

# Extended and improved criss-cross algorithms for computing the spectral value set abscissa and radius

Peter Benner\*      Tim Mitchell†

January 1, 2018

## Abstract

In this paper, we extend the original criss-cross algorithms for computing the  $\varepsilon$ -pseudo-spectral abscissa and radius to general spectral value sets. Furthermore, we greatly reduce the total number of expensive Hamiltonian eigenvalue decompositions needed by performing the horizontal/radial search subphases with a new root-finding-based approach that is both more numerically resilient and substantially faster, particularly for typical spectral value sets. Finally, we propose a new, more robust way of handling singular pencils that can arise when computing the  $\varepsilon$ -spectral value set radius. Compared to would-be direct extensions of the original criss-cross algorithms, that is, without the changes we propose here, our modified methods are more robust and often several times faster, improvements that can also be relevant for the special case of pseudospectra.

## 1 Introduction

Consider the continuous-time linear dynamical system

$$E\dot{x} = Ax + Bu, \quad (1a)$$

$$y = Cx + Du, \quad (1b)$$

where  $A \in \mathbb{C}^{n \times n}$ ,  $B \in \mathbb{C}^{n \times m}$ ,  $C \in \mathbb{C}^{p \times n}$ ,  $D \in \mathbb{C}^{p \times m}$ , and  $E \in \mathbb{C}^{n \times n}$ . Using feedback  $u = \Delta y$ , where  $\Delta \in \mathbb{C}^{m \times p}$ , so that input  $u$  varies linearly with respect to output  $y$ , (1a) can be rewritten as  $E\dot{x} = Ax + B\Delta y$  and (1b) as  $y = (I - D\Delta)^{-1}Cx$ , assuming that  $(I - D\Delta)$  is invertible. It is then clear that the input-output system (1) is equivalent to

$$E\dot{x} = M(\Delta)x, \quad (2)$$

where

$$M(\Delta) := A + B\Delta(I - D\Delta)^{-1}C \quad (3)$$

is called the *perturbed system matrix*. As a consequence, the dynamical properties of (1), which arise in many engineering applications, can be studied by examining the generalized eigenvalue problem of the matrix pencil  $(M(\Delta), E) = \lambda E - M(\Delta)$ .

For the special case of  $B = C = E = I$  and  $D = 0$ , (2) simply reduces to

$$\dot{x} = (A + \Delta)x. \quad (4)$$

---

\*Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg, 39106 Germany (benner@mpi-magdeburg.mpg.de).

†Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg, 39106 Germany (mitchell@mpi-magdeburg.mpg.de).

Considering  $\Delta = 0$ , the ordinary differential equation  $\dot{x} = Ax$  is asymptotically stable if its spectral abscissa, the maximal real part attained by the eigenvalues of matrix  $A$ , is strictly negative:

$$\alpha(A) < 0.$$

However, the spectrum only provides a limited perspective with respect to the dynamics of the system. If matrices close to an asymptotically stable matrix  $A$  have eigenvalues in the right half plane, then the solution of  $\dot{x} = Ax$  may still have large transient behavior before converging. Furthermore, in applications, where  $A$  is merely a model of some physical process or mechanism, the theoretical asymptotic stability of  $A$  may not be predictive of reality, particularly if small perturbations of the model  $A$  can result in unstable systems. Hence, there has been great interest to also consider the dynamical properties of (4), which is characterized by *pseudospectra* [TE05]: the set of eigenvalues of  $A$  under general perturbation, typically limited such that  $\|\Delta\|_2 \leq \varepsilon$  for some fixed choice  $\varepsilon \in \mathbb{R}^+$ . For a given  $\varepsilon \geq 0$ , the  $\varepsilon$ -*pseudospectral abscissa*:

$$\alpha_\varepsilon(A) := \max\{\operatorname{Re} \lambda : \lambda \in \sigma(A + \Delta), \|\Delta\|_2 \leq \varepsilon\},$$

where  $\sigma(\cdot)$  denotes the spectrum, provides a measure of *robust stability*: if  $\alpha_\varepsilon(A) < 0$ , then  $A + \Delta$  is stable for any perturbation such that  $\|\Delta\|_2 \leq \varepsilon$ . The norm of the smallest destabilizing perturbation, that is the value of  $\varepsilon$  that yields  $\alpha_\varepsilon(A) = 0$ , is called the *distance to instability*, introduced by [Van85]. Beyond robust stability measures, it is also possible to understand the transient behaviors of nonlinear processes via (stable) linearizations, provided that one looks beyond their associated linear spectra and instead at their pseudospectra [TTRD93].

Computationally, numerous techniques for plotting the boundaries of pseudospectra are discussed in [Tre99, WT01] while a “criss-cross” algorithm for computing the  $\varepsilon$ -pseudospectral abscissa, with global and local quadratic convergence guarantees, was proposed in [BLO03]. The criss-cross algorithm performs a sequence of alternating vertical and horizontal searches to find relevant boundary points of the  $\varepsilon$ -pseudospectrum along the respective search lines, which converge to a globally rightmost point of the  $\varepsilon$ -pseudospectrum; these vertical and horizontal searches are accomplished by computing eigenvalues of associated Hamiltonian matrix pencils. In fact, the techniques used in the criss-cross algorithm build upon those developed for the first algorithm for computing the distance to instability [Bye88], which actually dates back about fifteen years earlier. The criss-cross algorithm has also been adapted to compute the corresponding  $\varepsilon$ -*pseudospectral radius*:

$$\rho_\varepsilon(A) := \max\{|\lambda| : \lambda \in \sigma(A + \Delta), \|\Delta\|_2 \leq \varepsilon\},$$

relevant for discrete-time systems  $x_{k+1} = Ax_k$ , by using circular and radial searches instead of vertical and horizontal ones [MO05].

For the more general setting of (1), the analog of the  $\varepsilon$ -pseudospectrum is an  $\varepsilon$ -*spectral value set* while the analog of the distance to instability is the *complex stability radius* (perhaps better known by its reciprocal value, the  $\mathcal{H}_\infty$  norm). Spectral value sets are distinctly different from pseudospectra of generalized eigenvalue problems  $\lambda E - A$ , where both  $A$  and  $E$  would be considered under general perturbation. Instead, in spectral value sets, (1) only permits *structured* perturbations of the form  $B\Delta(I - D\Delta)^{-1}C$  to operator  $A$ , while  $E$  remains unperturbed. Fixed matrices  $A$ ,  $B$ ,  $C$ ,  $D$ , and  $E$  represent the certainties of the model while  $\Delta$  represents the uncertainties in the feedback loop. In order to identify dynamical properties of (1), it is natural to consider the worst outcome possible over the set of uncertainties. The complex stability radius encodes just that, as its value is the size of the smallest matrix  $\Delta$  such that  $B\Delta(I - D\Delta)^{-1}C$  destabilizes (1), assuming for now that  $(A, E)$  is stable itself.

Algorithms for computing the complex stability radius (or the  $\mathcal{H}_\infty$  norm) of general systems with input and output, (1) but with  $E = I$ , were first proposed by [BBK89], which was inspired by the Hamiltonian eigenvalue techniques developed in [Bye88] for computing the distance to instability (the special case where  $B = C = E = I$  and  $D = 0$ ). Shortly thereafter, the

now-considered standard algorithm, which converges quadratically and supersedes the earlier bisection methods of [Bye88] and [BBK89], was simultaneously and independently proposed by [BB90] and [BS90]. Extending these algorithms to so-called descriptor systems, where  $E \neq I$ , was addressed in [BSV12].<sup>1</sup> Techniques to accelerate these algorithms were proposed in [GVDV98] and, most recently, in [BM17]. As all these methods share a  $\mathcal{O}(n^3)$  amount of work per iteration complexity, they are typically only tractable choices for computing the  $\mathcal{H}_\infty$  norm of systems which are relatively small. However, while none of these algorithms make use of spectral values sets (at least not directly), spectral values sets have been a crucial computational tool in the recent introduction of methods (see [GGO13, BV14, MO16]) for *approximating* the complex stability radius of large-scale systems, using scalable routines. One of the key components of these new methods has been the development of efficient iterative algorithms for approximating the  $\varepsilon$ -spectral value set abscissa, which was first developed for the special case of approximating the  $\varepsilon$ -pseudospectral abscissa in [GO11].

Although the aforementioned pseudospectral criss-cross algorithm and its related variant for discrete-time systems provide dense algorithmic analogs to the newer sparse approximation techniques of [GO11], to the best of our knowledge corresponding dense algorithms for general spectral value sets have yet to be developed. In this paper, we extend and improve upon the work of [BLO03, MO05] to introduce the first criss-cross type methods for computing both the  $\varepsilon$ -spectral value set abscissa and radius. Some of the techniques we employed in this paper were developed in parallel with our recent efforts to accelerate the computation of the  $\mathcal{H}_\infty$  norm [BM17]. By taking advantage of specific properties of typical input-output systems, the algorithms we propose are faster than what would be achieved by a straightforward extension of [BLO03, MO05]. In fact, our efficiency improvements are also relevant to the pseudospectral case. We also introduce a new, more robust procedure for handling singular pencils that may arise when computing the  $\varepsilon$ -spectral value set radius. Finally, our improved algorithms are also more resilient to numerical issues in practice than the preceding methods.

The paper is organized as follows. In Section 2, we present the prerequisite definitions and theory for spectral value sets. Our new criss-cross algorithm for computing the  $\varepsilon$ -spectral value set abscissa is presented in Section 3 while Section 4 describes the necessary modifications to compute the  $\varepsilon$ -spectral value set radius instead, including new results for handling singular pencils. We present convergence results in Section 5. Numerical results are given in Section 6 with concluding remarks made in Section 7.

## 2 Spectral value sets and the transfer function

We now formally define the  $\varepsilon$ -spectral value set, its associated  $\varepsilon$ -spectral value set abscissa and radius, and give related theoretical results that will be needed in this paper. In addition, we discuss some of the new issues that arise when considering the dynamical behavior and stability of input-output systems, with possibly singular  $E$  matrices.

**Definition 2.1.** *Let  $\varepsilon \geq 0$  be such that  $\varepsilon \|D\|_2 \leq 1$  and define the  $\varepsilon$ -spectral value set*

$$\sigma_\varepsilon(A, B, C, D, E) = \bigcup \{ \sigma(M(\Delta), E) : \Delta \in \mathbb{C}^{m \times p}, \|\Delta\|_2 \leq \varepsilon \}. \quad (5)$$

Now consider the *transfer function* associated with input-output system (1):

$$G(\lambda) := C(\lambda E - A)^{-1} + D \quad \text{for } \lambda \in \mathbb{C} \setminus \sigma(A, E). \quad (6)$$

As shown in [HP05, Section 5.2] for  $E = I$ , spectral value sets can be equivalently defined in terms of the norm of the transfer function, instead of eigenvalues of  $(M(\Delta), E)$ . This fundamental

---

<sup>1</sup> Note that [BSV12] actually refers to computing the  $\mathcal{L}_\infty$  norm, not the  $\mathcal{H}_\infty$  norm, but the only difference between the two measures is the class of cases where an infinite value is attained. Determining this can be considered a preprocessing step to the convergent part of the algorithm, which is otherwise the same for both stability measures.

result easily extends to the case of generic  $E$  matrices we consider here; e.g. the proof of [GGO13, Theorem 2.1] readily generalizes by substituting all occurrences of  $(\lambda I - A)$  with  $(\lambda E - A)$ .

**Theorem 2.2.** *Let  $\varepsilon \geq 0$  be such that  $\varepsilon\|D\|_2 < 1$  and  $\|\Delta\|_2 \leq \varepsilon$  so that  $I - D\Delta$  is invertible. Then for  $\lambda \notin \sigma(A, E)$  the following are equivalent:*

$$\|G(\lambda)\|_2 \geq \varepsilon^{-1} \quad \text{and} \quad \lambda \in \sigma(M(\Delta), E) \quad \text{for some } \Delta \text{ with } \|\Delta\|_2 \leq \varepsilon. \quad (7)$$

By Theorem 2.2, the following corollary is immediate, providing an alternate spectral value set definition based on the norm of the transfer function.

**Corollary 2.3.** *Let  $\varepsilon \geq 0$  be such that  $\varepsilon\|D\|_2 < 1$ . Then*

$$\sigma_\varepsilon(A, B, C, D, E) = \sigma(A, E) \bigcup \{\lambda \in \mathbb{C} \setminus \sigma(A, E) : \|G(\lambda)\|_2 \geq \varepsilon^{-1}\}. \quad (8)$$

Note that the nonstrict inequalities in Definition 2.1 and Theorem 2.2 imply that the spectral value sets we consider are compact. Furthermore, the boundary of  $\sigma_\varepsilon(A, B, C, D, E)$  is characterized by the condition  $\|G(\lambda)\|_2 = \varepsilon^{-1}$  while for any matrix  $\Delta$  such that  $\lambda \in \sigma(M(\Delta), E)$  is a boundary point,  $\|\Delta\|_2 = \varepsilon$  must hold (though the reverse implication is not necessarily true).

**Lemma 2.4.** *Let  $\varepsilon > 0$  be such that  $\varepsilon\|D\|_2 < 1$  and let  $\lambda$  be a non-isolated boundary point of an  $\varepsilon$ -spectral value set, with associated perturbation matrix  $\Delta$ , that is,  $\lambda \in \sigma(M(\Delta), E)$ . Then for one or more  $\lambda_0 \in \sigma(A, E)$ , there exists a continuous path parameterized by  $t \in [0, 1]$  such that  $\lambda(t)$  is an eigenvalue of  $\sigma(M(t\Delta), E)$  taking  $\lambda(0) = \lambda_0$  to  $\lambda(1) = \lambda$ . Furthermore,  $\lambda(t)$  is only a boundary point at  $t = 1$ .*

*Proof.* By continuity of eigenvalues, the continuous path  $\lambda(t)$  exists and clearly,  $\|t\Delta\|_2 < \|\Delta\|_2$  holds for  $t \in [0, 1)$ . As  $\lambda$  is on the boundary,  $\|\Delta\|_2 = \varepsilon$  holds but then the necessary condition for  $\lambda(t)$  to be a boundary point is violated for all  $t \in [0, 1)$ .  $\square$

## 2.1 The spectral value set abscissa and radius

The  $\varepsilon$ -spectral value set abscissa, relevant for continuous-time systems (1), is formally defined as follows.

**Definition 2.5.** *Let  $\varepsilon \geq 0$  be such that  $\varepsilon\|D\|_2 < 1$  and define the  $\varepsilon$ -spectral value set abscissa*

$$\alpha_\varepsilon(A, B, C, D, E) := \max\{\operatorname{Re} \lambda : \lambda \in \sigma_\varepsilon(A, B, C, D, E)\}. \quad (9)$$

Now consider the discrete-time linear dynamical system

$$Ex_{k+1} = Ax_k + Bu_k \quad (10a)$$

$$y_k = Cx_k + Du_k, \quad (10b)$$

where the matrices are defined as before in (1). For the special case of  $B = C = E = I$  and  $D = 0$ , the simple ordinary difference equation  $x_{k+1} = Ax_k$  is stable if and only if its *spectral radius*, the maximal modulus attained by the eigenvalues of  $A$ , is strictly less than one:

$$\rho(A) < 1.$$

Thus, for discrete-time input-output systems of the form of (10), the relevant generalization of the  $\varepsilon$ -pseudospectral radius is defined as follows.

**Definition 2.6.** *Let  $\varepsilon \geq 0$  be such that  $\varepsilon\|D\|_2 < 1$  and define the  $\varepsilon$ -spectral value set radius*

$$\rho_\varepsilon(A, B, C, D, E) := \max\{|\lambda| : \lambda \in \sigma_\varepsilon(A, B, C, D, E)\}. \quad (11)$$

Unlike the pseudospectral case, it may also be preferable to consider modified definitions of the  $\varepsilon$ -spectral value set abscissa and radius. For continuous-time systems, if all the finite eigenvalues of  $(A, E)$  lie in the open left half plane, then  $(A, E)$  is a stable matrix pencil. Thus, there can be a motivation to exclude all infinite eigenvalues of  $(A, E)$  from consideration in computing the  $\varepsilon$ -spectral value set abscissa. However, if  $(A, E)$  has index more than one, where  $\text{rank}(E)$  is less than the number of finite eigenvalues of  $(A, E)$ , then infinitesimally small perturbations can be constructed which destabilize the pencil [BN93, Section 2]. In this case,  $(A, E)$  would be on the boundary of instability and would thus be essentially unstable, if not technically. As such, it can also be considered appropriate to sometimes include the infinite eigenvalues, so that the value of the  $\varepsilon$ -spectral value set abscissa would be  $\infty$ , for the sake of distinguishing such systems. Further complicating matters, for input-output systems of the form (1), eigenvalues may or may not be relevant depending whether they are *controllable* and/or *observable*, concepts which we now define.

**Definition 2.7.** *Let  $\lambda$  be a finite eigenvalue of the matrix pencil  $(A, E)$  from an input-output system. Eigenvalue  $\lambda$  is observable if  $Cx \neq 0$  holds for all of its right eigenvectors  $x$ , i.e.  $Ax = \lambda Ex, x \neq 0$ . Eigenvalue  $\lambda$  is controllable if  $B^*y \neq 0$  holds for all of its left eigenvectors  $y$ , i.e.  $y^*A = \lambda y^*E, y \neq 0$ .*

In a sense, the presence of uncontrollable and/or unobservable eigenvalues can be considered an artifact of redundancy in a specific system design. Any associated transfer function  $G(\lambda)$  of (1) can be reduced to what is called a *minimal realization*  $\widehat{G}(\lambda)$ , assuming  $G(\lambda)$  is not already minimal; e.g. see [Dai89, Theorem 2-6.3]. The  $A$  and  $E$  matrices of  $\widehat{G}(\lambda)$  are of minimal possible dimension so that the reduced transfer function is unaltered and its input-output behavior remains identical to  $G(\lambda)$ .

In terms of spectral value sets, consider an eigenvalue  $\lambda$  of  $(A, E)$  with right and left eigenvectors  $x$  and  $y$ . If  $\lambda$  is unobservable or uncontrollable, then  $Cx = 0$  or  $B^*y = 0$  respectively holds, and thus for any perturbation matrix  $\Delta \in \mathbb{C}^{m \times p}$ , either  $M(\Delta)x = Ax$  or  $y^*M(\Delta) = y^*A$  holds. Furthermore, if  $\lambda$  is a simple eigenvalue, then for sufficiently small  $\varepsilon > 0$ ,  $\lambda$  must be an isolated point of  $\sigma_\varepsilon(A, B, C, D, E)$ : letting  $\lambda(t)$  be some parameterization of  $\lambda$  with  $t \in \mathbb{R}$  and  $\lambda(0) = \lambda$ , via standard perturbation theory for simple eigenvalues, it is easily seen that  $\lambda'(0) = 0$  holds.

Despite the variety of  $\varepsilon$ -spectral value set abscissa or radius definitions one might be interested in, a specific choice merely dictates the initialization of the algorithms we present here while the main convergent phase will remain applicable to all cases without modification. For the remainder of the paper, we simply assume that the desired variant of the spectral value set is defined by parameters provided by the user, e.g. tolerances for eliminating uncontrollable/unobservable eigenvalues, whether or not a minimal realization is provided, the system's index is greater than one, etc.

## 2.2 Derivatives of the norm of the transfer function

As we will utilize first- and second-order information of the norm of the transfer function in several different ways, it will be less repetitious to establish the following somewhat abstract and out-of-context derivative results now and then apply them later in specific cases as needed. For technical reasons, we will first need the following assumption.

**Assumption 2.8.** *Let  $\varepsilon > 0$  with  $\varepsilon\|D\|_2 < 1$  and let  $\lambda \in \sigma_\varepsilon(A, B, C, D, E)$  with  $\lambda \notin \sigma(A, E)$ . Then the largest singular value of  $G(\lambda)$  is simple.*

**Remark 2.9.** *In fact, it can be shown that generically, that is, for all almost all quintuplets  $(A, B, C, D, E)$ , the largest singular value of  $G(\lambda)$  is indeed simple for all  $\lambda \in \mathbb{C} \setminus \sigma(A, E)$ ; e.g. see [BLO03, Section 2] for pseudospectra and [GG013, Remark 2.20] for general spectral value sets with  $E = I$ . Although counter examples can be constructed (see [GG013, Remark 2.20]), with*

probability one such examples will not be encountered in practice and as such, this technicality does not pose a problem for the algorithms we propose here.

Let  $\lambda(t) \in \mathbb{C}$  be parameterized with respect to  $t \in \mathbb{R}$  and  $Z(t) = \lambda(t)E - A$ . Then consider  $G \circ \lambda$ :

$$G(\lambda(t)) = C(\lambda(t)E - A)^{-1}B + D = CZ(t)^{-1}B + D. \quad (12)$$

By standard (matrix) differentiation techniques, we have that:

$$(G \circ \lambda)'(t) := -\lambda'(t)CZ(t)^{-1}EZ(t)^{-1}B \quad (13a)$$

$$(G \circ \lambda)''(t) := 2\lambda'(t)^2CZ(t)^{-1}EZ(t)^{-1}EZ(t)^{-1}B - \lambda''(t)CZ(t)^{-1}EZ(t)^{-1}B. \quad (13b)$$

Furthermore, let  $s(t)$  be the largest singular value of (12), i.e.  $\|G(\lambda(t))\|_2$ , with associated left and right singular vectors  $u(t)$  and  $v(t)$ . Assuming that  $s(t)$  is a simple singular value at say,  $t = 0$ , by standard perturbation theory, it follows that

$$s'(0) = \operatorname{Re}(u(0)^* [(G \circ \lambda)'(0)] v(0)) \quad (14a)$$

$$= -\operatorname{Re}(u(0)^* [\lambda'(0)CZ(0)^{-1}EZ(0)^{-1}B] v(0)). \quad (14b)$$

To compute  $s''(0)$ , we need the following result for the second derivative of eigenvalues, which can be found in various forms in [Lan64], [OW95], and [Kat82].

**Theorem 2.10.** *For  $t \in \mathbb{R}$ , let  $H(t)$  be a twice-differentiable  $n \times n$  Hermitian matrix family with distinct eigenvalues at  $t = 0$  with  $(\lambda_k, x_k)$  denoting the  $k$ th such eigenpair and where each eigenvector  $x_k$  has unit norm and the eigenvalues are ordered  $\lambda_1 > \dots > \lambda_n$ . Then:*

$$\lambda_1''(t) \Big|_{t=0} = x_1^* H''(0) x_1 + 2 \sum_{k=2}^n \frac{|x_1^* H'(0) x_k|^2}{\lambda_1 - \lambda_k}.$$

Since  $s(t)$  is the largest singular value of  $G(\lambda(t))$ , it is also the largest eigenvalue of the matrix:

$$H(t) = \begin{bmatrix} 0 & G(\lambda(t)) \\ G(\lambda(t))^* & 0 \end{bmatrix}, \quad (15)$$

which has first and second derivatives

$$H'(t) = \begin{bmatrix} 0 & (G \circ \lambda)'(t) \\ ((G \circ \lambda)'(t))^* & 0 \end{bmatrix} \quad \text{and} \quad H''(t) = \begin{bmatrix} 0 & (G \circ \lambda)''(t) \\ ((G \circ \lambda)''(t))^* & 0 \end{bmatrix}, \quad (16)$$

and where the nonzero blocks are given by (13). Thus, by constructing matrix (15) and its first and second derivatives given in (16),  $s''(0)$  can be computed by a straightforward application of Theorem 2.10, noting that eigenvector  $x_k = [u_k; v_k]$ , where  $u_k$  and  $v_k$  are the left and right singular vectors for the  $k$ th singular value of  $G(\lambda(0))$ , which is of course equal to the eigenvalue  $\lambda_k$  of  $H(0)$ .

### 3 The extended and improved criss-cross algorithm for computing the $\varepsilon$ -spectral value set abscissa

Our criss-cross algorithm is composed of two main subcomponents, vertical and horizontal searches, which we now describe.

### 3.1 Vertical cross sections of spectral value sets

We begin by presenting a fundamental theorem that relates singular values of the transfer function, evaluated at some point in the complex plane, to purely imaginary eigenvalues of an associated matrix pencil. As alluded to in Section 1, this correspondence, in various more specific forms, has been a key tool in the aforementioned complex stability radius class of algorithms [Bye88, BBK89, BB90, BS90, GVDV98, BSV12, BM17] and the  $\varepsilon$ -pseudospectral abscissa algorithm of [BLO03]. We defer the proof to Appendix A, as it is a straightforward extension of these earlier results.

**Theorem 3.1.** *Let  $x \in \mathbb{R}$ ,  $y \in \mathbb{R}$ ,  $\gamma > 0$  not a singular value of  $D$ , and  $\lambda E - A$  be regular. Consider the matrix pencil  $(\mathcal{M}_{\gamma x}, \mathcal{N})$ , where*

$$\mathcal{M}_{\gamma x} := \begin{bmatrix} A - xE - BR^{-1}D^*C & -\gamma BR^{-1}B^* \\ \gamma C^*S^{-1}C & -(A - xE - BR^{-1}D^*C)^* \end{bmatrix}, \quad \mathcal{N} := \begin{bmatrix} E & 0 \\ 0 & E^* \end{bmatrix}, \quad (17)$$

*$R = D^*D - \gamma^2 I$ , and  $S = DD^* - \gamma^2 I$ . Then  $\mathbf{i}y$  is a finite eigenvalue of matrix pencil  $(\mathcal{M}_{\gamma x}, \mathcal{N})$  if and only if  $\gamma$  is a singular value of  $G(x + \mathbf{i}y)$  and  $x + \mathbf{i}y$  is not an eigenvalue of  $(A, E)$ .*

By setting  $\gamma = \varepsilon^{-1}$ , Theorem 3.1 immediately leads to the ability to compute all the boundary points, if any, of an  $\varepsilon$ -spectral value set that lie on any desired vertical line specified by the value of  $x$ . Given these boundary points, it is then straightforward to obtain the set of cross sections of the  $\varepsilon$ -spectral value set with this vertical line.

**Remark 3.2.** *Note that the matrix pencil given by (17) cannot be singular. If it were, then  $\gamma$  would be a singular value of  $G(x + \mathbf{i}y)$  for all  $y \in \mathbb{R}$  and thus the entire vertical line specified by value  $x$  would be a part of  $\sigma_\varepsilon(A, B, C, D, E)$ . Since  $(A, E)$  is regular and  $\varepsilon$  is finite, this is not possible.*

It will be convenient to establish the following notation parameterizing the largest singular value of the transfer function by the vertical position  $y$  for a fixed horizontal position given by some  $x$ :

$$\lambda_x(y) := x + \mathbf{i}y \quad (18a)$$

$$Z_x(y) := \lambda_x(y)E - A \quad (18b)$$

$$g_x(y) := \|G(\lambda_x(y))\|_2 = \|CZ_x(y)^{-1}B + D\|_2. \quad (18c)$$

For a chosen vertical line defined by  $x = \eta$ , let  $\{y_1, \dots, y_l\}$  be the set of vertical positions, sorted in increasing order, given by the (we assume nonempty) set of imaginary eigenvalues of (17). Thus, each point  $\eta + \mathbf{i}y_j$  is a boundary point of  $\sigma_\varepsilon(A, B, C, D, E)$ . We first note that it is insufficient to merely assume that interval  $[y_j, y_{j+1}] \subset \sigma_\varepsilon(A, B, C, D, E)$  holds for odd values of  $j$ . This simple construction can fail if one of the eigenvalues  $\mathbf{i}y_j$  is double: if  $\eta + \mathbf{i}y_j$  happens to be a locally rightmost point of the  $\varepsilon$ -spectral value set, then neither  $[y_{j-1}, y_j]$  nor  $[y_j, y_{j+1}]$  will be inside the  $\varepsilon$ -spectral value set. As such, a robust algorithm must assert which intervals are cross sections. One possibility is via the intermediate value theorem and only requires evaluating  $g_{x=\eta}(y)$  at some interior point  $\tilde{y}_j \in (y_j, y_{j+1})$  for each candidate interval;  $[y_j, y_{j+1}]$  is a cross section of  $\sigma_\varepsilon(A, B, C, D, E)$  if and only if  $g_{x=\eta}(\tilde{y}_j) > \varepsilon^{-1}$ . Alternatively, in [SVDT95], it was shown that it is possible to cheaply compute  $g'_{x=0}(y_j)$  by using the eigenvector associated with eigenvalue  $\mathbf{i}y_j$ ; see [BM17] for this result when  $E \neq I$ . Furthermore, it is straightforward to adapt this approach for any vertical line  $x = \eta$ . Recalling that  $y_j < y_{j+1}$  holds, interval  $[y_j, y_{j+1}]$  is not a cross section of  $\sigma_\varepsilon(A, B, C, D, E)$  if either  $g'_{x=\eta}(y_j) < 0$  or  $g'_{x=\eta}(y_{j+1}) > 0$  hold. However, obtaining the values of these derivatives via their associated eigenvectors can sometimes considerably increase memory usage; e.g. if there are  $2n$  imaginary eigenvalues, then one would need an additional  $4n^2$  floating point numbers to store all the eigenvectors. Lastly, it is relatively inexpensive to compute  $g'_{x=\eta}(y_j)$  directly.

Suppose that  $g_x(\hat{y})$  is a simple singular value with associated left and right singular vectors  $\hat{u}$  and  $\hat{v}$ . As  $\lambda'_x(y) = \mathbf{i}$ , by (13a) it follows that

$$(G \circ \lambda_x)'(\hat{y}) = -\mathbf{i}CZ_x(\hat{y})^{-1}EZ_x(\hat{y})^{-1}B \quad (19)$$

and subsequently, by (14), the first derivative of (18c) at  $\hat{y}$  is

$$g'_x(\hat{y}) = -\operatorname{Re}(\hat{\mathbf{i}}\hat{u}^*CZ_x(\hat{y})^{-1}EZ_x(\hat{y})^{-1}B\hat{v}). \quad (20)$$

The main cost in computing  $g'_x(\hat{y})$  is computing the terms involving  $Z_x(\hat{y})^{-1}$ , but an  $LU$  factorization can be formed once when computing  $g_x(\hat{y})$  and then reused to compute its derivative. Furthermore, evaluating (18c), and possibly its derivative (20), at multiple points is an embarrassingly parallel task.

## 3.2 Horizontal cross sections of spectral value sets

For a given vertical line  $x = \eta$ , let  $\Omega_k = [y_k, y_{k+1}]$  denote a vertical cross section segment of  $\sigma_\varepsilon(A, B, C, D, E)$  and  $\Omega = \{\Omega_1, \dots, \Omega_q\}$  denote the set of all these cross sections, computed via the means described in Section 3.1. In the standard criss-cross algorithm for computing  $\varepsilon$ -pseudospectral abscissa [BLO03], horizontal cross sections were defined by the midpoints of each vertical cross section  $\Omega_k \in \Omega$ :

$$\psi_k = 0.5(y_k + y_{k+1}).$$

Assuming that there is at least one cross section of positive length (as otherwise, all the intervals would actually be just singleton points along the vertical line  $x = \eta$ , with  $\eta$  equal to the  $\varepsilon$ -spectral value set abscissa), it is then possible to move rightward in the spectral value set, since at least one of the points  $\eta + \mathbf{i}\psi_k$  must be strictly in the interior of  $\sigma_\varepsilon(A, B, C, D, E)$ . Computing the rightmost point  $\lambda$  in the union of these horizontal lines intersected with  $\sigma_\varepsilon(A, B, C, D, E)$  provides  $\operatorname{Re} \lambda > \eta$ , or equivalently written:

$$\max_{\Omega_k \in \Omega} \max\{\operatorname{Re} \lambda : \lambda \in \sigma_\varepsilon(A, B, C, D, E) \text{ and } \operatorname{Im} \lambda = \psi_k\}. \quad (21)$$

It is actually possible to compute (21) using Theorem 3.1 and [BLO03] proposed doing just that for the case of the  $\varepsilon$ -pseudospectral abscissa, by adapting it to consider horizontal cross sections of pseudospectra instead of vertical ones. However, we will now present a version of Theorem 3.1 that is applicable for finding cross sections of  $\sigma_\varepsilon(A, B, C, D, E)$  along any given line in the complex plane, as this more general form will be needed in Section 4. In order to present the theorem, it will be convenient to first establish the following notation for lines.

**Definition 3.3.** Let  $\theta \in [0, 2\pi)$  denote the angle between the  $x$ -axis and some ray from the origin, with the positive  $x$  and  $y$  directions respectively given by  $\theta = 0$  and  $\theta = \pi/2$ . Given  $s \in \mathbb{R}$ , we define  $L(\theta, s)$  as the parallel line to the left of the ray given by  $\theta$ , separated by distance  $s$ , with left defined with respect to the direction  $\theta$ .

**Theorem 3.4.** Given the line  $L(\theta, s)$ , let  $\{\mathbf{i}\omega_1, \dots, \mathbf{i}\omega_l\}$  be the set of purely imaginary eigenvalues of (17), where  $\gamma = \varepsilon^{-1}$ ,  $x = -s$ , and matrices  $A$  and  $B$  have been respectively replaced by  $e^{\mathbf{i}\theta_r}A$  and  $e^{\mathbf{i}\theta_r}B$ , with  $\theta_r = \pi/2 - \theta$ . Then the points  $\lambda_j = e^{-\mathbf{i}\theta_r}(-s + \mathbf{i}\omega_j)$  define the cross sections of  $\sigma_\varepsilon(A, B, C, D, E)$  along  $L(\theta, s)$ .

*Proof.* By Theorem 3.1 and Corollary 2.3, it follows that  $-s + \mathbf{i}\omega_j$  must be all the boundary points of  $\sigma_\varepsilon(e^{\mathbf{i}\theta_r}A, e^{\mathbf{i}\theta_r}B, C, D, E)$  along the vertical line defined by  $x = -s$ . Since this spectral value set is entirely composed of eigenvalues of  $(e^{\mathbf{i}\theta_r}M(\Delta), E)$ , recalling (3) defining  $M(\Delta)$ , multiplying  $e^{\mathbf{i}\theta_r}M(\Delta)$  by  $e^{-\mathbf{i}\theta_r}$  is equivalent to a rotation about the origin by angle  $-\theta_r$ , which yields  $\sigma_\varepsilon(A, B, C, D, E)$ . Since  $\theta_r = \pi/2 - \theta$ , this specific rotation also moves all points  $-s + \mathbf{i}\omega_j$  precisely onto the line  $L(\theta, s)$  and thus  $\lambda_j = e^{-\mathbf{i}\theta_r}(-s + \mathbf{i}\omega_j)$  are all the boundary points of  $\sigma_\varepsilon(A, B, C, D, E)$  along  $L(\theta, s)$ .  $\square$



Thus, for a midpoint  $\psi_k$  of some cross section on the vertical line  $x = \eta$ , it follows by Theorem 3.4 that the boundary points of  $\sigma_\varepsilon(A, B, C, D, E)$  along the horizontal line  $L(0, \psi_k)$  are given by  $\omega_j + \mathbf{i}\psi_k$ , where  $\{\mathbf{i}\omega_1, \dots, \mathbf{i}\omega_l\}$  are the imaginary eigenvalues of the version of (17) given by Theorem 3.4 and  $\omega_l + \mathbf{i}\psi_k$  is the rightmost boundary point with  $\omega_l > \eta$  holding. This procedure can then be applied independently for each interval  $\Omega_k \in \Omega$  to compute (21). However, this is where we make our main departure in extending the criss-cross algorithm to general spectral value sets, as there are several downsides to this approach.

The dominant cost of the criss-cross algorithm is computing the imaginary eigenvalues of (17), which requires  $\mathcal{O}(n^3)$  work, with a notably large constant since it is a  $2n \times 2n$  matrix pencil. Furthermore, in each horizontal search phase, computing (21) may require computing the imaginary eigenvalues of several versions of (17), one for each interval  $\Omega_k \in \Omega$ . Although parallelization could be used to solve these multiple eigenvalue problems, each requires at least  $8n^2$  floating point numbers (possibly complex valued) of memory just to store the corresponding matrix pencil, which may limit the achievable parallelization speedup, due to cache and memory contention issues.

We instead consider using a more direct root-finding approach. Analogously to (18), it will now be helpful to parameterize the largest singular value of the transfer function by the horizontal position  $x$  for a *fixed vertical position* given by some  $y$ :

$$\lambda_y(x) := x + \mathbf{i}y \quad (22a)$$

$$Z_y(x) := \lambda_y(x)E - A \quad (22b)$$

$$g_y(x) := \|G(\lambda_y(x))\|_2 = \|CZ_y(x)^{-1}B + D\|_2. \quad (22c)$$

Let  $\eta$  specify the current (nonempty) vertical cross section of  $\sigma_\varepsilon(A, B, C, D, E)$  and  $y \in (y_k, y_{k+1})$  define the position of an associated horizontal line for some interval of positive length  $\Omega_k \in \Omega$ . We aim to find a boundary point  $x + \mathbf{i}y$  of  $\sigma_\varepsilon(A, B, C, D, E)$  such that  $x > \eta$  by solving

$$g_y(x) - \varepsilon^{-1} = 0. \quad (23)$$

If  $x$  solves (23), then clearly  $x + \mathbf{i}y$  is a boundary point of  $\sigma_\varepsilon(A, B, C, D, E)$ . Since  $\eta + \mathbf{i}y$  is an interior point of  $\sigma_\varepsilon(A, B, C, D, E)$ , it follows that  $g_y(\eta) - \varepsilon^{-1} > 0$  and thus  $\eta$  provides an a priori lower bound on a permissible root. As  $\lim_{x \rightarrow \infty} g_y(x) = \|D\|_2$  and  $\varepsilon \|D\|_2 < 1$ , (23) must converge to some negative value as  $x \rightarrow \infty$  and thus it is straightforward to find some upper bound on a permissible root as well. Hence a simple bisection method, for example, would suffice to solve (23) such that  $x > \eta$  (albeit with only a linear rate of convergence). However, by exploiting first and possibly second derivatives of singular values, (23) can also be solved with faster first- or second-order root-finding methods, respectively. Nevertheless, it will still be necessary to enforce monotonically updating brackets, to both ensure convergence and that the solution is greater than the given value  $\eta$ . For example, a hybrid Newton-bisection (or Halley-bisection) root-finding method would guarantee convergence and near a root, we could still reasonably expect to observe quadratic (or cubic) convergence.

If the computed value  $x$  is the largest such permissible value, then  $x$  is equal to the globally optimal value  $\omega_l$  obtained by invoking Theorem 3.4. However, solving (23) via an iterative root-finding method may not always return the globally rightmost boundary point along the given horizontal line. Nonetheless, in our improved criss-cross algorithm, any boundary point on the horizontal line and to the right of  $\eta + \mathbf{i}y$  will suffice. This small concession has little to no negative impact in practice (as we will see in Section 6) and eventually, the root finding procedure is guaranteed to always return the globally rightmost boundary point along a given horizontal line, since as  $\eta$  becomes sufficiently close to the value of the  $\varepsilon$ -spectral value set abscissa, there will only be exactly one root of (23) such that  $x > \eta$ .

We now derive the first and second derivatives of (22c). Suppose that  $g_y(\hat{x})$  is a simple singular value with associated left and right singular vectors  $\hat{u}$  and  $\hat{v}$ . As  $\lambda'_y(x) = 1$  and  $\lambda''_y(x) = 0$ , by

(13), it follows that

$$(G \circ \lambda_y)'(x) = -CZ_y(x)^{-1}EZ_y(x)^{-1}B \quad (24a)$$

$$(G \circ \lambda_y)''(x) = 2CZ_y(x)^{-1}EZ_y(x)^{-1}EZ_y(x)^{-1}B. \quad (24b)$$

Again by (14), the first derivative of (22c) at  $\hat{x}$  is

$$g_y'(\hat{x}) = -\operatorname{Re}(\hat{u}^*CZ_y(\hat{x})^{-1}EZ_y(\hat{x})^{-1}B\hat{v}), \quad (25)$$

while its second derivative at  $\hat{x}$  can be computed via applying Theorem 2.10 to matrix (15) with first and second derivatives (16), respectively defined by  $G(\lambda_y(\hat{x}))$  and the matrix derivatives given in (24) evaluated at  $\hat{x}$ . Note that both  $g_y'(\hat{x})$  and  $g_y''(\hat{x})$  are relatively inexpensive to obtain once either (a)  $g_y(\hat{x})$ ,  $\hat{u}$ , and  $\hat{v}$  or (b) a full SVD of  $G(x+\mathbf{i}y)$  have been computed, respectively. Although computing the latter may not be reasonable if  $\min(m, p)$  is relatively large, the former can generally be computed efficiently and reliably by sparse methods.

Our proposed root-finding approach for the horizontal search phase has two key benefits over computing imaginary eigenvalues of pencils given by (17).

First, as recently benchmarked in [BM17, Table 2], it is typically much cheaper to compute the norm of the transfer function: for randomly generated dense systems with  $n = m = p = \{20, 100, 400\}$ , computing the norm of the transfer function was up to 2.47, 10.2, and 36.8 times faster, respectively, than computing the eigenvalues of the associated matrix pencils given by (17). Furthermore, if  $m, p \ll n$ , which is typical for input-output systems, then this performance gap can widen dramatically further: for a randomly generated dense system with  $n = 400$  and  $m = p = 10$  (again [BM17, Table 2]), computing the norm of the transfer function was up to 119 times faster than computing the eigenvalues of matrix pencil (17), whose cost to compute is generally unaffected by the number of inputs and outputs. Since the first and second derivatives can generally be computed with little additional cost over obtaining (22c), solving (23) for the horizontal searches via a first- or second-order root-finding method has the potential to be much faster than a direct extension of the original criss-cross algorithm would be. Such a direct extension would involve computing the imaginary eigenvalues of several large matrix pencils specified by Theorem 3.4 and (17), one for each of the cross sections computed in the preceding vertical search phase.

Second, like the eigenvalue-decomposition-based horizontal search phase, (23) can be solved in an embarrassingly parallel fashion for all the midpoints  $\psi_k$  of vertical cross sections  $\Omega_k$ , but our root-finding approach permits additional efficiencies and optimizations. Since there is no need to form the two large  $2n \times 2n$  matrices defining the matrix pencil given by Theorem 3.4, cache and memory contention issues should be much less severe, as only the  $n \times n$   $LU$  factorization of  $\lambda E - A$  is needed. Perhaps more importantly, and in contrast to the eigenvalue-based approach, is that in practice, we can actually often avoid needing to solve (23) for many of the midpoints  $\psi_k$ , thus perhaps negating a need to parallelize at all. Observe that the left side of (23) provides a measure of how far a given point  $x + \mathbf{i}y$  is inside the interior of  $\sigma_\varepsilon(A, B, C, D, E)$ . It thus stands to reason that a global optimizer of (21) might most likely lie on the horizontal line  $\mathbf{i}\psi_k$  that provides the maximal value of  $g_x(y)$  at  $y = \psi_k$ . Alternatively, a more promising strategy might be to take the version which has the largest initial Newton step to the right, that is, for the value of  $y \in \{\psi_1, \dots, \psi_q\}$  which maximizes:

$$-\frac{g_y(\eta) - \varepsilon}{g_y'(\eta)}. \quad (26)$$

Thus, for the horizontal search phase, we propose a priority of solving each instance of (23), for each  $\psi_k$ , where (26) is sorted in decreasing order for  $y \in \{\psi_1, \dots, \psi_q\}$ . For convenience, assume that the  $\psi_k$  midpoints are already sorted such that this ordering holds. Let  $\hat{x}$  be the solution of (23) for  $y = \psi_1$ , which had nothing but  $\eta$  to use as a starting point for finding an

---

**Subroutine 1**  $[x, y] = \text{horizontalSearch}(\eta, \{y_1, \dots, y_q\})$

---

**Constants:**

$\varepsilon > 0$  with  $\varepsilon \|D\|_2 < 1$  and matrices  $A, B, C, D, E$  defining  $\sigma_\varepsilon(A, B, C, D, E)$

**Input:**

$\eta \in \mathbb{R}$  defining a vertical line  $x = \eta$

$\{y_1, \dots, y_q\}$  with  $y_k \in \mathbb{R}$  and  $\eta + \mathbf{i}y_k$  in the interior of  $\sigma_\varepsilon(A, B, C, D, E)$

**Output:**

$x + \mathbf{i}y$  is a boundary point of  $\sigma_\varepsilon(A, B, C, D, E)$  with  $x > \eta$ , and  $y \in \{y_1, \dots, y_q\}$ .

```

1: reorder  $\{y_1, \dots, y_q\}$  s.t. (26) with  $y = y_k$  decreases with respect to all  $k$ 
2:  $x := \eta; y := y_1$ 
3: for  $k = 1, \dots, q$  do
4:   set function handle  $f(\cdot) := g_{y=y_k}(\cdot) - \varepsilon^{-1}$  // (23) defined for  $y = y_k$ 
5:   if  $f(x) > 0$  then
6:      $x := \text{findARootToTheRight}(f(\cdot), x)$ 
7:      $y := y_k$ 
8:   end if
9: end for

```

---

upper bound. Now, for the second solve, (23) for  $y = \psi_2$ , we can use  $\hat{x}$  as a starting point. If the left side of (23) is negative at  $\hat{x}$ , then we immediately have an upper bound on a root for  $\psi_2$  that is worse (to the left) than root  $\hat{x}$  for  $\psi_1$  and we have no evidence that there are any roots to the right; hence, we can completely skip solving (23) for  $\psi_2$  and proceed to  $\psi_3$ . Similarly, if the left side of (23) is exactly zero at  $\hat{x}$ , then (23) is already solved but does not yield a better root. On the other hand, if the left side of (23) for  $\psi_2$  is positive, then solving (23) for  $\psi_2$  initialized at  $\hat{x}$  must yield a better root  $\tilde{x} > \hat{x}$ , and so the solve should proceed. Furthermore, by initializing at  $\hat{x}$  instead of  $\eta$ , we have the additional benefit of warm starting the root-finding computation. We continue evaluating whether to solve or skip (23) for each of the  $\psi_k$  using the best (rightmost) root computed so far. Finally, note that our horizontal search procedure, as presented here, does not explicitly use parallelism but certainly parallelism could be added to it, potentially reducing its runtime further.

Pseudocode for our new horizontal search method is given in Subroutine 1, which makes use of the following root-finding method that we only define as function specification. The specification ensures that the root-finding routine will return a value corresponding to a point on an outer-facing spectral value set boundary as opposed to on an inner-facing boundary.

**Definition 3.5.** Let  $\text{findARootToTheRight}(f(\cdot), x_0)$  define some implementation of a bracketing and root-finding routine that given a function  $f(\cdot)$  and an initial guess  $x_0$  with  $f(x_0) > 0$ , returns a value  $r$  such that  $r > x_0$ ,  $f(r) = 0$ , and  $f(r + \delta) < 0$  for all  $\delta \in (0, \tau)$  for some fixed value  $\tau > 0$ .

**Remark 3.6.** It is interesting to note that while [BLO03, Theorem 4.1] also considered the pseudospectral analogs of the first derivatives given in (25) and (20), they actually only used them for analysis and did not make use of them computationally to improve efficiency and accuracy, as we do here.

### 3.3 The new criss-cross algorithm for spectral value sets

In order to compute the  $\varepsilon$ -spectral value set abscissa, we must first compute the spectrum of matrix pencil  $(A, E)$  and discard any eigenvalues the user does not wish to consider (e.g. uncon-

---

**Algorithm 1**  $[\eta] = \text{svsAbscissa}(\varepsilon, A, B, C, D, E)$

---

**Input:**

$\varepsilon > 0$  with  $\varepsilon \|D\|_2 < 1$  and matrices  $A, B, C, D, E$  defining  $\sigma_\varepsilon(A, B, C, D, E)$

**Output:**

$\eta$  equal to the value of  $\alpha_\varepsilon(A, B, C, D, E)$

```

1:  $\Lambda := \text{eig}(A, E)$ 
2: // As desired by user, discard uncontrollable/unobservable/infinite eigenvalues
3:  $\Lambda := \{\lambda \in \Lambda : \lambda \text{ meets user's inclusion criteria}\}$ 
4: if  $\infty \in \Lambda$  then
5:   return  $\eta = \infty$ 
6: end if
7:  $\lambda_0 := \arg \max\{\text{Re } \lambda : \lambda \in \Lambda\}$ 
8: // To reduce number of expensive vertical searches, first find a boundary point:
9:  $[\eta, y] := \text{horizontalSearch}(\text{Re } \lambda_0, \{\text{Im } \lambda_0\})$ 
10: while  $\eta < \alpha_\varepsilon(A, B, C, D, E)$  do
11:   compute imaginary eigenvalues  $\{\mathbf{i}y_1, \dots, \mathbf{i}y_l\}$  of (17) for  $x = \eta$  and  $\gamma = \varepsilon^{-1}$ 
12:   form all intervals  $\Omega_k = [y_k, y_{k+1}]$  s.t.  $\eta + \mathbf{i}y \in \sigma_\varepsilon(A, B, C, D, E) \forall y \in \Omega_k$ 
13:    $\Psi := \{\psi_1, \dots, \psi_q\}$  such that  $\psi_k$  is a midpoint of interval  $\Omega_k$ 
14:    $[\eta, y] := \text{horizontalSearch}(\eta, \Psi)$ 
15: end while

```

---

trollable/unobservable/infinite, as discussed in Section 2.1). This is an important preprocessing step as the algorithm will be initialized at a rightmost eigenvalue  $\lambda_0$  of those that remain. The original criss-cross algorithm for pseudospectra then proposes that a vertical search should be done slightly to the right of  $\lambda_0$  (it cannot be done exactly at  $\text{Re } \lambda_0$  because  $(\lambda_0 E - A)$  would be singular). Instead, we propose to first use Subroutine 1 to find a (locally) rightmost boundary point  $\lambda_{\text{bd}}$  of  $\sigma_\varepsilon(A, B, C, D, E)$  along the horizontal line passing through  $\lambda_0$ , with  $\text{Re } \lambda_{\text{bd}} > \text{Re } \lambda_0$ , and then commence the first vertical search at  $\text{Re } \lambda_{\text{bd}}$ . Besides the numerical benefits of avoiding the initial vertical search close to an eigenvalue of  $(A, E)$ , our modified scheme has the perhaps more important effect of potentially reducing both the number of vertical searches and the number of root finding problems (23) provided to each horizontal search phase (since the number of vertical cross sections is likely to decrease as the algorithm converges). Once  $\lambda_{\text{bd}}$  has been computed, the convergent part of the algorithm begins, first computing vertical cross sections of  $\sigma_\varepsilon(A, B, C, D, E)$  at the vertical line  $x = \text{Re } \lambda_{\text{bd}}$  and then performing our modified horizontal search given by Subroutine 1. The algorithm alternates between these updating vertical and horizontal searches until it has converged to a globally rightmost of  $\sigma_\varepsilon(A, B, C, D, E)$ , excluding the user-discarded eigenvalues of  $(A, E)$ , if any.

See Algorithm 1 for pseudocode of our new  $\varepsilon$ -spectral value set abscissa method.

## 4 Computing the $\varepsilon$ -spectral value set radius

Our method is also readily adapted to computing the  $\varepsilon$ -spectral value set radius.

### 4.1 Arc cross sections of spectral value sets.

First, instead of finding vertical cross sections of  $\sigma_\varepsilon(A, B, C, D, E)$ , as described in Section 3.1, we will find the set of arcs given by  $\sigma_\varepsilon(A, B, C, D, E)$  intersected with a chosen circle of some radius  $r > 0$  centered at the origin. In [MO05], this phase is referred to as a circular search.

Less general versions of the following result, the analog of Theorem 3.1, go back as far as [HS91, Section 3], where they considered a fixed radius  $r = 1$ ,  $D = 0$ , and  $E = I$ . We defer the proof to Appendix B.

**Theorem 4.1.** *Let  $r > 0$  be the radius of a circle centered at the origin, angle  $\theta \in [0, 2\pi)$ ,  $\gamma > 0$  not a singular value of  $D$ , and  $\lambda E - A$  be regular. Consider the matrix pencil  $(\mathcal{M}_{\gamma r}, \mathcal{N}_{\gamma r})$ , where*

$$\begin{aligned}\mathcal{M}_{\gamma r} &:= \begin{bmatrix} A - BR^{-1}D^*C & -\gamma BR^{-1}B^* \\ 0 & rE^* \end{bmatrix}, \\ \mathcal{N}_{\gamma r} &:= \begin{bmatrix} rE & 0 \\ -\gamma C^*S^{-1}C & A^* - C^*DR^{-1}B^* \end{bmatrix},\end{aligned}\tag{27}$$

$R = D^*D - \gamma^2 I$  and  $S = DD^* - \gamma^2 I$ . Then  $e^{i\theta}$  is an eigenvalue of matrix pencil  $(\mathcal{M}_{\gamma r}, \mathcal{N}_{\gamma r})$  if and only if  $\gamma$  is a singular value of  $G(re^{i\theta})$  and  $re^{i\theta}$  is not an eigenvalue of  $(A, E)$ .

Setting  $\gamma = \varepsilon^{-1}$ , Theorem 4.1 provides a means to compute all the boundary points, if any, of an  $\varepsilon$ -spectral value set that lie on any desired circle of radius  $r$  centered at the origin. More specifically, let  $\{\theta_1, \dots, \theta_l\}$  be the set of angles, all in  $[0, 2\pi)$  and sorted in increasing order, given by the (we assume nonempty) set of unit-modulus eigenvalues of (27). Thus, each point  $re^{i\theta_j}$  is a boundary point of  $\sigma_\varepsilon(A, B, C, D, E)$ . To determine the set of arcs on the circle of radius  $r$  that pass through the spectral value set, the additional “wrap-around” interval  $[\theta_l, \theta_1 + 2\pi]$  must be also be considered in addition to intervals  $\Omega_k = [\theta_k, \theta_{k+1}]$  for  $k = 1, \dots, l - 1$ . Determining whether an interval passes through the spectral value set can be done via any one of the three means described in Section 3.1.

**Remark 4.2.** *In contrast to the matrix pencils used in the vertical search phases for computing the spectral value set abscissa, the pencil defined by (27) can sometimes be singular, which presents an additional challenge. For now, we simply assume it is nonsingular and address this issue in Section 4.3.*

To provide the polar-coordinate analog of derivative (20), we again parameterize the largest singular value of the transfer function, similarly to (18) and (22), but now with respect to the angle  $\theta$  for a fixed radius  $r$  as follows:

$$\lambda_r(\theta) := re^{i\theta}\tag{28a}$$

$$Z_r(\theta) := \lambda_r(\theta)E - A\tag{28b}$$

$$g_r(\theta) := \|G(re^{i\theta})\|_2 = \|G_r(\theta)\|_2.\tag{28c}$$

Suppose that  $g_r(\hat{\theta})$  is a simple singular value with associated left and right singular vectors  $\hat{u}$  and  $\hat{v}$ . As  $\lambda_r'(\theta) = ire^{i\theta}$ , by (13a) it follows that

$$(G \circ \lambda_r)'(\theta) = -ire^{i\theta}CZ_r(\hat{\theta})^{-1}EZ_r(\hat{\theta})^{-1}B\tag{29}$$

and subsequently, by (14), the first derivative of (28c) at  $\hat{\theta}$  is

$$g_r'(\hat{\theta}) = -\operatorname{Re}\left(ire^{i\theta}\hat{u}^*CZ_r(\hat{\theta})^{-1}EZ_r(\hat{\theta})^{-1}B\hat{v}\right).\tag{30}$$

## 4.2 Radial cross sections of spectral value sets

Given a circle of radius  $r = \eta$ , let  $\Omega_k = [\theta_k, \theta_{k+1}]$  denote a non-zero length arc of this circle which also lies in  $\sigma_\varepsilon(A, B, C, D, E)$  and  $\Omega = \{\Omega_1, \dots, \Omega_q\}$  denote the set of all such arcs computed via the means described in Section 4.1, including the additional “wrap-around” arc. Similar to the

so-called radial searches employed in [MO05] for pseudospectra, we will consider the set of lines through the origin defined by angles

$$\psi_k = 0.5(\theta_k + \theta_{k+1}),$$

the midpoints of all the arcs in  $\Omega$ . Computing the outermost boundary point  $\lambda$  of the intersection of  $\sigma_\varepsilon(A, B, C, D, E)$  and these lines through the origin provides  $|\lambda| > \eta$ , or equivalently written:

$$\max_{\Omega_k \in \Omega} \max\{|\lambda| : \lambda \in \sigma_\varepsilon(A, B, C, D, E) \text{ and } \angle \lambda = \psi_k\}. \quad (31)$$

While Theorem 3.4 could be applied for lines  $L(\psi_k, 0)$  in order to solve (31), we will instead adapt our new root-finding-based horizontal search to the radial case.

Analogously to (23) for  $\varepsilon$ -spectral value set abscissa case, the corresponding root-finding subproblem of (31) is:

$$g_\theta(r) - \varepsilon^{-1} = 0, \quad (32)$$

where

$$\lambda_\theta(r) := r e^{i\theta} \quad (33a)$$

$$Z_\theta(r) := \lambda_\theta(r)E - A \quad (33b)$$

$$g_\theta(r) := \|G(\lambda_\theta(r))\|_2 = \|CZ_\theta(r)^{-1}B + D\|_2 \quad (33c)$$

parameterizes the largest singular value of the transfer function by the radius  $r$  for a *fixed angle*  $\theta$ . As  $\lambda'_\theta(r) = e^{i\theta}$  and  $\lambda''_\theta(r) = 0$ , by (13), it follows that

$$(G \circ \lambda_\theta)'(r) = -e^{i\theta} CZ_\theta(\hat{r})^{-1} E Z_\theta(\hat{r})^{-1} B \quad (34a)$$

$$(G \circ \lambda_\theta)''(r) = 2e^{2i\theta} CZ_\theta(r)^{-1} E Z_\theta(r)^{-1} E Z_\theta(r)^{-1} B. \quad (34b)$$

Assuming that  $g_\theta(\hat{r})$  is a simple singular value, with left and right singular vectors  $\hat{u}$  and  $\hat{v}$ , by (14), the first derivative of (33c) at  $\hat{r}$  is

$$g'_\theta(\hat{r}) = -\operatorname{Re}(e^{i\theta} \hat{u}^* CZ_\theta(\hat{r})^{-1} E Z_\theta(\hat{r})^{-1} B \hat{v}). \quad (35)$$

As before, the second derivative of (33c) at  $\hat{r}$  can be computed via Theorem 2.10. Finally, we similarly prioritize solving the root problems given by (32) such that

$$-\frac{g_\theta(\eta) - \varepsilon}{g'_\theta(\eta)}. \quad (36)$$

is sorted in decreasing order, for the angles  $\theta$  given by each midpoint  $\psi_k$  of interval  $\Omega_k \in \Omega$ .

**Remark 4.3.** *Similar to [BLO03], the pseudospectral analogs of the first derivatives given in (35) and (30) were considered in [MO05, Theorem 2.3], but they were only used for analysis and not computationally as we do here.*

### 4.3 The full algorithm and singular pencils

Before we can present a complete pseudocode for computing the  $\varepsilon$ -spectral value set radius, we must finally address that the circular searches may be interior and/or have associated pencils that are singular, either of which might cause the algorithm to break down. We will need the following theorem and an associated corollary, both generalizations of [MO05, Theorem 2.11 and Corollary 2.12], respectively; the argument given for [MO05, Theorem 2.11] extends directly, with only simple substitutions, so we omit a proof here.

---

**Subroutine 2**  $[r, \theta] = \text{radialSearch}(\eta, \{\theta_1, \dots, \theta_q\})$

---

**Constants:**

$\varepsilon > 0$  with  $\varepsilon \|D\|_2 < 1$  and matrices  $A, B, C, D, E$  defining  $\sigma_\varepsilon(A, B, C, D, E)$

**Input:**

$\eta > 0$  defining a circle of radius  $\eta$  centered at the origin

$\{\theta_1, \dots, \theta_q\}$  with  $\theta_k \in [0, 2\pi)$  and  $\eta e^{i\theta_k}$  in the interior of  $\sigma_\varepsilon(A, B, C, D, E)$

**Output:**

$re^{i\theta}$  is a boundary point of  $\sigma_\varepsilon(A, B, C, D, E)$  with  $r > \eta$  and  $\theta \in \{\theta_1, \dots, \theta_q\}$ .

```

1: reorder  $\{\theta_1, \dots, \theta_q\}$  s.t. (36) with  $\theta = \theta_k$  decreases with respect to all  $k$ 
2:  $r := \eta; \theta := \theta_1$ 
3: for  $k = 1, \dots, q$  do
4:   set function handle  $f(\cdot) := g_{\theta=\theta_k}(\cdot) - \varepsilon^{-1}$  // (32) defined for  $\theta = \theta_k$ 
5:   if  $f(r) > 0$  then
6:      $r := \text{findARootToTheRight}(f(\cdot), r)$ 
7:      $\theta := \theta_k$ 
8:   end if
9: end for

```

---

**Theorem 4.4.** *Suppose that the matrix pencil defined by (27) is singular for some value  $r > 0$  and that the largest singular value of  $G(re^{i\theta})$  is simple for all  $\theta \in [0, 2\pi)$ . Then either*

1. *the boundary of  $\sigma_\varepsilon(A, B, C, D, E)$  contains the circle of radius  $r$  or*
2. *the circle of radius  $r$  is strictly inside  $\sigma_\varepsilon(A, B, C, D, E)$ .*

**Corollary 4.5.** *Suppose that for some fixed  $r > 0$ ,  $\|G(re^{i\theta})\|_2 - \varepsilon^{-1} < 0$  holds for at least one angle  $\theta \in [0, 2\pi)$ . Then the matrix pencil defined by (27) is regular.*

In light of Corollary 4.5, the following strategy to avoid singular pencils was used in [MO05]. First, on initialization, they proposed computing the maximal value  $\eta$  of (31), for angle  $\angle\lambda$ , where  $\lambda$  is an outermost eigenvalue of  $A$ . This can be done by applying Theorem 3.4 for  $L(\angle\lambda, 0)$ , in other words, by initializing the algorithm with a radial search instead of a circular one.<sup>2</sup> By Corollary 4.5, it then follows that the matrix pencils defined by (27) will be regular for all  $r > \eta$ . However, in floating-point arithmetic, the computed value of  $\eta$  could be lower than its true value and so the matrix pencil for the initial circular search could still be a singular one, or even if not, could still correspond to a circular search done entirely in the interior of the pseudospectrum, which would not return any boundary points. Thus, [MO05] proposed increasing  $\eta$  by a small tolerance as necessary. If  $g_{\theta=\angle\lambda}(\eta) \geq 0$ , then they advocated increasing  $\eta$  by  $k\delta_0 > 0$ , where  $\delta_0$  is the Newton step for  $g_{\theta=\angle\lambda}(t)$  computed at  $t = \eta$ , such that  $g_{\theta=\angle\lambda}(\eta + k\delta_0) < 0$  for the smallest positive integer  $k$ . They noted that typically  $k = 1$  or  $2$  sufficed in practice. In exact arithmetic, this strategy ensures that all circular searches performed by the algorithm will be nonsingular.

Of course, the fact that this procedure cannot be done in exact arithmetic may lead to some issues. Since the algorithm must make a decision based on the sign of  $g_{\theta=\angle\lambda}(t) \approx 0$ , computed in floating-point arithmetic, the algorithm may increase  $\eta$  unnecessarily or not increase it when it should. Even if  $g_{\theta=\angle\lambda}(\eta + k\delta_0) < 0$  holds *numerically*, its sign could still be wrong (since it should be close to zero) and thus this is not a foolproof guarantee that the matrix pencil is

---

<sup>2</sup> This choice in [MO05] seems to only be a consequence for avoiding singular pencils. They make no mention that this choice also has the additional benefit of likely making the method more efficient overall, by reducing the total number of searches, as we motivate here in the beginning of Section 3.3, when proposing that our new criss-cross method should start with a horizontal search.

regular or that the circle of radius  $\eta + k\delta_0$  actually crosses the boundary of the pseudospectrum. Furthermore, increasing  $\eta$  by  $k\delta_0$  may make it greater than  $\rho_\varepsilon(A, B, C, D, E)$ , in which case, significant accuracy may sometimes be lost.

Motivated by such thorny issues, we propose a new approach for dealing with singular pencils or interior circular searches that may arise due to numerical inaccuracy. Furthermore, we will still be able to use our new `radialSearch` subroutine for all radial searches. If we were to extend the above procedure of [MO05], the very first radial search would have to be done via Theorem 3.4, as the root-finding method used by `radialSearch` is only guaranteed to find a root of (32) to the right of some initial guess, not necessarily the farthest one; as such, `radialSearch` is not always guaranteed to return a value  $\eta$  such that the assumptions of Corollary 4.5 would hold for all  $r > \eta$ . We first consider the following new result.

**Theorem 4.6.** *Given  $\varepsilon > 0$  with  $\varepsilon\|D\|_2 < 1$ , set  $\gamma = \varepsilon^{-1}$  and let  $\lambda$  be an outermost controllable and observable eigenvalue of  $(A, E)$ . For some  $\eta \geq |\lambda|$ , such that the circle of radius  $\eta$  centered at the origin is strictly in the interior of  $\sigma_\varepsilon(A, B, C, D, E)$ , let  $\delta > 0$  be the largest value such that, for all  $t \in [0, 1]$ , circles of radius  $\eta + t\delta$  are still subsets of  $\sigma_\varepsilon(A, B, C, D, E)$ . Finally, let  $R = \{r_1, \dots, r_l\}$  denote the set of radii corresponding to the boundary points of  $\sigma_\varepsilon(A, B, C, D, E)$  that lie on  $L(\theta, 0)$  but are outside the circle of radius  $\eta$ , where  $\theta \in [0, 2\pi)$  has been chosen randomly. Then for  $\hat{r} = \min\{r_1, \dots, r_l\} > \eta$ , either of the two following scenarios may hold:*

1.  $(\mathcal{M}_{\gamma r}, \mathcal{N}_{\gamma r})$ , defined by the matrices (27), is singular for  $r = \eta + \delta$  but  $\hat{r} = \eta + \delta = \rho_\varepsilon(A, B, C, D, E)$  or
2.  $(\mathcal{M}_{\gamma r}, \mathcal{N}_{\gamma r})$  is regular for  $r = \eta + \delta$  and, with probability one,  $\hat{r} > \eta + \delta$ .

*Proof.* We first consider the case where  $(\mathcal{M}_{\gamma r}, \mathcal{N}_{\gamma r})$  is singular at  $r = \eta + \delta$ . Since  $\eta + \delta \in R$ , it must be that  $re^{i\theta}$  is a boundary point of  $\sigma_\varepsilon(A, B, C, D, E)$ , and by Theorem 4.4, the circle of radius  $r$  centered at the origin must be a subset of the boundary of  $\sigma_\varepsilon(A, B, C, D, E)$ . Furthermore,  $\rho_\varepsilon(A, B, C, D, E) \geq r$ . If strict inequality holds, then there must exist some boundary point  $\hat{\lambda} \in \sigma_\varepsilon(A, B, C, D, E)$  with  $|\hat{\lambda}| > r$ . But this contradicts the conclusion of Lemma 2.4, that there exists a path taking some controllable and observable eigenvalue of  $(A, E)$  to  $\hat{\lambda}$  such that only  $\lambda(1) = \hat{\lambda}$  is a boundary point, since any such  $\lambda(t)$  must also cross the circle of radius  $r$  at some  $t < 1$ . Hence,  $r = \hat{r}$  as  $R$  only contains a single unique value, namely  $\eta + \delta$ .

Now suppose  $(\mathcal{M}_{\gamma r}, \mathcal{N}_{\gamma r})$  is regular at  $r = \eta + \delta$ . By assumption, the circle of radius  $\eta + \delta$  only touches the spectral value boundary but does not cross it. Furthermore, since the pencil is regular, by Theorem 4.1, there can only be a finite number (at most  $n$ ) of contact points between this circle and the spectral value set boundary. Suppose that  $\hat{r} = \eta + \delta$ , noting that by assumption,  $\hat{r}$  cannot be any smaller. Then, for boundary point  $\hat{r}e^{i\theta}$ , its angle  $\theta$  must be equal to one of the angles corresponding to the finite set of contact points. As  $\theta \in [0, 2\pi)$  was chosen randomly, the probability of this event occurring is zero. Therefore, with probability one,  $\theta$  will not correspond to any of the contact points on the circle of radius  $\eta + \delta$  and thus,  $\hat{r} > \eta + \delta$ .  $\square$

Theorem 4.6 says that if the current radius  $\eta$  corresponds to a problematic circular search (one within the interior of the spectral value set), then `radialSearch`( $\eta, \{\theta\}$ ), when  $\theta$  is chosen randomly from  $[0, 2\pi)$ , must return a value  $r > \eta$  such that either  $r$  is equal to  $\rho_\varepsilon(A, B, C, D, E)$  (and thus the computation is done) or  $r$  has been increased at least beyond the continuous problematic annular region of width  $\delta > 0$  where circular searches cannot be done. On the other hand, if the circular search for radius  $\eta$  is problematic because this circle is a subset of the spectral value set boundary, then by Theorem 4.4, it is clear that  $\eta = \rho_\varepsilon(A, B, C, D, E)$ . More precisely, it is not the case that singular pencils are so problematic for the algorithm, but circular searches done in the interior of  $\sigma_\varepsilon(A, B, C, D, E)$  (singular or not) are.

Unfortunately, detecting when a circular search is a problematic interior one is difficult in inexact arithmetic. Instead of trying to reliably avoid such scenarios, we simply accept that circular searches may sometimes fail to produce any arc segments due to numerical issues, and



that in this case, we must then distinguish between whether the computation is actually finished or not. If the computation is done, then  $\eta$  already equals  $\rho_\varepsilon(A, B, C, D, E)$  and calling `radialSearch`( $\eta, \{\theta\}$ ) for any angle cannot increase  $\eta$  any further. If the circular search was a failed interior one, then with probability one, `radialSearch` must increase  $\eta$  by more than  $\delta$ . The only remaining scenario is if the eigensolver simply failed to compute the correct boundary points but this is a problem to be addressed in the eigensolver, not our algorithm; even so, `radialSearch`( $\eta, \{\theta\}$ ) still has a nonzero (but not one either) chance of possibly increasing  $\eta$  to recover from this difficult situation.

We thus propose that when any circular search fails to return any arc segments, that the `radialSearch` subroutine should be called anyway, but for one or more randomly chosen angles. If `radialSearch` cannot increase  $\eta$ , we simply conclude that  $\eta = \rho_\varepsilon(A, B, C, D, E)$ . Otherwise,  $\eta$  has been increased and we proceed to the next (and larger) circular search. Even if the next circular search also fails, `radialSearch` will again be called with a new randomly chosen angle (or several), thus providing another opportunity for the algorithm to recover from numerical issues. Of course, Theorem 4.6 motivating our new strategy also assumes exact arithmetic, which is not possible in practice, and if it were, then the only time a problematic matrix pencil would ever be encountered is when the spectral value set boundary is a circle attaining the value of  $\rho_\varepsilon(A, B, C, D, E)$ , an easy case since  $\rho_\varepsilon(A, B, C, D, E)$  would have already been computed exactly anyway. Nevertheless, compared to the perturbation technique of [MO05], we feel that our new strategy is much better motivated and more robust in practice. Furthermore, unlike the previous method of [MO05], our new procedure has no danger of overshooting the true value of  $\rho_\varepsilon(A, B, C, D, E)$ .

Pseudocode for our new  $\varepsilon$ -spectral value set radius method is given in Algorithm 2.

## 5 Convergence of the algorithms

Although the proof of convergence arguments given in [BLO03] and [MO05] could be directly extended, the following argument we give here we feel is less technical and more intuitive.

**Theorem 5.1.** *Algorithms 1 and 2 converge to the  $\varepsilon$ -spectral value set abscissa and radius, respectively.*

*Proof.* Let  $\eta_\star$  be the value of the  $\varepsilon$ -spectral value set abscissa/radius and  $\{\eta_k\}$  be a sequence our methods generate, which by construction must be monotonically increasing and  $\eta_k \leq \eta_\star$  must hold. So suppose that  $\eta_k \rightarrow \hat{\eta} < \eta_\star$ . Let  $\Omega(\eta)$  denote the set of intervals corresponding to vertical (circular) cross sections varying by  $x = \eta$  ( $r = \eta$ ) and consider:

$$l(\eta) := \max_{\Omega_k \in \Omega(\eta)} \{|\omega_{k+1} - \omega_k| : \Omega_k = [\omega_k, \omega_{k+1}]\}.$$

The only way  $l(\eta)$  can be discontinuous at some value is if the corresponding matrix pencil is singular. As this can only happen for circular searches and when it does, it means the algorithm has converged to  $\eta_\star$ ,  $l(\eta)$  must be continuous for all  $\eta \in [\eta_0, \hat{\eta}]$ . Furthermore, as  $\eta \rightarrow \hat{\eta}$ ,  $l(\eta)$  must converge to zero; otherwise the corresponding matrix pencils would converge to a singular one, which we have ruled out cannot happen for  $\eta \in [\eta_0, \hat{\eta}]$ . Let  $\lambda_0$  be the starting point and  $\lambda_\star$  be a boundary point attaining the  $\varepsilon$ -spectral value set abscissa/radius, with associated perturbation matrix  $\Delta_\star$ , that is,  $\lambda_\star \in \sigma(M(\Delta_\star), E)$ . Furthermore, let  $\lambda(t)$  for  $t \in [0, 1]$  be a path described by Lemma 2.4 taking  $\lambda(0) = \lambda_0$  to  $\lambda(1) = \lambda_\star$ . Since by the lemma,  $\lambda(t)$  must be strictly in the interior of the  $\varepsilon$ -spectral value set for all  $t \in [0, 1)$ , and  $\hat{\eta}/\eta_\star < 1$ , there exists a fixed  $\delta > 0$  such that the neighborhood of radius  $\delta$  about  $\lambda(\hat{\eta}/\eta_\star)$  lays entirely in interior of the spectral value set. For any point  $\lambda(\hat{t})$  with  $\hat{t} \in (\hat{\eta}/\eta_\star - 0.5\delta, \hat{\eta}/\eta_\star + 0.5\delta)$ , the neighborhood of radius  $0.25\delta$  about  $\lambda(\hat{t})$  must also lie entirely in the interior spectral value set. Therefore, the length of any vertical/circular cross section passing through any such  $\lambda(\hat{t})$  must be at least  $0.5\delta$ . As  $\eta_k \rightarrow \hat{\eta}$ ,

---

**Algorithm 2**  $[\eta] = \text{svsRadius}(\varepsilon, A, B, C, D, E)$ 

---

**Input:**

$\varepsilon > 0$  with  $\varepsilon \|D\|_2 < 1$  and matrices  $A, B, C, D, E$  defining  $\sigma_\varepsilon(A, B, C, D, E)$   
 $r$  a positive integer, specifying how many random angles to try

**Output:**

$\eta$  equal to the value of  $\rho_\varepsilon(A, B, C, D, E)$

```
1:  $\Lambda := \text{eig}(A, E)$ 
2: // As desired by user, discard uncontrollable/unobservable/infinite eigenvalues
3:  $\Lambda := \{\lambda \in \Lambda : \lambda \text{ meets user's inclusion criteria}\}$ 
4: if  $\infty \in \Lambda$  then
5:   return  $\eta = \infty$ 
6: end if
7:  $\lambda_0 := \arg \max\{|\lambda| : \lambda \in \Lambda\}$ 
8:  $\Psi := \{\angle \lambda_0, \psi_1, \dots, \psi_r\}$  such that  $\psi_k$  is chosen randomly from  $[0, 2\pi)$ 
9:  $[\eta, \theta] := \text{radialSearch}(|\lambda_0|, \Psi)$ 
10: while  $\eta < \alpha_\varepsilon(A, B, C, D, E)$  do
11:   compute unimodular eigenvalues  $\{e^{i\theta_1}, \dots, e^{i\theta_l}\}$  of (27) for  $r = \eta$  and  $\gamma = \varepsilon^{-1}$ 
12:   form all intervals  $\Omega_k = [\theta_k, \theta_{k+1}]$  s.t.  $\eta e^{i\theta} \in \sigma_\varepsilon(A, B, C, D, E) \forall \theta \in \Omega_k$ 
13:   if no such intervals then
14:      $\Psi := \{\psi_1, \dots, \psi_r\}$  such that  $\psi_k$  is chosen randomly from  $[0, 2\pi)$ 
15:   else
16:      $\Psi := \{\psi_1, \dots, \psi_q\}$  such that  $\psi_k$  is a midpoint of interval  $\Omega_k$ 
17:   end if
18:    $[\eta, \theta] := \text{radialSearch}(\eta, \Psi)$ 
19: end while
```

---

$t \rightarrow \hat{\eta}/\eta_*$ , and so  $l(\eta)$  must converge to at least  $0.5\delta$ , a contradiction. Hence, the methods converge to  $\eta_*$ .  $\square$

## 6 Numerical experiments

We implemented algorithms `svsAbscissa` and `svsRadius` in a single completely new MATLAB code. When computing the  $\varepsilon$ -spectral value set radius, whenever no intervals are obtained, we configured our code to generate three random angles for `radialSearch`, for invoking Theorem 4.6 to avoid singular pencils and/or interior circular searches.

An evaluation of which bracketing and root-finding method would be most efficient to use for implementing the prerequisite subroutine `findARootToTheRight` (specified in Definition 3.5) is beyond the scope of this article. We implemented `findARootToTheRight` to first bracket a root by iteratively increasing the current guess by adding the larger of either two times the absolute value of the Newton step or the distance from the current guess and the initial guess  $x_0$  until an upper bound has been found (and also increasing the lower bound along the way) and then to subsequently compute a root using the Hermite-interpolation-based method of [CGL01], which uses first-order information but not second. The very first step of the upper bound search was set to increase the initial guess by at least  $\max\{10^{-6}, 0.01|x_0|\}$ . If the function given `findARootToTheRight` fails to return a finite value, our code simply updates the lower bound and increases the current guess.

As a practical optimization, for when all the matrices are real valued but  $\lambda_0$  is not, our

code always attempts to first find a root along the  $x$ -axis, either to the right of  $\lambda_0$  (or outward in either direction for the  $\varepsilon$ -spectral value set radius) before computing a solution to the root problem for  $\lambda_0$ . Assuming such a root exists along the  $x$ -axis, the initial  $\varepsilon$ -spectral value set abscissa (or radius) estimate  $\eta$  will be increased, from  $\eta = \alpha(A, E)$  (or  $\eta = \rho(A, E)$ ) to some larger value corresponding to a boundary point  $\sigma_\varepsilon(A, B, C, D, E)$  on the  $x$ -axis. Even though this strategy potentially introduces an additional horizontal search (or two radial searches), it often substantially reduces the overall number of *complex-valued* SVDs incurred, replacing them with *much cheaper real-valued ones*. This optimization can have a significant net benefit in terms of running time because it can sometimes require many iterations to find an upper bound for the root-finding problem for  $\lambda_0$ , since without this optimization, it would be initialized at  $\lambda_0$ , a pole of the transfer function.

Our new code has the following similarities to the `pspa` and `pspr` routines of [MO], the respective implementations of the original criss-cross type methods for computing the pseudospectral abscissa [BLO03] and the pseudospectral radius [MO05]. First, if the problem is real valued, the spectral value sets are symmetric with respect to the  $x$ -axis; in this case, any interval  $\Omega_k \in \Omega$  that corresponds to a section in the open lower half-plane is discarded (since it is “duplicated” by its positive conjugate). Second, as `pspr` does not use a structure-preserving eigensolver, we used `eig` from MATLAB for all codes in the benchmarks done here; note that any robust implementation should use structure-preserving eigensolvers, such as those available in SLICOT [BMS<sup>+</sup>99]. Third, our method simply terminates when the  $\varepsilon$ -spectral value set abscissa/radius estimate  $\eta$  can no longer be increased, by any amount; no tolerance is needed. Finally, if the current value of  $y$  (or  $\theta$  for the  $\varepsilon$ -spectral value set radius case) is in the interior of some interval  $\Omega_k \in \Omega$ , then  $\Omega_k$  is split into two separate intervals at this value (which is why our `horizontalSearch` and `radialSearch` pseudocodes presented here return these values). For explanations of these last two numeric-related choices, the latter of which helps ensure that the algorithms do not stagnate at non-globally optimal stationary points, see [BLO03, Section 6].

All experiments were conducted using MATLAB R2017a running on a Macbook Pro with an Intel i7-6567U dual-core CPU, 16GB of RAM, and Mac OS X v10.12. Running times were measured using `tic` and `toc` and, to account for variability, the time reported for each method-problem pair is the average of five trials. For consistency, `rng(100)` was called before each trial, since some test problems are randomly generated and `svsRadius` makes use of random numbers.

## 6.1 Pseudospectral examples

To compare to the existing `pspa` and `pspr` routines for pseudospectra, we used 21 demo problems from EigTool [Wri02], all with  $n = 100$ . For each example, we tested computing both the  $\varepsilon$ -pseudospectral abscissa (Table 1) and  $\varepsilon$ -pseudospectral radius (Table 2), using  $\varepsilon = 0.01$ . The values computed by our new methods typically agreed to those of `pspa` and `pspr` to machine precision (the median relative error was  $-7.26 \times 10^{-16}$ ). The magnitude of the worst relative error was  $1.99 \times 10^{-10}$  but only 6 of the 42 relative differences exceeded  $10^{-12}$  in magnitude. However, these six larger discrepancies were because `pspa` and `pspr` were less accurate: they returned points beyond the pseudospectral boundary (due to inaccuracy in the imaginary parts of the computed imaginary eigenvalues) while our root-finding approach found points on the boundary much more accurately (verified by plotting the boundaries near these points using `contour` on a sufficiently fine grid).

In Table 1, we see that our new method is faster than `pspa` on every single test problem, with the average speedup being 2.23. The maximum speedup of 6.24 faster was obtained on `companion` and we note that half of that was due to our optimization for avoiding complex-valued SVDs (since many iterations were necessary to find the upper bound for the first root-finding problem). Although part of the motivation for our root-finding was that it could be particularly efficient for general spectral value sets, (where  $m, p \ll n$  typically holds), it is now clear that our new approach can offer large performance increases for pseudospectra as well (where  $n = m = p$ ).

Pseudospectral Abscissa ( $\varepsilon = 0.01$ ): <b>pspa</b> versus new method											
Problem	# solves				# searches			time (sec.)	speedup		
	Eig		SVD		vert.	horz.					
airy(101)	13	4	10	46	4	4	9	4(6)	0.414	<b>0.306</b>	1.35
basor(100)	12	2	0	19	5	2	7	2	0.438	<b>0.148</b>	2.95
chebspec(101)	5	1	4	23	2	1	3	1(2)	0.141	<b>0.088</b>	1.61
companion(100)	167	2	396	291	2	2	165	3(11)	3.632	<b>0.582</b>	6.24
convdiff(101)	4	1	0	21	2	1	2	1	0.083	<b>0.072</b>	1.15
davies(101)	10	2	0	41	4	2	6	2(3)	0.264	<b>0.222</b>	1.19
demmel(100)	13	6	10	62	6	6	7	6	0.301	<b>0.270</b>	1.12
frank(100)	4	1	0	16	2	1	2	1	0.113	<b>0.064</b>	1.78
gaussseidel(100,'C')	4	4	4	21	2	4	2	4(6)	0.169	<b>0.166</b>	1.02
gaussseidel(100,'D')	6	1	0	14	3	1	3	1	0.182	<b>0.058</b>	3.16
gaussseidel(100,'U')	4	1	0	21	2	1	2	1	0.178	<b>0.074</b>	2.41
grcar(100)	4	2	4	23	2	2	2	2(3)	0.123	<b>0.115</b>	1.07
hatano(100)	8	1	0	10	4	1	4	1	0.258	<b>0.048</b>	5.43
kahan(100)	4	1	0	12	2	1	2	1	0.096	<b>0.051</b>	1.88
landau(100)	14	2	2	11	5	2	9	2	0.274	<b>0.086</b>	3.19
orrsommerfeld(101)	21	4	2	41	9	4	12	4	0.534	<b>0.260</b>	2.06
random(100)	4	1	4	8	2	1	2	1	0.121	<b>0.047</b>	2.57
randomtri(100)	3	2	14	18	2	2	1	2	0.104	<b>0.091</b>	1.14
riffle(100)	6	2	2	17	3	2	3	2	0.100	<b>0.082</b>	1.22
transient(100)	3	1	0	18	2	1	1	1	0.129	<b>0.122</b>	1.05
twisted(100)	12	2	10	18	5	2	7	2	0.367	<b>0.115</b>	3.18
Average Speedup:										2.23	

Table 1: For each labeled pair of columns, performance data is given on the left for **pspa** and on the right for the abscissa variant of our new method. The “Eig” column gives the total number of  $2n \times 2n$  eigensolves computed while the “SVD” column gives the total number of evaluations of the norm of the transfer function. The number of vertical and horizontal searches are given under the “vert.” and “horz.” headers, respectively; for our new method, if the total number of “horz.” searches is greater than the number that *actually needed to be solved*, the latter is given first, with the former given in parenthesis. The time for the faster of the two methods is in bold while the “speedup” column is simply **pspa**’s time divided by the time for our new method. The last row gives the arithmetic mean of all the speedup values. All pseudospectral examples have  $n = m = p = 100$ .

For the pseudospectral radius comparison, shown in Table 2, our new method was about 11% slower than **pspr** on average, and sometimes even up to three times slower (**companion** and **convdiff**). However, for the vast majority of problems (13 of 21), **pspr** only required two  $2n \times 2n$  eigenvalue decompositions before convergence, leaving little to no room for further optimization; as such, our new method was only faster on one of these problems (**hatano**, 1.24 speedup) and had a near tie on another (**frank**). Of the eight examples where **pspr** required at least three eigenvalue decompositions, our new method was generally faster, with speedups ranging from 1.28 to 1.91. Even though the average running time is slightly in favor of **pspr**, the accuracy and reliability improvements of our new method offer tangible benefits; indeed, **pspr** was responsible for four of the six aforementioned problems that had higher numerical error. Furthermore, we feel this small performance gap can likely be closed by further code optimizations.

Pseudospectral Radius ( $\varepsilon = 0.01$ ): <b>pspr</b> versus new method											
Problem	# solves				# searches			time (sec.)	speedup		
	Eig	SVD			circ.	rad.					
airy(101)	5	2	19	40	2	2	3	2(3)	<b>0.199</b>	0.232	0.86
basor(100)	4	3	11	35	2	3	2	4	<b>0.177</b>	0.268	0.66
chebspec(101)	2	1	10	29	1	1	1	1	<b>0.070</b>	0.137	0.51
companion(100)	3	1	402	269	1	1	2	2(4)	<b>0.491</b>	1.383	0.36
convdif(101)	2	1	6	40	1	1	1	1	<b>0.056</b>	0.174	0.32
davies(101)	2	1	10	40	1	1	1	1	<b>0.080</b>	0.193	0.41
demmel(100)	2	1	6	24	1	1	1	1	<b>0.056</b>	0.096	0.58
frank(100)	2	1	6	18	1	1	1	1	<b>0.072</b>	0.072	1.00
gaussseidel(100,'C')	2	3	2	20	1	3	1	3	<b>0.086</b>	0.142	0.61
gaussseidel(100,'D')	2	1	2	17	1	1	1	1	<b>0.067</b>	0.073	0.93
gaussseidel(100,'U')	2	1	2	30	1	1	1	2	<b>0.092</b>	0.112	0.83
grcar(100)	14	6	52	60	6	6	8	8	0.520	<b>0.357</b>	1.46
hatano(100)	2	1	2	13	1	1	1	1	0.081	<b>0.066</b>	1.24
kahan(100)	4	1	9	10	2	1	2	1(2)	0.117	<b>0.061</b>	1.91
landau(100)	8	3	15	21	3	3	5	3(4)	0.209	<b>0.163</b>	1.28
orrsommerfeld(101)	2	1	6	25	1	1	1	1	<b>0.080</b>	0.135	0.59
random(100)	9	4	34	36	4	4	5	5(6)	0.320	<b>0.236</b>	1.36
randomtri(100)	2	1	1	19	1	1	1	2	<b>0.066</b>	0.080	0.83
rifle(100)	2	1	2	14	1	1	1	1	<b>0.046</b>	0.063	0.74
transient(100)	7	2	10	21	3	2	4	2	0.301	<b>0.184</b>	1.64
twisted(100)	2	2	18	29	1	2	1	2(4)	<b>0.091</b>	0.164	0.56
Average Speedup:										0.89	

Table 2: The headers remain mostly as described in Table 1, except that now **pspr** is compared to our new method when computing the  $\varepsilon$ -pseudospectral radius; correspondingly, the number of circular and radial searches are respectively given under the “circ.” and “rad.” headers. All pseudospectral examples have  $n = m = p = 100$ .

## 6.2 General spectral value set examples

For our evaluation using spectral value sets, we used 16 test examples of varying dimensions: four problems (CBM, CM3, CM4, CSE2) from [GGO13] and another 12 from the SLICOT benchmark examples.<sup>3</sup> Since some of the examples have nonzero  $D$  matrices, setting  $\varepsilon := 0.01$  was not possible for all examples. Instead, we calculated both the continuous- and discrete-time  $\mathcal{L}_\infty$  norms for each example, via `getPeakGain` in MATLAB with a tolerance of  $10^{-14}$ , to be respectively used for the  $\varepsilon$ -spectral value set abscissa and radius evaluations. Let  $\gamma_*$  be the corresponding computed  $\mathcal{L}_\infty$ -norm value. For each problem, we set  $\varepsilon := 2\gamma_*$ , provided that  $\gamma_*$  was a finite positive value and  $\varepsilon\|D\|_2 < 0.5$  held. Otherwise, for problems with nonzero  $D$  matrices, we used  $\varepsilon := 0.5\|D\|_2^{-1}$  and  $\varepsilon := 0.01$  for the rest. Finally, each problem was initialized at a rightmost/outermost finite eigenvalue of  $(A, E)$  for computing the  $\varepsilon$ -spectral value set abscissa/radius, regardless of whether or not it was an uncontrollable or unobservable eigenvalue.

As our methods are the first to be able to compute the  $\varepsilon$ -spectral value set abscissa and radius, there are no directly competing codes available for comparison. Instead, we consider a second version of our own code, that mimics how “directly-extended” (DE) versions of **pspa** and **pspr** would perform, where the horizontal and radial searches are done by solving the corresponding eigenvalue problems instead of by our proposed root-finding alternatives. The values computed by both variants generally agreed with each other (the median relative error was precisely 0).

<sup>3</sup> Available at <http://slicot.org/20-site/126-benchmark-examples-for-model-reduction>

Spectral Value Set Abscissa: directly extended versus new method														
Problem	Dimensions			# solves			# searches			time (sec.)	speedup			
	$n$	$m$	$p$	Eig	SVD	vert.	horz.							
CBM	351	1	2	8	3	23	68	3	3	5	4(5)	3.159	<b>1.362</b>	2.32
CM3	123	1	3	6	2	14	19	3	2	3	2	0.242	<b>0.093</b>	2.60
CM4	243	1	3	8	2	26	20	3	2	5	2	1.453	<b>0.342</b>	4.25
CSE2	63	1	32	6	1	8	9	3	1	3	1(2)	0.064	<b>0.021</b>	3.07
beam	348	1	1	9	3	23	50	4	3	5	3	3.263	<b>1.039</b>	3.14
build	48	1	1	19	3	51	29	6	3	13	3	0.128	<b>0.031</b>	4.14
CDplayer	120	2	2	8	3	20	72	3	3	5	3(4)	0.230	<b>0.167</b>	1.38
eady	598	1	1	7	1	12	7	3	1	4	1(2)	8.149	<b>0.946</b>	8.62
fom	1006	1	1	11	3	25	43	3	3	8	4(6)	57.151	<b>9.758</b>	5.86
heat-cont	200	1	1	5	2	6	44	3	2	2	2	0.571	<b>0.187</b>	3.05
heat-disc	200	1	1	5	3	6	37	3	3	2	3	0.921	<b>0.501</b>	1.84
iss	270	3	3	8	3	19	32	3	3	5	3(4)	2.043	<b>0.652</b>	3.13
pde	84	1	1	6	1	10	18	3	1	3	1(2)	0.140	<b>0.035</b>	3.99
peec	480	1	1	1	1	0	1	1	1	0	0(1)	<b>2.266</b>	2.284	0.99
random	200	1	1	9	3	19	96	4	3	5	3	1.048	<b>0.584</b>	1.80
tline	256	2	2	1	1	0	39	1	1	0	1	<b>0.659</b>	0.913	0.72
Average speedup:													3.18	
Average speedup (directly extended with horz. search first):													2.62	

Table 3: The headers remain as described in Table 1, except that now a directly-extended variant of `pspa` is compared to our new method for computing the  $\varepsilon$ -spectral value set abscissa. The additional row gives the average speedup if the directly-extended variant were to begin with a horizontal search instead of a vertical one.

However, of the 8 of 32 relative differences that exceeded  $10^{-12}$  in magnitude, four of them even exceeded  $10^{-10}$  and sometimes excessively, with max magnitude error being  $2.00 \times 10^0$ . Like the earlier pseudospectral numerical discrepancies, these four specific larger observed numerical differences were again due to the DE approach being less accurate. We will discuss these examples in more detail after reporting the performance results.

For the  $\varepsilon$ -spectral value set abscissa tests, shown in Table 3, we compared against two versions of the DE approach: one using a vertical search first and an alternative using an initial horizontal search, though we only provide detailed per-problem performance statistics for the former. Overall, our method was much faster than the DE variant using a vertical search first: our method had an average speedup of 3.18, with the highest speedup being 8.62 (`eady`), and it was the faster of the two approaches on 14 out of 16 of the test examples. Even on the two problems where our method was slower (`peec` and `tline`), it was still fairly close. Compared to the DE variant using an initial horizontal search, our new approach was still 2.62 times faster on average, underscoring that the majority of acceleration achieved is due to our new root-finding-based method and not just the simple (though beneficial) swapping of the order of vertical and horizontal searches.

The  $\varepsilon$ -spectral value set radius tests, shown in Table 4, also validate our root-finding-based approach. Our method was again fastest on 14 of the 16 test problems, having an average speedup of 1.82 compared to the DE variant, with the fastest observed speedup being 4.59 (`heat-disc`). On the two problems where our new method was slower (`CM3` and `CDplayer`), it was only by very negligible amounts.

Returning the four examples where the DE variants had the highest errors, three came from the  $\varepsilon$ -spectral value abscissa tests. On `beam` ( $9.18 \times 10^{-10}$  relative error), the final vertical search produced a cross section that, due to inaccuracy in the imaginary parts of the computed

Spectral Value Set Radius: directly extended versus new method														
Problem	Dimensions			# solves		# searches			time (sec.)		speedup			
	$n$	$m$	$p$	Eig	SVD	circ.	rad.							
CBM	351	1	2	2	1	4	23	1	1	1	1	1.431	<b>0.954</b>	1.50
CM3	123	1	3	4	2	10	69	2	2	2	2	<b>0.198</b>	0.216	0.92
CM4	243	1	3	4	2	9	68	2	2	2	2	1.057	<b>0.983</b>	1.08
CSE2	63	1	32	4	2	9	21	2	2	2	2	0.052	<b>0.036</b>	1.46
beam	348	1	1	2	1	4	26	1	1	1	1	1.198	<b>0.891</b>	1.34
build	48	1	1	6	3	17	33	3	3	3	4(5)	0.046	<b>0.036</b>	1.27
CDplayer	120	2	2	2	1	5	46	1	1	1	1	<b>0.078</b>	0.079	0.99
eady	598	1	1	2	1	5	7	1	1	1	0	7.869	<b>6.416</b>	1.23
fom	1006	1	1	2	1	4	5	1	1	1	0	53.626	<b>47.797</b>	1.12
heat-cont	200	1	1	2	1	4	16	1	1	1	1	0.375	<b>0.205</b>	1.83
heat-disc	200	1	1	4	1	7	11	2	1	2	1(2)	0.772	<b>0.168</b>	4.59
iss	270	3	3	8	2	20	37	4	2	4	2	2.627	<b>0.940</b>	2.79
pde	84	1	1	7	1	14	18	3	1	4	1(2)	0.161	<b>0.039</b>	4.09
peec	480	1	1	2	1	3	5	1	1	1	0	5.181	<b>2.193</b>	2.36
random	200	1	1	2	1	4	13	1	1	1	1	0.359	<b>0.246</b>	1.46
tline	256	2	2	4	3	30	109	2	3	3	4(6)	2.523	<b>2.240</b>	1.13
Average speedup:											1.82			

Table 4: The headers remain mostly as described in Table 1, except that now a directly-extended variant of `pspr` is compared to our new method for computing the  $\varepsilon$ -spectral value set radius; correspondingly, the number of circular and radial searches are respectively given under the “circ.” and “rad.” headers.

eigenvalues, was much wider than it was in reality; as the interval was then split into two pieces, the midpoints of both segments actually corresponded to points outside of the  $\varepsilon$ -spectral value set and thus the method failed to make further rightward progress, terminating the code at a value below the actual  $\varepsilon$ -spectral value set abscissa. On `random` ( $-1.32 \times 10^{-9}$ ), inaccuracy in the imaginary parts of the computed imaginary eigenvalues resulted in the final horizontal search returning a point slightly outside of the  $\varepsilon$ -spectral value set, thus slightly overshooting the  $\varepsilon$ -spectral value set abscissa. In contrast, the very large error of  $2.00 \times 10^0$  observed on `tline` occurred because none of the computed eigenvalues for the horizontal search were sufficiently close to the imaginary axis and thus the first horizontal search failed to increase  $\eta$  at all, terminating the code with  $\eta = \alpha(A, E)$ . The remaining fourth high relative error was on `tline` ( $1.49 \times 10^{-7}$ ) in the  $\varepsilon$ -spectral value set radius evaluation. Here, the last radial searches failed to make outward progress because of numerical inaccuracy in the computed imaginary eigenvalues.

Using structure-preserving eigensolvers would have likely avoided at least some (perhaps even all) of the above failures but our root-finding-based approach is more numerically reliable even without them. However, had these numerical difficulties not occurred, the DE variants would have incurred more eigenvalue decompositions before converging, and thus the reported speedups would be even more in favor of our new approach. In particular, computing the  $\varepsilon$ -spectral value set abscissa for `tline` was likely only so much in favor of the DE variant because it stagnated so early; had it not, it is likely that our new approach would have been fastest on this problem.

## 7 Conclusion

We have proposed the first algorithms to compute, not just approximate, the general  $\varepsilon$ -spectral value set abscissa and radius to high accuracy, by both extending and improving upon the earlier  $\varepsilon$ -pseudospectral abscissa and radius algorithms of [BLO03] and [MO05]. Our experiments

validate that our new root-finding-based methodology is both much faster and more reliable than directly-extend approaches. In fact, our modified approach is typically over twice as fast as the original criss-cross algorithm for computing the  $\varepsilon$ -pseudospectral abscissa. Although the  $\varepsilon$ -pseudospectral radius evaluation was more or less a tie in terms of average running time, the increased numerical robustness of our new method, also partly due to our new procedure for handling singular pencils and/or interior circular searches, we feel is likely to outweigh performance concerns in practice.

There are several possible implementation-related modifications that may significantly improve upon our method further, but we have not yet fully explored these avenues. Certainly, there is the question of which root-finding method would be best in practice. Relatedly, one of the most expensive phases of our method can be finding an upper bound for the initial root-finding problem. In preliminary experiments, we found taking two times a Halley step instead of Newton one can dramatically reduce the number of evaluations of the norm of the transfer function needed to locate an upper bound. On a parallel architecture, one could also evaluate multiple trial upper-bound points simultaneously, which could significantly reduce the running time (if not the number of function evaluations). Finally, we also considered replacing the midpoint-of-intervals scheme with points in these intervals obtained from either cubic Hermite interpolants or local maximizers computed by an optimization code, the latter of which was proposed in [BM17] for accelerating the computation of the  $\mathcal{H}_\infty$  norm. While such strategies for this particular application were sometimes faster, sometimes not, optionally applying such an optimization approach may also be motivated by resiliency to numerical failures. If the eigenvalue solver fails to return intervals for the cross sections, or too inaccurate ones, optimizing in vertical/circular directions could allow additional horizontal/radial searches to be performed, thus recovering from the numerical failure.

Finally, for future theoretical work, both the numerical analysis done for the original pseudospectral algorithms as well as their local quadratic rate of convergence results can likely be extended to general spectral value sets, though we have not investigated either yet. Interestingly, the rate of convergence results from [BLO03] and [MO05] require a regularity assumption, which is not needed to prove the quadratic rate of the very related algorithm for computing the  $\mathcal{H}_\infty$  norm by [BB90, BS90]. The question of whether there is an alternative convergence rate proof that does not use such a regularity assumption remains open.

## References

- [BB90] 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. *Syst. Cont. Lett.*, 15:1–7, 1990.
- [BBK89] S. Boyd, V. Balakrishnan, and P. Kabamba. A bisection method for computing the  $\mathcal{H}_\infty$  norm of a transfer matrix and related problems. *Math. Control Signals Systems*, 2:207–219, 1989.
- [BLO03] 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.
- [BM17] P. Benner and T. Mitchell. Faster and more accurate computation of the  $\mathcal{H}_\infty$  norm via optimization. arXiv e-prints arXiv:1707.02497, Cornell University, July 2017. math.OC.
- [BMS<sup>+</sup>99] P. Benner, V. Mehrmann, V. Sima, S. Van Huffel, and A. Varga. SLICOT - a subroutine library in systems and control theory. In B. N. Datta, editor, *Applied and Computational Control, Signals, and Circuits*, volume 1, chapter 10, pages 499–539. Birkhäuser, Boston, MA, 1999.
- [BN93] R. Byers and N. K. Nichols. On the stability radius of a generalized state-space system. *Linear Algebra Appl.*, 188/189:113–134, 1993.
- [BS90] N. A. Bruinsma and M. Steinbuch. A fast algorithm to compute the  $H_\infty$ -norm of a transfer function matrix. *Syst. Cont. Lett.*, 14(4):287–293, 1990.



- [BSV12] P. Benner, V. Sima, and M. Voigt.  $\mathcal{L}_\infty$ -norm computation for continuous-time descriptor systems using structured matrix pencils. *IEEE Trans. Automat. Control*, 57(1):233–238, January 2012.
- [BV14] P. Benner and M. Voigt. A structured pseudospectral method for  $\mathcal{H}_\infty$ -norm computation of large-scale descriptor systems. *Math. Control Signals Systems*, 26(2):303–338, 2014.
- [Bye88] R. Byers. A bisection method for measuring the distance of a stable to unstable matrices. *SIAM J. Sci. Statist. Comput.*, 9:875–881, 1988.
- [CGL01] F. Costabile, M. I. Gualtieri, and R. Luceri. A new iterative method for the computation of the solutions of nonlinear equations. *Numer. Algorithms*, 28(1-4):87–100, 2001.
- [Dai89] L. Dai. *Singular Control Systems*. Number 118 in Lecture Notes in Control and Information Sciences. Springer-Verlag, Berlin, 1989.
- [GGO13] N. Guglielmi, M. Gürbüzbalaban, and M. L. Overton. Fast approximation of the  $H_\infty$  norm via optimization over spectral value sets. *SIAM J. Matrix Anal. Appl.*, 34(2):709–737, 2013.
- [GO11] N. Guglielmi and M. L. Overton. Fast algorithms for the approximation of the pseudospectral abscissa and pseudospectral radius of a matrix. *SIAM J. Matrix Anal. Appl.*, 32(4):1166–1192, 2011.
- [GVDV98] Y. Genin, P. Van Dooren, and V. Vermaut. Convergence of the calculation of  $\mathcal{H}_\infty$ -norms and related questions. In *Proceedings MTNS-98*, pages 429–432, Jul. 1998.
- [HP05] D. Hinrichsen and A. J. Pritchard. *Mathematical Systems Theory I*. Springer-Verlag, Berlin, 2005.
- [HS91] D. Hinrichsen and N. K. Son. Stability radii of linear discrete-time systems and symplectic pencils. *Int. J. Robust Nonlinear Control*, 1:79–97, 1991.
- [Kat82] T. Kato. *A short introduction to perturbation theory for linear operators*. Springer-Verlag, New York-Berlin, 1982.
- [Lan64] P. Lancaster. On eigenvalues of matrices dependent on a parameter. *Numer. Math.*, 6:377–387, 1964.
- [MO] E. Mengi and M. L. Overton. Software for Robust Stability and Controllability. <http://home.ku.edu.tr/~emengi/software/robuststability.html>.
- [MO05] E. Mengi and M. L. Overton. Algorithms for the computation of the pseudospectral radius and the numerical radius of a matrix. *IMA J. Numer. Anal.*, 25(4):648–669, 2005.
- [MO16] T. Mitchell and M. L. Overton. Hybrid expansion-contraction: a robust scaleable method for approximating the  $H_\infty$  norm. *IMA J. Numer. Anal.*, 36(3):985–1014, 2016.
- [OW95] M. L. Overton and R. S. Womersley. Second derivatives for optimizing eigenvalues of symmetric matrices. *SIAM J. Matrix Anal. Appl.*, 16(3):697–718, 1995.
- [SVDT95] J. Sreedhar, P. Van Dooren, and A. Tits. A fast algorithm to compute the real structured stability radius. In *Proc. Conf. Centennial Hurwitz on Stability Theory, Ticino (CH), May 21-26, 1995*.
- [TE05] L. N. Trefethen and M. Embree. *Spectra and pseudospectra: The behavior of nonnormal matrices and operators*. Princeton University Press, Princeton, NJ, 2005.
- [Tre99] L. N. Trefethen. Computation of pseudospectra. *Acta Numer.*, 8:247–295, 1999.
- [TTRD93] L. N. Trefethen, A. E. Trefethen, S. C. Reddy, and T. A. Driscoll. Hydrodynamic stability without eigenvalues. *Science*, 261(5121):578–584, 1993.
- [Van85] C. F. 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.
- [Wri02] T. G. Wright. EigTool. <http://www.comlab.ox.ac.uk/pseudospectra/eigtool/>, 2002.
- [WT01] T. G. Wright and L. N. Trefethen. Large-scale computation of pseudospectra using ARPACK and eigs. *SIAM J. Sci. Comput.*, 23(2):591–605, 2001. Copper Mountain Conference (2000).

## A Proof of Theorem 3.1

*Proof.* Let  $\gamma$  be a singular value of  $G(x+\mathbf{i}y)$  with left and right singular vectors  $u$  and  $v$ , that is, so that  $G(x+\mathbf{i}y)v = \gamma u$  and  $G(x+\mathbf{i}y)^*u = \gamma v$ . Using the expanded versions of these two equivalences

$$\begin{aligned} \left( C((x+\mathbf{i}y)E - A)^{-1}B + D \right) v &= \gamma u \quad \text{and} \\ \left( C((x+\mathbf{i}y)E - A)^{-1}B + D \right)^* u &= \gamma v, \end{aligned} \quad (37)$$

we define

$$q = ((x+\mathbf{i}y)E - A)^{-1}Bv \quad \text{and} \quad s = ((x-\mathbf{i}y)E^* - A^*)^{-1}C^*u. \quad (38)$$

Rewriting (37) using (38) yields the following matrix equation:

$$\begin{bmatrix} C & 0 \\ 0 & B^* \end{bmatrix} \begin{bmatrix} q \\ s \end{bmatrix} = \begin{bmatrix} -D & \gamma I \\ \gamma I & -D^* \end{bmatrix} \begin{bmatrix} v \\ u \end{bmatrix} \implies \begin{bmatrix} v \\ u \end{bmatrix} = \begin{bmatrix} -D & \gamma I \\ \gamma I & -D^* \end{bmatrix}^{-1} \begin{bmatrix} C & 0 \\ 0 & B^* \end{bmatrix} \begin{bmatrix} q \\ s \end{bmatrix}, \quad (39)$$

where

$$\begin{bmatrix} -D & \gamma I \\ \gamma I & -D^* \end{bmatrix}^{-1} = \begin{bmatrix} -R^{-1}D^* & -\gamma R^{-1} \\ -\gamma S^{-1} & -DR^{-1} \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} q \\ s \end{bmatrix} \neq 0. \quad (40)$$

Rewriting (38) as a matrix equation gives:

$$\left( \begin{bmatrix} (x+\mathbf{i}y)E & 0 \\ 0 & (x-\mathbf{i}y)E^* \end{bmatrix} - \begin{bmatrix} A & 0 \\ 0 & A^* \end{bmatrix} \right) \begin{bmatrix} q \\ s \end{bmatrix} = \begin{bmatrix} B & 0 \\ 0 & C^* \end{bmatrix} \begin{bmatrix} v \\ u \end{bmatrix}. \quad (41)$$

Substituting in (39) for the rightmost term of (41) yields

$$\left( \begin{bmatrix} (x+\mathbf{i}y)E & 0 \\ 0 & (x-\mathbf{i}y)E^* \end{bmatrix} - \begin{bmatrix} A & 0 \\ 0 & A^* \end{bmatrix} \right) \begin{bmatrix} q \\ s \end{bmatrix} = \begin{bmatrix} B & 0 \\ 0 & C^* \end{bmatrix} \begin{bmatrix} -D & \gamma I \\ \gamma I & -D^* \end{bmatrix}^{-1} \begin{bmatrix} C & 0 \\ 0 & B^* \end{bmatrix} \begin{bmatrix} q \\ s \end{bmatrix}. \quad (42)$$

Bringing over terms from the left side to separate out  $\mathbf{i}y$  and substituting the inverse on the right using (40) and then multiplying out the matrix terms, we have

$$\mathbf{i}y \begin{bmatrix} E & 0 \\ 0 & -E^* \end{bmatrix} \begin{bmatrix} q \\ s \end{bmatrix} = \begin{bmatrix} A - xE & 0 \\ 0 & A^* - xE^* \end{bmatrix} \begin{bmatrix} q \\ s \end{bmatrix} + \begin{bmatrix} -BR^{-1}D^*C & -\gamma BR^{-1}B^* \\ -\gamma C^*S^{-1}C & -C^*DR^{-1}B^* \end{bmatrix} \begin{bmatrix} q \\ s \end{bmatrix}.$$

Combining the matrices on the right and multiplying by

$$\begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}$$

yields:

$$\mathbf{i}y \begin{bmatrix} E & 0 \\ 0 & E^* \end{bmatrix} \begin{bmatrix} q \\ s \end{bmatrix} = \begin{bmatrix} (A - xE - BR^{-1}D^*C) & -\gamma BR^{-1}B^* \\ \gamma C^*S^{-1}C & -(A - xE - BR^{-1}D^*C)^* \end{bmatrix} \begin{bmatrix} q \\ s \end{bmatrix}.$$

It is now clear that  $\mathbf{i}y$  is an eigenvalue of the matrix pencil  $(\mathcal{M}_{\gamma x}, \mathcal{N})$ .

Now suppose that  $\mathbf{i}y$  is an eigenvalue of pencil  $(\mathcal{M}_{\gamma x}, \mathcal{N})$  with eigenvector given by  $q$  and  $s$  as above. Then it follows that (42) holds, which can be rewritten as (41) by defining  $u$  and  $v$  using the right-hand side equation of (39), noting that neither can be identically zero. It is then clear that the two equivalences in (38) both hold. Finally, substituting (38) into the left-hand side equation of (39), it is clear that  $\gamma$  is a singular value of  $G(x+\mathbf{i}y)$ , with left and right singular vectors  $u$  and  $v$ .  $\square$

## B Proof of Theorem 4.1

*Proof.* Let  $\gamma$  be a singular value of  $G(re^{i\theta})$  with left and right singular vectors  $u$  and  $v$ , that is, so that  $G(re^{i\theta})v = \gamma u$  and  $G(re^{i\theta})^*u = \gamma v$ . Using the expanded versions of these two equivalences

$$\left( C(re^{i\theta}E - A)^{-1}B + D \right)v = \gamma u \quad \text{and} \quad \left( C(re^{i\theta}E - A)^{-1}B + D \right)^*u = \gamma v, \quad (43)$$

we define

$$q = (re^{i\theta}E - A)^{-1}Bv \quad \text{and} \quad s = (re^{-i\theta}E^* - A^*)^{-1}C^*u. \quad (44)$$

Similar to the proof of Theorem 3.1, it follows that

$$\left( \begin{bmatrix} e^{i\theta}rE & 0 \\ 0 & re^{-i\theta}E^* \end{bmatrix} - \begin{bmatrix} A & 0 \\ 0 & A^* \end{bmatrix} \right) \begin{bmatrix} q \\ s \end{bmatrix} = \begin{bmatrix} B & 0 \\ 0 & C^* \end{bmatrix} \begin{bmatrix} -D & \gamma I \\ \gamma I & -D^* \end{bmatrix}^{-1} \begin{bmatrix} C & 0 \\ 0 & B^* \end{bmatrix} \begin{bmatrix} q \\ s \end{bmatrix}. \quad (45)$$

Furthermore, the rightmost three terms of (45) can again be replaced by first substituting the matrix inverse with its explicit form given by (40) and then multiplying these three terms together. Then, multiplying on the left by

$$\begin{bmatrix} I & 0 \\ 0 & -e^{i\theta}I \end{bmatrix}$$

and rearranging terms yields

$$e^{i\theta} \begin{bmatrix} rE & 0 \\ 0 & A^* \end{bmatrix} \begin{bmatrix} q \\ s \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & rE^* \end{bmatrix} \begin{bmatrix} q \\ s \end{bmatrix} + \begin{bmatrix} B & 0 \\ 0 & -e^{i\theta}C^* \end{bmatrix} \begin{bmatrix} -R^{-1}D^*C & -\gamma R^{-1}B^* \\ -\gamma S^{-1}C & -DR^{-1}B^* \end{bmatrix} \begin{bmatrix} q \\ s \end{bmatrix}.$$

Separating and then bringing the  $-e^{i\theta}$  terms over to the left side, we obtain

$$e^{i\theta} \begin{bmatrix} rE & 0 \\ -\gamma C^*S^{-1}C & A^* - C^*DR^{-1}B^* \end{bmatrix} \begin{bmatrix} q \\ s \end{bmatrix} = \begin{bmatrix} A - BDR^{-1}B^* & -\gamma BR^{-1}B^* \\ 0 & rE^* \end{bmatrix} \begin{bmatrix} q \\ s \end{bmatrix},$$

and thus it is clear that  $e^{i\theta}$  is an eigenvalue of the matrix pencil  $(\mathcal{M}_{\gamma r}, \mathcal{N}_{\gamma r})$ .

The reverse implication follows similarly to the reverse argument given in Appendix A for the proof of Theorem 3.1.  $\square$