

# Numerical Optimization of Eigenvalues of Hermitian Matrix Functions

Mustafa Kılıç\*      Emre Mengi†      E. Alper Yıldırım‡

September 12, 2011

## Abstract

The unordered eigenvalues of a Hermitian matrix function depending on one parameter analytically is analytic. Ordering these eigenvalues yields piece-wise analytic functions. For multi-variate Hermitian matrix functions depending on  $d$  parameters analytically the ordered eigenvalues are piece-wise analytic along lines in the  $d$ -dimensional space. These classical results imply the boundedness of the second derivatives of the pieces defining the eigenvalue functions along any direction. We derive an algorithm based on the boundedness of these second derivatives for the global minimization of an eigenvalue of an analytic Hermitian matrix function. The algorithm is globally convergent. It determines the globally minimal value of a piece-wise quadratic under-estimator for the eigenvalue function repeatedly. In the multi-variate case determining this globally minimal value is equivalent to the solution of a quadratic program. The derivatives of the eigenvalue functions are used to construct quadratic models yielding rapid global convergence as compared to traditional global optimization algorithms. The applications that we have in mind include the  $H_\infty$  norm of a linear system, numerical radius, distance to uncontrollability and distance to the nearest defective matrix.

**Key words.** Hermitian eigenvalues, analytic, global optimization, perturbation of eigenvalues, quadratic program

**AMS subject classifications.** 65F15, 90C26

---

\*Department of Mathematics, Koç University, Rumelifeneri Yolu, 34450 Sarıyer-İstanbul, Turkey (mukilic@ku.edu.tr)

†Department of Mathematics, Koç University, Rumelifeneri Yolu, 34450 Sarıyer-İstanbul, Turkey (emengi@ku.edu.tr). The work of this author was supported in part by the European Commission grant PIRG-GA-268355 and the TÜBİTAK (the scientific and technological research council of Turkey) carrier grant 109T660

‡Department of Industrial Engineering, Koç University, Rumelifeneri Yolu, 34450 Sarıyer-İstanbul, Turkey (alperyildirim@ku.edu.tr). This author was supported in part by TÜBİTAK (Turkish Scientific and Technological Research Council) Grant 109M149 and by TÜBA-GEBİP (Turkish Academy of Sciences Young Scientists Award Program).

## 1 Introduction

The main object of this work is a matrix-valued function  $\mathcal{A}(\omega) : \mathbb{R}^d \rightarrow \mathbb{C}^{n \times n}$  that satisfies the following two properties.

- (i) Analyticity : The function  $\mathcal{A}(\omega)$  has a convergent power series representation for all  $\omega \in \mathbb{R}^d$ .
- (ii) Self-adjointness : The function  $\mathcal{A}(\omega)$  is Hermitian (*i.e.*  $\mathcal{A}(\omega) = \mathcal{A}^*(\omega)$ ) for all  $\omega \in \mathbb{R}^d$ .

Here we consider the *global* minimization or maximization of a prescribed eigenvalue  $\lambda(\omega)$  of  $\mathcal{A}(\omega)$  over  $\omega \in \mathbb{R}^d$  numerically. From an application point of view prescribed typically refers to the  $j$ th largest eigenvalue, *i.e.*,  $\lambda(\omega) := \lambda_j(\mathcal{A}(\omega))$ . But it might as well refer to a particular eigenvalue with respect to a different criterion as long as some (piece-wise) analyticity properties discussed below and in Section 2 hold. For instance, for a *univariate* Hermitian matrix function,  $\lambda(\omega)$  might be anyone of the  $n$  roots of the characteristic polynomial of  $\mathcal{A}(\omega)$  that varies analytically with respect to  $\omega$  as described next.

Let us start by considering the univariate case. The roots of the characteristic polynomial of  $\mathcal{A}(\omega) : \mathbb{R} \rightarrow \mathbb{C}^{n \times n}$  can be arranged as  $\tilde{\lambda}_1(\omega), \dots, \tilde{\lambda}_n(\omega)$  so that each  $\tilde{\lambda}_j(\omega)$  is analytic over  $\mathbb{R}$  for  $j = 1, \dots, n$ . Remarkably this analyticity property of eigenvalues of Hermitian matrix functions holds even if some of the eigenvalues repeat, that is even if  $\tilde{\lambda}_j(\omega) = \tilde{\lambda}_k(\omega)$  for  $j \neq k$ . On the other hand this analyticity property of eigenvalues is intrinsic to Hermitian matrix functions only, and is not true for non-Hermitian matrix functions. For instance analytic perturbations of order  $O(\epsilon)$  to the entries of an  $n \times n$  Jordan block may yield variations in eigenvalues proportional to  $O(\epsilon^{1/n})$ .

When the eigenvalues  $\tilde{\lambda}_1(\omega), \dots, \tilde{\lambda}_n(\omega)$  are ordered from largest to smallest, they are no longer analytic, but continuous and piece-wise analytic. What we exploit in this paper for the optimization of an eigenvalue  $\lambda(\omega)$  is the boundedness of the derivatives of the analytic pieces. Particularly there exists a constant  $\gamma$  satisfying

$$\left| \tilde{\lambda}_j''(\omega) \right| \leq \gamma \quad \forall \omega \in \mathbb{R}$$

for  $j = 1, \dots, n$ . The function  $\lambda(\omega)$  is non-convex possibly with many local extrema, so its global optimization cannot be achieved solely based on derivatives. For global optimization one must also benefit from global properties such as the upper bound on the second derivatives  $\tilde{\lambda}_j''(\omega)$ .

Contrary to the univariate case, the roots of the characteristic polynomial of a multivariate Hermitian matrix function  $\mathcal{A}(\omega) : \mathbb{R}^d \rightarrow \mathbb{C}^{n \times n}$  are not analytic no matter how they are ordered. But along any line in  $\mathbb{R}^d$  there is an ordering that makes eigenvalues analytic, and ordering them from largest to smallest makes them continuous and piece-wise analytic. The (piece-wise) analyticity of eigenvalues along lines in  $\mathbb{R}^d$  is our main tool for the optimization of an eigenvalue  $\lambda(\omega)$  in the multi-variate case.

To the authors' knowledge the most elementary examples that require the optimization of eigenvalues of Hermitian matrix functions are the distance to instability

$$\begin{aligned} \inf\{\|\Delta A\|_2 : x'(t) = (A + \Delta A)x(t) \text{ is unstable}\} &= \inf_{\omega \in \mathbb{R}} \sigma_n(A - \omega i I), \\ \inf\{\|\Delta A\|_2 : x_{k+1} = (A + \Delta A)x_k \text{ is unstable}\} &= \inf_{\theta \in [0, 2\pi)} \sigma_n(A - e^{i\theta} I) \end{aligned}$$

for continuous and discrete systems, respectively. Above  $\sigma_n$  denotes the smallest singular value. For instance to minimize  $\sigma_n(A - \omega iI)$  the associated eigenvalue function  $\lambda(\omega)$  is the smallest eigenvalue of

$$\begin{bmatrix} 0 & A - \omega iI \\ A^* + \omega iI & 0 \end{bmatrix}$$

in absolute value, which is continuous and piece-wise analytic with respect to  $\omega$ . Some other examples include the numerical radius of a matrix,  $H_\infty$  norm of a transfer function, distance to uncontrollability from a linear time-invariant dynamical system, distance to the nearest matrix with an eigenvalue of specified algebraic multiplicity (in particular distance to the nearest defective matrix), distance to the nearest pencil with eigenvalues in a specified region. The eigenvalue optimization problems associated with these problems are listed in the table below. In some cases the supremum of an eigenvalue needs to be minimized. In these examples  $\lambda(\omega)$  is the supremum of the eigenvalue, instead of the eigenvalue itself. Note that if  $f(x, y)$  with  $x \in \mathbb{R}^d, y \in \mathbb{R}$  depends on each  $x_j$  and  $y$  piece-wise analytically, jointly. Then  $g(x) = \sup_y f(x, y)$  is also a piece-wise analytic function of each  $x_j$ , jointly. Below  $\Omega^r$  denotes the  $r$ -tuples of  $\Omega \subseteq \mathbb{C}$ ,  $e_r$  denotes the vector of ones of size  $r$ , the  $j$ th largest singular value is denoted by  $\sigma_j$ , the notation  $\otimes$  is reserved for the Kronecker product, and

$$C(\Lambda, \Gamma) := \begin{bmatrix} \lambda_1 & \gamma_{12}I & & \gamma_{1r}I \\ 0 & \lambda_2 & & \gamma_{2r}I \\ & & \ddots & \\ & & & \lambda_{r-1} & \gamma_{(r-1)r}I \\ 0 & & & 0 & \lambda_r \end{bmatrix}$$

for  $\Lambda = [\lambda_1 \ \dots \ \lambda_r] \in \Omega^r$  and  $\Gamma = [\gamma_{12} \ \dots \ \gamma_{(r-1)r}] \in \mathbb{C}^{r(r-1)/2}$ .

Problem	Optimization Characterization
Numerical Radius	$\sup_{\theta \in [0, 2\pi)} \lambda_1 \left( \frac{Ae^{i\theta} + A^*e^{-i\theta}}{2} \right)$
$H_\infty$ -Norm of a Linear Time-invariant System $(A, B, C, D)$	$\sup_{\omega \in \mathbb{R}} \sigma_1 (C(\omega iI - A)^{-1}B + D)$
Distance from a Linear Time-Invariant System $(A, B)$ to Uncontrollability	$\inf_{z \in \mathbb{C}} \sigma_n ([A - zI \ B])$
Distance from $A$ to Nearest Matrix with an Eigenvalue of Multiplicity $\geq r$	$\inf_{\lambda \in \mathbb{C}} \sup_{\Gamma \in \mathbb{C}^{r(r-1)/2}} \sigma_{nr-r+1} (I \otimes A - C(\lambda e_r, \Gamma) \otimes I)$
Distance from $A - \lambda B$ to Nearest Pencil with $r$ eigenvalues in region $\Omega \subseteq \mathbb{C}$	$\inf_{\Lambda \in \Omega^r} \sup_{\Gamma \in \mathbb{C}^{r(r-1)/2}} \sigma_{nr-r+1} (I \otimes A - C(\Lambda, \Gamma) \otimes B)$

The distance to instability was first considered by Van Loan [34] in 1984, and used to analyze the transient behavior of the dynamical system  $x'(t) = Ax(t)$ . Various algorithms have been suggested [8, 19, 18, 14] since then for its numerical computation. Specifically Byers' work [8] inspired many other algorithms, each of which is tailored for a particular eigenvalue optimization problem. In control theory it is essential to compute the  $H_\infty$  norm of the transfer function of a dynamical system for various purposes, e.g., controller synthesis,

model reduction, etc. In two independent works Boyd and Balakrishnan [5], and Bruinsma and Steinbuch [6] extended the Byers' algorithm for the computation of the  $H_\infty$  norm of the transfer function of a linear dynamical system. The distance to uncontrollability was originally introduced by Paige [28], and the eigenvalue optimization characterization was provided in [12]. Various algorithms appeared in the literature for the computation of the distance to uncontrollability; see for instance [9, 15, 16, 17]. Malyshev derived an eigenvalue optimization characterization for the distance to the nearest defective matrix [25], but did not elaborate on how to solve it numerically. The second author generalized the Malyshev's characterization for the distance to the nearest matrix with an eigenvalue of specified multiplicity [26]. More recently the eigenvalue optimization characterization with many unknowns is deduced for the distance to a nearest pencil with eigenvalues lying in a specified region [23]; [25, 26] are special cases of [23]. This last problem has an application in signal processing, namely estimating the shape of a region given the moments over the region as suggested by Elad, Milanfar and Golub [13].

All of the aforementioned algorithms are devised for particular problems. The algorithm that we introduce and analyze here is the first generic algorithm for the optimization of eigenvalues of Hermitian matrix functions depending on its parameters analytically. The algorithm is based on piece-wise quadratic models lying underneath the eigenvalue function. Consequently, the algorithm here is a reminiscent of the Piyavskii-Shubert algorithm [29, 33], that is well-known by the global optimization community and based on constructing piece-wise linear approximations for a Lipschitz continuous function lying underneath the function. The Piyavskii-Shubert algorithm is derivative-free, and later sophisticated variants, attempting to estimate the Lipschitz constant locally, appeared in the literature [21, 32]. The algorithm here exploits the derivatives, and the use of quadratic under-estimators yields faster convergence. It is applicable for the optimization of other (piece-wise) analytic functions. But it is particularly well-suited for the optimization of eigenvalues. For an eigenvalue function, once the eigenvalue and the associated eigenvector are evaluated, its derivative is available without any other significant work due to analytic formulas; see the next section, in particular equation (3).

The outline of this paper is as follows. In the next section we review the basic results concerning the analyticity of the eigenvalues of a Hermitian matrix function  $\mathcal{A}(\omega)$  that depends on  $\omega$  analytically. These basic results are in essence due to Rellich [30]. Also in the next section we derive expressions for the first two derivatives of the unordered eigenvalue  $\tilde{\lambda}(\omega)$ . To our knowledge these expressions first appeared in a Numerische Mathematik paper by Lancaster [24]. Section 3 is devoted to the derivation of the one-dimensional algorithm. Section 4 extends the algorithm to the multi-variate case. Section 5 focuses on the analysis of the multi-dimensional algorithm; specifically it establishes that there are subsequences of the sequence generated by the algorithm that converge to global minimizers. Section 6 describes a practical variant of the multi-dimensional algorithm, that is based on mesh-refinement. The algorithm suggested here and its convergence analysis applies to the global optimization of any continuous and piece-wise analytic function. Finally Section 7 discusses how in particular the algorithm can be applied to the eigenvalue optimization problems listed above.

## 2 Background on Perturbation Theory of Eigenvalues

In this section we first briefly summarize the analyticity results, mostly borrowed from [30, Chapter 1], related to the eigenvalues of matrix functions. Then expressions are derived for the derivatives of Hermitian eigenvalues in terms of eigenvectors and the derivatives of matrix functions. Finally, we elaborate on the analyticity of singular value problems as special Hermitian eigenvalue problems.

### 2.1 Analyticity of Eigenvalues

#### 2.1.1 Univariate Matrix Functions

For a univariate matrix function  $\mathcal{A}(\omega)$ , Hermitian or not, but depending on  $\omega$  analytically, the characteristic polynomial is of the form

$$g(\omega, \lambda) := \det(\lambda I - \mathcal{A}(\omega)) = a_n(\omega)\lambda^n + \cdots + a_1(\omega)\lambda + a_0(\omega)$$

where  $a_0(\omega), \dots, a_n(\omega)$  are analytic functions of  $\omega$ . It follows from the Puiseux' theorem (see for instance [35, Chapter 2]) that each root  $\tilde{\lambda}_j(\omega)$  such that  $g(\omega, \tilde{\lambda}_j(\omega)) = 0$  has a Puiseux series of the form

$$\tilde{\lambda}_j(\omega) = \sum_{k=0}^{\infty} c_{k,j} \omega^{k/r} \quad (1)$$

for all small  $\omega$  where  $r$  is the multiplicity of the root  $\tilde{\lambda}_j(0)$ .

Now suppose  $\mathcal{A}(\omega)$  is Hermitian for all  $\omega$ , and let  $\ell$  be the smallest integer such that  $c_{\ell,j} \neq 0$ . Then we have

$$\lim_{\omega \rightarrow 0^+} \frac{\tilde{\lambda}_j(\omega) - \tilde{\lambda}_j(0)}{\omega^{\ell/r}} = c_{\ell,j}$$

implying  $c_{\ell,j}$  is real, since  $\tilde{\lambda}_j(\omega)$  for all  $\omega$  and  $\omega^{\ell/r}$  are real numbers. Furthermore

$$\lim_{\omega \rightarrow 0^-} \frac{\tilde{\lambda}_j(\omega) - \tilde{\lambda}_j(0)}{(-\omega)^{\ell/r}} = \frac{c_{\ell,j}}{(-1)^{\ell/r}}$$

is real implying  $(-1)^{\ell/r}$  is real, equivalently  $\ell/r$  is integer. This shows that the first nonzero term in the Puiseux series of  $\tilde{\lambda}_j(\omega)$  is an integer power of  $\omega$ . The same argument applied to the derivatives of  $\tilde{\lambda}_j(\omega)$  and the associated Puiseux series indicates that only integer powers of  $\omega$  can appear in the Puiseux series (1), that is the Puiseux series reduces to a power series. This establishes that  $\tilde{\lambda}_j(\omega)$  is an analytic function of  $\omega$ . Indeed it can also be deduced that associated with  $\tilde{\lambda}_j(\omega)$  there is a unit eigenvector  $v_j(\omega)$  that varies analytically with respect to  $\omega$  (see [30] for details).

**Theorem 2.1** (Rellich). *Let  $\mathcal{A}(\omega) : \mathbb{R} \rightarrow \mathbb{C}^{n \times n}$  be a Hermitian matrix function that depends on  $\omega$  analytically.*

- (i) *The  $n$  roots of the characteristic polynomial of  $\mathcal{A}(\omega)$  can be arranged so that each root  $\tilde{\lambda}_j(\omega)$  for  $j = 1, \dots, n$  is an analytic function of  $\omega$ .*
- (ii) *There exists an eigenvector  $v_j(\omega)$  associated with  $\tilde{\lambda}_j(\omega)$  for  $j = 1, \dots, n$  satisfying*

- (1)  $(\tilde{\lambda}_j(\omega)I - \mathcal{A}(\omega))v_j(\omega) = 0 \quad \forall \omega \in \mathbb{R}$ ,
- (2)  $\|v_j(\omega)\|_2 = 1 \quad \forall \omega \in \mathbb{R}$ ,
- (3)  $v_j^*(\omega)v_k(\omega) = 0 \quad \forall \omega \in \mathbb{R}$  for  $k \neq j$ , and
- (4)  $v_j(\omega)$  is an analytic function of  $\omega$ .

For non-Hermitian matrix functions, since the eigenvalue  $\tilde{\lambda}_j(\omega)$  is not real, the argument above fails. In this case in the Puiseux series (1) non-integer rational powers remain in general. For instance the roots of the  $n \times n$  Jordan block

$$\begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ & & & \ddots & 0 \\ & & & & 1 \\ \epsilon & 0 & & & 0 \end{bmatrix}$$

with the lower left-most entry perturbed by  $\epsilon$  are given by  $\sqrt[n]{(-1)^n \epsilon}$ .

### 2.1.2 Multivariate Matrix Functions

The eigenvalues of a multivariate matrix function  $\mathcal{A}(\omega) : \mathbb{R}^d \rightarrow \mathbb{C}^{n \times n}$  depending on  $\omega$  analytically do not have power series representations in general even when  $\mathcal{A}(\omega)$  is Hermitian. As an example the unordered eigenvalues of

$$\mathcal{A}(\omega) = \begin{bmatrix} \omega_1 & \frac{\omega_1 + \omega_2}{2} \\ \frac{\omega_1 + \omega_2}{2} & \omega_2 \end{bmatrix}$$

are  $\tilde{\lambda}_{1,2}(\omega) = \frac{\omega_1 + \omega_2}{2} \pm \sqrt{\frac{\omega_1^2 + \omega_2^2}{2}}$ . On the other hand it follows from Theorem 2.1 that along any line in  $\mathbb{R}^d$  the unordered eigenvalues  $\tilde{\lambda}_j(\omega)$ ,  $j = 1, \dots, n$  of  $\mathcal{A}(\omega)$  are analytic when  $\mathcal{A}(\omega)$  is Hermitian. This analyticity property along lines in  $\mathbb{R}^d$  implies the existence of the partial derivatives of  $\tilde{\lambda}_j(\omega)$  everywhere. Expressions for the partial derivatives will be derived in the next subsection indicating their continuity. As a consequence of the continuity of the partial derivatives, each unordered eigenvalue  $\tilde{\lambda}_j(\omega)$  must be differentiable.

**Theorem 2.2.** *Let  $\mathcal{A}(\omega) : \mathbb{R}^d \rightarrow \mathbb{C}^{n \times n}$  be a Hermitian matrix function that depends on  $\omega$  analytically. Then the  $n$  roots of the characteristic polynomial of  $\mathcal{A}(\omega)$  can be arranged so that each root  $\tilde{\lambda}_j(\omega)$  is (i) analytic on every line in  $\mathbb{R}^d$ , and (ii) differentiable on  $\mathbb{R}^d$ .*

## 2.2 Derivatives of Eigenvalues and Eigenvectors

### 2.2.1 First Derivatives of Eigenvalues

Consider a univariate Hermitian matrix-valued function  $\mathcal{A}(\omega)$  depending on  $\omega$  analytically. An unordered eigenvalue  $\tilde{\lambda}_j(\omega)$  and the associated eigenvector  $v_j(\omega)$  as described in Theorem 2.1 satisfies

$$\mathcal{A}(\omega)v_j(\omega) = \tilde{\lambda}_j(\omega)v_j(\omega).$$

Taking the derivatives of both sides we obtain

$$\frac{d\mathcal{A}(\omega)}{d\omega}v_j(\omega) + \mathcal{A}(\omega)\frac{dv_j(\omega)}{d\omega} = \frac{d\tilde{\lambda}_j(\omega)}{d\omega}v_j(\omega) + \tilde{\lambda}_j(\omega)\frac{dv_j(\omega)}{d\omega}. \quad (2)$$

Multiplying both sides by  $v_j(\omega)^*$  and using the identities  $v_j(\omega)^*\mathcal{A}(\omega) = v_j(\omega)^*\tilde{\lambda}_j(\omega)$  as well as  $v_j(\omega)^*v_j(\omega) = \|v_j(\omega)\|_2^2 = 1$  yield

$$\frac{d\tilde{\lambda}_j(\omega)}{d\omega} = v_j(\omega)^*\frac{d\mathcal{A}(\omega)}{d\omega}v_j(\omega). \quad (3)$$

### 2.2.2 First Derivatives of Eigenvectors

It is also possible to deduce the first derivative of  $v_j(\omega)$  from (2). First the identity  $v_j(\omega)^*v_j(\omega) = 1$  for all  $\omega$  implies

$$\frac{d(v_j(\omega)^*v_j(\omega))}{d\omega} = 0 \implies v_j(\omega)^*\frac{dv_j(\omega)}{d\omega} = 0,$$

that is  $v_j(\omega)$  and its derivative are orthogonal at all  $\omega$ . Now by rearranging equation (2) and multiplying both sides by  $(\tilde{\lambda}_j(\omega)I - \mathcal{A}(\omega))^\dagger$ , the Moore-Penrose pseudoinverse of  $(\tilde{\lambda}_j(\omega)I - \mathcal{A}(\omega))$ , we have

$$(\tilde{\lambda}_j(\omega)I - \mathcal{A}(\omega))^\dagger(\tilde{\lambda}_j(\omega)I - \mathcal{A}(\omega))\frac{dv_j(\omega)}{d\omega} = (\tilde{\lambda}_j(\omega)I - \mathcal{A}(\omega))^\dagger\frac{d\mathcal{A}(\omega)}{d\omega}v_j(\omega) - \frac{d\tilde{\lambda}_j(\omega)}{d\omega}(\tilde{\lambda}_j(\omega)I - \mathcal{A}(\omega))^\dagger v_j(\omega). \quad (4)$$

Suppose that the multiplicity of  $\tilde{\lambda}_j(\omega)$  is one. Then

$$(\tilde{\lambda}_j(\omega)I - \mathcal{A}(\omega))^\dagger = \sum_{k \neq j} \frac{1}{\tilde{\lambda}_j(\omega) - \tilde{\lambda}_k(\omega)} v_k(\omega)v_k(\omega)^* \quad (5)$$

and

$$(\tilde{\lambda}_j(\omega)I - \mathcal{A}(\omega))^\dagger(\tilde{\lambda}_j(\omega)I - \mathcal{A}(\omega)) = \sum_{k \neq j} v_k(\omega)v_k(\omega)^*.$$

Consequently,

$$(\tilde{\lambda}_j(\omega)I - \mathcal{A}(\omega))^\dagger v_j(\omega) = 0 \quad \text{and} \quad (\tilde{\lambda}_j(\omega)I - \mathcal{A}(\omega))^\dagger(\tilde{\lambda}_j(\omega)I - \mathcal{A}(\omega))\frac{dv_j(\omega)}{d\omega} = \frac{dv_j(\omega)}{d\omega},$$

where the last equality is due to the orthogonality of  $\frac{dv_j(\omega)}{d\omega}$  to  $v_j(\omega)$  meaning  $\frac{dv_j(\omega)}{d\omega} \in \text{span}\{v_k(\omega) : k = 1, \dots, n \text{ and } k \neq j\}$ . Equation (4) simplifies as

$$\frac{dv_j(\omega)}{d\omega} = (\tilde{\lambda}_j(\omega)I - \mathcal{A}(\omega))^\dagger\frac{d\mathcal{A}(\omega)}{d\omega}v_j(\omega). \quad (6)$$

Now suppose that the multiplicity of  $\tilde{\lambda}_j(\omega)$  is greater than one at a given  $\omega_\ell \in \mathbb{R}$ . Indeed suppose that

$$\tilde{\lambda}_j(\omega_\ell) = \tilde{\lambda}_{\alpha_1}(\omega_\ell) = \dots = \tilde{\lambda}_{\alpha_{r-1}}(\omega_\ell).$$

We define a Hermitian matrix function  $\mathcal{C}(\omega)$  that has the same eigenvalues as  $\mathcal{A}(\omega)$  at  $\omega_\ell$ , but the multiplicity of  $\tilde{\lambda}_j(\omega)$  is one at all  $\omega \neq \omega_\ell$  close to  $\omega_\ell$ . If the eigenvalues of  $\mathcal{A}(\omega)$

already satisfies this property, let  $\mathcal{C}(\omega) := \mathcal{A}(\omega)$ . Otherwise the eigenvalues of  $\mathcal{A}(\omega)$  satisfies  $\tilde{\lambda}_j(\omega) = \tilde{\lambda}_{\alpha_k}(\omega)$  for all  $\omega$  for  $k = 1, \dots, r-1$ . In this case consider any Hermitian matrix function  $\mathcal{C}(\omega)$  with exactly the same eigenvectors and eigenvalues as  $\mathcal{A}(\omega)$  for all  $\omega$ , but with eigenvalue  $\tilde{\lambda}_{\alpha_k}(\omega)$  replaced by  $\tilde{\lambda}_{\alpha_k}(\omega) + k(\omega - \omega_\ell)^2$  for  $k = 1, \dots, r-1$ . Theorem 2.1 induces analytic eigenvalue decompositions for  $\mathcal{A}(\omega)$  and  $\mathcal{C}(\omega)$  of the form  $V(\omega)\Lambda(\omega)V(\omega)^*$  where  $V(\omega)$  is unitary and analytic, and  $\Lambda(\omega)$  is diagonal and analytic. It is apparent from these analytic eigenvalue decompositions that

$$\mathcal{A}(\omega_\ell) = \mathcal{C}(\omega_\ell) \quad \text{and} \quad \frac{d\mathcal{A}(\omega_\ell)}{d\omega} = \frac{d\mathcal{C}(\omega_\ell)}{d\omega}. \quad (7)$$

Since the eigenvectors of  $\mathcal{C}(\omega)$  and  $\mathcal{A}(\omega)$  are identical, their derivatives are equal. Furthermore  $\tilde{\lambda}_j(\omega_\ell)$  is an eigenvalue of  $\mathcal{C}(\omega_\ell)$  of multiplicity  $r$ , but at all  $\omega \neq \omega_\ell$  close to  $\omega_\ell$  the eigenvalue  $\tilde{\lambda}_j(\omega)$  of  $\mathcal{C}(\omega)$  has multiplicity one. Both the left-hand and right-hand sides of equation (6) are continuous. Consequently, applying the expression in (6) to  $\mathcal{C}(\omega)$  at  $\omega \neq \omega_\ell$  close to  $\omega_\ell$  and using the continuity as well as (7) yields

$$\begin{aligned} \frac{dv_j(\omega_\ell)}{d\omega} &= (\tilde{\lambda}_j(\omega_\ell)I - \mathcal{C}(\omega_\ell))^\dagger \frac{d\mathcal{C}(\omega_\ell)}{d\omega} v_j(\omega_\ell) \\ &= (\tilde{\lambda}_j(\omega_\ell)I - \mathcal{A}(\omega_\ell))^\dagger \frac{d\mathcal{A}(\omega_\ell)}{d\omega} v_j(\omega_\ell). \end{aligned}$$

This establishes that the expression given in (6) is valid even when the multiplicity of  $\tilde{\lambda}_j(\omega)$  is greater than one.

### 2.2.3 Second Derivatives of Eigenvalues

Finally by differentiating both sides of (3) we obtain

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

By plugging in the expression for the derivative of the eigenvector from (6) we have

$$\frac{d^2 \tilde{\lambda}_j(\omega)}{d\omega^2} = v_j(\omega)^* \frac{d^2 \mathcal{A}(\omega)}{d\omega^2} v_j(\omega) + 2v_j(\omega)^* \frac{d\mathcal{A}(\omega)}{d\omega} (\tilde{\lambda}_j(\omega)I - \mathcal{A}(\omega))^\dagger \frac{d\mathcal{A}(\omega)}{d\omega} v_j(\omega),$$

or by using (5) for the pseudoinverse

$$\frac{d^2 \tilde{\lambda}_j(\omega)}{d\omega^2} = v_j(\omega)^* \frac{d^2 \mathcal{A}(\omega)}{d\omega^2} v_j(\omega) + 2 \sum_{\tilde{\lambda}_k(\omega) \neq \tilde{\lambda}_j(\omega)} \frac{1}{\tilde{\lambda}_j(\omega) - \tilde{\lambda}_k(\omega)} \left| v_k(\omega)^* \frac{d\mathcal{A}(\omega)}{d\omega} v_j(\omega) \right|^2. \quad (8)$$

### 2.2.4 Derivatives of Eigenvalues for Multivariate Hermitian Matrix Functions

Let  $\mathcal{A}(\omega) : \mathbb{R}^d \rightarrow \mathbb{C}^{n \times n}$  be Hermitian and analytic. It follows from (3) that

$$\frac{\partial \tilde{\lambda}_j(\omega)}{\partial \omega_k} = v_j^*(\omega) \frac{\partial \mathcal{A}(\omega)}{\partial \omega_k} v_j(\omega). \quad (9)$$

Since  $\mathcal{A}(\omega)$  and  $v_j(\omega)$  are analytic, this implies the continuity (indeed analyticity) of the partial derivatives, meaning  $\tilde{\lambda}_j(\omega)$  is differentiable. As a consequence of the analyticity of the partial derivatives, all second partial derivatives exist everywhere. Differentiating both sides of (9) with respect to  $\omega_\ell$  would yield the following expressions for the second partial derivatives.

$$\frac{\partial^2 \tilde{\lambda}_j(\omega)}{\partial \omega_k \partial \omega_\ell} = v_j^*(\omega) \frac{\partial^2 \mathcal{A}(\omega)}{\partial \omega_k \partial \omega_\ell} v_j(\omega) + v_j^*(\omega) \frac{\partial \mathcal{A}(\omega)}{\partial \omega_k} \frac{\partial v_j(\omega)}{\partial \omega_\ell} + \left( \frac{\partial v_j(\omega)}{\partial \omega_\ell} \right)^* \frac{\partial \mathcal{A}(\omega)}{\partial \omega_k} v_j(\omega)$$

By eliminating the partial derivatives of the eigenvectors the expression above further simplifies as

$$\begin{aligned} \frac{\partial^2 \tilde{\lambda}_j(\omega)}{\partial \omega_k \partial \omega_\ell} = & v_j^*(\omega) \frac{\partial^2 \mathcal{A}(\omega)}{\partial \omega_k \partial \omega_\ell} v_j(\omega) + v_j^*(\omega) \frac{\partial \mathcal{A}(\omega)}{\partial \omega_k} (\tilde{\lambda}_j(\omega) I - \mathcal{A}(\omega))^\dagger \frac{\partial \mathcal{A}(\omega)}{\partial \omega_\ell} v_j(\omega) \\ & + v_j^*(\omega) \frac{\partial \mathcal{A}(\omega)}{\partial \omega_\ell} (\tilde{\lambda}_j(\omega) I - \mathcal{A}(\omega))^\dagger \frac{\partial \mathcal{A}(\omega)}{\partial \omega_k} v_j(\omega), \end{aligned}$$

or by further eliminating the pseudoinverses we have

$$\begin{aligned} \frac{\partial^2 \tilde{\lambda}_j(\omega)}{\partial \omega_k \partial \omega_\ell} = & v_j^*(\omega) \frac{\partial^2 \mathcal{A}(\omega)}{\partial \omega_k \partial \omega_\ell} v_j(\omega) + \sum_{\tilde{\lambda}_m(\omega) \neq \tilde{\lambda}_j(\omega)} \frac{1}{\tilde{\lambda}_j(\omega) - \tilde{\lambda}_m(\omega)} \left( v_j(\omega)^* \frac{d\mathcal{A}(\omega)}{d\omega_k} v_m(\omega) \right) \left( v_m(\omega)^* \frac{d\mathcal{A}(\omega)}{d\omega_\ell} v_j(\omega) \right) \\ & + \sum_{\tilde{\lambda}_m(\omega) \neq \tilde{\lambda}_j(\omega)} \frac{1}{\tilde{\lambda}_j(\omega) - \tilde{\lambda}_m(\omega)} \left( v_j(\omega)^* \frac{d\mathcal{A}(\omega)}{d\omega_\ell} v_m(\omega) \right) \left( v_m(\omega)^* \frac{d\mathcal{A}(\omega)}{d\omega_k} v_j(\omega) \right). \end{aligned}$$

### 2.3 Analyticity of Singular Values

Some of the applications (see Section 7) concern the optimization of the  $j$ th largest singular value of an analytic matrix function. The singular value problems are special Hermitian eigenvalue problems. In particular denote the  $j$ th largest singular value of an analytic matrix function  $\mathcal{B}(\omega) : \mathbb{R}^d \rightarrow \mathbb{C}^{n \times m}$ , not necessarily Hermitian, by  $\sigma_j(\omega)$ . Then the set of eigenvalues of the Hermitian matrix function

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

is  $\{\sigma_j(\omega), -\sigma_j(\omega) : j = 1, \dots, n\}$ . In the univariate case  $\sigma_j(\omega)$  is the  $j$ th largest of the  $2n$  unordered analytic eigenvalues,  $\tilde{\lambda}_1(\omega), \dots, \tilde{\lambda}_{2n}(\omega)$ , of  $\mathcal{A}(\omega)$ . The multivariate  $d$  dimensional case is similar, with the exception that each unordered eigenvalue  $\tilde{\lambda}_j(\omega)$  is differentiable and analytic along every line in  $\mathbb{R}^d$ . Let us focus on the univariate case throughout the rest of this subsection. Extensions to the multi-variate case are similar to the previous subsections.

Suppose  $v_j(\omega) := \begin{bmatrix} u_j(\omega) \\ w_j(\omega) \end{bmatrix}$  with  $u_j(\omega), w_j(\omega) \in \mathbb{C}^n$  is the analytic eigenvector function as specified in Theorem 2.1 of  $\mathcal{A}(\omega)$  associated with  $\tilde{\lambda}_j(\omega)$ , that is

$$\begin{bmatrix} 0 & \mathcal{B}(\omega) \\ \mathcal{B}(\omega)^* & 0 \end{bmatrix} \begin{bmatrix} u_j(\omega) \\ w_j(\omega) \end{bmatrix} := \tilde{\lambda}_j(\omega) \begin{bmatrix} u_j(\omega) \\ w_j(\omega) \end{bmatrix}.$$

The above equation implies

$$\mathcal{B}(\omega)w_j(\omega) = \tilde{\lambda}_j(\omega)u_j(\omega) \quad \text{and} \quad \mathcal{B}(\omega)^*u_j(\omega) = \tilde{\lambda}_j(\omega)w_j(\omega). \quad (10)$$

In other words  $u_j(\omega)$ ,  $w_j(\omega)$  are analytic, and consists of a pair of consistent left and right singular vectors associated with  $\tilde{\lambda}_j(\omega)$ . To summarize in the univariate case  $\tilde{\lambda}_j(\omega)$  can be considered as an unsigned and unordered singular value of  $\mathcal{B}(\omega)$ , and there is a consistent pair of analytic left  $u_j(\omega)$  and right  $w_j(\omega)$  singular vector functions.

Now in the univariate case we derive expressions for the first derivative of  $\tilde{\lambda}_j(\omega)$ , in terms of the corresponding left and right singular vectors. It follows from the singular value equations (10) above that  $\|u_j(\omega)\| = \|w_j(\omega)\| = 1/\sqrt{2}$  (if  $\tilde{\lambda}_j(\omega) = 0$  this equality follows from analyticity). Now the application of expression (3) yields

$$\begin{aligned} \frac{d\tilde{\lambda}_j(\omega)}{d\omega} &= \begin{bmatrix} u_j(\omega)^* & w_j(\omega)^* \end{bmatrix} \begin{bmatrix} 0 & d\mathcal{B}(\omega)/d\omega \\ d\mathcal{B}(\omega)^*/d\omega & 0 \end{bmatrix} \begin{bmatrix} u_j(\omega) \\ w_j(\omega) \end{bmatrix} \\ &= u_j(\omega)^* \frac{d\mathcal{B}(\omega)}{d\omega} w_j(\omega) + w_j(\omega)^* \frac{d\mathcal{B}(\omega)^*}{d\omega} u_j(\omega) \\ &= 2 \operatorname{Real} \left( u_j(\omega)^* \frac{d\mathcal{B}(\omega)}{d\omega} w_j(\omega) \right) \end{aligned}$$

In terms of the unit left  $\hat{u}_j(\omega) := \sqrt{2} \cdot u_j(\omega)$  and right  $\hat{w}_j(\omega) := \sqrt{2} \cdot w_j(\omega)$  singular vectors associated with  $\tilde{\lambda}_j(\omega)$  we obtain

$$\frac{d\tilde{\lambda}_j(\omega)}{d\omega} = \operatorname{Real} \left( \hat{u}_j(\omega)^* \frac{d\mathcal{B}(\omega)}{d\omega} \hat{w}_j(\omega) \right). \quad (11)$$

### 3 One-Dimensional Algorithm

We suppose  $f_1, \dots, f_n : \mathbb{R} \rightarrow \mathbb{R}$  are analytic functions. The function  $f : \mathbb{R} \rightarrow \mathbb{R}$  is a continuous and piece-wise function defined in terms of  $f_1, \dots, f_n$ . Here we describe an algorithm for finding a global minimizer of  $f$ . Certainly the ordered eigenvalues fit into this framework, i.e., the  $j$ th largest eigenvalue  $\lambda_j$  is a continuous and piece-wise analytic function defined in terms of the unordered eigenvalues  $\tilde{\lambda}_1, \dots, \tilde{\lambda}_n$ .

Any pair of functions among  $f_1, \dots, f_n$  can intersect each other only at finitely many isolated points on a finite interval, or otherwise they are identical due to analyticity. The finitely many isolated intersection points are the only points where the piece-wise function  $f$  is possibly not analytic. Clearly  $f$  is continuous but may not be differentiable at these points.

The algorithm is based on a quadratic model  $q_k(x)$  about a given point  $x_k \in \mathbb{R}$  that lies underneath  $f(x)$  at all  $x \in \mathbb{R}$ . For any point  $x \geq x_k$  denote the points where  $f$  is not differentiable on  $[x_k, x]$  by  $x^{(1)}, \dots, x^{(m)}$ . Then

$$f(x) = f(x_k) + \sum_{j=0}^m \int_{x^{(j)}}^{x^{(j+1)}} f'(t) dt \quad (12)$$

where  $x^{(0)} := x_k$  and  $x^{(m+1)} := x$ . Suppose that  $\gamma$  is a global upper bound on the second derivatives, that is

$$|f_j''(x)| \leq \gamma \quad \forall x \in \mathbb{R}$$

for  $j = 1, \dots, n$ . For some  $\eta \in (x_k, t)$  we have

$$f_j'(t) = f_j'(x_k) + f_j''(\eta)(t - x_k) \geq f_j'(x_k) - \gamma(t - x_k)$$

implying

$$f'(t) \geq \underline{f}'(x_k) - \gamma(t - x_k) \quad (13)$$

where  $\underline{f}'(x_k) := \min_{k=1,n} f'_j(x_k)$ . Using the lower bound (13) in (12) yields

$$\begin{aligned} f(x) &\geq f(x_k) + \sum_{j=0}^m \int_{x^{(j)}}^{x^{(j+1)}} \underline{f}'(x_k) - \gamma(t - x_k) dt \\ &= f(x_k) + \int_{x_k}^x \underline{f}'(x_k) - \gamma(t - x_k) dt. \end{aligned}$$

Finally by evaluating the integral on the right we deduce

$$f(x) \geq f(x_k) + \underline{f}'(x_k)(x - x_k) - \frac{\gamma}{2}(x - x_k)^2$$

at all  $x \geq x_k$ . Similarly at all  $x < x_k$  we have

$$f(x) \geq f(x_k) + \overline{f}'(x_k)(x - x_k) - \frac{\gamma}{2}(x - x_k)^2$$

where  $\overline{f}'(x_k) := \max_{k=1,n} f'_j(x_k)$ . Consequently the piece-wise quadratic function

$$q_k(x) := \begin{cases} f(x_k) + \overline{f}'(x_k)(x - x_k) - \frac{\gamma}{2}(x - x_k)^2 & x < x_k \\ f(x_k) + \underline{f}'(x_k)(x - x_k) - \frac{\gamma}{2}(x - x_k)^2 & x \geq x_k \end{cases} \quad (14)$$

about  $x_k$  satisfies  $f(x) \geq q_k(x)$  for all  $x \in \mathbb{R}$ . For eigenvalue functions generically the multiplicity of the eigenvalue at a global minimizer is one, meaning  $f(x) = f_j(x)$  at all  $x$  close to a global minimizer for some  $j$ . In this case  $f(x)$  is analytic, and the quadratic model could be simplified as

$$q_k(x) = f(x_k) + f'(x_k)(x - x_k) - \frac{\gamma}{2}(x - x_k)^2.$$

We assume the knowledge of an interval  $[\underline{x}, \overline{x}]$  in which the global minimizer is contained. A high-level description of the algorithm is given below.

#### Description of the Algorithm

1. Initially there are only two quadratic models  $q_0$  and  $q_1$  about  $x_0 := \underline{x}$  and  $x_1 := \overline{x}$ , respectively.
2. Find the global minimizer  $x_*$  of the piecewise quadratic function

$$\overline{q}_s(x) := \max_{k=0,s} q_k(x).$$

on  $[\underline{x}, \overline{x}]$  where  $s + 1$  is the number of quadratic models.

3. The lower and upper bounds for the globally minimal value of  $f(x)$  are given by

$$l = \overline{q}_s(x_*) \quad \text{and} \quad u = \min_{k=0,s} f(x_k).$$

4. Let  $x_{s+1} := x_*$ . Evaluate  $f(x_{s+1})$  and  $f'(x_{s+1})$ , form the quadratic model  $q_{s+1}$  about  $x_{s+1}$ , and increment  $s$ .
5. Repeat steps 2-4 until  $u - l$  is less than a prescribed tolerance.

Note that, for the optimization of Hermitian eigenvalues, evaluation of  $f(x_{s+1})$  corresponds to an eigenvalue computation of an  $n \times n$  matrix. In this case once  $f(x_{s+1})$ , that is the eigenvalue and the associated eigenvector, are evaluated, the derivative  $f'(x_{s+1})$  is cheap to calculate due to expression (3). The first five iterations of the algorithm applied to minimize  $\sigma_{\min}(A - \omega iI)$  over the real line are illustrated in Figure 1. The red curve is a plot of the graph of  $f(\omega) = \sigma_{\min}(A - \omega iI)$ , whereas the blue curves represent the quadratic models. The global minimizers of the piece-wise quadratic model are marked by asterisks.

## 4 Multi-Dimensional Algorithm

Now suppose  $f_1, \dots, f_n : \mathbb{R}^d \rightarrow \mathbb{R}$  are functions analytic along each line in  $\mathbb{R}^d$ , and differentiable on  $\mathbb{R}^d$ . The function  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  is a continuous and piece-wise function defined in terms of  $f_1, \dots, f_n$ . We would like to locate a global minimizer of  $f$ . For instance  $f$  could be considered as the  $j$ th largest eigenvalue of a Hermitian matrix function that varies analytically in  $\mathbb{R}^d$ , as the unordered eigenvalues of such matrix functions obey the properties required for  $f_1, \dots, f_n$  by Theorem 2.2.

We first derive a quadratic model  $q_k(x)$  about a given point  $x_k \in \mathbb{R}^d$  such that  $q_k(x) \leq f(x)$  for all  $x \in \mathbb{R}^d$ . Let us consider the direction  $p := (x - x_k)/\|x - x_k\|$ , the univariate function  $\phi(\alpha) := f(x_k + \alpha p)$  and the analytic functions  $\phi_j(\alpha) := f_j(x_k + \alpha p)$  for  $j = 1, \dots, n$  defining  $\phi$ . Also let us denote the finitely many points in the interval  $[0, \|x - x_k\|]$  where  $\phi(\alpha)$  is not differentiable by  $\alpha^{(1)}, \dots, \alpha^{(m)}$ . Then we have

$$f(x) = f(x_k) + \sum_{\ell=0}^m \int_{\alpha^{(\ell)}}^{\alpha^{(\ell+1)}} \phi'(t) dt \quad (15)$$

where  $\alpha^{(0)} := 0$  and  $\alpha^{(m+1)} := \|x - x_k\|$ . Now due to the differentiability of  $f_j$  we have

$$\phi'_j(\alpha) = \nabla f_j(x_k + \alpha p)^T p. \quad (16)$$

Furthermore since  $f_j(x_k + \alpha p)$  is analytic with respect to  $\alpha$  there exists a constant  $\gamma$  satisfying

$$|\phi''_j(\alpha)| \leq \gamma \quad \forall \alpha \in \mathbb{R} \quad (17)$$

for  $j = 1, \dots, n$ . Now as in Section 3 we have

$$\phi'(t) \geq \min_{j=1, n} \phi'_j(0) - \gamma t.$$

By plugging the last inequality in (15), then integrating the right-hand side of (15) we obtain

$$f(x) \geq f(x_k) + (\min_{j=1, n} \phi'_j(0)) \|x - x_k\| - \frac{\gamma}{2} \|x - x_k\|^2.$$

Finally using (16) at  $\alpha = 0$  and  $p := (x - x_k)/\|x - x_k\|$  yields

$$f(x) \geq q_k(x) := f(x_k) + (\min_{j=1, n} \nabla f_j(x_k)^T (x - x_k)) - \frac{\gamma}{2} \|x - x_k\|^2. \quad (18)$$

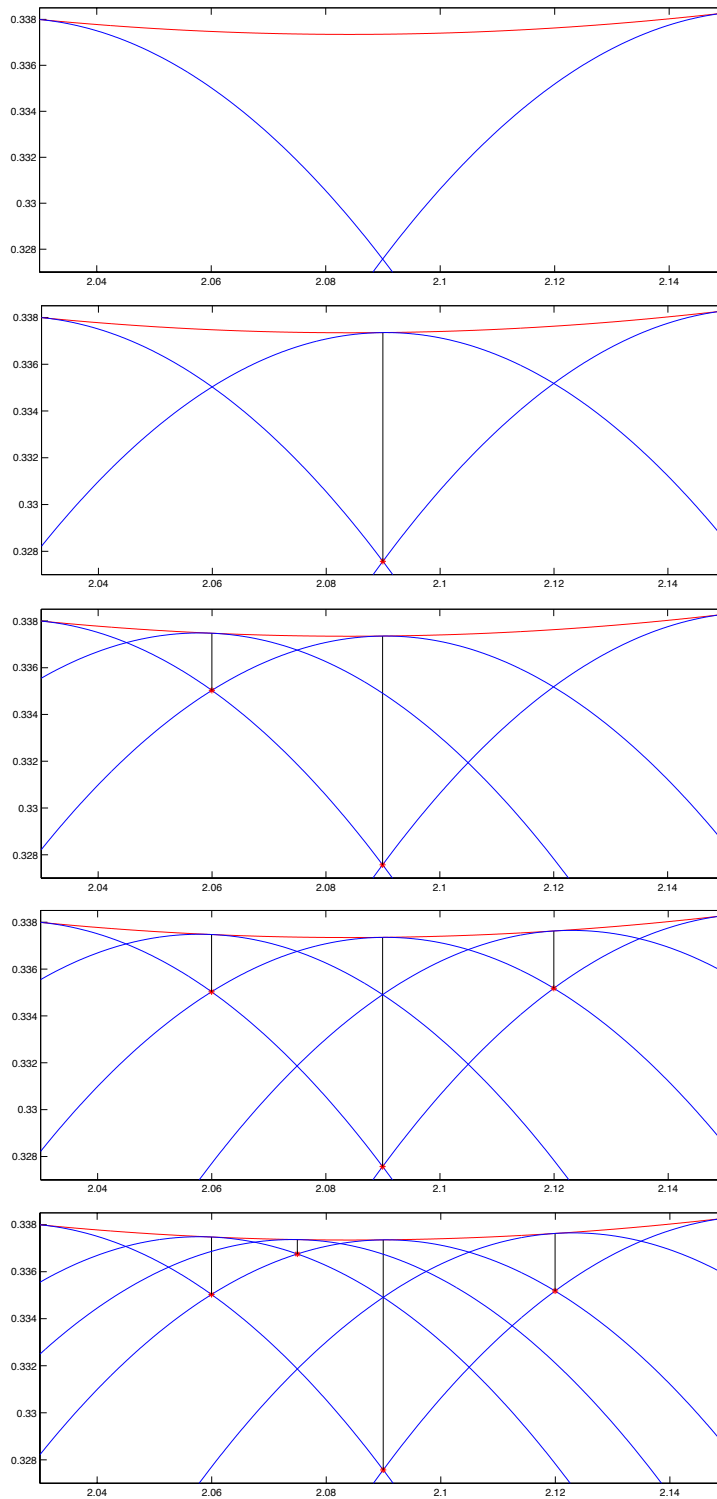


Figure 1: The first five iterations of the quadratic-model based algorithm for the minimization of piece-wise analytic functions

The algorithm in the multivariate case is same as the algorithm given in Section 3 for the univariate case, but with the definition of the quadratic model function generalized as in (18). As in the univariate case we assume that a box

$$\mathcal{B} := \mathcal{B}(\underline{x}_1, \bar{x}_1, \dots, \underline{x}_d, \bar{x}_d) := \{x \in \mathbb{R}^d : x_j \in [\underline{x}_j, \bar{x}_j] \text{ for } j = 1, \dots, d\} \quad (19)$$

containing a global minimizer is known, and the algorithm is performed inside this box. For convenience and for the convergence analysis in the next section the algorithm is formally presented below.

---

**Algorithm 1** Multidimensional Algorithm
 

---

**Require:** A continuous and piece-wise function  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  that is defined in terms of differentiable functions  $f_j$  for  $j = 1, \dots, n$  each of which is analytic along every line in  $\mathbb{R}^d$ , a scalar  $\gamma > 0$  satisfying (16), a set  $\mathcal{B}$  given by (19), and a tolerance parameter  $\epsilon > 0$ .

- 1: Pick an arbitrary  $x_0 \in \mathcal{B}$ .
  - 2:  $u_1 \leftarrow f(x_0)$ ;  $xbest \leftarrow x_0$ .
  - 3:  $\bar{q}_0(x) := q_0(x) := f(x_0) + \min_{j=1,n} \{\nabla f_j(x_0)^T (x - x_0)\} - (\gamma/2) \|x - x_0\|^2$ .
  - 4:  $x_1 \leftarrow \arg \min_{x \in \mathcal{B}} \bar{q}_0(x)$ .
  - 5:  $l_1 \leftarrow \bar{q}_0(x_1)$ .
  - 6: **if**  $f(x_1) < u_1$  **then**
  - 7:    $u_1 \leftarrow f(x_1)$ ;  $xbest \leftarrow x_1$ .
  - 8: **end if**
  - 9:  $s \leftarrow 1$ .
  - 10: **While**  $u_s - l_s > \epsilon$  **do**
  - 11: **loop**
  - 12:    $q_s(x) := f(x_s) + \min_{j=1,n} \{\nabla f_j(x_k)^T (x - x_s)\} - (\gamma/2) \|x - x_s\|^2$ .
  - 13:    $\bar{q}_s(x) := \max_{k=0,s} \{q_k(x)\}$ .
  - 14:    $x_{s+1} \leftarrow \arg \min_{x \in \mathcal{B}} \bar{q}_s(x)$ .
  - 15:    $l_{s+1} \leftarrow \bar{q}_s(x_{s+1})$ .
  - 16:   **if**  $f(x_{s+1}) < u_s$  **then**
  - 17:      $u_{s+1} \leftarrow f(x_{s+1})$ ;  $xbest \leftarrow x_{s+1}$ .
  - 18:   **else**
  - 19:      $u_{s+1} \leftarrow u_s$ .
  - 20:   **end if**
  - 21:    $s \leftarrow s + 1$ .
  - 22: **end loop**
  - 23: **Output:**  $l_s, u_s, xbest$ .
- 

In the multivariate case finding a global minimizer  $x_*$  of

$$\bar{q}_s(x) = \max_{k=0,s} q_k(x)$$

is a tough task. For the sake of simplicity first let us suppose  $f$  is defined in terms of one function, that is  $f$  is indeed analytic. Then the quadratic model (18) simplifies as

$$q_k(x) := f(x_k) + \nabla f(x_k)^T (x - x_k) - \frac{\gamma}{2} \|x - x_k\|^2. \quad (20)$$

Typically this is a reasonable assumption for the  $j$ th largest eigenvalue of a Hermitian matrix function in practice. Generically the  $j$ th largest eigenvalue is of multiplicity one at all points close to a global minimizer.

To minimize  $\bar{q}_s(x)$  with the quadratic model (20) we partition the box  $\mathcal{B}$  into regions  $\mathcal{R}_0, \dots, \mathcal{R}_s$  such that inside the region  $\mathcal{R}_k$  the quadratic function  $q_k(x)$  takes the largest value (see Figure 2). Therefore the minimization of  $\bar{q}_s(x)$  over the region  $\mathcal{R}_k$  is equivalent to the minimization of  $q_k(x)$  over the same region. This can be posed as the quadratic program below.

$$\begin{aligned} & \text{minimize}_{x \in \mathbb{R}^d} && q_k(x) \\ & \text{subject to} && q_k(x) \geq q_\ell(x), \quad \ell \neq k \\ & && x_j \in [\underline{x}_j, \bar{x}_j], \quad j = 1, \dots, d \end{aligned} \tag{21}$$

Note that the inequalities  $q_k(x) \geq q_\ell(x)$  are linear, as  $q_k(x)$  and  $q_\ell(x)$  have the same negative curvature, and consequently the quadratic terms of  $q_k(x)$  and  $q_\ell(x)$  cancel out. The minimization of  $\bar{q}_s(x)$  over  $\mathcal{B}$  can be performed by solving the quadratic program above for  $k = 0, \dots, s$ . The difficulty with the problem above is due to the negative-definiteness of  $\nabla^2 q_k(x)$ . This makes the problem non-convex, but the solution is guaranteed to be attained

at one of the vertices of the feasible region. Clearly there can be at most  $\binom{s+d-1}{d}$  vertices, where  $s+d-1$  is the number of constraints. In the 2-dimensional case the number of vertices can be proportional to  $O(s^2)$  in theory, but in practice we observe that the number of vertices does not exceed 5-6. In theory solving an indefinite quadratic program is NP-hard. The problem above can be expected to be solved efficiently for small  $d$  only. We are able to solve it efficiently for  $d = 2$  at the moment. Most of the problems mentioned in the opening are either one or two dimensional. For instance the distance to uncontrollability and the distance to the nearest matrix with an eigenvalue of specified multiplicity (in particular the distance to the nearest defective matrix) are two dimensional.

Let us comment on finding the global minimizer of  $\bar{q}_s(x)$  to the full generality with the quadratic model (18). We use the notation

$$q_{k,j}(x) := f(x_k) + \nabla f_j(x_k)^T(x - x_k) - \frac{\gamma}{2} \|x - x_k\|^2$$

so that  $q_k(x) = \min_{j=1,n} q_{k,j}(x)$ . Then the box  $\mathcal{B}$  could be partitioned into regions  $\mathcal{R}_{k,j}$ ,  $k = 0, \dots, s$ ,  $j = 1, \dots, n$  where

- $q_{k,j}$  is not larger than  $q_{k,\ell}$  for  $\ell \neq j$ , and
- $q_{k,j}$  is not smaller than at least one of  $q_{p,\ell}$ ,  $\ell = 1, \dots, n$  for  $p \neq k$ .

Minimization of  $\bar{q}_s(x)$  over  $\mathcal{B}$  can be achieved by minimizing  $q_{k,j}(x)$  over  $\mathcal{R}_{k,j}$  for  $k = 1, \dots, s$  and  $j = 1, \dots, n$ . The latter problem could be posed as a quadratic program of the form

$$\begin{aligned} & \text{minimize}_{x \in \mathbb{R}^d} && q_{k,j}(x) \\ & \text{subject to} && q_{k,j}(x) \leq q_{k,\ell}(x), \quad \ell \neq j \\ & && \forall \ell=1,n \quad q_{k,j}(x) \geq q_{p,\ell}(x), \quad p \neq k \\ & && x_j \in [\underline{x}_j, \bar{x}_j], \quad j = 1, \dots, d \end{aligned} \tag{22}$$

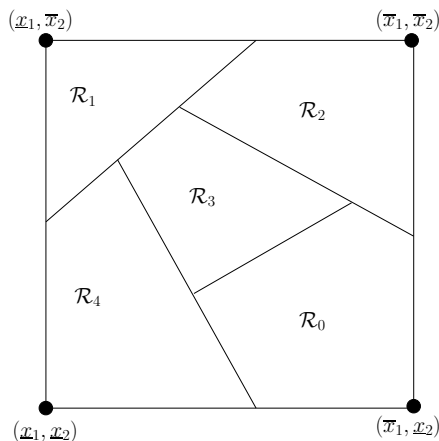


Figure 2: To minimize the piece-wise quadratic function  $q(x)$  over the box  $\mathcal{B}$  the box is split into regions  $\mathcal{R}_k$ ,  $k = 1, \dots, s$  such that  $q_k(x)$  is the largest inside the region  $\mathcal{R}_k$ . Above a possible partitioning is illustrated in the 2-dimensional case.

Above the notation  $\vee_{\ell=1,n} q_{k,j}(x) \geq q_{p,\ell}(x)$  means that at least one of the constraints  $q_{k,j}(x) \geq q_{p,\ell}(x)$  for  $\ell = 1, \dots, n$  must be satisfied. Clearly this is even a more difficult problem than (21). In practice it suffices to solve (21) instead of (22), because the multiplicity of the  $j$ th largest eigenvalue generically remains one close to a global minimizer.

## 5 Convergence Analysis of Multi-Dimensional Algorithm

In this section we analyze the convergence of Algorithm 1 for the following optimization problem.

$$(P) \quad f^* := \min_{x \in \mathcal{B}} f(x)$$

Recall that the algorithm starts off by picking an arbitrary point  $x_0 \in \mathcal{B}$ . At iteration  $s$ , the algorithm picks  $x_{s+1}$  to be a global minimizer of  $\bar{q}_s(x)$  over  $\mathcal{B}$ , where  $\bar{q}_s(x)$  is the maximum of the functions  $q_k(x)$  constructed at the points  $x_k$ ,  $k = 0, \dots, s$  in  $\mathcal{B}$ . Note that  $\{l_s\}$  is a non-decreasing sequence of lower bounds on  $f^*$ , while  $\{u_s\}$  is a non-increasing sequence of upper bounds on  $f^*$ .

We require that  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  is a continuous and piece-wise function defined in terms of the differentiable functions  $f_j : \mathbb{R}^d \rightarrow \mathbb{R}$ ,  $j = 1, \dots, n$ . Differentiability of each  $f_j$  on  $\mathbb{R}^d$  implies the boundedness of  $\|\nabla f_j(x)\|$  on  $\mathbb{R}^d$ . Consequently, we define

$$\mu := \max_{j=1,n} \max_{x \in \mathcal{B}} \|\nabla f_j(x)\|.$$

We furthermore require each piece  $f_j$  to be analytic along every line in  $\mathbb{R}^d$ , which implies the existence of a scalar  $\gamma > 0$  satisfying (17). Our convergence analysis depends on the scalars  $\mu$  and  $\gamma$ . We now establish the convergence of Algorithm 1 to a global minimizer of (P).

**Theorem 5.1.** *Let  $\{x_s\}$  be the sequence of iterates generated by Algorithm 1. Every limit point of this sequence is a global minimizer of the problem (P).*

*Proof.* Since  $\mathcal{B}$  is a compact set, it follows that the sequence  $\{x_s\}$  has at least one limit point  $x^* \in \mathcal{B}$ . By passing to a subsequence if necessary, we may assume that  $\{x_s\}$  itself is a convergent sequence. Let  $l^*$  denote the limit of the bounded nondecreasing sequence  $\{l_s\}$ . Since  $l_s \leq f^* \leq f(x_s)$  for each  $s \geq 0$ , it suffices to show that  $l^* = f(x^*)$ .

Suppose, for a contradiction, that there exists a real number  $\delta > 0$  such that

$$f(x^*) \geq l^* + \delta. \quad (23)$$

By the continuity of  $f$ , there exists  $s_1 \in \mathbb{N}$  such that

$$f(x_s) \geq l^* + \frac{\delta}{2}, \quad \text{for all } s \geq s_1. \quad (24)$$

Since  $x^*$  is the limit of the sequence  $\{x_s\}$ , there exists  $s_2 \in \mathbb{N}$  such that

$$\|x_{s'} - x_{s''}\| < \min \left\{ \sqrt{\frac{\delta}{6\gamma}}, \frac{\delta}{12\mu} \right\}, \quad \text{for all } s' \geq s'' \geq s_2, \quad (25)$$

where we define  $1/\mu := +\infty$  if  $\mu = 0$ . Let  $s_* = \max\{s_1, s_2\}$ . For each  $s \geq s_*$ , it follows from the definition of the functions  $\bar{q}_s(x)$  that

$$\begin{aligned} \bar{q}_s(x_{s+1}) &\geq \bar{q}_{s_*}(x_{s+1}), \\ &\geq q_{s_*}(x_{s+1}), \\ &= f(x_{s_*}) + \nabla f_{j_*}(x_{s_*})^T (x_{s+1} - x_{s_*}) - \frac{\gamma}{2} \|x_{s+1} - x_{s_*}\|^2. \end{aligned}$$

where  $j_* \in \{1, \dots, n\}$  is the index of the gradient that determines the value of  $q_{s_*}(x_{s+1})$ . Now by applying the Cauchy-Schwarz inequality and then using the inequalities (24) and (25) we arrive at

$$\begin{aligned} \bar{q}_s(x_{s+1}) &\geq f(x_{s_*}) - \|\nabla f_{j_*}(x_{s_*})\| \|x_{s+1} - x_{s_*}\| - \frac{\gamma}{2} \|x_{s+1} - x_{s_*}\|^2, \\ &\geq \left( l^* + \frac{\delta}{2} \right) - \left( \mu \cdot \frac{\delta}{12\mu} \right) - \left( \frac{\gamma}{2} \cdot \frac{\delta}{6\gamma} \right), \\ &= l^* + \frac{\delta}{3}, \end{aligned}$$

Using the definition  $l_{s+1} = \bar{q}_s(x_{s+1})$ , it follows that

$$l_{s+1} \geq l^* + \frac{\delta}{3}, \quad \text{for all } s \geq s_*.$$

Since  $\delta > 0$ , this contradicts our assumption that  $l^*$  is the limit of the non-decreasing sequence  $\{l_s\}$ . Therefore, we have  $f(x^*) < l^* + \delta$  for all  $\delta > 0$ , or equivalently  $f(x^*) \leq l^*$ . Since  $l_s \leq f(x)$  for all  $s \in \mathbb{N}$  and  $x \in \mathcal{B}$ , it follows that  $l^* \leq f(x^*)$ , which establishes that  $f(x^*) = l^* \leq f(x)$  for all  $x \in \mathcal{B}$ . Therefore,  $x^*$  is a global minimizer of (P). The assertion is proved by repeating the same argument for any other limit point of the sequence  $\{x_k\}$ .  $\square$

## 6 Practical Issues

The distance to uncontrollability from a time-invariant linear control system  $(A, B)$  (see Section 7.3 for the formal definition of the distance to uncontrollability) where  $A \in \mathbb{C}^{n \times n}$  and  $B \in \mathbb{C}^{n \times m}$  with  $n \geq m$  has the eigenvalue optimization characterization

$$\inf_{z \in \mathbb{C}} \sigma_n \left( \begin{bmatrix} A - zI & B \end{bmatrix} \right).$$

Below we apply Algorithm 1 to calculate the distance to uncontrollability for a pair of random matrices (of size  $100 \times 100$  and  $100 \times 30$ , respectively) whose entries are selected from a normal distribution with zero mean and unit variance. The running times (in seconds) in Matlab for every thirty iterations are given in the table below.

<b>iterations</b>	1-30	31-60	61-90	91-120	121-150
<b>cpu-time</b>	3.283	6.514	10.372	17.069	28.686

Without doubt the later iterations are expensive. At iteration  $s$  there are  $s$  quadratic programs of the form (21). Each of the  $s$  quadratic programs has  $s - 1 + d$  constraints. When we add the  $s$ th quadratic model, we create a new quadratic program for which we need to calculate its potentially  $\binom{s - 1 + d}{2}$  vertices. (Many of these potential vertices would turn out to be infeasible in practice; but this is difficult to know in advance.) Furthermore each of the existing  $s - 1$  quadratic programs has now one more constraint, consequently each of the existing  $s - 1$  quadratic programs has potentially  $s - 2 + d$  additional vertices. We need to calculate these vertices as well. The total number of vertices that need to be calculated at iteration  $s$  is  $O(s^2)$ . Calculation of these vertices dominates the computation time eventually even for a system with large system matrices  $(A, B)$ . This is a rare situation for which the computation time is dominated by the solution of  $2 \times 2$  linear systems! Obviously the situation does not get any better in higher dimensions. In the  $d$  dimensional case the work at the  $s$ th iteration would be  $O(s^d)$ .

Consider the following two schemes both of which use  $s$  quadratic functions in the 2-dimensional case.

- (1) Form a piece-wise quadratic model consisting of  $s$  quadratic functions on a  $\ell_1 \times \ell_2$  box.
- (2) Split the  $\ell_1 \times \ell_2$  box into four sub-boxes, each of size  $\frac{\ell_1}{2} \times \frac{\ell_2}{2}$ . Then use a piece-wise quadratic model consisting of  $s/4$  quadratic functions inside each sub-box.

The latter scheme is 16 times cheaper. Furthermore our quadratic models capture the eigenvalue function much better in a small box, since our quadratic models are defined in terms of derivatives, that is the local information.

It appears wiser not to let  $s$  grow too large. But we also would like  $s$  to be not too small so that the piece-wise quadratic model can capture the eigenvalue function to a degree. Once the cost of adding a new quadratic function becomes expensive, we can split a box into sub-boxes, and construct piece-wise quadratic functions inside each sub-box from scratch separately. We start with  $2^d$  sub-boxes of equal size. Apply Algorithm 1 on each sub-box until either the prescribed accuracy  $\epsilon$  is reached, or the number of quadratic functions exceeds a prescribed value  $n_q$ . Then we have a global upper bound resulting from the function values evaluated so

far on various sub-boxes. We further partition each sub-box into sub-boxes if the lower bound for the sub-box (resulting from the piece-wise quadratic model particularly constructed for the sub-box) is less than the global upper bound minus the prescribed tolerance  $\epsilon$ . In practice we observe that many of the sub-boxes do not need to be further partitioned, because their lower bounds are larger than the global upper bound minus  $\epsilon$ . The practical algorithm is summarized below.

---

**Algorithm 2** Mesh-Adaptive Multidimensional Algorithm
 

---

**Require:** A continuous and piece-wise function  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  that is defined in terms of differentiable functions  $f_j$  for  $j = 1, \dots, n$  each of which is analytic along every line in  $\mathbb{R}^d$ , a scalar  $\gamma > 0$  satisfying (16), a set  $\mathcal{B}$  given by (19), a tolerance parameter  $\epsilon > 0$ , and a parameter  $n_q$  for the number of quadratic functions.

- 1: Partition  $\mathcal{B}$  into boxes  $\mathcal{B}_1, \dots, \mathcal{B}_{2^d}$  of equal size.
  - 2:  $u \leftarrow \infty$
  - 3: For  $j = 1, \dots, 2^d$  do
  - 4: **loop**
  - 5:   Apply Algorithm 1 on  $\mathcal{B}_j$  but with the constraint that  $s$  does not exceed  $n_q$ . Let  $l_j$  and  $u_j$  be the returned lower, upper bounds. Let  $x_j$  be the returned *xbest*.
  - 6:   **if**  $u_j < u$  **then**
  - 7:      $u \leftarrow u_j; x \leftarrow x_j$
  - 8:   **end if**
  - 9: **end loop**
  - 10:  $l \leftarrow \min_{j=1, 2^d} \{l_j\}$
  - 11: **if**  $u - l > \epsilon$  **then**
  - 12:   Sort the boxes  $\mathcal{B}_1, \dots, \mathcal{B}_{2^d}$  according to their upper bounds  $u_1, \dots, u_{2^d}$  from smallest to largest. Sort also the lower and upper bounds of the boxes accordingly.
  - 13:   For  $j = 1, \dots, 2^d$  do
  - 14:    **loop**
  - 15:     **if**  $(u - l_j) > \epsilon$  **then**
  - 16:      Apply Algorithm 2 on  $\mathcal{B}_j$ . Let  $l_j$  and  $u_j$  be the returned lower, upper bounds. Let  $x_j$  be the returned *xbest*.
  - 17:      **if**  $u_j < u$  **then**
  - 18:        $u \leftarrow u_j; x \leftarrow x_j$
  - 19:      **end if**
  - 20:     **end if**
  - 21:    **end loop**
  - 22:    $l \leftarrow \min_{j=1, 2^d} \{l_j\}$
  - 23: **end if**
  - 24: **Output:**  $l, u, x$ .
- 

Sorting the boxes according to their upper-bounds makes the algorithm greedy. When we further partition, we start with the box yielding the smallest function value so far, continue with the one yielding the second smallest function value and so on. There are a few further practical improvements that could be made.

- When Algorithm 1 is called inside Algorithm 2, it is possible to benefit from the global

upper bound  $u$ . Algorithm 1 could terminate once  $\min\{u, u_s\} - l_s < \epsilon$  instead of  $u_s - l_s < \epsilon$ .

- Furthermore, in Algorithm 1, for each of the quadratic program if the optimal value exceeds  $u - \epsilon$ , then the quadratic program could be discarded at the later iterations. The reasoning is simple; the eigenvalue function over the feasible region of the quadratic program is verified to be no smaller than  $u - \epsilon$ . If the eigenvalue function really takes a value smaller than  $u - \epsilon$ , it will be in some other region. If the eigenvalue function does not take a value smaller than  $u - \epsilon$ , then there is no reason to further search anyway; eventually the lower bounds from other regions will also approach or exceed  $u - \epsilon$ .

For the example mentioned at the opening of this section Algorithm 1 calculates the distance to uncontrollability as 0.785, and guarantees that the exact value can differ from this computed value by at most 0.23 after 150 iterations and 66 seconds. Algorithm 2 on the other hand with  $n_q = 30$  requires 55, 121, 171 and 214 seconds to calculate the distance to uncontrollability within an error of  $10^{-2}$ ,  $10^{-4}$ ,  $10^{-6}$  and  $10^{-8}$ , respectively. Just to guarantee  $10^{-2}$  accuracy Algorithm 1 iterates 305 times and uses 1298 seconds of cpu time.

## 7 Applications to Eigenvalue Optimization

We reserve this section for the applications of the algorithm introduced and analyzed in the previous four sections to particular eigenvalue optimization problems. For each problem we deduce expressions for derivatives in terms of eigenvectors using (3) and (11). It may be possible to deduce the bound  $\gamma$  for the second derivatives using (8), though we will not attempt it here.

### 7.1 Numerical Radius

The numerical radius  $r(A)$  of a square matrix  $A \in \mathbb{C}^{n \times n}$  is the modulus of the outer-most point in its field of values [20]

$$F(A) = \{z^*Az \in \mathbb{C} : z \in \mathbb{C}^n \text{ s.t. } \|z\|_2 = 1\}.$$

The numerical radius gives information about the powers of  $A$ , e.g.,  $\|A^k\| \leq 2r(A)^k$ . In literature it is used to analyze the convergence of iterative methods for the solution of linear systems [11, 3].

The facts that the right-most intersection point of  $F(A)$  with the real axis is given by

$$\lambda_1 \left( \frac{A + A^*}{2} \right),$$

and  $F(Ae^{i\theta})$  is same as  $F(A)$  rotated  $\theta$  radians in the counter clock-wise direction together imply the eigenvalue optimization characterization

$$r(A) = \max_{\theta \in [0, 2\pi]} \lambda_1(\mathcal{A}(\theta))$$

where  $\mathcal{A}(\theta) := (Ae^{i\theta} + A^*e^{-i\theta})/2$ .

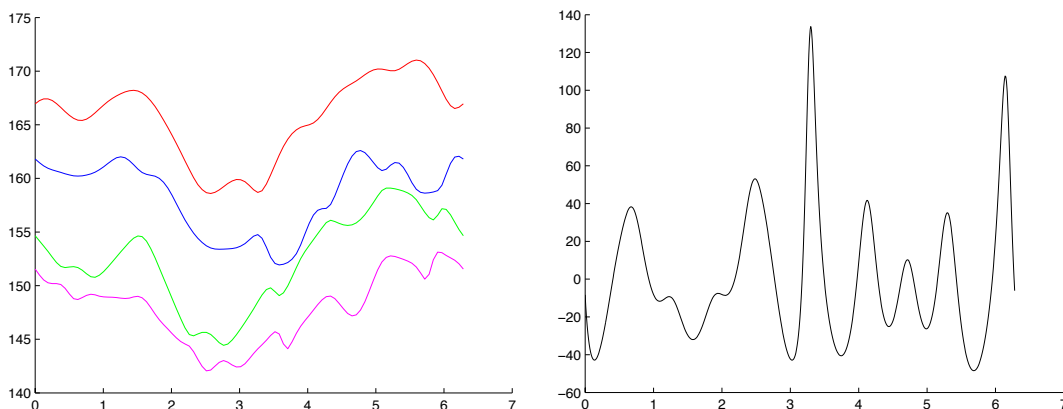


Figure 3: On the left the four largest eigenvalues of  $\mathcal{A}(\theta)$  are plotted on  $[0, 2\pi]$  for a Poisson-random matrix example. On the right the second derivative of the largest eigenvalue of  $\mathcal{A}(\theta)$  is shown for the same example.

Clearly  $\mathcal{A}(\theta)$  is analytic and Hermitian at all  $\theta$ . If the multiplicity of  $\lambda_1(\mathcal{A}(\theta))$  is one, then it is equal to one of the unordered eigenvalues in Theorem 2.1 in a neighborhood of  $\theta$ . In this case  $\lambda_1(\theta) := \lambda_1(\mathcal{A}(\theta))$  is analytic, and its derivative can be deduced as

$$\frac{d\lambda_1(\theta)}{d\theta} = v_1^*(\theta) \operatorname{Real}(iAe^{i\theta}) v_1(\theta). \quad (26)$$

from (3) where  $v_1(\theta)$  is the analytic unit eigenvector associated with  $\lambda_1(\theta)$ . The bound  $\gamma$  for the second derivative depends on the norm of  $A$ ; it is larger for  $A$  with larger norm.

Here we specifically illustrate the algorithm on matrices

$$A_n = P_n - (n/20) \cdot iR_n$$

of various sizes where  $P_n$  is an  $n \times n$  matrix obtained from a finite difference discretization of the Poisson operator, and  $R_n$  is a random  $n \times n$  matrix with entries selected from a normal distribution with zero mean and unit variance. This is a carefully chosen challenging example, as  $\lambda_1(\theta)$  has many local maxima. The largest four eigenvalues of  $\mathcal{A}(\theta)$  are displayed in Figure 3 on the left when  $n = 100$ . The number of local maxima typically increases as the size of the input matrix increases. Note that the largest four eigenvalues do not intersect each other, consequently all of them are analytic for this particular example. The plot of the second derivative is also given in Figure 3 on the right. For the particular example the second derivative varies in between -49 and 134, and  $\|A_{100}\| = 241$ .

The number of function evaluations by our one-dimensional algorithm applied to calculate  $r(A_n)$  for  $n = 100, 400, 900$  are summarized in Table 1. The cpu-times in seconds are also provided in Table 1 in paranthesis. We set  $\gamma = \|A_n\|$  for each  $A_n$ , even though this is a gross over-estimate. Smaller  $\gamma$  values obviously require fewer function evaluations. For instance for the last line in Table 1 we choose  $\gamma = \|A_{900}\| = 2689$ , whereas in reality the second derivative never drops below  $-300$ . Thus choosing  $\gamma = 300$  yields exactly the same numerical radius value for  $\epsilon = 10^{-12}$  but after 41 function evaluations and 88 seconds of cpu-time (instead of

$n / \epsilon$	$10^{-4}$	$10^{-6}$	$10^{-8}$	$10^{-10}$	$10^{-12}$
100	45 (1.0)	54 (1.2)	64 (1.4)	73 (1.6)	81 (1.9)
400	44 (9.0)	54 (10.9)	65 (12.9)	74 (14.6)	83 (17.5)
900	67 (156)	77 (177)	88 (201)	99 (225)	119 (267)

Table 1: Number of function evaluations (or iterations) and cpu-times in seconds (in parenthesis) of the one-dimensional algorithm on the Poisson-random matrices  $A_n$  of various sizes

$\epsilon$	$10^{-1}$	$10^{-2}$	$10^{-3}$	$10^{-4}$
<b># of func. eval.</b>	573	1427	4013	11591

Table 2: Number of function evaluations (or iterations) for the Piyavskii-Shubert algorithm on the matrix  $A_{100}$  for various accuracy

119 function evaluations and 267 seconds). But it may be difficult to know such a value of  $\gamma$  in advance.

The notable thing from Table 1 is that the asymptotic rate of convergence appears to be linear, i.e., every two decimal digit accuracy requires about ten additional iterations. For instance the number of iterations required by the Piyavskii-Shubert algorithm applied to calculate  $r(A_n)$  for  $n = 100$  to reach  $\epsilon = 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}$  accuracy are listed in Table 2. The Piyavskii-Shubert algorithm requires a global Lipschitz constant for the eigenvalue function. Here we choose it as  $\gamma = \|A_{100}\|$ , i.e., the expression (26) implies that in the worst case the derivative of the eigenvalue function can be as large as  $\|A_{100}\|$ . Clearly Table 2 indicates sub-linear convergence for the Piyavskii-Shubert algorithm. Significantly more iterations are required to reach  $10^{-3}$  accuracy from  $10^{-2}$  as compared to the number of iterations required to reach  $10^{-2}$  accuracy from  $10^{-1}$  accuracy. For the optimization of a Lipschitz continuous function with Lipschitz constant  $\gamma$  the first thing that comes to mind is a brute-force grid search. The idea is to split an interval of length  $\ell$  containing a global minimizer or global maximizer into equal sub-intervals each of length  $\ell(\gamma/(2\epsilon))$ . Then evaluating the function at all grid-point and taking the smallest would guarantee that error cannot exceed  $\epsilon$ . For  $A_{100}$  this naive idea would require more than  $10^{14}$  function evaluations with  $\epsilon = 10^{-12}$ .

The algorithm described in [27] for the numerical radius is one of the most reliable techniques at a reasonable cost at the moment. It is not based on derivatives, rather it is based on finding the level sets of  $\lambda_1(\theta)$ . The results for the numerical radius of  $A_n$  listed in the last column of Table 1 match with the algorithm in [27] up to 12 decimal digits. But the specialized algorithm in [27] appears to be slower as indicated by Table 3.

$n$	100	400	900
<b>cpu-time</b>	1.9	48	603

Table 3: cpu-times in second for the algorithm in [27] on  $A_n$  for various  $n$

## 7.2 $H_\infty$ norm

One of the two most widely-used norms in practice for the time-invariant linear control system

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

is the  $H_\infty$  norm (with the other common norm being the  $H_2$  norm). Above  $u(t)$  is called the control input,  $y(t)$  is called the output, and  $A \in \mathbb{C}^{n \times n}$ ,  $B \in \mathbb{C}^{n \times m}$ ,  $C \in \mathbb{C}^{p \times n}$ ,  $D \in \mathbb{C}^{p \times m}$  with  $m, p \leq n$  are the system matrices. We say that the system above (more precisely the state-space description of the system) is of order  $n$ . In the Laplace domain this system with  $x(0) = 0$  can be represented as

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

where  $U(s), Y(s)$  denote the Laplace transformations of  $u(t), y(t)$ , respectively, and  $H(s) := (C(s)I - A)^{-1}B(s) + D(s)$  is called the transfer function of the system. The  $H_\infty$  norm of the transfer function is defined as

$$\sup_{s \in \mathbb{R}} \sigma_1(H(is)),$$

and is same as the  $\infty$  norm of the operator that maps  $u(t)$  to  $y(t)$  in the time domain. For instance in  $H_\infty$  model reduction [10, 2] the purpose is to find a smaller system of order  $r$  such that the operator of the reduced-order system is as close to the operator of the original systems as possible with respect to the  $\infty$ -norm.

The  $H_\infty$  norm is well-defined only when  $A$  is stable, i.e., all of its eigenvalues lie on the left half of the complex plane. In this case the matrix function  $\mathcal{A}(s) := H(is)$  is analytic over the real line. Whenever  $\sigma_1(s) := \sigma_1(H(is))$  is of multiplicity one and non-zero, the singular value  $\sigma_1(s)$  matches with one of the unordered and unsigned analytic singular values discussed in Section 2.3 in a small neighborhood of  $s$ . Then its derivative is given by

$$\frac{d\sigma_1}{ds} = \text{Imag}(u_1(s)^* C(siI - A)^{-2} B v_1(s))$$

from expression (11) where  $u_1(s), v_1(s)$  is a consistent pair of unit left and right singular vectors associated with  $\sigma_1(s)$ .

We experiment with the system  $(A_n, B_n, C_n, D_n)$  of order  $n$  for various values of  $n$  resulting from a finite difference discretization of the heat equation with a control input and a control output [22, Example 3.2]. We slightly perturb  $A_n$  in each case so that the optimization problem becomes more challenging. In Figure 4 the function  $\sigma_1(s)$  is displayed together with  $(s_*, \sigma_1(s_*))$  marked by an asterisk where  $s_*$  is the computed global maximizer of  $\sigma_1(s)$ . Figure 5 displays the second derivative of  $\sigma_1(s)$ , which seems to lie in the interval  $[-11, 3]$ .

The number of iterations and cpu-times required by the one-dimensional algorithm for a perturbed variant of the control system  $(A_n, B_n, C_n, D_n)$  resulting from the heat equation of order  $n = 100, 200, 400, 800$  are listed in Table 4. Here we set  $\gamma = \|A^{-1}\| = \frac{1}{\sigma_n(A)}$ . Once again the algorithm appears to be converging linearly, i.e., every two decimal digit accuracy requires about a fixed number additional function evaluations. Piyavskii-Shubert algorithm again converges sub-linearly; for instance for the system of order  $n = 100$  the number of function evaluations necessary to reach  $\epsilon = 10^{-2}, 10^{-4}, 10^{-6}$  accuracy are 71, 635, 5665, respectively.

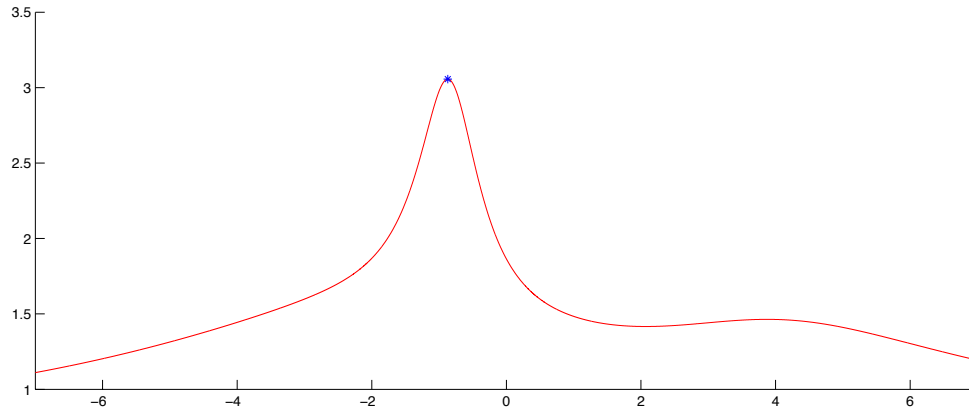


Figure 4: The plot of  $\sigma_1(s) := \sigma_1(H(is))$  for the dynamical system resulting from the Heat equation together with  $(s_*, \sigma_1(s_*))$  marked by an asterisk where  $s_*$  is the computed global maximizer

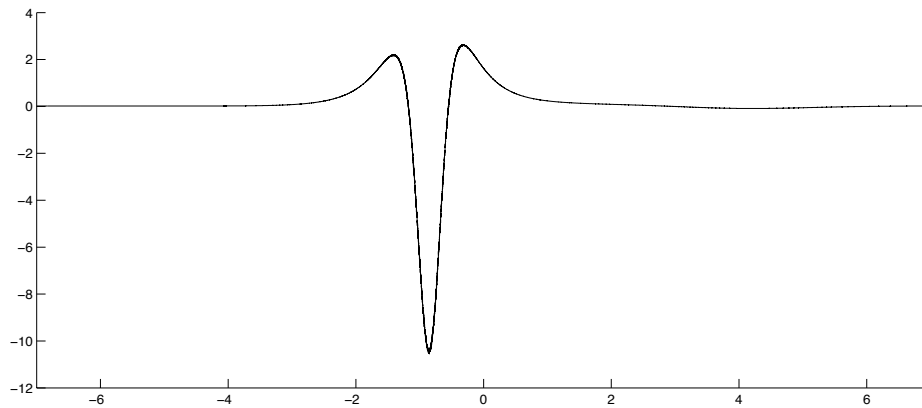


Figure 5: The plot of the second derivative of  $\sigma_1(s) := \sigma_1(H(is))$  for the dynamical system resulting from the Heat equation

$n / \epsilon$	$10^{-4}$	$10^{-6}$	$10^{-8}$	$10^{-10}$
100	23 (0.3)	32 (0.5)	39 (0.5)	47 (0.6)
200	22 (1.5)	29 (1.9)	36 (2.3)	44 (2.8)
400	18 (8.3)	24 (10.8)	29 (12.9)	34 (17.6)
800	16 (53)	19 (63)	22 (73)	27 (92)

Table 4: Number of function evaluations (or iterations) and cpu-times in seconds (in parenthesis) of the one-dimensional algorithm for calculating the  $H_\infty$  norm of the control systems resulting from the heat equation

### 7.3 Distance to Uncontrollability

The controllability of a time-invariant linear control system means that the system can be driven into any state at a particular time by some input  $u(t)$ . This property solely depends on the differential part

$$x'(t) = Ax(t) + Bu(t)$$

of the state-space description from the previous subsection, and could be equivalently characterized as

$$\text{rank} \left( \begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix} \right) = n$$

or

$$\text{rank} \left( \begin{bmatrix} A - zI & B \end{bmatrix} \right) = n \quad \forall z \in \mathbb{C}.$$

The controllability is a fundamental property just as stability. For instance if a system is not controllable, then it is not minimal in the sense that there are systems of smaller order mapping the input to the output in exactly the same manner as the original system.

Paige [28] suggested the distance to uncontrollability

$$\inf \left\{ \left\| \begin{bmatrix} \Delta A & \Delta B \end{bmatrix} \right\|_2 : x'(t) = (A + \Delta A)x(t) + (B + \Delta B)u(t) \right\}$$

as a robust measure of controllability. This problem has the eigenvalue optimization characterization [12]

$$\min_{z \in \mathbb{C}} \sigma_n \left( \begin{bmatrix} A - zI & B \end{bmatrix} \right).$$

As in the previous subsection, since the matrix function  $\mathcal{A}(z) := \begin{bmatrix} A - zI & B \end{bmatrix}$  is analytic, we conclude that  $\sigma_n(\omega_1, \omega_2) := \sigma_n(\mathcal{A}(\omega + i\omega_2))$  is differentiable and analytic along every line in  $\mathbb{R}^2$  whenever it is non-zero and is of multiplicity one. Let  $u_n(\omega) \in \mathbb{C}^n$ ,  $v_n(\omega) = \begin{bmatrix} \tilde{v}_n(\omega) \\ \hat{v}_n(\omega) \end{bmatrix} \in \mathbb{C}^{n+m}$  with  $\tilde{v}_n(\omega) \in \mathbb{C}^n$ ,  $\hat{v}_n(\omega) \in \mathbb{C}^m$  be a consistent pair of unit left and right singular vectors associated with  $\sigma_n(\omega)$ . Then by (11) the gradient is given by

$$\nabla \sigma_n(\omega) = (-\text{Real}(u_n^*(\omega)\tilde{v}_n(\omega)), \text{Imag}(u_n^*(\omega)\tilde{v}_n(\omega)))$$

The level sets of the function  $\sigma_n(\omega)$  are shown in Figure 6 for the perturbed control system resulting from the heat equation of the previous subsection of order  $n = 30$ . Clearly the function is highly non-convex with multiple local minima. We apply Algorithm 2 to calculate the distance to uncontrollability for the heat equation examples of the previous subsection. The number of function evaluations and cpu-times in seconds are listed for the systems of order  $n = 100, 200, 400$  in Table 5. In all cases we set  $\gamma = 2$ .

The rate of convergence is not a very meaningful criterion for Algorithm 2, since it is mesh-based. But we again observe that every two decimal digit accuracy does not increase the number of function evaluations significantly. A brute-force grid based method for a Lipschitz function with Lipschitz constant  $\gamma$  on a rectangle of size  $\ell_1 \times \ell_2$  would require  $(\ell_1 \cdot \gamma) \times (\ell_2 \cdot \gamma) / (2 \cdot \epsilon^2)$  function evaluations for  $\epsilon$  accuracy. For the heat example and with tolerance  $\epsilon = 10^{-8}$  a brute-force grid approach would amount to more than  $10^{18}$  function evaluations. None of the existing algorithms that we are aware of, such as [9, 15, 16, 17], is capable of solving a  $400 \times 400$  example to half of the precision at a reasonable time.

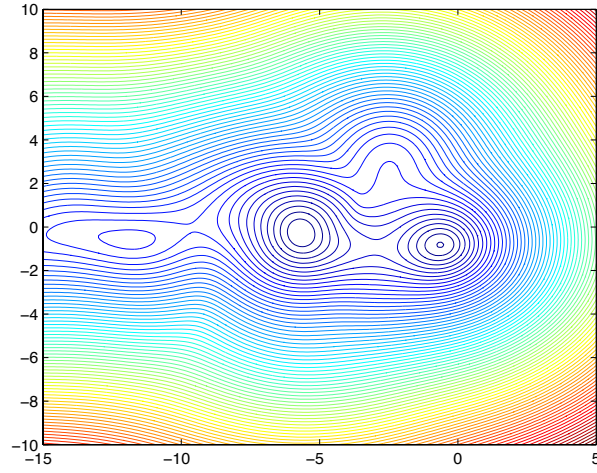


Figure 6: The level-sets of the function  $\sigma_n(\omega_1, \omega_2) := \sigma_n(\mathcal{A}(\omega_1 + i\omega_2))$  for the distance to uncontrollability from the perturbed control system of order  $n = 30$  resulting from the heat equation are displayed.

$n / \epsilon$	$10^{-2}$	$10^{-4}$	$10^{-6}$	$10^{-8}$
100	345 (38)	548 (56)	747 (73)	850 (82)
200	456 (53)	569 (65)	767 (84)	1066 (113)
400	615 (315)	734 (374)	849 (427)	1047 (521)

Table 5: Number of function evaluations (or iterations) and cpu-times in seconds (in paranthesis) of Algorithm 2 for calculating the distances to uncontrollability from the control systems resulting from the heat equation

## 7.4 Distance to Defectiveness

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

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

is mentioned in the book of [36] as a possible measure of the sensitivity of the worst-conditioned eigenvalue of  $A$ . Later it is confirmed [31, 37] that indeed the distance to defectiveness from  $A$  is large if and only if  $A$  has a highly sensitive eigenvalue. For this distance Malyshev [25] deduced the eigenvalue optimization characterization

$$\min_{\lambda \in \mathbb{C}} \max_{\gamma \in [0,1]} \sigma_{2n-1} \left( \begin{bmatrix} A - \lambda I & \gamma I \\ 0 & A - \lambda I \end{bmatrix} \right).$$

Unlike the problems in the previous subsections the eigenvalue characterization is in the min-max form. But Algorithm 2 is still applicable. The function that we need to minimize is defined in terms of the functions

$$f_j(\lambda_1, \lambda_2) := \max_{\gamma \in [0,1]} \tilde{\lambda}_j(\mathcal{A}(\lambda_1, \lambda_2, \gamma)) \quad (27)$$

for  $j = 1, \dots, 4n$  where

$$\mathcal{A}(\lambda_1, \lambda_2, \gamma) := \begin{bmatrix} 0 & \mathcal{B}(\lambda_1, \lambda_2, \gamma) \\ \mathcal{B}^*(\lambda_1, \lambda_2, \gamma) & 0 \end{bmatrix}$$

with

$$\mathcal{B}(\lambda_1, \lambda_2, \gamma) := \begin{bmatrix} A - (\lambda_1 + i\lambda_2)I & \gamma I \\ 0 & A - (\lambda_1 + i\lambda_2)I \end{bmatrix}.$$

It follows from the derivation of the eigenvalue optimization characterization (see [25]) that the maximization problem in (27) for all  $f_j(\lambda)$  such that  $|f_j(\lambda)| \leq \sigma_{2n-1}(\lambda)$  are unimodal where

$$\sigma_{2n-1}(\lambda_1, \lambda_2) := \max_{\gamma \in [0,1]} \sigma_{2n-1} \left( \begin{bmatrix} A - (\lambda_1 + i\lambda_2)I & \gamma I \\ 0 & A - (\lambda_1 + i\lambda_2)I \end{bmatrix} \right).$$

This means that all such  $f_j(\lambda)$  are differentiable and analytic along every line in  $\mathbb{R}^2$ . These are the pieces defining the function  $\sigma_{2n-1}$  at  $\lambda$ . By continuity they remain to be the defining functions for  $\sigma_{2n-1}$  in a neighborhood of  $\lambda$ . Perform an analytic extension to  $f_j$  if necessary outside of this neighborhood. At other  $\lambda$  outside the neighborhood defining functions may be different. But there are finitely many such neighborhoods on a bounded domain. Consequently,  $\sigma_{2n-1}(\lambda)$  is defined by finitely many functions each of which is differentiable and analytic along every line in  $\mathbb{R}^d$ .

In practice the defining functions remain the same inside the box for all  $\lambda \in \mathcal{B}$ . Indeed it seems reasonable to assume that  $\sigma_{2n-1}$  is defined only by one function  $f_j(\lambda)$  inside  $\mathcal{B}$  excluding the non-generic cases. Then the function  $\sigma_{2n-1}(\lambda)$  is differentiable and analytic along lines in  $\mathbb{R}^2$ . Suppose that the maximization problem (27) is attained at  $\gamma_*$ . From [4] we have

$$\nabla \sigma_{2n-1}(\lambda) = \nabla f_j(\lambda) = \left( \frac{\partial \tilde{\lambda}_j(\mathcal{A}(\lambda, \gamma_*))}{\partial \lambda_1}, \frac{\partial \tilde{\lambda}_j(\mathcal{A}(\lambda, \gamma_*))}{\partial \lambda_2} \right).$$

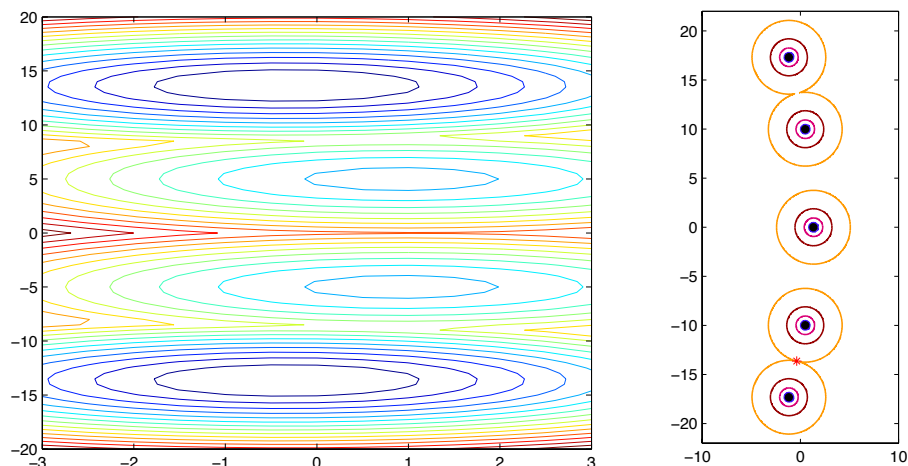


Figure 7: (Left) the level-sets of the function  $\sigma_{2n-1}(\lambda_1, \lambda_2)$  that needs to be minimized for the distance to defectiveness for a  $5 \times 5$  penta-diagonal Toeplitz matrix; (Right) the  $\epsilon$  pseudospectrum of the same matrix for  $\epsilon$  equal to its distance to defectiveness together with asterisks marking the defective eigenvalue of the nearest matrix

It follows from (11) that

$$\nabla \sigma_{2n-1}(\lambda) = (-\text{Real}(u_{2n-1}^*(\lambda)v_{2n-1}(\lambda)), \text{Imag}(u_{2n-1}^*(\lambda)v_{2n-1}(\lambda)))$$

where  $u_{2n-1}(\lambda), v_{2n-1}(\lambda)$  is a consistent pair of unit left and right singular vectors associated with

$$\sigma_{2n-1}(\lambda) := \sigma_{2n-1} \left( \begin{bmatrix} A - \lambda I & \gamma_* I \\ 0 & A - \lambda I \end{bmatrix} \right).$$

The level sets of the function  $\sigma_{2n-1}(\lambda)$  is displayed for the  $5 \times 5$  matrix  $T = \text{diag}(1, -10, 0, 10, 1)$  on the left in Figure 7 which reveals the non-convex nature of the function to be minimized. It is well-known that the distance from  $A$  to the nearest defective matrix is related to the  $\epsilon$ -pseudospectrum of  $A$  defined as

$$\Lambda_\epsilon(A) := \bigcup_{\|X-A\|_2 \leq \epsilon} \Lambda(X),$$

where  $\Lambda(\cdot)$  denotes the spectrum of its argument. For an  $n \times n$  matrix with distinct eigenvalues this set consists of  $n$  disconnected components, one component around each eigenvalue. The distance from  $A$  to the nearest defective matrix is the smallest  $\epsilon$  such that two components of  $\Lambda_\epsilon(A)$  coalesce [1, 7]. Furthermore the point of coalescence is the defective eigenvalue of the nearest defective matrix. For the  $5 \times 5$  example  $T$  the  $\epsilon$ -pseudospectrum is displayed on the right in Figure 7 for various  $\epsilon$ . The outer-most curves represent the boundary of  $\Lambda_\epsilon(T)$  for  $\epsilon = 3.753$ , the computed distance to defectiveness by Algorithm 2. Two components of the outer-most curve coalesce at  $\lambda_* = -0.336 - i13.6$  marked by an asterisks, which is the computed defective eigenvalue of the nearest matrix.

The inner minimization problems are solved by means of the secant method, which requires the derivatives with respect to  $\gamma$  for a fixed  $\lambda$ . Analytic expressions can again be derived from (11) for these derivatives. The distance to the nearest defective matrix is same as the distance to the nearest matrix with a multiple eigenvalue. The reason is that for any matrix  $A$  with a multiple eigenvalue, there are defective matrices arbitrarily close to  $A$ . In [26] eigenvalue optimization characterizations were derived for the more general problem, the distance to the nearest matrix with an eigenvalue of specified algebraic multiplicity. The discussions in this subsection extends for the numerical solution of these more general distances as well.

## 8 Software

Matlab implementations of the one-dimensional algorithm and the 2-dimensional version of Algorithm 2 are available on the web<sup>1</sup>. The user of the routines is expected write down a matlab routine calculating the eigenvalue function as well as its derivative, or gradient in the 2-dimensional case, at a given point. The user must also provide  $\gamma$ , an upper bound on the second derivatives in absolute value.

## 9 Conclusion

We introduced the first generic algorithm tailored for the optimization of the eigenvalues of a Hermitian matrix function depending on its parameters analytically. The algorithm is guaranteed to converge to a global optimizer. In practice we observe linear convergence unlike other global optimization algorithms exploiting the Lipschitzness or boundedness of the function and converging sub-linearly. This is largely due to the fact that the algorithm makes use of the derivatives, and constructs piece-wise quadratic models.

The computational difficulty with the algorithm is that in the multi-dimensional case negative definite quadratic functions need to be minimized subject to linear constraints. This problem is NP-hard, but the solution is guaranteed to be attained at one of the vertices of its feasible region. In small dimensions these quadratic problems can be solved efficiently; but in high dimensions they are not tractable. A Matlab implementation of the 2-dimensional algorithm is available. We plan to extend the implementation for up to 4 or 5-dimensional case. But for higher dimensional problems the negative definite quadratic programs need to be replaced by positive definite programs. This is a subject that we hope to tackle in the near future.

## References

- [1] R. Alam and S. Bora. On sensitivity of eigenvalues and eigendecompositions of matrices. *Linear Algebra Appl.*, 396:273–301, 2005.
- [2] A.C. Antoulas. *Approximation of large scale dynamical systems*. SIAM, Philadelphia, 2005.

---

<sup>1</sup><http://home.ku.edu.tr/~emengi/software>

- [3] O. Axelsson, H. Lu, and B. Polman. On the numerical radius of matrices and its application to iterative solution methods. *Linear and Multilinear Algebra*, 37:225–238, 1994.
- [4] J. F. Bonnans and A. Shapiro. *Perturbation Analysis of Optimization Problems*. Springer-Verlag, New York, 2000.
- [5] S. Boyd and V. Balakrishnan. A regularity result for the singular values of a transfer matrix and a quadratically convergent algorithm for computing its  $L_\infty$ -norm. *Systems Control Lett.*, 15(1):1–7, 1990.
- [6] N.A. Bruinsma and M. Steinbuch. A fast algorithm to compute the  $H_\infty$ -norm of a transfer function matrix. *Systems and Control Letters*, 14:287–293, 1990.
- [7] J.V. Burke, A.S. Lewis, and M.L. Overton. Spectral conditioning and pseudospectral growth. *Numer.Math.*, 107:27–37, 2007.
- [8] R. Byers. A bisection method for measuring the distance of a stable to unstable matrices. *SIAM J. Sci. Statist. Comput.*, 9:875–881, 1988.
- [9] R. Byers. The descriptor controllability radius. In *Numerical Methods Proceedings of the International Symposium MTNS-93*, volume II, pages 85–88. Uwe Helmke, Reinhard Mennicken, and Hosef Saurer, eds., Akademie Verlag, Berlin, 1993.
- [10] G. E. Dullerud and F. Paganini. *A Course in Robust Control Theory*. Springer-Verlag, New York, 2000.
- [11] M. Eiermann. Field of values and iterative methods. *Linear Algebra and Its Applications*, 180:167–197, 1993.
- [12] R. Eising. Between controllable and uncontrollable. *Systems Control Lett.*, 4(5):263–264, 1984.
- [13] M. Elad, P. Milanfar, and G. H. Golub. Shape from moments—an estimation theory perspective. *IEEE Trans. Signal Process.*, 52(7):1814–1829, 2004.
- [14] M.A. Freitag and A. Spence. A newton-based method for the calculation of the distance to instability. *Linear Algebra Appl.*, 435(12):3189–3205, 2011.
- [15] M. Gao and M. Neumann. A global minimum search algorithm for estimating the distance to uncontrollability. *Linear Algebra Appl.*, 188-189:305–350, 1993.
- [16] M. Gu. New methods for estimating the distance to uncontrollability. *SIAM J. Matrix Analysis Appl.*, 21(3):989–1003, 2000.
- [17] M. Gu, E. Mengi, M.L. Overton, J. Xia, and J. Zhu. Fast methods for estimating the distance to uncontrollability. *SIAM J. Matrix Anal. Appl.*, 28(2):477–502, 2006.
- [18] C. He and G.A. Watson. An algorithm for computing the distance to instability. *SIAM J. Matrix Anal. Appl.*, 20:101–116, 1999.

- [19] D. Hinrichsen and M. Motscha. Optimization problems in the robustness analysis of linear state space systems. In *Proceedings of the international seminar on Approximation and optimization*, pages 54–78, New York, NY, USA, 1988. Springer-Verlag New York, Inc.
- [20] R.A. Horn and C.R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991.
- [21] D. R. Jones, C. D. Perttunen, and B. E. Stuckman. Lipschitzian optimization without the Lipschitz constant. *J. Optim. Theory Appl.*, 79(1):157–181, 1993.
- [22] D. Kressner, V. Mehrmann, and T. Penzl. Ctdsx - a collection of benchmarks for state-space realizations of continuous-time dynamical systems. *SLICOT Working Note*, 1998.
- [23] D. Kressner, E. Mengi, I. Nakic, and N. Truhar. Generalized eigenvalue problems with specified eigenvalues. *IMA J. Numer. Anal.*, 2011. submitted.
- [24] P. Lancaster. On eigenvalues of matrices dependent on a parameter. *Numer. Math.*, 6:377–387, 1964.
- [25] A. N. Malyshev. A formula for the 2-norm distance from a matrix to the set of matrices with multiple eigenvalues. *Numer. Math.*, 83:443–454, 1999.
- [26] E. Mengi. Locating a nearest matrix with an eigenvalue of prespecified algebraic multiplicity. *Numer. Math.*, 118:109–135, 2011.
- [27] E. Mengi and M.L. Overton. Algorithms for the computation of the pseudospectral radius and the numerical radius of a matrix. *IMA Journal of Numerical Analysis*, 25:648–669, 2005.
- [28] C.C. Paige. Properties of numerical algorithms relating to computing controllability. *IEEE Trans. Automat. Control*, 26:130–138, 1981.
- [29] S. A. Piyavskii. An algorithm for finding the absolute extremum of a function. *USSR Comput. Math. and Math. Phys.*, 12:57–67, 1972.
- [30] F. Rellich. *Perturbation Theory of Eigenvalue Problems*. Gordon and Breach, 1969.
- [31] A. Ruhe. Properties of a matrix with a very ill-conditioned eigenproblem. *Numer. Math.*, 15:57–60, 1970.
- [32] Y.D. Sergeyev and D.E. Kvasov. Global search based on efficient diagonal partitions and a set of lipschitz constants. *SIAM J. on Optimization*, 16(3):910–937, 2006.
- [33] B. Shubert. A sequential method seeking the global maximum of a function. *SIAM J. Numer. Anal.*, 9:379–388, 1972.
- [34] C. F. Van Loan. How near is a matrix to an unstable matrix? *Lin. Alg. and its Role in Systems Theory*, 47:465–479, 1984.
- [35] C.T.C. Wall. *Singular Points of Plane Curves*. Cambridge University Press, Cambridge, 2004.

- [36] J.H. Wilkinson. *The Algebraic Eigenvalue Problem*. Clarendon Press, Oxford, 1965.
- [37] J.H. Wilkinson. Note on matrices with a very ill-conditioned eigenproblem. *Numer. Math.*, 19:176–178, 1972.