Theory and methods for problems arising in robust stability, optimization and quantization

by

Mert Gürbüzbalaban

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Mathematics

New York University

May 2012

Advisors - C. Sinan Güntürk and Michael L. Overton

To my family

Acknowledgements

Firstly, I would like to thank my advisors Sinan Güntürk and Michael L. Overton for their endless support and guidance during my stay at Courant. Not only they introduced me to fascinating problems and guided me through them, but also were an infinite source of inspiration, creativity, and motivation. Sinan was also my mentor in my first year and has been a great source of career advice since. Michael, as a meticulous editor, taught me plenty of tips for writing mathematics with an elegant style which improved my writing skills substantially. Beyond research, it was also a great opportunity to know them personally from which my character and personal development have significantly benefited. Both Michael and Sinan are among the "nicest" people I have ever met.

I wish to express my gratitude to Nicola Guglielmi with whom I had a chance to work closely on the computation of the H_∞ norm for large sparse systems. I benefited much from his mathematical insight and knowledge in our various discussions. I also thank Vincent Blondel and Alexandre Megretski for our discussions which led to a fruitful collaboration. I am also indebted to Jonathan Goodman for teaching me a treasure-trove of beautiful mathematics. In addition, I would like to thank Daniel Kressner, Emre Mengi and Robert Kohn for the discussions and suggestions about my work.

I would also like to thank my professors at Boğaziçi University who provided me support, valuable advice and background before I came to Courant. I particularly thank Kadri Özçaldiran, Alp Eden, Burak Gürel and Ahmet Feyzioğlu.

I met many nice people at Courant. I thank the students at Courant for their friendship and for making Courant a fun place. In addition, I cannot thank my close friends Dr. Özgür Kalenci and Dr. Hikmet Dursun enough for their everlasting

friendship which helped me shape my personal life and academic career so far.

Finally, I thank my family Mine, Mehmet and Melis Gürbüzbalaban and Burçe Ergel for supporting me in every possible way during my PhD years. I always felt their support when I needed; none of my achievements would be possible without them.

Abstract

This thesis is composed of three independent parts:

Part I concerns spectral and pseudospectral robust stability measures for linear dynamical systems. Popular measures are the H_∞ norm, the distance to instability, numerical radius, spectral abscissa and radius, pseudospectral abscissa and radius. Firstly, we develop and analyze the convergence of a new algorithm to approximate the H_∞ norm of large sparse systems. Secondly, we tackle the static output feedback problem, a problem closely related to minimizing the abscissa (largest real part of the roots) over a family of monic polynomials. We show that when there is just one affine constraint on the coefficients of the monic polynomials, this problem is tractable, deriving an explicit formula for the optimizer when it exists and an approximate optimizer otherwise, and giving a method to compute it efficiently. Thirdly, we develop a new Newton-based algorithm for the calculation of the distance to discrete instability and prove that for generic matrices the algorithm is locally quadratically convergent. For the numerical radius, we give a proof of the fact that the Mengi-Overton algorithm is always quadratically convergent. Finally, we give some regularity results on pseudospectra, the pseudospectral abscissa and the pseudospectral radius. These results answer affirmatively a conjecture raised by Lewis & Pang in 2008.

Part II concerns nonsmooth optimization. We study two interesting nonsmooth functions introduced by Nesterov. We characterize Clarke stationary and Mordukhovich stationary points of these functions. Nonsmooth optimization algorithms have an interesting behavior on the second function, converging very often to a nonminimizing Clarke stationary point that is not Mordukhovich stationary.

Part III concerns the equivalence between one-bit sigma-delta quantization and

a recent optimization-based halftoning method. Sigma-delta quantization is a popular method for the analog-to-digital conversion of signals, whereas halftoning is a core process governing most digital printing and many display devices, by which continuous tone images are converted to bi-level images. The halftoning problem was recently formulated as a global optimization problem. We prove that the same objective function is minimized in one-bit sigma-delta quantization.

Contents

List of Figures

List of Tables

Introduction

This thesis consists of three independent parts; each part is independent from the others in terms of notation and content.

Part I (Chapters 1-4) is on the spectral and pseudospectral robust stability measures for linear dynamical systems with input and output. Such a system is stable if it has bounded output for the types of input that we are interested in. However, besides stability, we desire systems that are robustly stable, i.e., systems that remain stable under perturbations. One approach to design robust systems is to optimize a robust stability measure. Popular robust stability measures are the H_∞ norm, the distance to continuous instability (a.k.a. complex stability radius), the distance to discrete instability, numerical radius, spectral abscissa and radius, pseudospectral abscissa and radius [BHLO06a], all which we call robust stability functions (RSFs). The difficulty of the optimal system design problems with RSFs is due to the nonsmoothness and nonconvexity of RSFs. This is a topic where optimization and numerical linear algebra meet, closely linked to nonsmooth optimization techniques, eigenvalue and pseudospectrum problems, and perturbation theory of linear operators.

Our primary contribution in Part I is in the large scale computation of the H_∞ norm. Algorithms to compute the H_∞ norm accurately exist [BB90b, BS90, Rob89, GDV98], but they are impractical when the dimension is large and they do not exploit the structure of many interesting problems where large sparse matrices arise, especially in the control of partial differential equations (PDE), for example in the control of the heat equation with boundary control [LV07]. In Chapter 1 we develop a novel algorithm to approximate the H_∞ norm of large sparse systems and discuss its local convergence. We also provide a freely available MATLAB package

that implements the algorithm. In addition, the new algorithm will be included in the HIFOO package [Hif] eventually so that it can be used to design controllers in PDE applications, many of which are collected in [Lei06].

In Chapter 2, we tackle the static output feedback problem where the question is whether a controller that would make the closed-loop system stable exists, a problem closely related to minimizing the abscissa (largest real part of the roots) over a family of monic polynomials. It is known that this problem is NP-hard in many cases [BT95, Nem93]. However, in Chapter 2, we show that in some cases this problem is tractable, deriving an explicit formula for the optimizer when it exists and an approximation when it does not, and giving a method to compute it efficiently. This work has been accepted for publication in *IEEE Trans. Auto. Control* [BGMO].

In Chapter 3, we study the distance to discrete instability and the numerical radius (the latter arises in the analysis of the convergence of iterative solution methods for linear systems and the stability of hyperbolic finite-difference schemes). Inspired by the work [FS11, SP05], we develop a new Newton-based algorithm for the calculation of the distance to discrete instability and prove that for generic matrices the algorithm is locally quadratically convergent. For the numerical radius, firstly we prove that the assumption made in the Mengi-Overton algorithm [MO05] always holds unconditionally. Secondly, inspired by [GDV98], we improve this algorithm by developing a cubically convergent variant.

In Chapter 4, we study the regularity of the pseudospectral abscissa and radius functions and of pseudospectra. Besides having applications to robust control, pseudospectra, pseudospectral abscissa and radius are useful for studying the convergence and stability behavior of operators and matrices arising in many fields

such as numerical analysis, fluid dynamics and Markov chains [TE05]. It has been argued that for nonnormal operators, pseudoeigenvalues and pseudoeigenvectors reveal more information than the eigenvalues and eigenvectors. Understanding the regularity of the ε-pseudospectrum, ε-pseudospectral abscissa and ε-pseudospectral radius as a function of the underlying matrix and the scalar parameter ε is important both for understanding the sensitivity of pseudoeigenvalues and pseudoeigenvectors and for the design of efficient algorithms to compute pseudospectra-related quantities. Some exciting recent work has been done in this area [Kar03, LP08, BL10]. In Chapter 4, we give a proof of the fact that the pseudospectral abscissa and pseudospectral radius are globally Lipschitz together with the local Lipschitzness and differentiability of the boundary of the pseudospectrum where the pseudospectral abscissa or pseudospectral radius value is attained (this extends to more general boundary points by rotating the pseudospectrum). The results, published in [GO12b], answer affirmatively a conjecture raised by Lewis and Pang [LP08] and have applications to the design and study of pseudospectra-related algorithms, for instance, to the convergence proof of a recent algorithm by Kressner and Vandereycken for computing the stability radius [KV12]. In addition, they justify the use of nonsmooth optimization algorithms developed for locally Lipschitz functions such as the gradient sampling method [BLO05] to optimize the pseudospectral abscissa and radius of a parameter dependent matrix. Furthermore, an improvement of the differentiability of the pseudospectrum boundary result to the twice continuously differentiability of the boundary seems possible with similar machinery, and this would imply that the criss-cross algorithm of [BLO03a] is always, not just almost always, quadratically convergent.

Part II consists of Chapter 5 and concerns nonsmooth optimization. Convex

optimization problems have many important properties, including a powerful duality theory and the property that any local minimum is also a global minimum. Nonsmooth optimization refers to the more general problem of minimizing functions that are typically not differentiable at their minimizers. The RSFs mentioned above are typical examples. Nonsmooth optimization requires the study of subgradients (a generalization of the notion of a gradient in the absence of smoothness) and uses tools and ideas from the field *variational analysis*. In Chapter 5 we study two interesting nonsmooth functions introduced by Nesterov. Our contribution is to characterize Clarke stationary and Mordukhovich stationary points of these functions using variational analysis techniques. Nonsmooth optimization algorithms have an interesting behavior on the second function, converging very often to a nonminimizing Clarke stationary point that is not Mordukhovich stationary. Our results, published in [GO12a], motivated some other recent related research on the behavior of optimization algorithms on smooth variants of these functions (Jarre [Jar], Cartis et al. [CGT11]).

Part III consists of Chapter 6 and is related to the quantization of signals. Quantization is the process of mapping a large set of input values to a smaller set, and is an integral part of many electronic devices that convert analog (continuous) signals to digital ones. Quantization is inherent in many applications. For example, converting an image to GIF format reduces the file size by limiting the number of colors to 256. Halftoning of images, being an important part of most digital printing and many display devices, is another quantization problem, by which images of continuous tones are converted to ensembles of discrete dots in a limited number of colors. A recent approach to halftoning of images is to formulate the halftoning problem as a global energy minimization problem where the energy is a

difference of two convex functionals [TSG+11]. The objective function is an energy functional inspired by electrostatics. In Chapter 6, we demonstrate the connections of this approach with the sigma-delta quantization. In particular, we prove that the same energy functional is also minimized by one-bit first-order sigma-delta quantization.

Chapter 1 is done in collaboration with Nicola Guglielmi and Michael Overton [GGO]. Chapter 2 is done in collaboration with Vincent Blondel, Alexandre Megretski and Michael Overton [BGMO10, BGMO]. Chapters 4-5 are joint work with Michael Overton [GO12a, GO12b] and Chapter 6 is joint work with Sinan Güntürk [GG12].

Part I

Robust Stability and Pseudospectra

Chapter 1

Fast approximation of the H_∞ norm

1.1 Introduction

Consider the continuous-time linear dynamical system with input and output defined by

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{1.1}$$

where $A \in \mathbb{C}^{n,n}$, $B \in \mathbb{C}^{n,p}$ and $C \in \mathbb{C}^{m,n}$ and $D \in \mathbb{C}^{m,p}$. The discrete time analogue is

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k + Du_k. \end{aligned} \tag{1.2}$$

In this paper we present new methods for computing the H_∞ norm of the transfer function associated with these systems, a well known important quantity for measuring robust stability [HP05, ZGD95]. We build on two foundations. The first is the theory of spectral value sets presented in [HP05], as the H_∞ norm can be viewed as the reciprocal of the largest value of ε such that the associated ε-spectral value set is contained in the stability region for the dynamical system (the left-half plane in the continuous-time case and the unit disk in the discrete-time case). The second is an algorithm recently introduced by Guglielmi and Overton [GO11] for computing the rightmost point (or the largest point in modulus) in the ε-pseudospectrum of a matrix A. We extend this algorithm from pseudospectra to spectral value sets, and then give a Newton-bisection method to approximate the H_∞ norm. The algorithm is much faster than the standard Boyd-Balakrishnan-Bruinsma-Steinbuch algorithm to compute the H_∞ norm when $n \gg \max(m, p)$ and the matrix A is sparse.

The paper is organized as follows. In the next section we establish the fundamental properties of spectral value sets that we will need and we define the H_∞ norm. In Section 1.3 we generalize the algorithm of [GO11] for computing the pseudospectral abscissa of a matrix A to a spectral value set abscissa for (A, B, C, D). We briefly discuss local convergence analysis for this method, including the characterization of fixed points of the iteration, and we give a variation for the spectral value set radius. Then in Section 1.4 we introduce a Newton-bisection method to approximate the H_∞ norm. Every step of this method requires the approximation of a spectral value set abscissa (or radius) and each of these is carried out by an iteration which requires only the computation of the rightmost eigenvalue (or eigenvalue with largest modulus) of a sequence of matrices that are

rank-one perturbations of A. In Sections 1.5.1 and 1.5.2 we present numerical examples.

1.2 Spectral Value Sets

The first part of this section follows the development in [HP05, Section 5.1]; more detailed attribution appears below. Given A, B, C, D defining the linear dynamical system (1.1), consider the *perturbed system matrix*

$$M(E) = A + BE(I - DE)^{-1}C \quad \text{for } E \in \mathbb{C}^{p\times m} \tag{1.3}$$

assuming $I - DE$ is invertible and the associated *transfer matrix*

$$G(\lambda) = C(\lambda I - A)^{-1}B + D \quad \text{for } \lambda \in \mathbb{C}\backslash\sigma(A)$$

where $\sigma(\cdot)$ denotes spectrum. The following fundamental theorem relates the norm of the transfer matrix evaluated at eigenvalues of the perturbed system matrices to the norms of the underlying perturbations E. Here and throughout the paper, $\|\cdot\|$ denotes the matrix or vector 2-norm $\|\cdot\|_2$, or equivalently the maximum singular value. The dimension of the identity matrix I depends on the context.

Theorem 1.2.1. *Let $\varepsilon \in \mathbb{R}$, with $\varepsilon > 0$ and $\varepsilon\|D\| < 1$. Then for $\lambda \notin \sigma(A)$ the following are equivalent:*

$$\|G(\lambda)\| \geq \varepsilon^{-1} \qquad \text{and} \qquad \lambda \in \sigma(M(E)) \text{ for some } E \text{ with } \|E\| \leq \varepsilon. \tag{1.4}$$

Proof. Suppose the first statement holds with $\xi = \|G(\lambda)\|^{-1} \leq \varepsilon$. Let u and v

respectively be right and left singular vectors of $G(\lambda)$ corresponding to the largest singular value ξ^{-1}, so that $\xi G(\lambda)u = v$, $\xi v^* G(\lambda) = u^*$ and $\|u\| = \|v\| = 1$. Set $E = \xi uv^*$ so that $\|E\| = \xi \leq \varepsilon$. We have $G(\lambda)E = vv^*$, so

$$(C(\lambda I - A)^{-1}B + D)Ev = v. \tag{1.5}$$

Define $Y = (I - DE)^{-1}C$ and $Z = (\lambda I - A)^{-1}BE$, so we have $YZv = v$. It follows that $ZYx = x$, with $x = Zv \neq 0$ an eigenvector of ZY. Multiplying through by $\lambda I - A$, we have

$$BE(I - DE)^{-1}Cx = (\lambda I - A)x. \tag{1.6}$$

proving the second statement in (1.4).

Conversely, suppose that the second statement holds. Then $\exists x \neq 0$ such that (1.6) holds. We have $ZYx = x$, so x is an eigenvector of ZY corresponding to the eigenvalue 1. Consequently, $YZw = w$ where $w = Yx \neq 0$ is an eigenvector of YZ. Multiplying by $I - DE$ and rearranging we have

$$(C(\lambda I - A)^{-1}B + D)Ew = w$$

so

$$\varepsilon \|G(\lambda)\| \geq \|G(\lambda)E\| \geq 1$$

establishing the first statement in (1.4). □

Remark 1.2.2. *The equivalence* (1.4) *also holds if we restrict E in the first statement to have rank one. The proof remains unchanged. Furthermore, if, given λ, we choose $\varepsilon = \xi = \|G(\lambda)\|^{-1}$ and $E = \xi uv^*$ as in the proof, then the inequalities in both statements hold with equality. Note that u and v are each uniquely defined up*

to a unimodular scalar if and only if the maximum singular value ξ^{-1} is simple.

Definition 1.2.3. *Let $\varepsilon \in \mathbb{R}$, with $\varepsilon \geq 0$ and $\varepsilon\|D\| < 1$, and define the* spectral value set

$$\sigma_\varepsilon(A, B, C, D) = \bigcup \left\{ \sigma(M(E)) : E \in \mathbb{C}^{p\times m}, \|E\| \leq \varepsilon \right\}.$$

Note that $\sigma_\varepsilon(A, B, C, D) \supset \sigma_0(A, B, C, D) = \sigma(A)$. The following corollary of Theorem 1.2.1 and Remark 1.2.2 is immediate.

Corollary 1.2.4. *Let $\varepsilon \in \mathbb{R}$, with $\varepsilon > 0$ and $\varepsilon\|D\| < 1$. Then*

$$\begin{aligned} \sigma_\varepsilon(A, B, C, D)\backslash\sigma(A) &= \bigcup \left\{ \lambda \in \mathbb{C}\backslash\sigma(A) : \|G(\lambda)\| \geq \varepsilon^{-1} \right\} \\ &= \bigcup \left\{ \sigma(M(E)) : E \in \mathbb{C}^{p\times m}, \|E\| \leq \varepsilon, \mathit{rank}(E) = 1 \right\}. \end{aligned}$$

Remark 1.2.5. *Theorem 1.2.1 is implied by the more general development in [HP05, Theorem 5.2.9], where the norm need not be the 2-norm and the admissible perturbations E may be restricted to have a specified structure; see also [Kar03] for the case $D = 0$. The basic idea of our proof is from [HP05, Lemma 5.2.7], but the relationship between eigenvectors of $M(E)$ and singular vectors of $G(\lambda)$ revealed by our proof and developed further below is essential for this paper. Remark 1.2.2 may also be found in [HP05, Remark 5.2.20 (iii)]; this observation does not generally apply when structure is imposed on E. The sets σ_ε are called spectral value sets in [HP05, Kar03] and are also sometimes known as structured pseudospectra. In the case $B = C = I, D = 0$, the σ_ε are called pseudospectra [TE05]. In all the references just mentioned, the sets are defined with strict inequalities instead of the non-strict inequalities used above. Using our definition, the set σ_ε is compact for fixed ε.*

Definition 1.2.6. *An eigenvalue λ of A is* observable *if all its corresponding right eigenvectors x (with $Ax = \lambda x$, $x \neq 0$) satisfy $Cx \neq 0$, and it is* controllable *if all its corresponding left eigenvectors y (with $y^*A = \lambda y^*$, $y \neq 0$) satisfy $y^*B \neq 0$ [AM07, Corollary 6.9].*

Remark 1.2.7. *If an eigenvalue λ of A is either uncontrollable or unobservable, that is $Cx = 0$ or $y^*B = 0$ for some right eigenvector x or left eigenvector y, then from (1.3) we have either $M(E)x = \lambda x$ for all E or $y^*M(E) = \lambda y^*$ for all E. Therefore, λ is an eigenvalue of $M(E)$ for all E, so $\lambda \in \sigma_\varepsilon(A, B, C, D)$ for all ε and furthermore, by eigenvalue continuity, λ must be an isolated point of $\sigma_\varepsilon(A, B, C, D)$ for all sufficiently small ε.*

Next, we show that as long as E is chosen to have rank one, $E(I - DE)^{-1}$ can be simplified.

Lemma 1.2.8. *Let $\varepsilon \in \mathbb{R}$, with $\varepsilon > 0$ and $\varepsilon\|D\| < 1$. Then for all E with $\|E\| \leq \varepsilon$, we have $E(I-DE)^{-1} = (I-ED)^{-1}E$, and if $E = \varepsilon uv^*$, where $u \in \mathbb{C}^p$ and $v \in \mathbb{C}^m$ are arbitrary vectors with unit norm, we have*

$$E(I - DE)^{-1} = \frac{1}{1 - \varepsilon v^* Du} E.$$

Proof. The proof of the first statement is immediate. For the second, by the Sherman-Morrison-Woodbury formula [GV83], we have

$$\begin{aligned} E(I - DE)^{-1} &= \varepsilon uv^*(I - \varepsilon Duv^*)^{-1} = \varepsilon uv^* \left(I + \frac{\varepsilon}{1 - \varepsilon v^* Du} Duv^* \right) \\ &= \varepsilon uv^* + \frac{\varepsilon^2 v^* Du}{1 - \varepsilon v^* Du} uv^* = \frac{1}{1 - \varepsilon v^* Du} E. \end{aligned}$$

□

We now show that again provided E is rank-one, there is a key relationship between the right and left eigenvectors of $M(E)$ and the right and left singular vectors of $G(\lambda)$.

Theorem 1.2.9. *Let $\varepsilon \in \mathbb{R}$, with $\varepsilon > 0$ and $\varepsilon\|D\| < 1$, and suppose that $u \in \mathbb{C}^p$ and $v \in \mathbb{C}^m$ with $\|u\| = \|v\| = 1$ satisfy*

$$\varepsilon G(\lambda)u = v \quad \text{and} \quad \varepsilon v^* G(\lambda) = u^* \tag{1.7}$$

i.e., that u and v are respectively right and left singular vectors of $G(\lambda)$ corresponding to a singular value ε^{-1}. Then, defining $E = \varepsilon uv^*$, we have

$$M(E)x = \lambda x \quad \text{and} \quad y^* M(E) = \lambda y^* \tag{1.8}$$

with

$$x = \varepsilon(\lambda I - A)^{-1}Bu \quad \text{and} \quad y = \varepsilon(\lambda I - A)^{-*}C^* v \tag{1.9}$$

both nonzero, so that x and y are respectively right and left eigenvectors of $M(E)$ corresponding to the eigenvalue λ. Furthermore,

$$Cx + \varepsilon Du = v \quad \text{and} \quad B^* y + \varepsilon D^* v = u \tag{1.10}$$

and

$$u = \left(I - \varepsilon^2 D^* D\right)^{-1} \left(B^* y + \varepsilon D^* C x\right) \tag{1.11}$$

$$v = \left(I - \varepsilon^2 D D^*\right)^{-1} \left(Cx + \varepsilon D B^* y\right). \tag{1.12}$$

Conversely, suppose $E = \varepsilon uv^*$ for some $u \in \mathbb{C}^p$ and $v \in \mathbb{C}^m$ with $\|u\| = \|v\| = 1$

and that equation (1.8) *holds with x and y nonzero. Then we can scale x and y so that*

$$v^*Cx + \varepsilon v^*Du = 1 \quad \text{and} \quad u^*B^*y + \varepsilon u^*D^*v = 1 \tag{1.13}$$

and so that (1.9) *also holds, and, if we assume further that* (1.10) *holds, it follows that* (1.7) *holds.*

Proof. Suppose that (1.7) holds, so $G(\lambda)E = vv^*$ and hence (1.5) holds. Defining $Y = (I - DE)^{-1}C$ and $Z = (\lambda I - A)^{-1}BE$ as in the proof of the first part of Theorem 1.2.1 and using the same argument given there, we have (1.6) with $x = Zv \neq 0$, proving the first statement in (1.8). Hence

$$x = Zv = (\lambda I - A)^{-1}BEv = \varepsilon(\lambda I - A)^{-1}Bu$$

giving the first part of (1.9). Furthermore, we have $EG(\lambda) = uu^*$, so

$$u^*E(C(\lambda I - A)^{-1}B + D) = u^*.$$

Defining $\widetilde{Z} = EC(\lambda I - A)^{-1}$ and $\widetilde{Y} = B(I - ED)^{-1}$, we have $u^*\widetilde{Z}\widetilde{Y} = u^*$, so u is a left eigenvector of $\widetilde{Z}\widetilde{Y}$. Hence $y^*\widetilde{Y}\widetilde{Z} = y^*$, with $y = \widetilde{Z}^*u \neq 0$ a left eigenvector of $\widetilde{Y}\widetilde{Z}$. Multiplying through by $(\lambda I - A)$ on the right, we find

$$y^*B(I - ED)^{-1}EC = y^*(\lambda I - A) \tag{1.14}$$

with

$$y = \widetilde{Z}^*u = (\lambda I - A)^{-*}C^*E^*u = \varepsilon(\lambda I - A)^{-*}C^*v$$

so y is a left eigenvector of $A + B(I - ED)^{-1}EC$, and hence by Lemma 1.2.8 a left

eigenvector of $M(E)$. This proves the second statement in (1.8) and the second part of (1.9). Also, (1.9) implies that

$$Cx = \varepsilon C(\lambda I - A)^{-1}Bu \quad \text{and} \quad B^*y = \varepsilon B^*(\lambda I - A)^{-*}C^*v$$

and combining this with (1.7) we obtain (1.10). Solving (1.10) for u and v gives (1.11) and (1.12), which are well defined as $\varepsilon\|D\| < 1$. Note that the right-hand sides of (1.11) and (1.12) must have unit norm as we assumed *a priori* that u and v have unit norm.

Conversely, given (1.8), it follows that (1.6) holds, and hence using Lemma 1.2.8 we have

$$x = \psi(v^*Cx)(\lambda I - A)^{-1}Bu \quad \text{with} \quad \psi = \frac{\varepsilon}{1 - \varepsilon v^*Du}$$

giving the first parts of (1.13) and (1.9) by scaling x so that $\psi v^*Cx = \varepsilon$. Similarly, we have (1.14), which implies

$$y = \bar{\psi}(u^*B^*y)(\lambda I - A)^{-*}C^*v$$

giving the second parts of (1.13) and (1.9) by scaling y so that $\bar{\psi}u^*By = \varepsilon$. Note that scaling x and y does not change the norms of u and v which are one by assumption. It follows from (1.9) that

$$Cx + \varepsilon Du = \varepsilon G(\lambda)u \quad \text{and} \quad B^*y + \varepsilon D^*v = \varepsilon G(\lambda)^*v.$$

So, if u and v satisfy (1.10), then (1.7) must hold.

□

Remark 1.2.10. *Theorem 1.2.9 generalizes the far more trivial Lemma 1.1 of [GO11]. In the case $D = 0$, equations* (1.10), (1.11) *and* (1.12) *simplify considerably to $u = B^*y$, $v = Cx$.*

Remark 1.2.11. *If either $Cx = 0$ or $y^*B = 0$, then λ is also an eigenvalue of A and is either uncontrollable or unobservable. In this case, the normalization* (1.13) *is not possible, given the assumption $\varepsilon\|D\| < 1$, and consequently neither the assumptions of the theorem nor its converse can hold.*

1.2.1 The H_∞ norm for continuous-time systems

We start by defining spectral abscissa and spectral value set abscissa.

Definition 1.2.12. *The* spectral abscissa *of the matrix A is*

$$\alpha(A) = \max\{Re\ \lambda : \lambda \in \sigma(A)\},$$

with A (Hurwitz) stable *if $\alpha(A) < 0$. For $\varepsilon \geq 0$, the* spectral value set abscissa *is*

$$\alpha_\varepsilon(A, B, C, D) = \max\{Re\ \lambda : \lambda \in \sigma_\varepsilon(A, B, C, D)\} \tag{1.15}$$

with $\alpha_0(A, B, C, D) = \alpha(A)$.

Definition 1.2.13. *A* rightmost *point of a set $S \subset \mathbb{C}$ is a point where the maximal value of the real part of the points in S is attained.*

Remark 1.2.14. *Since $\sigma_\varepsilon(A, B, C, D)$ is compact, its rightmost points, that is the maximizers of the optimization problem in* (1.15)*, lie on its boundary. There can*

only be a finite number of these; otherwise, the boundary would need to contain an infinite number of points with the same real part which can be ruled out by an argument similar to [GO11, Lemma 2.5], exploiting [HP05, Lemma 5.3.30].

Definition 1.2.15. *The H_∞ norm of the transfer function G for continuous-time systems is*

$$\|G\|_\infty^c = \sup_{\delta>0} \left\{ \delta : \delta = \varepsilon^{-1} \text{ and } \alpha_\varepsilon(A,B,C,D) \geq 0 \right\}. \tag{1.16}$$

Remark 1.2.16. *The reciprocal of the H_∞ norm is called the* complex stability radius *[HP05, Section 5.3] (complex because complex perturbations are admitted even if the data are real, and radius in the sense of the perturbation space, not the complex plane). When $B = C = I$ and $D = 0$ this is also known as the* distance to instability *[VL85] for the matrix A.*

The following lemma states an equivalent definition of the H_∞ norm which is actually the standard one:

Lemma 1.2.17.

$$\|G\|_\infty^c = \begin{cases} \infty \text{ if } \alpha(A) \geq 0 \\ \sup_{\omega\in\mathbb{R}} \|G(i\omega)\| \text{ otherwise.} \end{cases} \tag{1.17}$$

Proof. Clearly, the supremum in (1.16) is bounded if and only if A is stable. For stable A and sufficiently small ε, rightmost points of the spectral value set are in the open left-half plane. If $\sigma_\varepsilon(A,B,C,D)$ does not intersect the imaginary axis for arbitrarily large ε then we take the supremum in (1.16) to be zero as no $\delta > 0$ satisfies the conditions, while by Corollary (1.2.4), $G(i\omega) = 0$ for all $\omega \in \mathbb{R}$ and hence the supremum in (1.17) is also zero. Otherwise, there must exist a smallest $\tilde{\varepsilon}$ for which a rightmost point $\tilde{\lambda}$ in $\sigma_{\tilde{\varepsilon}}(A,B,C,D)$ is on the imaginary axis, and

by choosing E to have rank one as explained in Remark 1.2.2 we have $\|E\| = \tilde{\varepsilon}$ and $\|G(\tilde{\lambda})\| = \tilde{\varepsilon}^{-1}$. Furthermore, supposing that there is another point on the imaginary axis with a norm larger than $\tilde{\varepsilon}$ leads immediately to a contradiction. $\square$

The standard method to compute the H_∞ norm is the Boyd-Balakrishnan-Bruinsma-Steinbuch algorithm [BB90a, BS90], henceforth called the BBBS algorithm, which generalizes and improves an algorithm of Byers [Bye88] for computing the distance to instability for A. The method relies on Lemma 1.2.17: for stable A, it needs only to maximize $\|G(i\omega)\|$ for $\omega \in \mathbb{R}$. The key idea is that, given any $\delta > 0$, it is possible to determine whether or not $\omega \in \mathbb{R}$ exists such that $\|G(i\omega)\| = \delta$ by computing all eigenvalues of an associated Hamiltonian matrix and determining whether any are imaginary. The algorithm is quadratically convergent, but the computation of the eigenvalues and the evaluation of the norm of the transfer matrix both require of the order of n^3 operations which is not practical when n is large. Furthermore, some implementations of the algorithm may be problematic because small real rounding errors in the imaginary eigenvalues may result in incorrectly concluding that there is no ω for which $\|G(i\omega)\|$ equals a given value δ, giving an incorrect upper bound on the H_∞ norm.

Our new algorithm is *not* based on evaluating the norm of the transfer matrix. Instead, it works directly with spectral value sets. The first step is to generalize the algorithm of [GO11] for approximating the pseudospectral abscissa of a matrix to the more general setting of the spectral value set abscissa $\alpha_\varepsilon(A, B, C, D)$ defined in (1.15), as explained in the next section. For this we will need the following concept.

Definition 1.2.18. *A* locally rightmost *point of a set $S \subset \mathbb{C}$ is a point λ which is a rightmost point of $S \cap \mathcal{N}$ for some neighborhood $\mathcal{N}$ of λ.*

We now state our main assumption:

Assumption 1.2.1. *Let $\varepsilon \in \mathbb{R}$, with $\varepsilon > 0$ and $\varepsilon\|D\| < 1$, and let $\lambda \notin \sigma(A)$ be a locally rightmost point of $\sigma_\varepsilon(A, B, C, D)$. Then:*

1. *the largest singular value ε^{-1} of $G(\lambda)$ is simple.*

2. *letting u and v be corresponding right and left singular vectors and setting $E = \varepsilon uv^*$, the eigenvalue λ of $M(E)$ is simple. (That λ is an eigenvalue of $M(E)$ follows from Theorem 1.2.9.)*

We shall assume throughout the paper that Assumption 1.2.1 holds. It can be shown by similar arguments to those used in [BLO03a, Section 2] that generically, that is for almost all quadruples (A, B, C, D), the largest singular value of $G(\lambda)$ is simple for all $\lambda \in \mathbb{C}\backslash\sigma(A)$.

Remark 1.2.19. *In the case $B = C = I, D = 0$, Part 2 of the Assumption is implied by Part 1 [GO11, Lemma 2.6].*

Although we do not use the transfer matrix $G(\lambda)$ as a computational tool, we instead use it to characterize maxima of the optimization problem on the right-hand side of (1.15). First, note that it follows from Corollary 1.2.4 that, for $\varepsilon > 0$, the definition of the spectral value set abscissa in (1.15) is equivalent to

$$\alpha_\varepsilon(A, B, C, D) = \max\left\{\text{Re } \lambda : \lambda \in \sigma(A) \text{ or } \|G(\lambda)\| \geq \varepsilon^{-1}\right\}. \tag{1.18}$$

The set of admissible λ must include $\sigma(A)$ because of the possibility that the spectral value set $\sigma_\varepsilon(A, B, C, D)$ has isolated points. Excluding such points, we obtain local optimality conditions for (1.18) as follows:

Lemma 1.2.20. *Under Assumption 1.2.1, a necessary condition for $\lambda \notin \sigma(A)$ to be a local maximizer of the optimization problem in* (1.18) *is*

$$\|G(\lambda)\| = \varepsilon^{-1} \quad \text{and} \quad v^* C \left(\lambda I - A\right)^{-2} Bu \in \mathbb{R}^{++}, \tag{1.19}$$

where $\mathbb{R}^{++}$ denotes the positive real numbers and u and v are respectively right and left singular vectors corresponding to the largest singular value ε^{-1} of $G(\lambda)$.

Proof. We have already observed that by compactness of $\sigma_\varepsilon(A, B, C, D)$, maximizers must lie on the boundary, and hence the first statement in (1.19) holds. The standard first-order necessary condition for ζ to be a local maximizer of an optimization problem $\max\{f(\zeta) : g(\zeta) \leq 0,\ \zeta \in \mathbb{R}^2\}$, when f, g are continuously differentiable and $g(\zeta) = 0$, $\nabla g(\zeta) \neq 0$, is the existence of a Lagrange multiplier $\mu \geq 0$ such that $\nabla f(\zeta) = \mu \nabla g(\zeta)$. In our case, identifying $\lambda \in \mathbb{C}$ with $\zeta \in \mathbb{R}^2$, the gradient of the maximization objective is the real number 1, while the constraint

$$\frac{1}{\varepsilon} - \|C \left(\lambda I - A\right)^{-1} B + D\|$$

is differentiable with respect to λ because of the first part of Assumption 1.2.1, and it has gradient $v^* C(\lambda I - A)^{-2} Bu$ using standard perturbation theory for singular values [GO11, Lemma 2.3]. Defining $E = \varepsilon u v^*$ and applying Theorem 1.2.9 we know that x and y as defined in (1.8) are respectively right and left eigenvectors of $M(E)$, with inner product

$$y^* x = \varepsilon^2 v^* C (\lambda I - A)^{-2} B u. \tag{1.20}$$

By the second part of Assumption 1.2.1, λ is a simple eigenvalue of $M(E)$ and so

$y^*x \neq 0$. Therefore, the constraint gradient is nonzero implying that the Lagrange multiplier $\mu \geq 0$ exists with $v^*C(\lambda I - A)^{-2}Bu = 1/\mu \in \mathbb{R}^{++}$. □

Corollary 1.2.21. *Let $\lambda \notin \sigma(A)$ be a local maximizer of the optimization problem in* (1.15) *and let u, v be respectively right and left singular vectors of $G(\lambda)$ corresponding to the largest singular value ε^{-1}. Let $E = \varepsilon uv^*$. Define x and y to be eigenvectors of $M(E)$ corresponding to the eigenvalue λ and scaled as in* (1.9)*. Then, under Assumption 1.2.1, y^*x must be real and positive.*

Proof. Since the optimization problems in (1.15) and (1.18) are equivalent, the result follows directly from Lemma 1.2.20 using (1.20). □

For this reason the following definition is very useful:

Definition 1.2.22. *A pair of complex vectors x and y is called* RP-compatible *if $\|x\| = \|y\| = 1$ and $y^*x \in \mathbb{R}^{++}$, and therefore in the interval $(0, 1]$.*

1.2.2 The H_∞ norm for discrete-time systems

We have analogous definitions relevant to discrete-time systems.

Definition 1.2.23. *The* spectral radius *of the matrix A is*

$$\alpha(A) = \max\{|\lambda| : \lambda \in \sigma(A)\},$$

with A (Schur) stable *if $\rho(A) < 0$. For $\varepsilon \geq 0$, the* spectral value set radius *is*

$$\rho_\varepsilon(A, B, C, D) = \max\{|\lambda| : \lambda \in \sigma_\varepsilon(A, B, C, D)\}. \tag{1.21}$$

Definition 1.2.24. *An* outermost *point of a set $S \subset \mathbb{C}$ is a point where the maximal value of the modulus of the points in S is attained.*

Definition 1.2.25. *The H_∞ norm of the transfer function G for discrete-time systems is*

$$\|G\|_\infty^d = \sup_{\delta>0} \left\{ \delta : \delta = \varepsilon^{-1} \text{ and } \rho_\varepsilon(A,B,C,D) \geq 1 \right\}. \tag{1.22}$$

The more standard equivalent definition of the H_∞ norm is given by:

Lemma 1.2.26.

$$\|G\|_\infty^d = \begin{cases} \infty \text{ if } \rho(A) \geq 1 \\ \sup_{\theta\in\mathbb{R}} \|G(e^{i\theta})\| \text{ otherwise.} \end{cases} \tag{1.23}$$

We omit the proof.

There is a variant of the BBBS algorithm for computing the discrete-time H_∞ norm, based on computing eigenvalues of symplectic pencils instead of Hamiltonian matrices [HS89, GDV98].

Definition 1.2.27. *A* locally outermost *point of a set $S \subset \mathbb{C}$ is a point λ which is an outermost point of $S \cap \mathcal{N}$ for some neighborhood $\mathcal{N}$ of λ.*

From Corollary 1.2.4, for $\varepsilon > 0$, the definition of the spectral value set radius in (1.21) is equivalent to

$$\rho_\varepsilon(A,B,C,D) = \max \left\{ |\lambda| : \lambda \in \sigma(A) \text{ or } \|G(\lambda)\| \geq \varepsilon^{-1} \right\}. \tag{1.24}$$

Excluding possibly isolated points in $\sigma(A)$, we obtain local optimality conditions for (1.24) as follows.

Lemma 1.2.28. *Extending Assumption 1.2.1 to locally outermost points in addition to locally rightmost points, a necessary condition for $\lambda \notin \sigma(A)$ to be a local*

maximizer of the optimization problem in (1.24) *is*

$$\|G(\lambda)\| = \varepsilon^{-1} \quad \text{and} \quad \lambda\left(v^* C\left(\lambda I - A\right)^{-2} Bu\right) \in \mathbb{R}^{++}, \tag{1.25}$$

where u and v are respectively right and left singular vectors corresponding to the largest singular value ε^{-1} of $G(\lambda)$.

The proof is the same as the proof of Lemma 1.2.20, except that the derivative of the complex modulus replaces the derivative of the real part.

So, we generalize the definition of RP-compatibility as follows:

Definition 1.2.29. *A pair of complex vectors x and y is called* RP(λ)-compatible *if $\|x\| = \|y\| = 1$ and y^*x is a positive real multiple of λ.*

1.3 Approximating the spectral value set abscissa and radius

We now show how to generalize the algorithm of [GO11] to approximate the spectral value set abscissa $\alpha_\varepsilon(A, B, C, D)$. We address the spectral value set radius $\rho_\varepsilon(A, B, C, D)$ in Section 1.3.5 below. We write *approximate*, not *compute*, because the algorithm aims to find local maximizers of the optimization problem in (1.15). There will be no assurance that these are global maximizers, but in practice this is very often the case, and even if it is not we obtain guaranteed lower bounds on α_ε. We remark that we could easily extend the criss-cross algorithm of [BLO03a] to compute the global optimum, but the cost would be comparable to that of the BBBS algorithm.

We have seen from Theorem 1.2.1 and Remark 1.2.2 that, without loss of generality, we can restrict the perturbation matrix $E \in \mathbb{C}^{p\times m}$ parameterizing $\sigma_\varepsilon(A, B, C, D)$ to have rank one. The idea of the algorithm is to generate a sequence of rank-one perturbations with norm ε, say $\varepsilon u_k v_k^*$, $k = 0, 1, 2, \ldots$, with $u_k \in \mathbb{C}^p$, $v_k \in \mathbb{C}^m$ and $\|u_k\| = \|v_k\| = 1$. The goal is to choose the sequence so that $M(\varepsilon u_k v_k^*)$ converges to a matrix $M(E)$ with an eigenvalue that is a rightmost point of $\sigma_\varepsilon(A, B, C, D)$. The primary matrix operation needed by the algorithm is the computation of eigenvalues with largest real part and their corresponding right and left eigenvectors, which can be done efficiently using an iterative method assuming A is sparse and $\max(m, p) \ll n$.

We know from Lemma 1.2.8 that

$$M(\varepsilon u_k v_k^*) = A + BF_kC \quad \text{where} \quad F_k = \frac{\varepsilon u_k v_k^*}{1 - \varepsilon v_k^* D u_k}.$$

The first step of the algorithm is to compute the rightmost eigenvalue λ_0 of A and corresponding RP-compatible right and left eigenvectors x_0, y_0. Assume that λ_0 is simple, controllable and observable and consider the matrix-valued function

$$K(t) = A + tBF_0C = A + tB\frac{\varepsilon u_0 v_0^*}{1 - \varepsilon v_0^* D u_0}C$$

where u_0 and v_0 are to be determined. Let $\lambda(t)$ denote the eigenvalue of $K(t)$ converging to λ_0. Using standard eigenvalue perturbation theory [HJ90, Theorem 6.3.12], [GO11, Lemma 2.1], we have

$$\lambda'(0) := \left.\frac{d\lambda(t)}{dt}\right|_{t=0} = \frac{y_0^* B \left(\frac{u_0 v_0^*}{1-\varepsilon v_0^* D u_0}\right) C x_0}{y_0^* x_0}. \tag{1.26}$$

We choose (u_0, v_0) to maximize the real part of this expression, as this choice is the one that moves $\lambda(t)$ to the right as fast as possible as t is increased from zero. Since x_0, y_0 are RP-compatible, their inner product $y_0^* x_0$ is a fixed positive real number. Therefore, we choose u_0 and v_0 as maximizers of

$$\max_{\substack{u\in\mathbb{C}^p,\|u\|=1\\ v\in\mathbb{C}^m,\|v\|=1}} \operatorname{Re}\left(y_0^* B\left(\frac{uv^*}{1-\varepsilon v^* D u}\right) C x_0\right). \tag{1.27}$$

When $D = 0$, we obtain $u_0 = B^* y_0/\|B^* y_0\|$ and $v_0 = C x_0/\|C x_0\|$. We will discuss the case $D \neq 0$ in detail below.

Now, let us consider how to compute (u_k, v_k) from (u_{k-1}, v_{k-1}) for $k = 1, 2, \ldots$. The algorithm computes the rightmost eigenvalue λ_k of $M(\varepsilon u_{k-1} v_{k-1}^*) = A + BF_{k-1}C$ and corresponding RP-compatible right and left eigenvectors x_k, y_k. Assume that λ_k is simple, controllable and observable and consider the matrix-valued linear function

$$K(t) = A + BF_{k-1}C + tB\left(F_k - F_{k-1}\right)C$$

with $t \in \mathbb{R}$, which satisfies $K(0) = M(\varepsilon u_{k-1} v_{k-1}^*)$ and $K(1) = M(\varepsilon u_k v_k^*)$. Define $\lambda(t)$ to be the eigenvalue of $K(t)$ that converges to λ_k as $t \to 0$. Again using standard first-order eigenvalue perturbation theory we have

$$\begin{aligned} \lambda'(0) &:= \left.\frac{d\lambda(t)}{dt}\right|_{t=0} = \frac{y_k^* B\left(F_k - F_{k-1}\right) C x_{k-1}}{y_{k-1}^* x_{k-1}} \\ &= \varepsilon \frac{y_k^* B\left(\frac{u_k v_k^*}{1-\varepsilon v_k^* D u_k}\right) C x_k}{y_k^* x_k} - \varepsilon \frac{y_k^* B\left(\frac{u_{k-1} v_{k-1}^*}{1-\varepsilon v_{k-1}^* D u_{k-1}}\right) C x_k}{y_k^* x_k}. \end{aligned} \tag{1.28}$$

The second term is fixed so we choose u_k, v_k to maximize the real part of the first term; clearly, the real part of the second term is a lower bound. Since x_k, y_k are

RP-compatible, their inner product $y_k^* x_k$ is a fixed positive real number. Therefore, we choose u_k and v_k as maximizers of

$$\max_{\substack{u\in\mathbb{C}^p,\|u\|=1\\ v\in\mathbb{C}^m,\|v\|=1}} \operatorname{Re}\left(y_k^* B\left(\frac{uv^*}{1-\varepsilon v^* Du}\right)Cx_k\right), \tag{1.29}$$

an optimization problem with the same form as (1.27). When $D = 0$, we obtain $u_k = B^* y_k / \|B^* y_k\|$ and $v_k = Cx_k / \|Cx_k\|$.

1.3.1 Solving the optimization subproblem when $D \neq 0$

We address here the following unconstrained optimization problem:

$$\max_{\substack{u\in\mathbb{C}^p,\|u\|=1\\ v\in\mathbb{C}^m,\|v\|=1}} \operatorname{Re}\, g(u,v) \tag{1.30}$$

with

$$g(u,v) = \frac{g_1(u,v)}{g_2(u,v)}, \quad g_1(u,v) = b^* uv^* c, \quad g_2(u,v) = 1 - \varepsilon v^* Du$$

which is equivalent to (1.27) when $b = B^* y_0$ and $c = Cx_0$ and to (1.29) when $b = B^* y_k$ and $c = Cx_k$. Assume furthermore that $b \neq 0$ and $c \neq 0$, otherwise the optimization problem (1.30) is trivial. Note that since we assume $\varepsilon\|D\| < 1$ the denominator $g_2(u,v)$ is always nonzero.

By compactness, a maximizer must exist. Let us define the Lagrangian

$$\mathcal{L}(u,v,\mu,\nu) = \operatorname{Re}\, g(u,v) - \frac{1}{2}\mu\left(u^*u - 1\right) - \frac{1}{2}\nu\left(v^*v - 1\right)$$

where $\mu \in \mathbb{R}$, $\nu \in \mathbb{R}$ are Lagrange multipliers, and impose the classical optimality conditions. Observe that replacing g by its complex conjugate $\overline{g}$ leaves the problem

unchanged.

Denoting the jth component of u by $u^j = u^j_R + iu^j_I$, let us consider the partial derivatives of g with respect to u^j_R and u^j_I and impose the conditions $\partial\mathcal{L}/\partial u^j_R = 0$ and $\partial\mathcal{L}/\partial u^j_I = 0$. Since the function $g(u,v)$ is holomorphic with respect to u^j, the Cauchy-Riemann equations yield

$$\begin{aligned} \mu u^j_R &= \frac{\partial \text{Re } g(u,v)}{\partial u^j_R} = \frac{\partial \text{Im } g(u,v)}{\partial u^j_I} \\ \mu u^j_I &= \frac{\partial \text{Re } g(u,v)}{\partial u^j_I} = -\frac{\partial \text{Im } g(u,v)}{\partial u^j_R} \end{aligned}$$

which imply, using $\dfrac{\partial}{\partial u^j} = \dfrac{1}{2}\left(\dfrac{\partial}{u^j_R} - i\dfrac{\partial}{u^j_I}\right)$,

$$\frac{\partial \text{Re } g(u,v)}{u^j} = \frac{1}{2}\mu\overline{u}^j \quad \text{and} \quad i\frac{\partial \text{Im } g(u,v)}{u^j} = \frac{1}{2}\mu\overline{u}^j$$

so that we can write

$$\frac{\partial g(u,v)}{\partial u^j} = \mu\overline{u}^j.$$

The gradients of g_1 and g_2 with respect to u are the row vectors

$$\nabla_u g_1(u,v) = v^* c b^* \quad \text{and} \quad \nabla_u g_2(u,v) = -v^*\varepsilon D.$$

Imposing $\nabla_u g(u,v) = \mu u^*$ we obtain

$$\frac{v^* c b^* \left(1 - \varepsilon v^* D u\right) - b^* u v^* c \left(-\varepsilon v^* D\right)}{\left(1 - \varepsilon v^* D u\right)^2} = \mu u^*. \tag{1.31}$$

A right multiplication by u gives a formula for the Lagrange multiplier

$$\mu = \frac{b^* u v^* c}{\left(1 - \varepsilon v^* D u\right)^2} \in \mathbb{R}. \tag{1.32}$$

At the maximal value of $g(u, v)$, we have

$$\operatorname{Re}\left(\frac{b^* u v^* c}{1 - \varepsilon v^* D u}\right) > 0 \quad \text{and} \quad \operatorname{Re}\left(1 - \varepsilon v^* D u\right) > 0$$

with the first inequality holding because we may take $u = b$, $v = c$, and the second because $\varepsilon \|D\| < 1$. Therefore, we have $\mu > 0$. Substituting (1.32) into (1.31), dividing through by $b^* u v^* c$ and conjugating gives

$$\beta b + \varepsilon D^* v = u \quad \text{with} \quad \beta = \frac{1 - \varepsilon u^* D^* v}{u^* b}. \tag{1.33}$$

In order to obtain a similar formula for the gradient with respect to v we replace g by $\overline{g}$ in (1.30), which has the same optimal solution. Doing so we obtain

$$\nabla_v \overline{g}_1(u, v) = u^* b c^* \quad \text{and} \quad \nabla_v \overline{g}_2(u, v) = 1 - u^* \varepsilon D^*.$$

Imposing $\nabla_v \overline{g}(u, v) = \nu v^*$ we get

$$\frac{u^* b c^* \left(1 - \varepsilon u^* D^* v\right) - c^* v u^* b \left(-\varepsilon u^* D^*\right)}{\left(1 - \varepsilon u^* D^* v\right)^2} = \nu v^*. \tag{1.34}$$

A right multiplication by v gives

$$\nu = \frac{c^* v u^* b}{\left(1 - \varepsilon u^* D^* v\right)^2} = \overline{\mu} = \mu. \tag{1.35}$$

Substituting (1.35) into (1.34), conjugating and dividing through by b^*uv^*c gives

$$\gamma c + \varepsilon Du = v \quad \text{with} \quad \gamma = \frac{1 - \varepsilon v^* Du}{v^* c}. \tag{1.36}$$

We have

$$\frac{\beta}{\gamma} = \frac{1 - \varepsilon u^* D^* v}{u^* b} \frac{v^* c}{1 - \varepsilon v^* Du} = \frac{\mu |1 - \varepsilon u^* D^* v|^2}{|b^* u|^2}, \tag{1.37}$$

a positive real number.

Now, combining equations (1.33) and (1.36), we find

$$u = \Delta \left(\beta b + \gamma \varepsilon D^* c \right) \quad \text{and} \quad v = \widetilde{\Delta} \left(\gamma c + \beta \varepsilon D b \right)$$

where

$$\Delta = \left(I - \varepsilon^2 D^* D \right)^{-1} \quad \text{and} \quad \widetilde{\Delta} = \left(I - \varepsilon^2 D D^* \right)^{-1}. \tag{1.38}$$

Note the equivalences

$$D\Delta = \widetilde{\Delta} D \quad \text{and} \quad \Delta D^* = D^* \widetilde{\Delta}.$$

Therefore, we have

$$u = \beta \widetilde{b} + \gamma \varepsilon D^* \widetilde{c} \quad \text{and} \quad v = \gamma \widetilde{c} + \beta \varepsilon D \widetilde{b}$$

where

$$\widetilde{b} = \Delta b \qquad \text{and} \qquad \widetilde{c} = \widetilde{\Delta} c. \tag{1.39}$$

From (1.37) we can assume

$$\gamma = \rho\beta, \qquad \text{with } \rho > 0 \tag{1.40}$$

which implies

$$u = \beta\left(\widetilde{b} + \rho\varepsilon D^*\widetilde{c}\right) \quad \text{and} \quad v = \beta\left(\rho\widetilde{c} + \varepsilon D\widetilde{b}\right). \tag{1.41}$$

Substituting u, v given by (1.41) into the function $g(u, v)$ to be optimized, we observe that the argument of β does not play any role, that is the function g depends only on $|\beta|$ whose purpose is to normalize the vectors u and v. So we can choose β real and positive. Note also that (1.33) and (1.36) remain unchanged if we scale u, v, β and γ by any unimodular scalar $e^{i\theta}$.

From (1.41) we require

$$0 = \|u\|^2 - \|v\|^2 = \beta^2\left(\|\widetilde{b}\|^2 + \rho^2\varepsilon^2\|D^*\widetilde{c}\|^2 - \rho^2\|\widetilde{c}\|^2 - \varepsilon^2\|D\widetilde{b}\|^2\right)$$

so

$$\rho = \sqrt{\frac{\|\widetilde{b}\|^2 - \|\varepsilon D\widetilde{b}\|^2}{\|\widetilde{c}\|^2 - \|\varepsilon D^*\widetilde{c}\|^2}}. \tag{1.42}$$

The last step is to choose $\beta > 0$ such that $\|u\| = 1$, which yields

$$\beta = \frac{1}{\|\widetilde{b} + \rho\varepsilon D^*\widetilde{c}\|}. \tag{1.43}$$

Substituting the optimal values (1.42)–(1.43) into (1.41) we obtain a pair (u, v) that solves (1.30). This pair is unique up to multiplication of u and v by a unimodular factor $e^{i\theta}$.

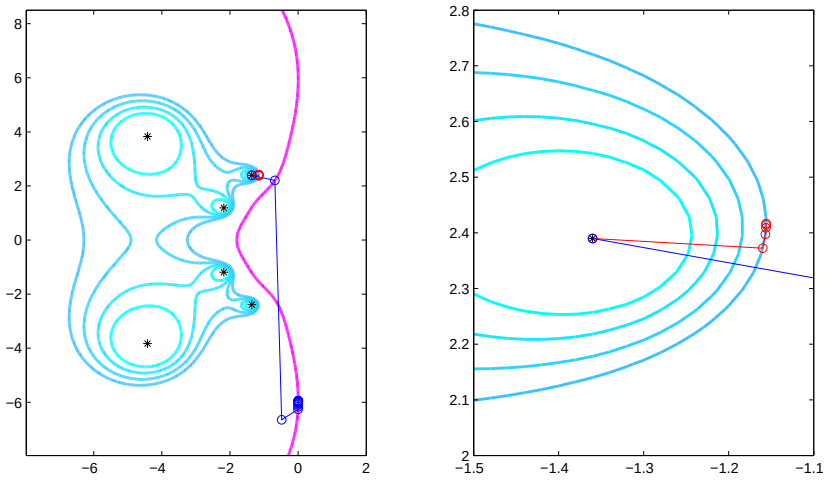

Figure 1.1: Iterates $\{\lambda_k\}$ on a simple example.

1.3.2 Basic algorithm statement

The derivation given above leads to the following algorithm. To make it well defined, we interpret "rightmost eigenvalue" below to mean the rightmost eigenvalue with largest imaginary part, in case there is more than one with largest real part, although in practice we make no attempt to break ties except in the case of complex conjugate pairs of eigenvalues of real matrices. We adopt the convention that the algorithm *breaks down* if it generates a rightmost eigenvalue λ_k which is not simple, controllable and observable. For later use, we include as inputs to the algorithm the scalar ε and an initial pair of RP-compatible vectors x_0, y_0. In the absence of any other estimates, these should in principle be set to right and left eigenvectors corresponding to λ_0, the rightmost eigenvalue of A that is simple, controllable and observable, although checking these conditions is not actually practical.

Algorithm SVSA0(ε, x_0, y_0)

Set u_0, v_0 to u and v as defined by (1.41) using (1.42), (1.43), (1.39) and (1.38), where $b = B^*y_0$ and $c = Cx_0$. Set $F_0 = \varepsilon u_0 v_0^*/(1 - \varepsilon v_0^* D u_0)$. For $k = 1, 2, \ldots$

> Let λ_k be the rightmost eigenvalue of $A + BF_{k-1}C$, with corresponding RP-compatible right and left eigenvectors x_k and y_k. Set u_k, v_k to u and v as defined by (1.41) using (1.42), (1.43), (1.39) and (1.38), where $b = B^*y_k$ and $c = Cx_k$. Set $F_k = \varepsilon u_k v_k^*/(1 - \varepsilon v_k^* D u_k)$.

Figure 1.1 shows iterates of Algorithm SVSA0 for computing the pseudospectral abscissa of a simple example [HP05, Example 5.2.21] with $n = 6$, $m = 6$ and $p = 1$. The contours are the boundaries of the spectral value sets for $\varepsilon = 0.5, 0.66, 0.83, 1.0, 8.77$. In the left panel we see that the convergence of the iterates for $\varepsilon = 8.77$ and in the right panel, a close-up view for $\varepsilon = 1$. In both cases we initialize x_0 and y_0 to right and left eigenvectors for the rightmost eigenvalue of A.

By construction, the sequence $\{\text{Re } \lambda_k\}$ is bounded above by $\alpha_\varepsilon(A, B, C, D)$. Also, the real part of the quantities $\lambda'(0)$ in (1.26) and (1.28) are nonnegative for all k. This is not enough to guarantee monotonicity of the sequence $\{\text{Re } \lambda_k\}$; however we discuss how to achieve monotonicity below in Section 1.3.4. First, we characterize fixed points of the iteration described by Algorithm SVSA0.

1.3.3 Fixed points

Now denote by $\mathcal{T}_\varepsilon$ the map that generates the pair (u_k, v_k) from the pair (u_{k-1}, v_{k-1}) as defined by Algorithm SVSA0. Equivalently, $\mathcal{T}_\varepsilon$ maps a rank-one matrix $u_{k-1}v_{k-1}^*$ with norm one to a rank-one matrix $u_k v_k^*$ with norm one.

Definition 1.3.1. *The pair* (u_{k-1}, v_{k-1}) *is a* fixed point *of the map* $\mathcal{T}_\varepsilon$ *if* $u_k =$

$e^{i\theta}u_{k-1}$, $v_k = e^{i\theta}v_{k-1}$ *for some* $\theta \in \mathbb{R}$. *Equivalently,* $u_{k-1}v_{k-1}^*$ *is a fixed point of the iteration if* $u_k v_k^* = u_{k-1}v_{k-1}^*$. *It follows that* $\lambda_k = \lambda_{k-1}$ *and, implicitly, that* λ_k *is simple, controllable and observable.*

Theorem 1.3.2. *Assume* $0 < \varepsilon\|D\| < 1$ *and suppose that* (u, v) *is a fixed point of* $\mathcal{T}_\varepsilon$ *corresponding to the rightmost eigenvalue* λ *of* $M(\varepsilon uv^*)$. *Then* $G(\lambda)$ *has a singular value equal to* ε^{-1}, *and furthermore, if it is the largest singular value, then* λ *satisfies the first-order necessary condition for a local maximizer of* (1.18) *given in* (1.19).

Conversely, assume $0 < \varepsilon\|D\| < 1$ *and suppose that* $\lambda \notin \sigma(A)$ *satisfies* (1.19), *and let* u *and* v *denote unit right and left singular vectors corresponding to the largest singular value* ε^{-1} *of* $G(\lambda)$. *Then* λ *is an eigenvalue of* $M(\varepsilon uv^*)$, *and if it is the rightmost eigenvalue and is simple, then* (u, v) *is a fixed point of* $\mathcal{T}_\varepsilon$.

Proof. Suppose (u, v) is a fixed point. This means that u and v satisfy (1.33) and (1.36) with $b = B^*y$, $c = Cx$, and x, y respectively right and left RP-compatible eigenvectors of $M(\varepsilon uv^*)$. By definition of x and y, it follows that (1.8) holds with $E = \varepsilon uv^*$, and by replacing x and y by βx and γy respectively, we have (1.10). Therefore, from the second part of Theorem 1.2.9, it follows that (1.7) also holds, that is u and v are respectively right and left singular vectors of $G(\lambda)$ corresponding to the singular value ε^{-1}, and if this is the largest singular value, then Lemma 1.2.20 shows that the first-order optimality conditions hold, using (1.20) and the positivity of y^*x. The latter is not changed by the scaling of x by β and y by γ because β/γ is real and positive, as shown in (1.37).

Conversely, if λ satisfies the first-order necessary conditions then ε^{-1} is the largest singular value of $G(\lambda)$ and the corresponding unit right and left singular vectors u and v satisfy the inequality in (1.19). Applying the first part of Theorem

1.2.9 with $E = \varepsilon uv^*$ we see that (1.8) holds for nonzero x and y defined by (1.9) so λ is an eigenvalue of $M(\varepsilon uv^*)$, and furthermore u and v satisfy (1.10), and therefore also (1.33) and (1.36) with $\beta = \gamma = 1$. Also, y^*x is real and positive using (1.19) and (1.20). Thus, if λ is the rightmost eigenvalue of $M(\varepsilon uv^*)$ and it is simple, then (x, y) is a fixed point of $\mathcal{T}_\varepsilon$. Note that Remark 1.2.11 shows that λ must be controllable and observable. □

As in [GO11, Section 4], we conjecture that the only *attractive* fixed points for Algorithm SVSA0 correspond to points λ that are local maximizers of (1.18).

1.3.4 A monotonic variant

Algorithm SVSA0 does not always generate a monotonically increasing sequence $\{\text{Re } \lambda_k\}$. Consider the continuous matrix family

$$N(t) = A + BF(t)C \quad \text{where} \quad F(t) = \frac{\varepsilon u(t)v(t)^*}{1 - \varepsilon v(t)^* Du(t)} \tag{1.44}$$

with

$$u(t) = \frac{tu_k + (1-t)u_{k-1}}{\|tu_k + (1-t)u_{k-1}\|} \quad \text{and} \quad v(t) = \frac{tv_k + (1-t)v_{k-1}}{\|tv_k + (1-t)v_{k-1}\|}. \tag{1.45}$$

The idea is that in case the rightmost eigenvalue of $N(1)$ does not have real part greater than that of λ_k, the rightmost eigenvalue of $N(0) = A + BF_{k-1}C$, we may instead choose $t \in (0, 1)$ so that the rightmost eigenvalue of $N(t)$ has this property.

As in [GO11, Section 6], using $'$ to denote differentiation with respect to t, we have

$$\begin{aligned} N'(t) &= B\left(\frac{\varepsilon u(t)v(t)^*}{1-\varepsilon v(t)^* Du(t)}\right)' C = N_1'(t) + N_2'(t) \\ N_1'(t) &= \frac{1}{1-\varepsilon v(t)^* Du(t)} B\Big(\varepsilon u(t)v(t)^*\Big)' C \\ N_2'(t) &= \varepsilon B u(t)v(t)^* C\Big(\frac{1}{1-\varepsilon v(t)^* Du(t)}\Big)'. \end{aligned}$$

Evaluating these at $t = 0$, we find

$$N'(0) = N_1'(0) + N_2'(0)$$

with

$$N_1'(0) = \frac{\varepsilon}{1-\varepsilon v_{k-1}^* Du_{k-1}} B\Big(\big(u_k - \mathrm{Re}(u_k^* u_{k-1})u_{k-1}\big)v_{k-1}^* + u_{k-1}\big(v_k - \mathrm{Re}(v_k^* v_{k-1})v_{k-1}\big)^*\Big)C,$$

$$N_2'(0) = \frac{\varepsilon^2}{(1-\varepsilon v_{k-1}^* Du_{k-1})^2}\Big(v_k^* Du_{k-1} - (v_{k-1}^* Du_{k-1})\mathrm{Re}(v_k^* v_{k-1}) + v_{k-1}^* Du_k - (v_{k-1}^* Du_{k-1})\mathrm{Re}(u_k^* u_{k-1})\Big)Bu_{k-1}v_{k-1}^* C.$$

Now let $\lambda(t)$ denote the eigenvalue of $N(t)$ converging to λ_k as $t \to 0$. From standard eigenvalue perturbation theory

$$\lambda'(0) = \frac{y_k^* N'(0) x_k}{y_k^* x_k} = \frac{\psi_k}{y_k^* x_k} \tag{1.46}$$

where

$$\psi_k = \frac{\varepsilon}{1-\varepsilon v_{k-1}^* D u_{k-1}}\Big((v_{k-1}^* C x_k)\big(y_k^* B u_k - (y_k^* B u_{k-1})\mathrm{Re}(u_k^* u_{k-1})\big) + (y_k^* B u_{k-1})\big(v_k^* C x_k - (v_{k-1}^* C x_k)\mathrm{Re}(v_k^* v_{k-1})\big)\Big) + \frac{\varepsilon^2 (y_k^* B u_{k-1})(v_{k-1}^* C x_k)}{(1-\varepsilon v_{k-1}^* D u_{k-1})^2}\Big(v_k^* D u_{k-1} - (v_{k-1}^* D u_{k-1})\mathrm{Re}(v_k^* v_{k-1}) + v_{k-1}^* D u_k - (v_{k-1}^* D u_{k-1})\mathrm{Re}(u_k^* u_{k-1})\Big). \quad (1.47)$$

We know from the RP-compatibility of x_k, y_k that the denominator of (1.46) is real and positive. Furthermore, if Re $\psi_k < 0$, we can change the sign of both u_k and v_k so that Re $\psi_k > 0$. Excluding the unlikely event that Re $\psi_k = 0$, defining u_k, v_k in this way guarantees that Re $\lambda(t) >$ Re λ_k for sufficiently small t, so that the following algorithm generates monotonically increasing {Re λ_k}. As before, we say that the algorithm breaks down if it generates a rightmost eigenvalue λ_k that is not simple, controllable and observable. Note however that provided x_0 and y_0 are RP-compatible right and left eigenvectors corresponding to a rightmost eigenvalue λ_0 that is simple, controllable and observable, and provided that Re $\lambda_1 >$ Re λ_0 and Re $\psi_k \neq 0$ for all k, then for $k > 1$, as long as λ_k is simple, it must also be controllable and observable.

Algorithm SVSA1(ε, x_0, y_0)

Set u_0, v_0 to u and v as defined by (1.41) using (1.42), (1.43), (1.39) and (1.38), where $b = B^* y_0$ and $c = C x_0$. Set $F_0 = \varepsilon u_0 v_0^* / (1 - \varepsilon v_0^* D u_0)$, and let λ_1 be the rightmost eigenvalue of $A + BF_0C$. For $k = 1, 2, \ldots$

1. Set x_k and y_k to be right and left eigenvectors of $A + BF_{k-1}C$ corresponding to the eigenvalue λ_k, normalized so they are RP-compatible. Set u_k, v_k to

u and v as defined by (1.41) using (1.42), (1.43), (1.39) and (1.38), where $b = B^* y_k$ and $c = C x_k$. Furthermore, compute ψ_k defined in (1.47). If Re $\psi_k < 0$ then replace u_k by $-u_k$ and v_k by $-v_k$. Set $t = 1$ and set z to the rightmost eigenvalue of $N(t)$ as defined in (1.44),(1.45).

2. Repeat the following zero or more times until Re $z >$ Re λ_k: replace t by $t/2$ and set z to the rightmost eigenvalue of $N(t)$ as defined in (1.44),(1.45).

3. Set $F_k = F(t)$ as defined in (1.44) and set $\lambda_{k+1} = z$.

Note that if t is always 1, then Algorithm SVSA1 generates the same iterates as Algorithm SVSA0, and if we omit Step 2, Algorithm SVSA1 reduces to Algorithm SVSA0.

Remark 1.3.3. *In the case $B = C = I$, $D = 0$, Algorithm SVSA1 reduces to a monotonically increasing algorithm for the pseudospectral abscissa as derived by a similar argument in [GO11]. However, the analogous Algorithm PSA1 stated there contains several errors: in Step 1, z should be set to the rightmost eigenvalue of $A + \varepsilon y x^*$; in Step 2, the stopping criterion should be Re $z >$ Re z_k; and in Step 3, z_{k+1}, not z_k, should be set to z. The errors in the algorithm statement did not affect the experimental results, except that Table 8.2 of [GO11] should show that the two Boeing examples need just one bisection each, not two.*

Remark 1.3.4. *We have established that, for sufficiently small ε, Algorithms SVSA0 and SVSA1 converge locally to rightmost points of the spectral value set with a linear rate of convergence. We omit the details since the proof is a rather lengthy generalization of the development in [GO11,Section 5] but offers little additional insight. In practice, we find that the algorithm normally converges to*

rightmost points, without assuming that ε is small, and though although convergence to global rather than local maximizers of (2.16) cannot be guaranteed, this is common in practice.

1.3.5 Approximating the spectral value set radius

Algorithms for the spectral value set radius $\rho_\varepsilon(A, B, C, D)$, defined in (1.21) and (1.24), are obtained by simple variants of Algorithms SVSA0 and SVSA1. Observe that

$$\left.\frac{d\left(|\lambda(t)|^2\right)}{dt}\right|_{t=0} = 2\,\mathrm{Re}\left(\overline{\lambda(0)}\lambda'(0)\right).$$

Thus, in order to maximize the modulus of the left-hand side of (1.26) or (1.28) instead of the real part, we will obtain the same optimization problems (1.27) and (1.29) as before if we simply require x_k and y_k to be $\mathrm{RP}(\overline{\lambda}_k)$-compatible, using Definition 1.2.29. (Note the conjugate.)

Likewise, in order to ensure that the modulus of the left-hand side of (1.46) is positive we again need only that Re ψ_k is positive, assuming that x_k and y_k are $\mathrm{RP}(\overline{\lambda}_k)$-compatible. This leads to the following algorithm. To ensure that it is well defined we say that if there is a tie for the outermost eigenvalue, the one whose nonnegative complex argument is closest to zero is used. We say that the algorithm breaks down if it generates an outermost eigenvalue that is not simple, controllable and observable. In the absence of other estimates, x_0 and y_0 are to be set to an $\mathrm{RP}(\overline{\lambda}_0)$-compatible pair of right and left eigenvectors for the outermost eigenvalue λ_0 that is simple, controllable and observable.

Algorithm SVSR1(ε, x_0, y_0)

Set u_0, v_0 to u and v as defined by (1.41) using (1.42), (1.43), (1.39) and (1.38),

where $b = B^* y_0$ and $c = Cx_0$. Set $F_0 = \varepsilon u_0 v_0^* / (1 - \varepsilon v_0^* D u_0)$, and let λ_1 be the outermost eigenvalue of $A + BF_0C$. For $k = 1, 2, \ldots$

1. Set x_k and y_k to be right and left eigenvectors of $A + BF_{k-1}C$ corresponding to the eigenvalue λ_k, normalized so they are $\mathrm{RP}(\overline{\lambda}_k)$-compatible. Set u_k, v_k to u and v as defined by (1.41) using (1.42), (1.43), (1.39) and (1.38), where $b = B^* y_k$ and $c = Cx_k$. Furthermore, compute ψ_k defined in (1.47). If $\mathrm{Re}\ \psi_k < 0$ then replace u_k by $-u_k$ and v_k by $-v_k$. Set $t = 1$ and set z to the outermost eigenvalue of $N(t)$ as defined in (1.44),(1.45).

2. Repeat the following zero or more times until $|z| > |\lambda_k|$: replace t by $t/2$ and set z to the outermost eigenvalue of $N(t)$ as defined in (1.44),(1.45).

3. Set $F_k = F(t)$ as defined in (1.44) and set $\lambda_{k+1} = z$.

Remark 1.3.5. *In the case $B = C = I$, $D = 0$, Algorithm SVSR1 reduces to a monotonically increasing algorithm for the pseudospectral radius as derived by a similar argument in [GO11]. However, Algorithm PSR1 stated in [GO11] contains the same errors as Algorithm PSA1 as described in Remark 1.3.3.*

Let us also define Algorithm SVSR0, a variant of Algorithm SVSA0 for the spectral value set radius, as Algorithm SVSR1 with Step 2 omitted. The fixed point theorem for Algorithm SVSA0, Theorem 1.3.2, extends in a straightforward way to Algorithm SVSR0, replacing "rightmost" by "outermost" and using the first-order optimality conditions for (1.24) given in (1.25). The local convergence results mentioned in Remark 1.3.4 also apply.

1.4 Approximating the H_∞ norm

Recall that the H_∞ norm was defined for the continuous-time and discrete-time case respectively in Sections 1.2.1 and 1.2.2.

1.4.1 The continuous-time case

We wish to compute $\|G\|_\infty^c$, defined in (1.16) and (1.17). Assume that A is Hurwitz stable, so the norm is finite. We start by observing that since the spectral value set abscissa $\alpha_\varepsilon(A, B, C, D)$ is a monotonically increasing function of ε, we need only to solve the equation

$$f(\varepsilon) = \alpha_\varepsilon(A, B, C, D) = 0 \tag{1.48}$$

for $\varepsilon \in \mathbb{R}^{++}$. The first step is to characterize how α_ε depends on ε.

Theorem 1.4.1. *Let $\lambda(\varepsilon)$ denote the rightmost point of $\sigma_\varepsilon(A, B, C, D)$ for $\varepsilon > 0$, $\varepsilon\|D\| < 1$, and assume that Assumption 1.2.1 holds for all such ε. Define $u(\varepsilon)$ and $v(\varepsilon)$ as right and left singular vectors with unit norm corresponding to ε^{-1}, the largest singular value of $G(\lambda(\varepsilon))$, and applying Theorem 1.2.9 with $E(\varepsilon) = \varepsilon u(\varepsilon)v(\varepsilon)^*$, define $x(\varepsilon)$ and $y(\varepsilon)$ by* (1.8) *and* (1.9)*. Furthermore, assume that for a given value $\hat{\varepsilon}$, the rightmost point $\lambda(\hat{\varepsilon})$ is unique. Then λ is continuously differentiable at $\hat{\varepsilon}$ and its derivative is real, with*

$$\left.\frac{d}{d\varepsilon}\alpha_\varepsilon(A, B, C, D)\right|_{\varepsilon=\hat{\varepsilon}} = \frac{d}{d\varepsilon}\lambda(\hat{\varepsilon}) = \frac{1}{y(\hat{\varepsilon})^* x(\hat{\varepsilon})} \in \mathbb{R}^{++}. \tag{1.49}$$

Proof. For the purposes of differentiation, we identify $\lambda \in \mathbb{C}$ with $\xi \in \mathbb{R}^2$ as in the proof of Lemma 1.2.20. The first part of Assumption 1.2.1 ensures that the largest

singular value of $G(\lambda)$ is differentiable with respect to λ and that the singular vectors $v(\varepsilon)$ and $u(\varepsilon)$ are well defined up to multiplication of both by a unimodular scalar, and that $E(\varepsilon)$ is not only well defined but differentiable with respect to ε. The second part ensures that $y(\varepsilon)^* x(\varepsilon)$ is nonzero, while the assumption that $\lambda(\hat{\varepsilon})$ is unique ensures that $\lambda(\varepsilon)$ is unique in a neighborhood of $\hat{\varepsilon}$ and, as an eigenvalue of $M(\varepsilon)$, is differentiable at $\hat{\varepsilon}$ using standard eigenvalue perturbation theory. As in the proof of Lemma 1.2.20, observe that

$$\frac{1}{\varepsilon} - \left\| C\left(\lambda I - A\right)^{-1} B + D \right\| = 0$$

so differentiating this with respect to ε at $\hat{\varepsilon}$ and using the chain rule yields

$$\left.\frac{d\lambda(\varepsilon)}{d\varepsilon}\right|_{\varepsilon=\hat{\varepsilon}} = \frac{1}{\varepsilon^2 v^* C(\lambda(\varepsilon)I - A)^{-2} B u}.$$

Furthermore, (1.20) follows (for $\lambda = \lambda(\varepsilon)$) from (1.9). Combining these with the first-order optimality conditions for (1.18) in (1.19) gives the result. □

Corollary 1.4.2. *Make the same assumptions as in Theorem 1.4.1, except normalize $x(\varepsilon)$ and $y(\varepsilon)$ so that they are RP-compatible. This is equivalent to scaling $x(\varepsilon)$ and $y(\varepsilon)$ by $1/\beta(\varepsilon)$ and $1/\gamma(\varepsilon)$ respectively where these are defined as in* (1.33) *and* (1.36)*, or equivalently* (1.43)*,* (1.40) *and* (1.42)*. So*

$$\left.\frac{d}{d\varepsilon}\alpha_\varepsilon(A,B,C,D)\right|_{\varepsilon=\hat{\varepsilon}} = \frac{d}{d\varepsilon}\lambda(\hat{\varepsilon}) = \frac{1}{\beta(\hat{\varepsilon})\gamma(\hat{\varepsilon})\big(y(\hat{\varepsilon})^* x(\hat{\varepsilon})\big)} \in \mathbb{R}^{++}. \tag{1.50}$$

Remark 1.4.3. *If A, B, C, D are all real then $\sigma_\varepsilon(A,B,C,D)$ is symmetric with respect to the real axis and hence its rightmost points must either be real or part of a conjugate pair. In the latter case, the assumption that $\lambda(\hat{\varepsilon})$ is unique does not*

hold but the result still holds as long as there is no third rightmost point.

The derivative formula (1.50) naturally leads to a formulation of Newton's method for computing $\|G\|_\infty^c$. We first state this in an idealized form:

Algorithm NHC0(ε^1) (Newton's method for H_∞ norm for continuous-time systems)

For $j = 1, 2, \ldots$

1. Compute the spectral value set abscissa $\alpha_{\varepsilon^j}(A, B, C, D)$, along with the rightmost point λ^j and corresponding RP-compatible right and left eigenvectors x^j, y^j and scalars β^j, γ^j defined as in (1.33) and (1.36), or equivalently (1.43), (1.40) using (1.38), (1.39) and (1.42), where $b = B^* y^j$ and $c = C x^j$.

2. Set
$$\varepsilon^{j+1} = \varepsilon^j - \big(\mathrm{Re}\ \lambda^j\big)\beta^j\gamma^j\big((y^j)^* x^j\big).$$

Let $\varepsilon_{\mathrm{opt}} = (\|G\|_\infty^c)^{-1}$, so that the rightmost point of $\sigma_{\varepsilon_{\mathrm{opt}}}(A, B, C, D)$ lies on the imaginary axis, and suppose this rightmost point is unique. It follows from Definition 1.2.15, Assumption 1.2.1, Theorem 1.4.1 that $\alpha_\varepsilon(A, B, C, D)$ is differentiable with respect to ε at $\varepsilon = \varepsilon_{\mathrm{opt}}$ and that the derivative is positive. Thus, the nonzero derivative condition for Newton's method to converge quadratically holds, so the sequence $\{\varepsilon^j\}$ defined by Algorithm NHC0 converges quadratically to $\varepsilon_{\mathrm{opt}}$ if $|\varepsilon^1 - \varepsilon_{\mathrm{opt}}|$ is sufficiently small.

In practice, each step of Algorithm NHC0 requires a call to Algorithm SVSA1 to compute the spectral value set abscissa via an "inner iteration" that must be terminated appropriately, perhaps without computing $\alpha_{\varepsilon^j}(A, B, C, D)$ very accurately. Furthermore, it is clearly desirable to "warm start" this computation by

providing as input to Algorithm SVSA1 not only the new value of ε, but also the final right and left eigenvectors already computed for the previous value of ε, as opposed to repeatedly initializing Algorithm SVSA1 with right and left eigenvectors corresponding to a rightmost eigenvalue of A. In the absence of any other estimates, x_0 and y_0 are to be set to an RP-compatible pair of right and left eigenvectors for the rightmost eigenvalue λ_0 that is simple, controllable and observable.

Algorithm NHC1(ε^1, x^0, y^0)

For $j = 1, 2, \ldots$

1. Call Algorithm SVSA1(ε^j,x^{j-1},y^{j-1}) to compute the spectral value set abscissa $\alpha_{\varepsilon^j}(A, B, C, D)$, also returning rightmost point λ^j, corresponding RP-compatible right and left eigenvectors x^j, y^j and corresponding scalars β^j, γ^j defined as in (1.33) and (1.36), or equivalently (1.43), (1.40) using (1.38), (1.39) and (1.42), where $b = B^* y^j$ and $c = C x^j$.

2. Set
$$\varepsilon^{j+1} = \varepsilon^j - \big(\text{Re } \lambda^j\big)\beta^j\gamma^j\big((y^j)^* x^j\big).$$

Since Newton's method may not converge if it is not initialized near the solution, it is standard practice to combine it with a bisection method to enforce convergence. While there are many variants of Newton-bisection methods in the literature, a good choice is the `rtsafe` routine [PFTV86, Hig88], a hybrid Newton-bisection method that maintains an interval known to contain the root, bisecting when the Newton step is either outside the interval or does not yield a sufficient decrease in the absolute function value (in this case, $|f(\varepsilon^j)| = |\alpha_{\varepsilon^j}(A, B, C, D)| = |\text{Re } \lambda^j|$). This safeguard is also useful in the unlikely event that f is not differentiable at some values of ε^j. If one has nonnegative lower and upper bounds h_{lb}

and h_{ub} on $\|G\|_\infty^c$, then these can be used to define initial lower and upper bounds $\varepsilon_{\mathrm{lb}}$ and $\varepsilon_{\mathrm{ub}}$ on $\varepsilon_{\mathrm{opt}}$ by

$$\varepsilon_{\mathrm{lb}} = \frac{1}{h_{\mathrm{ub}}} \leq \varepsilon_{\mathrm{opt}} \leq \varepsilon_{\mathrm{ub}} = \frac{1}{h_{\mathrm{lb}}}.$$

Assuming that such bounds are not known, we use the trivial initial lower bound $\varepsilon_{\mathrm{lb}} = 0$, and for $D \neq 0$, we use the initial upper bound $\varepsilon_{\mathrm{ub}} = 1/\|D\|$. For $D = 0$, we set $\varepsilon_{\mathrm{ub}}$ to the initial Newton step from 0, and if $f(\varepsilon_{\mathrm{ub}}) < 0$, we keep doubling it until $f(\varepsilon_{\mathrm{ub}}) \geq 0$. Putting all this together, we call the resulting algorithm NBHC1 (Newton-bisection method for the H_∞ norm for continuous-time systems).

1.4.2 The discrete-time case

In this case, the H_∞ norm is the quantity $\|G\|_\infty^d$ defined in (1.22) and (1.23). Assume that A is Schur stable so that the norm is finite. Equation (1.48) is replaced by

$$f(\varepsilon) = \rho_\varepsilon(A, B, C, D) - 1 = 0$$

where, as in the continuous-time case, f is a monotonically increasing function of ε. Defining $\lambda(\varepsilon)$ as the outermost point of $\sigma_\varepsilon(A, B, C, D)$, and assuming that it is nonzero and unique for a given value $\hat{\varepsilon}$, equation (1.49) is replaced by

$$\frac{d}{d\varepsilon}\rho_\varepsilon(A, B, C, D)\bigg|_{\varepsilon=\hat{\varepsilon}} = \frac{d}{d\varepsilon}|\lambda(\hat{\varepsilon})|, \qquad \frac{d}{d\varepsilon}\lambda(\hat{\varepsilon}) = \frac{1}{y(\hat{\varepsilon})^* x(\hat{\varepsilon})} \tag{1.51}$$

when the eigenvectors are normalized by (1.9). Applying the first-order optimality conditions for (1.24) given in (1.25), we find that the right-hand side of (1.51) is a

multiple of $\lambda(\hat{\varepsilon})$. Equation (1.50) is replaced by

$$\frac{d}{d\varepsilon}\rho_\varepsilon(A,B,C,D)\Big|_{\varepsilon=\hat{\varepsilon}} = \frac{d}{d\varepsilon}|\lambda(\hat{\varepsilon})|, \quad \frac{d}{d\varepsilon}\lambda(\hat{\varepsilon}) = \frac{1}{\beta(\hat{\varepsilon})\gamma(\hat{\varepsilon})\big(y(\hat{\varepsilon})^* x(\hat{\varepsilon})\big)}$$

when the eigenvectors are normalized to be $\mathrm{RP}(\overline{\lambda}(\hat{\varepsilon}))$-compatible, with the right-hand side again a multiple of $\lambda(\hat{\varepsilon})$. Algorithm NHC0 is replaced by:

Algorithm NHD0(ε^1) (Newton's method for H_∞ norm for discrete-time systems)

For $j = 1, 2, \ldots$

1. Compute the spectral value set radius $\rho_{\varepsilon^j}(A,B,C,D)$, along with the outermost point λ^j, corresponding $\mathrm{RP}(\overline{\lambda}^j)$-compatible right and left eigenvectors x^j, y^j and corresponding scalars β^j, γ^j defined as in (1.33) and (1.36), or equivalently (1.43), (1.40) using (1.38), (1.39) and (1.42), where $b = B^* y^j$ and $c = Cx^j$.

2. Set
$$\varepsilon^{j+1} = \varepsilon^j - \big(|\lambda^j| - 1\big)\beta^j\gamma^j\big|(y^j)^* x^j\big|.$$

This algorithm is quadratically convergent. A less idealized version is the following, where x^0, y^0 are set, in the absence of any other estimates, to right and left eigenvectors for λ_0, the outermost eigenvalue of A, normalized to be $\mathrm{RP}(\overline{\lambda}_0)$-compatible:

Algorithm NHD1(ε^1, x_0, y_0)

For $j = 1, 2, \ldots$

1. Call Algorithm SVSR1(ε^j, x^{j-1}, y^{j-1}) to compute the spectral value set radius $\rho_{\varepsilon^j}(A, B, C, D)$, also returning outermost point λ^j, corresponding RP $(\overline{\lambda^j})$-compatible right and left eigenvectors x^j, y^j and corresponding scalars β^j, γ^j defined as in (1.33) and (1.36), or equivalently (1.43), (1.40) using (1.38), (1.39) and (1.42), where $b = B^* y^j$ and $c = C x^j$.

2. Set
$$\varepsilon^{j+1} = \varepsilon^j - \big(|\lambda^j| - 1\big)\beta^j \gamma^j \big|(y^j)^* x^j\big|.$$

This algorithm has only local convergence guarantees, but can be globalized by combining it with the `rtsafe` routine, using an initial interval for ε as described above, giving an algorithm that we call NBHD1 (Newton-bisection method for H_∞ norm for discrete-time systems).

1.5 Numerical results

MATLAB codes SVSAR & HINFNORM implementing the spectral value set and H_∞ norm approximating algorithms (Algorithms SVSA1, SVSR1, NBHC1 and NBHD1) are freely available on the website http://cims.nyu.edu/~mert/software/hinfinity.html. We have tested this on many examples from the Compleib [Lei06] and EigTool [Wri02a] collections. The example data and the script used to generate the numerical examples in this section are also available on the website together with the code.

We use the following stopping condition in Step 1 of the SVSA1 and SVSR1

algorithms: termination takes place at iteration $k \geq 1$ if

$$|\phi(\lambda_k) - \phi(\lambda_{k-1})| < \max(1, |\phi(\lambda_{k-1}|) \texttt{ ftol}$$

where ϕ is the real part or modulus function, respectively, and we use the value $\texttt{ftol} = 10^{-12}$. We also set the maximum number of iterations of SVSA1 and SVSR1 to 100. The `rtsafe` routine, which is used to implement the NBHC1 and NBHD1 algorithms as explained above, uses both relative and absolute tolerances. We set the termination condition to

$$\left|\varepsilon_k - \varepsilon_{k-1}\right| \leq \max(\texttt{atol}, |\varepsilon_{k-1}| \texttt{ rtol})$$

where the tolerances for the relative error and absolute error are set to $\texttt{rtol}=10^{-10}$ and $\texttt{atol}=10^{-10}$ respectively.

1.5.1 Dense examples

We first test the algorithms on small dense problems for which we can compare the results with a standard implementation of the BBBS algorithm for computing the H_∞ norm, namely the `ss/norm` function in the Control Systems Toolbox of MATLAB [Mat], with tolerance 10^{-10}. For these problems, in Algorithms SVSA1 and SVSR1, all eigenvalues and right eigenvectors of the matrices $M_k := M(\varepsilon u_k v_k) = M + BF_kC$ are computed by calling the standard MATLAB eigenvalue routine `eig`. To compute the left eigenvectors of M_k, we make a second call to `eig`, computing the right eigenvectors of the transposed matrix M_k^T instead of inverting the possibly ill-conditioned matrix of right eigenvectors. Once left and right

eigenvectors are computed, they are normalized to satisfy the RP-compatibility condition.

Compleib is a database of continuous-time control-design examples, many from real applications, collected from the engineering literature. Each of these examples defines an "open-loop plant", described by a system of the form (1.1). In most cases, this open-loop system is unstable, and hence its H_∞ norm is $+\infty$, according to our definition (1.17). However, by designing an appropriate controller, the open-loop system can typically be stabilized, defining a "closed-loop plant" associated with a different system of the form (1.1): one with a finite H_∞ norm. We obtained these stabilized closed-loop systems by computing third-order controllers using the HIFOO package [BHLO06a].

Table 1.1 compares the results of the new NBHC1 algorithm with the BBBS algorithm for computing the H_∞ norm of the closed-loop systems obtained in this way for 17 different examples from Compleib. The columns headed n, m and p specify the dimension of the state space and the number of outputs and inputs in (1.1). The column headed $\|G\|_\infty^c$ shows the value of the norm computed by the new algorithm. The column headed "diff" shows the difference between the value of the H_∞ norm computed by Algorithm NBHC1 and that obtained using `ss/norm`; this is clarified further below. The mostly small values shown in this column indicates that our algorithm converges to a global maximizer of the optimization problem in (1.16) for all these examples with the exception of `CM4`, where our algorithm converges to a local maximizer of the spectral value set and hence an estimate of the H_∞ norm which is too small. However, when Algorithm NBHC1 was initialized with an eigenvector pair corresponding to a different eigenvalue of A (not a rightmost one) then convergence to the global maximizer was observed.

The columns headed "ni" and "bi" show the number of Newton steps and the number of bisection steps taken in the `rtsafe` routine by Algorithm NBHC1, so ni+bi is the number of calls to Algorithm SVSA1 for different values of ε^j. The first step in `rtsafe` is a bisection step, but after the first step, we observe that in most of the examples only Newton steps are taken and termination takes place rapidly (the `CM4` example is an exception).

According to (1.17), for stable A the norm $\|G\|_\infty^c$ is the maximum of $\|G(\lambda)\|$ over the imaginary axis. Algorithm NBHC1 does not verify the norm computation explicitly, but returns a value for $\hat{\varepsilon}$ for which the rightmost point $\hat{\lambda}$ of $\sigma_{\hat{\varepsilon}}(A, B, C, D)$ is estimated to lie on the imaginary axis, and hence $\hat{\varepsilon}^{-1}$ is an estimate of the norm. Thus, for validation purposes, we need to actually compute $\|G(i\mathrm{Im}\ \hat{\lambda})\|$ to obtain a guaranteed lower bound for $\|G\|_\infty^c$, neglecting rounding errors in the computation of the largest singular value. Similarly, the BBBS algorithm implemented in MATLAB returns a value that it estimates to be the norm, along with a second output argument, which we denote $\hat{\omega}$, which is the algorithm's estimate of the corresponding point on the imaginary axis where the maximum is attained. So, again for validation purposes, we compute $\|G(i\hat{\omega})\|$ to obtain a guaranteed lower bound on the norm. The quantities reported in the column `diff` are the differences of $\|G(i\mathrm{Im}\ \hat{\lambda})\|$ and $\|G(i\hat{\omega})\|$. When this number is positive, the new NBHC1 algorithm computed a better (larger) lower bound than the BBBS algorithm implemented in MATLAB while when it is negative, the new algorithm computed a worse (lower) lower bound.

The Compleib examples correspond to physical control systems that are all posed in continuous time. In order to create discrete-time examples, we sampled these systems with sampling time $T_s = 1$ (to obtain discrete-time open-loop sys-

tems of the form (1.2)) but these are usually not stable. So, we attempted to stabilize these discrete-time systems with the HIFOOd package [PWM10], which is an extension of HIFOO for discrete-time systems. In these examples, the order of the controller was taken to be 5 except for some of the smaller dimensional examples with $n < 10$ where we used a fourth order controller. Since the examples in Table 1.1 are posed in continuous time, some of them could not be stabilized in discrete-time by HIFOOd, so we added some new examples instead of these. The results for these discrete-time problems are shown in Table 1.2. Again, the mostly small numbers in the column headed "diff" indicate that Algorithm NBHD1 mostly computed globally optimal results, an exception being the `AC16` example where our algorithm apparently found a local maximizer. As previously, the Newton step is preferred to bisection most of the time. Note that HIFOOd usually yields control-systems that are barely stable so these examples are quite challenging and the performance of our algorithm on randomly created examples is much better. The validation of the results was done in the same way as explained for the continuous-time case.

1.5.2 Sparse matrices

As in [GO11], our MATLAB implementation supports three kinds of matrix input: dense matrices, sparse matrices and function handles, which specify the name of a MATLAB file implementing matrix-vector products. In the last two cases, we use the MATLAB routine `eigs`, which is an interface for ARPACK, a well-known code implementing the implicitly restarted Arnoldi method [LSY98]. Since `eigs` does not require M_k explicitly, but needs only the ability to do matrix-vector products with M_k, it also accepts as input either a sparse matrix or a

Table 1.1: Results for dense continuous-time problems from Compleib. The column headed "diff" shows the difference between the $\|G\|_\infty^c$ norm computed by Algorithm NBHC1 and that computed by the BBBS algorithm implemented in MATLAB. The last two columns show the number of Newton iterates and the number of bisection steps in Algorithm NBHC1.

example	n	m	p	$\|\|G\|\|_\infty^c$	diff	ni	bi
CBM	351	2	1	$2.630e-001$	$-3.9e-015$	5	1
CSE2	63	32	1	$2.034e-002$	$-2.7e-014$	9	4
CM1	23	3	1	$8.165e-001$	$+0.0e+000$	3	4
CM3	123	3	1	$8.214e-001$	$-8.2e-015$	6	3
CM4	243	3	1	$1.445e+000$	$-1.2e-001$	2	33
HE6	23	16	6	$4.929e+002$	$+0.0e+000$	25	13
HE7	23	16	9	$3.465e+002$	$+0.0e+000$	4	1
ROC1	12	2	2	$1.217e+000$	$-5.7e-004$	4	3
ROC2	13	1	4	$1.334e-001$	$+0.0e+000$	4	1
ROC3	14	11	11	$1.723e+004$	$+3.9e-006$	2	1
ROC4	12	2	2	$2.957e+002$	$-6.8e-004$	3	1
ROC5	10	2	3	$9.800e-003$	$+0.0e+000$	4	11
ROC6	8	3	3	$2.576e+001$	$+0.0e+000$	3	1
ROC7	8	3	1	$1.122e+000$	$-3.2e-010$	16	1
ROC8	12	7	1	$6.599e+000$	$+2.4e-010$	7	1
ROC9	9	5	1	$3.294e+000$	$-2.6e-012$	5	1
ROC10	9	2	2	$1.015e-001$	$-4.4e-016$	4	1

Table 1.2: Results for dense discrete-time version of problems from Compleib. The column headed "diff" shows the difference between the $\|G\|_\infty^d$ norm computed by Algorithm NBHD1 and that computed by the BBBS algorithm implemented in MATLAB. The last two columns show the number of Newton iterates and the number of bisection steps in Algorithm NBHD1.

example	n	m	p	ord	$\|\|G\|\|_\infty^d$	diff	ni	bi
AC5	8	4	4	4	$7.626e+001$	$-1.6e-011$	2	1
AC12	8	1	3	4	$1.082e+001$	$-4.4e-010$	2	3
AC15	8	6	4	4	$2.369e+001$	$-7.3e-003$	2	1
AC16	8	6	4	4	$1.818e+001$	$-1.1e-001$	4	3
AC17	8	4	4	4	$3.001e+005$	$-2.9e-004$	2	1
REA1	8	4	4	4	$7.438e+002$	$+2.0e-008$	3	1
AC1	10	2	3	5	$1.500e-001$	$-3.2e-004$	5	1
AC2	10	5	3	5	$3.056e-001$	$-3.9e-013$	4	1
AC3	10	5	5	5	$1.912e+001$	$-1.1e-009$	4	1
AC6	12	7	7	5	$5.294e+007$	$+5.9e+002$	3	4
AC11	10	5	5	5	$2.185e+007$	$+1.3e+000$	2	1
ROC3	16	11	11	5	$2.337e+001$	$-9.8e-010$	3	3
ROC5	12	2	3	5	$3.911e+003$	$+3.5e-007$	3	1
ROC6	10	3	3	5	$1.720e+001$	$-2.7e-004$	6	5
ROC7	10	3	1	5	$1.109e+000$	$-1.1e-007$	8	3
ROC8	14	7	1	5	$6.283e+004$	$+1.4e-003$	3	1
ROC9	11	5	1	5	$2.861e+001$	$-1.6e-003$	3	1

function handle. The last is crucial, because we must avoid computing the dense matrix $M_k = A + BF_kC$ explicitly. On the other hand, writing an efficient function to compute matrix-vector products with M_k is straightforward, and it is a handle for this function that we pass to `eigs`, which computes the largest eigenvalue with respect to real part or modulus, respectively. Our results are summarized in Tables 1.3 and 1.4 for continuous-time and discrete-time respectively.

As in the dense case, we compute the corresponding left eigenvector of M_k by a second call to `eigs`, to find the right eigenvector of the transposed matrix M_k^T. Thus, when the input to our implementation is a function handle, it must implement matrix-vector products with A^T as well as with A. The appearance of NaN in Tables 1.3 and 1.4 means that `eigs` failed to compute the desired eigenvalue to the default required accuracy.

We used the same tolerances as in the dense case: `ftol` = 10^{-12}, `rtol`=10^{-10} and `atol`=10^{-10}. Although we are not able to check global optimality as in the previous section, it seems likely that globally optimal values were again computed in many cases. As far as we know, this is the first time the H_∞ norm has been estimated for such large systems.

In Tables 1.3 and 1.4, the examples `NN18` and `HF1` are from Compleib; the other examples are from EigTool. For EigTool examples, we generate B,C and D matrices randomly. The column "shift" in Table 1.3 shows the shift applied to the corresponding A matrices to make it (Hurwitz) stable, i.e., we add the shift term to matrices A obtained from EigTool to make them Hurwitz stable. Similarly, the column "scale" in Table 1.4 shows the scaling applied, i.e., we divide the matrices A obtained from EigTool by the scale factor shown in column "scale" to make them Schur stable. In both continuous-time and discrete-time examples, we see

Table 1.3: Results of Algorithm NBHC1 on sparse continuous-time problems from EigTool and Compleib. The last four columns, respectively, show the computed $\|G\|_\infty^c$ norm, the number of Newton iterates, the number of bisection iterates and the total number of calls to the routine eigs.

example	shift	n	m	p	$\|\|G\|\|_\infty^c$	ni	bi	# `eigs`
`NN18`	0	1006	2	1	1.02336	3	1	24
`HF1`	0	130	2	1	1.41421	3	1	23
`convdiff_fd`	-90I	400	6	4	NaN	NaN	NaN	NaN
`dwave`	$-$I	2048	6	4	38020	4	1	21
`markov`	-2I	5050	6	4	6205.53	2	1	19
`olmstead`	-5I	500	6	4	504.564	3	1	18
`pde`	-10I	2961	6	4	368.75	4	1	35
`rdbrusselator`	$-$I	3200	6	4	1868.3	3	1	53
`skewlap3d`	0	24389	6	4	217.395	6	1	29
`sparserandom`	-3I	10000	6	4	141905	2	1	13
`supg`	$-$I	400	6	4	497.608	5	1	32
`tolosa`	0	4000	6	4	NaN	NaN	NaN	NaN

that except for the most challenging `pde`, `tolosa` and `conv_diff_fd` examples, the H_∞ norm computation reduces to just dozens of times as much work as the computation of the spectral abscissa or spectral radius alone.

1.6 Conclusion of the chapter

The H_∞ norm of a transfer matrix of a control system is the reciprocal of the largest value of ε such that the associated ε-spectral value set is contained in the stability region (the left half-plane for a continuous-time system and the unit disk for a discrete-time system). We extended an algorithm recently introduced by Guglielmi and Overton [GO11] for approximating the maximal real part or modulus of points in a matrix pseudospectrum to spectral value sets, characterizing its fixed points. We then introduced a Newton-bisection method to approximate the H_∞

Table 1.4: Results of Algorithm NBHD1 on sparse discrete-time problems from EigTool. The last four columns, respectively, show the computed $\|G\|_\infty^d$ norm, the number of Newton iterates, the number of bisection iterates and the total number of calls to the routine eigs.

example	scale	n	m	p	$\|\|G\|\|_\infty^d$	ni	bi	# `eigs`
`convdiff_fd`	1	400	6	4	NaN	NaN	NaN	NaN
`dwave`	1	2048	6	4	39026.7	4	1	22
`markov`	2	5050	6	4	12365.4	5	1	24
`olmstead`	3000	500	6	4	617.456	4	1	26
`pde`	10	2961	6	4	3645.62	3	9	631
`rdbrusselator`	120	3200	6	4	3891.82	5	1	24
`skewlap3d`	11000	24389	6	4	29357.9	6	1	31
`sparserandom`	3	10000	6	4	3.94791e+006	2	1	11
`supg`	1	400	6	4	499.276	5	1	78
`tolosa`	5000	4000	6	4	5.66293e+006	4	4	360

norm, for which each step requires optimization of the real part or the modulus over an ε-spectral value set. The algorithm is much faster than the standard Boyd-Balakrishnan-Bruinsma-Steinbuch algorithm to compute the H_∞ norm when the system matrices are large and sparse. The main work required by the algorithm is the computation of the spectral abscissa or radius of a sequence of large sparse matrices.

Chapter 2

Explicit Solutions for Root Optimization of a Polynomial Family

2.1 Introduction

A fundamental general class of problems is as follows: given a set of monic polynomials of degree n whose coefficients depend on parameters, determine a choice for these parameters for which the polynomial is stable, or show that no such stabilization is possible. Variations on this stabilization problem have been studied for more than half a century and several were mentioned in [BGL95] as being among the "major open problems in control systems theory".

In this paper, we show that there is one important special case of the polynomial stabilization problem which is explicitly solvable: when the dependence on parameters is affine and the number of parameters is $n-1$, or equivalently, when

there is a single affine constraint on the coefficients. In this setting, regardless of whether the coefficients are allowed to be complex or restricted to be real, the problem of globally minimizing the root radius (defined as the maximum of the moduli of the roots) or root abscissa (maximum of the real parts) may be solved efficiently, even though the minimization objective is nonconvex and not Lipschitz continuous at minimizers. The polynomial is Schur (respectively Hurwitz) stabilizable if and only if the globally minimal value of the root radius (abscissa) is less than one (zero). This particular class of polynomial stabilization problems includes two interesting control applications. The first is the classical static output feedback stabilization problem in state space with one input and $m-1$ independent outputs, where m is the system order [Che79a]. The second is a frequency-domain stabilization problem for a controller of order $m-2$ [Ran89, p. 651]. In the second case, if stabilization is not possible, then the minimal order required for stabilization is $m-1$. How to compute the minimal such order in general is a long-standing open question.

As a specific continuous-time example, consider the classical two-mass-spring dynamical system. It was shown in [HO06] that the minimal order required for stabilization is 2 and that the problem of maximizing the closed-loop asymptotic decay rate in this case is equivalent to the optimization problem

$$\min_{p \in P} \max_{z \in \mathbb{C}} \{\text{Re } z \mid p(z) = 0\}$$

where

$$P = \{(z^4 + 2z^2)(x_0 + x_1 z + z^2) + y_0 + y_1 z + y_2 z^2 \mid x_0, x_1, y_0, y_1, y_2 \in \mathbb{R}\}.$$

Thus P is a set of monic polynomials with degree 6 whose coefficients depend affinely on 5 parameters. A construction was given in [HO06] of a polynomial with one distinct root with multiplicity 6 and its local optimality was proved using techniques from nonsmooth analysis. Theorem 2.3.1 below validates this construction in a more general setting and proves global optimality.

The global minimization methods just mentioned are explained in a sequence of theorems that we present below. Theorem 2.2.1 shows that in the discrete-time case with real coefficients, the optimal polynomial can always be chosen to have at most two distinct roots, regardless of n, while Theorem 2.2.6 shows that in the discrete-time case with complex coefficients, the optimal polynomial can always be chosen to have just one distinct root. The continuous-time case is more subtle, because the globally infimal value of the root abscissa may not be attained. Theorem 2.3.1 shows that if it is attained, the corresponding optimal polynomial may be chosen to have just one distinct root, while Theorem 2.3.7 treats the case in which the optimal value is not attained. As in the discrete-time case, two roots play a role, but now one of them may not be finite. More precisely, the globally optimal value of the root abscissa may be arbitrarily well approximated by a polynomial with two distinct roots, only one of which is bounded. Finally, Theorem 2.3.8 shows that in the continuous-time case with complex coefficients, the optimal value is always attained by a polynomial with just one distinct root.

Our work was originally inspired by a combination of numerical experiments and mathematical analysis of special cases reported in [BLO01, BHLO06b, HO06]. As we began investigating a more general theory, A. Rantzer drew our attention to a remarkable 1979 Ph.D. thesis of Raymond Chen [Che79b], which in fact derived a method to compute the globally infimal value of the abscissa in the continuous-

time case with real coefficients. Chen also obtained some key related results for the discrete-time case with real coefficients, as explained in detail below. However, he did *not* provide generally applicable methods for constructing globally optimal or approximately optimal solutions, indeed remarking that he was lacking such methods [Che79b, p. 29 and p. 71]. Neither did he consider the complex case, for which it is a curious fact that our theorems are easier to state but apparently harder to prove than in the real case when the globally optimal value is attained.

This paper is concerned only with closed-form solutions. The problem of generating the entire root distribution of a polynomial subject to an affine constraint can also be approached by computational methods based on value set analysis (see [Bar93] for details). This has the advantage that it can be generalized to handle more than one affine constraint.

The theorems summarized above are presented in Sections 2.2 and 2.3 for the discrete-time and continuous-time cases, respectively. The algorithms implicit in the theorems are implemented in a publicly available MATLAB code. Examples illustrating various cases, including the subtleties involved when the globally optimal abscissa is not attained, are presented in Section 2.4. We make some concluding remarks about possible generalizations in Section 2.5.

2.2 Discrete-time stability

Let $\rho(p)$ denote the *root radius* of a polynomial p,

$$\rho(p) = \max\{|z| \mid p(z) = 0,\ z \in \mathbb{C}\}\,.$$

The following result shows that when the root radius is minimized over monic polynomials with real coefficients subject to a single affine constraint, the optimal polynomial can be chosen to have at most two distinct roots (zeros), and hence at least one multiple root when $n > 2$.

Theorem 2.2.1. *Let $B_0, B_1, \ldots, B_n$ be real scalars (with $B_1, \ldots, B_n$ not all zero) and consider the affine family of monic polynomials*

$$P = \{z^n + a_1 z^{n-1} + \ldots + a_{n-1} z + a_n \mid B_0 + \sum_{j=1}^{n} B_j a_j = 0, a_i \in \mathbb{R}\}.$$

The optimization problem

$$\rho^* := \inf_{p \in P} \rho(p)$$

has a globally optimal solution of the form

$$p^*(z) = (z - \gamma)^{n-k}(z + \gamma)^k \in P$$

for some integer k with $0 \leq k \leq n$, where $\gamma = \rho^*$.

Proof. Existence of an optimal solution is easy. Take any $p_0 \in P$ and define $P_0 = \{p \in P | \rho(p) \leq \rho(p_0)\}$. The set P_0 is bounded and closed. Since $\inf_{p \in P} \rho(p) = \inf_{p \in P_0} \rho(p)$, optimality is attained for some $p \in P_0 \subseteq P$.

We now prove the existence of an optimal solution that has the claimed structure. Let

$$p(z) = \prod_{i=1}^{n_1} (z + c_i) \prod_{i=n_1+1}^{n_2} (z^2 + 2d_i z + e_i)$$

be an optimal solution with $n_1 + 2(n_2 - n_1) = n$, $c_i, d_i, e_i \in \mathbb{R}$, $e_i > |d_i|$ and $\rho(p) = r$. We first show that there is an optimal solution whose roots all have

magnitude r. Consider therefore the perturbed polynomial

$$\begin{aligned} p_\Delta(z) &= \prod_{i=1}^{n_1}(z + c_i(1+\Delta_i)) \prod_{i=n_1+1}^{n_2} (z^2 + 2d_i z + e_i(1+\Delta_i)) \\ &= z^n + a_1(\Delta)z^{n-1} + \ldots + a_{n-1}(\Delta)z + a_n(\Delta), \end{aligned}$$

with $p_\Delta \in P$. The function

$$L(\Delta) = B_0 + B_1 a_1(\Delta) + \ldots + B_{n-1}a_{n-1}(\Delta) + B_n a_n(\Delta)$$

is a multilinear function from $\mathbb{R}^{n_2}$ to $\mathbb{R}$ and it satisfies $L(0) = 0$. Observe that the case $n_2 = 1$ can occur only if $n = 1$ or $n = 2$ and in that case the result is easy to verify, so assume that $n_2 \geq 2$. Consider now a perturbation Δ_j associated with a root or a conjugate pair of roots that do not have maximal magnitude (i.e., $1 \leq j \leq n_1$ and $|c_j| < r$, or $n_1 + 1 \leq j \leq n_2$ and $e_j < r^2$), and define

$$\mu_j := \frac{\partial L}{\partial \Delta_j}(0).$$

If $\mu_j \neq 0$ then by the implicit function theorem one can find some Δ in a neighborhood of the origin for which $\Delta_i < 0$ for $i \neq j$ with $L(\Delta) = 0$ and therefore for which $\rho(q_\Delta) < \rho^*$, contradicting the optimality of q. On the other hand, if $\mu_j = 0$, then, since L is linear in Δ_j, we have $L(0, \ldots, 0, \Delta_j, 0, \ldots, 0) = L(0) = 0$ for all Δ_j, and so Δ_j can be chosen so that the corresponding root or conjugate pair of roots has magnitude exactly equal to r. Thus, an optimal polynomial whose roots have equal magnitudes can always be found.

If $r = 0$, the result is established, so in what follows suppose that $r > 0$. We need to show that all roots can be chosen to be real. We start from some optimal

solution whose roots have magnitude $r > 0$, say

$$p(z) = \prod_{i=1}^{n_1}(z^2 + 2d_i z + r^2) \prod_{i=1}^{n_2}(z + r) \prod_{i=1}^{n_3}(z - r),$$

with $d_i \in \mathbb{R}$. Consider the perturbed polynomial

$$\begin{aligned} p_\Delta(z) &= \prod_{i=1}^{n_1} \left(z^2 + 2d_i(1 + \Delta_{2i})z + r^2(1 + \Delta_{2i-1})\right) \times \\ &\quad \prod_{i=1}^{n_2} (z + r(1 + \Delta_{2n_1+i})) \prod_{i=1}^{n_3} (z - r(1 + \Delta_{2n_1+n_2+i})) \\ &= z^n + a_1(\Delta)z^{n-1} + \ldots + a_{n-1}(\Delta)z + a_n(\Delta), \end{aligned}$$

now including a perturbation to d_i, so the function

$$L(\Delta) = B_0 + B_1 a_1(\Delta) + \ldots + B_{n-1}a_{n-1}(\Delta) + B_n a_n(\Delta)$$

is now a multilinear function from $\mathbb{R}^n$ to $\mathbb{R}$ that satisfies $L(0) = 0$. Let j be an index $1 \leq j \leq n_1$ for which $d_j \neq \pm r$ and define

$$\mu_j := \frac{\partial L}{\partial \Delta_{2j}}(0).$$

If $\mu_j \neq 0$ then by the same argument as above one can find a value of Δ in the neighborhood of the origin for which $\Delta_i < 0$ for $i \neq 2j$ with $L(\Delta) = 0$ and therefore for which $\rho(p_\Delta) < r$, which contradicts the optimality of p. So we must have $\mu_j = 0$. But then Δ_{2j} can be modified as desired while preserving the condition $L(\Delta) = 0$ and so in particular it may be chosen so that $d_i(1+\Delta_{2j}) = \pm r$. Repeated application of this argument leads to a polynomial $p^*(z)$ whose roots are all $\pm r$. □

Notice that $p^*(z) \in P$ if and only if γ satisfies a certain polynomial equality once k is fixed. The following corollary is a direct consequence of this fact, showing that γ in Theorem 2.2.1 can be computed explicitly.

Corollary 2.2.2. *Let γ be the globally optimal value whose existence is asserted in Theorem 2.2.1, and consider the set*

$$\Xi = \{r \in \mathbb{R} \mid g_k(r) = 0 \text{ for some } k \in \{0, 1, \ldots, n\}\}$$

where

$$g_k(z) = B_0 v_0 + B_1 v_1 z + \ldots + B_{n-1} v_{n-1} z^{n-1} + B_n v_n z^n$$

and $(v_0, \ldots, v_n)$ is the convolution of the vectors

$$\left(\binom{n-k}{0}, \binom{n-k}{1}, \ldots \binom{n-k}{n-k}\right) \text{ and } \left(\binom{k}{0}, -\binom{k}{1}, \ldots (-1)^k \binom{k}{k}\right)$$

for $k = 0, \ldots, n$. Then, $-\gamma$ is an element of Ξ with smallest magnitude.

Although Theorem 2.2.1 and Corollary 2.2.2 are both new, they are related to results in [Che79b], as we now explain. Let

$$H_P = \{(a_1, a_2, \ldots, a_n) \in \mathbb{R}^n \mid z^n + a_1 z^{n-1} + \ldots + a_n \in P\} \tag{2.1}$$

be the set of coefficients of polynomials in P. The set H_P is a hyperplane, by which we mean an $n-1$ dimensional affine subspace of $\mathbb{R}^n$. Let

$$C_r^n = \left\{(a_1, a_2, \ldots, a_n) \in \mathbb{R}^n \mid p(z) = z^n + a_1 z^{n-1} ... + a_n \text{ and } \rho(p) < r\right\}$$

be the set of coefficients of monic polynomials with root radius smaller than r.

Clearly, $\rho^* < r$ if and only if $H_P \cap C_r^n \neq \varnothing$. The root optimization problem is then equivalent to finding the infimum of r such that the hyperplane H_P intersects the set C_r^n. The latter set is known to be nonconvex, characterized by several algebraic inequalities, so this would appear to be difficult. However, since C_r^n is open and connected, it intersects a given hyperplane if and only if its convex hull intersects the hyperplane:

Lemma 2.2.3. (Chen [Che79b, Lemma 2.1.2]; see also [Che79a, Lemma 2.1]) *Let H be a hyperplane in $\mathbb{R}^n$, that is an $n-1$ dimensional affine subspace of $\mathbb{R}^n$, and let $S \subset \mathbb{R}^n$ be an open connected set. Then $H \cap S \neq \varnothing$ if and only if $H \cap \mathrm{conv}(S) \neq \varnothing$.*

The set $\mathrm{conv}(C_r^n)$ is an open simplex so it is easy to characterize its intersection with H_P:

Theorem 2.2.4. (Chen, special case of [Che79b, Prop. 3.1.7] and also Fam and Meditch [FM78], for the case $r = 1$; see also [**?**, Prop. 4.1.26].) *We have*

$$\mathrm{conv}(C_r^n) = \mathrm{conv}(\nu_1, \nu_2, \ldots, \nu_{n+1})$$

where the vertices

$$\nu_k = \{(a_1, a_2, \ldots, a_n) \in \mathbb{R}^n \mid (z-r)^{n-k}(z+r)^k = z^n + \sum_{j=1}^{n} a_j z^j\}$$

are the coefficients of the polynomials $(z-r)^{n-k}(z+r)^k$.

Since the optimum ρ^* is attained, the closure of $\mathrm{conv}(C_{\rho^*}^n)$ and the hyperplane H_P must have a non-empty intersection. Theorem 2.2.1 says that, in fact, the intersection of H_P with $C_{\rho^*}^n$ must contain at least one vertex of $\mathrm{conv}(C_{\rho^*}^n)$, and

Corollary 2.2.2 explains how to find it. In contrast, Chen uses Theorem 2.2.4 to derive a procedure (his Theorem 3.2.2) for testing whether the minimal value ρ^* of Theorem 2.2.1 is greater or less than a given value r (see also [Che79a, Theorem 2.6]). This could be used to define a bisection method for approximating ρ^*, but it would not yield the optimal polynomial $p^*(z)$. Note that the main tool used in the proof of Theorem 2.2.1 is the implicit function theorem, in contrast to the sequence of algebraic results leading to Theorem 2.2.4.

Remark 2.2.5. *The techniques used in Theorem 2.2.1 are all local. Thus, any locally optimal minimizer can be perturbed to yield a locally optimal minimizer of the form* $(z-\beta)^{n-k}(z+\beta)^k \in P$ *for some integer* k*, where* β *is the root radius attained at the local minimizer. Furthermore, all real roots* $-\beta$ *of the polynomials* g_k *in Corollary 2.2.2 define candidates for local minimizers, and while not all of them are guaranteed to be local minimizers, those with smallest magnitude (usually there will only be one) are guaranteed to be global minimizers.*

The work of Chen [Che79b] was limited to polynomials with real coefficients. A complex analogue of Theorem 2.2.1 is simpler to state because the optimal polynomial may be chosen to have only one distinct root, a multiple root if $n > 1$. However, the proof is substantially more complicated than for the real case and is given in [BGMO].

Theorem 2.2.6. *Let* $B_0, B_1, \ldots, B_n$ *be complex scalars (with* $B_1, \ldots, B_n$ *not all zero) and consider the affine family of polynomials*

$$P = \{z^n + a_1 z^{n-1} + \ldots + a_{n-1}z + a_n \mid B_0 + \sum_{j=1}^{n} B_j a_j = 0, a_i \in \mathbb{C}\}.$$

The optimization problem

$$\rho^* := \inf_{p \in P} \rho(p)$$

has an optimal solution of the form

$$p^*(z) = (z - \gamma)^n \in P$$

with $-\gamma$ *given by a root of smallest magnitude of the polynomial*

$$h(z) = B_n z^n + B_{n-1} \binom{n}{n-1} z^{n-1} + \ldots + B_1 \binom{n}{1} z + B_0.$$

2.3 Continuous-time stability

Let $\alpha(p)$ denote the *root abscissa* of a polynomial p,

$$\alpha(p) = \max \{ \mathrm{Re}(z) \mid p(z) = 0,\ z \in \mathbb{C} \}.$$

We now consider minimization of the root abscissa of a monic polynomial with real coefficients subject to a single affine constraint. In this case, the infimum may not be attained.

Theorem 2.3.1. *Let* $B_0, B_1, \ldots, B_n$ *be real scalars (with* $B_1, \ldots, B_n$ *not all zero) and consider the affine family of polynomials*

$$P = \{ z^n + a_1 z^{n-1} + \ldots + a_{n-1} z + a_n \mid B_0 + \sum_{j=1}^{n} B_j a_j = 0, a_i \in \mathbb{R} \}.$$

Let $k = \max\{j : B_j \neq 0\}$. *Define the polynomial of degree* k

$$h(z) = B_n z^n + B_{n-1}\binom{n}{n-1} z^{n-1} + \ldots + B_1\binom{n}{1} z + B_0.$$

Consider the optimization problem

$$\alpha^* := \inf_{p \in P} \alpha(p).$$

Then

$$\alpha^* = \min\left\{\beta \in \mathbb{R} \mid h^{(i)}(-\beta) = 0 \text{ for some } i \in \{0, \ldots, k-1\}\right\},$$

where $h^{(i)}$ *is the* i*-th derivative of* h. *Furthermore, the optimal value is attained by a minimizing polynomial* p^* *if and only if* $-\alpha^*$ *is a root of* h, *that is* $i = 0$, *and in this case we can take*

$$p^*(z) = (z - \gamma)^n \in P$$

with $\gamma = \alpha^*$.

The first part of this result, the characterization of the infimal value, is due to Chen [Che79b, Theorem 2.3.1]. Furthermore, Chen also observed the "if" part of the second statement, showing [Che79b, p.29] that if $-\alpha^*$ is a root of h (as opposed to one of its derivatives), the optimal value α^* is attained by the polynomial with a single distinct root α^*. However, he noted on the same page that he did not have a general method to construct a polynomial with an abscissa equal to a given value $\tilde{\alpha} > \alpha^*$. Nor did he characterize the case when the infimum is attained. We now address both these issues.

Because the infimum may not be attained, we cannot prove Theorem 2.3.1 using

a variant of the proof of Theorem 2.2.1. Instead, we follow Chen's development. Define the hyperplane of feasible coefficients as previously (see equation (2.1)). Let

$$S_\zeta^n := \left\{(a_1, a_2, \ldots, a_n) \in \mathbb{R}^n \mid z^n + a_1 z^{n-1} + \ldots + a_n = 0 \text{ implies } \mathrm{Re}(z) < \zeta\right\}$$

denote the set of coefficients of monic polynomials with root abscissa less than ζ, where $\zeta \in \mathbb{R}$ is a given parameter.

Definition 2.3.2. (S_ζ^n-stabilizability) *A hyperplane $H_P \subset \mathbb{R}^n$ is said to be S_ζ^n-stabilizable if $H_P \cap S_\zeta^n \neq \varnothing$.*

As in the root radius case, Lemma 2.2.3 shows that although S_ζ^n is a complicated nonconvex set, a hyperplane H_P is S_ζ^n-stabilizable if and only if H_P intersects $\mathrm{conv}S_\zeta^n$, a polyhedral convex cone which can be characterized as follows:

Theorem 2.3.3. (Chen [Che79b, Theorem 2.1.8]) *We have*

$$\mathrm{conv}(S_\zeta^n) = \nu + \mathrm{pos}(\{\tilde{e}_i\}) = \{\nu + \sum_{i=1}^{n} r_i \tilde{e}_i \mid r_i \geq 0\},$$

an open polyhedral convex cone with vertex

$$\nu = \sum_{j=1}^{n} \binom{n}{j} (-\zeta)^j e_j$$

and extreme rays

$$\tilde{e}_i = \sum_{j=i}^{n} \binom{n-i}{j-i} (-\zeta)^{j-i} e_j,$$

where $\{e_j\}_{j=1}^n$ is the standard basis of $\mathbb{R}^n$.

This leads to the following characterization of S_ζ^n-stabilizability:

Theorem 2.3.4. (Chen, a variant of [Che79b, Theorem 2.2.2]; see also [Che79a, Theorem 2.4]) *Define the hyperplane H_P as in equation* (2.1)*, the polynomial h and the integer k as in Theorem 2.3.1. Then the following statements are equivalent:*

1. *H_P is S_ζ^n-stabilizable*
2. *There exist nonnegative integers $j, \tilde{j}$ with $0 \leq j < \tilde{j} \leq k$ such that*

$$h^{(j)}(-\zeta)h^{(\tilde{j})}(-\zeta) < 0$$

where $h^{(j)}(-\zeta)$ denotes the j-th derivative of $h(z)$ at $z = -\zeta$.

To prove the last part of Theorem 2.3.1, we need the following lemma.

Lemma 2.3.5. *We have $h(-\zeta) = 0$ if and only if $(z-\zeta)^n \in P$. Furthermore, for $i \in \{1, 2, \ldots, k-1\}$, $h^{(i)}(-\zeta) = 0$ if and only if exactly one of the following two conditions hold:*

1. *$L_i \cap H_P = \varnothing$ and $h(-\zeta) \neq 0$*
2. *$L_i \in H_P$ and $h(-\zeta) = 0$*

where

$$L_i = \{\nu + r_i\tilde{e}_i \mid r_i \geq 0\},\ i = 1, 2, \ldots, n.$$

is the i-th extreme ray of the cone $\mathrm{conv}(S_\zeta^n)$ *given in Theorem 2.3.3.*

Proof. We have

$$h(-\zeta) = \sum_{j=0}^{n} B_j \binom{n}{j} (-\zeta)^j = B_0 + (B_1, B_2, \ldots, B_n) \cdot \nu$$

where $\cdot$ denotes the usual dot product in $\mathbb{R}^n$. Therefore,

$$\begin{aligned} h(-\zeta) = 0 &\iff B_0 + (B_1, B_2, \ldots, B_n) \cdot \nu = 0 \\ &\iff \nu \in H_P \\ &\iff z^n + \sum_{i=1}^{n} \nu_j z^{n-j} = (z-\zeta)^n \in P \end{aligned} \tag{2.2}$$

proves the first part of the lemma. Now, let $i \in \{1, 2, \ldots, k-1\}$. A straightforward calculation gives

$$\begin{aligned} h^{(i)}(-\zeta) &= \frac{n!}{(n-i)!} \sum_{j=i}^{n} B_j \binom{n-i}{j-i} (-\zeta)^{j-i} \\ &= \frac{n!}{(n-i)!} (B_1, B_2, \ldots, B_n) \cdot \tilde{e}_i \end{aligned}$$

Hence,

$$\begin{aligned} h^{(i)}(-\zeta) = 0 &\iff (B_1, B_2, \ldots, B_n) \cdot \tilde{e}_i = 0 \\ &\iff L_i \in H := \{(a_1, a_2, \ldots, a_n) \mid -(B_1, B_2, \ldots, B_n) \cdot \nu \\ &\qquad + \sum_{j=1}^{n} B_j a_j = 0\} \end{aligned}$$

If $B_0 = -(B_1, B_2, \ldots, B_n) \cdot \nu$, then $H = H_P$, $\nu \in H_P$ and from (2.2), we get $h(-\zeta) = 0$ (case (1)). Otherwise, the hyperplane H is parallel to H_P and $H \cap H_P = \varnothing$, so that $L_i \cap H_P = \varnothing$, and also $h(-\zeta) \neq 0$ (otherwise by (2.2), $\nu \in L_i \cap H_P$ which would be a contradiction); this is case (2). $\square$

Now we are ready to complete the proof of Theorem 2.3.1.

Proof. Chen's theorem [Che79b, Theorem 2.3.1] establishes the characterization of the optimal value,

$$\inf_{p\in P} \alpha(p) = \alpha^* = \min\{\beta \mid \prod_{i=0}^{k-1} h^{(i)}(-\beta) = 0\}.$$

Let $l \in \{0, 1, \ldots, k-1\}$ be the smallest integer such that $h^{(l)}(-\alpha^*) = 0$. If $l = 0$, then $-\alpha^*$ is a root of h and by Lemma 2.3.5, $p^*(z) = (z-\gamma)^n \in P$ is an optimizer with $\gamma = \alpha^*$.

Suppose now that $l > 0$. We will show that the infimum is not attained. Suppose the contrary, that is $H_P \cap \mathrm{cl}(S^n_{\alpha^*}) \neq \varnothing$ so that $H_P \cap \mathrm{cl}(\mathrm{conv}S^n_{\alpha^*}) \neq \varnothing$. Without loss of generality, assume $B_k > 0$ so that $h^{(k)}$ is the constant function $k!B_k > 0$ and the derivatives $h^{(j)}$, $j = 1, 2, \ldots, k-1$ each have leading coefficient (coefficient of z^{k-j}) also having positive sign. By Theorem 2.3.4, $h^{(j)}(-\tilde{\alpha}) > 0$ for any $j = 1, 2, .., k$ and $\tilde{\alpha} < \alpha^*$ and, in addition, $h^{(j)}(-\alpha^*) > 0$ for $0 \leq j < l$. By continuity of $h^{(j)}$, we have

$$h^{(j)}(-\alpha^*) \begin{cases} > 0 & \text{if } 0 \leq j < l \\ = 0 & \text{if } j = l \\ \geq 0 & \text{if } l < j < k \\ > 0 & \text{if } j = k \end{cases}$$

It thus follows from Theorem 2.3.4 that H_p is not $S^n_{\alpha^*}$-stabilizable, which means $H_P \cap S^n_{\alpha^*} = \varnothing$, or equivalently, by Lemma 2.2.3, that $H_P \cap \mathrm{conv}S^n_{\alpha^*} = \varnothing$. Since $\mathrm{conv}S^n_{\alpha^*}$ is an open set, it follows from the assumption made above that its boundary intersects H_P. Pick a point $y \in H_P \cap \mathrm{bd}(\mathrm{conv}S^n_{\alpha^*})$. It is easy to show that H_P is a supporting hyperplane to the convex cone $\mathrm{conv}S^n_{\alpha^*}$ at the boundary point y.

Since every hyperplane supporting a convex cone must pass through the vertex of the cone [HUL93, A.4.2], it follows that $\nu \in H_P$. On the other hand, since $l > 0$, Lemma 2.3.5 implies $L_l \cap H_p = \varnothing$. This is a contradiction. $\square$

Remark 2.3.6. *If $-\beta$ is a real root of $h(z)$, then $(z-\beta)^n \in P$. Such a polynomial is often, though not always, a local minimizer of $\alpha(p)$, but it is a global minimizer if and only if $-\beta$ is the largest such real root* and *no other roots of derivatives of h are larger than $-\beta$.*

We now address the case where the infimum is not attained.

Theorem 2.3.7. *Assume that $-\alpha^*$ is not a root of h. Let ℓ be the smallest integer $i \in \{1, \ldots, k-1\}$ for which $-\alpha^*$ is a root of $h^{(i)}$. Then, for all sufficiently small $\epsilon > 0$ there exists a real scalar M_ϵ for which*

$$p_\epsilon(z) := (z - M_\epsilon)^m (z - (\alpha^* + \epsilon))^{n-m} \in P$$

where $m = \ell$ or $\ell + 1$, and $M_\epsilon \to -\infty$ as $\epsilon \to 0$.

Proof. By Theorem 2.3.1, the optimal abscissa value α^* is not attained. Without loss of generality, assume $\alpha^* = 0$. Otherwise, write $z = \tilde{z} + \alpha^*$ and rewrite P as the set of monic polynomials in $\tilde{z}$ with an affine constraint.

For $0 < m \leq n$, we have $p_\epsilon(z) = (z + K)^m (z - \epsilon)^{n-m} \in P$ if and only if its coefficients are real and

$$
\begin{aligned}
0 &= \left(B_0 + B_1\binom{n-m}{1}(-\epsilon) + B_2\binom{n-m}{2}(-\epsilon)^2 + \cdots + B_{n-m}(-\epsilon)^{n-m}\right) \\
&\quad + \binom{m}{1}\left(B_1 + B_2\binom{n-m}{1}(-\epsilon) + B_3\binom{n-m}{2}(-\epsilon)^2 + \cdots B_{n-m+1}(-\epsilon)^{n-m}\right)K \\
&\quad + \binom{m}{2}\left(B_2 + B_3\binom{n-m}{1}(-\epsilon) + \cdots + B_{n-m+2}(-\epsilon)^{n-m}\right)K^2 \\
&\quad + \cdots + \left(B_m + B_{m+1}\binom{n-m}{1}(-\epsilon) + \cdots + B_n(-\epsilon)^{n-m}\right)K^m \\
&= \eta_0(\epsilon) + \eta_1(\epsilon)K + \cdots \eta_m(\epsilon)K^m =: f_\epsilon(K).
\end{aligned}
$$

Thus, $p_\epsilon \in P$ if and only if K is a real root of f_ϵ, a polynomial of degree m whose coefficients depend on ϵ. By Theorem 2.3.4, the $h^{(j)}(\epsilon)$ have the same sign for all $\epsilon > 0$ and for all $j \in \{0, 1, \ldots, k\}$, which we take to be positive. By the definiton of ℓ, $h^{(j)}(0) > 0$ for $j < \ell$ and $h^{(\ell)}(0) = 0$ which gives $\eta_j(0) = \binom{m}{j}\frac{h^{(j)}(0)}{n!(n-j)!} > 0$ for $j < \ell$ and similarly $\eta_\ell(0) = 0$. We have also

$$
\eta_m(\epsilon) = \sum_{j=m}^{n} B_j\binom{n-m}{j-m}(-\epsilon)^{j-m} = \frac{(n-m)!}{n!}h^{(m)}(-\epsilon) \tag{2.3}
$$

and

$$
\begin{aligned}
\eta_{m-1}(\epsilon) &= m\sum_{j=m}^{n} B_{j-1}\binom{n-m}{j-m}(-\epsilon)^{j-m} && (2.4)\\
&= m\frac{(n-m)!}{n!}\left((n-m+1)h^{(m-1)}(-\epsilon) + \epsilon h^{(m)}(-\epsilon)\right). && (2.5)
\end{aligned}
$$

Let $m = \ell$. We have $\eta_\ell(\epsilon) > 0$ for $\epsilon < 0$ and $\eta_\ell(0) = 0$. The polynomial η_ℓ might change sign around 0, depending on the multiplicity of 0 as a root. If 0 is a root of η_ℓ with an odd multiplicity, $\eta_\ell(\epsilon) < 0$ for $\epsilon > 0$ small enough and so the

coefficients of f_ϵ have one and only one sign change. By Descartes' rule of signs, f_ϵ has one and only one root K with positive real part which must therefore be real. Setting $M_\epsilon = -K$, we have $p_\epsilon(z) = (z-\epsilon)^{n-m}(z+K)^m \in P$ as desired. If the multiplicity is even, then the multiplicity of 0 as a root of $h^{(\ell)}$ is also even by (2.3). Then, $h^{(\ell+1)}$ must have 0 as a root with odd multiplicity and $h^{(\ell+1)}$ changes sign around 0. Set $m = \ell + 1$ in this case and repeat a similar argument: By (2.3), η_m changes sign around 0, i.e. $\eta_m < 0$ for $\epsilon > 0$ small enough. Furthermore, from (2.5), $\eta_{m-1} > 0$ for $\epsilon > 0$, ϵ small enough. As a result, the coefficients of f_ϵ have one and only one sign change, for $\epsilon > 0$, ϵ small enough. We again get the existence of p_ϵ in P with the desired structure.

Finally, let us show that $M_\epsilon \to -\infty$. Suppose this is not the case. Then, there exists a sequence $\epsilon_\kappa \downarrow 0$ and a positive number R such that $\sup_\kappa \rho(p_{\epsilon_\kappa}) \leq R$. Since $\mathrm{cl}(C_R^n)$ is compact by Theorem 2.2.4, there exists a positive constant $\tilde{R}$ such that all of the coefficients of the polynomial p_{ϵ_κ} are bounded by $\tilde{R}$, uniformly over κ. By compactness, there exists a subsequence $p_{\epsilon_{\kappa_\iota}}$ converging to a limit p_* pointwise. Furthermore, $p_* \in P$ since P is closed. By continuity of the abscissa mapping, $\alpha(p_*) = \lim_{\iota\to\infty} \alpha(p_{\epsilon_{\kappa_\iota}}) = 0$. This implies that the optimal abscissa is attained on P, which is a contradiction. □

Theorem 2.3.1 showed that in the real case the infimal value is not attained if and only if the polynomial h has a derivative of any order between 1 and $k-1$ with a real root to the right of the rightmost real root of h. However, it is not possible that a derivative of h has a complex root to the right of the rightmost complex root of h. This follows immediately from the Gauss-Lucas theorem, which states that the roots of the derivative of a polynomial p must lie in the convex hull of the roots of p [BLO04, Mar66]. This suggests that the infimal value of the optimal

abscissa problem with complex coefficients is always attained at a polynomial with a single distinct root, namely a rightmost root of h. Indeed, this is established in the following theorem, whose proof can be found in [BGMO].

Theorem 2.3.8. *Let $B_0, B_1, \ldots, B_n$ be complex scalars (with $B_1, \ldots, B_n$ not all zero) and consider the affine family of polynomials*

$$P = \{z^n + a_1 z^{n-1} + \ldots + a_{n-1} z + a_n \mid B_0 + \sum_{j=1}^{n} B_j a_j = 0, a_i \in \mathbb{C}\}.$$

The optimization problem

$$\alpha^* := \inf_{p \in P} \alpha(p)$$

has an optimal solution of the form

$$p^*(z) = (z - \gamma)^n \in P$$

with $-\gamma$ given by a root with largest real part of the polynomial h where

$$h(z) = B_n z^n + B_{n-1} \binom{n}{n-1} z^{n-1} + \ldots + B_1 \binom{n}{1} z + B_0.$$

2.4 Examples

Example 2.4.1. *The following simple example is from [BLO01], where it was proved using the Gauss-Lucas theorem that $p_*(z) = z^n$ is a global optimizer of the abscissa over the set of polynomials*

$$P = \{z^n + a_1 z^{n-1} + \ldots + a_{n-1} z + a_n \mid a_1 + a_2 = 0, a_i \in \mathbb{C}\}.$$

We calculate $h(z) = \binom{n}{2} z(z + \frac{2}{n-1})$. *Theorem 2.3.1 proves global optimality over* $a_i \in \mathbb{R}$ *and Theorem 2.3.8 proves global optimality over* $a_i \in \mathbb{C}$.

Example 2.4.2. *As mentioned in Section 1, Henrion and Overton [HO06] showed that the problem of finding a second-order linear controller that maximizes the closed-loop asymptotic decay rate for the classical two-mass-spring system is equivalent to an abscissa minimization problem for a monic polynomial of degree 6 whose coefficients depend affinely on 5 parameters, or equivalently with a single affine constraint on the coefficients. Theorem 2.3.1 (as well as Theorem 2.3.8) establishes global optimality of the locally optimal polynomial constructed in [HO06], namely,* $(z - \beta)^6$, *where* $\beta = -\sqrt{15}/5$.

Example 2.4.3. *This is derived from a "Belgian chocolate" stabilization challenge problem of Blondel [Blo94]: given* $a(z) = z^2 - 2\delta z + 1$ *and* $b(z) = z^2 - 1$, *find the range of real values of* δ *for which there exist polynomials* x *and* y *such that* $deg(x) \geq deg(y)$ *and* $\alpha(xy(ax + by)) < 0$. *This problem remains unsolved. However, inspired by numerical experiments, [BHLO06b] gave a solution for* $\delta < \bar{\delta} = (1/2)\sqrt{2 + \sqrt{2}} \approx 0.924$. *When* x *is constrained to be a monic polynomial with degree* 3 *and* y *to be a constant, the minimization of* $\alpha(xy(ax + by))$ *reduces to*

$$\inf_{p \in P} \alpha(p)$$

where

$$P = \{(z^2 - 2\delta z + 1)(z^3 + \sum_{k=0}^{2} w_k z^k) + (z^2 - 1)v \mid w_0, w_1, w_2, v \in \mathbb{C}\}.$$

For nonzero fixed δ, P *is a set of monic polynomials with degree 5 whose coefficients*

depend affinely on 4 parameters, or equivalently with a single affine constraint on the coefficients. In [BHLO06b] a polynomial in P with one distinct root of multiplicity 5 was constructed and proved to be locally optimal using nonsmooth analysis. Theorems 2.3.1 and 2.3.8 prove its global optimality. They also apply to the case when x is constrained to be monic with degree 4; then, as shown in [BHLO06b], stabilization is possible for $\delta < \tilde{\delta} = (1/4)\sqrt{10 + 2\sqrt{5}} \approx 0.951$.

Example 2.4.4. *The polynomial achieving the minimal root radius may not be unique. Let $P = \{z^2 + a_1 z + a_2 \mid 1 + a_1 + a_2 = 0, a_i \in \mathbb{R}\}$. We have*

$$\rho^* := \inf_{p \in P} \rho(p) = \inf_{a_2 \in \mathbb{R}} \rho(z^2 - (a_2 + 1)z + a_2) = \inf_{a_2 \in \mathbb{R}} \rho\left((z - a_2)(z - 1)\right) = 1.$$

The minimal value is attained on a continuum of polynomials of the form $(z - a_2)(z-1)$ for any $-1 \leq a_2 \leq 1$ and hence minimizers are not unique. The existence of the minimizers $(z-1)^2$ and $(z+1)(z-1)$ is consistent with Theorem 2.2.1. The same example shows that the minimizer for the radius optimization problem with complex coefficients may not be unique.

Example 2.4.5. *Likewise, a polynomial achieving the minimal root abscissa may not be unique. Let $P = \{z^2 + a_1 z + a_2 \mid a_1 = 0, a_2 \in \mathbb{R}\}$. We have*

$$\alpha^* = \inf_{p \in P} \alpha(p) = \inf_{a_2 \in \mathbb{R}} \alpha(z^2 + a_2) = 0.$$

Here $B_0 = B_2 = 0$, $B_1 = 1$. The optimum is attained at $p^*(z) = z^2$, where $-\alpha^* = 0$ is a root of the polynomial $h(z) = z$, as claimed in Theorem 2.3.1. However, the optimum is attained at a continuum of polynomials of the form $z^2 + a_2$ for any $a_2 > 0$.

Example 2.4.6. *In this example, the infimal root abscissa is not attained. Let* $P = \{z^2 + a_1 z + a_2 \mid a_1 \in \mathbb{R} \text{ and } a_2 = -1\}$. *We have* $h(z) = z^2 + 1$, *so* $-\alpha^* = 0$ *is a root of* $h^{(1)}$ *but not of* h. *Thus, Theorem 2.3.7 applies with* $\ell = 1$. *Indeed*

$$\begin{aligned} \alpha^* = \inf_{p \in P} \alpha(p) &= \inf_{a_1 \in \mathbb{R}} \alpha(z^2 + a_1 z - 1) \\ &= \inf_{a_1 \in \mathbb{R}} \max \left\{ \frac{-a_1 - \sqrt{a_1^2 + 4}}{2}, \frac{-a_1 + \sqrt{a_1^2 + 4}}{2} \right\} = 0. \end{aligned}$$

This infimum is not attained, but as $a_1 \to \infty$, *setting* $\epsilon = \frac{-a_1 + \sqrt{a_1^2+4}}{2} \to 0$ *and* $M_\epsilon = \frac{-a_1 - \sqrt{a_1^2+4}}{2} \to -\infty$ *gives* $(z - M_\epsilon)(z - \epsilon) \in P$ *as claimed in Theorem 2.3.7.*

Example 2.4.7. *Consider the family* $P = \{z^3 + a_1 z^2 + a_2 z + a_3 \mid a_1, a_2 \in \mathbb{R} \text{ and } a_3 = -1\}$. *We have* $h(z) = z^3 + 1$, *so* $-\alpha^* = 0$ *is a root of both* $h^{(1)}$ *and* $h^{(2)}$. *Thus, the assumptions of Theorem 2.3.7 are again satisfied with* $\ell = 1$. *However, this example shows the necessity of setting* $m = \ell + 1$ *when* $h^{(\ell)}$ *has a root of even multiplicity at* $-\alpha^*$. *Setting* $m = \ell = 1$ *is impossible since then* $(z - M_\epsilon)^m (z - \epsilon)^{n-m} \in P$ *implies* $M_\epsilon = \frac{1}{\epsilon^2} \to +\infty$ *as* $\epsilon \to 0$. *On the other hand, when* $m = \ell + 1 = 2$, *we have* $(z - M_\epsilon)^m (z - \epsilon)^{n-m} \in P$ *with* $M_\epsilon = -\frac{1}{\sqrt{\epsilon}} \to -\infty$ *as* $\epsilon \downarrow 0$.

Example 2.4.8. *This is a SIMO static output feedback example going back to 1975 [ABJ75]. Given a linear system* $\dot{x} = Fx + Gu$, $y = Hx$, *we wish to determine whether there exists a control law with* $u = Ky$ *stabilizing the system, i.e., so that the eigenvalues of* $F + GKH$ *are in the left half-plane. For this particular example, the gain matrix* $K \equiv [w_1, w_2] \in \mathbb{R}^{2\times 1}$, *and the problem is equivalent to finding a*

stable polynomial in the family

$$P = \{(z^3 - 13z) + (z^2 - 5z)w_1 + (z + 1)w_2 \mid w_1, w_2 \in \mathbb{R}\}.$$

A very lengthy derivation in [ABJ75] based on the decidability algorithms of Tarski and Seidenberg yields a stable polynomial $p \in P$ *with abscissa* $\alpha(p) \approx -0.0656$. *In 1979, Chen [Che79b, p.31], referring to [ABJ75], mentioned that his results show that the infimal value of the abscissa* α *over all polynomials in* P *is approximately* -5.91, *but he did not provide an optimal or nearly optimal solution. In 1999, the same example was used to illustrate a numerical method given in [PS99], which, after 20 iterations, yields a stable polynomial in* $p \in P$ *with abscissa* $\alpha(p) \approx -0.0100$. *The methods of [ABJ75] and [PS99] both generate stable polynomials, but their abscissa values are nowhere near Chen's infimal value. Applying Theorem 2.3.1, we find that the rightmost real root of* h *is* $-\beta \approx 5.91$ *and none of the derivatives of* h *have larger real roots, so* $(z - \beta)^3$ *is the global minimizer of the abscissa in the family* P. *Theorem 2.3.8 shows that allowing* K *to be complex does not reduce the optimal value.*

Example 2.4.9. *Consider the SISO system with the transfer function ([SMM92, Example 1], [GAB08])*

$$\frac{s^2 + 15s + 50}{s^4 + 5s^3 + 33s^2 + 79s + 50}.$$

We seek a second-order controller of the form

$$\frac{w_3 s^2 + w_4 s + w_5}{s^2 + w_1 s + w_2}$$

that stabilizes the resulting closed-loop transfer function

$$T(s) = (s^4+5s^3+33s^2+79s+50)(s^2+w_1s+w_2)+(s^2+15s+50)(w_3s^2+w_4s+w_5).$$

Applying the software package HIFOO *[BHLO06a] to locally optimize the abscissa of T results in a stabilizing controller with $\alpha(T) \approx -0.6640$. But since $T(s)$ is a monic polynomial with degree 6 depending affinely on 5 parameters, Theorems 2.3.1 and 2.3.8 apply, showing that the optimal closed-loop transfer function is $(z-\beta)^6$ where $\beta \approx -12.0801$.*

More examples may be explored by downloading a publicly available[1] MATLAB code implementing the constructive algorithms implicit in Theorems 2.2.1, 2.2.6, 2.3.1 and 2.3.8 as well as Corollary 2.2.2 and Theorem 2.3.7. A code generating all the examples of this section and two other examples mentioned in [BGMO10] is also available at the same website. In general, there does not seem to be any difficulty obtaining an accurate globally optimal value for the root abscissa or root radius in the real or complex case. However, even in the cases where an optimal solution exists, the coefficients may be large, so that rounding errors in the computed coefficients result in a large constraint residual, and the difficulty is compounded when the optimal abscissa value is not attained and a polynomial with an approximately optimal abscissa value is computed: hence, it is inadvisable to choose ϵ in Theorem 2.3.7 too small. Furthermore, the multiple roots of the optimal polynomials are not robust with respect to small perturbations in the coefficients. Optimizing a more robust objective such as the so-called complex stability "radius" (in the data-perturbation sense) of the polynomial may be of more practical use;

[1] www.cs.nyu.edu/overton/software/affpoly/

see [BHLO06b, Section II]. Since it is not known how to compute global optima for this problem, one might use local optimization with the starting point chosen by first globally optimizing the root abscissa or radius respectively.

2.5 Concluding remarks

Suppose there are κ constraints on the coefficients. In this case, we conjecture, based on numerical experiments, that there always exists an optimal polynomial with at most $\kappa - 1$ roots having modulus less than ρ^* or having real part less than α^* respectively. However, there does not seem to be a useful bound on the number of possible distinct roots. Thus, computing global optimizers appears to be difficult.

When there are κ constraints, we can obtain upper and lower bounds on the optimal value as follows. Lower bounds can be obtained by solving many problems with only *one* constraint, each of which is obtained from random linear combinations of the prescribed κ constraints. Upper bounds can be obtained by local optimization of the relevant objective ρ or α over an affine parametrization which is obtained from computing the null space of the given constraints. However, the gap between these bounds cannot be expected to be small.

The results do not extend to the more general case of an affine family of $n \times n$ matrices depending on $n - 1$ parameters. For example, consider the matrix family

$$A(\xi) = \begin{bmatrix} \xi & 1 \\ -1 & \xi \end{bmatrix}.$$

This matrix depends affinely on a single parameter ξ, but its characteristic poly-

nomial, a monic polynomial of degree 2, does not, so the results given here do not apply. The minimal spectral radius (maximum of the moduli of the eigenvalues) of $A(\xi)$ is attained by $\xi = 0$, for which the eigenvalues are $\pm i$. Nonetheless, experiments show that it is often the case that optimizing the spectral radius or spectral abscissa of a matrix depending affinely on parameters yields a matrix with multiple eigenvalues, or several multiple eigenvalues with the same radius or abscissa value; an interesting example is analyzed in [GO07].

Chapter 3

The distance to discrete instability and the numerical radius

3.1 Introduction

Consider a linear dynamical system of the form

$$x(t+1) = Ax(t), \quad x(0) = x_0 \in \mathbb{C}^n \tag{3.1}$$

where $A \in \mathbb{C}^{n\times n}$. Let $\rho(A)$ denote the spectral radius of A (largest of the modulus of the eigenvalues of A). It is well known that the linear system (3.1) is stable if and only if $\rho(A) < 1$. By stability of the system, we mean that $x(t) \to 0$ as $t \to \infty$ regardless of the initial value $x(0)$. A stable system will remain stable under sufficiently small perturbations to the entries of A. Assume that the perturbed

system matrix has the form $A+E$. The norm of the smallest perturbation necessary to make the system unstable is called the *discrete distance to instability* of A and is given by

$$d_{\mathrm{DI}}(A) = \inf\{||E|| : \rho(A+E) \geq 1\},$$

where above and throughout this chapter, we are interested in the case $||\cdot|| = ||\cdot||_2$. The quantity d_{DI} is important in robustness analysis of control systems (Hinrichsen & Pritchard, 2005). The following characterization of d_{DI} is well known in the literature:

$$d_{\mathrm{DI}}(A) = \min_{\theta \in [0,2\pi)} \sigma_{\min}(A - e^{i\theta} I). \tag{3.2}$$

where I is the $n \times n$ identity matrix and $\sigma_{\min}$ denotes the smallest singular value. Consider the following continuous-time counterpart of the discrete-time system (3.1)

$$\frac{dx(t)}{dt} = Ax(t), \quad x(0) = x_0 \in \mathbb{C}^n, \tag{3.3}$$

where $A \in \mathbb{C}^{n \times n}$ as before. The system (3.3) is stable if and only if $\alpha(A) < 0$ where $\alpha(A)$ is the abscissa of A (largest real part of the eigenvalues of A). *The continuous distance to instability* is defined similarly and is given by

$$d_{\mathrm{CI}}(A) = \inf\{||E|| : \alpha(A+E) \geq 0\}.$$

[Bye88] shows how the continuous distance to instability d_{CI} can be computed by solving Hamiltonian eigenvalue problems. More specifically, by [Bye88, Theorem 1], we have $d_{\mathrm{CI}}(A)$ equal to the smallest non-negative ρ such that the $2n \times 2n$

Hamiltonian matrix

$$H(\rho) = \begin{bmatrix} A & -\rho I \\ \rho I & -A^H \end{bmatrix}$$

has a pure imaginary eigenvalue, where the superscript ^{H} denotes the complex conjugate transpose. As a consequence, the quantity $d_{\mathrm{CI}}(A)$ is equal to the smallest positive value of ρ such that the two-parameter Hamiltonian matrix $H(\rho) - i\omega I$ is singular for some value of $\omega \in \mathbb{R}$. To solve this two-parameter Hamiltonian eigenvalue problem, Freitag & Spence [FS11] recently introduced a fast method that extends the implicit determinant method described in [SP05]. The implicit determinant method was originally developed to deal with one-parameter nonlinear eigenvalue problems and goes back to the work of [GR84].

For the discrete case, computing $d_{\mathrm{DI}}(A)$ involves solving generalized symplectic eigenvalue problems rather than standard Hamiltonian eigenvalue problems. Indeed, as follows from Lemma 3.2.1 at the start of Section 3.2, the discrete distance $d_{\mathrm{DI}}(A)$ is equal to the smallest positive value of ε such that two-parameter matrix family $P(\varepsilon) - e^{i\theta}Q(\varepsilon)$ is singular for some value of $\theta \in \mathbb{R}$ ($P(\varepsilon)$ and $Q(\varepsilon)$ are to be defined in Section 3.2). A natural idea to compute d_{DI} is to extend the implicit determinant method to deal with generalized eigenvalue problems; however this is not straightforward and requires a detailed analysis. In Section 3.2, we give the background on the implicit determinant method and explain how this extension can be done. Section 3 includes our new algorithm to compute the distance to instability, its convergence and complexity analysis and numerical experiments that demonstrate the effectiveness of our method by comparing it with the existing Boyd-Balakrishnan type algorithm described in [HS89, Men06]. We show that under generic assumptions, our algorithm is locally quadratically convergent.

The local quadratic convergence of the method in [FS11] was shown under an assumption. Here, we improve the results of [FS11] by proving that this assumption holds generically. We then turn our attention in Section 3.4 to another quantity of interest for the robust stability analysis, *the numerical radius* [MO05, HS89]. The numerical radius of a square matrix is a key quantity which has applications to the analysis of the convergence of iterative solution methods for linear systems [ALP94, Eie93], and to the stability of hyperbolic finite-difference schemes [GT82]. Furthermore, since it provides the upper bound

$$||A^k|| \leq 2r(A)^k$$

on the norm of the powers of A [GT82], the numerical radius bounds the asymptotic behavior of the dynamical system (3.1).

For the computation of the numerical radius, two recent methods are by [HW97] and [MO05]. The former method is based on finding a local maximum of an eigenvalue optimization problem whose global maximum is equal to the numerical radius. Hence, the method is not globally convergent. The latter method is the first globally convergent method known in the literature. The quadratic convergence of the method was mentioned in [MO05] under an assumption. Here we show that in fact this assumption always holds and develop a cubically convergent variant of this algorithm. The cubic method is inspired by the algorithm of [GDV98] which was originally developed for the H_∞ norm of a transfer matrix.

Notation: We use the subscripts "ε" and "θ" to denote partial derivatives with respect to ε and θ respectively. A *matrix pencil* is the set of all matrices of the form $G - \lambda H$, where $G, H \in \mathbb{C}^{n\times n}$ and $\lambda \in \mathbb{C}$. A number $\mu \in \mathbb{C}$ is a *generalized*

eigenvalue of the pencil $G-\lambda H$ if μ is a root of the polynomial $\psi(\lambda) = \det(G-\lambda H)$. The $2n \times 2n$ pencil $G - \lambda H$ is said to be *singular* if $\det(G - \lambda H) = 0$ for all $\lambda \in \mathbb{C}$; otherwise it is said to be *regular* in which case it has at most $2n$ finite eigenvalues. A non-zero vector x satisfying $(G-\lambda H)x = 0$ is called a "generalized eigenvector". From now on, we shall drop the adjective "generalized" except when its absence would be confusing.

3.2 Background and the method

We start with a result that characterizes the singular values of $A - e^{i\theta} I$ by the eigenvalues of a matrix pencil.

Lemma 3.2.1. *([Bye88, MO05]) $A - e^{i\theta} I$ has ε as one of its singular values if and only if the pencil $R_\varepsilon(\lambda) := P(\varepsilon) - \lambda Q(\varepsilon)$ has the eigenvalue $e^{i\theta}$ or it is singular where*

$$P(\varepsilon) = \begin{bmatrix} -\varepsilon I & A \\ I & 0 \end{bmatrix} \quad \textit{and} \quad Q(\varepsilon) = \begin{bmatrix} 0 & I \\ A^H & -\varepsilon I \end{bmatrix}.$$

Furthermore, the pencil $R_\varepsilon^H(\lambda) = P^H(\varepsilon) - \lambda Q^H(\varepsilon)$ is symplectic.

The matrix $M(\varepsilon, \theta) = D(\theta) R_\varepsilon(e^{i\theta})$ is Hermitian for all θ [MO05], where

$$D(\theta) = \begin{bmatrix} I & 0 \\ 0 & -e^{-i\theta} I \end{bmatrix} \quad \text{and} \quad M(\varepsilon, \theta) = \begin{bmatrix} -\varepsilon I & A - e^{i\theta} I \\ A^H - e^{-i\theta} I & -\varepsilon I \end{bmatrix}.$$

If R_ε has an eigenvalue $\lambda = e^{i\theta}$, for some $\theta \in \mathbb{R}$, and the corresponding eigenvector

$x \in \mathbb{C}^{2n}$ is partitioned as $x = \begin{bmatrix} u^T & v^T \end{bmatrix}^T$ with $u, v \in \mathbb{C}^n$, then

$$R_\varepsilon(e^{i\theta})x = 0 \iff M(\varepsilon,\theta)x = 0 \iff \begin{bmatrix} 0 & A - e^{i\theta}I \\ A^H - e^{-i\theta}I & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \varepsilon \begin{bmatrix} u \\ v \end{bmatrix} \tag{3.4}$$

$$\iff (A - e^{i\theta}I)v = \varepsilon u \quad \text{and} \quad (A - e^{i\theta}I)^H u = \varepsilon v. \tag{3.5}$$

Hence, ε is a singular value of $A - e^{i\theta}I$ with left and right singular vectors u and v. This is the idea behind the proof of Lemma 3.2.1. Define the real-valued function $g : [0, 2\pi) \to \mathbb{R}$

$$g(\theta) := \sigma_{\min}(A - e^{i\theta}I).$$

By simple continuity and periodicity arguments, it is easy to see that g attains its minimum and maximum value. From (3.2), we see that the minimum value is $d_{\mathrm{DI}}(A)$. Denote the maximum value by $\Gamma(A)$. From Lemma 3.2.1, we see that if $\varepsilon < d_{\mathrm{DI}}(A)$, then R_ε has no eigenvalue λ of unit modulus. In fact, when $\varepsilon = 0$, the set of eigenvalues of R_ε, denoted by $\Lambda(R_\varepsilon)$, consists of the eigenvalues of A and their mirror images with respect to the unit circle. As ε increases, the set $\Lambda(R_\varepsilon)$ approaches the unit circle from both sides and reach the unit circle when ε is exactly equal to the optimal value $\epsilon_* := d_{\mathrm{DI}}(A)$. At this point, at least one of the eigenvalues of R_{ϵ_*} will be of unit modulus. Let $e^{i\theta_*}$ be such an eigenvalue. The angle θ_* is a global minimizer of g (conversely, any global minimizer of g correspond to the angle of a unit modulus eigenvalue of R_{ε_*}). By the intermediate value theorem, for any $d_{\mathrm{DI}}(A) \le \varepsilon \le \Gamma(A)$, there exists a $\theta \in [0, 2\pi)$ such that

$\varepsilon = g(\theta)$ and for such ε by Lemma 3.2.1 we conclude that R_ε has a unit modulus eigenvalue.

It is known that $g(\theta)$ is C^{2k} ($2k$-times continuously differentiable) for some integer $k \geq 1$ around a local minimizer of g, admitting a Taylor series expansion around θ_* [GDV98]

$$g(\theta) = \epsilon_* + \beta(\theta - \theta_*)^{2k} + \mathcal{O}\Big((\theta - \theta_*)^{2k+1}\Big) \tag{3.6}$$

for some $\beta \geq 0$. We start with an assumption that is common in the literature [MO05, BLO03a, GO11]. This is a generic assumption that holds for almost all matrices in $\mathbb{C}^{n\times n}$ [BLO03a].

Assumption 3.2.1. *At a global minimizer θ_* of g, the smallest singular value of $A - e^{i\theta_*}I$ is simple.*

Assumption 3.2.1 implies that g is real analytic around a global minimizer. In addition, under Assumption 3.2.1, we have the following two mutually exclusive cases about the shape of the graph of g around θ_*.

(C1) $\underline{R_{\epsilon_*} \text{ is singular:}}$ It is well known that almost all perturbations to a square singular pencil makes it regular. In this sense, this is a degenerate case. From Lemma 3.2.1, it follows that ϵ_* is a singular value of $A - e^{i\theta}I$ for all θ. By continuity, for θ around θ_*, ϵ_* must be the smallest singular value. Hence, $g(\theta)$ is a constant function around θ_* and is equal to ϵ_*. The converse of this statement is true as we establish in Theorem 3.2.2: If $g(\theta)$ is a constant function around θ_*, then R_{ϵ_*} is singular.

(C2) $\underline{R_{\epsilon_*} \text{ is regular:}}$ The pencil R_{ϵ_*} has at most $2n$ eigenvalues. Hence, by Lemma

3.2.1, the minimum of g can only be attained at finitely many (at most $2n$) points. This implies that $\Gamma(A) > d_{\mathrm{DI}}(A)$ and hence g cannot be a constant function around the minimizer(s). The representation (3.6) is valid for some $k \geq 1$ and $\beta > 0$. For some small $\delta > 0$, since $\beta > 0$, $g(\theta)$ is a strictly decreasing function of θ on $[\theta_* - \delta, \theta_*]$ and a strictly increasing function on $[\theta_*, \theta_* + \delta]$. Hence, ε has inverse functions $\theta_1 : [\epsilon_*, g(\theta_* - \delta)] \to [\theta_* - \delta, \theta_*]$ and $\theta_2 : [\epsilon_*, g(\theta_* + \delta)] \to [\theta_*, \theta_* + \delta]$. These functions have Puiseux series expansions [Die57, p. 246], so a simple calculation shows

$$\theta_1(\varepsilon) = \theta_* - \beta^{-1/2k}(\varepsilon - \epsilon_*)^{1/2k} + \mathcal{O}(\varepsilon - \epsilon_*)^{1/k}, \tag{3.7}$$

$$\theta_2(\varepsilon) = \theta_* + \beta^{-1/2k}(\varepsilon - \epsilon_*)^{1/2k} + \mathcal{O}\left(\varepsilon - \epsilon_*\right)^{1/k}. \tag{3.8}$$

Note that $\theta_* = \theta_1(\epsilon_*) = \theta_2(\epsilon_*)$. Using Lemma 3.2.1, it follows that $e^{i\theta_1(\varepsilon)}$ and $e^{i\theta_2(\varepsilon)}$ are eigenvalues of R_ε when ε is in the interval $(\epsilon_*, \epsilon_* + \delta)$. As $\varepsilon \downarrow \epsilon_*$, $\theta_1(\varepsilon), \theta_2(\varepsilon) \to \theta_*$ showing us that the multiplicity of $e^{i\theta_*}$ as an eigenvalue is at least two and that $e^{i\theta_*}$ splits into two different eigenvalues $e^{i\theta_1(\varepsilon)}$ and $e^{i\theta_2(\varepsilon)}$ as ε exceeds ϵ_*.

Since the pencil $R_\varepsilon^H(\lambda)$ is symplectic, the set of its eigenvalues is symmetric with respect to the unit circle, and so is the set of eigenvalues of R_ε denoted by $\Lambda(R_\varepsilon)$. As $\varepsilon \uparrow \epsilon_*$, the set $\Lambda(R_\varepsilon)$ approaches the unit circle from both sides and touches the unit circle when $\varepsilon = \epsilon_*$. Since symmetric eigenvalue pairs have the same multiplicity [Wim91, Theorem 1.3], as long as R_{ϵ_*} is not singular, the multiplicity of $e^{i\theta_*}$ as an eigenvalue of R_{ϵ_*} must be an even integer. In addition, it turns out that the multiplicity of $e^{i\theta_*}$ as an eigenvalue determines the curvature of g around θ_* as the following theorem shows.

Theorem 3.2.2. *Let Assumption 3.2.1 be satisfied and θ_* be a global minimizer of g. We have the following.*

i) The pencil R_{ε_*} is singular if and only if $g(\theta) = \epsilon_*$ for θ in a neighborhood of θ_*.

ii) If the pencil R_{ε_*} is regular then the multiplicity of $e^{i\theta_*}$ as an eigenvalue of R_{ε_*} is an even positive integer. In this case, the multiplicity is equal to m if and only if

$$g(\theta) = \epsilon_* + c_m(\theta - \theta_*)^m + \mathcal{O}(|\theta - \theta_*|^{m+1})$$

for θ around θ_* and for some $c_m > 0$.

Proof. A direct computation yields

$$\begin{aligned}
\det R_\varepsilon(e^{i\theta}) &= \det(D(\theta)^{-1}M(\varepsilon,\theta)) && (3.9)\\
&= \det(D(\theta))^{-1}\det(M(\varepsilon,\theta)) && (3.10)\\
&= (-e^{i\theta})^n \det\begin{bmatrix} -\varepsilon I & A - e^{i\theta}I \\ (A - e^{i\theta}I)^H & -\varepsilon I \end{bmatrix} && (3.11)\\
&= e^{in\theta}\prod_{j=1}^{n}[\sigma_j(A - e^{i\theta}I) - \varepsilon][\sigma_j(A - e^{i\theta}I) + \varepsilon]. && (3.12)
\end{aligned}$$

where $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n = \sigma_n$ denotes the singular values listed by multiplicity[1].

Since ϵ_* is the minimum of $g(\theta)$, we have $\sigma_j(A - e^{i\theta}I) \geq \varepsilon$ for all θ and $j = 1, 2, \ldots, n$. By Assumption 3.2.1 and the continuity of the singular values, the smallest singular value of $A - e^{i\theta}I$ is simple for θ close enough to θ_*. Thus, for θ

[1]Note that equation (3.12) leads to a proof of the first statement in Lemma 3.2.1.

in a neighborhood of θ_*, we have

$$\sigma_1(A - e^{i\theta}I) \geq \ldots \geq \sigma_{n-1}(A - e^{i\theta}I) \geq \min_{\tilde{\theta}\in[0,2\pi)} \sigma_{n-1}(A - e^{i\tilde{\theta}}I) > \sigma_n(A - e^{i\theta}I) \geq \epsilon_*.$$

Hence for (ε, θ) close enough to (ϵ_*, θ_*), we have

$$\det R_\varepsilon(e^{i\theta}) = 0 \iff \sigma_n(A - e^{i\theta}I) = g(\theta) = \varepsilon. \tag{3.13}$$

The pencil R_{ε_*} is singular if and only if $\det R_{\epsilon_*}(e^{i\theta}) = 0$ for all θ in a neighborhood of θ_*. Then plugging $\varepsilon = \epsilon_*$ into (3.13) proves i). For part ii), we start by noting that if $e^{i\theta_*}$ is a generalized eigenvalue of $P(\epsilon_*) - \lambda Q(\epsilon_*)$ with multiplicity m, then by definition we have the first order expansion

$$\det R_{\epsilon_*}(e^{i\theta}) = \hat{c}(e^{i\theta} - e^{i\theta_*})^m + \mathcal{O}(|e^{i\theta} - e^{i\theta_*}|^2) \tag{3.14}$$

$$= \tilde{c}(\theta - \theta_*)^m + \mathcal{O}(|\theta - \theta_*|^2) \tag{3.15}$$

for some non-zero $\hat{c}, \tilde{c} \in \mathbb{C}$ which holds if and only if

$$g(\theta) - \epsilon_* = \sigma_n(A - e^{i\theta}I) - \epsilon_* \approx c_m(\theta - \theta_*)^m$$

for some non-zero complex constant c_m, where in the last step we used the fact from (3.12) that $\sigma_j(A - e^{i\theta}I) - \epsilon_*$ is nonzero for $j < n$ and for θ around θ_*. The fact that m is an even integer and $c_m > 0$ follows from the fact that θ_* is a global minimizer of $g(\theta)$. This completes the proof of *ii*). □

From Assumption 3.2.1, (3.4) and (3.5), it follows that

$$\dim\ker(R_{\epsilon_*}(e^{i\theta_*})) = 1 \quad \text{and} \quad \dim\ker(M(\epsilon_*, \theta_*)) = 1. \tag{3.16}$$

Furthermore, null vectors of $R_{\epsilon_*}(e^{i\theta_*})$ and $M(\epsilon_*, \theta_*)$ are related. Let $x_* = \begin{bmatrix} u_*^T & v_*^T \end{bmatrix}^T$ be a right null vector of $M(\epsilon_*, \theta_*)$. Clearly, x_* is a right null vector of $R_{\epsilon_*}(e^{i\theta_*}) = D(\theta_*)^H M(\epsilon_*, \theta_*)$ as well. Since $M(\epsilon_*, \theta_*)$ is Hermitian, x_*^H is a left null vector of $M(\epsilon_*, \theta_*)$. It is easy to verify that $y_* = D(\theta_*)^H x_*$ is a left null vector of $R(\epsilon_*, e^{i\theta_*})$.

To extend the implicit determinant method [SP05] for computing the smallest ε such that the pencil R_ε has a unit norm eigenvalue, we start with a key lemma on which our method is based.

Lemma 3.2.3. *Let Assumption 3.2.1 be satisfied and assume*

$$c^H x_* \neq 0 \tag{3.17}$$

for some column vector $c \in \mathbb{C}^{n\times 1}$. Then the $(2n+1) \times (2n+1)$ complex matrix

$$K(\varepsilon, \theta) = \begin{bmatrix} R_\varepsilon(e^{i\theta}) & D(\theta)^H c \\ c^H & 0 \end{bmatrix} \tag{3.18}$$

is nonsingular at $\varepsilon = \epsilon_*$, $\theta = \theta_*$.

Proof. Equation (3.16) and Lemma 2.8 of [Kel77] prove that $K(\epsilon_*, \theta_*)$ is nonsingular if $c^H x_* \neq 0$ and $y_*^H D(\theta_*)^H c \neq 0$. The first inequality is satisfied by assumption and the second inequality reduces to the first since we have $y_* = D(\theta_*)^H x_*$ and $D(\theta_*)D(\theta_*)^H = I$. □

Since $K(\epsilon_*, \theta_*)$ is nonsingular, so is $K(\varepsilon, \theta)$ for (ε, θ) around (ϵ_*, θ_*). Consider

the following linear system where $c \in \mathbb{R}^n$ satisfies the condition (3.17) and (ε, θ) is around (ϵ_*, θ_*):

$$\begin{bmatrix} R_\varepsilon(e^{i\theta}) & D(\theta)^H c \\ c^H & 0 \end{bmatrix} \begin{bmatrix} x(\varepsilon,\theta) \\ f(\varepsilon,\theta) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{3.19}$$

It follows easily from Cramer's Rule that

$$f(\varepsilon,\theta) = \frac{\det R_\varepsilon(e^{i\theta})}{\det K(\varepsilon,\theta)}. \tag{3.20}$$

Since $\det K(\varepsilon,\theta) \neq 0$ around (ϵ_*, θ_*), by Lemma 3.2.3, we have

$$f(\varepsilon,\theta) = 0 \iff \det R_\varepsilon(e^{i\theta}) = 0. \tag{3.21}$$

Also, from (3.19) we have

$$f(\varepsilon,\theta) = 0 \iff x(\varepsilon,\theta) \in \ker R_\varepsilon(e^{i\theta}) \text{ and } c^H x(\varepsilon,\theta) = 1. \tag{3.22}$$

Equation (3.21) is a characterization of the zero set of the trigonometric polynomial $\det R_\varepsilon(e^{i\theta})$ as the zero set of the function $f(\varepsilon,\theta)$ near (ϵ_*, θ_*), an equivalence that we will exploit in our method. The main idea of our method is to seek solutions of

$$f(\varepsilon,\theta) = 0 \tag{3.23}$$

and hence recover values of ε and θ such that R_ε has the eigenvalue $e^{i\theta}$ and then find the corresponding eigenvector. In particular, we are looking for the smallest such ε which is equal to $d_{\mathrm{DI}}(A)$.

To see that $f(\varepsilon,\theta)$ is real, multiply the first row of (3.19) from left by the row

vector $x(\varepsilon,\theta)^H D(\theta)$ to get

$$f(\varepsilon,\theta) = -x(\varepsilon,\theta)^H M(\varepsilon,\theta) x \tag{3.24}$$

where we have again used the fact that $D(\theta)D(\theta)^H = I$ and $c^H x(\varepsilon,\theta) = 1$. Since M is Hermitian, we verify from (3.24) that f is real-valued.

By multiplying (3.19) from left with the invertible matrix $\begin{bmatrix} D(\theta) & 0 \\ 0 & I \end{bmatrix}$, we see that the system (3.19) is equivalent to the Hermitian system

$$\begin{bmatrix} M(\varepsilon,\theta) & c \\ c^H & 0 \end{bmatrix} \begin{bmatrix} x(\varepsilon,\theta) \\ f(\varepsilon,\theta) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{3.25}$$

The linear systems (3.25) and (3.19) are equivalent. For numerical computations, we prefer to work with the Hermitian system (3.25) which has more structure.

3.2.1 The $f(\varepsilon,\theta) = 0$ curve around (ϵ_*, θ_*)

The characterization (3.21) says that analyzing the path $\det(R_\varepsilon(e^{i\theta})) = 0$ in the (ε,θ) plane is equivalent to analyzing the path $f(\varepsilon,\theta) = 0$. This amounts to looking for the roots of $f(\varepsilon,\theta)$ in the (ε,θ) plane. We will use Newton's method but local quadratic convergence of Newton's method is guaranteed only when the Jacobian of $f(\varepsilon,\theta)$ is nonsingular at the optimizer. Our next aim is to compute this Jacobian and to characterize when it is singular.

Lemma 3.2.4. *Let Assumption 3.2.1 be satisfied and assume* (3.17) *holds. Con-*

sider the real curve $f(\varepsilon,\theta)=0$. Then, for (ε,θ) near (ϵ_*,θ_*) we have

$$f_\varepsilon(\varepsilon,\theta)=||x(\varepsilon,\theta)||^2>0. \tag{3.26}$$

Proof. Differentiating the linear system (3.25) with respect to ε, we obtain

$$\begin{bmatrix} M(\varepsilon,\theta) & c \\ c^H & 0 \end{bmatrix} \begin{bmatrix} x_\varepsilon(\varepsilon,\theta) \\ f_\varepsilon(\varepsilon,\theta) \end{bmatrix} = \begin{bmatrix} x(\varepsilon,\theta) \\ 0 \end{bmatrix} \tag{3.27}$$

where we used the fact that $M_\varepsilon(\varepsilon,\theta)=-I$ and $x^H(\varepsilon,\theta)c=1$. Notice from (3.25) that when $f(\varepsilon,\theta)=0$, $x(\varepsilon,\theta)^H$ is a left null vector of $M(\varepsilon,\theta)$. Thus multiplying the first row from the left by $x(\varepsilon,\theta)^H$ gives

$$f_\varepsilon(\varepsilon,\theta)=x(\varepsilon,\theta)^H x(\varepsilon,\theta)=||x(\varepsilon,\theta)||^2>0 \tag{3.28}$$

where we used the fact that $c^H x(\varepsilon,\theta)=1$. □

We now turn our attention to the (first and higher order) partial derivatives of f with respect to θ at (ϵ_*,θ_*) .

Theorem 3.2.5. *Let Assumption 3.2.1 be satisfied. Then the following statements are true.*

i) In the (ε,θ) plane, near (ϵ_*,θ_*), the $f(\varepsilon,\theta)=0$ curve is identical with the curve $\varepsilon=g(\theta)$.

ii) The pencil R_{ε_*} is regular and $e^{i\theta_*}$ is an eigenvalue of R_{ε_*} with multiplicity m

if and only if

$$f^*_{\theta,j} := \frac{d^j f}{d\theta^j}(\epsilon_*, \theta_*) = 0 \text{ for } 0 \le j < m \quad \text{and} \quad f^*_{\theta,m} = \frac{d^m f}{d\theta^m}(\epsilon_*, \theta_*) < 0$$

with the convention that $f^*_{\theta,0} := f(\epsilon_*, \theta_*)$.

iii) The pencil R_{ε_*} *is singular if and only if*

$$f^*_{\theta,j} = 0 \quad \text{for} \quad j = 0, 1, 2, \ldots .$$

Proof. The part i) is a direct consequence of (3.21) and (3.13). For the part ii), observe that by (3.21) we have $f^*_{\theta,0} = 0$. Let ℓ be the smallest positive integer such that the ℓ-th derivative is non-zero, i.e.,

$$f^*_{\theta,j} = 0 \text{ for } 0 \le j < \ell \quad \text{and} \quad f^*_{\theta,\ell} \neq 0.$$

Plugging $\varepsilon = \epsilon_*$ into (3.20), we have

$$\det R_{\epsilon_*}(e^{i\theta}) = f(\epsilon_*, \theta) \det K(\epsilon_*, \theta)$$

for θ in a neighborhood of θ_*. Taking the j-th derivatives of both sides with respect to θ, and evaluating it at $\theta = \theta_*$ we get

$$\frac{d^j \Big(\det R_{\epsilon_*}(e^{i\theta}) \Big)}{d\theta^j}\Big|_{\theta=\theta_*} = \begin{cases} 0 & \text{for } 0 \le j < \ell \\ f^*_{\theta,\ell} \det K(\epsilon_*, \theta_*) & \text{for } j = \ell \end{cases} \tag{3.29}$$

In addition, from Lemma 3.2.3, we know that $\det K(\epsilon_*, \theta_*) \neq 0$ which implies

$$\frac{d^\ell \Big(\det R_{\epsilon_*}(e^{i\theta}) \Big)}{d\theta^\ell}\Big|_{\theta=\theta_*} \neq 0. \tag{3.30}$$

From (3.15), (3.29) and (3.30) we get $\ell = m$. It remains to prove that $f^*_{\theta,m} < 0$. Consider the equation $f(\varepsilon, \theta) = 0$. Since $f^*_\varepsilon \neq 0$ by Lemma 3.2.4, we can write ε as a function of θ around θ_* using the implicit function theorem. A simple calculus computation leads to

$$\varepsilon(\theta) = \epsilon_* - \frac{f^*_{\theta,m}}{m!}(\theta - \theta_*)^m + \mathcal{O}(|(\theta - \theta_*)|^{m+1}).$$

On the other hand, from part *i*), we must have $\varepsilon(\theta) = g(\theta)$. From Theorem 3.2.2, we obtain

$$c_m = -\frac{f^*_{\theta,m}}{m!}$$

with $c_m > 0$. We conclude that $f^*_{\theta,m} < 0$ and this proves part *ii*). The proof of part *iii*) is very similar and is omitted.

□

Corollary 3.2.6. *Under Assumption 3.2.1, we have*

$$f^*_\theta := f^*_{\theta,1} = \frac{df}{d\theta}(\epsilon_*, \theta_*) = 0.$$

Remark 3.2.7. *Corollary 3.2.6 can also be obtained by a direct computation as*

follows. Differentiating the linear system (3.25) *with respect to* θ*, we obtain*

$$\begin{bmatrix} M(\varepsilon,\theta) & c \\ c^H & 0 \end{bmatrix} \begin{bmatrix} x_\theta \\ f_\theta \end{bmatrix} = \begin{bmatrix} -M_\theta(\theta,\varepsilon)x(\varepsilon,\theta) \\ 0 \end{bmatrix} \tag{3.31}$$

where

$$M_\theta(\theta,\varepsilon) = \begin{bmatrix} 0 & -ie^{i\theta}I \\ ie^{-i\theta}I & 0 \end{bmatrix}. \tag{3.32}$$

Multiplying the first equality in (3.31) *by* x_*^H*, the left null vector of* $M(\epsilon_*,\theta_*)$*, and evaluating it at* (ϵ_*,θ_*)*, we obtain*

$$f_\theta^* = -x_*^H M_\theta(\epsilon_*,\theta_*)x_* = -2\ Im(e^{i\theta_*}u_*^H v_*). \tag{3.33}$$

Note that we have u_* *and* v_* *as the left and right singular vectors of* $A - e^{i\theta_*}I$ *corresponding to the smallest singular value* $g(\theta_*)$ (see (3.5))*. From the Taylor expansion* (3.6)*, we get*

$$g_\theta(\theta_*) = Im(e^{i\theta_*}u_*^H v_*) \tag{3.34}$$

Combining (3.33) *and* (3.34) *gives* $f_\theta^* = 0$*.*

Figure 3.1 illustrates Theorem 3.2.5. When $m = 2$, the $f(\varepsilon,\theta)$ curve around (ϵ_*,θ_*) looks like the graph of a quadratic polynomial in θ (on the left). In the middle, the case when $m > 2$ is illustrated. In this case, the $f(\varepsilon,\theta)$ curve looks like the graph of an m-th degree polynomial. Clearly, the graph moves closer to the vertical line $\varepsilon = \epsilon_*$ as m increases. Note that only even values of m are admitted by Theorem 3.2.2. When R_{ε_*} is singular, $f(\varepsilon,\theta)$ becomes a vertical line. In this sense, a singular pencil is behaving like the case $m = \infty$. In any case, we have

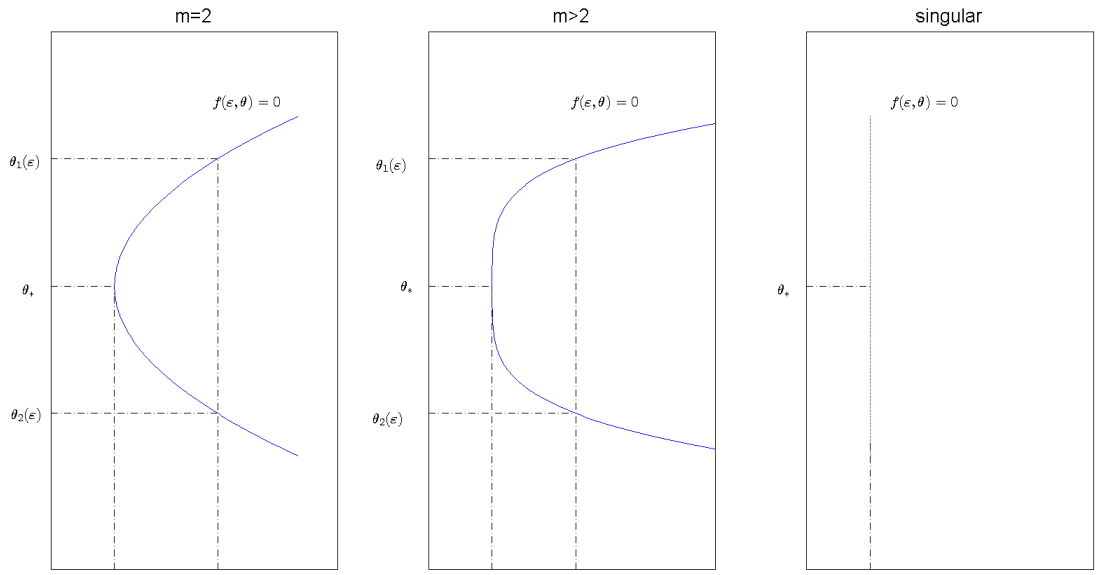

Figure 3.1: $f(\varepsilon, \theta) = 0$ curve in the (ε, θ) plane. The cases $m = 2$ (on the left), $m > 2$ (in the middle) and when R_{ε_*} is singular (on the right).

$f(\epsilon_*, \theta_*) = f_\theta(\epsilon_*, \theta_*) = 0$. A natural idea to compute (ϵ_*, θ_*) is to use Newton's method to find a root of the nonlinear equations

$$F(\varepsilon, \theta) = \begin{bmatrix} f(\varepsilon, \theta) \\ f_\theta(\varepsilon, \theta) \end{bmatrix} \tag{3.35}$$

which we will discuss in the next section. Newton's method is quadratically convergent as long as the Jacobian of F at the optimizer is nonsingular. Indeed, we will show in Section 3.3.1 that this is the case for almost all A.

3.3 Computing $d_{\mathrm{DI}}(A)$

We will use Newton's method to compute (ϵ_*, θ_*) which is a root of $F(\varepsilon, \theta)$. Newton's method with a starting guess $(\theta^{(0)}, \varepsilon^{(0)})$ requires solving a sequence of

linear systems

$$J(\varepsilon^{(i)}, \theta^{(i)}) \begin{bmatrix} \Delta\varepsilon^{(i)} \\ \Delta\theta^{(i)} \end{bmatrix} = F(\varepsilon^{(i)}, \theta^{(i)}) \tag{3.36}$$

and setting

$$\begin{bmatrix} \varepsilon^{(i+1)} \\ \theta^{(i+1)} \end{bmatrix} = \begin{bmatrix} \varepsilon^{(i)} \\ \theta^{(i)} \end{bmatrix} - \begin{bmatrix} \Delta\varepsilon^{(i)} \\ \Delta\theta^{(i)} \end{bmatrix} \tag{3.37}$$

for $i = 0, 1, 2, \ldots$ until convergence, where the Jacobian is

$$J(\varepsilon^{(i)}, \theta^{(i)}) = \begin{bmatrix} f_\varepsilon(\varepsilon^{(i)}, \theta^{(i)}) & f_\theta(\varepsilon^{(i)}, \theta^{(i)}) \\ f_{\theta\varepsilon}(\varepsilon^{(i)}, \theta^{(i)}) & f_{\theta\theta}(\varepsilon^{(i)}, \theta^{(i)}) \end{bmatrix}. \tag{3.38}$$

The values of $f_\varepsilon(\varepsilon^{(i)}, \theta^{(i)})$, $f_\theta(\varepsilon^{(i)}, \theta^{(i)})$ are calculated by solving the systems (3.27) and (3.31) by LU decomposition. Similarly, the values of $f_{\theta\varepsilon}(\varepsilon^{(i)}, \theta^{(i)})$ can be calculated by solving the system

$$\begin{bmatrix} M(\varepsilon, \theta) & c \\ c^H & 0 \end{bmatrix} \begin{bmatrix} x_{\theta\varepsilon}(\varepsilon, \theta) \\ f_{\theta\varepsilon}(\varepsilon, \theta) \end{bmatrix} = \begin{bmatrix} x_\theta(\varepsilon, \theta) - M_\theta(\varepsilon, \theta) x_\varepsilon(\varepsilon, \theta) \\ 0 \end{bmatrix} \tag{3.39}$$

obtained by differentiating (3.31) with respect to ε. Finally, $f_{\theta\theta}(\varepsilon^{(i)}, \theta^{(i)})$ can be calculated by solving the system

$$\begin{bmatrix} M(\varepsilon, \theta) & c \\ c^H & 0 \end{bmatrix} \begin{bmatrix} x_{\theta\theta}(\varepsilon, \theta) \\ f_{\theta\theta}(\varepsilon, \theta) \end{bmatrix} = \begin{bmatrix} -M_{\theta\theta}(\varepsilon, \theta) x(\varepsilon, \theta) - 2M_\theta(\varepsilon, \theta) x_\theta(\varepsilon, \theta) \\ 0 \end{bmatrix} \tag{3.40}$$

obtained by differentiating (3.31) with respect to θ where

$$M_{\theta\theta}(\varepsilon,\theta)=\begin{bmatrix}0 & e^{i\theta}I\\ e^{-i\theta}I & 0\end{bmatrix}. \tag{3.41}$$

Algorithm DDI (Newton's method) Given $(\varepsilon^{(0)},\theta^{(0)})$ and $c\in\mathbb{C}^n$ satisfying (3.17), set $i=0$.

1. Solve (3.25) and (3.31) (with $\varepsilon=\varepsilon^{(i)}$, $\theta=\theta^{(i)}$, using $x(\varepsilon,\theta)$ obtained in (3.25)) for the right side of (3.31)) in order to compute

$$F(\varepsilon^{(i)},\theta^{(i)})=\begin{bmatrix}f(\varepsilon^{(i)},\theta^{(i)})\\ f_\theta(\varepsilon^{(i)},\theta^{(i)})\end{bmatrix}, x(\varepsilon^{(i)},\theta^{(i)}) \text{ and } x_\theta(\varepsilon^{(i)},\theta^{(i)}).$$

2. Solve (3.27), (3.40) and (3.39) respectively in order to find the Jacobian $J(\varepsilon^{(i)},\theta^{(i)})$ given by (3.38) (using $x(\varepsilon^{(i)},\theta^{(i)})$ and $x_\theta(\varepsilon^{(i)},\theta^{(i)})$ computed in Step 1).

3. Newton update: Solve (3.37) in order to get $(\varepsilon^{(i+1)},\theta^{(i+1)})$.

4. Increment i, repeat Steps 1-3 until convergence.

We analyze next the convergence of this algorithm.

3.3.1 Convergence analysis

By Corollary 3.2.6, we have $f^*_\theta=0$. So the Jacobian at (ϵ_*,θ_*) is

$$J_*=\begin{bmatrix}f^*_\varepsilon & 0\\ f^*_{\theta\varepsilon} & f^*_{\theta\theta}\end{bmatrix}$$

where $f^*_{\theta\theta} = f^*_{\theta,2}$ is the second derivative with respect to θ at the optimizer. Newton's method (Algorithm DDI) is locally quadratically convergent if J_* is nonsingular. From Theorem 3.2.5 and Lemma 3.2.4, we see that J_* is nonsingular if and only if $f^*_{\theta\theta} \neq 0$. When J_* is singular, it is known that Newton's method generally exhibits linear convergence in practice [DKK83]. Depending on the order of singularity, there are some linear convergence results available. Let m be the multiplicity of $e^{i\theta_*}$ as an eigenvalue of R_{ε_*} (with the convention that $m = \infty$ if R_{ε_*} is singular). As a consequence of Theorem 3.2.2, m is even if it is finite. Applying Theorem 3.2.5, we see that the following mutually exclusive three cases arise.

(D1) $\underline{m = 2}$: The Jacobian $J(\epsilon_*, \theta_*)$ is nonsingular and Algorithm DDI has local quadratic convergence.

(D2) $\underline{m \geq 4}$: We have $f^*_{\theta,m} \neq 0$, J_* is singular and $F(\varepsilon, \theta)$ has a singularity of order $m - 2$ at (ϵ_*, θ_*) [Gri80]. It is known that in a starlike domain (that can be made explicit) around (ϵ_*, θ_*), Newton's method will converge linearly with rate $\frac{m-2}{m-1}$ [Gri80].

(D3) $\underline{m = \infty}$: We have $f^*_{\theta,m} \neq 0$ for all $m \geq 0$. $F(\varepsilon, \theta)$ does not have a singularity of finite order and J_* is singular. Linear convergence results mentioned above do not apply.

First, let's consider the case (D3). We start with the observation that almost all perturbations to a singular pencil makes it regular and a square matrix pencil is generically regular, i.e. for almost all ε and A, the pencil R_ε is not singular. However, this does not help because we are interested in a particular value of ε, namely $\varepsilon = d_{\mathrm{DI}}(A)$ for A fixed. To show that case (D3) is degenerate, we will

need to count the number of constraints on A and ε that are needed to make R_ε singular. Note that R_ε is singular if and only if the polynomial (in λ) of degree $2n$

$$p_{A,\varepsilon}(\lambda) = \det\Big(P(\varepsilon) - \lambda Q(\varepsilon)\Big)$$

is the zero polynomial. The polynomial $p_{A,\varepsilon}(\lambda)$ has $2n+1$ complex coefficients, each of which are polynomial functions of $2n^2+1$ real variables corresponding to ε and n^2 complex entries of A. The vanishing coefficients of $p_{A,\varepsilon}(\lambda)$ amount to $4n+2$ real-valued polynomial equations (or $2n+1$ complex-valued polynomial equations). Finally, there is one more constraint on ε, which must equal the stability radius of A. In total, we have $2n^2+1$ real variables and $4n+3$ equality constraints. Thus, for $n > 2$, R_{ε_*} is generically regular. Hence, for almost all matrices A, (D3) does not hold.

Let us show that (D1) holds for almost all matrices. It suffices to show that case (D2) does not hold generically. The complex space of $2n \times 2n$ matrices M_{2n} has real dimension $8n^2$. The manifold of $2n \times 2n$ matrices with only one Jordan block of size r and with $2n-r+1$ different eigenvalues, which we denote by M_r^{2n}, has real dimension of $8n^2 - 2r + 2$ in M_{2n} [Kel08]. As a consequence, for $A \in \mathbb{C}^{n\times n}$ fixed, the manifold

$$\mathcal{S}_A = \{P(\varepsilon) - \lambda Q(\varepsilon) : \varepsilon \in \mathbb{R}, \lambda \in \mathbb{C}\}$$

of real dimension 3 will generically not intersect M_r^{2n} for $r \geq 3$. This shows that for almost all matrices A, the pencil $R_\varepsilon(\lambda) = P(\varepsilon) - \lambda Q(\varepsilon)$ does not have a Jordan block of size ≥ 3 for any value of ε. Thus, (D2) is a degenerate case and (D1) is the generic case. These observations lead to the following theorem.

Theorem 3.3.1. *Let $n > 2$ be given. For a generic matrix $A \in \mathbb{C}^{n\times n}$, Algorithm DDI is locally quadratically convergent.*

3.3.2 Numerical experiments

Other existing methods to compute the discrete distance d_{DI} are the bisection algorithm of [HS89] and the Boyd-Balakrishnan (BB) algorithm [Men06, Algorithm 2], [HS89] (BB algorithm is originally developed to compute H_∞ norm of a transfer function in [BB90b] and it is adapted to compute d_{DI} in [HS89]). The bisection algorithm has linear convergence, whereas the BB method is quadratically convergent. Hence, we compare our method only with the BB algorithm.

Algorithm BB Given A, set $\theta \leftarrow 0$, $\varepsilon \leftarrow \sigma_n(A - e^{i\theta}I)$.

1. Compute the unit norm eigenvalues $e^{i\theta_1}$, $e^{i\theta_2}$, …, $e^{i\theta_j}$ of R_ε ordered so that $0 \le \theta_1 \le \theta_2 \le \cdots \le \theta_j < 2\pi$.

2. Find the midpoints, setting $\beta_\ell = \frac{1}{2}(\theta_\ell + \theta_{\ell+1})$ for $\ell < j$ and $\beta_j = \frac{1}{2}(\theta_j + \theta_1 + 2\pi) \mod 2\pi$.

3. Update $\varepsilon = \min_{1\le\ell\le j} \sigma_n(A - e^{i\beta_\ell})$.

4. Repeat until convergence.

In this section, we compare both algorithms in terms of computational cost. Note that Algorithm BB is globally convergent, whereas Algorithm DDI is, although quick, not guaranteed to find a globally optimal solution (although it does in all our numerical tests). In order to compare these algorithms in a fair way, inspired by the checking step introduced by Watson & He for the continuous distance to instability [HW98], we augment Algorithm DDI to a global method by

adding a checking step that checks global optimality, and in case global optimum is not certified, we restart it from a better initial guess. To be more precise, let $\bar{\varepsilon}$ be the distance computed by Algorithm DDI. We first compute the eigenvalues of the $R_{\bar{\varepsilon}-\delta}$ (by a symplectic eigenvalue decomposition (see Lemma 3.2.1)) where δ is a small tolerance. If $R_{\bar{\varepsilon}-\delta}$ does not have a unit modulus eigenvalue, then global convergence is achieved, we stop. Else, we reduce the value of $\bar{\varepsilon}$ as

$$\bar{\varepsilon} = \zeta\bar{\varepsilon}, \quad \zeta \in (0,1)$$

until $\bar{\varepsilon}$ is small enough that $R_{\bar{\varepsilon}-\delta}$ does not have a unit modulus eigenvalue. Then, restart Algorithm DDI by setting the initial values $\varepsilon^{(0)}$ to $\bar{\varepsilon}$, $\theta^{(0)}$ to the angle of an outermost eigenvalue (an eigenvalue whose modulus is larger than or equal to the modulus of the others) of $R_{\bar{\varepsilon}}$ and c to a unit norm eigenvector corresponding to this eigenvalue. Note that such a choice of c would be our best guess (up to a complex constant) for x_*.

In the simulations, we insert the following termination condition to the Step 4 of the Algorithm DDI: The iteration stops if $i > 0$ and

$$|\varepsilon^{(i+1)} - \varepsilon^{(i+1)}| \leq \texttt{tol} \quad \text{and} \quad |F(\varepsilon^{(i)}, \theta^{(i)})| \leq \texttt{tol}$$

for some small tolerance `tol`. We take $\texttt{tol} = 10^{-12}$ and $\delta = 10^{-6}$. We initialize Algorithm DDI by setting $\varepsilon^{(0)}$ to 0, $\theta^{(0)}$ to the angle of an outermost eigenvalue of A and c to a unit norm eigenvector corresponding to this eigenvalue.

We used the publicly available implementation of the BB algorithm[2]. Example 3.3.2 is taken from [HS89, Example 5.2] whereas the other examples are taken

[2]http://www.cs.nyu.edu/mengi/robuststability.html

from the EigTool package [Wri02b]. The experiments are made on a PC with an i7-620M processor and 4GB of memory using Matlab Version 7.13.0.564.

The cost of the BB algorithm in each iteration consists of a $2n \times 2n$ symplectic eigenvalue decomposition in Step 1 and $n \times n$ SVD decompositions in Step 4, whereas for the DDI algorithm, the main cost is, ignoring the triangular system solves, a couple of $(2n+1) \times (2n+1)$ LU decompositions in Step 1-3 and $2n \times 2n$ Hamiltonian eigenvalue decomposition(s) in the checking step explained above. For dense matrices, the cost of the $(2n+1) \times (2n+1)$ LU decomposition and the $n \times n$ SVD decomposition are $\frac{2}{3}(2n+1)^3 \approx \frac{16}{3}n^3$ and $\frac{20}{3}n^3$ respectively [GVL96]. The cost of a $2n \times 2n$ symplectic (generalized) eigenvalue decomposition is $\frac{10}{3}(2n)^3 = \frac{80}{3}n^3$ [GVL96, p. 100] assuming the QR algorithm is used. In Tables 3.2, 3.4, 3.6, 3.8 and 3.10 the columns headed as "#LU", "#SVD" and "#SYMP" demonstate the number of LU decompositions, SVD decompositions and symplectic eigenvalue decompositions needed for each algorithm respectively. In Tables 3.1, 3.3, 3.5, 3.7 and 3.9, we report $f_{\theta\theta}$ in each step of the DDI algorithm. We observe that $f_{\theta\theta}$ never approaches zero, this verifies theoretically that the DDI algorithm has local quadratic convergence in all our test examples.

Example 3.3.2. *We take the A matrix to be the* 7×7 *stable matrix in [HS89, Example 5.2], a matrix that arises in the stability analysis of a heated rod controlled by a dead-beat controller. Table 3.1 shows the iterates of the Algorithm DDI. We observe that the method converges fast, in a couple of iterations. Table 3.1 shows the local quadratic convergence of the DDI algorithm. Table 3.2 shows that the cost of BB Algorithm is approximately three times of the cost of the DDI algorithm.*

Example 3.3.3. *Orr-Sommerfeld matrix of EigTool [Wri02b] is a test matrix ob-*

Table 3.1: Iterates of the Algorithm DDI on Example 3.3.2

Algorithm DDI				
i	ε_i	θ_i	$f_{\theta\theta}(\varepsilon_i, \theta_i)$	$\|\|F(\varepsilon_i, \theta_i)\|\|$
0	0	2.118832e-01	-7.792936e-01	3.410160e-01
1	6.818395e-01	2.042892e-01	-2.635779e-02	5.717587e-03
2	6.793675e-01	-1.203319e-02	-3.149541e-02	9.143296e-04
3	6.806481e-01	-4.858668e-04	-2.986638e-02	1.698551e-05
4	6.806575e-01	-2.208443e-07	-2.984173e-02	9.258642e-09
5	6.806575e-01	-7.199569e-14	-2.984171e-02	2.745284e-15

Table 3.2: Comparison of algorithms for Example 3.3.2

Method	#LU	#SVD	#SYMP
BB	0	16	3
IDM	6	0	1

tained by the discretization of the Orr-Sommerfeld operator. Here we take A to be the Orr-Sommerfeld matrix of dimension 200×200 *(divided) scaled by 50000. The scaling is necessary for making A stable. Table 3.3 shows quadratic convergence of the DDI algorithm. Table 3.4 shows that high number of symplectic eigenvalue decompositions make the BB algorithm slower in this case whereas the DDI algorithm is again much faster.*

Example 3.3.4. *The Tolosa matrix arises in the stability analysis of a model of an*

Table 3.3: Iterates of the Algorithm DDI on Example 3.3.3

Algorithm DDI				
i	ε_i	θ_i	$f_{\theta\theta}(\varepsilon_i, \theta_i)$	$\|\|F(\varepsilon_i, \theta_i)\|\|$
0	0	-3.141593e+000	-2.420602e+000	3.334017e-003
1	6.668033e-003	-3.141593e+000	-1.111981e+000	9.971928e-009
2	6.668033e-003	-3.141593e+000	-1.111981e+000	2.352849e-016

Table 3.4: Comparison of algorithms for Example 3.3.3

Method	#LU	#SVD	#SYMP
BB	0	36	7
IDM	3	0	1

Table 3.5: Iterates of the Algorithm DDI on Example 3.3.4

Algorithm DDI				
i	ε_i	θ_i	$f_{\theta\theta}(\varepsilon_i, \theta_i)$	$\|\|F(\varepsilon_i, \theta_i)\|\|$
0	0	1.873922e+000	-5.631099e-003	2.887954e-004
1	5.705540e-004	1.881527e+000	-1.747326e-003	2.949183e-005
2	5.696610e-004	1.898392e+000	-1.730629e-003	2.792801e-007
3	5.701565e-004	1.898490e+000	-1.728859e-003	1.506027e-010
4	5.701565e-004	1.898490e+000	-1.728859e-003	8.786281e-018

airplane in flight. Here, A is the 1090×1090 *matrix from EigTool [Wri02b] divided by* 2000 *to make the Tolosa matrix stable. Tables 3.5 and 3.6 demonstrate the results. For this example the DDI algorithm is much faster since the BB algorithm requires many SVD decompositions.*

Example 3.3.5. *The matrix* `markov_demo`*(30) is a* 465×465 *example that is a Markov chain transition matrix for a random walk on a 30 by 30 triangular lattice, we choose A to be this matrix divided by* 2 *to make the transition matrix stable. Table 3.7 shows the iterates of the DDI algorithm. Table 3.8 demonstrates that*

Table 3.6: Comparison of algorithms for Example 3.3.4

Method	#LU	#SVD	#SYMP
BB	0	459	6
IDM	5	0	1

Table 3.7: Newton Iterates on Example 3.3.5

Algorithm DDI				
i	ε_i	θ_i	$f_{\theta\theta}(\varepsilon_i, \theta_i)$	$\|\|F(\varepsilon_i, \theta_i)\|\|$
0	0	0	-1.517494e+000	2.371689e-001
1	4.743378e-001	0	-2.162368e-001	8.189826e-017
2	4.743378e-001	0	-2.162368e-001	1.183417e-017

Table 3.8: Comparison of algorithms for Example 3.3.5

Method	#LU	#SVD	#SYMP
BB	0	19	2
IDM	2	0	1

DDI is faster more than a factor of 2.

Example 3.3.6. *Here, we set* $A =$ `pde_demo`$(900)/10$. *The* `pde_demo`(900) *example is a 900* $\times 900$ *matrix obtained by a five-point central finite difference discretization of a two-dimensional variable-coefficient linear elliptic equation. Tables 3.9 and 3.10 demonstrate the results showing that DDI is at least 7 times faster.*

Table 3.9: Iterates of the Algorithm DDI on Example 3.3.6

Algorithm DDI				
i	ε_i	θ_i	$f_{\theta\theta}(\varepsilon_i, \theta_i)$	$\|\|F(\varepsilon_i, \theta_i)\|\|$
0	0	1.810990e-001	-1.799665e+001	7.623228e-002
1	1.795854e-002	1.768948e-001	-4.518592e+000	5.441128e-002
2	1.679771e-002	1.652988e-001	-3.192297e+000	6.395900e-003
3	1.731956e-002	1.620758e-001	-2.637687e+000	1.354115e-003
4	1.732250e-002	1.615521e-001	-2.603736e+000	9.504772e-006
5	1.732309e-002	1.615463e-001	-2.603017e+000	3.116738e-009
6	1.732309e-002	1.615463e-001	-2.603017e+000	1.140411e-016

Table 3.10: Comparison of algorithms for Example 3.3.6

Method	#LU	#SVD	#SYMP
BB	0	59	7
IDM	7	0	1

3.4 Computing the numerical radius

The numerical radius $r(A)$ of a square matrix A admits the following well-known characterization

$$r(A) = \max_{\theta \in [0,2\pi)} \lambda_{\max} H(Ae^{i\theta})$$

where $H(Ae^{i\theta}) := \frac{1}{2}(Ae^{i\theta} + A^H e^{-i\theta})$ is the Hermitian part of $Ae^{i\theta}$ and $\lambda_{\max} H(Ae^{i\theta})$ denotes the largest eigenvalue of $H(Ae^{i\theta})$ (whose eigenvalues are all real). Hence, computing $r(A)$ amounts to maximizing the objective function $p : [0, 2\pi) \to \mathbb{R}$ where

$$p(\theta) = \lambda_{\max} H(Ae^{i\theta})$$

is a piecewise smooth function.

The first globally convergent algorithm for numerical radius is the Mengi-Overton (MO) algorithm [MO05]. This algorithm is an extension of the BB algorithm [BB90b] to the numerical radius. Here we sketch the algorithm; the details can be found in [Men06]. The MO algorithm produces increasing estimates r^j, $j = 1, 2, \ldots$ that converge to $r(A)$. Given an estimate r^j, at the first phase we find the intervals $I_1^{j+1}, I_2^{j+1}, \cdots, I_{m^{j+1}}^{j+1}$ on which p is strictly bigger than the current estimate r^j. This phase is accomplished by finding the unit norm eigenvalues of

the symplectic pencil

$$W_\alpha(\lambda) := \begin{bmatrix} 2\alpha I & -A^H \\ I & 0 \end{bmatrix} - \lambda \begin{bmatrix} A & 0 \\ 0 & I \end{bmatrix}$$

for $\alpha = r^j$, followed by several eigenvalue decompositions of order $2n$ as a checking step. In the second phase, p is evaluated at the midpoints of these intervals and the maximum value observed is taken as the next estimate r^{j+1}. As long as no singular pencils are encountered, the algorithm does not break down and the iterates r^j are strictly increasing.

The quadratic convergence of this algorithm was proved in [MO05] under an assumption that holds generically. Here we show that in fact the assumption always holds. We start by showing some regularity properties around a local maximizer of p.

Consider the function $T : \mathbb{R} \to \mathbb{C}^{n\times n}$

$$T(\theta) = H(Ae^{i\theta}).$$

Note that $T(\theta)$ is Hermitian for all $\theta \in \mathbb{R}$ and is an analytic function of θ. By [GLR82, Theorem S6.3] (see also [Kat82, Theorem II-§6.1] or [Kat82, Theorem II-§1.10]), there exist real-valued functions $\mu_1(\theta), \cdots, \mu_n(\theta)$ and a matrix-valued function $U(\theta)$, which are analytic functions of θ, such that for every $\theta \in \mathbb{R}$,

$$T(\theta) = U(\theta)^{-1}\mathrm{diag}\Big[\mu_1(\theta), \cdots, \mu_n(\theta)\Big]U(\theta), \quad U(\theta)U(\theta)^H = I. \tag{3.42}$$

The smooth scalar functions $\mu_1(\theta), \cdots, \mu_n(\theta)$ are the unordered eigenvalues of

$T(\theta)$. Let

$$\lambda_{\max}(T(\theta)) = \lambda_1(\theta) \geq \lambda_2(\theta) \geq \ldots \geq \lambda_n(\theta)$$

denote the ordered eigenvalues of $T(\theta)$ obtained by ordering the functions $\mu_j(\theta)$. The graphs of unordered eigenvalues may cross each other for some values of θ. If such a crossing takes place, the graph of $\lambda_j(\theta)$ jumps from one smooth curve to another at the crossing point. However, in any finite interval of θ, there are only finitely many crossing points and $\lambda_j(\theta)$ is piecewise analytic for $j = 1, 2 \ldots, n$ [Kat82, II-§6.4]. This observation leads to the following lemma.

Lemma 3.4.1. *Given any $\theta_0 \in \mathbb{R}$ and $1 \leq j \leq n$, there are two functions $p_-, p_+ : \mathbb{R} \to \mathbb{R}$ and $p_+ : \mathbb{R} \to \mathbb{R}$ such that in a neighborhood of θ_0, p_- and p_+ are real analytic and*

$$\lambda_j(\theta) = \begin{cases} p_-(\theta) & \text{if} \quad \theta \in (-\infty, \theta_0] \\ p_+(\theta) & \text{else.} \end{cases}$$

Furthermore, when $j = 1$, we have $\lambda_1(\theta) = \max(p_-, p_+)$, or equivalently $p_-(\theta) \geq p_+(\theta)$ for $\theta \leq \theta_0$, and $p_-(\theta) \leq p_+(\theta)$ for $\theta \geq \theta_0$.

Note that the function $\lambda_1(\theta)$ is the 2π-periodic extension of $p(\theta)$ to the real line and is equal to $p(\theta)$ on $[0, 2\pi)$. We have by definition

$$r(A) = \max_{\theta \in \mathbb{R}} \lambda_1(\theta).$$

Consider $\lambda_1(\theta)$ around a local maximum which we denote by θ_M. From Lemma 3.4.1, it follows that θ_M is a local maximum of both p_- and p_+. Thus, the first derivatives of p_- and p_+ at θ_M match and they are both equal to 0. It is also true that the second derivatives of p_- and p_+ must match. If this were not the case, we

would have $p_-(\theta) > p_+(\theta)$ or $p_-(\theta) < p_+(\theta)$ in a neighborhood of θ_M except when $\theta = \theta_M$, which would contradict Lemma 3.4.1. This leads to the following result.

Lemma 3.4.2. *$\lambda_1(\theta)$ is C^2 around a local maximizer.*

The BB algorithm [BB90b] was developed originally to compute the H_∞ norm of a transfer matrix. Lemma 3.4.2 is analogous to [BB90b, Theorem 2.3]. So the quadratic convergence proof of BB algorithm [BB90b] can be adapted to our case, yielding:

Theorem 3.4.3. *The MO algorithm is locally quadratically convergent.*

By quadratic convergence, we mean that the set $\cup_k I_k^{j+1}$ converges quadratically to the set of maximizers

$$\Omega_{max} = \{\theta \in [0, 2\pi) \ : \ f(\theta) = r(A)\}.$$

3.4.1 An improved algorithm for the numerical radius

To compute the H_∞ norm of a transfer matrix, Genin et al. [GDV98] proposed a variant of the quadratically convergent BB algorithm [BB90b]. The BB algorithm keeps tracks of some intervals where the objective function is bigger than a threshold that is to be updated at each step, and in the next step the midpoints of these intervals are selected and the objective function is re-evaluated at these points. Genin et al. [GDV98] show that one can obtain cubic convergence and ultimate quartic convergence just by using special points instead of the midpoints. The special point corresponding to each interval is chosen to be the one that maximizes a cubic polynomial model of the objective function whose values

and derivatives match those of the objective function at the endpoints of the interval. The derivatives of the objective function at the endpoints of the intervals are obtained from the eigenvectors of a corresponding pencil. We will extend the algorithm of [GDV98] to the numerical radius computation with a modification to improve the performance.

The proposed new algorithm has two differences from the MO algorithm. The first difference is that cubic interpolation is used instead of the midpoint rule. As in the MO algorithm, the intervals $I_k^{j+1} = (\gamma_k^{j+1}, \zeta_k^{j+1}) \subset [0, 2\pi)$ except that the last interval might wrap around the circle, i.e., $I_{m^{j+1}}^{j+1} = (\gamma_{m^{j+1}}^{j+1}, 2\pi) \cup [0, \zeta_{m^{j+1}}^{j+1})$. We use the notation $cubic(\theta_a, \theta_b)$ to denote the maximizer of the unique cubic polynomial in θ which has the same values and derivatives as p at the points θ_a and θ_b, with the convention that $p(\theta_b) := p(\theta_b - 2\pi)$ for $4\pi > \theta_b \geq 2\pi$. The second difference is the way we determine the intervals. In the MO algorithm, once angles θ for which $p(\theta) = r^{j+1}$ are found, then these angles are sorted in a vector, and p is evaluated at the midpoint of any two elements of this vector. This causes sampling from unnecessary intervals, as p can be less than the threshold r^{j+1} at some of these midpoints. However, from the derivative information at the angle vector, one can easily determine the right intervals I_k^{j+1} on which p is greater than r^{j+1}. This saves some eigenvalue decompositions.

For the cubic interpolation, one needs to compute the derivative of the objective function $p(\theta)$. This can be computed with little extra cost from the eigenvectors of W_α as follows. Assume that α is a simple eigenvalue of $H(Ae^{i\theta})$ and u is a normalized eigenvector corresponding to α. Using standard eigenvalue perturbation theory,

$$p'(\theta) = u^H H_\theta(Ae^{i\theta})u \tag{3.43}$$

where $H_\theta(Ae^{i\theta})$ denotes the derivative of $H(Ae^{i\theta})$ with respect to θ.

More generally, for an eigenspace of dimension k spanned by $n \times k$ orthonormal matrix U, the eigenvalues of the $k \times k$ matrix $U^H H_\theta(Ae^{i\theta})U$ are the derivatives of the k unordered smooth eigenvalues that cross each other at θ. The signs of the derivatives determine whether eigenvalues decrease or increase around a crossing point. The eigenvectors of $H(Ae^{i\theta})$ can be computed from the eigenvalues of the symplectic pencil $W_\alpha(\lambda)$ easily. In fact, for fixed θ, it follows from a straightforward computation that

$$H(Ae^{i\theta})u = \alpha u \iff W_\alpha(e^{i\theta}) \begin{bmatrix} I & 0 \\ 0 & e^{-i\theta}I \end{bmatrix} \begin{bmatrix} u \\ u \end{bmatrix} = 0 \tag{3.44}$$

$$\iff W_\alpha(e^{i\theta}) \begin{bmatrix} u \\ e^{-i\theta}u \end{bmatrix} = 0, \tag{3.45}$$

i.e., the first block of an eigenvector corresponding to a unit modulus eigenvalue $e^{i\theta}$ of W_α is in fact an eigenvector of $H(Ae^{i\theta})$.

A new algorithm for numerical radius

1. Set $j = 0$ and $\phi^0 = [0]$.

2. Update the numerical radius estimate: Compute r^{j+1} using the formula

$$r^{j+1} = \max\{p(\theta) : \theta \in \phi^j\}.$$

3. Update the intervals: Find θ values for which $p(\theta) = r^{j+1}$ holds and compute the derivatives at these points. From these, infer the intervals I_1^{j+1}, I_2^{j+1}, $\cdots$, $I_{m^{j+1}}^{j+1}$ in which $p(\theta) > r^{j+1}$.

4. Calculate the new set of points

$$\phi^{j+1} = \{\phi_1^{j+1}, \phi_2^{j+1}, \ldots, \phi_{m^{j+1}}^{j+1}\}$$

where ϕ_k^{j+1} is a point in the open interval I_k^{j+1} obtained by cubic interpolation

$$\phi_k^{j+1} = \begin{cases} \text{cubic}\left(\gamma_k^{j+1}, \zeta_k^{j+1}\right) & \text{if } \gamma_k^{j+1} < \zeta_k^{j+1} \\ \text{cubic}\left(\gamma_k^{j+1}, \zeta_k^{j+1} + 2\pi\right) \mod 2\pi & \text{otherwise} \end{cases}$$

5. Increment j by one, go to Step 2.

3.4.2 Relation with the inner radius

Another quantity is the distance from the boundary of the field of values to the origin, which is called the *inner radius*. It follows from [CH99, Theorem 2.1] that, the inner radius of a matrix A admits the characterization

$$\hat{r}(A) = \left| \min_{0 \le \theta < 2\pi} \lambda_{\max} H(Ae^{i\theta}) \right| \tag{3.46}$$

$$= \left| \max_{0 \le \theta < 2\pi} \lambda_{\min} H(Ae^{i\theta}) \right|. \tag{3.47}$$

The MO algorithm and our proposed new algorithm for the numerical radius extend easily to the computation of the inner radius, with the only change being that the maximization objective changes from $\lambda_{\max} H(Ae^{i\theta})$ to $\lambda_{\min} H(Ae^{i\theta})$.

The inner radius is closely related to the Crawford number of two Hermitian matrices. Two Hermitian matrices C, D are said to form a *definite pair* if their

Crawford number $\gamma(C, D) > 0$. It is shown in [HTVD02] that

$$\gamma(C, D) = \max\Big(\max_{0\leq\theta<2\pi} \lambda_{\min} H(\hat{A}e^{i\theta}\Big), 0)$$

where $\hat{A} := C + iD$. Hence, the Crawford number of a definite pair (C, D) is equal to the inner radius of $\hat{A}$. In [HTVD02], authors propose an algorithm that tests definiteness by solving one quadratic eigenvalue problem (QEP). If the pair is definite, then a level-set algorithm is applied to compute the Crawford number. This algorithm can be used to compute the inner radius of a matrix as well and reduces to the inner radius algorithm derived from the MO algorithm if the related QEP is linearized to give a symplectic pencil.

Note that our proof of the quadratic convergence of the MO algorithm uses the fact that $j = 1$ in Lemma 3.4.1 and hence does not apply to the inner radius case. However, if $\lambda_{\max} H(Ae^{i\theta})$ is not multiple at its maximum value then there is no crossing of unordered eigenvalues and quadratic convergence is obtained. On the other hand, a slight modification of the argument in [BLO03a, Section 2] that counts the codimension of the set of Hermitian matrices with multiple eigenvalues would prove that for almost all matrices $A \in \mathbb{C}^{n\times n}$, the eigenvalues of $H(Ae^{i\theta})$ are simple for all values of θ. This shows that the level-set algorithm of [HTVD02] and the inner radius algorithm mentioned are quadratically convergent for almost all matrices $A \in \mathbb{C}^{n\times n}$.

3.5 Conclusion of the chapter

In this chapter, in order to compute the distance to instability of a stable matrix, we have presented a new algorithm, the DDI algorithm. Numerical ex-

periments indicate that this algorithm is competitive with and in almost all cases outperforms the existing BB algorithm. The extension of these ideas to the more general problem of computing the H_∞ norm of a transfer matrix is under investigation.

We have also presented an improved algorithm for computing the numerical radius. The algorithm is cubically and globally convergent and improves the existing MO algorithm. Besides, we proved that the assumption made in the MO algorithm is not needed since it always holds.

Chapter 4

Regularity of the Pseudospectral Abscissa and Radius

4.1 Introduction

Let $\|\cdot\|$ denote the vector or matrix 2-norm (spectral norm). For real $\varepsilon \geq 0$, the *ε-pseudospectrum* of a matrix $A \in \mathbb{C}^{n\times n}$ [TE05] is the union of the spectra of nearby matrices,

$$\Lambda_\varepsilon(A) = \{z \in \mathbb{C} : z \in \Lambda(A+E) \text{ for some } E \in \mathbb{C}^{n\times n} \text{ with } \|E\| \leqslant \varepsilon\} \tag{4.1}$$

where $\Lambda(A)$ denotes the spectrum (set of eigenvalues) ofb A. Equivalently, Λ_ε is the upper level set of the norm of the resolvent of $A - zI$,

$$\Lambda_\varepsilon(A) = \{z : \|(A-zI)^{-1}\| \geq \frac{1}{\varepsilon}\} \tag{4.2}$$

and the lower level set of the smallest singular value of $A - zI$,

$$\Lambda_\varepsilon(A) = \{z \in \mathbb{C} : \sigma_n(A - zI) \leq \varepsilon\}. \tag{4.3}$$

The *ε-pseudospectral abscissa* of A is the largest of the real parts of the elements of the pseudospectrum, i.e.,

$$\alpha_\varepsilon(A) = \max\{\text{Re } z : z \in \Lambda_\varepsilon(A)\}. \tag{4.4}$$

The case $\varepsilon = 0$ reduces to the spectral abscissa $\alpha(A)$, which measures the growth or decay of solutions to the continuous-time dynamical system $\dot{x} = Ax$; in particular, $\alpha(A)$ is negative if and only if the solution decays to zero for all initial states. For $\varepsilon > 0$, the pseudospectral abscissa of A characterizes asymptotic behavior when A is subject to perturbations bounded in norm by ε. Furthermore, as ε varies from 0 to ∞, the map α_ε ranges from measuring asymptotic behavior to measuring initial behavior of the solutions to $\dot{x} = Ax$ [BLO03b, p. 86].

The analogous measure for discrete-time systems $x_{k+1} = Ax_k$ is the *ε-pseudospectral radius*

$$\rho_\varepsilon(A) = \max\{|z| : z \in \Lambda_\varepsilon(A)\}.$$

The case $\varepsilon = 0$ reduces to $\rho(A)$, the spectral radius of A, which is less than one if and only if solutions decay to zero for all initial states.

Below, we will refer to points where α_ε or ρ_ε is attained. By this we mean the points $z \in \Lambda_\varepsilon$ where the real part or the modulus respectively is maximized.

For fixed ε, $\Lambda_\varepsilon : A \to \Lambda_\varepsilon(A)$ is continuous [Kar03, Theorem 2.3.7]. Since Λ_ε is set-valued, continuity is to be understood in the Hausdorff metric. Recently, Lewis

and Pang [LP08] proved that Λ_ε has further regularity properties. Specifically, they showed that Λ_ε has a local Lipschitz property known as the Aubin property everywhere except at *resolvent-critical* points (to be defined in the next section). It was also proved that for fixed $\varepsilon > 0$, α_ε (respectively ρ_ε) is Lipschitz continuous at a matrix A if the points where α_ε (respectively ρ_ε) are attained are not resolvent-critical (a consequence of [LP08, Corollary 7.2] and [LP08, Theorem 5.2]). The fact that for a fixed matrix A the number of resolvent-critical points is finite leads to the property that Λ_ε, α_ε and ρ_ε are Lipschitz around a given matrix A for all but finitely many $\varepsilon > 0$ [LP08, Corollary 8.5]. It was conjectured that the points where α_ε is attained are not resolvent-critical [LP08, Conjecture 8.9]. We prove this conjecture, which implies that for fixed $\varepsilon > 0$, α_ε is locally Lipschitz continuous on $\mathbb{C}^{n\times n}$. Our proof also applies to ρ_ε, proving that it is also locally Lipschitz. We also prove the Aubin property of the ε-pseudospectrum with respect to both ε and A for the points $z \in \mathbb{C}$ where α_ε and ρ_ε are attained. Finally, we give a proof showing that Λ_ε can never be "pointed outwards".

4.2 Previous results and notation

Before stating the conjecture, we need the following known results and definitions from [LP08]. We write $\mathrm{MSV} : M^n \rightrightarrows \mathbb{C}^n \times \mathbb{C}^n$, with

$$\mathrm{MSV}(A) := \{(u, v) \mid u, v \text{ minimal left and right singular vectors of } A\}.$$

In this definition, u, v are minimal left and right singular vectors of A if they are unit vectors satisfying

$$\sigma_n(A)u = Av \quad \text{and} \quad \sigma_n(A)v = A^*u,$$

where A^* is the Hermitian transpose of A. The set

$$Y(A) := \{v^*u \mid (u, v) \in MSV(A)\}$$

will be a key tool in our analysis since, for a fixed A and for $z \notin \Lambda(A)$, we have [LP08, Proposition 4.5]

$$Y(A - zI) = \partial(-\sigma_n(A - zI)), \tag{4.5}$$

where ∂ is the subdifferential in the sense of [RW98, Definition 8.3]. This leads to:

Definition 4.2.1. *A point $z \in \mathbb{C}$ is resolvent-critical for $A \in \mathbb{C}^{n\times n}$ if either $z \in \Lambda(A)$ or $0 \in Y(A - zI)$.*

Thus, a resolvent-critical point is either an eigenvalue of A or a critical point of the norm of the resolvent in the nonsmooth sense (see [LP08, Proposition 4.7 and Definition 4.8]).

Now we are ready to state Lewis and Pang's conjecture:

Conjecture 4.2.2. *[LP08, Conjecture 8.9] The points $z \in \Lambda_\varepsilon(A)$ where the pseudospectral abscissa $\alpha_\varepsilon(A)$ is attained are not resolvent-critical.*

In the following, we will also need the Aubin property, a local Lipschitz property for set-valued mappings.

Definition 4.2.3. *(see [RW98, Definition 9.36]) A mapping $S : \mathbb{R}^n \rightrightarrows \mathbb{R}^m$ has the Aubin property at $\bar{x}$ for $\bar{u}$, where $\bar{x} \in \mathbb{R}^n$ and $\bar{u} \in S(\bar{x})$, if gph S is locally closed at $(\bar{x}, \bar{u})$ and there are neighborhoods V of $\bar{x}$ and W of $\bar{u}$, and a constant $\kappa \in \mathbb{R}_+$, such that*

$$S(x') \cap W \subset S(x) + \kappa |x' - x| \mathbb{B} \text{ for all } x, x' \in V$$

where $\mathbb{B}$ is the unit ball in $\mathbb{R}^m$.

4.3 New results

Let bd $\Lambda_\varepsilon(A)$ denote the boundary of the pseudospectrum of A. We now state our main result on the resolvent-critical points of bd $\Lambda_\varepsilon(A)$, which is based on a result in [ABBO11]:

Theorem 4.3.1. *If $z \in$ bd $\Lambda_\varepsilon(A)$ is resolvent-critical for some $\varepsilon > 0$ and a matrix A, then there exists an integer $m \geq 2$, $\tilde{\theta}$ real and $\tilde{\rho}$ positive real such that for all $\omega < \pi/m$, Λ_ε contains m equally spaced circular sectors of angle at least ω centered at z, that is*

$$\Lambda_\varepsilon(A) \supset \{z + \rho e^{i\theta} \mid \theta \in [\tilde{\theta} + 2\pi k/m - \omega/2, \tilde{\theta} + 2\pi k/m + \omega/2], \rho \leq \tilde{\rho}\}$$

for all $k = 0, 1, 2, \ldots, m-1$.

Proof. Assume that $z \in$ bd $\Lambda_\varepsilon(A)$ is resolvent-critical. Since $\varepsilon > 0$, $z \notin \Lambda(A)$, so there exists a pair of singular vectors $(\tilde{u}, \tilde{v}) \in \text{MSV}(A - zI)$ such that $\tilde{v}^* \tilde{u} = 0$. If the smallest singular value of $A - zI$ is simple, then it follows from [ABBO11, Theorem 9 and subsequent discussion] that there exists an integer $m \geq 2$ such that for all $\omega < \pi/m$, Λ_ε contains m circular sectors of angle at least ω centered at z

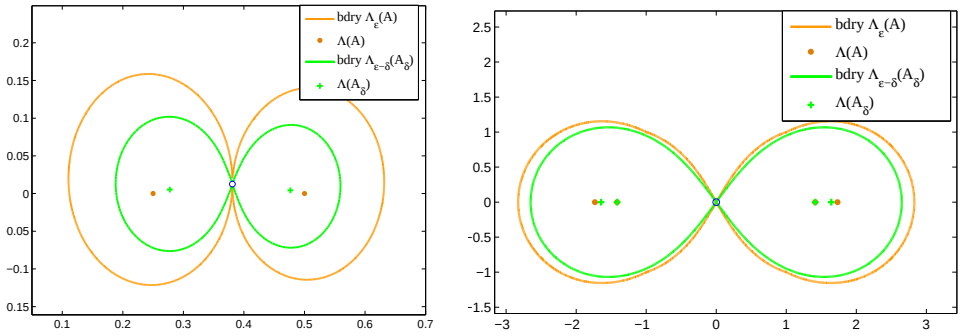

Figure 4.1: Figure illustrates the inclusion $\Lambda_{\varepsilon-\delta}(A_\delta) \subset \Lambda_\varepsilon(A)$. On the left, A is the 4×4 matrix with tangential coalescence given in [ABBO11, right panel of Figure 1] with $\varepsilon = 0.0136$ and $\delta = 0.005$. On the right, A is the reverse diagonal matrix with entries 1,1,3 and 2 (Gracia's example), $\varepsilon = 1$ and $\delta = 0.1$. The smallest singular value of $A - zI$ has multiplicity 2 in both cases, and $m = 2$ in both cases. The plots are obtained with the software package EigTool [Wri02b].

as stated, and so the result is proved. If the smallest singular value of $A - zI$ is not simple, consider a perturbed matrix $A_\delta = A - \delta\tilde{u}\tilde{v}^*$ for $\delta \in (0, \varepsilon)$. Then, the smallest singular value of $A_\delta - zI$ is simple with value $\varepsilon - \delta$ and corresponding singular vectors $\tilde{u}, \tilde{v}$ with $\tilde{u}^*\tilde{v} = 0$. Thus, we can apply [ABBO11, Theorem 9] to A_δ, finding that for all $\omega < \pi/m$, $\Lambda_{\varepsilon-\delta}(A_\delta)$ contains $m \geq 2$ circular sectors of angle at least ω centered at z. But immediately from the definition, using the triangle inequality for the norm, we have (see Figure 4.3)

$$\Lambda_\varepsilon(A) \supset \Lambda_{\varepsilon-\delta}(A_\delta),$$

proving the result. □

We conjecture that the only possible value for m in Theorem 4.3.1 is 2. See [ABBO11, Figure 3].

Clearly, at a point where the pseudospectral abscissa or pseudospectral radius is attained, the pseudospectrum cannot contain $m \geq 2$ circular sectors as defined

above. As a consequence, we have:

Corollary 4.3.2. *For any $\varepsilon > 0$, the points where the pseudospectral abscissa α_ε or pseudospectral radius ρ_ε are attained are not resolvent-critical.*

Thus, Conjecture 4.2.2 is proved. Furthermore, Theorem 4.3.1 implies the following regularity results about $\alpha_\varepsilon, \rho_\varepsilon$ and Λ_ε:

Corollary 4.3.3. *Let $\varepsilon > 0$ be given, and $z_* \in bd\, \Lambda_\varepsilon(A_*)$ be a point where the pseudospectral abscissa $\alpha_\varepsilon(A_*)$ or pseudospectral radius $\rho_\varepsilon(A_*)$ is attained for some matrix A_*. Then, the map $A \to \Lambda_\varepsilon(A)$ has the Aubin property at A_* for z_*.*

Proof. By Corollary 4.3.2, z_* is not resolvent-critical. The result follows from [LP08, Theorem 5.2]. □

Corollary 4.3.4. *For any fixed $\varepsilon > 0$, α_ε and ρ_ε are Lipschitz continuous at any matrix A.*

Proof. Let $A \in \mathbb{C}^{n\times n}$ be given. By Corollary 4.3.3, Λ_ε has the Aubin property at A at all the points where the pseudospectral abscissa or pseudospectral radius is attained. An application of [LP08, Corollary 7.2(a)] with $F = \Lambda_\varepsilon$, $g(x) = \mathrm{Re}(-x)$ proves the Lipschitz continuity of α_ε while using $F = \Lambda_\varepsilon$, $g(x) = -|x|$ proves the Lipschitz continuity of ρ_ε. □

Corollary 4.3.5. *Let $z_* \in \mathbb{C}$ be a point where the pseudospectral abscissa $\alpha_{\varepsilon_*}(A)$ or pseudospectral radius $\rho_{\varepsilon_*}(A)$ is attained for some $\varepsilon_* > 0$ and $A \in \mathbb{C}^{n\times n}$. Then the map $\varepsilon \to \Lambda_\varepsilon(A)$ has the Aubin property at ε_* for z_*.*

Proof. From (4.5) and Corollary 4.3.2, we have $0 \notin Y(A - z_*I) = \partial(-\sigma_n(A - z_*I))$. The result then follows from [LP08, Proposition 5.3], using, as is done there, the inclusion $-\partial(\sigma_n(A - zI)) \subset \partial(-\sigma_n(A - zI))$. □

In the terminology of [BLO03b, Definition 4.5 and its corrigendum], a point z is called nondegenerate with respect to $\Lambda_\varepsilon(A)$ if $Y(A - zI) \neq \{0\}$. Thus, Corollary 4.3.2 implies that a point where the pseudospectral abscissa or pseudospectral radius is attained is nondegenerate. This leads to the following generalization of [BLO03b, Proposition 4.8]:

Proposition 4.3.6. *Let z_* be a point where $\alpha_\varepsilon(A)$ or $\rho_\varepsilon(A)$ is attained for some $\varepsilon > 0$ and a matrix A. Then the boundary of $\Lambda_\varepsilon(A)$ is differentiable at z_*, i.e., the boundary of $\Lambda_\varepsilon(A)$ around z_* is a curve that is differentiable at z_*.*

Proof. This follows from [BLO03b, Proposition 4.8] and the fact that z_* is nondegenerate. □

It was proved in [BLO03b, Proposition 4.8] that the pseudospectrum cannot be "pointed outwards" at nondegenerate points. By this, one means that around a nondegenerate point z_*, the pseudospectrum can never be contained in a circular sector of angle strictly less than π centered at z_*. It was further stated that a more detailed analysis due to Trefethen shows that the pseudospectrum is never pointed outwards. Since the latter result, based on eigenvalue perturbation theory, was never published, we give a new proof here.

Proposition 4.3.7. *Let z_* be on the boundary of the pseudospectrum, i.e. $z_* \in \operatorname{bd} \Lambda_\varepsilon(A)$ for some $\varepsilon > 0$ and a matrix A. The pseudospectrum near z_* cannot be contained in a circular sector of angle $< \pi$ centered at z_*, that is, for all $\omega \in [0, \pi)$, $\tilde{\theta} \in [0, 2\pi)$ and $\tilde{\rho}$ positive real, there exists a point $y \in \Lambda_\epsilon(A)$ such that we have $|y - z_*| \leq \tilde{\rho}$ but y is not contained in the circular sector*

$$z_* + \{\rho e^{i\theta} \mid [\tilde{\theta} - \omega/2, \tilde{\theta} + \omega/2], \rho \leq \tilde{\rho}\}.$$

Proof. If z_* is nondegenerate, then the result follows from [BLO03b, Proposition 4.7]. Otherwise, z_* is resolvent-critical and the result follows from Theorem 4.3.1. □

Part II

Nonsmooth Optimization

Chapter 5

On Nesterov's Nonsmooth Chebyshev - Rosenbrock Functions

5.1 Introduction

In 2008, Nesterov [Nes08] introduced the following *smooth* (differentiable, in fact polynomial) function on $\mathbb{R}^n$:

$$\tilde{f}(x) = \frac{1}{4}(x_1 - 1)^2 + \sum_{i=1}^{n-1}(x_{i+1} - 2x_i^2 + 1)^2.$$

The only stationary point of f is the global minimizer $x^* = [1, 1, \ldots, 1]^T$. Consider the point $\hat{x} = [-1, 1, 1, \ldots, 1]^T$ and the manifold

$$\mathcal{M} = \{x : x_{i+1} - 2x_i^2 + 1 = 0, \quad i = 1, \ldots, n-1\}$$

which contains both x^* and $\hat{x}$. For $x \in \mathcal{M}$,

$$x_{i+1} = 2x_i^2 - 1 = T_2(x_i) = T_{2^i}(x_1), \quad i = 1, \ldots, n-1,$$

where $T_k(x)$ denotes the kth Chebyshev polynomial of the first kind [Sze39, Section 2.4].

The function $\tilde{f}$ is the sum of a quadratic term and a nonnegative sum whose zero set is the manifold $\mathcal{M}$. Minimizing $\tilde{f}$ is equivalent to minimizing the first quadratic term on $\mathcal{M}$. Standard optimization methods, such as Newton's method and the BFGS quasi-Newton method, when applied to minimize $\tilde{f}$ and initiated at $\hat{x}$, generate iterates that, as in the well known Rosenbrock example [GMW81] and its extensions [NW06], approximately "track" $\mathcal{M}$ to approach the minimizer. The iterates do not track $\mathcal{M}$ exactly, but because they typically follow this highly oscillatory manifold fairly closely, the tracking process requires many iterations. To move from $\hat{x}$ to x^* along $\mathcal{M}$ *exactly* would require x_n to trace the graph of the 2^{n-1}th Chebyshev polynomial, which has $2^{n-1} - 1$ extrema in $(-1, 1)$, as x_1 increases from -1 to 1. Hence, $\tilde{f}$ is a challenging test problem for optimization methods.

Nesterov also introduced two *nonsmooth* variants of $\tilde{f}$, the first being

$$\hat{f}(x) = \frac{1}{4}(x_1 - 1)^2 + \sum_{i=1}^{n-1} |x_{i+1} - 2x_i^2 + 1|. \tag{5.1}$$

A contour plot of this function when $n = 2$ is shown on the left side of Figure 5.1. Again, the unique global minimizer is x^*. Like $\tilde{f}$, the function $\hat{f}$ is the sum of a quadratic term and a nonnegative sum whose zero set is the manifold $\mathcal{M}$, so, as previously, minimizing $\hat{f}$ is equivalent to minimizing the first quadratic term on

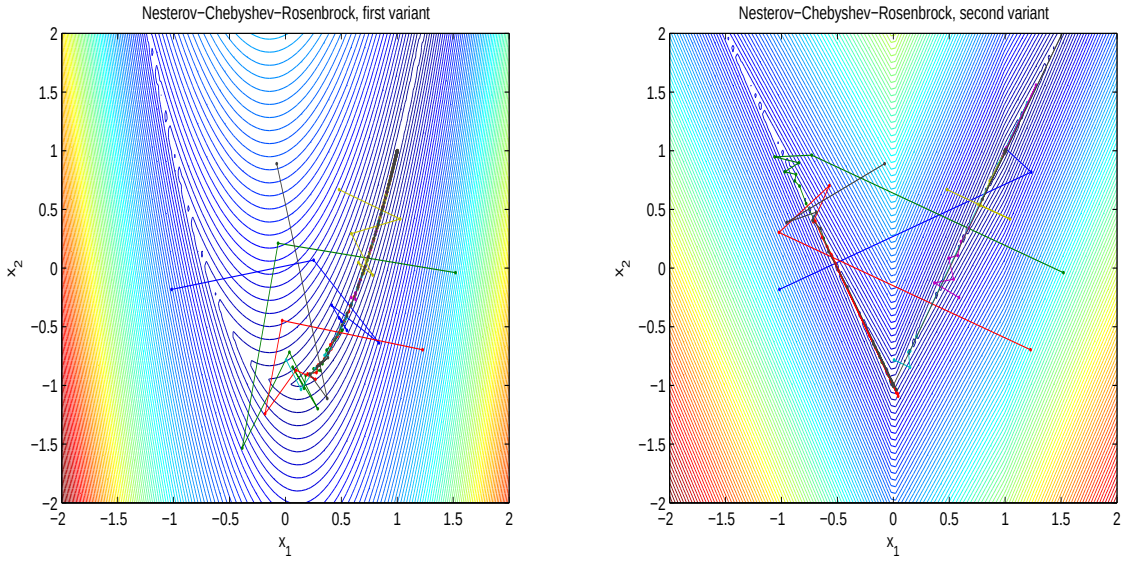

Figure 5.1: Contour plots for Nesterov's first (left) and second (right) nonsmooth Chebyshev-Rosenbrock functions $\hat{f}$ and f respectively, with $n = 2$. Points connected by line segments show the iterates generated by the BFGS method (see Section 5.3) initialized at 7 different randomly generated starting points (iterates plotted later may overwrite those plotted earlier). For the first variant $\hat{f}$, convergence always takes place to the only Clarke stationary point: the global minimizer $x^* = [1, 1]^T$. For the second variant f, some runs of BFGS generate iterates that approximate the nonminimizing Clarke stationary point $[0, -1]^T$ while others converge to the minimizer $[1, 1]^T$.

$\mathcal{M}$, but unlike $\tilde{f}$, the function $\hat{f}$ is not differentiable at points in $\mathcal{M}$. However, as we show in the next section, $\hat{f}$ is *partly smooth* with respect to $\mathcal{M}$, in the sense of [Lew03], at points in $\mathcal{M}$. It follows that, like $\tilde{f}$, the function $\hat{f}$ has only one stationary point — the global minimizer x^* — where by stationary point we mean both in the sense of Clarke and of Mordukhovich. It follows that the global minimizer x^* is the only local minimizer of $\hat{f}$.

The second nonsmooth variant is

$$f(x) = \frac{1}{4}|x_1 - 1| + \sum_{i=1}^{n-1} |x_{i+1} - 2|x_i| + 1|. \tag{5.2}$$

Again, the unique global minimizer is x^*. Consider the set

$$S = \{x : x_{i+1} - 2|x_i| + 1 = 0, \quad i = 1, \ldots, n-1\}. \tag{5.3}$$

Minimizing f is equivalent to minimizing its first term on S. Like $\mathcal{M}$, the set S is highly oscillatory, but it has "corners": it is not a manifold around any point x where any of the components $x_1, \ldots, x_{n-1}$ vanishes. For example, consider the case $n = 2$, for which a contour plot is shown on the right side of Figure 5.1. It is easy to verify that the point $[0, -1]^T$ is Clarke stationary (zero is in the convex hull of gradient limits at the point), but not a local minimizer ($[1, 2]^T$ is a direction of linear descent from $[0, -1]^T$). We will show in the next section that, in fact, f has 2^{n-1} Clarke stationary points, that the only local minimizer is the global minimizer x^*, and furthermore that the only stationary point in the sense of Mordukhovich is x^*.

In the next section, we define stationarity in both senses and present the main results. In Section 5.3, we report on numerical experiments showing the behavior

of nonsmooth minimization algorithms on these functions.

5.2 Main results

Before stating our main results, we will need the following well-known definitions. The Clarke subdifferential or generalized gradient [Cla83] of a locally Lipschitz function on a finite-dimensional space can be defined as follows [BL05, Theorem 6.2.5]. Let ∇ denote gradient.

Definition 5.2.1. *(Clarke subdifferential) Consider a function $\phi : \mathbb{R}^n \to R$ and a point $x \in \mathbb{R}^n$, and assume that ϕ is locally Lipschitz around x. Let $\mathcal{G} \subset \mathbb{R}^n$ be the set of all points where ϕ is differentiable, and $A \subset \mathbb{R}^n$ be an arbitrary set with measure zero. Then the Clarke subdifferential of ϕ at x is*

$$\partial^C \phi(x) = \text{conv } \{ \lim_{m \to \infty} \nabla \phi(x^m) \ : \ x^m \to x, \ x_m \in \mathcal{G}, \ x^m \notin A \}. \tag{5.4}$$

Note that by Rademacher's Theorem [EG92], locally Lipschitz functions are differentiable almost everywhere so A can be chosen as the set of points at which ϕ is not differentiable.

As expounded in [RW98], the Mordukhovich [Mor76] subdifferential is defined as follows.

Definition 5.2.2. *(Mordukhovich subdifferential) Consider a function $\phi : \mathbb{R}^n \to R$ and a point $x \in \mathbb{R}^n$. A vector $v \in \mathbb{R}^n$ is a* regular subgradient *of ϕ at x (written $v \in \hat{\partial}\phi(x)$) if*

$$\liminf_{\substack{z \to x \\ z \neq x}} \frac{\phi(z) - \phi(x) - \langle v, z - x \rangle}{|z - x|} \geq 0,$$

where $\langle \cdot\ ,\ \cdot \rangle$ *is the usual inner product on* $\mathbb{R}^n$*. A vector* $v \in \mathbb{R}^n$ *is a* Mordukhovich subgradient *of* ϕ *at* x *(written* $v \in \partial^M \phi(x)$*) if there exist sequences* x^m *and* v^m *in* $\mathbb{R}^n$ *satisfying*

$$\begin{aligned} x^m &\to x \\ \phi(x^m) &\to \phi(x) \\ v^m &\in \hat{\partial}\phi(x^m) \\ v^m &\to v. \end{aligned}$$

We say that ϕ is *Clarke stationary* at x if $0 \in \partial^C \phi(x)$. Similarly, ϕ is *Mordukhovich stationary* at x if $0 \in \partial^M \phi(x)$. For a locally Lipschitz function ϕ, we have [RW98, Theorem 8.49]

$$\partial^C \phi(x) = \text{conv } \{\partial^M \phi(x)\}. \tag{5.5}$$

The following simple example illustrates equation (5.5).

Example 5.2.3. *For* $g(x) = |x_1| - |x_2|$*,* $x \in \mathbb{R}^2$*, explicit formulas for the Clarke and Mordukhovich subdifferentials can be derived at* $x = [0, 0]^T$*. Using Definitions 5.2.1 and 5.2.2, a straightforward computation leads to*

$$\partial^C g([0,0]^T) = [-1,1] \times [-1,1] \quad \textit{and} \quad \partial^M g([0,0]^T) = [-1,1] \times \{-1,1\},$$

where the former subdifferential is the convex hull of the latter one.

We will need the concept of regularity (also known as subdifferential regularity or Clarke regularity) [RW98], which can be characterized for locally Lipschitz

functions as follows [Lew06, Theorem 6.10].

Definition 5.2.4. *(regularity) A locally Lipschitz function $\phi : \mathbb{R}^n \to \mathbb{R}$ is regular at a point x if and only if its ordinary directional derivative satisfies*

$$\phi'(x; d) = \limsup_{z \to x} \ \langle \nabla \phi(z), d \rangle$$

for every direction $d \in \mathbb{R}^n$.

One consequence of regularity of ϕ at a point x is that $\partial^C \phi(x) = \partial^M \phi(x)$ [BLO02, Proposition 4.1(iii)] and another is that the Clarke stationarity condition $0 \in \partial^C \phi(x)$ is equivalent to the first-order optimality condition $\phi'(x, d) \geq 0$ for all directions d [SY06, Section 14.1].

A property that will be central in our analysis is *partial smoothness* [Lew03].

Definition 5.2.5. *A function ϕ is* partly smooth *at x relative to a manifold $\mathcal{X}$ containing x if*

1. *its restriction to $\mathcal{X}$, denoted by $\phi_{|\mathcal{X}}$, is twice continuously differentiable at x,*
2. *at every point close to $x \in \mathcal{X}$, the function ϕ is regular and has a Mordukhovich subgradient,*
3. *par $\{\partial^M \phi(x)\}$, the subspace parallel to the affine hull of the subdifferential of ϕ at x, is the normal subspace to $\mathcal{X}$ at x, and*
4. *the Mordukhovich subdifferential map $\partial^M \phi$ is continuous at x relative to $\mathcal{X}$.*

We illustrate the definition by proving that $\hat{f}$ is partly smooth.

Lemma 5.2.6. *Nesterov's first nonsmooth Chebyshev-Rosenbrock function $\hat{f}$, defined in* (5.1)*, is partly smooth with respect to $\mathcal{M}$ at all points in $\mathcal{M}$.*

Proof. For each $i \in \{1, \ldots, n-1\}$, consider the function $h_i(x) = |x_{i+1} - 2x_i^2 + 1|$ and the manifold $\mathcal{M}_i := \{x : H_i(x) := x_{i+1} - 2x_i^2 + 1 = 0\}$. By the chain rule [RW98, Proposition 10.5], h_i is globally regular as a composition of two regular functions and we have

$$\partial^M h_i(x) = \nabla H_i(x) \left[\left\{ \partial^M |\cdot| \right\} (x_{i+1} - 2x_i^2 + 1) \right]$$

for any $x \in \mathbb{R}^n$. We have the normal space $N_{\mathcal{M}_i} = \text{Range}(\nabla H_i(x))$ [RW98, Ex. 6.8] which is clearly parallel to the subdifferential $\partial^M h_i(x)$. Since $h_{i|\mathcal{M}_i} = 0$ is smooth and $\partial^M h_i$ is continuous at x relative to $\mathcal{M}_i$, it follows from Definition 5.2.5 that h_i is partly smooth with respect to the manifold $\mathcal{M}_i$. We conclude from [Lew03, Corollary 4.6] and [Lew03, Corollary 4.7] that $\hat{f}(x) = \frac{1}{4}(x_1 - 1)^2 + \sum_{i=1}^{n-1} h_i(x)$ is partly smooth with respect to the manifold $\mathcal{M} = \cap_{i=1}^{n-1} \mathcal{M}_i$ at all points in $\mathcal{M}$. □

It follows that $\hat{f}$ has only one stationary point.

Theorem 5.2.7. *Nesterov's first nonsmooth Chebyshev-Rosenbrock function $\hat{f}$ is Clarke stationary or Mordukhovich stationary only at the unique global minimizer $x^* = [1, 1, \ldots, 1]^T$.*

Proof. If $x \notin \mathcal{M}$, then $\hat{f}$ is smooth and nonstationary at x as the partial derivative of $\hat{f}$ with respect to x_n is ± 1. On the other hand, if $x \in \mathcal{M}$, then the restricted function $\hat{f}_{|\mathcal{M}} = \frac{1}{4}(x_1 - 1)^2$ is smooth and has a critical point only at the global minimizer x^*. If $x \in \mathcal{M}$ and $x \neq x^*$, it follows from [Lew03, Proposition 2.4] that $0 \notin \text{aff } \partial^M \hat{f}(x)$. Thus, $0 \notin \partial^M \hat{f}(x)$. By regularity, we have $\partial^C \hat{f}(x) = \partial^M \hat{f}(x)$ and the result follows. □

The main results of the paper concern Nesterov's second nonsmooth example.

For this we will need the usual sign function:

$$\operatorname{sign}(x) = \begin{cases} 1 & : x > 0, \\ 0 & : x = 0, \\ -1 & : x < 0. \end{cases}$$

We start by stating a simple lemma.

Lemma 5.2.8. *Let S be defined as in (5.3). There are $2^{n-1} - 1$ points in S such that $x_j = 0$ for some $j < n$. Let $\bar{x}$ be such a point. Then $\bar{x}_i$ takes non-integer values between -1 and 1 for $i < j$, $\bar{x}_i = 0$ for $i = j$, $\bar{x}_i = -1$ for $i = j + 1$ and $\bar{x}_i = 1$ if $n \geq i > j + 1$. In particular, $\bar{x}_1 < 1$ (with $\bar{x}_1 = 0$ if $j = 1$).*

Proof. For $j < n$ fixed, it is easy to see that there are 2^{j-1} points in S such that $x_j = 0$. Summing over j, we obtain $2^{n-1} - 1 = \sum_{j=1}^{n-1} 2^{j-1}$ points. The rest of the proof is straightforward. □

Theorem 5.2.9. *Nesterov's second nonsmooth Chebyshev-Rosenbrock function f, defined in (5.2), is Clarke stationary at the $2^{n-1} - 1$ points in the set S with a vanishing x_j for some $j < n$.*

Proof. Let $\bar{x}$ be such a point. Then, using Lemma 5.2.8, we see that around $\bar{x}$ the function $\frac{|x_1-1|}{4}$ is equal to $\frac{1-x_1}{4}$ and furthermore $\bar{x}_i \neq 0$ if $i \neq j$. These observations allow us to write f in a simpler form eliminating most of the absolute values. We first prove the case $j = n - 1$. We will show that in an arbitrarily small neighborhood of $\bar{x}$ the gradient vector (if defined) can take arbitrary signs in each coordinate. This will ensure that $0 \in \partial^C f(\bar{x})$ by (5.4).

Around $\bar{x}$, the function $f(x)$ may be rewritten as

$$f(x) = \frac{1-x_1}{4} + |x_2 + 2c_1x_1 + 1| + \ldots + |x_{n-1} + 2c_{n-2}x_{n-2} + 1| + |x_n - 2|x_{n-1}| + 1| \quad (5.6)$$

where $c_i = -\text{sign}(\bar{x}_i)$, $i = 1, 2, \ldots, n-2$, depend only on the point $\bar{x}$ and are fixed. (Note that $\bar{x}_i \neq 0$ for $i < j = n-1$). Since $\bar{x} \in S$ and $\bar{x}_{n-1} = 0$, all the absolute value terms appearing in (5.6) are equal to 0 at $\bar{x}$. By the continuity of f at $\bar{x}$, it is possible to find points x arbitrarily close to $\bar{x}$ such that each of the absolute value terms evaluated at x has arbitrary sign and at those points

$$\nabla f(x) = \left[-\frac{1}{4} + 2c_1d_1, d_1 + 2c_2d_2, \ldots, d_{n-2} + 2c_{n-1}d_{n-1}, d_{n-1} \right]^T$$

where $c_{n-1} := -\text{sign}(x_{n-1})$ and each of d_1, d_2,..., d_{n-1} can be chosen to be $+1$ or -1 as desired. Hence, it is possible to have $\nabla f(x)$ in any of the 2^n quadrants of $\mathbb{R}^n$. Consequently, 0 lies in the convex combination of these gradient vectors and we conclude from (5.4) that $0 \in \partial^C f(\bar{x})$.

The case $j < n-1$ is handled similarly. For a choice of x around $\bar{x}$, we get

$$\nabla f(x) = \left[-\frac{1}{4} + 2c_1d_1, d_1 + 2c_2d_2, \ldots, d_{j-1} + 2c_jd_j, d_j + 2d_{j+1}, d_{j+1} - 2d_{j+2}, \ldots, d_{n-1} \right]^T$$

where $c_i = -\text{sign}(x_i)$, $i = 1, 2, \ldots, j-1$, are fixed (when $j > 1$) and $c_j = -\text{sign}(x_j)$, d_1, d_2,..., d_{n-1} are free parameters to choose from $\{-1, 1\}$. Suppose $d_j = d_j^0$, $d_{j+1} = d_{j+1}^0$,..., $d_{n-1} = d_{n-1}^0$ are fixed. By choosing $c_j, d_1, \ldots, d_{j-1}$ appropriately, the signs of the first j components of $\nabla f(x)$ vector can be chosen to be positive

or negative. Thus, by convexity,

$$\left[0, \ldots, 0, d_j^0 + 2d_{j+1}^0, d_{j+1}^0 - 2d_{j+2}^0, \ldots, d_{n-1}^0\right]^T \in \partial^C f(\bar{x}).$$

Choosing $d_j = -d_j^0, d_{j+1} = -d_{j+1}^0, \ldots, d_{n-1} = -d_{n-1}^0$, we have

$$\left[0, \ldots, 0, -(d_j^0 + 2d_{j+1}^0), -(d_{j+1}^0 - 2d_{j+2}^0), \ldots, -d_{n-1}^0\right]^T \in \partial^C f(\bar{x}).$$

and so by convexity $0 \in \partial^C f(\bar{x})$, completing the proof. □

The following theorem characterizes all the stationary points of f in the sense of both subdifferentials.

Theorem 5.2.10. *Nesterov's second nonsmooth Chebyshev-Rosenbrock function f is Mordukhovich stationary only at the global minimizer $x^* = [1, 1, \ldots, 1]^T$. Furthermore, f is Clarke stationary only at x^* and the $2^{n-1} - 1$ points in S with a vanishing x_j for some $j < n$. None of the Clarke stationary points of f except the global minimizer are local minimizers of f and there exists a direction of linear descent from each of these points.*

Proof. If $x \notin S$, f is smooth at x and we have $0 \notin \partial^M f(x) = \partial^C f(x) = \{\nabla f(x)\}$ since the partial derivative of f with respect to x_n at x is ± 1.

When $x = x^* \in S$, we have $0 \in \hat{\partial} f(x) \subset \partial^M f(x) \subset \partial f^C(x)$. If $x \in S$, $x \neq x^*$ ($x_1 \neq 1$) and $x_j \neq 0$ for $j = 1, 2, \ldots, n-1$, then the set S is a manifold around x. The function f is partly smooth with respect to S at x, with $f_{|S}(x) = \frac{|1-x_1|}{4}$, the restriction of f to S, smooth around x, and x is not a critical point of $f_{|S}$. It follows from [Lew03, Proposition 2.4] that $0 \notin \text{aff } \{\partial^M f(x)\}$. This implies directly that $0 \notin \partial^M f(x)$ and $0 \notin \partial^C f(x) = \text{conv } \{\partial^M f(x)\}$, using (5.5).

The remaining case is when $x \in S$ is such that $x_j = 0$ for some $j < n$. We have $x_1 < 1$. Let $\delta > 0$ be small and x^δ be the unique point near x such that $x^\delta \in S$ and $x_1^\delta = x_1 + \delta$. It follows from the definition of S that $x^\delta = x + \delta v$ where v is a fixed vector independent of $\delta > 0$ for δ sufficiently small. Since $f_{|S}(x) = \frac{1-x_1}{4}$ around x, we have $f(x^\delta) = f(x + \delta v) = f(x) - \frac{1}{4}\delta < f(x)$ which shows that v is a direction of linear descent. Furthermore, we have $0 \notin \hat{\partial} f(x)$ since the existence of the descent direction at $\bar{x}$ implies

$$\liminf_{\substack{z \to \bar{x} \\ z \neq x}} \frac{f(z) - f(x)}{|z - x|} \leq \liminf_{\delta \downarrow 0} \frac{f(x + \delta v) - f(x)}{\delta |v|} = -\frac{1}{4|v|} < 0.$$

We want to prove that $0 \notin \partial^M f(x)$. This requires an investigation of the regular subdifferential $\hat{\partial} f(y)$ for y near x. Let y be a point near x, $y \neq x$. We have $x_j = 0$, so we distinguish two cases: $y \notin S$, $\{y \in S$ and $y_j \neq 0\}$. (If $y \in S$ and $y_j = 0$, then, for y to be near x, we would need $y = x$.)

1. $\underline{y \notin S}$: $\nabla f(y)$ exists, we have $\hat{\partial} f(y) = \{\nabla f(y)\}$ and the n-th coordinate of $\nabla f(y)$ is ± 1. This shows that there exists no sequence $y^m \to x$ such that $y^m \notin S$ for all m with $\hat{\partial} f(y^m) = \{\nabla f(y^m)\} \ni v^m \to 0$.

2. $\underline{y \in S \text{ and } y_j \neq 0}$: We have, for y sufficiently close to x, that $y_k \neq 0$ for $k = 1$, $\ldots$, n and

$$S = \{x : F_i(x) = 0, \quad i = 1, \ldots, n-1\}$$

where $F_i(x) = x_{i+1} - 2|x_i| + 1$ is smooth at y. Hence, S is a manifold around y and it is easy to see that f is partly smooth at y with respect to S. The restricted function $f_{|S}(x) = \frac{1-x_1}{4}$ is smooth at y and since $y_1 < 1$, y is not a critical point of f, so from [Lew03, Proposition 2.4] we conclude

that $0 \notin \text{aff}\,\{\partial^M f(y)\}$ which leads to $0 \notin \hat{\partial} f(y)$. Furthermore, by [Lew03, Proposition 2.2], we have

$$\hat{\partial} f(y) \subset \nabla g(y) + N_S(y) \tag{5.7}$$

where $g(x) = \frac{1-x_1}{4}$ and $N_S(y)$ is the normal space to S at y. The normal space to S at y coincides with the normal cone to S at y so by [RW98, Ex. 6.8]

$$N_S(y) = \text{Range}(\nabla F)$$

where

$$\nabla F(y)^T = \left[\frac{\partial F_i}{\partial x_j}(y)\right]_{i,j=1}^{n-1,n} \in \mathbb{R}^{(n-1)\times n}$$

is the Jacobian matrix. We have

$$\nabla F(y) = \begin{bmatrix} -2\,\text{sign}(y_1) & & & & \\ 1 & -2\,\text{sign}(y_2) & & & \\ & 1 & \ddots & & \\ & & \ddots & -2\,\text{sign}(y_{n-1}) & \\ & & & 1 & \end{bmatrix} \in \mathbb{R}^{n\times(n-1)} \tag{5.8}$$

and $\nabla g(y) = [-1/4, 0, \ldots, 0]^T$. From (5.7) and (5.8), we see that $0 \in \hat{\partial} f(y)$ is possible only if $[1/4, 0, \ldots, 0]^T \in N_S(y)$. A straightforward calculation shows that this is impossible. We conclude that $0 \notin \hat{\partial} f(y)$. The next step is to investigate the possible limits of $v^m \in \hat{\partial} f(y^m)$ as the sequence $y^m \in S$ approaches x. Let y^m be a sequence such that $y^m \to x$, $y^m \in S$ and $y^m \neq x$ for all m (this implies $y_j^m \neq 0$ for all m as before). Without loss of generality, assume $y_j^m > 0$ for all m. For fixed $k \in \{1, 2, \ldots, n\}$, the quantity $\text{sign}(y_k^m)$ does not depend on m and is nonzero. Thus, $G := \nabla F(y^m)$ does not depend

on m. Let $v \in \mathbb{R}^n$ be such that $\hat{\partial} f(y^m) \ni v^m \to v$. From (5.7), we have $v^m = [-1/4, 0, \ldots, 0]^T + Gc^m$ for some $c^m \in \mathbb{R}^{n-1}$. Since $v^m \to v$ and G has full rank, we have $c^m \to c \in \mathbb{R}^{n-1}$ and $v = [-1/4, 0, \ldots, 0]^T + Gc$. As previously, $v = 0$ is impossible.

We conclude that $0 \notin \partial^M f(x)$. Since we already know from Theorem 5.2.9 that $0 \in \partial^C f(x)$, this completes the proof of the theorem. □

It follows immediately from Theorem 5.2.10 that f is not regular at the $2^{n-1} - 1$ non-locally-minimizing Clarke stationary points of f: see the comments after Definition 5.2.4.

5.3 Numerical experiments

Nesterov has observed that Newton's method with an inexact line search, when applied to minimize the *smooth* function $\tilde{f}$ initiated at $\hat{x}$, takes many iterations to reduce the value of the function below a small tolerance ϵ. Indeed, the number of iterations is typically exponential in n, although quadratic convergence is observed eventually if the method is run for long enough. Our experimental results are mainly obtained using the BFGS quasi-Newton algorithm with a line search based on the Armijo and "weak Wolfe" conditions, a well-known method generally used to optimize *smooth* functions [NW06]. However, as explained in [LO], BFGS with the same line search is surprisingly effective for nonsmooth functions too. For the results reported below, we used a publicly available MATLAB implementation.[1]

For smooth but nonconvex functions such as $\tilde{f}$, there is no theorem known that guarantees that the BFGS iterates will converge to a stationary point, and

[1] http://www.cs.nyu.edu/overton/software/hanso

pathological counterexamples have been constructed [Dai02, Mas04], although, unlike $\tilde{f}$, these are not analytic. However, it is widely accepted that BFGS generally produces sequences converging to local minimizers of smooth, nonconvex functions [LF01], so it is not surprising that this is the case for $\tilde{f}$, with superlinear convergence to x^* in the limit. As with Newton's method, many iterations are required. For $n = 8$, starting at $\hat{x}$ and with the initial inverse Hessian approximation H set to the identity matrix I, the BFGS method requires about 6700 iterations to reduce $\tilde{f}$ below 10^{-15}, and for $n = 10$, nearly 50,000 iterations are needed.

For nonsmooth functions, there is no general convergence theory for the BFGS method, but as discussed in [LO], when applied to locally Lipschitz functions the method seems to always generate sequences of function values converging linearly to Clarke stationary values, and our experiments confirm this observation for small n for both nonsmooth functions studied in this paper. To apply BFGS to Nesterov's first nonsmooth variant $\hat{f}$, we cannot use $\hat{x}$ for the initial point as the method immediately breaks down, $\hat{f}$ being nondifferentiable at $\hat{x}$. Instead, we initialize x randomly, retaining the identity matrix for initializing H. The left panel of Figure 5.1 shows the iterates generated by BFGS for the case $n = 2$ using 7 random starting points: all sequences of iterates converge to the global minimizer $x^* = [1, 1]^T$. However, the accuracy to which BFGS can minimize $\hat{f}$ drops rapidly as n increases. Because of the difficulty of the problem combined with the limited machine precision, the method breaks down, that is the line search fails to return a point satisfying the Armijo and weak Wolfe conditions, at an iterate x that is close to $\mathcal{M}$ but not very near x^*. When the calculations are carried out to higher precision, more accurate results are obtained [Kak11]. For example, for $n = 4$, using standard IEEE "double" precision (about 16 decimal digits), from

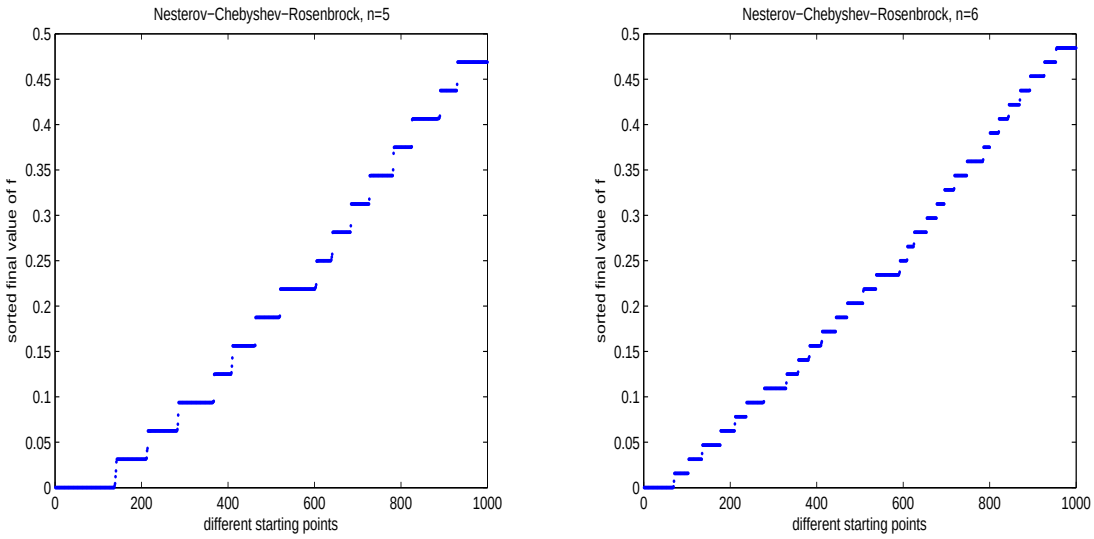

Figure 5.2: Left: sorted final values of f for 1000 randomly generated starting points, when $n = 5$: BFGS finds all 16 Clarke stationary points. Right: same with $n = 6$: BFGS finds all 32 Clarke stationary points.

most starting points BFGS reduces $\hat{f}$ to final values ranging from 10^{-3} to 10^{-2}, while using "double double" precision (about 32 decimal digits), from the same starting points, the final values that are obtained range from 10^{-4} to 10^{-3}.

For Nesterov's second nonsmooth variant f, we find that BFGS generates iterates approximating Clarke stationary points, but not necessarily the minimizer x^*. The iterates for the case $n = 2$, again for 7 randomly generated starting points, are shown in the right panel of Figure 5.1. Most of the runs converge to the minimizer $[1, 1]^T$, but some terminate near the Clarke stationary point $[0, -1]^T$. For $n \leq 6$, given enough randomly generated starting points, BFGS finds, that is approximates well, all 2^{n-1} Clarke stationary points. The left and right panels of Figure 5.2 plot final values of f found by 1000 runs of BFGS starting with random x and $H = I$, sorted into increasing order, for the cases $n = 5$ and $n = 6$ respectively. Most runs find either the minimizer or one of the $2^{n-1} - 1$ nonminimizing Clarke

stationary points, although a few runs break down away from these points. For $n = 7$, the method usually breaks down without finding any Clarke stationary point, presumably because of the limitations of machine precision.

Experiments with the gradient sampling algorithm [BLO05] and Kiwiel's bundle code [Kiw08] give similar results. Both of these methods have well established convergence theories ensuring convergence to Clarke stationary points. However, it remains an open question whether the nonminimizing Clarke stationary points are points of attraction for any of these algorithms. For small n, the computations usually terminate near Clarke stationary points because, eventually, rounding error prevents the method from obtaining a lower point in the line search. But this does not establish whether, in exact arithmetic, the methods would actually generate sequences converging to the nonminimizing Clarke stationary points. Indeed, experiments in [Kak11] suggest that the higher the precision used, the more likely BFGS is to move away from the neighborhood of a nonminimizing Clarke stationary point and eventually find a lower one, perhaps the minimizer.

Another observation is the difficulty of finding descent directions from the nonminimizing Clarke stationary points using random search. Although we know that such descent directions exist by Theorem 5.2.10, numerical experiments show that finding a descent direction by random search typically needs exponentially many trials. For example, when $n = 5$, usually 100,000 random trials do not suffice to find a descent direction. This illustrates the difficulty faced by an optimization method in moving away from these points.

5.4 Conclusion of the chapter

Nesterov's Chebyshev-Rosenbrock functions provide very interesting examples for optimization, both in theory and in practice. Specifically, the smooth function $\tilde{f}$, the first nonsmooth function $\hat{f}$ and the second nonsmooth function f are very challenging nonconvex instances of smooth functions, partly smooth functions and non-regular functions respectively. As far as we know, Nesterov's function f is the first documented case for which methods for nonsmooth optimization result in the approximation of Clarke stationary points from which there exist directions of linear descent. This observation is primarily due to Kiwiel [Kiw08]. Furthermore, since all first-order nonsmooth optimization methods, including bundle methods [Kiw85], the gradient sampling method [BLO05] and the BFGS method [LO], are based on sampling gradient or subgradient information, the results given here for f suggest that limitation of convergence results to Clarke stationary points may be unavoidable, in the sense that one may not in general be able to expect stronger results such as convergence only to Mordukhovich stationary points. Nonetheless, it remains an open question as to whether the nonminimizing Clarke stationary points of f are actually points of attraction for methods using exact arithmetic.

Part III

Halftoning and sigma-delta quantization

Chapter 6

Optimization-based halftoning and sigma-delta quantization

6.1 Introduction

Digital halftoning is a core process governing most digital printing and many display devices by which continuous tone images are converted to discrete-tone images where only a limited number of tones are available. In this chapter, we focus only on gray-scale images although (digital) halftoning applies to color images as well. Given a gray-scale image $f : G \to [0, 1]$ on an integer grid $G := \{1, \ldots, n_x\} \times \{1, \ldots, n_y\}$, (digital) halftoning is the process of forming an image consisting of only black and white dots that resembles the original gray-scale image as much as possible. The number of black dots depends on how dark the image is.

There have been many different approaches to this problem: error diffusion, least square error minimization, global search or direct binary search (DBS), threshold halftoning methods such as ordered dithering and stochastic dithering

Figure 6.1: A halftone image obtained with the minimization of the energy (6.1), taken from [TSG+11].

to name a few (see [Uli87] for a detailed reference). Perhaps the most popular approach is the error diffusion algorithm of Floyd and Steinberg [Flo76] due to its efficiency and low computational complexity. In fact, it is crucial to design halftoning algorithms with linear or almost linear complexity in the number of pixel points; otherwise, it would not be practical for printers and display devices.

Since the judge of the quality of the halftoned image is a human observer, halftoning algorithms are often based on the properties of the human visual system (HVS). For example, in the DBS algorithm, the resulting halftoned image minimizes an HVS-based cost function that measures the perceived error between the gray-scale image and the halftone image.

Mots of the existing halftoning methods are local in the sense that the decision of placing a black dot in a certain position is made by considering only nearby gray values but not the whole picture. A recent approach, introduced in [TSG+11], is

a non-local method that formulates halftoning as a global energy minimization problem for the first time to our knowledge. Their energy functional is a difference of two convex functionals, one for the attraction of the black dots to be inserted for halftoning, which we denote by *halftoning dots*, to the image gray values, and the other one for the repulsion between the dots. Figure 6.1 shows an example of a halftone image obtained this way. Let m be the number of halftoning dots, located at the positions $p := \{p_k\}_{k=1}^m$ where the k-th location p_k has coordinates $(p_{k,x}\ p_{k,y})^T \in G$ and where T denotes transpose. The optimal locations are given by the minimizer(s) of the energy

$$E(p) = \sum_{k=1}^{m} \sum_{(i,j)\in G} w_{ij} \left| p_k - \begin{pmatrix} i \\ j \end{pmatrix} \right| - \sum_{k=1}^{m} \sum_{l=k+1}^{m} |p_k - p_l| \tag{6.1}$$

where $w_{ij} = 1 - f_{ij}$ is the gray value of the pixel (i, j) (0 for white, 1 for black) and $m := \lfloor \sum_{i,j} w_{ij} \rfloor$. This choice of m can be explained by the fact that HVS is a low-pass filter that averages pixel values. For the halftone image to resemble the original image, the average of the gray values of the halftone image should approximately match that of the original image, i.e., $\frac{m}{n_x n_y}$ should be close to $\frac{\sum_{i,j} w_{ij}}{n_x n_y}$. Since m is an integer, it is reasonable to set m to the integer part of $\sum_{i,j} w_{ij}$ as above. For simplicity of the presentation, we assume that $\sum_{i,j} w_{ij}$ is an integer, i.e. we have

$$m = \sum_{i,j} w_{ij}. \tag{6.2}$$

The following generalized energy

$$E_\varphi(p) = \sum_{k=1}^{m} \sum_{(i,j)\in G} w_{ij} \varphi \left(\left| p_k - \begin{pmatrix} i \\ j \end{pmatrix} \right| \right) - \lambda \sum_{k=1}^{m} \sum_{l=k+1}^{m} \varphi \left(|p_k - p_l| \right) \tag{6.3}$$

is also considered in [TSG+11] where the function $\varphi : [0, \infty] \to \mathbb{R}$ is chosen to make E_φ coercive. It is possible to motivate such an energy by electrostatics, viewing the halftoning dots and pixels as point charges. Note that in two dimensions the force that a point particle k at position p_k with charge w_k applies to an other point particle l at position p_l with charge w_l is given by

$$F_{k,l} = \frac{cw_k w_l}{||p_k - p_l||} e_{k,l} = \frac{cw_k w_l}{||p_k - p_l||^2}(p_k - p_l) \tag{6.4}$$

where $e_{k-l} := \frac{p_k - p_l}{||p_k - p_l||}$ is the unit vector from particle k to l and c is a positive constant. A positive force implies it is repulsive, while a negative force implies it is attractive. It is reasonable to introduce an electrostatic repulsion between the halftoning dots because in practice, we want dispersion of the dots in the uniformly colored regions. On the other hand, in the textured regions or at image boundaries, we want halftoning dots be close to darker regions. These constraints can be satisfied by adding an attractive electrostatic force between each pixel and each halftoning dot, a force proportional to the gray values w_{ij} of the image. It is also natural to choose the charge of the halftoning dots to be the same, as otherwise some points would have a bigger impact on the overall system. Without loss of generality we can choose the charge of p_k's to be $+1$. The overall system can be thought of as consisting of m positively charged particles (halftoning dots) with charge $+1$ that repel each other, and are attracted by the particles with charge $-w_{ij}$ at pixel (i, j) for all $(i, j) \in G$. For an equilibrium to be reached, the overall charge of the system should be zero, as indeed is the case by the condition (6.2).

The overall force acting on a particle at position p_k will be

$$F_k = c\left(\sum_{i,j} -w_{ij}\frac{\left(p_k - \binom{i}{j}\right)}{||p_k - \binom{i}{j}||^2} + \sum_{l\neq k}\frac{(p_l - p_k)}{||p_l - p_k||^2}\right). \tag{6.5}$$

Let $\tilde{E}(p)$ be the total electrostatic potential of this system of point particles. Using $-\frac{\partial \tilde{E}}{\partial p_k} = F_k$ and integrating (6.5) gives the electrostatic potential energy of the system

$$\tilde{E}(p) = c\left(\sum_{k=1}^{m}\sum_{(i,j)\in G} w_{ij}\log\left(\left|p_k - \binom{i}{j}\right|\right) - \sum_{k=1}^{m}\sum_{l=k+1}^{m}\log\left(|p_k - p_l|\right)\right) \tag{6.6}$$

which is (6.3) (up to a constant factor c) with $\varphi(s) = \log(s)$. The appearance of log terms in the energy is due to the fact that $\varphi(s) = \log(s)$ is a fundamental solution (a nonconstant solution with rotational symmetry) for Laplace equation in two dimensions.

Note that the energy (6.1) corresponds to $\varphi(s) = |s|$ which amounts to a non-decaying repulsion (or attraction) with distance $|s|$ when compared to the $\varphi(s) = \log(s)$ case where the decay is $\mathcal{O}(1/|s|)$. In this sense, the energy (6.1) is more sensitive to the long-distance interactions among the halftoning dots and pixels.

It is easy to find examples showing that the energy (6.3) is in general not convex. Consider a uniformly colored 2×2 grid with $w_{ij} = 1/2$ for $i = 1, 2$ and $j = 1, 2$ with $\varphi(s) = |s|$. The points $\hat{p}$ and $\bar{p}$ are both optimizers with $\hat{p}_1 = (1, 1), \hat{p}_2 = (2, 2)$, $\bar{p}_1 = (2, 1)$ and $\bar{p}_2 = (1, 2)$; however $\frac{\hat{p}+\bar{p}}{2}$ is not, showing nonconvexity of E.

Approximating optimizer(s) of the energy (6.1) or (6.3) with an explicit formula is hard in its full generality; however in one dimension, the energy (6.1) is convex

and an explicit solution is available [TSG+11]. It will be shown in this thesis that this solution is related to (one-bit first-order) sigma-delta quantization. In the next section, we discuss this connection.

6.2 One-dimensional problem

For a one-dimensional signal $\{w_j\}_{j=1}^n$ the energy functional (6.1) becomes

$$E(p) = \sum_{k=1}^{m} \sum_{j=1}^{n} w_j \, |p_k - j| - \sum_{k=1}^{m} \sum_{l=k+1}^{m} |p_k - p_l| \tag{6.7}$$

with $n = n_x$. Assuming that p_k's are in ascending order, i.e., $p_1 \leq p_2 \leq ... \leq p_m$, this energy becomes

$$E(p) = \sum_{k=1}^{m} \Big(\sum_{j=1}^{n} w_j \, |p_k - j| + (m - 2k + 1) p_k \Big) \tag{6.8}$$

which, being a sum of convex and linear functionals, is convex.

By convex calculus, it is possible to compute the minimizer(s) of (6.8) (see [TSG+11, Theorem 3.1]). In particular, $\hat{p}$ is a minimizer with

$$\hat{p}_k = \min\{r \ : \ a_{r-1} < k - \frac{1}{2} \leq a_r, r \geq 1\}, \quad k = 1, 2, \cdots, n \tag{6.9}$$

where $a_0 := 0$ and $a_r := \sum_{j=1}^{r} w(j)$ for $r \geq 1$ is the cumulative sum of the pixel values $\{w_j\}_{j=1}^n$. The k-th halftoning dot $\hat{p}_k$ depends only on the positions of the previous ones $\hat{p}_1, \hat{p}_2, \cdots, \hat{p}_{k-1}$; in other words, only on the values w_j for $j \leq \hat{p}_k$. This is somehow unexpected since there are interactions between all the pairs of halftoning dots.

Note that in (6.9) the halftoning dots are all different from each other. Indeed, placing two black dots or placing only one dot at a particular location would result in the same halftone image. So we assume that the halftoning dots are all distinct from each other. We also assume that the dots are lying on the same grid as the pixels, which is the case in practice. Then, one can identify the points $\{p_k\}_{k=1}^m$ with the $\{0,1\}$-valued sequence $\{Q_j\}_{j=1}^n$ where $Q_j = 1$ if and only if there exists a halftoning dot at the j-th pixel location. This allows us to reformulate the energy functional (6.7) in the space of binary-valued sequences, and the resulting energy functional is

$$E(Q) = \sum_i \sum_j |i-j| w_j Q_i - \frac{1}{2} \sum_i \sum_j |i-j| Q_i Q_j \tag{6.10}$$

$$= \left\langle Q, |\cdot| * (w - \frac{1}{2} Q) \right\rangle \tag{6.11}$$

where, for notational convenience, w and Q are one-sided infinite sequences obtained from $\{w_j\}_{j=1}^n$, $\{Q_j\}_{j=1}^n$ by zero padding, i.e. $w_j := 0$, $Q_j := 0$ for $j > n$ and $j = 0$. The convolution operator $*$ and the dot product $\langle \cdot, \cdot \rangle$ are defined in the usual way with $(a * b)_j := \sum_{k=0}^j a_k b_{j-k}$, $\langle a, b \rangle := \sum_{j=0}^\infty a_j b_j$ for square-summable sequences a and b.

6.2.1 Necessary conditions

The energy functional (6.10) is to be minimized over the set of {0,1}-valued sequences whose sum is equal to m. Let $\hat{Q}$ be a local optimizer of (6.10) (hence a global optimizer by convexity in the p-coordinates). At an optimizer, the energy cannot decrease if two entries of $\hat{Q}$ are swapped, i.e., if the values of Q_ℓ and Q_m are swapped for any ℓ and m. In particular, if $Q_\ell = 1$ and $Q_m = 0$ for some ℓ and

m (necessarily distinct), we have

$$E(\hat{Q} + \delta_m - \delta_\ell) \geq E(\hat{Q}) \tag{6.12}$$

where we use δ_j to denote the sequence consisting of zeros except a one in location j, i.e.

$$\delta_{jk} = \begin{cases} 1 & \text{if } k = j \\ 0 & \text{if } k \neq j \quad . \end{cases} \tag{6.13}$$

Computing the left-hand side of (6.12) using (6.10), we obtain

$$\begin{aligned}
E(\hat{Q}) &\leq E(\hat{Q} + \delta_m - \delta_\ell) && (6.14)\\
&= \Big\langle Q + \delta_m - \delta_\ell, |\cdot| * (w - \frac{1}{2}Q + \frac{\delta_\ell}{2} - \frac{\delta_m}{2}) \Big\rangle && (6.15)\\
&= E(\hat{Q}) + \Big\langle \delta_m - \delta_\ell, |\cdot| * (w - \frac{1}{2}Q) \Big\rangle + \Big\langle \delta_\ell, |\cdot| * \frac{\delta_m}{2} \Big\rangle && (6.16)\\
&\quad + \Big\langle \delta_m, |\cdot| * \frac{\delta_\ell}{2} \Big\rangle + \Big\langle Q, |\cdot| * (\frac{\delta_\ell}{2} - \frac{\delta_m}{2}) \Big\rangle && (6.17)\\
&= E(\hat{Q}) + \Big\langle |\cdot - m| - |\cdot - \ell|, q - \frac{Q}{2} \Big\rangle + |\ell - m| && (6.18)\\
&\quad + \Big\langle \frac{Q}{2}, |\cdot - \ell| \Big\rangle - \Big\langle Q, \frac{|\cdot - m|}{2} \Big\rangle && (6.19)\\
&= E(\hat{Q}) + \Big\langle w - Q, |\cdot - m| - |\cdot - \ell| \Big\rangle + |\ell - m| && (6.20)
\end{aligned}$$

where in the second equality we used the fact that $\langle \delta_\ell, |\cdot| * \frac{\delta_\ell}{2} \rangle = \langle \delta_m, |\cdot| * \frac{\delta_m}{2} \rangle = 0$. The equation (6.20) is only valid for all ℓ and m with $Q(\ell) = 1$ and $Q(m) = 0$; however by swapping the variables ℓ and m, it easily generalizes to the following necessary condition for optimality which applies to all ℓ and m:

$$\Big(Q_\ell - Q_m\Big)\Big\langle w - Q, |\cdot - m| - |\cdot - \ell| \Big\rangle + \Big|\ell - m\Big| \geq 0 \quad \forall \ell, \forall m. \tag{6.21}$$

(Note that by modifying more than 2 pixel values at a time a second order necessary condition can also be obtained with similar arguments.)

Let $\ell, m \in \mathbb{N}$ be such that $\hat{Q}_\ell = 1$, $\hat{Q}_m = 0$ and $m = \ell - 1$. Then we have from (6.21)

$$-\sum_{j=1}^{\ell-1}(w_j - \hat{Q}_j) + \sum_{\ell}^{n}(w_j - \hat{Q}_j) + 1 \geq 0. \tag{6.22}$$

Since we have $\sum_{j=1}^{n}(w_j - \hat{Q}_j) = 0$ and $w_\ell - \hat{Q}_\ell = w_\ell - 1 \leq 0$ in (6.22), we obtain

$$\sum_{j=1}^{\ell}(w_j - \hat{Q}_j) \leq \sum_{j=1}^{\ell-1}(w_j - \hat{Q}_j) \leq \frac{1}{2}. \tag{6.23}$$

Similarly, for ℓ, m satisfying $\hat{Q}_\ell = 1$, $\hat{Q}_m = 0$ and $m = \ell + 1$, we get

$$-\frac{1}{2} \leq \sum_{j=1}^{\ell}(w_j - \hat{Q}_j). \tag{6.24}$$

Let $\ell_2 \geq \ell_1 \geq 1$ be integers such that $\hat{Q}_s = 1$ for $s \in [\ell_1, \ell_2]$, $Q_{\ell_1 - 1} = 0$ and $\hat{Q}_{\ell_2+1} = 0$. Applying (6.23) with $\ell = \ell_1$ and (6.24) with $\ell = \ell_2$ we obtain

$$-\frac{1}{2} \leq \sum_{j=1}^{l_2}(w_j - \hat{Q}_j) \leq \sum_{j=1}^{s}(w_j - \hat{Q}_j) \leq \sum_{j=1}^{\ell_1}(w_j - \hat{Q}_j) \leq \frac{1}{2} \tag{6.25}$$

where in the second and third inequality above we used the fact that $w_s - \hat{Q}_s \leq 0$ for $s \in [l_1, l_2]$. The upper and lower bounds obtained in (6.25) for $\sum_{j=1}^{s}(w_j - \hat{Q}_j)$ makes the assumption that $\hat{Q}_s = 1$. The following proposition shows that this assumption can actually be removed.

Proposition 6.2.1. *Let $\hat{Q}$ be an optimizer of the energy functional* (6.10)*. For*

all $k \geq 1$, we have the upper and lower bounds

$$-\frac{1}{2} \leq \sum_{j=1}^{k}(w_j - \hat{Q}_j) \leq \frac{1}{2}.$$

Proof. Let $k \geq 1$ be given. If $\hat{Q}_k = 1$, then the proof follows from (6.25). Otherwise, we have $\hat{Q}_k = 0$. The main idea is to look for integers $\underline{\ell}$, $\overline{\ell}$ close to k to which inequalities (6.25) apply. Let $b_k := \sum_{j=1}^{k} \hat{Q}_j$, the cumulative sum of $\hat{Q}$. It is clear that $0 \leq b_k \leq m$. There are three cases:

(1) $0 < b_k < m$: There exist positive integers $\underline{\ell}$ and $\overline{\ell}$ where $\underline{\ell}$ is the largest integer such that $\underline{\ell} < k$ and $\hat{Q}_{\underline{\ell}} = 1$ and $\overline{\ell}$ is the smallest integer such that $\overline{\ell} > k$ and $\hat{Q}_{\overline{\ell}} = 1$. We have also $b_{\underline{\ell}} = b_k$ and $b_{\overline{\ell}} = b_k + 1$. By an application of (6.24) with $\ell = \underline{\ell}$, we get

$$-\frac{1}{2} \leq \sum_{j=1}^{\underline{\ell}}(w_j - \hat{Q}_j) \leq \big(\sum_{j=1}^{k} w_j\big) - b_{\underline{\ell}} = \sum_{j=1}^{k}(w_j - \hat{Q}_j).$$

Similarly, choosing $\ell = \overline{\ell}$ in (6.23) we get

$$\sum_{j=1}^{k}(w_j - \hat{Q}_j) \leq \sum_{j=1}^{\overline{\ell}-1}(w_j - \hat{Q}_j) \leq \frac{1}{2}.$$

(2) $b_k = 0$: Proving the lower bound is straightforward, since we have

$$\sum_{j=1}^{k}(w_j - \hat{Q}_j) = \sum_{j=1}^{k} w_j \geq 0.$$

For the upper bound, it suffices to apply (6.23) with $l = \overline{\ell}$ where $\overline{\ell}$ is the smallest positive integer (larger than k) such that $Q_{\overline{\ell}} = 1$.

(3) $b_k = m$: The upper bound follows from

$$\sum_{j=1}^{k}(w_j - \hat{Q}_j) = \Big(\sum_{j=1}^{k} w_j\Big) - m \leq 0$$

where we used $\sum_j w_j = m$. The lower bound follows from similar arguments.

□

Proposition 6.2.1 provides a connection between the energy functional and the sigma-delta ($\Sigma\Delta$) quantization as we shall explain. Given an input sequence $\{w_j\}_{j=1}^{\infty}$ with values in $[0, 1]$, the first-order $\Sigma\Delta$ scheme constructs a binary sequence $\tilde{Q}$ such that

$$\sup_{n_1,n_2\in\mathbb{N}^+} \Big| \sum_{n_1}^{n_2}(w_j - \tilde{Q}_j)\Big| \leq M \tag{6.26}$$

for some constant M. One way to achieve this bound is to look for a bounded solution u of the difference equation

$$u_k - u_{k-1} = w_k - \tilde{Q}_k, \quad u_0 = 0 \tag{6.27}$$

so that

$$u_k = \sum_{j=1}^{k}(w_j - \tilde{Q}_j) \tag{6.28}$$

and hence the supremum in (6.26) stays bounded uniformly over k. The initial value u_0 in (6.27) can be taken arbitrarily, but we set it to zero for convenience. A standard quantization rule that leads to a bounded solution u is given by the *greedy rule* which minimizes $|u_k|$ given u_{k-1} and w_k, i.e.

$$\tilde{Q}_k = \arg\min_{b\in\{0,1\}} |u_{k-1} + w_k - b|.$$

Proposition 6.2.1 shows that a global optimizer $\hat{Q}$ (of the energy functional (6.10)) satisfies (6.26) and so it corresponds to the output of a first-order $\Sigma\Delta$ scheme for the input signal $\{w_j\}_{j=1}^{\infty}$. The following proposition complements this result by showing that one-bit sigma-delta quantization with greedy rule can be formulated as an optimization problem where the minimization objective coincides with the energy (6.10) (up to an additive constant).

Proposition 6.2.2. *Let u be a bounded solution of the difference equation* (6.27)*. The energy functional* (6.10) *can be rewritten as*

$$E(Q) = \sum_{k=1}^{n} u_k^2 + \text{constant}(w, n) \tag{6.29}$$

and is minimized in a first-order $\Sigma\Delta$ scheme with the greedy rule.

Proof. The greedy rule minimizes $|u_k|$ given u_{k-1} and w_k for all k. Furthermore, by induction and using (6.28), it is easy to see that it minimizes $|u_k|$ for all k over all $\{Q_j\}_{j=1}^{n}$. Thus, the energy functional

$$\bar{E}(Q) := \sum_{k=1}^{n} u_k^2 = \sum_{k=1}^{n} \Big(\sum_{j=1}^{k} (w_j - Q_j) \Big)^2 \tag{6.30}$$

is minimized in a first-order $\Sigma\Delta$ scheme. We will show that $\bar{E}(Q)$ is equal to $E(Q)$ up to a constant. We compute

$$\begin{aligned} \bar{E}(Q) &= \sum_{k=1}^{n} \Big(\sum_{j=1}^{k} (w_j - Q_j) \Big)^2 \\ &= \sum_{k=1}^{n} \sum_{i=1}^{k} \sum_{j=1}^{k} w_i w_j + \sum_{k=1}^{n} \sum_{i=1}^{k} \sum_{j=1}^{k} Q_i Q_j - 2 \sum_{k=1}^{n} \sum_{i=1}^{k} \sum_{j=1}^{k} w_j Q_i \end{aligned}$$

We have

$$\begin{aligned}
\sum_{k=1}^{n}\sum_{i=1}^{k}\sum_{j=1}^{k} Q_iQ_j &= \sum_{i=1}^{n}\sum_{j=1}^{n}(n-\max(i,j)+1)Q_iQ_j \\
&= (n+1)\sum_{i=1}^{n}\sum_{j=1}^{n} Q_iQ_j - \sum_{i=1}^{n}\sum_{j=1}^{n}\max(i,j)Q_iQ_j \\
&= (n+1)m^2 - \sum_{k=1}^{m}\sum_{l=1}^{m}\max(p_k,p_l) \\
&= (n+1)m^2 - \sum_{k=1}^{m}\sum_{l=1}^{m}\Big(\frac{p_k+p_l}{2}+\frac{|p_k-p_l|}{2}\Big) \\
&= (n+1)m^2 - \sum_{k=1}^{m}\sum_{l=1}^{m}\Big(\frac{p_k+p_l}{2}\Big) - \sum_{k=1}^{m}\sum_{l=k+1}^{m}|p_k-p_l| \\
&= (n+1)m^2 - \sum_{k=1}^{m}\sum_{l=1}^{m}p_k - \sum_{k=1}^{m}\sum_{l=k+1}^{m}|p_k-p_l| \\
&= (n+1)m^2 - m\sum_{k=1}^{m}p_k - \sum_{k=1}^{m}\sum_{l=k+1}^{m}|p_k-p_l|
\end{aligned}$$

where we used $\sum_i Q_i = m$, a condition that is automatically satisfied in $\Sigma\Delta$ given that $\sum_j w_j = m$. Similarly, we have

$$\begin{aligned}
\sum_{k=1}^{n}\sum_{i=1}^{k}\sum_{j=1}^{k} w_jQ_i &= \sum_{i=1}^{n}\sum_{j=1}^{n}(n-\max(i,j)+1)w_jQ_i \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n}(n+1)w_jQ_i - \sum_{i=1}^{n}\sum_{j=1}^{n}\max(i,j)w_jQ_i \\
&= (n+1)m^2 - \sum_{k=1}^{m}\sum_{j=1}^{n}\max(p_k,j)w_j \\
&= (n+1)m^2 - \sum_{k=1}^{m}\sum_{j=1}^{n}\Big(\frac{p_k+j}{2}+\frac{|p_k-j|}{2}\Big)w_j \\
&= (n+1)m^2 - \frac{m}{2}\sum_{k=1}^{m}p_k - \frac{m}{2}\sum_{j=1}^{n}jw_j - \sum_{k=1}^{m}\sum_{j=1}^{n}\frac{|p_k-j|}{2}w_j
\end{aligned}$$

where we have used again the fact that $\sum_i Q_i = \sum_j w_j = m$. Hence,

$$\begin{aligned}
\bar{E}(Q) &= \sum_{k=1}^{n}\sum_{i=1}^{k}\sum_{j=1}^{k} w_i w_j + \sum_{k=1}^{n}\sum_{i=1}^{k}\sum_{j=1}^{k} Q_i Q_j - 2\sum_{k=1}^{n}\sum_{i=1}^{k}\sum_{j=1}^{k} w_j Q_i \quad (6.31)\\
&= \sum_{k=1}^{m}\sum_{j=1}^{n} |p_k - j| w_j - \sum_{k=1}^{m}\sum_{l=k+1}^{m} |p_k - p_l| + \text{constant}(w, n) \quad (6.32)
\end{aligned}$$

is the same as the energy functional (6.7) (which is equivalent to (6.10)) up to an additive constant that depends on w and n. This completes the proof. □

Remark 6.2.3. *Computing $\bar{E}(Q)$ using matrix notation is an alternative. Let S be the operator that maps a sequence to its cumulative sums, i.e.,*

$$(Sz)_j := \sum_{m=1}^{j} z_m.$$

The matrix representation of S is given by $S_{ij} = 1$ if $i \geq j$, $S_{ij} = 0$ otherwise. We compute

$$\begin{aligned}
\bar{E}(Q) &= ||S(w - Q)||_2^2 \quad (6.33)\\
&= \langle S(w - Q), S(w - Q)\rangle \quad (6.34)\\
&= \langle Sw, Sw\rangle + \langle SQ, SQ\rangle - 2\langle SQ, Sw\rangle. \quad (6.35)
\end{aligned}$$

We have also

$$\begin{aligned}
\langle SQ, SQ\rangle &= \langle \sum_{k=1}^{m} S(:,p_k), \sum_{l=1}^{m} S(:,p_l)\rangle && (6.36)\\
&= \sum_{k=1}^{m}\sum_{l=1}^{m} \langle S(:,p_k), S(:,p_l)\rangle && (6.37)\\
&= \sum_{k=1}^{m}\sum_{l=1}^{m}\sum_{j=1}^{n} S(j,p_k)S(j,p_l) && (6.38)\\
&= \sum_{k=1}^{m}\sum_{l=1}^{m} (n - \max(p_k,p_l) + 1) && (6.39)
\end{aligned}$$

where we used the MATLAB *notation* $S(:,j)$ *to refer to* j*th column of* S*. Similarly,*

$$\begin{aligned}
\langle SQ, Sw\rangle &= \langle \sum_{k=1}^{m} S(:,p_k), \sum_{l=1}^{n} S(:,l)w_l\rangle && (6.40)\\
&= \sum_{k=1}^{m}\sum_{l=1}^{n} \langle S(:,p_k), S(:,l)w_l\rangle && (6.41)\\
&= \sum_{k=1}^{m}\sum_{l=1}^{n}\sum_{j=1}^{n} S(j,p_k)S(j,l)w_l && (6.42)\\
&= \sum_{k=1}^{m}\sum_{l=1}^{n} (n - \max(p_k,l) + 1)w_l. && (6.43)
\end{aligned}$$

Plugging (6.39) *and* (6.43) *into* (6.35) *and using the identities* $\sum_j w_j = m$ *and* $\max(i,j) = \frac{i+j}{2} + \frac{|i-j|}{2}$*, we get the formula* (6.32).

6.3 Conclusion of the chapter

We have shown the equivalence between an optimization-based halftoning approach and first-order $\Sigma\Delta$ quantization. Any minimizer of the energy (6.1) in one dimension, corresponds to the output of a first-order $\Sigma\Delta$ scheme which is a dy-

namical system. Hence, halftoning can be performed in linear time (in the number of pixels). In addition, we showed that a first-order $\Sigma\Delta$ scheme with greedy rule minimizes the same energy.

In two dimensions, simple examples show that an optimizer cannot be the output of a recursive dynamical system such as a first-order $\Sigma\Delta$ quantizer due to the interactions between the halftoning dots; however it might be possible to approximate the optimizer(s) with a dynamical system. Sticking to $\varphi(s) = \log(s)$ seems to be a better choice in search for the approximate equivalence since then the interactions between the distant halftoning dots would be much smaller in magnitude as explained in Section 6.1. This is a subject for future work. In addition, applying higher-order sigma-delta quantization techniques to the halftoning of images is currently under investigation.

Bibliography

[ABBO11] R. Alam, S. Bora, R. Byers, and M.L. Overton. Characterization and construction of the nearest defective matrix via coalescence of pseudospectral components. *Linear Algebra and its Applications*, 435:494–513, 2011.

[ABJ75] B. Anderson, N. Bose, and E. Jury. Output feedback stabilization and related problems—solution via decision methods. *IEEE Transactions on Automatic Control*, 20(1):53–66, 1975.

[ALP94] O. Axelsson, H. Lu, and B. Polman. On the numerical radius of matrices and its application to iterative solution methods. *Linear and Multilinear Algebra*, 37(1-3):225–238, 1994. Special Issue: The numerical range and numerical radius.

[AM07] P.J. Antsaklis and A.N. Michel. *A linear systems primer*. Birkhäuser Boston Inc., Boston, MA, 2007.

[Bar93] B.R. Barmish. *New tools for robustness of linear systems*. Macmilllan, New York, 1993.

[BB90a] 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):1–7, 1990.

[BB90b] S. Boyd and V. Balakrishnan. A regularity result for the singular values of a transfer matrix and a quadratically convergent algorithm for computing its L_∞-norm. *Systems and Control Letters*, 15:1–7, 1990.

[BGL95] V. Blondel, M. Gevers, and A. Lindquist. Survey on the state of systems and control. *European Journal of Control*, 1:5–23, 1995.

[BGMO] V.D. Blondel, M. Gürbüzbalaban, A. Megretski, and M.L. Overton. Explicit solutions for root optimization of a polynomial family with one affine constraint. *IEEE Transactions on Automatic Control*. To appear.

[BGMO10] V.D. Blondel, M. Gürbüzbalaban, A. Megretski, and M.L. Overton. Explicit solutions for root optimization of a polynomial family. In *Proceedings of the 49th IEEE Conference on Decision and Control*, pages 485–488. IEEE, 2010.

[BHLO06a] J.V. Burke, D. Henrion, A.S. Lewis, and M.L. Overton. HIFOO - a MATLAB package for fixed-order controller design and H_∞ optimization. In *Fifth IFAC Symposium on Robust Control Design, Toulouse*, 2006.

[BHLO06b] J.V. Burke, D. Henrion, A.S. Lewis, and M.L. Overton. Stabilization via nonsmooth, nonconvex optimization. *IEEE Transactions on Automatic Control*, 51:1760–1769, 2006.

[BL05] J.M. Borwein and A.S. Lewis. *Convex Analysis and Nonlinear Optimization: Theory and Examples*. Springer, New York, second edition, 2005.

[BL10] Lyonell Boulton and Peter Lancaster. Schur decompositions of a matrix and the boundary of its pseudospectrum. *SIAM J. Matrix Anal. Appl.*, 31(5):2921–2933, 2010.

[Blo94] V. Blondel. *Simultaneous Stabilization of Linear Systems*. Lecture Notes in Control and Information Sciences 191. Springer, Berlin, 1994.

[BLO01] J.V. Burke, A.S. Lewis, and M.L. Overton. Optimizing matrix stability. *Proceedings of the American Mathematical Society*, 129:1635–1642, 2001.

[BLO02] J.V. Burke, A.S. Lewis, and M.L. Overton. Approximating subdifferentials by random sampling of gradients. *Math. Oper. Res.*, 27:567–584, 2002.

[BLO03a] J. V. Burke, A. S. Lewis, and M. L. Overton. Robust stability and a criss-cross algorithm for pseudospectra. *IMA J. Numer. Anal.*, 23(3):359–375, 2003.

[BLO03b] J.V. Burke, A.S. Lewis, and M.L. Overton. Optimization and pseudospectra, with applications to robust stability. *SIAM Journal on*

Matrix Analysis and Applications, 25:80–104, 2003. Corrigendum: http://www.cs.nyu.edu/overton/papers/pseudo_corrigendum.html.

[BLO04] J.V. Burke, A.S. Lewis, and M.L. Overton. Variational analysis of the abscissa mapping for polynomials via the Gauss-Lucas theorem. *Journal of Global Optimization*, 28:259–268, 2004.

[BLO05] J.V. Burke, A.S. Lewis, and M.L. Overton. A robust gradient sampling algorithm for nonsmooth, nonconvex optimization. *SIAM Journal on Optimization*, 15:751–779, 2005.

[BS90] N.A. Bruinsma and M. Steinbuch. A fast algorithm to compute the H^∞-norm of a transfer function matrix. *Systems Control Letters*, 14:287–293, 1990.

[BT95] V. Blondel and J.N. Tsitsiklis. NP-hardness of some linear control design problems. In *Proceedings of the 34th IEEE Conference on Decision and Control (CDC)*, volume 3, pages 2910–2915. IEEE, 1995.

[Bye88] R. Byers. A bisection method for measuring the distance of a stable matrix to the unstable matrices. *SIAM Journal on Scientific and Statistical Computing*, 9:875–881, 1988.

[CGT11] C. Cartis, N. Gould, and P.L. Toint. A note about the complexity of minimizing Nesterov's smooth Chebyshev-Rosenbrock function. Technical Report naXys-20-2011, Namur Center for Complex Systems, University of Namur, Belgium, 2011.

[CH99] S.H. Cheng and N.J. Higham. The nearest definite pair for the Hermitian generalized eigenvalue problem. *Linear Algebra and Its Applications*, 302:63–76, 1999.

[Che79a] R. Chen. On the problem of direct output feedback stabilization. In *Proceedings of MTNS*, pages 412–414, 1979.

[Che79b] R. Chen. *Output Feedback Stabilization of Linear Systems*. PhD thesis, University of Florida, 1979.

[Cla83] F. H. Clarke. *Optimization and Nonsmooth Analysis*. John Wiley, New York, 1983. Reprinted by SIAM, Philadelphia, 1990.

[Dai02] Y.-H. Dai. Convergence properties of the BFGS algorithm. *SIAM Journal on Optimization*, 13:693–701, 2002.

[Die57] P. Dienes. *The Taylor series: an introduction to the theory of functions of a complex variable*. Dover Publications Inc., New York, 1957.

[DKK83] D.W. Decker, H.B. Keller, and C.T. Kelley. Convergence rates for Newton's method at singular points. *SIAM Journal on Numerical Analysis*, 20(2):296–314, 1983.

[EG92] L. C. Evans and R. F. Gariepy. *Measure Theory and Fine Properties of Functions*. Studies in Advanced Mathematics. CRC Press, Boca Raton, FL, 1992.

[Eie93] M. Eiermann. Fields of values and iterative methods. *Linear Algebra and its Applications*, 180:167–197, 1993.

[Flo76] R.W. Floyd. An adaptive algorithm for spatial gray-scale. In *Proc. Soc. Inf. Disp.*, volume 17, pages 75–77, 1976.

[FM78] A. T. Fam and J. S. Meditch. A canonical parameter space for linear systems design. *IEEE Transactions on Automatic Control*, 23(3):454–458, 1978.

[FS11] M.A. Freitag and A. Spence. A newton-based method for the calculation of the distance to instability. *Linear Algebra and its Applications*, 435(12):3189 – 3205, 2011.

[GAB08] S. Gugercin, A. C. Antoulas, and C. Beattie. H_2 model reduction for large-scale linear dynamical systems. *SIAM J. Matrix Anal. Appl.*, 30(2):609–638, 2008.

[GDV98] Y. Genin, P. Van Dooren, and V. Vermaut. Convergence of the calculation of H_∞-norms and related questions. In *Proceedings of MTNS*, pages 429–432, July 1998.

[GG12] S. Güntürk and M. Gürbüzbalaban. On the equivalence between an optimization-based variational halftoning algorithm and sigma-delta quantization. 2012. In preparation.

[GGO] N. Guglielmi, M. Gürbüzbalaban, and M.L. Overton. Fast approximation of the H_∞ norm via optimization over spectral value sets. In preparation.

[GLR82] I. Gohberg, P. Lancaster, and L. Rodman. *Matrix Polynomials*. Academic Press Inc. [Harcourt Brace Jovanovich Publishers], New York, 1982. Computer Science and Applied Mathematics.

[GMW81] P.E. Gill, W. Murray, and M.H. Wright. *Practical Optimization*. Academic Press, New York and London, 1981.

[GO07] K. K. Gade and M. L. Overton. Optimizing the asymptotic convergence rate of the Diaconis-Holmes-Neal sampler. *Adv. in Appl. Math.*, 38(3):382–403, 2007.

[GO11] N. Guglielmi and M.L. Overton. Fast algorithms for the approximation of the pseudospectral abscissa and pseudospectral radius of a matrix. *SIAM Journal on Matrix Analysis and Applications*, 32(4):1166–1192, 2011.

[GO12a] M. Gürbüzbalaban and M.L. Overton. On Nesterov's nonsmooth Chebyshev-Rosenbrock functions. *Nonlinear Analysis: Theory, Methods and Applications*, 75(3):1282 – 1289, 2012.

[GO12b] M. Gürbüzbalaban and M.L. Overton. Some regularity results for the pseudospectral abscissa and pseudospectral radius of a matrix. *SIAM Journal on Optimization*, 22(2):281–285, 2012.

[GR84] A. Griewank and G. W. Reddien. Characterization and computation of generalized turning points. *SIAM J. Numer. Anal.*, 21(1):176–185, 1984.

[Gri80] A. O. Griewank. Starlike domains of convergence for Newton's method at singularities. *Numerische Mathematik*, 35:95–111, 1980.

[GT82] M. Goldberg and E. Tadmor. On the numerical radius and its applications. *Linear Algebra Appl*, 42:263–284, 1982.

[GV83] G.H. Golub and C. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, 1983.

[GVL96] G.H. Golub and C.F. Van Loan. *Matrix computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, third edition, 1996.

[Hif] HIFOO (H-infinity fixed-order optimization). http://www.cs.nyu.edu/overton/software/hifoo/.

[Hig88] N.J. Higham. Computing a nearest symmetric positive semidefinite matrix. *Linear Algebra and its Applications*, 103(0):103 – 118, 1988.

[HJ90] R.A. Horn and C.R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, 1990. Corrected reprint of the 1985 original.

[HO06] D. Henrion and M. L. Overton. Maximizing the closed loop asymptotic decay rate for the two-mass-spring control problem. Technical Report 06342, LAAS-CNRS, March 2006.
http://homepages.laas.fr/henrion/Papers/massspring.pdf.

[HP05] D. Hinrichsen and A.J. Pritchard. *Mathematical Systems Theory I: Modelling, State Space Analysis, Stability and Robustness*. Springer, Berlin, Heidelberg and New York, 2005.

[HS89] D. Hinrichsen and N. K. Son. The complex stability radius of discrete-time systems and symplectic pencils. In *Proceedings of the 28th IEEE Conference on Decision and Control, Vol. 1–3 (Tampa, FL, 1989)*, pages 2265–2270, New York, 1989. IEEE.

[HTVD02] N.J. Higham, F. Tisseur, and P.M. Van Dooren. Detecting a definite Hermitian pair and a hyperbolic or elliptic quadratic eigenvalue problem, and associated nearness problems. *Linear Algebra and its applications*, 351:455–474, 2002.

[HUL93] J.B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms*. Springer-Verlag, New York, 1993. Two volumes.

[HW97] C. He and G. A. Watson. An algorithm for computing the numerical radius. *IMA J. Numer. Anal.*, 17(3):329–342, 1997.

[HW98] 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.

[Jar] F. Jarre. On Nesterov's smooth Chebyshev–Rosenbrock function. *Optimization Methods and Software*. To appear. DOI: 10.1080/10556788.2011.638924.

[Kak11] A. Kaku. Implementation of high precision arithmetic in the BFGS method for nonsmooth optimization. Master's thesis, NYU, Jan 2011. http://www.cs.nyu.edu/overton/mstheses/kaku/msthesis.pdf.

[Kar03] M. Karow. *Geometry of Spectral Value Sets*. PhD thesis, Universität Bremen, 2003.

[Kat82] T. Kato. *A Short Introduction to Perturbation Theory for Linear Operators*. Springer-Verlag, New York, 1982.

[Kel77] H.B. Keller. Numerical solution of bifurcation and nonlinear eigenvalue problems. In *Applications of bifurcation theory (Proc. Advanced Sem., Univ. Wisconsin, Madison, Wis., 1976)*, pages 359–384. Publ. Math. Res. Center, No. 38. Academic Press, New York, 1977.

[Kel08] J.B. Keller. Multiple eigenvalues. *Linear Algebra Appl.*, 429(8-9):2209–2220, 2008.

[Kiw85] K.C. Kiwiel. *Methods of Descent for Nondifferentiable Optimization.* Lecture Notes in Mathematics 1133. Springer-Verlag, Berlin and New York, 1985.

[Kiw08] K.C. Kiwiel, 2008. Private communication.

[KV12] D. Kressner and B. Vandereycken. Subspace methods for computing the pseudospectral abscissa and the stability radius. Technical report, MATHICSE Nr. 13.2012, École Polytechnique Fédérale de Lausanne, Switzerland, March 2012.

[Lei06] F. Leibfritz. Compleib: Constrained matrix optimization problem library. 2006.

[Lew03] A.S. Lewis. Active sets, nonsmoothness and sensitivity. *SIAM Journal on Optimization*, 13:702–725, 2003.

[Lew06] A.S. Lewis. Eigenvalues and nonsmooth optimization. In *Foundations of Computational Mathematics, Santander 2005*, volume 331 of *London Math. Soc. Lecture Note Ser.*, pages 208–229. Cambridge Univ. Press, Cambridge, 2006.

[LF01] D.-H. Li and M. Fukushima. On the global convergence of the BFGS method for nonconvex unconstrained optimization problems. *SIAM Journal on Optimization*, 11:1054–1064, 2001.

[LO] A.S. Lewis and M.L. Overton. Nonsmooth optimization via quasi-Newton methods. *Mathematical Programming*. To Appear.

[LP08] A.S. Lewis and C.H.J. Pang. Variational analysis of pseudospectra. *SIAM Journal on Optimization*, 19:1048–1072, 2008.

[LSY98] R. B. Lehoucq, D. C. Sorensen, and C. Yang. *ARPACK users' guide*, volume 6 of *Software, Environments, and Tools*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1998. Solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods.

[LV07] F. Leibfritz and S. Volkwein. Numerical feedback controller design for PDE systems using model reduction: techniques and case studies. In L.T. Biegler, O. Ghattas, M. Heinkenschloss, D. Keyes, and B. van Bloemen Waanders, editors, *Real-time PDE-constrained optimization*, Comput. Sci. Eng. 3. SIAM, Philadelphia, PA, 2007.

[Mar66] M. Marden. *Geometry of Polynomials*. American Mathematical Society, 1966.

[Mas04] W.F. Mascarenhas. The BFGS method with exact line searches fails for non-convex objective functions. *Mathematical Programming*, 99:49–61, 2004.

[Mat] MATLAB Control System Toolbox. http://www.mathworks.com.

[Men06] E. Mengi. *Measures for Robust Stability and Controllability*. PhD thesis, Courant Institute of Mathematical Sciences, New York, NY, 2006.

[MO05] E. Mengi and M.L. Overton. Algorithms for the computation of the pseudospectral radius and the numerical radius of a matrix. *IMA Journal on Numerical Analysis*, 25:648–669, 2005.

[Mor76] B.S. Mordukhovich. Maximum principle in the problem of time optimal response with nonsmooth constraints. *Journal of Applied Mathematics and Mechanics*, 40:960–969, 1976.

[Nem93] A. Nemirovskii. Several NP-hard problems arising in robust stability analysis. *Math. Control Signals Systems*, 6:99–105, 1993.

[Nes08] Y. Nesterov, 2008. Private communication.

[NW06] J. Nocedal and S.J. Wright. *Nonlinear Optimization*. Springer, New York, second edition, 2006.

[PFTV86] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical recipes*. Cambridge University Press, Cambridge, 1986. The art of scientific computing.

[PS99] B.T. Polyak and P.S. Shcherbakov. Numerical search of stable or unstable element in matrix or polynomial families: a unified approach to robustness analysis and stabilization. In A.Garulli, A.Tesi, and A.Vicino, editors, *Robustness in Identification and Control*, Lect. Notes Control and Inform. Sci. 245, pages 344–358. Springer, London, 1999.

[PWM10] A.P. Popov, H. Werner, and M. Millstone. Fixed-structure discrete-time H_∞ controller synthesis with HIFOO. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 3152 –3155, Dec. 2010.

[Ran89] A. Rantzer. Equivalence between stability of partial realizations and feedback stabilization—applications to reduced order stabilization. *Linear Algebra Appl.*, 122/123/124:641–653, 1989.

[Rob89] G. Robel. On computing the infinity norm. *IEEE Transactions on Automatic Control*, 34(8):882–884, 1989.

[RW98] R.T. Rockafellar and R.J.B. Wets. *Variational Analysis*. Springer, New York, 1998.

[SMM92] J. T. Spanos, M. H. Milman, and D. L. Mingori. A new algorithm for L_2 optimal model reduction. *Automatica*, 28:897–909, September 1992.

[SP05] A. Spence and C. Poulton. Photonic band structure calculations using nonlinear eigenvalue techniques. *J. Comput. Phys.*, 204(1):65–81, 2005.

[SY06] W.-Y. Sun and Y.-X. Yuan. *Optimization Theory and Methods*, volume 1 of *Springer Optimization and Its Applications*. Springer, New York, 2006. Nonlinear programming.

[Sze39] G. Szegö. *Orthogonal Polynomials*. American Mathematical Society, New York, 1939. American Mathematical Society Colloquium Publications, v. 23.

[TE05] L. N. Trefethen and M. Embree. *Spectra and Pseudospectra: the Behavior of Nonnormal Matrices and Operators*. Princeton University Press, 2005.

[TSG+11] T. Teuber, G. Steidl, P. Gwosdek, C. Schmaltz, and J. Weickert. Dithering by differences of convex functions. *SIAM Journal on Imaging Sciences*, 4:79, 2011.

[Uli87] R. Ulichney. *Digital Halftoning*. Mit Press, 1987.

[VL85] C. Van Loan. How near is a stable matrix to an unstable matrix? In *Linear algebra and its role in systems theory (Brunswick, Maine, 1984)*, volume 47 of *Contemp. Math.*, pages 465–478. Amer. Math. Soc., Providence, RI, 1985.

[Wim91] H.K. Wimmer. Normal forms of symplectic pencils and the discrete-time algebraic Riccati equation. *Linear Algebra and its Applications*, 147:411–440, 1991.

[Wri02a] T.G. Wright. Eigtool: a graphical tool for nonsymmetric eigenproblems. *Oxford University Computing Laboratory*, 2002. http://www.comlab.ox.ac.uk/pseudospectra/eigtool/.

[Wri02b] T.G. Wright. EigTool: a graphical tool for nonsymmetric eigenproblems, 2002. Oxford University Computer Laboratory, http://web.comlab.ox.ac.uk/pseudospectra/eigtool/.

[ZGD95] K. Zhou, K. Glover, and J. Doyle. *Robust and Optimal Control*. Prentice Hall, 1995.