R}^d$ ordering them from largest to smallest yields continuous and piece-wise analytic functions.

The algorithm that we use in this thesis is in essence adopted from a global optimization algorithm due to Breiman and Cutler [23] for the optimization of a prescribed eigenvalue $\lambda(\epsilon)$ of $\mathcal{A}(\epsilon)$. It heavily depends on boundedness of the second derivatives of the analytic pieces defining the prescribed eigenvalue. The eigenvalues of $\mathcal{A}(\epsilon)$ are typically non-convex functions, i.e., they may have many local extrema. Traditional derivative-based optimization algorithms are guaranteed to converge to only local minimizers. Here we benefit from global properties such as an upper bound on the second derivatives of analytic eigenvalues to converge to global minimizers.

Some of the most elementary examples that require the optimization of eigenvalues of Hermitian matrix functions are the distance to instability, the numerical radius of a matrix, H_∞ norm of a transfer function, the distance to uncontrollability from a linear time-invariant dynamical system, and the distance to defectiveness. Many researchers came up with specialized algorithms for the solution of each one of these problems. Van Loan has worked on the distance to instability concerning the dynamical system $x'(t) = Ax(t)$, and offered a heuristic-based algorithm [8]. Later, this problem has been considered by many researchers [37, 12, 9]. More recently Freitag and Spence offered a Newton-based algorithm [26]. The bisection method due to Byers [37] for the distance to instability inspired many algorithms. Boyd and Balakrishnan [43], and Bruinsma and Steinbuch [33] extended the bisection method to evaluate the H_∞ norm of the transfer function of a linear dynamical system. Paige [7] introduced the distance to uncontrollability for which the eigenvalue optimization characterization was presented by Eising [39]. In time, many algorithms have been suggested for the numerical solution of this problem [38, 28, 31, 30]. In another direction specific problems related to the nearest matrix with an eigenvalue of specified

algebraic multiplicity have been studied by many mathematicians. For instance the eigenvalue optimization characterization for the distance to a defective matrix was introduced by Malyshev [3], the eigenvalue optimization characterization for the distance to a nearest matrix with an eigenvalue of specified multiplicity was presented in [15], and the eigenvalue optimization characterization for the distance to a nearest pencil with eigenvalues lying in a specified region was studied in [13].

To our knowledge all algorithms for eigenvalue optimization have been derived for particular problems. The algorithm that is described here is based on piece-wise quadratic models lying underneath the objective function. The under-estimators for global optimization were first utilized by Piyavskii and Shubert [41, 6]. These algorithms [6] are derivative-free, and later the algorithms [14, 46] have been suggested attempting to estimate bounds for the first derivatives of the objective functions efficiently. The algorithm here exploits the derivatives, and the use of quadratic under-estimators based on derivatives yields faster convergence. A factor making the use of these quadratic models for eigenvalue optimization more appealing is the fact that the derivatives of the eigenvalue functions can be calculated by means of the analytic formulas in terms of the eigenvectors and the derivative of the matrix function.

OUTLINE

In Chapter 2 we review the basic results concerning the analyticity of the eigenvalues of a Hermitian matrix function $\mathcal{A}(\epsilon)$ that depends on ϵ analytically. These basic results are inherited from Rellich [17]. Also, we derive expressions for the first two derivatives of an analytic eigenvalue $\tilde{\lambda}(\epsilon)$, which is used in the definition of a piece-wise analytic eigenvalue $\lambda(\epsilon)$. These expressions first appeared in a Numerische Mathematik paper by Lancaster [35]. Chapter 3 focuses on some applications where eigenvalue optimization problems occur, and Chapter 4 focuses on some of the existing algorithms which can be applied to some of the elementary eigenvalue problems mentioned above.

Chapter 5 is devoted to the derivation of the one-dimensional algorithm, and the extension of the algorithm to the multivariate case. Also, this chapter focuses on the analysis of the multi-dimensional algorithm; specifically it establishes that there are subsequences of the sequence generated by the algorithm that converge to a global minimizer. Section 5.3 is devoted to some observations that can lead us to an efficient implementation of the algorithm in the multi-variate case. These observations are mostly due to Breiman and Cutler [23]. Chapter 6 presents the numerical results for the algorithm derived in Chapter 5 on some of the specific eigenvalue optimization problems given in Chapter 3.

Chapter 2

Perturbation Theory of Eigenvalues

This chapter summarizes the analyticity results concerning the eigenvalues of matrix functions dependent on real-parameters, mostly borrowed from Rellich's [17] and Lewis' works [5].

2.1 Eigenvalue Perturbation Theory

2.1.1 Univariate Case

In this part, we will analyze the analyticity of the eigenvalues and associated eigenvectors of Hermitian matrix functions depending on one variable analytically. We consider the eigenvalue problem

$$\mathcal{A}(\epsilon)u(\epsilon) = \tilde{\lambda}(\epsilon)u(\epsilon) \tag{2.1}$$

where $\mathcal{A}(\epsilon) \in \mathbb{C}^{n\times n}$ is analytic and $\mathcal{A}(\epsilon)^* = \mathcal{A}(\epsilon)$. By definition $u(\epsilon)$ is a nonzero vector in $\mathbb{C}^n$. We furthermore assume $\|u(\epsilon)\|_2 = 1$. Since $\mathcal{A}(\epsilon)$ is Hermitian, the

eigenvalue $\tilde{\lambda}(\epsilon)$ is real. Equation (2.1) implies that

$$\begin{aligned}(\mathcal{A}(\epsilon) - \tilde{\lambda}(\epsilon)I_n)u(\epsilon) = 0 \quad &\Leftrightarrow \quad (\mathcal{A}(\epsilon) - \tilde{\lambda}(\epsilon)I_n) \text{ is singular} \\ &\Leftrightarrow \quad \det(\mathcal{A}(\epsilon) - \tilde{\lambda}(\epsilon)I_n) = 0.\end{aligned}$$

Therefore $\tilde{\lambda}(\epsilon)$ is a root of the characteristic polynomial of $\mathcal{A}(\epsilon)$ of the form

$$p(\lambda, \epsilon) = \det(\mathcal{A}(\epsilon) - \lambda I_n) = \lambda^n + c_1\lambda^{n-1} + ... + c_{n-1}\lambda + c_n$$

where each coefficient c_j has a power series in terms of ϵ convergent for small ϵ in absolute value. By Puiseux' Theorem, any root $\tilde{\lambda}(\epsilon)$ of $p(\lambda, \epsilon)$ has an expansion of the form

$$\tilde{\lambda}(\epsilon) = \tilde{\lambda}(0) + b_1\epsilon^{\frac{1}{m}} + b_2\epsilon^{\frac{2}{m}} + ... = \tilde{\lambda}(0) + \sum_{k=1}^{\infty} b_k\epsilon^{\frac{k}{m}} \tag{2.2}$$

where m is the algebraic multiplicity of $\tilde{\lambda}(0)$. As we stated above, $\tilde{\lambda}(\epsilon)$ is real for all ϵ.

Lemma 2.1.1. *Only the integral powers appear in the expansion (2.2).*

Proof. Let b_j be the first nonzero coefficient in the expansion (2.2). Then we have

$$\lim_{\epsilon \to 0^+} \frac{\tilde{\lambda}(\epsilon) - \tilde{\lambda}(0)}{\epsilon^{j/m}} = b_j$$

implying b_j is real, since $\tilde{\lambda}(\epsilon)$, for all ϵ, and $\epsilon^{j/m}$ are real numbers. Furthermore

$$\lim_{\epsilon \to 0^-} \frac{\tilde{\lambda}(\epsilon) - \tilde{\lambda}(0)}{(-\epsilon)^{j/m}} = \frac{b_j}{(-1)^{j/m}}$$

must be a real number implying $(-1)^{j/m}$ is real, consequently $\frac{j}{m}$ is an integer.

□

This shows that $\tilde{\lambda}(\epsilon)$ is analytic at $\epsilon = 0$. Similarly, it can be decuded that $\tilde{\lambda}(\epsilon)$ is analytic at all ϵ. This implies that there exist coefficients a_k such that

$$\tilde{\lambda}(\epsilon) = \sum_{k=0}^{\infty} a_k \epsilon^k$$

for small *epsilon* in absolute value. Furthermore it can be shown that the associated eigenvectors are also analytic functions. More precisely associated with each analytic eigenvalue $\tilde{\lambda}(\epsilon)$ there exists a unit eigenvector $u(\epsilon) = \begin{bmatrix} u_1(\epsilon) & \dots & u_n(\epsilon) \end{bmatrix}^* \in \mathbb{C}^n$ such that $u(\epsilon)$ has a power series in terms of ϵ and convergent for small ϵ in absolute value. This follows from the following argument. Let $\gamma(\epsilon) := \mathcal{A}(\epsilon) - \tilde{\lambda}(\epsilon) I_n$. If $\gamma(\epsilon)$ is identically zero for all ϵ, then choose $u(\epsilon) = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix}^*$ and we are done. So suppose $\gamma(\epsilon)$ is nonzero. Since $\tilde{\lambda}(\epsilon)$ is an eigenvalue of $\mathcal{A}(\epsilon)$, $\gamma(\epsilon)$ is rank-deficient. Let $\text{rank}(\gamma(\epsilon)) = r$, where $1 \leq r \leq n-1$. Without loss of generality assume that $\det(\tilde{\gamma}_r(\epsilon)) \neq 0$, where

$$\tilde{\gamma}_r(\epsilon) := \begin{bmatrix} \gamma_{11}(\epsilon) & \dots & \gamma_{1r}(\epsilon) \\ \vdots & \ddots & \vdots \\ \gamma_{r1}(\epsilon) & \dots & \gamma_{rr}(\epsilon) \end{bmatrix}.$$

We define Γ_{ji} as the (i,j) cofactor of $\tilde{\gamma}_{r+1}(\epsilon)$ and

$$f_k(\epsilon) := \begin{cases} \Gamma_{k,r+1} & \text{if } k = 1, 2, \dots, r+1 \\ 0 & \text{if } k = r+2, \dots, n \end{cases}$$

Analyticity of $\mathcal{A}(\epsilon)$ and $\tilde{\lambda}(\epsilon)$ for all real ϵ implies that each $f_k(\epsilon)$ has a convergent power series in terms of ϵ for small ϵ in absolute value and $f(\epsilon) := [f_1(\epsilon) \ \dots \ f_n(\epsilon)]^T$ is not identically zero since $f_{r+1}(\epsilon) = \det(\tilde{\gamma_r}) \neq 0$. Moreover, the ith entry of the vector

$\gamma(\epsilon)f(\epsilon)$ vanishes for $i = 1, \ldots, r+1$. The ith entry of the vector is given by

$$\sum_{k=1}^{n}(\gamma(\epsilon))_{ik}f_k(\epsilon) = \sum_{k=1}^{r+1}(\gamma(\epsilon))_{ik}\Gamma_{k,r+1} = 0. \tag{2.3}$$

Observe that the second sum is the determinant of the matrix

$$\hat{\gamma}_{r+1}(\epsilon) = \begin{bmatrix} \gamma_{11}(\epsilon) & \ldots & \ldots & \ldots & \gamma_{1,r+1}(\epsilon) \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \gamma_{i1}(\epsilon) & \ldots & \ldots & \ldots & \gamma_{i,r+1}(\epsilon) \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \gamma_{r1}(\epsilon) & \ldots & \ldots & \ldots & \gamma_{r,r+1}(\epsilon) \\ \gamma_{i1}(\epsilon) & \ldots & \ldots & \ldots & \gamma_{i,r+1}(\epsilon) \end{bmatrix} \in \mathbb{C}^{(r+1)\times(r+1)},$$

which is equal to zero. To see this the second sum in (2.3) corresponds to the cofactor expansion of $\hat{\gamma}_{r+1}(\epsilon)$ across the $(r+1)$th row. Equation (2.3)holds also for $i = r+2, \ldots, n$, because in this case $\hat{\gamma}_{r+1}(\epsilon)$ corresponds to a $(r+1)\times(r+1)$ submatrix of $\gamma(\epsilon)$, which cannot be full rank due to rank$(\gamma(\epsilon) = r$ meaning $\det(\hat{\gamma}_{r+1}(\epsilon) = 0$. Define

$$u(\epsilon) := \frac{f(\epsilon)}{\|f(\epsilon)\|}, \qquad \text{where } \|f\| = (|f_1|^2 + \ldots + |f_n|^2)^{1/2} \text{ for real } \epsilon.$$

Then clearly $u(\epsilon)$ is an analytic function of real ϵ and $\|u(\epsilon)\| = 1$.

Let $\tilde{\lambda}_1(\epsilon)$ and $\tilde{\lambda}_2(\epsilon)$ be two distinct analytic eigenvalues of the Hermitian matrix $\mathcal{A}(\epsilon)$, and $\epsilon_1, \ldots, \epsilon_\ell$ be the points where $\tilde{\lambda}_1(\epsilon)$ and $\tilde{\lambda}_2(\epsilon)$ intersect each other, i.e, $\tilde{\lambda}_1(\epsilon_j) = \tilde{\lambda}_2(\epsilon_j)$ for $j = 1, \ldots, \ell$. Also let $v_1(\epsilon)$ and $v_2(\epsilon)$ be the eigenvectors, associated with $\tilde{\lambda}_1(\epsilon)$ and $\tilde{\lambda}_2(\epsilon)$ respectively. We know that there are finitely many such intersection points on a finite interval, say $[a, b]$, due to the analyticity of the

eigenvalues. By assumption, we have the following equalities

$$\mathcal{A}(\epsilon)v_1(\epsilon) = \tilde{\lambda}_1(\epsilon)v_1(\epsilon) \tag{2.4}$$

$$\mathcal{A}(\epsilon)v_2(\epsilon) = \tilde{\lambda}_2(\epsilon)v_2(\epsilon) \tag{2.5}$$

Premultiply equation (2.4) by $v_2^*(\epsilon)$ and equation (2.5) by $v_1^*(\epsilon)$. Now complex-conjugating the first equation and subtracting it from the second equation yield

$$(\tilde{\lambda}_2(\epsilon) - \tilde{\lambda}_1(\epsilon))v_1^*(\epsilon)v_2(\epsilon) = 0.$$

This implies that $v_1(\epsilon)$ is orthogonal to $v_2(\epsilon)$ for all $\epsilon \in \{[a,b] - \{\epsilon_1, \ldots, \epsilon_\ell\}\}$. Orthogonality of the eigenvectors remains valid for $\epsilon = \epsilon_j$ for $j = 1, \ldots, \ell$ due to the continuity of the inner product $v_1^*(\epsilon)v_2(\epsilon)$.

Theorem 2.1.2 (Rellich)**.** *For any Hermitian matrix function $\mathcal{A}(\epsilon) : \mathbb{R} \to \mathbb{C}^{n\times n}$ that depends on ϵ analytically, the following holds.*

(i) *The eigenvalue functions of $\mathcal{A}(\epsilon)$ can be arranged so that each $\tilde{\lambda}_j(\epsilon)$ for $j = 1, \ldots, n$ is an analytic function of ϵ.*

(ii) *There exists an orthonormal set $\{v_1(\epsilon), v_2(\epsilon), \ldots, v_n(\epsilon)\}$ of eigenvectors where $v_j(\epsilon)$ is a unit eigenvector associated with $\tilde{\lambda}_j(\epsilon)$ and analytic at all ϵ.*

The results above hold strictly for Hermitian matrices. For non-Hermitian matrices the eigenvalues are not necessarily real, consequently the Puiseux' series may have fractional powers. In other words, the eigenvalues may not be analytic no matter how they are ordered. For instance, consider the matrix $\begin{bmatrix} 0 & -1 \\ \epsilon & 0 \end{bmatrix}$, with the eigenvalues $\lambda_{1,2} = \pm\sqrt{-\epsilon}$ that are not analytic around $\epsilon = 0$.

2.1.2 Multivariate Case

In the multi-variate case, even if $\mathcal{A}(\epsilon) \in \mathbb{C}^{n\times n}$ is Hermitian and analytic for $\epsilon \in \mathbb{R}^d$ with $d > 1$, its eigenvalues may not be analytic regardless of their ordering. As an example, consider $\begin{bmatrix} 1+2\epsilon_1 & \epsilon_1+\epsilon_2 \\ \epsilon_1+\epsilon_2 & 1+2\epsilon_2 \end{bmatrix}$ with the eigenvalues $\lambda(\epsilon_1,\epsilon_2) = 1+\epsilon_1+\epsilon_2 \pm \sqrt{2}\sqrt{\epsilon_1^2+\epsilon_2^2}$, which do not have convergent power series around $\epsilon_1 = \epsilon_2 = 0$. Theorem 2.1.2 implies that for any direction $p \in \mathbb{R}^d$ the eigenvalues can be ordered as $\tilde{\lambda}_1, \ldots, \tilde{\lambda}_n$ so that each eigenvalue $\tilde{\lambda}_j(tp)$ is analytic at all $t \in \mathbb{R}$

2.2 Non-Smooth Analysis

We will analyze the differential properties of the eigenvalue functions of Hermitian matrices which are not differentiable in the usual sense. As an example, consider the following problem.

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & g(x) = u \end{aligned}$$

Let $h(u)$ denote the globally minimal value of the problem above. By using the *Newton's method* applied to Lagrangian function [20], we could solve the problem above. But $f(x)$ and $g(x)$ have to be differentiable functions to use that method. Unfortunately, there is no guarantee for the differentiability of $f(x)$ and $g(x)$. Now, define the problem above as follows

$$H(x) = \begin{cases} f(x) & \text{if } g(x) = u \\ \infty & else \end{cases}$$

Then we know that the minimum of $H(x)$ must be attained at a stationary point, and in the generalized sense this stationary point must satisfy the necessary conditions for optimality. Whenever f does not have a derivative in the usual sense, we define a

set called approximate subdifferential $\partial f(x)$ of $f(x)$ which behaves very much like a derivative. Then the optimality conditions can be stated in terms of the approximate subdifferentials instead of the derivatives. For instance, if f has a local minimum or maximum at x_0, then $0 \in \partial f(x_0)$ (see [16]).

Definition 2.2.1 (Regular Subgradient)**.** *Let E be an Euclidean space, and $f : E \to \mathbb{R}$. A vector $y \in E$ is called a regular subgradient of f at $x \in E$ if for all sequences $z = \{z_n\}$ with $\lim_{n\to\infty} z_n = 0$ for all n sufficiently large*

$$f(x + z_n) \geq f(x) + (y, z_n) + o(z_n)$$

where $(\cdot, \cdot)$ denotes the inner product and $o(z_n)$ is such that $lim_{z\to 0} \frac{o(z)}{\|z\|} = 0$.

The set of regular subgradients at x is denoted by $\hat{\partial} f(x)$. The problem with the set of regular subgradients is that it may be an empty set. The *approximate subdifferential* solves this problem.

Definition 2.2.2 (Approximate Subdifferential)**.** *Let E be an Euclidean space, and $f : E \to \mathbb{R}$. A $y \in E$ vector is called an approximate subdifferential of f at $x \in E$ if there exists a sequence of points x^r in E with $x^r \to x$, and $f(x^r) \to f(x) < \infty$ satisfying $y^r \to y$ for some sequence $\{y^r\}$ of regular subgradients $y^r \in \hat{\partial} f(x^r)$.*

The set of approximate subdifferentials of f at x is denoted by $\partial f(x)$.

Now we derive the subgradient of $f \circ \lambda$ at any given matrix X. Let $\mathcal{O}(n)$ denote the set of $n \times n$ real orthogonal matrices, and

$$U \cdot X := U^T X U. \tag{2.6}$$

Consider the eigenvalue function

$$\lambda : \mathcal{S}(n) \to \mathbb{R}^n, \quad \text{where } \lambda(\mathcal{A}) := \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_n \end{bmatrix},$$

$\mathcal{S}(n)$ is the set of $n \times n$ real symmetric matrices and $\lambda_1, \ldots, \lambda_n$ denote the eigenvalues of $\mathcal{A}$. For any extended real valued function f we will focus on

$$f \circ \lambda \to [-\infty, \infty].$$

Before deriving the subdifferential formula, we present three lemmas. For their proofs we refer to [5].

Lemma 2.2.3 (Diagonal Subgradient)**.** *For any pair of vectors x and y in $\mathbb{R}^n$ and any eigenvalue function $f \circ \lambda$, we have*

$$y \in \partial f(\lambda(X)) \Leftrightarrow Diag\ y \in \partial(f \circ \lambda)(Diag(\lambda(X))).$$

Lemma 2.2.4 (Subgradient Invariance)**.** *If a function $h : \mathcal{O}(n) \to \mathbb{R}$ is invariant with respect to the operation defined in (2.6) under a subgroup U of $\mathcal{O}(n)$, then all x in the domain of h and all $u \in U$ satisfy $\partial h(u \cdot x) = u \cdot \partial h(x)$.*

Lemma 2.2.5 (Commutativity)**.** *Suppose $Y \in \hat{\partial}(f \circ \lambda)(X)$ where $X \in \mathcal{S}(n)$, then $YX = XY$.*

A formula for the subdifferential of $f \circ \lambda$ can be deduced by using the lemmas above.

Theorem 2.2.6 (Subgradients of Eigenvalues)**.** *The set of approximate subdifferen-*

tials of any eigenvalue function $f \circ \lambda$ at any matrix X in $S(n)$ is given by

$$\partial(f \circ \lambda)(X) = \mathcal{O}(n)^X \cdot Diag(\partial f(\lambda(X)))$$

where $\mathcal{O}(n)^X = \{U \in \mathcal{O}(n) : U^T Diag(\lambda(X))U = X\}$.

Proof. First suppose $y \in \partial f(\lambda(X))$, then by Lemma 2.2.3 we have $Diag\ y \in \partial(f \circ \lambda)(Diag(\lambda(X)))$. Let $U \in \mathcal{O}(n)^X$. It follows from Lemma 2.2.4 that

$$Diag\ y \in \partial(f\circ\lambda)(Diag(\lambda(X))) \Leftrightarrow U\cdot Diag\ y \in \partial(f\circ\lambda)(U\cdot Diag\ \lambda(X)) = \partial(f\circ\lambda)(X).$$

So far we proved that $\mathcal{O}(n)^X \cdot Diag\,(\partial f(\lambda(X))) \subseteq \partial(f \circ \lambda)(X)$. For the reverse inequality assume

$$Y \in \partial(f \circ \lambda)(X) \tag{2.7}$$

By Lemma 2.2.5 this means $YX = XY$ implying X and Y are simultaneously diagonalizable, i.e., there exists a $U \in \mathcal{O}(n)^X$ such that

$$U^T \cdot Y = diag\ y$$

for some $y \in \mathbb{R}^n$. Applying Lemma 2.2.4 to (2.7) yields

$$diag\ y = U^T \cdot Y \in \big(\partial(f \circ \lambda)(U^T \cdot X)\big) = \partial(f \circ \lambda)(diag(\lambda(X)))$$

Now we deduce from Lemma 2.2.3 that $y \in \partial f(\lambda(X))$ meaning $diag\ y \in diag\ (\partial f(\lambda(X)))$, or equivalently $U \cdot diag\ y = Y \in U \cdot diag\ \partial f(\lambda(X))$. Consequently, $\partial(f \circ \lambda)(X) \subseteq \mathcal{O}(n)^X \cdot Diag\,(\partial f(\lambda(X)))$. Therefore $\partial(f \circ \lambda)(X) = \mathcal{O}(n)^X \cdot Diag\,(\partial f(\lambda(X)))$. $\square$

2.3 Derivatives of Eigenvalues and Eigenvectors

This chapter is devoted to the derivation of the first two derivatives of an analytic eigenvalue of a given Hermitian matrix function, mostly borrowed from [32].

2.3.1 First Derivatives of Eigenvalues

Let $\mathcal{A}(\epsilon)$ be a univariate Hermitian matrix-valued function depending on ϵ analytically and let $\tilde{\lambda}_j(\epsilon)$ and $v_j(\epsilon)$ be any analytic eigenvalue and the associated analytic eigenvector of $\mathcal{A}(\epsilon)$ as stated in Theorem 2.1.2. They satisfy

$$\mathcal{A}(\epsilon)v_j(\epsilon) = \tilde{\lambda}_j(\epsilon)v_j(\epsilon), \tag{2.8}$$

$$v_j(\epsilon)^*\mathcal{A}(\epsilon) = v_j(\epsilon)^*\tilde{\lambda}_j(\epsilon), \text{ and} \tag{2.9}$$

$$v_j(\epsilon)^*v_j(\epsilon) = 1. \tag{2.10}$$

Taking the derivatives of both sides of equation (2.10) with respect to ϵ, we get the following property

$$\frac{dv_j(\epsilon)^*}{d\epsilon}v_j(\epsilon) = -v_j(\epsilon)^*\frac{dv_j(\epsilon)}{d\epsilon}. \tag{2.11}$$

And differentiating the both sides of equation (2.8) yields

$$\frac{d\mathcal{A}(\epsilon)}{d\epsilon}v_j(\epsilon) + \mathcal{A}(\epsilon)\frac{dv_j(\epsilon)}{d\epsilon} = \frac{d\tilde{\lambda}_j(\epsilon)}{d\epsilon}v_j(\epsilon) + \tilde{\lambda}_j(\epsilon)\frac{dv_j(\epsilon)}{d\epsilon}. \tag{2.12}$$

By multiplying both sides of equation (2.12) by $v_j(\epsilon)^*$, and using (2.9), (2.11) as well as (2.10), we arrive at

$$\frac{d\tilde{\lambda}_j(\epsilon)}{d\epsilon} = v_j(\epsilon)^*\frac{d\mathcal{A}(\epsilon)}{d\epsilon}v_j(\epsilon). \tag{2.13}$$

2.3.2 First Derivatives of Eigenvectors

First observe that equation (2.11) implies that

$$\frac{d\left(v_j(\epsilon)^* v_j(\epsilon)\right)}{d\epsilon} = 0 \quad \Longrightarrow \quad v_j(\epsilon)^* \frac{dv_j(\epsilon)}{d\epsilon} = 0, \tag{2.14}$$

which means $v_j(\epsilon)$ and its derivative are orthogonal at all ϵ. In other words,

$$\frac{dv_j(\epsilon)}{d\epsilon} \in \ span\ \{v_k(\epsilon) : k \neq j\}. \tag{2.15}$$

For simplicity assume that the multiplicity of $\tilde{\lambda}_j(\epsilon)$ is one. In this case

$$(\tilde{\lambda}_j(\epsilon) I_n - \mathcal{A}(\epsilon))^\dagger = \sum_{k \neq j} \frac{1}{\tilde{\lambda}_j(\epsilon) - \tilde{\lambda}_k(\epsilon)} v_k(\epsilon) v_k(\epsilon)^* \tag{2.16}$$

and

$$(\tilde{\lambda}_j(\epsilon) I_n - \mathcal{A}(\epsilon))^\dagger (\tilde{\lambda}_j(\epsilon) I_n - \mathcal{A}(\epsilon)) = \sum_{k \neq j} v_k(\epsilon) v_k(\epsilon)^*, \tag{2.17}$$

where $\dagger$ denotes the Moore-Penrose pseudoinverse. By equation (2.16),

$$(\tilde{\lambda}_j(\epsilon) I_n - \mathcal{A}(\epsilon))^\dagger v_j(\epsilon) = 0.$$

Rearrange equation (2.12) so that

$$(\tilde{\lambda}_j(\epsilon) I - \mathcal{A}(\epsilon)) \frac{dv_j(\epsilon)}{d\epsilon} = \frac{d\mathcal{A}(\epsilon)}{d\epsilon} v_j(\epsilon) - \frac{d\tilde{\lambda}_j(\epsilon)}{d\epsilon} v_j(\epsilon). \tag{2.18}$$

Now multiply the both sides of equation (2.18) by $(\tilde{\lambda}_j(\epsilon) I_n - \mathcal{A}(\epsilon))^\dagger$ to obtain

$$\begin{aligned} (\tilde{\lambda}_j(\epsilon) I_n - \mathcal{A}(\epsilon))^\dagger (\tilde{\lambda}_j(\epsilon) I_n - \mathcal{A}(\epsilon)) \tfrac{dv_j(\epsilon)}{d\epsilon} \ &= \ (\tilde{\lambda}_j(\epsilon) I_n - \mathcal{A}(\epsilon))^\dagger \tfrac{d\mathcal{A}(\epsilon)}{d\epsilon} v_j(\epsilon) \\ &\quad - \tfrac{d\tilde{\lambda}_j(\epsilon)}{d\epsilon} \underbrace{(\tilde{\lambda}_j(\epsilon) I_n - \mathcal{A}(\epsilon))^\dagger v_j(\epsilon)}_{=0}. \end{aligned} \tag{2.19}$$

The expression $(\tilde{\lambda}_j(\epsilon)I_n - \mathcal{A}(\epsilon))^\dagger(\tilde{\lambda}_j(\epsilon)I_n - \mathcal{A}(\epsilon))$ is indeed an orthogonal projector onto $span\{v_k : k \neq j\}$. By using equation (2.15), we deduce

$$(\tilde{\lambda}_j(\epsilon)I_n - \mathcal{A}(\epsilon))^\dagger(\tilde{\lambda}_j(\epsilon)I_n - \mathcal{A}(\epsilon))\frac{dv_j(\epsilon)}{d\epsilon} = \frac{dv_j(\epsilon)}{d\epsilon}. \tag{2.20}$$

Finally, by substituting equation (2.20) in equation (2.19), we obtain

$$\frac{dv_j(\epsilon)}{d\epsilon} = (\tilde{\lambda}_j(\epsilon)I_n - \mathcal{A}(\epsilon))^\dagger \frac{d\mathcal{A}(\epsilon)}{d\epsilon} v_j(\epsilon). \tag{2.21}$$

2.3.3 Second Derivatives of Eigenvalues

Here for simplicity we assume that the multiplicity of $\tilde{\lambda}_j(\epsilon)$ is one. Taking the derivatives of both sides of (2.13) we get

$$\begin{aligned} \frac{d^2\tilde{\lambda}_j(\epsilon)}{d\epsilon^2} &= \left(\frac{dv_j(\epsilon)}{d\epsilon}\right)^* \frac{d\mathcal{A}(\epsilon)}{d\epsilon} v_j(\epsilon) + v_j(\epsilon)^* \frac{d^2\mathcal{A}(\epsilon)}{d\epsilon^2} v_j(\epsilon) + v_j(\epsilon)^* \frac{d\mathcal{A}(\epsilon)}{d\epsilon} \frac{dv_j(\epsilon)}{d\epsilon} \\ &= v_j(\epsilon)^* \frac{d^2\mathcal{A}(\epsilon)}{d\epsilon^2} v_j(\epsilon) + 2\ \Re\left(v_j(\epsilon)^* \frac{d\mathcal{A}(\epsilon)}{d\epsilon} \frac{dv_j(\epsilon)}{d\epsilon}\right). \end{aligned}$$

By substituting for $\frac{dv_j(\epsilon)}{d\epsilon}$ using the formula in (2.21) we have

$$\frac{d^2\tilde{\lambda}_j(\epsilon)}{d\epsilon^2} = v_j(\epsilon)^* \frac{d^2\mathcal{A}(\epsilon)}{d\epsilon^2} v_j(\epsilon) + 2\ \Re\left(v_j(\epsilon)^* \frac{d\mathcal{A}(\epsilon)}{d\epsilon} (\tilde{\lambda}_j(\epsilon)I_n - \mathcal{A}(\epsilon))^\dagger \frac{d\mathcal{A}(\epsilon)}{d\epsilon} v_j(\epsilon)\right),$$

or by using (2.16) for the pseudoinverse,

$$\frac{d^2\tilde{\lambda}_j(\epsilon)}{d\epsilon^2} = v_j(\epsilon)^* \frac{d^2\mathcal{A}(\epsilon)}{d\epsilon^2} v_j(\epsilon) + 2\sum_{k\neq j} \frac{1}{\tilde{\lambda}_j(\epsilon) - \lambda_k(\epsilon)} \left| v_k(\epsilon)^* \frac{d\mathcal{A}(\epsilon)}{d\epsilon} v_j(\epsilon)\right|^2. \tag{2.22}$$

2.3.4 Derivatives of Eigenvalues for Multivariate Hermitian Matrix Functions

Let $\mathcal{A}(\epsilon) : \mathbb{R}^d \to \mathbb{C}^{n\times n}$ be a Hermitian and analytic matrix valued function. It follows from (2.13) that

$$\frac{\partial\tilde{\lambda}_j(\epsilon)}{\partial\epsilon_k} = v_j^*(\epsilon)\frac{\partial\mathcal{A}(\epsilon)}{\partial\epsilon_k}v_j(\epsilon). \tag{2.23}$$

The first partial derivatives of the eigenvalue $\tilde{\lambda}_j(\epsilon)$ are continuous, which implies the existence of the second partial derivatives. By differentiating both sides of (2.23) with respect to ϵ_l, we get

$$\frac{\partial^2\tilde{\lambda}_j(\epsilon)}{\partial\epsilon_k\,\partial\epsilon_l} = v_j^*(\epsilon)\frac{\partial^2\mathcal{A}(\epsilon)}{\partial\epsilon_k\,\partial\epsilon_l}v_j(\epsilon) + v_j^*(\epsilon)\frac{\partial\mathcal{A}(\epsilon)}{\partial\epsilon_k}\frac{\partial v_j(\epsilon)}{\partial\epsilon_\ell} + \left(\frac{\partial v_j(\epsilon)}{\partial\epsilon_l}\right)^*\frac{\partial\mathcal{A}(\epsilon)}{\partial\epsilon_k}v_j(\epsilon)$$

One can simplify this expression by eliminating the derivatives of the eigenvectors and using (2.21) to obtain

$$\begin{aligned}\frac{\partial^2\tilde{\lambda}_j(\epsilon)}{\partial\epsilon_k\,\partial\epsilon_l} &= v_j^*(\epsilon)\frac{\partial^2\mathcal{A}(\epsilon)}{\partial\epsilon_k\,\partial\epsilon_l}v_j(\epsilon) + v_j^*(\epsilon)\frac{\partial\mathcal{A}(\epsilon)}{\partial\epsilon_k}(\tilde{\lambda}_j(\epsilon)I_n - \mathcal{A}(\epsilon))^\dagger\frac{\partial\mathcal{A}(\epsilon)}{\partial\epsilon_l}v_j(\epsilon)\\ &\quad + v_j^*(\epsilon)\frac{\partial\mathcal{A}(\epsilon)}{\partial\epsilon_l}(\tilde{\lambda}_j(\epsilon)I_n - \mathcal{A}(\epsilon))^\dagger\frac{\partial\mathcal{A}(\epsilon)}{\partial\epsilon_k}v_j(\epsilon),\end{aligned}$$

or, by replacing the pseudoinverses above with the right-hand sides of (2.16), we have

$$\begin{aligned}\frac{\partial^2\tilde{\lambda}_j(\epsilon)}{\partial\epsilon_k\,\partial\epsilon_l} = \;& v_j^*(\epsilon)\frac{\partial^2\mathcal{A}(\epsilon)}{\partial\epsilon_k\,\partial\epsilon_l}v_j(\epsilon)\\ &+\textstyle\sum_{m\neq j}\frac{1}{\tilde{\lambda}_j(\epsilon)-\lambda_m(\epsilon)}\left(v_j(\epsilon)^*\frac{d\mathcal{A}(\epsilon)}{\partial\epsilon_k}v_m(\epsilon)\right)\left(v_m(\epsilon)^*\frac{d\mathcal{A}(\epsilon)}{\partial\epsilon_l}v_j(\epsilon)\right)\\ &+\textstyle\sum_{m\neq j}\frac{1}{\tilde{\lambda}_j(\epsilon)-\lambda_m(\epsilon)}\left(v_j(\epsilon)^*\frac{d\mathcal{A}(\epsilon)}{\partial\epsilon_l}v_m(\epsilon)\right)\left(v_m(\epsilon)^*\frac{d\mathcal{A}(\epsilon)}{\partial\epsilon_k}v_j(\epsilon)\right).\end{aligned} \tag{2.24}$$

2.4 Analyticity of Singular Values

Let $\sigma_j(\epsilon)$ denote the jth largest singular value of an analytic $n \times m$ matrix function $\mathcal{A}(\epsilon)$. In this section $\mathcal{A}(\epsilon)$ is not necessarily Hermitian. Clearly the set of eigenvalues

of the Hermitian matrix function

$$\mathcal{B}(\epsilon) := \begin{bmatrix} 0 & \mathcal{A}(\epsilon) \\ \mathcal{A}(\epsilon)^* & 0 \end{bmatrix}$$

is $\{\sigma_j(\epsilon), -\sigma_j(\epsilon) : j = 1, \ldots, n\}$. Let us focus on the univariate case. Observe that $\sigma_j(\epsilon)$ is the jth largest eigenvalue of $\mathcal{B}(\epsilon)$. Suppose $\epsilon \in \mathbb{R}$, $v_j(\epsilon) := \begin{bmatrix} v_1(\epsilon) \\ v_2(\epsilon) \end{bmatrix}$ is the unit eigenvector, as specified in Theorem 2.1.2, associated with $\sigma_j(\epsilon)$, where $v_1(\epsilon) \in \mathbb{C}^n$ and $v_2(\epsilon) \in \mathbb{C}^m$ satisfying

$$\begin{bmatrix} 0 & \mathcal{A}(\epsilon) \\ \mathcal{A}(\epsilon)^* & 0 \end{bmatrix} \begin{bmatrix} v_1(\epsilon) \\ v_2(\epsilon) \end{bmatrix} := \sigma_j(\epsilon) \begin{bmatrix} v_1(\epsilon) \\ v_2(\epsilon) \end{bmatrix}.$$

Rewrite this as

$$\begin{aligned} \mathcal{A}(\epsilon)v_2(\epsilon) &= \sigma_j(\epsilon)v_1(\epsilon), \quad \text{and} \\ \mathcal{A}(\epsilon)^*v_1(\epsilon) &= \sigma_j(\epsilon)v_2(\epsilon). \end{aligned} \tag{2.25}$$

By multiplying the first equation by $v_1(\epsilon)^*$ and the second equation by $v_2(\epsilon)^*$, we obtain $v_1(\epsilon)^*v_1(\epsilon) = v_2(\epsilon)^*v_2(\epsilon)$, so $\|v_1(\epsilon)\| = \|v_2(\epsilon)\| = 1/\sqrt{(}2)$. By Theorem 2.1.2, $v_1(\epsilon)$, $v_2(\epsilon)$ are analytic. Also observe that the right and left singular vectors associated with the singular value $\sigma_j(\epsilon)$ are $v_1(\epsilon)$ and $v_2(\epsilon)$ due to (2.25). Now we can derive expressions for the first derivative of $\sigma_j(\epsilon)$, in terms of the corresponding right and left singular vectors of $\mathcal{A}(\epsilon)$. By using (2.13), we have

$$\begin{aligned} \frac{d\sigma_j(\epsilon)}{d\epsilon} &= \begin{bmatrix} v_1(\epsilon)^* & v_2(\epsilon)^* \end{bmatrix} \begin{bmatrix} 0 & d\mathcal{B}(\epsilon)/d\epsilon \\ d\mathcal{B}(\epsilon)^*/d\epsilon & 0 \end{bmatrix} \begin{bmatrix} v_1(\epsilon) \\ v_2(\epsilon) \end{bmatrix} \\ &= v_1(\epsilon)^* \tfrac{d\mathcal{B}(\epsilon)}{d\epsilon} v_2(\epsilon) + v_2(\epsilon)^* \tfrac{d\mathcal{B}(\epsilon)^*}{d\epsilon} v_1(\epsilon) \\ &= 2\,\Re\left(v_1(\epsilon)^* \tfrac{d\mathcal{B}(\epsilon)}{d\epsilon} v_2(\epsilon)\right) \end{aligned}$$

In terms of the unit left $\hat{v}_1(\epsilon) := \sqrt{2} \cdot v_1(\epsilon)$ and right $\hat{v}_2(\epsilon) := \sqrt{2} \cdot v_2(\epsilon)$ singular vectors associated with $\tilde{\lambda}_j(\epsilon)$ we obtain

$$\frac{d\sigma_j(\epsilon)}{d\epsilon} = \Re\left(\hat{v}_1(\epsilon)^* \frac{d\mathcal{B}(\epsilon)}{d\epsilon} \hat{v}_2(\epsilon)\right). \tag{2.26}$$

Chapter 3

Applications

This chapter presents some applications of eigenvalue optimization problems. We start the chapter with applications from numerical linear algebra and control theory, specifically some distance problems, H_∞-norm, some quantities related to field of values, and the role of eigenvalue optimization in the design of optimal preconditioners are discussed. This is followed by an information-theory motivated graph theory application. Then the relation between semidefinite programming and eigenvalue optimization problems is elaborated. In the last section applications of the eigenvalue optimization to logarithmic Chebyshev approximation, and some geometrical problems involving quadratic forms are presented. In this last section applications to structural optimization is also shortly mentioned.

3.1 Distance Problems

3.1.1 Distance to Defectiveness

Distance to a nearest defective matrix from a square matrix $\mathcal{A} \in \mathbb{C}^{n\times n}$ is defined as

$$\min \left\{ \|\Delta\mathcal{A}\|_2 : \Delta\mathcal{A} \in \mathbb{C}^{n\times n} \text{ s.t. } (\mathcal{A}+\Delta\mathcal{A}) \text{ is defective} \right\}$$

For this distance Malyshev [3] deduced the eigenvalue optimization characterization

$$\min_{\lambda\in\mathbb{C}} \max_{\gamma\in\mathbb{R}^+} \sigma_{2n-1}\left(\begin{bmatrix} \mathcal{A}-\lambda I_n & \gamma I_n \\ 0 & \mathcal{A}-\lambda I_n \end{bmatrix}\right).$$

The distance to defectiveness from $\mathcal{A}$ is small if and only if $\mathcal{A}$ has a highly sensitive eigenvalue [4, 19]. In [13] eigenvalue optimization characterizations were derived for the more general problem, the distance to a nearest matrix with an eigenvalue of specified algebraic multiplicity.

Unlike most of the other eigenvalue optimization problems that we will encounter in this section the eigenvalue characterization above is in the min-max form. There is no algorithm which is designed specifically for this problem based on the characterization above. On the other hand, the algorithm that we describe in Chapter 5 is applicable.

3.1.2 Distance to Instability

Suppose $\mathcal{A} \in \mathbb{C}^{n\times n}$ and $\Lambda(\mathcal{A}) \in \mathbb{C}^n$ is the set of the eigenvalues of $\mathcal{A}$ in the complex plane. If $\Lambda(\mathcal{A}) \subseteq \mathbb{C}^- := \{\lambda \in \mathbb{C} : \Re(\lambda) < 0\}$, then $\mathcal{A}$ is called a stable matrix. This terminology is related to the asymptotic behavior of the ODE $x'(t) = \mathcal{A}x(t)$. Stability is a very important property in engineering applications. Yet properties stronger than stability are desirable in practice. For instance even if $\mathcal{A}$ is stable, small perturbations may yield unstable matrices. In other words, $\mathcal{A} + E$ may have eigenvalues crossing the imaginary axis in the complex plane for E with small norm. Consequently, the distance to instability is introduced by Van Loan [8]. Let us use the notation $\sigma_{\min}(\mathcal{A})$ for the smallest singular value of $\mathcal{A}$, which satisfies

$$\sigma_{\min}(\mathcal{A}) = \min\{\|E\|_2 : \operatorname{rank}(\mathcal{A}+E) < n, E \in \mathbb{C}^{n\times n}\}$$

Then the distance to instability is defined as

$$\beta(\mathcal{A}) := \min\{\|E\|_2 : x'(t) = (\mathcal{A} + E)x(t) \text{ is unstable}\}$$

and has the eigenvalue optimization characterization

$$\beta(\mathcal{A}) = \min_{\omega \in \mathbb{R}} \sigma_{\min}(\mathcal{A} - i\omega I_n).$$

3.1.3 Distance to Uncontrollability

Two of the most important concepts in control theory are the controllability of a time-invariant linear system and its observability as the dual problem. For given matrices $A \in \mathbb{C}^{n\times n}$ and $B \in \mathbb{C}^{n\times m}$, the linear control system

$$x'(t) = Ax(t) + Bu(t), \tag{3.1}$$

is controllable if the state function $x(t)$ in (3.1) can be directed from any given state to a desired one in finite time by choosing the input function $u(t)$ appropriately. Any controllable system may be very close to an uncontrollable one. Furthermore uncontrollability of the system in (3.1) is equivalent to the rank deficiency of

$$\begin{bmatrix} B & AB & \dots & A^{n-1}B \end{bmatrix},$$

and this cannot be determined reliably in the presence of rounding errors. Thus, a continuous measure such as how far a controllable system is from a nearest uncontrollable system would be more useful.

The distance to uncontrollability of the system in (3.1) was introduced in [7] as

$$\mu(A, B) := \min\{\|[\delta A \ \delta B]\| : (A + \delta A, B + \delta B) \text{ is uncontrollable}\} \tag{3.2}$$

where $\|.\|$ denotes 2-norm or Frobenius norm[1]. This distance can equivalently be characterized as [39].

$$\mu(A, B) := \min_{\lambda \in \mathbb{C}} \sigma_{\min}([\lambda I_n - A, B]). \tag{3.3}$$

3.2 H_∞-norm

Consider the dynamical system

$$\begin{aligned} x'(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t), \end{aligned} \tag{3.4}$$

with $x(0) = 0$, where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$ and $D \in \mathbb{R}^{p\times m}$. Here $u(t) : [0, \infty) \to \mathbb{R}^m$ is called the control input, $y(t) : [0, \infty) \to \mathbb{R}^p$ is called the output of the system and the system is called of order n. In the Laplace domain this dynamical system can be represented as

$$Y(s) = H(s)U(s)$$

where $U(s)$ and $Y(s)$ denote the Laplace transformations of $u(t)$and $y(t)$ respectively, and

$$H(s) := C(sI - A)^{-1}B + D \tag{3.5}$$

is called the transfer function of system (3.4).

One very important quantity related to the transfer function, for instance in model reduction or controller synthesis, is its H_∞-norm given by

$$\|H\|_\infty := \max_{\omega \in \mathbb{R}} \sigma_{\max}(H(i\omega)). \tag{3.6}$$

[1]The definitions are equivalent with respect to the 2-norm and Frobenius norm.

3.3 Field of Values related quantities

3.3.1 Numerical Radius

The *Field of Values* of $\mathcal{A} \in C^{n\times n}$ is defined by

$$F(\mathcal{A}) := \{x^*\mathcal{A}x : \ x \in \mathbb{C}^n, \ \ x^*x = 1\}.$$

The numerical radius of $\mathcal{A}$ is the modulus of the outermost point in $F(\mathcal{A})$, that is

$$r(\mathcal{A}) := \max_{x\in F(\mathcal{A})} |x|. \tag{3.7}$$

Observe that $r(\mathcal{A})$ satisfies the following matrix norm properties.

i) $r(\mathcal{A}) = 0$ if and only if $\mathcal{A} = 0$.

ii) $r(\alpha\mathcal{A}) = \alpha r(\mathcal{A})$ for any complex scalar α.

iii) Triangular inequality $r(\mathcal{A} + \mathcal{B}) \leq r(\mathcal{A}) + r(\mathcal{B})$ holds.

But note, on the other hand, that the submultiplicative property does not hold. The norm of $\mathcal{A}$ is bounded by the numerical radius of $\mathcal{A}$ as follows.

$$r(\mathcal{A}) \leq \|\mathcal{A}\| \leq 2r(\mathcal{A}).$$

Additionally, the numerical radius does satisfy the power inequality $r(\mathcal{A}^k) \leq r(\mathcal{A})^k$ for any nonnegative k [36]. Therefore $r(\mathcal{A})$ captures the norm of $\mathcal{A}$ and also asymptotic behavior of the discrete first-order autonomous system $x_k = Ax_{k-1}$. The numerical radius have been used to analyze the convergence of classical iterative methods for linear systems by Axelsson [45] and Eiermann [27].

Another measure related to the field of values is the numerical abscissa motivated by continuous time autonomous dynamical systems. The numerical abscissa is the

real part of the rightmost point in the field of values.

$$\alpha_F(\mathcal{A}) := \{\Re(z) : z \in F(\mathcal{A})\}$$

It can be reduced to an eigenvalue problem. An eigenvalue optimization characterization for the numerical radius can be derived by exploiting this eigenvalue problem associated with the numerical abscissa. Notice that the field of values of $Ae^{i\theta}$ is the same as the field of values of $\mathcal{A}$ rotated θ radians in the counter clock-wise direction. Consequently, the numerical radius can be considered as a global optimization problem over θ of the form

$$r(\mathcal{A}) = \max_{\theta\in[0,2\pi)} \alpha_f(\mathcal{A}e^{i\theta}) \tag{3.8}$$

where $\alpha_f(\mathcal{A}e^{i\theta})$ denotes the numerical abscissa of $\mathcal{A}e^{i\theta}$. By using the eigenvalue characterization $\alpha_f(\mathcal{A}) = \lambda_{\max}(H(\mathcal{A}))$ for the numerical abscissa where $H(\mathcal{A}) = \frac{\mathcal{A}+\mathcal{A}^*}{2}$ [36], (3.8) can be rewritten as

$$r(\mathcal{A}) = \max_{\theta\in[0,2\pi)} \lambda_{\max}(H(\mathcal{A}e^{i\theta})). \tag{3.9}$$

3.3.2 Crawford Number

We can geometrically interpret the numerical radius of a matrix $\mathcal{A}$ as the maximal distance from $0 \in \mathbb{C}$ to the field of values $F(\mathcal{A})$. On the other hand, the minimal distance from $0 \in \mathbb{C}$ to the field of values $F(\mathcal{A})$ is called the Crawford number [44], defined by

$$\gamma(\mathcal{A}) := \min_{x\in F(\mathcal{A})} |x|. \tag{3.10}$$

By similar steps that led us to (3.9) this problem can be posed as

$$\gamma(\mathcal{A}) = \min_{\theta\in[0,2\pi)} \left|\lambda_{\max}(H(\mathcal{A}e^{i\theta}))\right|, \tag{3.11}$$

assuming $0 \notin F(\mathcal{A})$.

3.3.3 Inner Numerical Radius

The numerical radius of a matrix $\mathcal{A}$ is always attained at a point on the boundary of the field of values $F(\mathcal{A})$, while the Crawford number of $\mathcal{A}$ is attained on the boundary if and only if $0 \notin F(\mathcal{A})$. The inner numerical radius of $\mathcal{A}$ is defined by

$$\zeta(\mathcal{A}) := \min\{|\omega| : \omega \text{ is on the boundary of } F(\mathcal{A})\} \tag{3.12}$$

which is the same as $\gamma(\mathcal{A})$ whenever $0 \notin F(\mathcal{A})$. When the field of values does contain the origin, then $\gamma(\mathcal{A}) = 0$, while $\zeta(\mathcal{A})$ is the radius of the largest circle centered at the origin contained inside the field of values of $\mathcal{A}$.

The inner numerical radius is equivalent to the eigenvalue optimization problem

$$\zeta(\mathcal{A}) = \min_{\theta\in[0,2\pi)} \left|\lambda_{\max}\left(H(e^{-i\theta}\mathcal{A})\right)\right| \tag{3.13}$$

regardless whether $0 \notin F(\mathcal{A})$ or not.

3.4 Design of Optimal Preconditioners

Greenbaum and Rodrigue [2] has worked on the following optimization problem; for a given positive definite symmetric matrix B, find the positive definite symmetric matrix M with the prescribed sparsity pattern that minimizes the 2-norm condition

number of $M^{-1/2}BM^{1/2}$. This problem is equivalent to

$$M = \arg\min_{M} \lambda_{\max}(I_n - L^{-1}ML^{-1}),$$

where LL^T is the Cholesky factorization of B. Finding and applying such preconditioners would speed up the convergence of Krylov subspace methods when they are used to solve linear systems or eigenvalue problems associated with B.

3.5 Graph Problem

Let $M \in \mathbb{R}^{n\times n}$ be a symmetric adjacency matrix associated with a graph with the following properties. Its entries along the diagonal are zero. Its (i,j)-entry for $i \neq j$ is equal to zero whenever the vertices i and j are not adjacent, and equal to one whenever the vertices i and j are adjacent.

Suppose $A(x) := M + ee^T$ where x denotes the vector consisting of strictly lower triangular entries of M and $e = [1 \dots 1]^T$. Then, the Shannon Capacity [24, 29] of the graph is defined as

$$\min_{x} \lambda_{\max}(A(x)), \tag{3.14}$$

which is an upper bound on the amount of information that can be reliably transmitted over the given graph M.

3.6 Semidefinite Programming

We consider the problem of minimizing a linear function depending on $x \in \mathbb{R}^m$ subject to a matrix inequality of the form

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & F(x) \succeq 0 \end{aligned}$$

where $F(x) = F_0 + \sum_{i=1}^{m} x_i F_i$, $c \in \mathbb{R}^m$ and $F_0, \ldots, F_m \in \mathbb{R}^{n \times n}$ are symmetric matrices.

Above $F(x) \succeq 0$ means $F(x)$ is positive semi-definite, or equivalently its smallest eigenvalue is non-negative. The *logarithmic barrier method* to solve this constrained optimization problem will focus on

$$\min_{x \in \mathbb{R}^m} L(x, \mu)$$

where $L(x, \mu) = c^T x - \mu \cdot \ln(\lambda_{\min}(F(x)))$ and μ is a small positive real number. Throughout the rest of this section we will present specific semidefinite programming problems.

3.6.1 Logarithmic Chebyshev Approximation

Consider the problem of approximating the solution of $Ax = b$, where $A = [a_1 \ldots a_p]^T \in \mathbb{R}^{p \times k}$ and $b \in \mathbb{R}^p$. In logarithmic Chebyshev approximation we minimize the ℓ_∞-norm of the residual

$$\max_i |\log(a_i^T x) - \log(b_i)| \tag{3.15}$$

where we assume that $b_i > 0$. Equation (3.15) can be cast as a semidefinite programming problem of the form [25]

$$\text{min } t, \qquad \text{subject to } \begin{bmatrix} t - a_i^T x/b_i & 0 & 0 \\ 0 & a_i^T x/b_i & 1 \\ 0 & 1 & t \end{bmatrix} \succeq 0, \quad i = 1, \ldots, p.$$

3.6.2 Geometrical problems involving quadratic forms

Many geometrical problems involving quadratic functions can be cast as semidefinite programs. We will focus on only one example. Consider the quadratic functions

$$f_i(x) = x^T A_i x + 2b_i^T x + c, \quad i = 1, \ldots, k,$$

which are ellipsoids. Our goal is to find the smallest ball that contains all k of these ellipsoids. We can find the smallest ball by solving the semidefinite programming problem [25]

$$\begin{array}{ll} \min & t \\ \text{subject to} & \begin{bmatrix} I & -x_c \\ x_c^T\gamma & 0 \end{bmatrix} \preceq \eta_i \begin{bmatrix} A_i & b_i \\ b_i^T & c_i \end{bmatrix}, \quad i = 1, \ldots, k \\ & \eta_i \geq 0, \quad i = 1, \ldots, k \\ & \begin{bmatrix} I_n & x_c \\ x_c^T & t+\gamma \end{bmatrix} \succeq 0 \end{array}$$

where the variables are $x_c, \eta_1, \ldots, \eta_k, \gamma$ and t.

3.6.3 Structural optimization

Consider a structure consisting of k elastic bars connected at a set of p nodes. The task is to determine the optimal cross-sectional areas for the bars. For simplicity, assume that the external nodal forces $f_i,\ i = 1, \ldots, p$ are fixed, let d denote the displacement vector of the nodes caused by the nodal forces f. The objective is to minimize the elastic energy stored $\frac{1}{2}f^T d$ with the constraints on the total volume, upper and lower bounds on the cross-sectional area of each bar. The relation between f and d is linear, that is $f = A(x)d$ where $A(x) = \sum_{i=1}^k x_i A_i$ for some $A_i \in \mathbb{C}^{p\times p}$ for $i = 1, \ldots, k$.

The optimization problem can be posed as

$$\begin{array}{ll} \text{minimize} & f^T d \\ \text{subject to} & f = A(x)d \\ & \sum_{i=1}^k \ell_i x_i \leq v \\ & \underline{x}_i \leq x_i \leq \bar{x}_i, \quad i = 1, \ldots, k \end{array}$$

where d and x are the variables, v is the maximal total volume of the elastic bars, ℓ_i are the lengths of the bars, and $\underline{x}_i, \bar{x}_i$ are the lower and upper bounds on the cross-sectional areas (see [1] for details).

By eliminating d, the structural problem can be posed as a semidefinite programming problem of the form

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \begin{bmatrix} t & f^T \\ f & A(x) \end{bmatrix} \succeq 0 \\ & \sum_{i=1}^{k} \ell_i x_i \leq v \\ & \underline{x}_i \leq x_i \leq \bar{x}_i, \quad i = 1, \ldots, k \end{array}$$

in terms of the variables x and t.

Chapter 4

Existing Numerical Algorithms

This chapter presents a survey on the widely-used algorithms for some of the elementary eigenvalue optimization problems. Specifically we will focus on the algorithms for the distance to instability, distance to uncontrollability, H_∞-norm and the numerical radius in this order.

4.1 Distance to Instability

Here we describe some of the widely-used algorithms to compute the distance to instability. One of the oldest algorithms is due to Van Loan [8]. But since it is heuristic-based, and not able to return estimates with a prescribed accuracy, we skip Van Loan's algorithm. Instead we consider the algorithms by Byers [37], Boyd and Balakrishnan [43], and the recent Newton-based algorithm of Freitag and Spence [26]

4.1.1 Byers' Method

Distance to instability is a one dimensional unconstrained real global optimization problem of the form

$$\min_{\omega \in \mathbb{R}} f(\omega)$$

where $f(\omega) := \sigma_{\min}(\mathcal{A} - \omega i I_n)$. The bisection method due to Byers that we present in this section gives upper bound and lower bound for the globally minimal value of $f(\omega)$, and it converges to the globally minimal value of $f(\omega)$ even when f has multiple local extrema. However, it converges slowly.

This algorithm heavily depends on the Hamiltonian matrix

$$H(\sigma) = \begin{bmatrix} \mathcal{A} & -\sigma I_n \\ \sigma I_n & -\mathcal{A}^* \end{bmatrix} \in \mathbb{C}^{2n \times 2n}$$

where $\sigma \geq 0$ and $\mathcal{A}^*$ denotes the adjoint (complex conjugate transpose) of $\mathcal{A}$. Let us also use the notation $\mathbb{C}^0 := \{\lambda \in \mathbb{C} : \Re(\lambda) = 0\}$ for the imaginary axis in the next theorem.

Theorem 4.1.1 (Byers)**.** *There exists* $\lambda_j \in (\Lambda(H(\sigma)) \cap \mathbb{C}^0)$ *if and only if* $\sigma \geq \beta(\mathcal{A})$.

Proof. ($\Rightarrow$) Assume $\lambda_j \in (\Lambda(H(\sigma)) \cap \mathbb{C}^0)$ meaning $\lambda_j = i\eta$ for some $\eta \in \mathbb{R}$. By the definition of an eigenvalue, there exists a nonzero vector $u \in \mathbb{C}^{2n}$ such that

$$H(\sigma)u = \eta i u. \tag{4.1}$$

If we partition u into vectors $u_1,\ u_2 \in \mathbb{C}^n$, (4.1) can be expressed as

$$\begin{bmatrix} \mathcal{A} & -\sigma I_n \\ \sigma I_n & -\mathcal{A}^* \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \eta i \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{4.2}$$

By simple calculations, (4.2) yields

$$\begin{aligned} (\mathcal{A} - \eta i I_n)u_1 &= \sigma u_2, \quad \text{and} \\ (\mathcal{A} - \eta i I_n)^* u_2 &= \sigma u_1, \end{aligned} \tag{4.3}$$

implying that σ is a singular value of $(\mathcal{A} - \eta i I_n)$. So, for some integer $k \in [1, n]$ we have $\sigma = \sigma_k(\mathcal{A} - \eta i I_n) \geq \sigma_{\min}(\mathcal{A} - \eta i I_n) \geq \min_\eta \sigma_{\min}(\mathcal{A} - \eta i I_n) = \beta(\mathcal{A})$.

($\Leftarrow$) Assume $\sigma \geq \beta(\mathcal{A})$. Now consider $f(\omega) = \sigma_{\min}(\mathcal{A} - \omega i I_n)$, which is a continuous positive real-valued function, and it is unbounded above as $\omega \to \infty$. Since $\sigma \geq \beta(\mathcal{A}) \geq 0$, by continuity of f there exists $\eta \in \mathbb{R}$ such that $f(\eta) = \sigma$. That means σ is a singular value of $(\mathcal{A} - \eta i I_n)$, that is there exist unit vectors $u_1, u_2 \in \mathbb{C}^n$ such that $(\mathcal{A} - \eta i I_n)u_1 = \sigma u_2$, and $(\mathcal{A} - \eta i I_n)^* u_2 = \sigma u_1$. By reversing the steps that yield (4.3) from (4.2), we deduce $H(\sigma)u = \eta i u$ where $u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \in \mathbb{C}^{2n}$. □

Corollary 4.1.2. *There exists $\omega \in \mathbb{R}$ such that $i\omega \in \Lambda(H(\sigma))$ if and only if $\sigma_k(\mathcal{A} - i\omega I_n) = \sigma$ for some $k \in [1, n]$.*

The Bisection Algorithm is simply based on Theorem 4.1.1. It uses naive lower and upper bounds, namely $\beta(\mathcal{A}) \geq 0$ and $\beta(\mathcal{A}) \leq \frac{1}{2}\|\mathcal{A} + \mathcal{A}^*\|_2$, respectively

Algorithm 1 Bisection Algorithm

1: **INPUT**: $\mathcal{A} \in \mathbb{C}^{n\times n}$ stable matrix, desired tolerance:$\epsilon > 0$
2: Initialization: $\alpha \leftarrow 0$, and $\gamma \leftarrow \frac{1}{2}\|\mathcal{A} + \mathcal{A}^*\|_2$
3: $\sigma \leftarrow \sqrt{\gamma \max(\epsilon, \alpha)}$
4: While $\gamma > 10\max(\epsilon, \alpha)$ do
5: **loop**
6: **if** $H(\sigma)$ has an eigenvalue with zero real part **then**
7: $\gamma \leftarrow \sigma$
8: **else**
9: $\alpha \leftarrow \sigma$
10: **end if**
11: **end loop**
12: **OUTPUT** α, γ such that $\beta(\mathcal{A}) \in [\alpha, \gamma]$.

Algorithm 1 starts with these naive bounds $\alpha = 0$ and $\gamma = \frac{1}{2}\|\mathcal{A} + \mathcal{A}^*\|_2$ initially. Then it chooses a point $\sigma \in (\alpha, \gamma)$. By using Theorem 4.1.1, it updates either α or γ. Finding the eigenvalues of $H(\sigma)$ is the most expensive computational step above. On exit, it returns σ which is an approximation for the distance to instability.

4.1.2 Boyd and Balakrisnan's Method

Originally, Boyd and Balakrishnan's method was used to compute the H_∞-norm of a given transfer function. However, by simple modification, it can be adopted to compute the distance to instability of a given stable matrix $\mathcal{A} \in \mathbb{C}^{n\times n}$.

The Boyd and Balakrishnan method is simply based on the Byers' results specifically Corollary 4.1.2, level sets and frequency intervals. It finds all pure imaginary eigenvalues of $H(\sigma)$. Let us label these eigenvalues as $i\lambda_1, i\lambda_2, \ldots, i\lambda_k$ such that $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_k$. By Corollary 4.1.2 these imaginary eigenvalues would yield all λ such that $\sigma = \sigma_{\min}(\mathcal{A} - \lambda i I_n)$. Then the algorithm determines the frequency intervals $\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_\ell$ in which $f(\lambda) = \sigma_{\min}(\mathcal{A} - \lambda i I_n) \leq \sigma$. The next estimate for $\beta(\mathcal{A})$ is given by

$$\sigma = \min_{j=1,\ldots,\ell} \sigma_{\min}(\mathcal{A} - s_j i I_n)$$

where s_j is the midpoint of the interval $\mathcal{I}_j$.

Algorithm 2 Boyd and Balakrishnan Algorithm

1: **INPUT**: $\mathcal{A} \in \mathbb{C}^{n\times n}$ stable matrix, desired tolerance:$\epsilon > 0$
2: Initialization: $\sigma \leftarrow \frac{1}{2}\|\mathcal{A} + \mathcal{A}^*\|_2$
3: Compute all pure imaginary eigenvalues $i\lambda_1, i\lambda_2, \ldots, i\lambda_k$ of $H(\sigma)$ where $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_k$.
4: Find all frequency interval $\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_\ell$ in which $\sigma_{\min}(\mathcal{A} - \lambda i I_n) \leq \sigma$.
5: Define s_j as the midpoint of $\mathcal{I}_j$ for $j = 1, \ldots, \ell$
6: Update $\sigma = \min_{j=1,\ldots,\ell} \sigma_{\min}(\mathcal{A} - s_j i I_n) - \epsilon$
7: Repeat **step 3-7** until convergence (Terminate if no such λ is found at step 3)
8: **OUTPUT** $\sigma \in [\beta(\mathcal{A}) - \epsilon, \beta(\mathcal{A})]$

The Boyd and Balakrishnan algorithm converges quadratically, however it has some disadvantages. At each step, as in Byers' Algorithm, it needs to solve a $2n \times 2n$ eigenvalue problem. Also, it needs to solve singular value decompositions for $n \times n$ matrices to determine σ. As n gets larger, these calculations become expensive. He and Watson's [9] algorithm attempts to reduce the cost of $2n \times 2n$ eigenvalue problem by using *inverse power iteration*, and their algorithm works efficiently especially for

sparse matrices.

4.1.3 Freitag and Spence's Method

Freitag and Spence have recently devised a new quadratically convergent algorithm [26]. A function h, seemingly not connected to $f(\omega) = \sigma_{\min}(\mathcal{A} - \omega i I_n)$, will be derived in this section such that its globally minimal value is the distance to instability. Newton's method is used to determine a locally optimal solution of h. The converged local minimizer will be checked to see whether it is indeed a global minimizer by means of Theorem 4.1.1. If it is not a global minimizer then a new initial point will be determined without any significant work.

The rest of this section provides a brief description of the algorithm. The algorithm is based on the following observation. Suppose ω_* is a local minimizer of $f(\omega) = \sigma_{\min}(\mathcal{A} - \omega i I_n)$ and $\sigma_* = \sigma_{\min}(\mathcal{A} - \omega i I_n)$. Then typically $H(\sigma_*)$ has $i\omega_*$ as a multiple defective eigenvalue, and $H(\sigma)$ does not have any imaginary eigenvalue for $\sigma < \sigma_*$ provided σ_* is the globally minimal value. Let σ_* denote the smallest σ such that $H(\sigma)$ has a pure imaginary eigenvalue. Denote this imaginary eigenvalue with $i\lambda_*$ and associated the right eigenvector $\underline{x}$. Assume that $i\lambda_*$ is a defective eigenvalue with algebraic multiplicity two.

Then it can be shown that

$$\begin{bmatrix} H(\sigma_*) - i\lambda_* & Jc \\ c^* & 0 \end{bmatrix} \text{ where } J = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}$$

is nonsingular for all $c \in \mathbb{C}^n$ such that $c^*\underline{x} \neq 0$. For the proof we refer to [26]. Therefore the linear system

$$\left(M(\sigma, \lambda) := \begin{bmatrix} H(\sigma) - i\lambda & Jc \\ c^* & 0 \end{bmatrix} \right) \begin{bmatrix} x \\ h \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{4.4}$$

has a unique solution near (σ_*, λ_*). Observe that x and h are dependent on λ and σ. Indeed by Cramer's Rule

$$h(\sigma, \lambda) = \frac{\det(H(\sigma) - i\lambda)}{\det(M(\sigma, \lambda))}$$

implying

$$h(\sigma, \lambda) = 0 \Leftrightarrow det(H(\sigma) - i\lambda) = 0$$

Specifically, $h(\sigma_*, \lambda_*) = 0$. Furthermore, under the assumptions that (i) $i\lambda_*$ is a defective eigenvalue, (ii) $c \in \mathbb{C}^n$ is such that $c^*\underline{x} \neq 0$, and (iii) $\mathcal{A}$ is a stable matrix, it can be deduced that $h_\lambda(\sigma_*, \lambda_*) = 0$.

Let

$$g(\sigma, \lambda) := \begin{bmatrix} h(\sigma, \lambda) \\ h_\lambda(\sigma, \lambda) \end{bmatrix},$$

then (σ_*, λ_*) is a root of $g(\sigma, \lambda)$. Since we intend to use Newton's Method to find a root of $g(\sigma, \lambda)$, each iteration of Newton's method requires the solution p of the linear system $g'(\sigma, \lambda)p = -g(\sigma, \lambda)$. Therefore we need to evaluate $g(\sigma, \lambda)$ and its Jacobian

$$g'(\sigma, \lambda) = \begin{bmatrix} h_\sigma(\sigma, \lambda) & h_\lambda(\sigma, \lambda) \\ h_{\lambda\sigma}(\sigma, \lambda) & h_{\lambda\lambda}(\sigma, \lambda) \end{bmatrix}.$$

The function values $g(\sigma, \lambda)$ and its derivatives can be constructed by the following steps. All these expressions can be obtained by differentiating (4.4) repeatedly.

i) The function values $x(\sigma, \lambda), h(\sigma, \lambda)$ can be found by solving

$$M(\sigma, \lambda) \begin{bmatrix} x \\ h \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

ii) The derivatives with respect to λ $x_\lambda(\sigma,\lambda), h_\lambda(\sigma,\lambda)$ can be found by solving

$$M(\sigma,\lambda)\begin{bmatrix} x_\lambda \\ h_\lambda \end{bmatrix} = \begin{bmatrix} ix \\ 0 \end{bmatrix}.$$

iii) The derivatives with respect to λ $x_\sigma(\sigma,\lambda), h_\sigma(\sigma,\lambda)$ can be found by solving

$$M(\sigma,\lambda)\begin{bmatrix} x_\sigma \\ h_\sigma \end{bmatrix} = \begin{bmatrix} Jx \\ 0 \end{bmatrix}.$$

iv) The second derivatives with respect to λ $x_{\lambda\lambda}(\sigma,\lambda), h_{\lambda\lambda}(\sigma,\lambda)$ can be found by solving

$$M(\sigma,\lambda)\begin{bmatrix} x_{\lambda\lambda} \\ h_{\lambda\lambda} \end{bmatrix} = \begin{bmatrix} 2ix_\lambda \\ 0 \end{bmatrix}.$$

v) The second derivatives with respect to σ $x_{\lambda\sigma}(\sigma,\lambda), h_{\lambda\sigma}(\sigma,\lambda)$ can be found by solving

$$M(\sigma,\lambda)\begin{bmatrix} x_{\lambda\sigma} \\ h_{\lambda\sigma} \end{bmatrix} = \begin{bmatrix} Jx_\lambda + ix_\sigma \\ 0 \end{bmatrix}.$$

Note also that it can be shown that $f(\sigma,\lambda)$ satisfying (4.4) is a real for all $\sigma,\lambda \in \mathbb{R}$. Therefore our problem is a *real* root finding problem.

The computation of an LU-factorization of the $(2n+1)\times(2n+1)$ matrix $M(\sigma,\lambda)$ requires $\frac{16}{3}n^3$ flops, and this dominates the work needed for forward and back substitutions. If we need m Newton steps for convergence, then the algorithm requires $m\frac{16}{3}n^3$ flops in addition to the work to solve $2n\times 2n$ eigenvalue problems.

Algorithm 3 Freitag and Spence Algorithm

1: **INPUT**: Initial points (σ_0, λ_0), $c \in \mathbb{C}$ such that $M(\sigma_0, \lambda_0)$ is nonsingular, desired tolerance $\epsilon > 0$ and maximum number of iterations $maxiter \in \mathbb{Z}^+$
2: $k \leftarrow 0$
3: While { *desired accuracy is satisfied* **and** $k \leq$ *maxiter* } do
4: **loop**
5: $[L_k, U_k] \leftarrow LU - factorization\ of\ M(\sigma_k, \lambda_k)$
6: $(x^k, h^k) \leftarrow$Solve $L_k U_k \begin{bmatrix} x \\ h \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ by forward and back substitution
7: $(x^k_\lambda, h^k_\lambda) \leftarrow$Solve $L_k U_k \begin{bmatrix} x_\lambda \\ h_\lambda \end{bmatrix} = \begin{bmatrix} ix^k \\ 0 \end{bmatrix}$ by forward and back substitution
8: $g_k \leftarrow \begin{bmatrix} h^k \\ h^k_\lambda \end{bmatrix}$
9: $(x^k_\sigma, h^k_\sigma) \leftarrow$Solve $L_k U_k \begin{bmatrix} x_\sigma \\ h_\sigma \end{bmatrix} = \begin{bmatrix} Jx^k \\ 0 \end{bmatrix}$ by forward and back substitution
10: $(x^k_{\lambda\lambda}, h^k_{\lambda\lambda}) \leftarrow$Solve $L_k U_k \begin{bmatrix} x_{\lambda\lambda} \\ h_{\lambda\lambda} \end{bmatrix} = \begin{bmatrix} 2ix^k_\lambda \\ 0 \end{bmatrix}$ by forward and back substitution
11: $(x^k_{\lambda\sigma}, h^k_{\lambda\sigma}) \leftarrow$Solve $L_k U_k \begin{bmatrix} x_{\lambda\sigma} \\ h_{\lambda\sigma} \end{bmatrix} = \begin{bmatrix} Jx^k_\lambda + ix^k_\sigma \\ 0 \end{bmatrix}$ by forward and back substitution
12: $G_k \leftarrow \begin{bmatrix} h^k_\sigma & h^k_\lambda \\ h^k_{\lambda\sigma} & h^k_{\lambda\lambda} \end{bmatrix}$
13: $p_k \leftarrow -(G_k) \backslash g_k$
14: $(\sigma^{k+1}, \lambda^{k+1}) \leftarrow (\sigma^k, \lambda^k) + p_k$
15: $k \leftarrow k + 1$.
16: **end loop**
17: **if** $H(\sigma_k(1-\epsilon))$ does have a pure imaginary eigenvalue **then**
18: **GO TO step 2**
19: **else**
20: return $\sigma_* \leftarrow \sigma_k$
21: **end if**
22: **OUTPUT** (σ_*, λ_*) where $\beta(\mathcal{A}) \in [\sigma_k(1-\epsilon), \sigma_k]$.

4.2 Distance to Uncontrollability

In this section we present Byers' method [38], which constructs one dimensional and two dimensional grids to minimize $\sigma_{\min}([A - \lambda I_n, B])$, Gu's bisection algorithm [31], which is the first algorithm that computes the globally minimal value of $\sigma_{\min}([A - \lambda I_n,\ B])$ within a factor of two without depending on a grid, Burke, Lewis and Overton's trisection algorithm [21], which computes $\mu(A, B)$ to any desired precision, and Gu, Mengi, Overton, Xia and Zhu's algorithm[30], which uses inverse iteration and shift-and-invert preconditioned Arnoldi to reduce the complexity of the trisection algorithm.

4.2.1 Byers' Method

i) Algorithm based on two dimensional optimization problem: Distance to uncontrollability is a nonconvex global eigenvalue optimization problem of the form

$$\min_{\lambda} f(\lambda) \quad \text{where } f(\lambda) := \sigma_{\min}([A - \lambda I_n, B]).$$

Byers' method is a grid-based algorithm. It evaluates $f(\lambda)$ so many times so that it needs an efficient way for these function evaluations. Observe that for any unitary matrix $Q \in \mathbb{C}^{n\times n}$, we have

$$\sigma_{\min}([A - \lambda I_n, B]) = \sigma_{\min}\left(Q^*[A - \lambda I_n, B]\begin{bmatrix} Q & 0 \\ 0 & I_n \end{bmatrix}\right) = \sigma_{\min}([Q^*AQ - \lambda I_n, Q^*B]).$$

Therefore Byers' algorithm initially computes a Hessenberg factorization of A. Let $\tilde{A} = Q^*AQ$ be Hessenberg and $\tilde{B} = Q^*B$, then unitary matrices U, V in terms of Givens' rotators can be constructed at a cost of $O(n^2m)$ such that $U[\tilde{A} - \lambda I_n, \tilde{B}]V = [R,\ 0]$ is upper triangular. Therefore evaluating $f(\lambda)$ costs $O(n^2m)$ after finding Q initially as described above.

Instead of searching all complex plane, we will search for λ, which is an estimation to a global minimizer of $f(\lambda)$, over $|\lambda| \leq 2(\|[A \quad B]\|_2)$. This observation is a direct corollary of Wielandt-Hoffman theorem applied to the singular value decomposition of $[A - \lambda I_n,\ B]$. We divide the region into squares with side-lengths equal to $\epsilon\sqrt{2}$ to satisfy ϵ accuracy. Since the function is Lipschitz continuous with Lipschitz constant one, the difference between $f(\lambda)$ and $f(\lambda+\delta\lambda)$ cannot exceed $|\delta\lambda|$. Consequently, it is sufficient to evaluate the function only at the grid points. The prescribed ϵ-accuracy is ensured as the distance of the global minimizer to one of the grid-points is ϵ or smaller. Unfortunately, this requires too many evaluations of $f(\lambda)$, e.g., for $\epsilon = 10^{-3}$ we need $\frac{10^6}{2}$ evaluations. Even though this method is reliable, the number of function evaluations needs to be reduced.

Let

$$g(z) := \lambda_{\min}\left([A - zI_n \ B] \begin{bmatrix} A^* - \bar{z}I_n \\ B^* \end{bmatrix} \right).$$

Rellich's work shows that $g(z)$ is analytic everywhere except at a discrete set of values of z, and the second derivative satisfies $|g''(z)| \leq 2$ whenever the function is analytic. Let $h(z)$ be the quadratic function interpolating $g(z)$ at $z_1, z_2, z_3 \in \mathbb{C}$. If z is inside the simplex with vertices z_1, z_2, z_3, then it can be shown that $g^2(z) \geq h(z)$, implying $\min_z g(z) \geq \sqrt{\min_z h(z)}$. By this action we can construct a lower bound for $\min_z g(z)$ at every step efficiently, since evaluating the right hand side is a trivial quadratic programming problem.

Let $\mathbb{S}$ denote the set of defined simplices, and ℓ_S denote a lower bound for $\min_z g(z)$ over a given simplex S. The algorithm keeps a set of simplices which may contain the global minimizer of g. An upper bound for the distance to uncontrollability is given by the square root of the smallest value of $g(z)$ over all defined vertices. Also, for each simplex, the algorithm finds a lower bound ℓ_S, and discards the simplices which cannot contain a global minimizer, i.e., its lower bound ℓ_S is greater than the upper

bound provided by the function values. The feasible simplices are subdivided until the upper bound and lower bound are within the desired tolerance.

Algorithm 4 Two dimensional Algorithm

1: **INPUT**: $A \in \mathbb{C}^{n\times n}$, $B \in \mathbb{C}^{n\times m}$, desired tolerance $\epsilon > 0$
2: Initialization: $z \leftarrow 2\|[A,B]\|_2$; $\mathbb{S} \leftarrow \{(-z-iz, -z+iz, z+iz), (-z-iz, z-iz, z+iz)\}$; $lb = \{\ell_S : S \in \mathbb{S}\}$; $ub = \min\{g(z), g(zi), g(-zi), g(-z)\}$
3: While $ub - lb > \epsilon$ do
4: **loop**
5: Define $T \leftarrow \{S \in \mathbb{S} : \ell_S \leq ub\}$;
6: $v \leftarrow \min_{s\in T} g(w_s)$, where w_s is the center of the simplex S;
7: $ub \leftarrow \min\{v, ub\}$;
8: $U \leftarrow \{S \in T : \ell_S \leq ub\}$;
9: $\mathbb{S} \leftarrow \{$ divide all $S \in U$ into three by introducing the vertex $w_S\}$;
10: Calculate ℓ_S for all simplices in $\mathbb{S}$.
11: $l_b \leftarrow \min\{\ell_S : S \in \mathbb{S}\}$;
12: **end loop**
13: **OUTPUT** $lb, ub \in \mathbb{R}$ such that $\sqrt{ub} \geq \mu(A,B) \geq \sqrt{lb}$ with $ub - lb \geq \epsilon$.

Observe that subdivisions at an iteration cost only $O(n^2m) \times$ number of simplices. However long and thin simplices may yield numerical problems. Also it is hard to predict the shape of the simplices beforehand.

ii) Algorithm based on one dimensional optimization problem: We have a two dimensional optimization problem of the form

$$\min_{x,y} \sigma_{\min}\left([A - (x+iy)I_n, B]\right).$$

Furthermore our feasible region $F = \{(x,y) : x, y \leq 2\|[A,B]\|_2\}$ is bounded meaning

$$\min_{x,y} \sigma_{\min}\left([A - (x+iy)I_n, B]\right) = \min_{x} \min_{y} \sigma_{\min}\left([A - (x+iy)I_n, B]\right).$$

In what follows we use the notation $h_1(x) := \min_y \sigma_{\min}\left([A - (x+iy)I_n, B]\right)$ and $\mu(A,B) := \min_x h_1(x)$.

Theorem 4.2.1. *Let $A \in \mathbb{C}^{n\times n}$ and $B \in \mathbb{C}^{n\times m}$. For all $\sigma > 0$, the inequality $\sigma \geq \mu(A, B)$ holds if and only if there exist $x, y \in \mathbb{R}$ such that $det(G-\sigma I_n - xK - iyL) = 0$ where*

$$G = \begin{bmatrix} 0 & 0 & A^* \\ 0 & 0 & B^* \\ A & B & 0 \end{bmatrix},\ K = \begin{bmatrix} 0 & 0 & I_n \\ 0 & 0_{mm} & 0 \\ I_n & 0 & 0 \end{bmatrix}, \text{and}\ \ L = \begin{bmatrix} 0 & 0 & -I_n \\ 0 & 0_{mm} & 0 \\ I_n & 0 & 0 \end{bmatrix}$$

The proof of Theorem 4.2.1 follows by rearranging $\det(G - \sigma I_n - xK - iyL) = 0$ so that

$$\det\left(\begin{bmatrix} 0 & 0 & (A-\lambda I_n)^* \\ 0 & 0 & B^* \\ (A-\lambda I_n) & B & 0 \end{bmatrix} - \sigma I_n\right) = 0$$

where $\lambda = x + iy$. For any $x \in \mathbb{R}$ and $\sigma \in \mathbb{R}^+$, this theorem also implies that $\sigma > h_1(x)$ if and only if $(G - \sigma I_n - xK) - \lambda L$ has a pure imaginary generalized eigenvalue. The function $h_1(x)$ can be evaluated based on this observation by means of a bisection method. Unfortunately, the solution of the generalized eigenvalue problem costs $O(n^3)$, and bisection method is not meant for the calculation of $h_1(x)$ with high accuracy. A helpful observation is that we can use a modified version of the Boyd and Balakrishnan method to evaluate $h_1(x)$ based on extracting the imaginary eigenvalues of the pencil $\mathcal{L}(\lambda) := (G-\sigma I_n - xK) - \lambda L$. Still the method is too expensive, therefore interpolation techniques are used in the two-dimensional case. Specifically, the facts

$$\left|\frac{d(h_1(x)^2)}{dx}\right| \leq 1, \qquad \left|\frac{d^2(h_1(x)^2)}{dx^2}\right| \leq 2$$

are exploited. It follows from these facts that if $f(x) = a + bx + cx^2$ interpolates $h_1(x)^2$ at two points x_1, x_2 such that $x_1 < x_2$, then $h_1(x)^2 \geq f(x)$ for all $x \in [x_1, x_2]$.

Algorithm 5 below keeps a set of intervals each of which may contain a global

minimizer of h_1, a lower bound for each of these intervals, a lower bound (lb) and a upper bound (ub) for the function $h_1(x)$.

Algorithm 5 One dimensional Byers' Algorithm

INPUT: $A \in \mathbb{C}^{n\times n}$, $B \in \mathbb{C}^{n\times m}$ desired tolerance: $\epsilon > 0$
Initialization: $z \leftarrow 2\|[A, B]\|_2$; $\mathbb{S} \leftarrow \{[-z, z]\}$; $lb = 0$; $ub = \min\{h_1(-z), h_1(z)\}$
While $ub - lb > \epsilon$ do
loop
  $U \leftarrow \{l \in \mathbb{S} : \ell_l \leq ub\}$;
  $v \leftarrow \min_{l\in U} h_1(\omega_l)$ where ω_l is the midpoint of the lth interval;
  $ub \leftarrow \min\{ub, v\}$;
  $\mathbb{S} = \left\{\left[a_l, \frac{a_l+b_l}{2}\right], \left[\frac{a_l+b_l}{2}, b_l\right] : \forall [a_l, b_l] \in U\right\}$;
  calculate ℓ_l for each $l \in \mathbb{S}$;
  $lb \leftarrow \min\{\ell_l : l \in \mathbb{S}\}$;
end loop
OUTPUT $lb, ub \in \mathbb{R}$ such that $ub \geq \mu(A, B) \geq lb$ with $ub - lb \leq \epsilon$.

Two algorithms mentioned above due to Byers are expensive. On the other hand, they give approximate values for $\mu(A, B)$ with any desired accuracy.

4.2.2 Gu's bisection Method

Gu presented a quadratically convergent algorithm based on generalized eigenvalue problems that does not utilize a grid. Consider the objective function

$$f(x, y) := \sigma_{\min}([A - (x + iy)I_n, B])$$

again such that $\mu(A, B) = \min_{x,y} f(x, y)$.

Theorem 4.2.2 (Gu)**.** *Let $\sigma > \mu(A, B)$. Then there exist x, y such that*

$$f(x, y) = f(x + \eta, y) = \sigma \tag{4.5}$$

for all $\eta \in (0, 2(\sigma - \mu(A, B))]$.

Next we will outline the derivation of a generalized eigenvalue problem such that

x is a real eigenvalue of this problem, if there exists (x, y) satisfying (4.5). For a given x it can be shown that if $f(x, y) = \sigma$ for some y, then iy is a pure imaginary eigenvalue of a pencil $\mathcal{L}(\lambda) := \mathcal{A}(\sigma, x) - \lambda\mathcal{B}(\sigma, x)$. Thus to check the existence of a pair (x, y) satisfying (4.5) in Theorem 4.2.2 we need to determine whether two pencils $\mathcal{A}(\sigma, x) - \lambda\mathcal{B}(\sigma, x)$ and $\mathcal{A}(\sigma, x + \eta) - \lambda\mathcal{B}(\sigma, x + \eta)$ share a common pure imaginary eigenvalue. Now suppose that

$$\mathcal{A}(\sigma, x)v_1 = iy\mathcal{B}(\sigma, x) \tag{4.6}$$

$$\mathcal{A}(\sigma, x + \eta)v_1 = iy\mathcal{B}(\sigma, x + \eta) \tag{4.7}$$

for some nonzero v_1, v_2. Then the matrix equation

$$\mathcal{A}(\sigma, x)X\mathcal{B}(\sigma, x + \eta) - \mathcal{B}(\sigma, x)X\mathcal{A}(\sigma, x + \eta) = 0 \tag{4.8}$$

must have a nonzero solution

$$X = \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix} \in \mathbb{R}^{2n\times 2n}.$$

By vectorizing the matrix equation above an eigenvalue problem of the form $\mathcal{C}u = x\mathcal{D}u$ of size $2n^2 \times 2n^2$ can be obtained. The pencil depends on σ as well as η. In while control of the algorithm, we set η to a small positive number, then we check if $\mathcal{C} - \lambda\mathcal{D}$ has a real eigenvalue σ. If it does, we will check whether the two pencils $\mathcal{A}(\sigma, x)-\lambda\mathcal{B}(\sigma, x)$ and $\mathcal{A}(\sigma, x+\eta)-\lambda\mathcal{B}(\sigma, x+\eta)$ share a common imaginary eigenvalue. If they have a common imaginary eigenvalue, then there exist x, y satisfies equation (4.5), which implies $\mu(A, B) \leq \sigma$. On the other hand if there is no such pair, then $\eta > 2(\sigma - \mu(A, B)$. This implies $\mu(A, B) > \sigma - \eta/2$. By using this fact, we will halve σ values repeatedly. Observe that, at termination, $\mu(A, B)$ satisfies the following

inequality

$$\sigma - \eta/2 < \mu(A, B) \leq 2\sigma.$$

One needs to set η sufficiently small to have higher precision. Unfortunately, this leads to numerical difficulties. For example, the comparison of imaginary eigenvalues of two pencils mentioned above, cannot be carried out without rounding errors.

Algorithm 6 Gu's Algorithm

1: **INPUT**: $A \in \mathbb{C}^{n\times n}$, $B \in \mathbb{C}^{n\times m}$
2: Initialization: $\sigma \leftarrow \sigma_{\min}([A, B])$
3: calculate a pair of real numbers (x_*, y_*) and η by using Gu's verification scheme.
4: While Gu's verification scheme succeeds do
5: **loop**
6: $\sigma \leftarrow \frac{\sigma}{2}$
7: **end loop**
8: **OUTPUT** $\sigma \in \mathbb{R}$ such that $\sigma - \frac{\eta}{2} < \mu(A, B) \leq 2\sigma$.

At each iteration while-control requires the solution of a $2n^2 \times 2n^2$ eigenvalue problem. The efficiency may be improved by benefitting from the sparse matrix computations and focusing only on the real eigenvalues of the pencil $\mathcal{C} - \lambda\mathcal{D}$ as we shall see in Section 4.2.4.

4.2.3 Burke, Lewis and Overton's Trisection Method

This section presents trisection algorithm, which is a variant of Gu's algorithm. Instead of using the midpoints the trisection method divides an interval into three and uses the one-third and two-third of the interval as the checkpoints.

The trisection algorithm starts with an upper bound (U) and a lower bound (L) on the distance to uncontrollability of a pair (A, B). At each iteration, two quantities are evaluated; δ_1 is the one-third of the interval, δ_2 is the two-third of the interval. Gu's verification scheme is applied with $\delta = \delta_1$, and $\eta = 2(\delta_1 - \delta_2)$. Theorem 4.2.2 implies that, if a pair (x, y) satisfying (4.5) exists for $\delta = \delta_1$, then $\mu(A, B) \leq \delta_1$. On the other hand, when no such pair exists, theinequality $\eta > 2(\delta_1 - \mu(A, B))$ is

satisfied. By simplifying this inequality, we have $\mu(A,B) > \delta_1 - \frac{\eta}{2} = \delta_2$. By using these facts, one of L and U will be updated repeatedly. At each iteration, this update reduces the feasible interval by a factor of 2/3.

Algorithm 7 Trisection Algorithm

INPUT: $A \in \mathbb{C}^{n\times n}$, $B \in \mathbb{C}^{n\times m}$, desired tolerance $\epsilon > 0$
Initialization: $L \leftarrow 0$ and $U \leftarrow \sigma_{\min}([A,B])$;
While $|U-L| > \epsilon$ do
loop
  $\delta_1 \leftarrow L + 2\frac{U-L}{3}$ and $\delta_2 \leftarrow L + \frac{U-L}{3}$
  perform Gu's verification test which is a direct consequence of Theorem 4.2.2 given in Section 4.2.2.
  if $\mu(A,B) \leq \delta_1$ **then**
    $U \leftarrow \delta_1$;
  end if
  if $\mu(A,B) > \delta_2$ **then**
    $L \leftarrow \delta_2$;
  end if
end loop
OUTPUT $L, U \in \mathbb{R}$ such that $L \leq \mu(A,B) \leq U$ with $|U-L| \leq \epsilon$.

4.2.4 Gu, Mengi, Overton, Xia and Zhu's Method

This paper presents a new fast verification scheme and two new real eigenvalue searching strategies to reduce the cost of Gu's method. In the new verification scheme it is still necessary to find the real eigenvalues of $2n^2 \times 2n^2$ matrices. So if the QR-algorithm is used, there would be no efficiency gain over Gu's method. Therefore this paper takes advantage of the preconditioned Arnoldi method, which costs $O(n^3)$, to find the eigenvalue closest to a given real number. For extracting the real eigenvalues "divide and conquer" and "adaptive progress" methods are suggested. It has been proven that extracting all real eigenvalues costs $O(n^4)$ for the divide and conquer method on average [30]. For the adaptive progress method, no such analysis has been performed. But, in practice, it is observed that the divide and conquer method is much more efficient.

We first describe the new verification scheme in detail, then we will present the adaptive progress and divide and conquer schemes for real eigenvalue extraction. According to Theorem 4.2.2, there exist x, y such that $\sigma_{\min}([A-(x+iy)I_n, B]) = \sigma_{\min}([A-(x+\eta+iy)I_n, B]) = \sigma$ for all $\eta \in (0, 2(\sigma-\mu(A,B)))$ meaning

$$\begin{aligned} [A-(x+iy)I_n,\ B]\begin{bmatrix} u_1 \\ v_1 \end{bmatrix} &= \sigma z_1 \\ \begin{bmatrix} A^*-(x-iy)I_n \\ B^* \end{bmatrix} z_1 &= \sigma \begin{bmatrix} u_1 \\ v_1 \end{bmatrix}, \end{aligned} \tag{4.9}$$

and

$$\begin{aligned} [A-(x+\eta+iy)I_n,\ B]\begin{bmatrix} u_2 \\ v_2 \end{bmatrix} &= \sigma z_2 \\ \begin{bmatrix} A^*-(x+\eta-iy)I_n \\ B^* \end{bmatrix} z_2 &= \sigma \begin{bmatrix} u_2 \\ v_2 \end{bmatrix} \end{aligned} \tag{4.10}$$

for some $u_1, u_2, z_1, z_2 \in \mathbb{C}^n$ and $v_1, v_2 \in \mathbb{C}^m$ not all zero.

By noting that $v_1 = \frac{1}{\sigma}B^*z_1$ and defining $\hat{B} := \frac{BB^*-\sigma I_n}{\sigma}$ from (4.9) we have

$$H(x)\begin{bmatrix} z_1 \\ u_1 \end{bmatrix} = iy\begin{bmatrix} z_1 \\ u_1 \end{bmatrix}, \quad \text{where } H(x) := \begin{bmatrix} -(A^*-xI) & \sigma I_n \\ \hat{B} & A-xI_n \end{bmatrix}.$$

Similarly, we can deduce from (4.10) that

$$H(x+\eta)\begin{bmatrix} z_2 \\ u_2 \end{bmatrix} = iy\begin{bmatrix} z_2 \\ u_2 \end{bmatrix}.$$

Therefore $H(x)$ and $-H(x+\eta)^*$ share the common pure eigenvalue iy implying that

the matrix equation

$$H(x)X + XH(x+\eta)^* = \begin{bmatrix} -(A^* - xI_n) & \sigma I_n \\ \hat{B} & A - xI \end{bmatrix} X + X \begin{bmatrix} -A^* - (x+\eta)I_n & \sigma I_n \\ \hat{B} & A - (x+\eta)I_n \end{bmatrix}^* = 0 \tag{4.11}$$

has a nonzero solution. Let $\mathrm{vec}(X)$ denote the vector formed by stacking the columns of X. We will benefit from the identity

$$\mathrm{vec}(AXB) = (B^T \otimes A)\mathrm{vec}(X)$$

where $\otimes$ denotes the *Kronecker product*. By the partitioning

$$X = \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix},$$

writing equation (4.11) in vectorial form, applying the identity above and rearranging, we obtain

$$\begin{bmatrix} -A_1^* - A_2^T & \sigma I_n & \sigma I_n & 0 \\ B_2^T & -A_1^* + \bar{A}_2 & 0 & \sigma I_n \\ B_1 & 0 & A_1 - A_2^T & \sigma I_n \\ 0 & B_1 & B_2^T & A_1 + \bar{A}_2 \end{bmatrix} \begin{bmatrix} \mathrm{vec}(X_{11}) \\ \mathrm{vec}(X_{12}) \\ \mathrm{vec}(X_{21}) \\ \mathrm{vec}(X_{22}) \end{bmatrix} = \begin{bmatrix} -2x\mathrm{vec}(X_{11}) \\ 0 \\ 0 \\ -2x\mathrm{vec}(X_{22}) \end{bmatrix}$$

where $A_1 = I_n \otimes A,\ A_2 = (A - \eta I_n) \otimes I_n,\ B_1 = I_n \otimes \hat{B},\ B_2 = \hat{B} \otimes I_n$ and $\bar{A}_2$ denotes the complex conjugate of A_2 entry-wise.

Eliminating $\mathrm{vec}(X_{12})$ and $\mathrm{vec}(X_{21})$ in the equation above yields the eigenvalue problem

$$\mathcal{A}v = xv, \text{ where } v = \begin{bmatrix} \mathrm{vec}(X_{11}) \\ \mathrm{vec}(X_{22}) \end{bmatrix} \tag{4.12}$$

and

$$\mathcal{A} = \frac{1}{2}\begin{bmatrix} A_1^* + A_2 & 0 \\ 0 & A_1 + \bar{A}_2 \end{bmatrix} - \frac{1}{2}\left(\begin{bmatrix} -\sigma I_n & -\sigma I_n \\ B_1 & B_2^T \end{bmatrix}\begin{bmatrix} -A_1^* + \bar{A}_2 & 0 \\ 0 & A_1 + A_2^T \end{bmatrix}^{-1}\begin{bmatrix} B_2^T & \sigma I_n \\ B_1 & \sigma I_n \end{bmatrix}\right).$$

To summarize we deduce the following; if there exists (x, y) satisfying (4.5) in Theorem 4.2.2, then $\mathcal{A}$ must have x as a real eigenvalue. For each real eigenvalue x of $\mathcal{A}$, we also need to check whether $H(x)$ and $H(x + \eta)$ have the same imaginary eigenvalue in order to ensure (4.5) in Theorem 4.2.2. The trisection algorithm can be applied based on this numerical verification. The real innovative idea in this algorithm is the efficient extraction of the real eigenvalues of $\mathcal{A}$ as described next.

By using the direct Sylvester equation solvers, for instance the one in [40], and the inverse iteration one can compute the eigenvalue of $\mathcal{A}$ closest to any $\sigma \in \mathbb{C}$ at a cost of $O(n^3)$. This is due to the special structure of $\mathcal{A}$, specifically due to the fact that it is derived from a Sylvester equation of size $2n \times 2n$. The details are presented in [30]. The algorithms that extract the real eigenvalues of $\mathcal{A}$ use this scheme for computing the eigenvalue closest to $\sigma \in \mathbb{C}$ as a basic block. Two algorithms, an adaptive progress and a divide and conquer algorithm, are presented below for real eigenvalue extraction.

Adaptive Progress: We assume that a positive number such that

$$d \leq \min_{\lambda_i, \lambda_j \ are \ distinct} |\lambda_i - \lambda_j|$$

is known a priori. Also, we assume that a bound D exists such that all real eigenvalues lie in $[-D, D]$. The algorithm starts with the rightmost point $\nu = D$ as a shift. By using the inverse iteration the eigenvalue λ closest to ν can be found. If

λ is real and if it is an eigenvalue that is discovered by the algorithm before, then there is no real eigenvalue in $(\lambda - d, \lambda]$ and $(\nu - |\lambda - \nu|, \nu]$, therefore the new shift can be chosen as $\nu = \nu - \max\{|\lambda - \nu|, d - |\lambda - \nu|\}$. If it is real and not discovered, the updated shift will be $\lambda - d$. Finally, if λ is not real, there are no eigenvalues inside two circles, centered at ν and λ with radius $|\lambda - \nu|$ and d, respectively. In this case ν is set to be the left most intersection point of these two circles with the real line.

Algorithm 8 Adaptive Progress

INPUT: $A \in \mathbb{C}^{n\times n}$, $D \in \mathbb{R}$ (the upper bound for the real eigenvalues), $d \in \mathbb{R}$ (the distance between two closest eigenvalues)
Initialization: $\nu \leftarrow D$, $\Lambda \leftarrow \{\}$;
While $\nu \geq -D$ do
loop
  Find the eigenvalue λ closest to ν by using the inverse iteration.
  if λ is real and discovered **then**
    $\nu \leftarrow \nu - max\{|\lambda - \nu|, d - |\lambda - \nu|\}$;
  end if
  if λ is real and not discovered **then**
    $\nu \leftarrow \lambda - d$;
    $\Lambda \leftarrow \Lambda \cup \{\lambda\}$;
  end if
  if $\lambda = \alpha + i\beta$ where β is nonzero **then**
    if $d \geq |\beta|$ **then** $\nu \leftarrow \min\{\alpha - \sqrt{d^2 - \beta^2}, \nu - |\nu - \lambda|\}$, **else** $\nu \leftarrow \nu - |\nu - \lambda|$;
  end if
end loop
OUTPUT Λ, the set of all real eigenvalue of the given matrix A.

There are some disadvantages in using the adaptive progress algorithm. The first one is that it requires d, the minimum distance between the two closest eigenvalues. For reliability d needs to be set very small and this action increases the number of iterations. The other disadvantage of the algorithm is the need for the upper bound D on the real eigenvalues. Again for reliability, the upper bound must be chosen very large, but this clearly degrades the efficiency.

Divide and Conquer: For this method, the distance d is not needed. However

upper and lower bounds for the real eigenvalues are required. The basic idea of the algorithm is that for a given interval find the eigenvalue closest to the midpoint of the interval. Then divide the interval into three subintervals, where subintervals are of the form *undiscovered, discovered* and *undiscovered* from left to the right. Then call the algorithm again for the undiscovered parts. The discovered subinterval corresponds to an interval where it is known that no eigenvalues lie. This recursion will continue until the whole interval is completely discovered. It has been proven that extracting all of the real eigenvalues requires $O(n^4)$ operations on average and $O(n^5)$ operations in the worst case [30].

Algorithm 9 Divide and Conquer

1: **INPUT**: $A \in \mathbb{C}^{n\times n}$, $L, U \in \mathbb{R}$ (the lower and upper bound for the real eigenvalues, respectively)
2: Initialization: $\nu \leftarrow \frac{U+L}{2}$;
3: Find the eigenvalue λ closest to ν by using the inverse iteration.
4: **if** $U - L < 2|\lambda - \nu|$ **then**
5: interval contains no eigenvalue **return** { }.
6: **else**
7: $\Lambda_{left} \leftarrow$ call divide-and-conquer algorithm recursively with L_n, U_n as the lower and upper bounds where $L_n = L$ and $U_n = \nu - |\lambda - \nu|$;
8: $\Lambda_{right} \leftarrow$ call divide-and-conquer algorithm recursively with lower bound $L_n = \nu + |\lambda\nu|$ and upper bound $U_n = U$.
9: **if** λ is real **then** return $(\lambda \cup \Lambda_{left} \cup \Lambda_{right})$ **else** return $(\Lambda_{left} \cup \Lambda_{right})$
10: **OUTPUT** Λ, the set of all real eigenvalues of the matrix A.
11: **end if**

4.3 H_∞ norm

In this section, we start with the bisection method by Boyd, Balakrishnan and Kabamba [42]. (See [18] for a similar algorithm due to Robel). Then we present Boyd and Balakrishnan's quadratically convergent algorithm [43], which is widely-used. Finally, we state how Bruinsma and Steinbuch's Algorithm [33] chooses the

starting point to speed up the convergence of the algorithm due to Boyd and Balakrishnan.

4.3.1 Bisection method by Boyd, Balakrishnan and Kabamba

Recall that

$$\|H\|_\infty = \max_{\omega\in\mathbb{R}} \sigma_{\max}\left(H(i\omega)\right))$$

where $H(s)$ denotes the transfer function of the system.

We first establish a connection between the singular values of the transfer function and the eigenvalues of the Hamiltonian matrix

$$\begin{aligned} M_\gamma &= \begin{bmatrix} A & 0 \\ 0 & -A^T \end{bmatrix} + \begin{bmatrix} B & 0 \\ 0 & -C^T \end{bmatrix} \begin{bmatrix} -D & -\gamma I_n \\ \gamma I_n & -D^T \end{bmatrix} \begin{bmatrix} C & 0 \\ 0 & B^T \end{bmatrix} \\ &= \begin{bmatrix} A - BR^{-1}D^TC & -\gamma BR^{-1}B^T \\ \gamma C^TS^{-1}C & -a^T + C^TDR^{-1}B \end{bmatrix} \end{aligned} \tag{4.13}$$

for a given $\gamma > 0$ where $R = D^TD - \gamma^2 I_n$ and $S = DD^T - \gamma^2 I_n$.

The following theorem can be considered as a generalization of Theorem 4.1.1 for the distance to instability.

Theorem 4.3.1. *For any $\omega \in \mathbb{R}$, the scalar γ is a singular value of $H(i\omega)$ if and only if M_γ has the pure imaginary eigenvalue $i\omega$.*

The next theorem enables us to check whether our estimates γ_k satisfy $\|H\|_\infty \geq \gamma_k$ or $\|H\|_\infty < \gamma_k$.

Theorem 4.3.2. *Let A be a stable matrix and $\gamma > \sigma_{\max}(D)$. Then M_γ has at least one pure imaginary eigenvalue if and only if $\|H\|_\infty \geq \gamma$.*

Proof. Note that $\sigma_{\max}(D) = \lim_{t\to\infty} \sigma_{\max}(H(it))$. Suppose $\|H\|_\infty \geq \gamma > \sigma_{\max}(D)$. By the continuity of $\sigma_{\max}(H(i\omega))$, for all $\gamma \in (\sigma_{\max}(D), \|H\|_\infty]$ there must exists an

ω such that $\sigma_{\max}(H(i\omega)) = \gamma$. Now by Theorem 4.3.1 M_γ has the pure imaginary eigenvalue $i\omega$.

Conversely if M_γ has a pure imaginary eigenvalue $i\omega$, then by Theorem 4.3.1 $\|H\|_\infty \geq \sigma_{\max}(H(i\omega)) \geq \gamma$. □

We will use the bounds for the H_∞-norm of the transfer function derived by Enns [11] and Glover [22] given by

$$lb = \max\{\sigma_{\max}(D), \sigma_{H_1}\}, \tag{4.14}$$

$$ub = \sigma_{\max}(D) + 2\sum_i \sigma_{H_i} \tag{4.15}$$

where σ_{H_i} denotes the Hankel singular values of the system $\{A, B, C, D\}$, and σ_{H_1} denotes the largest Hankel singular value.

The bisection Algorithm is given below. Observe that, at termination $\frac{\gamma_{lb}+\gamma_{ub}}{2}$ is guaranteed to approximate the H_∞-norm of the transfer function within a relative accuracy of ϵ. Unfortunately, the algorithm converges linearly, and as n gets larger, it is expensive to evaluate the eigenvalues of M_γ as well as the initial bounds.

Algorithm 10 Bisection Algorithm for $\|H\|_\infty$-norm

1: **INPUT**: $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$ and $D \in \mathbb{R}^{p\times m}$ and the desired tolerance $\epsilon > 0$
2: Initialization: $\gamma_{lb} \leftarrow lb$ from equation (4.14) and $\gamma_{ub} \leftarrow ub$ from equation (4.15);
3: **repeat**
4: $\gamma \leftarrow \frac{\gamma_{lb}+\gamma_{ub}}{2}$;
5: **if** M_γ has at least one pure imaginary eigenvalue **then**
6: $\gamma_{lb} \leftarrow \gamma$;
7: **else**
8: $\gamma_{ub} \leftarrow \gamma$;
9: **end if**
10: **until**$(\gamma_{ub} - \gamma_{lb} \leq 2\epsilon\gamma_{lb})$
11: **OUTPUT** γ_{lb} and γ_{ub} satisfying $\frac{\gamma_{ub}-\gamma_{lb}}{2\gamma_{lb}} \leq \epsilon$.

4.3.2 Boyd and Balakrishnan Method

The algorithm is tailored based on Theorem 4.3.1. Let $i\omega_1, \ldots, i\omega_r$ denote the pure imaginary eigenvalues of M_γ. Then we can determine the frequency intervals $\mathcal{I}_k$ for $k = 1, \ldots, \ell$ where $\sigma_{\max}(H(i\omega)) > \gamma, \ \forall\omega \in \mathcal{I}_k$. By Theorem 4.3.1 the set $\{\omega_1, \ldots, \omega_r\}$ is a superset of the set of points satisfying $\sigma_{\max}(H(i\omega)) = \sigma$. Specifically if ω_1, ω_2 denote two consecutive ω values such that $\sigma_{\max}(H(i\omega)) = \gamma$, by checking whether

$$\sigma_{\max}\left(H\left(i\frac{(\omega_1+\omega_2)}{2}\right)\right) > \sigma$$

we can decide whether $[\omega_1, \omega_2]$ is one of those frequency intervals. We will define the new value of γ by $\gamma := \max_{k=1,\ldots,\ell} \sigma_{\max}(H(i\hat{\omega}_k))$ where $\hat{\omega}_k$ is the midpoint of $\mathcal{I}_k$.

Algorithm 11 Boyd and Balakrishnan Algorithm

1: **INPUT**: $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$ and $D \in \mathbb{R}^{p\times m}$ and the desired tolerance $\epsilon > 0$
2: Initialization: $\gamma \leftarrow \sigma_{\max}(D)$;
3: **repeat**
4: find the frequency interval $\mathcal{I}_1, \ldots, \mathcal{I}_\ell$ in which $\sigma_{\max}(H(i\omega)) > \gamma(1+\epsilon)$.
5: **if** $\ell > 0$ **then**
6: $\hat{\omega}_k \leftarrow$ midpoints of $\mathcal{I}_k$ for $k = 1, \ldots, \ell$;
7: $\gamma \leftarrow \max_{k=1,\ldots,\ell} \sigma_{\max}(H(i\hat{\omega}_k))$
8: **end if**
9: **until**$(\ell = 0)$
10: **OUTPUT** γ such that $\gamma \leq \|H\|_\infty \leq \gamma(1+\epsilon)$.

Normally lower bound for γ is chosen as $\sigma_{\max}(D)$, which is very cheap to compute. However one can prefer the lower bound given in (4.14). On the other hand, computation of the Hankel singular values requires the solution of two Lyapunov equations with dimension n, which is rather expensive. Many systems attain their $\|H\|_\infty$-norm at frequency $\omega = 0$. One reasonable choice suggested by Bruinsma and Steinbuch is

$$\gamma = \max\{\sigma_{\max}(H(0)), \sigma_{\max}(D), \sigma_{\max}(H(i\omega_p))\} \tag{4.16}$$

where $\omega_p = |\lambda_i|$ and λ_i is a pole of $H(s)$ satisfying $\lambda_i = \arg\max \left| \frac{Im(\lambda_i)}{Re(\lambda_i)} \frac{1}{|\lambda_i|} \right|$. Bruinsma and Steinbuch's method is the same as Algorithm 11, but with the initial guess for γ given by (4.16). This initialization reduces the number of function evaluations.

4.4 Numerical Radius

The numerical radius of a matrix $A \in \mathbb{C}^{n\times n}$ can be posed as the eigenvalue optimization problem

$$r(A) = \lambda_{\max}(H(Ae^{i\theta})),$$

where $H(A) = (A + A^*)/2$. Here we present level-set ideas for the computation of the numerical radius. Specifically this section concerns an algorithm due to *He and Watson* and a *modified Boyd and Balakrishnan method.*

4.4.1 He and Watson Algorithm

In this section we give the steps of a simple iterative algorithm, which we call simple iteration, and a corollary which is necessary to describe the algorithm in detail.

Simple iteration: Let z_0 be an initial nonzero complex vector. For $k = 1, 2, \ldots$, define $\omega_{k-1} := z_{k-1}^* A z_{k-1}$, and an z_k as

$$\gamma_k z_k = \omega_{k-1} A^* z_{k-1} + \bar{w}_{k-1} A z_{k-1}$$

where γ_k is such that $\|z_k\| = 1$. It has been shown by Watson that this iteration will converge to a locally maximal value of $f(\theta) := \lambda_{\max}\left(H(Ae^{i\theta})\right)$. However, it does not converge to a globally maximal value in general. To check whether a converged locally maximal value α is a globally maximal value the following theorem can be used.

Theorem 4.4.1 (He and Watson)**.** *The pencil $R(\alpha) - \lambda S$ has an eigenvalue on the*

unit circle or is singular if and only if $H(Ae^{i\theta})$ has α as one of its eigenvalues, where

$$R(\alpha) = \begin{bmatrix} 2\alpha I_n & -A^* \\ I_n & 0 \end{bmatrix} \text{ and } S = \begin{bmatrix} A & 0 \\ 0 & I_n \end{bmatrix}. \tag{4.17}$$

Proof. First observe that

$$R(\alpha) - e^{i\theta}S = \begin{bmatrix} 2\alpha I_n - e^{i\theta}A & -A^* \\ I_n & -e^{i\theta}I_n \end{bmatrix}.$$

This matrix is singular if and only if $e^{i\theta}A + e^{-i\theta}A^* - 2\alpha I_n$ is singular. Therefore there exists a nonzero vector v such that $(e^{i\theta}A + e^{-i\theta}A^* - 2\alpha I_n)v = 0$. By rearranging this equation, we have

$$\left(\frac{e^{i\theta}A + e^{-i\theta}A^*}{2}\right) v = \alpha v, \tag{4.18}$$

that is $H(Ae^{i\theta})$ has α as one of its eigenvalues. □

In case α is only a locally optimal solution, a better estimate for $r(A)$ is provided by the next corollary.

Corollary 4.4.2. *For a given α, let v be an eigenvector of the pencil $R(\alpha) - \lambda S$ associated with an eigenvalue η on the unit circle. If z is formed from the last n components of v and normalized to be a unit vector, then $|z^*Az| \geq \alpha$.*

Proof. Let $[v_1^T \ v_2^T]^T$ be an eigenvector of $R(\alpha) - \eta S$ where η is on the unit circle. Clearly, v_2 is nonzero, then by using the definitions of $R(\alpha)$, S and defining $z := \frac{v_2}{\|v_2\|}$, we have

$$\frac{\eta Az + \bar{\eta}A^*z}{2} = \alpha z.$$

Multiplying from the left by z^*, we obtain $|z^*Az| \geq \Re(\eta z^*Az) = \alpha$. □

The algorithm summarized here is dominated by the QZ algorithm for calculating

Algorithm 12 He and Watson

1: **INPUT**: $A \in \mathbb{R}^{n\times n}$ and the desired tolerance $\epsilon > 0$
2: Initialization: $lb \leftarrow 0$, $ub \leftarrow \|A\|_1$ and z_0 is any complex vector;
3: While $ub - lb > 2\epsilon$, do
4: $z \leftarrow$ run simple iteration with the initial vector z_0;
5: $lb \leftarrow \max\{lb, |z^*Az|\}$;
6: Extract the unit eigenvalues of $R(\alpha) - \lambda S$ for $\alpha = lb + \epsilon$.
7: **if** (4.18) has no eigenvalue **then**
8: $ub \leftarrow lb + \epsilon$; **return**
9: **else**
10: $z_0 \leftarrow$ last n components of an eigenvector of $R(\alpha) - \lambda S$ associated with a unit eigenvalue normalized to be a unit vector.
11: $ub \leftarrow |z_0^* A z_0|$;
12: **go to** step 4
13: **end if**
14: **OUTPUT** lower bound lb for $r(A)$ such that $lb \le r(A) \le lb + \epsilon$.

the eigenvalues of $R(\alpha) - \lambda S$ which costs $368 \times n^3$ flops, and thus is quite expensive. On the other hand, in most circumstances only one call to the QZ solver is required.

4.4.2 Modified Boyd and Balakrishnan Method

Thereom 4.4.1 provides the capability to find all θ such that $f(\theta) = \lambda_{\max}(H(Ae^{i\theta})) = \alpha$. Therefore the Boyd and Balakrishnan idea from the previous section to compute the $\|H\|_\infty$ is applicable. At the jth step, given an estimate r^j for $r(A)$, we find all θ such that $f(\theta) = r^j$. Then the intervals $\mathcal{I}_k^j$ such that

$$\mathcal{I}_k^j = \{[a_j^k, b_j^k] : f(\theta) > r^j, \text{ where } \theta \in [a_j^k, b_j^k]\}$$

can be determined. Next we evaluate f at the midpoints of these intervals, and set r^{j+1} to the maximum value attained by f among these midpoints.

Algorithm 13 is an extension of the Boyd and Balakrishnan algorithm originally introduced for the H_∞-norm, thus a similar convergence analysis is valid. Specifically, this algorithm converges quadratically to a global solution. On the other hand, finding

Algorithm 13 Modified Boyd and Balakrishnan Algorithm

1: **INPUT**: $A \in \mathbb{C}^{n\times n}$, and a desired tolerance ϵ
2: Initialization: $j \leftarrow 0$, $\phi^0 \leftarrow [0]$ and $r^0 \leftarrow 0$;
3: repeat
4: Find θ values such that $f(\theta) = r^j$.
5: Find the intervals I_k^j for $k = 1, \ldots, m_j$, where $f(\theta) > r^j$ for all $\theta \in \mathcal{I}_k^j$.
6: $\phi_j \leftarrow \{\phi_k^j : k = 1, \ldots, m_j\}$ where ϕ_k^j is the midpoint of $\mathcal{I}_k^j$;
7: $r^{j+1} \leftarrow max\{f(\theta) : \theta \in \phi^j\}$
8: $j \leftarrow j + 1$
9: until $(r_{j+1} - r_j < \epsilon)$
10: **OUTPUT** r_j an estimate for $r(A)$.

the eigenvalues of the pencil $R(\alpha) + \lambda S$ is still necessary. One can benefit from the fact that $R(\alpha) + \lambda S$ is symplectic, which implies that this problem can be reduced to a Hamiltonian eigenvalue problem and solved by means of a structure preserving algorithm respecting the Hamiltonian structure [34]. This reduces the computational complexity slightly. More importantly, the unit eigenvalues can be extracted reliably by exploiting the symplectic structure.

Chapter 5

Derivation of an Eigenvalue Optimization Algorithm

This chapter derives generic *one-dimensional* and *multi-dimensional* algorithms for the global solution of eigenvalue optimization problems, also described in [32]. We also discuss how the algorithm can be implemented efficiently by exploiting the observations due to Breiman and Cutler [23].

5.1 One Dimensional Algorithm

We assume $\mathcal{A}(\epsilon) : \mathbb{R} \to \mathbb{C}^{n \times n}$ is a Hermitian, analyic matrix function. It was discussed in Chapter 2 that the eigenvalues of such matrix functions can be ordered so that each eigenvalue is an analytic function of ϵ. Thus ordering the eigenvalues of $\mathcal{A}(\epsilon)$ from largest to smallest causes these eigenvalue functions intersect each other at finitely many isolated points on a finite interval. These intersection points do not violate the continuity, but analyticity. On the other hand, piece-wise analyticity is preserved, but the eigenvalue functions are not differentiable at these isolated points. Nevertheless, for the sake of simplicity we assume that the eigenvalue function $\lambda(\epsilon)$ to be optimized

is analytic everywhere. This is a reasonable assumption for many functions of interest. For instance for a generic matrix A, the multiplicity of $\sigma_{\min}(A - \omega i I_n)$ is one at all $\omega \in \mathbb{R}$, similarly for $\lambda_{\max}\left(\frac{Ae^{i\theta}+Ae^{-i\theta}}{2}\right)$ and for all $\theta \in [0, 2\pi]$.

We derive a quadratic model about a given $x_k \in \mathbb{R}$, which lies underneath the objective eigenvalue function $\lambda(\epsilon)$. Due to the analyticity of $\lambda(\epsilon)$ we have

$$\lambda(x) = \lambda(x_k) + \int_{x_k}^{x} \lambda'(t)\, dt. \tag{5.1}$$

As mentioned before we assume the knowledge of an upper bound γ on the second derivatives satisfying

$$\left|\lambda''(x)\right| \leq \gamma \quad \forall x \in \mathbb{R}.$$

Consequently, for all $t \in \mathbb{R}$ we have

$$\lambda'(t) \geq \lambda'(x_k) - \gamma(t - x_k). \tag{5.2}$$

By substituting the lower bound given by (5.2) for $\lambda'(t)$ in (5.1), we obtain the following inequality

$$\begin{array}{rcl} \lambda(x) & \geq & \lambda(x_k) + \int_{x_k}^{x} \lambda'(x_k) - \gamma(t - x_k)\, dt \\ & = & \lambda(x_k) + \lambda'(x_k)(x - x_k) - \frac{\gamma}{2}(x - x_k)^2. \end{array} \tag{5.3}$$

We conclude with

$$q_k(x) := \lambda(x_k) + \lambda'(x_k)(x - x_k) - \frac{\gamma}{2}(x - x_k)^2 \leq \lambda(x) \tag{5.4}$$

for all $x \in \mathbb{R}$. More generally for piece-wise analytic and continuous eigenvalue functions it is possible to extend the derivation above yielding piece-wise quadratic models lying underneath the eigenvalue functions (see [32] for details). The algorithm

largely depends on the piece-wise quadratic functions

$$\mathcal{M}_s(x) := \max_{k=0,\ldots,s} q_k(x)$$

The outline of the algorithm is given below. Initially it constructs the two quadratic models about the end points of a given interval $[\underline{x},\ \overline{x}]$ containing a global minimizer. Their intersection point is the starting point x_0 for the algorithm and $\mathcal{M}_0(x) := q_0(x)$ is the initial quadratic model. The algorithm keeps track of a sequence $\{x_k\}$ of estimates for a global minimizer of $\lambda(\epsilon)$ as well as a sequence $\{\mathcal{M}_k\}$ of piece-wise quadratic models. The point x_{k+1} is selected as a global minimizer of $\mathcal{M}_k(x)$. At the kth iteration, a quadratic model $q_{k+1}(x)$ about x_{k+1} is constructed and $\mathcal{M}_{k+1}(x)$ is set to the maximum of $\mathcal{M}_k(x)$ and $q_{k+1}(x)$. The algorithm terminates whenever $\left|\min_{j=0,\ldots,s} \lambda(x_j) - \min_{x\in[\underline{x},\overline{x}]} \mathcal{M}_s(x)\right|$ is less than or equal to a given tolerance.

Algorithm 14 One-dimensional Algorithm

1: **INPUT**: An analytic function $\lambda(x)$, the bound γ on second derivatives, initial box $[\underline{x}, \overline{x}]$ and a desired tolerance ϵ
2: let x_0 be the intersection of the quadratic models about $\underline{x}$ and $\overline{x}$.
3: $k \leftarrow 0$;
4: repeat
5: $x_{k+1} \leftarrow \arg\min_{x\in[\underline{x},\overline{x}]} \mathcal{M}_k(x)$ and $lb \leftarrow \min_{x\in[\underline{x},\overline{x}]} \mathcal{M}_k(x)$;
6: $ub \leftarrow \min_{j=0,\ldots,k} \lambda(x_j)$;
7: Evaluate $\lambda(x_{k+1})$ and $\lambda'(x_{k+1})$.
8: $q_{k+1} \leftarrow$ form the quadratic model about x_{k+1};
9: $\mathcal{M}_{k+1}(x) \leftarrow \max\{\mathcal{M}_k(x), q_{k+1}(x)\}$;
10: $k \leftarrow k + 1$;
11: until $(ub - lb < \epsilon)$
12: **OUTPUT** lb and ub such that $lb \le \lambda(x_*) \le ub$ with $ub - lb \le \epsilon$, where x_* is a global minimizer of $\lambda(x)$.

Evaluation of $\lambda(x)$ at a given value x_{k+1} requires the solution of an $n\times n$ Hermitian eigenvalue problem at each iteration. Once $\lambda(x_{k+1})$, that is the eigenvalue and the associated eigenvector, are evaluated, the derivative $\lambda'(x_{k+1})$ can be evaluated without

any significant work due to formula (2.13).

5.2 Multi-Dimensional Algorithm

Suppose $\mathcal{A}(\epsilon) : \mathbb{R}^d \to \mathbb{C}^{n\times n}$ is Hermitian and an analytic function of ϵ. Then the eigenvalues along any direction p can be ordered so that each eigenvalue is analytic in the direction p. If the eigenvalues are sorted from largest to smallest, they are continuous and piece-wise analytic along p. However, as in Section 5.1, for the sake of simplicity we assume each sorted eigenvalue is analytic along any direction p. Again this is a reasonable assumption for many eigenvalue functions of interest, for instance for $\sigma_{\min}([A - zI,\ B])$ over $z \in \mathbb{C}$ as the corresponding eigenvalue functions do not cross each other generically and therefore analytic along any direction. Extensions to piece-wise analytic case are given in [32].

First, we derive the quadratic model about $x_k \in \mathbb{R}^d$ which lie underneath the objective function. Let $x \in \mathbb{R}^d$ and define $\phi(\alpha) := \lambda(x_k + \alpha p)$ where p is the direction, defined as $p := \frac{(x-x_k)}{\|x-x_k\|}$, and λ is the eigenvalue function to be minimized that is analytic along p.

Since λ is analytic along p, we have

$$\lambda(x) = \lambda(x_k) + \int_0^{\|x-x_k\|} \phi'(t)dt. \tag{5.5}$$

As in Section 5.1, we assume the knowledge of a γ such that

$$\left|\phi''(\alpha)\right| \leq \gamma, \quad \forall \alpha \in \mathbb{R} \ \text{ and } \ \forall p \in \mathbb{R}^d. \tag{5.6}$$

Thus we have the inequality

$$\phi'(t) \geq \phi'(0) - \gamma t$$

holding for all $t \in \mathbb{R}$. Substitute the last inequality in (5.5), and integrate the right-

hand side of (5.5). Finally use the identity $\phi'(0) = \nabla\lambda(x_k)^T p = \nabla\lambda(x_k)^T \frac{x-x_k}{\|x-x_k\|}$ to get

$$q_k(x) := \lambda(x_k) + \nabla\lambda(x_k)^T(x - x_k) - \frac{\gamma}{2}\|x - x_k\|^2 \quad \leq \quad \lambda(x). \tag{5.7}$$

The algorithm in the multi-variate case (Algorithm 15) to solve the eigenvalue optimization problem

$$\min_{x\in\mathcal{B}} \lambda(x) \tag{5.8}$$

with

$$\mathcal{B} := \mathcal{B}(\underline{x}_1, \overline{x}_1, \ldots, \underline{x}_d, \overline{x}_d) := \{x \in \mathbb{R}^d : x_j \in [\underline{x}_j, \overline{x}_j] \text{ for } j = 1, \ldots, d\} \tag{5.9}$$

is the same as the univariate case described in Section 5.1

In the multi-variate algorithm a major computational difficulty is the solution of the optimization problem

$$\min_{x\in\mathcal{B}} \mathcal{M}_k(x) := \max_{\ell=0,\ldots,k} q_\ell(x) \tag{5.10}$$

at the kth iteration. Recall that x_{k+1} is chosen as the global minimizer of this problem. This problem can equivalently be posed as $k+1$ quadratic programs of the form

$$\begin{array}{lll} \text{minimize}_{x\in\mathbb{R}^d} & q_p(x) & \\ \text{subject to} & q_p(x) \geq q_\ell(x), & \ell \neq p \\ & x_j \in [\underline{x}_j, \overline{x}_j], & j = 1, \ldots, d \end{array} \tag{5.11}$$

for $p = 0, \ldots, k$. Observe that all the quadratic models have the same curvature, which implies that all constraints above are linear. The solution is guaranteed to be attained at one of the vertices. Unfortunately, since the quadratic models have negative curvature, the problem is non-convex, i.e., there may be locally, but not

globally, optimal solutions. Consequently, $q_p(x)$ needs to be evaluated at all vertices of the feasible region. Suppose we have ℓ linear constraints (including the box constraints), then we have at most $\begin{pmatrix} \ell \\ d \end{pmatrix}$ vertices. For higher dimensional case, solving this quadratic problem is NP-hard as the number of vertices grows exponentially. Currently, we have an implementation of the algorithm working efficiently in the two-dimensional case.

Algorithm 15 Multidimensional Algorithm

1: **INPUT**: An eigenvalue function $\lambda : \mathbb{R}^d \to \mathbb{R}$ that is analytic along any direction $p \in \mathbb{R}^d$, a scalar $\gamma > 0$ that is an upper bound on $|\phi''(\alpha)|$ at all α and $p \in \mathbb{R}^d$, a set $\mathcal{B}$ given by (5.9), and a tolerance parameter $\epsilon > 0$.
2: $x_0 \leftarrow$ arbitrary point from the box $\mathcal{B}$;
3: $\mathcal{M}_0(x) \leftarrow q_0(x)$ which is the quadratic model about x_0.
4: $k \leftarrow 0$;
5: repeat
6: $x_{k+1} \leftarrow \arg\min_{x\in\mathcal{B}} \mathcal{M}_k(x)$ and $lb \leftarrow \min_{x\in\mathcal{B}} \mathcal{M}_k(x)$;
7: $ub \leftarrow \min_{j=0,\ldots,k} \lambda(x_j)$;
8: Evaluate $\lambda(x_{k+1})$ and $\nabla\lambda(x_{k+1})$.
9: $q_{k+1}(x) := \lambda(x_{k+1}) + \nabla\lambda(x_{k+1})^T(x - x_{k+1}) - (\gamma/2)\|x - x_{k+1}\|^2$;
10: $\mathcal{M}_{k+1}(x) \leftarrow \max\{\mathcal{M}_k(x), q_{k+1}(x)\}$;
11: $k \leftarrow k + 1$;
12: until $(ub - lb < \epsilon)$
13: $x_{best} \leftarrow \arg\min_{j=0,\ldots,k-1} \lambda(x_j)$;
14: **Output:** lb, ub, x_{best} such that $lb \le \lambda(x_*) \le ub = \lambda(x_{best})$ with $ub - lb \le \epsilon$, where x_* is a global minimizer of $\lambda(x)$.

Now we present the convergence proof of Algorithm 15 given in [32].

Theorem 5.2.1. *Let $\{x_k\}$ be the sequence of iterates generated by Algorithm 15. Every limit point of this sequence is a global minimizer of the problem $\min_{x\in\mathcal{B}} \lambda(x)$.*

Notation for the proof

Let $\mu := \max_{x\in\mathcal{B}} \|\nabla\lambda(x)\|$ with the assumption $1/\mu := +\infty$ if $\mu = 0$. Also λ is analytic along every line in $\mathbb{R}^d$ implying the existence of a scalar $\gamma > 0$ satisfying (5.6). Also note that $\{lb_k\}$ is such that $lb_k := \min_{x\in\mathcal{B}} \mathcal{M}_k(x)$ is a non-decreasing sequence of

lower bounds on $\lambda_* := \min_{x\in\mathcal{B}} \lambda(x)$, while $\{ub_k\}$ is such that $ub_k := \min_{j=0,\ldots,k} \lambda(x_j)$ is a non-increasing sequence of upper bounds on λ_*.

Proof. The sequence $\{x_k\}$ has at least one limit point $x_* \in \mathcal{B}$, since $\mathcal{B}$ is a compact set. Without loss of generality we may assume that $\{x_k\}$ itself is a convergent sequence. Let lb_* denote the limit of the bounded nondecreasing sequence $\{lb_k\}$. We note $lb_k \leq \lambda_* \leq \lambda(x_k)$ for $k \geq 0$ implying

$$lb_* \leq \lambda_* \leq \lambda(x_*). \tag{5.12}$$

We claim indeed $\lambda(x_*) = lb_*$. We prove this claim by *contradiction*. Assume that there exists a real number $\delta > 0$ such that

$$\lambda(x_*) \geq lb_* + \delta. \tag{5.13}$$

meaning for some $k_1 \in \mathbb{N}$ we have

$$\lambda(x_k) \geq lb_* + \frac{\delta}{2}, \quad \text{for all } k \geq k_1. \tag{5.14}$$

Since x_* is the limit of the sequence $\{x_k\}$, there exists $k_2 \in \mathbb{N}$ that satisfies

$$\|x_{k'} - x_{k''}\| < \min\left\{\sqrt{\frac{\delta}{6\gamma}}, \frac{\delta}{12\mu}\right\}, \quad \text{for all } k' \geq k'' \geq k_2. \tag{5.15}$$

Let k_* denote the maximum of k_1 and k_2. For each $k \geq k_*$, it follows from the definition of the functions $\mathcal{M}_k(x)$ that

$$\begin{aligned} \mathcal{M}_k(x_{k+1}) &\geq \mathcal{M}_{k_*}(x_{k+1}) \geq q_{k_*}(x_{k+1}), \\ &= \lambda(x_{k_*}) + \nabla\lambda(x_{k_*})^T(x_{k+1} - x_{k_*}) - \frac{\gamma}{2}\|x_{k+1} - x_{k_*}\|^2. \end{aligned}$$

By using the Cauchy-Schwarz inequality as well as inequalities (5.14) and (5.15) we get the following

$$\begin{aligned} lb_{k+1} = \mathcal{M}_k(x_{k+1}) &\geq \lambda(x_{k_*}) - \|\nabla\lambda(x^*)\|\|x_{k+1} - x_{k_*}\| - \frac{\gamma}{2}\|x_{k+1} - x_{k_*}\|^2 \\ &\geq \left(lb^* + \frac{\delta}{2}\right) - \left(\mu \cdot \frac{\delta}{12\mu}\right) - \left(\frac{\gamma}{2} \cdot \frac{\delta}{6\gamma}\right) \\ &= lb^* + \frac{\delta}{3}. \end{aligned}$$

Since $\delta > 0$, this contradicts our assumption that lb_* is the limit of the non-decreasing sequence $\{lb_k\}$. Therefore, we have $\lambda(x_*) < lb_* + \delta$ for all $\delta > 0$, or equivalently $\lambda(x_*) \leq lb_*$. This combined with (5.12) yields $\lambda(x_*) = lb_* \leq \lambda(x)$ for all $x \in \mathcal{B}$. Therefore, x_* is a global minimizer of $\lambda(x)$ over $\mathcal{B}$. □

5.3 The Deterministic Algorithm for Global Optimization

The idea of using quadratic models of the form given by (5.7) was explored by Breiman and Cutler in [23] in the context of global optimization. In particular they have come up with remarkable observations for the solutions of the quadratic programs in (5.11). Here we present some of their observations that would improve the efficiency of Algorithm 15.

At the kth iteration Algorithm 15 solves $k+1$ quadratic programs each of form (5.11). Associated with each quadratic program there is a feasible region, which is a polytope. Clearly these feasible regions are disjoint, and their union forms $\mathcal{B}$. Intuitively inside the feasible region for the jth quadratic model, this quadratic model dominates the others, so the minimization of $\mathcal{M}_k(x)$ inside this region is same as the minimization of $q_j(x)$. Figure 5.1 illustrates a possible partitioning of $\mathcal{B}$ into these polytopes when $k = 20$. An efficient implementation of Algorithm 15 must update

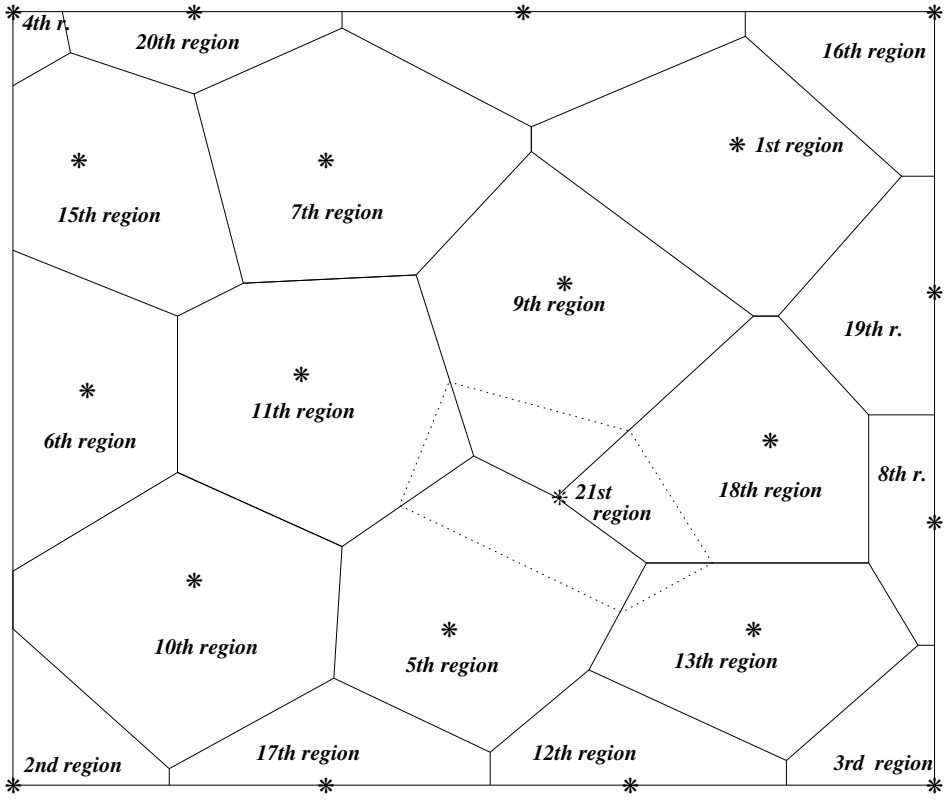

Figure 5.1: Polytope structure, $k = 20$

this polytope structure efficiently when a new quadratic model is added.

As shown in Figure 5.1 corresponding to a new quadratic model a new polytope appears. Some previous vertices are contained inside this polytope, and are not vertices anymore. These we call dead vertices. A dead vertex x_D satisfies $q_{k+1}(x_D) > M_k(x_D)$. We need to remove these dead vertices. Additionally we need to determine the new vertices and their connections forming the boundary of the new polytope. Finally we need to update the old polytopes adjacent to the new polytope (e.g., 11th, 9th, 5th, 13th, 18th regions in Figure 5.1).

At the kth step the minimum-valued vertex x_{k+1} of $\mathcal{M}_k(x)$ will be a dead vertex. Additionally Breiman and Cutler established that the set of dead vertices at the kth iteration is graph-connected. Therefore the set of dead vertices can be determined

efficiently by starting from x_{k+1} and expanding to the adjacent vertices until vertices that are not dead are reached. Additionally each new vertex of the new polytope appears on the edge connecting a dead vertex to an adjacent vertex that is not dead. This again is an observation due to Breiman and Cutler. The edges into these new vertices also can be determined efficiently.

Chapter 6

Numerical Comparisons

We reserve this chapter for the numerical applications of Algorithm 15. Let us first focus on the distance to instability problems. Specifically, we illustrate Algorithm 15 on matrices

$$\mathcal{A}_n := \mathcal{R}_n + i\mathcal{R}_n$$

where $\mathcal{R}_n$ is a random matrix with entries selected independently from a normal distribution with zero mean and unit variance. If $\mathcal{A}_n$ is not a stable matrix, we compute the eigenvalue decomposition and make the positive eigenvalues negative to obtain a related stable matrix.

To apply Algorithm 15 we need a lower and an upper bound for a feasible interval containing a global minimizer as well as an upper bound for the second derivatives in absolute value. We assume that a global minimizer is contained in the interval $[-5, 5]$. As for the bound on the second derivatives, we will use two which we numerically observe as an upper bound for $|f''(\omega)|$ where $f(\omega) = \sigma_{\min}(A - i\omega I_n)$. The number of function evaluations by Algorithm 14 applied to calculate the distance to instability for $\mathcal{A}_n$ for various n is presented in Table 6.1. The CPU times in seconds are also provided in parenthesis. It can be observed that the asymptotic rate of convergence appears to be linear, i.e., every two-decimal-digit accuracy requires about

n / ϵ	10^{-4}	10^{-6}	10^{-8}	10^{-10}	10^{-12}
100	37 (0.92)	55 (1.29)	72 (1.74)	83 (1.97)	100 (2.38)
400	45 (15.29)	64 (21.67)	80 (26.95)	96 (32.40)	114 (38.25)
800	43 (61.84)	57 (82.13)	69 (99.31)	81 (116.38)	90 (129.71)

Table 6.1: Number of function evaluations (or iterations) and cpu-times in seconds (in parenthesis) of the one-dimensional algorithm on the random matrices $\mathcal{A}_n$ of various sizes

n / ϵ	10^{-4}	10^{-6}	10^{-8}	10^{-10}
100	36 (0.87)	49 (0.98)	63 (1.25)	75 (1.52)
400	57 (15.17)	79 (19.77)	104 (26.01)	123 (29.99)
900	55 (96.66)	62 (111.46)	69 (122.47)	74 (132.31)

Table 6.2: Number of function evaluations (or iterations) and cpu-times in seconds (in parenthesis) of the one-dimensional algorithm on the Poisson-random matrices $\mathcal{B}_n$ of various sizes

fixed number of additional iterations for a given n in Table 6.1.

Secondly, we focus on the Crawford number. We illustrate Algorithm 14 on matrices

$$\mathcal{B}_n := \mathcal{P}_n - i\mathcal{R}_n$$

of various sizes, where $\mathcal{P}_n$ is an $n \times n$ matrix obtained from a finite difference discretization of the Poisson operator. Here optimization needs to be performed on $[0, 2\pi]$. We assume that the bound on the second derivative in absolute value is the 2-norm of the associated matrix $\mathcal{B}_n$. The number of function evaluations by Algorithm 14 applied to calculate the Crawford number for $\mathcal{B}_n$ for various n is listed in Table 6.2 together with the CPU times in seconds in parenthesis. The asymptotic rate of convergence again appears to be linear.

Finally we apply Algorithm 15 to calculate the distance to defectiveness. We illustrate Algorithm 15 on $n \times n$ tridiagonal matrices $\mathcal{T}_n$ for various n. In all examples we set $\gamma = 20$. Also we assume that a global minimizer is contained in the box $[-1, -1] \times [1, 1]$. The inner minimization problems are solved by means of the secant method. The number of function evaluations that the algorithm requires is given in

n / ϵ	10^{-2}	10^{-3}	10^{-4}
20	109 (34.05)	212 (61.74)	223 (67.08)
40	158 (153.06)	217 (212.07)	221 (220.29)

Table 6.3: Number of function evaluations (or iterations) and cpu-times in seconds (in parenthesis) of Algorithm 15 for calculating the distance to defectiveness from tridiagonal matrices of various sizes

Table 6.3. Again the CPU times are provided in parenthesis in the table.

For the two dimensional case, we use the *mesh-adaptive* version of the multi-dimesional algorithm described in [32]. Instead of running the algorithm once on the whole box, mesh-adaptive algorithm divides the initial box into smaller boxes, and call the algorithm on each box. By this modification, the algorithm discards some of the infeasible regions that do not contain a global minimizer efficiently. Also the quadratic models capture the objective function much faster in smaller boxes. The asymptotic rate of convergence cannot be interpreted directly, since the algorithm is mesh-based.

Chapter 7

Conclusion

We presented a generic algorithm for the optimization of the eigenvalues of a Hermitian matrix function depending on its parameters analytically. Similar ideas were explored in the general setting of global optimization by Breiman and Cutler, but, to our knowledge, never considered in the context of eigenvalue optimization. The convergence of the algorithm is given by Theorem 5.2.1. In practice we observe linear convergence.

In the multi-dimensional case the algorithm requires the solutions of many quadratic programs (i.e., optimization problems with quadratic objective functions and linear constraints). It might be possible to solve these quadratic programs more efficiently by incorporating the observations due to Breiman and Cutler concerning how to modify the polytope structure when a new quadratic model is added.

The quadratic programs are NP-hard, but the solutions are guaranteed to be attained at the vertices of the feasible regions. So far, we implemented the algorithm to solve the global eigenvalue optimization problems in one and two dimensional cases efficiently. In the near future, we will extend the implementation for higher dimensional cases. For specific problems, we hope to derive naive bounds for second derivatives and tight initial intervals containing global minimizers.

Bibliography

[1] A. Ben-Tal and M. P. Bendsøe, *A new method for optimal truss topology design*, SIAM. J. Optim., 3:322–358, 1993.

[2] A. Greenbaum and G. H. Rodrigue, *Optimal preconditioners of a given sparsity pattern*, BIT Numer. Math., 29: 610–634, 1989.

[3] A. N. Malyshev, *A Formula for the 2-norm Distance from a Matrix to the Set of Matrices with Multiple Eigenvalues*, Numer. Math. ,83:443–454, 1999.

[4] A. Ruhe, *Properties of a matrix with a very ill-conditioned eigenproblem*, Numer. Math., 15:57–60, 1970.

[5] A. S. Lewis, *Nonsmooth Analysis of Eigenvalues*, Math. Prog., 84:1–24, 1999.

[6] B. Shubert, *A sequential method seeking the global maximum of a function*, SIAM J. Numer. Anal., 9:379–388, 1972.

[7] C. C. Paige, *Properties of numerical algorithms relating to computing controllability*, IEEE Trans. Automat. Control, 26:130–138, 1981.

[8] C. F. Van Loan, *How near is a matrix to an unstable matrix?*, Lin. Alg. and its Role in Systems Theory, 47:465–479, 1984.

[9] C. He and G.A. Watson, *An algorithm for computing the distance to instability*, SIAM journal on matrix analysis and applications, 20:101–116, 1998.

[10] C. He and G.A. Watson, *An algorithm for computing the numerical radius*, IMA J. Numer. Anal., 17:329–342, 1997.

[11] D. F. Enns, *Model reduction with balanced realizations: error bound and a frequency weighted generalization*, Decision and Control, The 23rd IEEE Conference on 1984, 23: 127–132

[12] D. Hinrichsen and M. Motscha, *Optimization problems in the robustness analysis of linear state space systems*, Proceedings of the international seminar on Approximation and optimization, 54–78, 1988.

[13] D. Kressner and E. Mengi and I. Nakic and N. Truhar, *Generalized Eigenvalue Problems with Specified Eigenvalues*, **accepted subject to minor revision**, IMA J. Numer. Anal., 2011.

[14] D. R. Jones and C. D. Perttunen and B. E. Stuckman, *Lipschitzian optimization without the Lipschitz constant*, J. Optim. Theory Appl., 79:157–181,1993.

[15] E. Mengi, *Locating a nearest matrix with an eigenvalue of prespecified algebraic multiplicity*, Numer. Math, 118:109-135, 2011.

[16] F. H. Clarke, *Optimization and nonsmooth analysis*, Society for Industrial and Applied Mathematics, 38-39, 1983.

[17] F. Rellich, *Perturbation Theory of Eigenvalue Problems*, Gordon and Breach, 1969.

[18] G. Robel, *On Computing the Infinity Norm*, IEEE Trans. Automat. Control, 34:882–884, 1989.

[19] J. H. Wilkinson, *Note on matrices with a very ill-conditioned eigenproblem*, Numm. Math., 19:176–178, 1972.

[20] J. Nocedal and S. Wright, *Numerical optimization*, Springer-Verlag New York Inc, 2nd Ed., 519–523, 2006.

[21] J. V. Burke and A.S. Lewis and M.L. Overton, *Pseudospectral components and the distance to uncontrollability*, SIAM J. Matrix Analysis Appl., 26:350–361, 2004.

[22] K. Glover, *All optimal Hankel-norm approximations of linear multivariable systems and their L^∞-error bounds*, Onternat. J. Control, 39: 1115–1193, 1984.

[23] L. Breiman and A. Cutler, *A deterministic algorithm for global optimization*, Mathematical Programming, 58:179–199, 1993.

[24] L. Lovász, *On the Shannon capacity of a graph*, IEEE Trans. on Inf. Theory, 25:1–7, 1979.

[25] L. Vandenberge and S. Boyd, *Semidefinite programming*, SIAM J. Numer. Anal, 38:49–95, 1996.

[26] M. A. Freitag and A. Spence, *A Newton-based method for the calculation of the distance to instability*, Linear Algebra and Its Applications, 435: 3189–3205, 2011.

[27] M. Eiermann, *Fields of values and iterative methods*, Linear Algebra and its App., 180:167–197, 1993.

[28] M. Gao and M. Neumann, *A global minimum search algorithm for estimating the distance to uncontrollability*, Linear Algebra Appl.,188-189:305–350, 1993.

[29] M. Grötschel and L. Lovász and A. Schriver, *Geometric algorithms and combinatorial optimization*, Springer-Verlag, New York, 1988.

[30] M. Gu and E. Mengi and M.L. Overton and J. Xia and J. Zhu, *Fast methods for estimating the distance to uncontrollability*, SIAM J. Matrix Anal. Appl., 28:477–502, 2006.

[31] M. Gu, *New methods for estimating the distance to uncontrollability*, SIAM J. Matrix Analysis Appl., 21:989–1003, 2000.

[32] M. Kılıç and E. Mengi and E. A. Yıldırım, *Numerical optimization of eigenvalues of hermitian matrix functions*, **submitted to** SIAM. J. Matrix Anal. Appl., 2012.

[33] N. A. Bruinsma and M. Steinbuch, *A fast algorithm to compute the H_∞-norm of a transfer function matrix*, Systems and Control Letters, 14:287–293, 1990.

[34] P. Benner and V. Mehmann and H. Xu, *A numerically stable, structure preserving method for computing the eigenvalues of real Hamiltonian or symplectic pencils*, Numer. Math., 78:329–358, 1998.

[35] P. Lancaster, *On Eigenvalues of Matrices Dependent on a Parameter*, Numer. Math.,6:377–387, 1964.

[36] R. A. Horn and C. R. Johnson, *Matrix analysis*, Cambridge University Press, 1985

[37] R. Byers, *A bisection method for measuring the distance of a stable to unstable matrices*, SIAM J. Sci. Statist. Comput., 9:875–881, 1988.

[38] R. Byers, *Detecting nearly uncontrollable pairs*, Proceeding of the International Symposium MTNS-89 (Amsterdam, 1989).

[39] R. Eising, *Between controllable and uncontrollable*, Systems Control Lett., 4:263–264, 1984.

[40] R. H. Bartels and G. W. Stewart, *Solution of the equation $AX + XB = C$*, Comm. ACM, 15:820–826, 1972.

[41] S. A. Piyavskii, *An algorithm for finding the absolute extremum of a function*, USSR Comput. Math. and Math. Phys., 12:57–67, 1972.

[42] S. Boyd and V. Balakrishnan and P. Kabamba, *A Bisection Method for Computing the H_∞ Norm of a Transfer Matrix and Related Problems*, Mathematics of Control, Signals and Systems, 207–219, 1989.

[43] S. Boyd and V. Balakrishnan, *A regularity result for the singular values of a transfer matrix and a quadratically convergent algorithm for computing its $\mathbf{L}_\infty$-norm*, Systems Control Lett., 15:1–7, 1990.

[44] S. H. Cheng and N. J. Higham, *The nearest definite pair for the Hermitian generalized eigenvalue problem*, Linear Alg. and its App., 302-303:63-76, 1999.

[45] O. Axelsson and H. Lu and B. Polman, *On the numerical radius of matrices and its application to iterative solution methods*, Linear and Mult. Algebra, 37:225–238, 1994.

[46] Y. D. Sergeyev and D. E. Kvasov, *Global Search Based on Efficient Diagonal Partitions and a Set of Lipschitz Constants*, SIAM J. on Optimization, 16:910–937, 2006.

VITA

Mustafa Kılıç was born in 1988 in Yıldırım in Bursa. He graduated from ITU, Mathematical Engineering Department in 2010. In late 2010, he started to expertise on numerical optimization under supervision of Assist. Prof. Emre Mengi from KU. He was both research and teaching assistant in KU, and recently he has qualified to have Master degree on Mathematics.