

## APPROXIMATING THE REAL STRUCTURED STABILITY RADIUS WITH FROBENIUS-NORM BOUNDED PERTURBATIONS\*

N. GUGLIELMI<sup>†</sup>, M. GÜRBÜZBALABAN<sup>‡</sup>, T. MITCHELL<sup>§</sup>, AND M. L. OVERTON<sup>¶</sup>

**Abstract.** We propose a fast method to approximate the real stability radius of a linear dynamical system with output feedback, where the perturbations are restricted to be real valued and bounded with respect to the Frobenius norm. Our work builds on a number of scalable algorithms that have been proposed in recent years, ranging from methods that approximate the complex or real pseudospectral abscissa and radius of large sparse matrices (and generalizations of these methods for pseudospectra to spectral value sets) to algorithms for approximating the complex stability radius (the reciprocal of the  $H_\infty$  norm). Although our algorithm is guaranteed to find only upper bounds to the real stability radius, it seems quite effective in practice. As far as we know, this is the first algorithm that addresses the Frobenius-norm version of this problem. Because the cost is dominated by the computation of the eigenvalue with maximal real part for continuous-time systems (or modulus for discrete-time systems) of a sequence of matrices, our algorithm remains very efficient for large-scale systems provided that the system matrices are sparse.

**Key words.**  $H_\infty$  (H-infinity) norm, linear dynamical systems, spectral value sets, structured pseudospectra, robust stability

**AMS subject classifications.** 65F99, 93D09

**DOI.** 10.1137/16M1110169

**1. Introduction.** Consider a linear time-invariant dynamical system with output feedback defined, for continuous-time systems, by matrices  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times p}$ ,  $C \in \mathbb{R}^{m \times n}$ , and  $D \in \mathbb{R}^{m \times p}$  as

$$(1) \quad \dot{x} = Ax + Bw,$$

$$(2) \quad z = Cx + Dw,$$

where  $w$  is a disturbance feedback depending linearly on the output  $z$  [HP05, p. 538]. For simplicity, we restrict our attention to continuous-time systems for most of the paper, but we briefly explain how to extend our results and methods to discrete-time systems in section 6.

The real stability radius, sometimes called the real structured stability radius, is a well-known quantity for measuring robust stability of linear dynamical systems with output feedback [HP90a, HP90b, HK94, HP05, ZGD95, Kar03]. It measures

---

\*Received by the editors January 3, 2017; accepted for publication (in revised form) July 31, 2017; published electronically November 7, 2017.

<http://www.siam.org/journals/simax/38-4/M111016.html>

**Funding:** The work of the first author was supported in part by the Italian Ministry of Education, Universities and Research (M.I.U.R.) and by Istituto Nazionale di Alta Matematica - Gruppo Nazionale di Calcolo Scientifico (INdAM-G.N.C.S). The work of the third author was supported by National Science Foundation grant DMS-1317205 at New York University. The work of the fourth author was supported in part by National Science Foundation grant DMS-1620083.

<sup>†</sup>Dipartimento di Matematica Pura e Applicata, Università dell'Aquila, I-67010 L'Aquila, Italy (guglielm@univaq.it).

<sup>‡</sup>Department of Management Science and Information Systems, Rutgers University, New Brunswick, NJ 08901 (mgurbuzbalaban@business.rutgers.edu).

<sup>§</sup>Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg 39106, Germany (mitchell@mpi-magdeburg.mpg.de).

<sup>¶</sup>Courant Institute of Mathematical Sciences, New York University, New York, NY 10012 (overton@cims.nyu.edu).

stability under a certain class of real perturbations where the size of the perturbations is measured by a given norm  $\|\cdot\|$ . Most of the literature has focused on spectral norm bounded perturbations in which case there exists a characterization in terms of an explicit formula [QBR<sup>+</sup>95] and a level-set algorithm [SVDT96]. This algorithm has been proven to be convergent; however, it is not practical for systems where large and sparse matrices arise as it requires a sequence of Hamiltonian eigenvalue decompositions, each with a complexity of  $\mathcal{O}(n^3)$ . For larger-scale problems, see [FS14], which treats the special case where  $B = C = I$  and  $D = 0$ .

As an alternative to the spectral norm, Frobenius-norm bounded perturbations have also been of interest to the control community [LKL96, BS99, BS98, BB01, Bob99, BBD01]. It has been argued that the Frobenius norm is easier to compute and is more advantageous to consider in certain types of control systems [Bob99, BBD01], admitting natural extensions to infinite-dimensional systems [BB01]. In the special case  $B = C = I, D = 0$ , there exists an algorithm [Bob99, BBD01] that gives upper and lower bounds for the Frobenius-norm bounded real stability radius; however, there seems to be no algorithm that is applicable in the general case. Indeed, [BV14] describes this as an unsolved research problem. In this paper, we present the first method to the best of our knowledge that provides good approximations to the Frobenius-norm bounded real stability radius.

Our method relies on two foundations. The first is the theory of spectral value sets associated with the dynamical system (1)–(2) as presented in [HP05, Chapter 5]. The second is the appearance of a number of recent iterative algorithms that find rightmost points of spectral value sets of various sorts, beginning with the special case of matrix pseudospectra (the case  $B = C = I, D = 0$ ) [GO11], followed by a related method for pseudospectra [KV14] and extensions to real-structured pseudospectra [GL13, GM15, Ros15, Gug16], and to spectral value sets associated with (1)–(2) [GGO13, MO16] and related descriptor systems [BV14].

The paper is organized as follows. In section 2 we introduce spectral value sets, establishing a fundamental relationship between the spectral value set abscissa and the stability radius which was not previously known for Frobenius-norm bounded real spectral value sets. In section 3 we introduce an ordinary differential equation (ODE) that generates a sequence of points in the spectral value set moving monotonically to the right as far as possible. In section 4 we present a practical iterative method to approximate rightmost points efficiently. This leads to our method for approximating the Frobenius-norm bounded real stability radius, presented in section 5. We outline extensions to discrete-time systems in section 6, present numerical results in section 7, and make concluding remarks in section 8.

**2. Fundamental concepts.** Throughout the paper,  $\|\cdot\|_2$  denotes the matrix 2-norm (maximum singular value), whereas  $\|\cdot\|_F$  denotes the Frobenius norm (associated with the trace inner product). The usage  $\|\cdot\|$  means that the norm may be either  $\|\cdot\|_2$  or  $\|\cdot\|_F$ , or both when they coincide, namely for vectors or rank-one matrices. We use the notation  $\mathbb{C}_-$  to denote the open *left* half-plane  $\{\lambda : \operatorname{Re}(\lambda) < 0\}$  and  $\overline{\mathbb{H}}$  to denote the closed *upper* half-plane  $\{\lambda : \operatorname{Im}(\lambda) \geq 0\}$ .

**2.1. Spectral value sets and  $\mu$ -values.** Given real matrices  $A, B, C$ , and  $D$  defining the linear dynamical system (1)–(2), linear feedback  $w = \Delta z$  leads to a *perturbed system matrix* with the linear fractional form

$$(3) \quad M(\Delta) = A + B\Delta(I - D\Delta)^{-1}C \quad \text{for } \Delta \in \mathbb{K}^{p \times m},$$

where the field  $\mathbb{K}$  is either  $\mathbb{R}$  or  $\mathbb{C}$ . Note that since

$$(4) \quad \|D\Delta\|_2 \leq \|D\|_2 \|\Delta\|_2 \leq \|D\|_2 \|\Delta\|_F,$$

we can ensure that  $M(\Delta)$  is well defined by assuming  $\|\Delta\| \leq \varepsilon$  and  $\varepsilon\|D\|_2 < 1$ , regardless of whether  $\|\Delta\|$  is  $\|\Delta\|_2$  or  $\|\Delta\|_F$ .

DEFINITION 2.1. *Let  $\varepsilon \in \mathbb{R}$ , with  $\varepsilon\|D\|_2 < 1$ . Define the spectral value set with respect to the norm  $\|\cdot\|$  and the field  $\mathbb{K}$  as*

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

Here  $\sigma$  denotes spectrum. Note that

$$\sigma_\varepsilon^{\mathbb{K}, \|\cdot\|_2}(A, B, C, D) \supseteq \sigma_\varepsilon^{\mathbb{K}, \|\cdot\|_F}(A, B, C, D) \supseteq \sigma_0^{\mathbb{K}, \|\cdot\|}(A, B, C, D) = \sigma(A).$$

It is well known that when  $\mathbb{K} = \mathbb{C}$ , the spectral value set can equivalently be defined as the set of points  $s \in \mathbb{C}$  for which the spectral norm, i.e., the largest singular value, of the transfer matrix

$$G(s) = C(sI - A)^{-1}B + D$$

takes values at least  $1/\varepsilon$  [HP05, Chap. 5]. Furthermore, it is well known that

$$(5) \quad \sigma_\varepsilon^{\mathbb{C}, \|\cdot\|}(A, B, C, D) = \bigcup \{ \sigma(M(\Delta)) : \Delta \in \mathbb{C}^{p \times m}, \|\Delta\| \leq \varepsilon \text{ and } \text{rank}(\Delta) \leq 1 \}.$$

As a consequence of this rank-one property, it is clear that

$$\sigma_\varepsilon^{\mathbb{C}, \|\cdot\|_2}(A, B, C, D) = \sigma_\varepsilon^{\mathbb{C}, \|\cdot\|_F}(A, B, C, D).$$

In contrast, when perturbations are restricted to be real, the inclusion

$$\sigma_\varepsilon^{\mathbb{R}, \|\cdot\|_2}(A, B, C, D) \supseteq \sigma_\varepsilon^{\mathbb{R}, \|\cdot\|_F}(A, B, C, D)$$

is generically strict. Instead of ordinary singular values, we must consider *real structured singular values* or *real  $\mu$ -values*, that is, with perturbations restricted to real matrices. This topic is discussed at length in [HP05, sec. 4.4], allowing additional structure to be imposed on  $\Delta$  beyond simply  $\Delta \in \mathbb{K}^{p \times m}$ , and treating a general class of operator norms, including the spectral norm, but not, however, the Frobenius norm. See also [Kar03, Chapter 6].

DEFINITION 2.2. *The  $\mu$ -value of a matrix  $H \in \mathbb{C}^{m \times p}$  with respect to the field  $\mathbb{K}$  and the norm  $\|\cdot\|$  is defined by*

$$(6) \quad \mu_{\mathbb{K}}^{\|\cdot\|}(H) = \left[ \inf \{ \|\Delta\| : \Delta \in \mathbb{K}^{p \times m}, \det(I - H\Delta) = 0 \} \right]^{-1}.$$

We use the convention that taking the infimum over the empty set always yields  $\infty$  and that  $\infty^{-1} = 0$ , so that  $\mu_{\mathbb{K}}^{\|\cdot\|}(0) = 0$ . Definition 2.2 defines the *real  $\mu$ -value* when  $\mathbb{K} = \mathbb{R}$  for complex matrices  $H$  as well as real matrices. In particular, when  $mp = 1$  with  $H \in \mathbb{C} \setminus \mathbb{R}$ , the equation  $\det(I - H\Delta) = 0$  is not solvable for real  $\Delta$ , and so  $\mu_{\mathbb{R}}^{\|\cdot\|}(H)$  is zero. On the other hand, when  $mp > 1$ , the equation  $\det(I - H\Delta) = 0$  is generically solvable for real  $\Delta$ , even if  $H$  is complex, and hence  $\mu_{\mathbb{R}}^{\|\cdot\|}(H)$  is nonzero.

The following lemma is well known in the case of the spectral norm, where it is usually known as the Eckart–Young theorem. See [QBR<sup>+</sup>95, Lemma 1] and [HP05, Prop. 4.4.11] for extensions to other structures and other operator norms. A key point to note here is that while this result holds both for  $\mathbb{K} = \mathbb{C}$  and  $\mathbb{K} = \mathbb{R}$ , it does *not* hold for the real  $\mu$ -value when  $H$  is complex.

LEMMA 2.3. *Let  $H \in \mathbb{K}^{m \times p}$ , and let  $\|\cdot\|$  be either  $\|\cdot\|_2$  or  $\|\cdot\|_F$ . Then,  $\mu_{\mathbb{K}}^{\|\cdot\|}(H) = \|H\|_2$ .*

*Proof.* If  $H = 0$ , the result is clear. Assume  $H \neq 0$ . If  $\|\Delta\| < \|H\|_2^{-1}$ , then we have from (4) that  $\det(I - H\Delta) \neq 0$ . This shows that  $\mu_{\mathbb{K}}^{\|\cdot\|}(H) \leq \|H\|_2$ . For the reverse inequality, let  $H$  have the singular value decomposition  $U\Sigma V^T$ , where  $U$  and  $V$  are unitary and

$$\Sigma = \text{diag}\{\sigma_1(H), \sigma_2(H), \dots, \sigma_{\min\{p,m\}}(H)\}$$

is a diagonal matrix with singular values on the diagonal in descending order by magnitude. Define

$$\Delta = V \text{diag}\{\sigma_1(H)^{-1}, 0, \dots, 0\} U^T.$$

Then  $\|\Delta\|_F = \|\Delta\|_2 = \|H\|_2^{-1}$  and  $\det(I - H\Delta) = 0$ . Furthermore, if  $\mathbb{K} = \mathbb{R}$ , then since  $H$  is real,  $\Delta$  is also real. This shows that  $\mu_{\mathbb{K}}^{\|\cdot\|}(H) \geq \|H\|_2$ .  $\square$

Combining [HP05, Lemma 5.2.7] with Lemma 2.3 results in the following corollary. This may be compared with [HP05, Corollary 5.2.8], which treats a more general class of structured perturbations but is restricted to operator norms.

COROLLARY 2.4. *Let  $\|\cdot\|$  be either  $\|\cdot\|_2$  or  $\|\cdot\|_F$ . Let  $s \in \mathbb{C} \setminus \sigma(A)$  and  $\|D\| < \mu_{\mathbb{K}}^{\|\cdot\|}(G(s))$ , and define*

$$\varepsilon(s) = \min \{ \|\Delta\| : \Delta \in \mathbb{K}^{p \times m}, s \in \sigma(M(\Delta)) \}.$$

Then,

$$\varepsilon(s) = \left( \mu_{\mathbb{K}}^{\|\cdot\|}(G(s)) \right)^{-1}.$$

This leads to the following theorem, which may be compared to [HP05, Theorem 5.2.9], which again does not treat the Frobenius norm.

THEOREM 2.5. *Let  $\|\cdot\|$  be either  $\|\cdot\|_2$  or  $\|\cdot\|_F$ . Suppose  $\varepsilon > 0$  and  $\varepsilon \|D\|_2 < 1$ . Then*

$$\sigma_{\varepsilon}^{\mathbb{K}, \|\cdot\|}(A, B, C, D) = \sigma(A) \cup \left\{ s \in \mathbb{C} \setminus \sigma(A) : \mu_{\mathbb{K}}^{\|\cdot\|}(G(s)) \geq \varepsilon^{-1} \right\}.$$

*Proof.* Suppose  $s \in \sigma(M(\Delta)) \setminus \sigma(A)$ ,  $\Delta \in \mathbb{K}^{p \times m}$ , and  $\|\Delta\| \leq \varepsilon$ . By [HP05, Lemma 5.2.7], we have  $(\mu_{\mathbb{K}}^{\|\cdot\|}(G(s)))^{-1} \leq \|\Delta\| \leq \varepsilon$ . Conversely, if  $s \in \mathbb{C} \setminus \sigma(A)$  and  $(\mu_{\mathbb{K}}^{\|\cdot\|}(G(s))) \geq \varepsilon^{-1}$ , then we have  $\mu_{\mathbb{K}}^{\|\cdot\|}(G(s)) > \|D\|_2$ , and by Corollary 2.4, there exists  $\Delta$  with  $\|\Delta\| = (\mu_{\mathbb{K}}^{\|\cdot\|}(G(s)))^{-1} = \varepsilon$  such that  $s \in \sigma(M(\Delta))$ .  $\square$

Note that even when  $A, B, C, D$  are real, the transfer function  $G(s)$  is normally complex for  $s \notin \mathbb{R}$ , so it is not generally the case that  $\mu_{\mathbb{R}}^{\|\cdot\|}(G(s)) = \|G(s)\|_2$ . For real spectral value sets defined by the spectral norm, the optimal perturbation that appears in Definition 2.2 of the  $\mu$ -value can in fact always be chosen to have rank at most two [QBR<sup>+</sup>95, sec. 2], leading to the formula

$$\sigma_{\varepsilon}^{\mathbb{R}, \|\cdot\|_2}(A, B, C, D) = \bigcup \{ \sigma(M(\Delta)) : \Delta \in \mathbb{R}^{p \times m}, \|\Delta\|_2 \leq \varepsilon, \text{ and } \text{rank}(\Delta) \leq 2 \}.$$

We make the assumption in this paper that the same property holds for the Frobenius norm. This assumption seems reasonable, because the real part of a complex rank-one perturbation has rank at most two, and as we shall see in sections 3 and 4, such

rank-two real perturbations arise in a natural way. The assumption is also supported by numerical experiments.

We also make a second assumption that  $mp > 1$  because when  $m = p = 1$ , the spectral value set  $\sigma_\varepsilon^{\mathbb{R}, \|\cdot\|}(A, B, C, D)$  has an atypical structure with no interior points, being parametrized by a single real scalar  $\Delta \in [-\varepsilon, \varepsilon]$ .

**2.2. The stability radius.** Because we are focusing on the continuous-time dynamical system (1)–(2), the stability region of interest is the open left half-plane  $\mathbb{C}_-$ . We say that  $A$  is *stable* if  $\sigma(A) \in \mathbb{C}_-$ , in which case, for sufficiently small  $\varepsilon$ , the spectral value set  $\sigma_\varepsilon^{\mathbb{K}, \|\cdot\|}(A, B, C, D)$  is also in  $\mathbb{C}_-$ . The stability radius  $r_{\mathbb{K}}^{\|\cdot\|}$  measures the size of the minimal perturbation that destabilizes the matrix or results in  $M(\Delta)$  being undefined [HP05, Def. 5.3.1].

**DEFINITION 2.6.** *The stability radius  $r_{\mathbb{K}}^{\|\cdot\|}$  is defined with respect to the field  $\mathbb{K}$  and the norm  $\|\cdot\|$  as*

$$r_{\mathbb{K}}^{\|\cdot\|}(A, B, C, D) = \inf\{\|\Delta\| : \Delta \in \mathbb{K}^{p \times m}, \det(I - D\Delta) = 0 \text{ or } \sigma(M(\Delta)) \not\subset \mathbb{C}_-\}.$$

The characterization

$$(7) \quad r_{\mathbb{K}}^{\|\cdot\|}(A, B, C, D) = \min \left( \left[ \mu_{\mathbb{K}}^{\|\cdot\|}(D) \right]^{-1}, \inf_{\omega \in \mathbb{R}} \left[ \mu_{\mathbb{K}}^{\|\cdot\|}(G(i\omega)) \right]^{-1} \right)$$

is well known for operator norms [HP05, Theorem 5.3.3]. Corollary 2.4 and Lemma 2.3 extend [HP05, Theorem 5.3.3] beyond operator norms to the Frobenius norm leading to the formula

$$(8) \quad r_{\mathbb{K}}^{\|\cdot\|}(A, B, C, D) = \min \left( \|D\|_2^{-1}, \inf_{\omega \in \mathbb{R}} \left[ \mu_{\mathbb{K}}^{\|\cdot\|}(G(i\omega)) \right]^{-1} \right),$$

where  $\|\cdot\|$  is either the spectral or the Frobenius norm.

*Remark 2.7.* As the  $\mu_{\mathbb{R}}^{\|\cdot\|}$  function is upper semicontinuous both for operator norms and the Frobenius norm (see [Kar03, Lemma 1.7.1]), we have  $\lim_{|\omega| \rightarrow \infty} G(i\omega) = D$  but

$$(9) \quad \liminf_{|\omega| \rightarrow \infty} \left[ \mu_{\mathbb{K}}^{\|\cdot\|}(G(i\omega)) \right]^{-1} \geq \left[ \mu_{\mathbb{K}}^{\|\cdot\|}(D) \right]^{-1} = \|D\|_2^{-1}$$

with a possible strict inequality (see [HP05, Remark 5.3.17(i)] and [HP05, Example 5.3.18] for an example with  $p = 1$ ). Therefore, when  $D \neq 0$ , we cannot eliminate the first term in (8). Either  $r_{\mathbb{K}}^{\|\cdot\|}(A, B, C, D) = \|D\|_2^{-1}$  or the infimum in (8) is strictly less than  $\|D\|_2^{-1}$ , in which case it has to be attained at a finite  $\omega$ ; otherwise, we would obtain a contradiction as  $|\omega| \rightarrow \infty$  by the inequality (9). However, in the special case when  $D = 0$ , we can interpret  $[\mu_{\mathbb{K}}^{\|\cdot\|}(D)]^{-1} = \|D\|_2^{-1} = \infty$  (see the paragraph after Definition 2.2) and eliminate the first term in (8).

In the complex case  $\mathbb{K} = \mathbb{C}$ , the spectral norm and Frobenius norms define the same stability radius  $r_{\mathbb{C}}^{\|\cdot\|}$ . In this case also we can eliminate the first term in (8), since (9) holds with equality, and the second term is simply the reciprocal of the  $H_\infty$  norm of the transfer matrix  $G$  on the boundary of the stability region. The standard method to compute it is the Boyd–Balakrishnan–Bruinsma–Steinbuch (BBBS) algorithm [BB90, BS90]. This algorithm is globally and quadratically convergent but is not practical

when  $n$  is large due to its computational complexity: it requires repeated computation of all eigenvalues of  $2n \times 2n$  Hamiltonian matrices. The first constructive formula to compute  $\mu_{\mathbb{R}}^{\|\cdot\|_2}$  and hence  $r_{\mathbb{R}}^{\|\cdot\|_2}$ , the real  $\mu$ -value and the real stability radius for the spectral norm, was given in [QBR<sup>+</sup>95]; this led to a practical level-set algorithm [SVDT96]. However, this is significantly more involved than the BBBS algorithm and hence is also impractical in the large-scale case. As noted in the introduction, much less attention has been given to the Frobenius-norm case, though it is clearly of interest in applications. As far as we know, no constructive method has been given to approximate  $\mu_{\mathbb{R}}^{\|\cdot\|_F}$  or  $r_{\mathbb{R}}^{\|\cdot\|_F}$ , even if  $n$  is small.

**2.3. The spectral value set abscissa.** The spectral value set abscissa measures how far the spectral value set extends rightward into the complex plane for a prescribed value of  $\varepsilon$ .

DEFINITION 2.8. For  $\varepsilon \geq 0$ ,  $\varepsilon\|D\|_2 < 1$ , the spectral value set abscissa (with respect to the norm  $\|\cdot\|$  and the field  $\mathbb{K}$ ) is

$$(10) \quad \alpha_{\varepsilon}^{\mathbb{K},\|\cdot\|}(A, B, C, D) = \max\{\operatorname{Re} \lambda : \lambda \in \sigma_{\varepsilon}^{\mathbb{K},\|\cdot\|}(A, B, C, D)\}$$

with  $\alpha_0^{\mathbb{K},\|\cdot\|}(A, B, C, D) = \alpha(A)$ , the spectral abscissa of  $A$ .

In the case  $\mathbb{K} = \mathbb{R}$ ,  $\alpha_{\varepsilon}^{\mathbb{R},\|\cdot\|}(A, B, C, D)$  is called the *real spectral value set abscissa*.

DEFINITION 2.9. A rightmost point of a set  $S \subset \mathbb{C}$  is a point where the maximal value of the real part of the points in  $S$  is attained. A locally rightmost point of a set  $S \subset \mathbb{C}$  is a point  $\lambda$  which is a rightmost point of  $S \cap \mathcal{N}$  for some neighborhood  $\mathcal{N}_{\delta} = \{s : |s - \lambda| < \delta\}$  of  $\lambda$  with  $\delta > 0$ .

Remark 2.10. Since  $A, B, C, D$  are real,  $\sigma_{\varepsilon}^{\mathbb{K},\|\cdot\|}(A, B, C, D)$  is symmetric with respect to the real axis, so without loss of generality, when we refer to a rightmost  $\lambda$  in  $\sigma_{\varepsilon}^{\mathbb{K},\|\cdot\|}$ , we imply that  $\lambda \in \overline{\mathbb{H}}$ , the closed upper half-plane, and when we say that the rightmost point  $\lambda$  is unique, we mean considering only points in  $\overline{\mathbb{H}}$ . The same convention applies to the spectrum, so that a rightmost eigenvalue is understood to be in  $\overline{\mathbb{H}}$ .

There is a key relationship between the spectral value set abscissa and the stability radius that is a consequence of Theorem 2.5.

COROLLARY 2.11.

$$(11) \quad r_{\mathbb{K}}^{\|\cdot\|}(A, B, C, D) = \inf \left\{ \varepsilon : \varepsilon\|D\|_2 < 1 \text{ or } \alpha_{\varepsilon}^{\mathbb{K},\|\cdot\|}(A, B, C, D) \geq 0 \right\}$$

$$(12) \quad = \min \left( \|D\|_2^{-1}, \inf \left\{ \varepsilon : \alpha_{\varepsilon}^{\mathbb{K},\|\cdot\|}(A, B, C, D) \geq 0 \right\} \right).$$

*Proof.* That the right-hand sides of (11) and (12) are the same is immediate. Hence, it suffices to show that both of the infimum terms in (8) and (12) are attained and are equal when the upper bound  $\|D\|_2^{-1}$  is not active. The infimum in (12) is attained because  $\alpha_{\varepsilon}^{\mathbb{K},\|\cdot\|}(A, B, C, D)$  is a monotonically increasing continuous function of  $\varepsilon$  (see [Kar03, Chapter 2] for continuity properties of real spectral value sets), and the infimum in (8) is attained at a finite point by Remark 2.7. Finally, the infimal values are equal by Theorem 2.5.  $\square$

The algorithm developed in this paper for approximating the real stability radius  $r_{\mathbb{R}}^{\|\cdot\|_F}$  when  $n$  is large depends on the fundamental characterization (12). This

was also true of the recent algorithms developed in [GGO13, BV14, MO16] for the complex stability radius  $r_{\mathbb{C}}^{\|\cdot\|}$  when  $n$  is large, for which the equivalence (12) is more straightforward and well known.<sup>1</sup>

**3. Rightward trajectories of spectral value set eigenvalues.** In this section, let  $\varepsilon > 0$  be fixed with  $\varepsilon \|D\|_2 < 1$ . We consider the variational behavior of eigenvalues of the perturbed system matrix  $M(\Delta)$  defined in (3). It is convenient to consider a smooth parametrization  $t \mapsto \Delta(t)$  mapping  $\mathbb{R}$  to  $\mathbb{R}^{p \times m}$ . We will use  $\dot{\Delta}$  to denote the derivative  $(d/dt)\Delta(t)$ .

**3.1. An ordinary differential equation approach.** We extend the method of Guglielmi and Lubich [GL13, sec. 2.1] for approximating the real pseudospectral abscissa (the real spectral value set abscissa in the case  $B = C = I, D = 0$ ) to the spectral value set abscissa  $\alpha_{\varepsilon}^{\mathbb{R}, \|\cdot\|_F}$  for general  $A, B, C, D$ , using the Frobenius norm. As we shall see, the extension is not straightforward as additional subtleties arise in the general case that are not present in the pseudospectral case.

We need the following lemma.

LEMMA 3.1. *Given a smooth parametrization  $\Delta(t)$  with  $\|\Delta(t)\|_F \|D\|_2 < 1$ , we have*

$$(13) \quad \frac{d}{dt} \left( \Delta(t) (I - D\Delta(t))^{-1} \right) = (I - \Delta(t)D)^{-1} \dot{\Delta}(t) (I - D\Delta(t))^{-1}.$$

*Proof.* For conciseness, we omit the dependence on  $t$ , differentiate, and regroup terms as

$$(14) \quad \begin{aligned} \frac{d}{dt} \left( \Delta (I - D\Delta)^{-1} \right) &= \dot{\Delta} (I - D\Delta)^{-1} + \Delta \frac{d}{dt} (I - D\Delta)^{-1} \\ &= \dot{\Delta} (I - D\Delta)^{-1} + \Delta (I - D\Delta)^{-1} D \dot{\Delta} (I - D\Delta)^{-1} \\ &= \left( I + \Delta (I - D\Delta)^{-1} D \right) \dot{\Delta} (I - D\Delta)^{-1}. \end{aligned}$$

We then observe that

$$(15) \quad I + \Delta (I - D\Delta)^{-1} D = I + \Delta \left( \sum_{k=0}^{\infty} (D\Delta)^k \right) D = I + \sum_{k=1}^{\infty} (\Delta D)^k = (I - \Delta D)^{-1}.$$

Combining (14) and (15) yields the result. □

The following definition from [GO11, MO16] is useful.

DEFINITION 3.2. *Let  $\lambda$  be a simple eigenvalue of a matrix  $M$  with associated right eigenvector  $x$  satisfying  $Mx = \lambda x$  and left eigenvector  $y$  satisfying  $y^*M = \lambda y^*$ . We refer to  $(\lambda, x, y)$  as an RP-compatible eigentriple of  $M$  if  $y^*x$  is real and positive and  $\|x\| = \|y\| = 1$ , and as a rightmost eigentriple if  $\lambda$  is a rightmost eigenvalue of  $M$  in  $\mathbb{H}$ .*

Note that if  $(\lambda, x, y)$  is an RP-compatible eigentriple of  $M$ , so is  $(\lambda, e^{i\theta}x, e^{i\theta}y)$  for any  $\theta \in [0, 2\pi)$ .

Then we have the following lemma.

---

<sup>1</sup>For a different approach to approximating  $r_{\mathbb{C}}^{\|\cdot\|}$  when  $n$  is large, namely the “implicit determinant” method, see [FSVD14].

LEMMA 3.3. *Given a smooth parametrization  $\Delta(t)$  with  $\|\Delta(t)\|_F \|D\|_2 < 1$ , let  $\lambda(t)$  be a continuously varying simple eigenvalue of*

$$M(\Delta(t)) = A + B\Delta(t)(I - D\Delta(t))^{-1}C.$$

Then  $\lambda(t)$  is differentiable with

$$(16) \quad \operatorname{Re} \dot{\lambda}(t) = \frac{1}{y(t)^*x(t)} \operatorname{Re} \left( u(t)^* \dot{\Delta}(t)v(t) \right),$$

where  $(\lambda(t), x(t), y(t))$  is an RP-compatible eigentriple of  $M(\Delta(t))$  and

$$u(t) = (I - \Delta(t)D)^{-T} B^T y(t), \quad v(t) = (I - D\Delta(t))^{-1} Cx(t).$$

*Proof.* Applying standard eigenvalue perturbation theory [HJ90, Theorem 6.3.12], together with Lemma 3.1, we find that  $\lambda$  is differentiable with

$$(17) \quad \dot{\lambda} = \frac{y^* \dot{M}(\Delta) x}{y^* x} \quad \text{and} \quad \dot{M}(\Delta) = B(I - \Delta D)^{-1} \dot{\Delta} (I - D\Delta)^{-1} C,$$

where we omitted the dependence on  $t$  for conciseness. The result now follows.  $\square$

Now define  $E(t) = \Delta(t)/\varepsilon$ , so that  $\|\Delta(t)\|_F = \varepsilon$  when  $\|E(t)\|_F = 1$ . Then we have

$$(18) \quad \operatorname{Re} \dot{\lambda}(t) = \frac{\varepsilon}{y(t)^*x(t)} \operatorname{Re} \left( u(t)^* \dot{E}(t)v(t) \right) = \frac{\varepsilon}{y(t)^*x(t)} \left\langle \operatorname{Re} (u(t)v(t)^*), \dot{E}(t) \right\rangle,$$

where for  $R, S \in \mathbb{R}^{p \times m}$ ,

$$\langle R, S \rangle = \operatorname{Tr} R^T S = \sum_{i,j} R_{ij} S_{ij},$$

the trace inner product on  $\mathbb{R}^{p \times m}$  associated with the Frobenius norm. The condition that  $\|E(t)\|_F$  is constant is equivalent to

$$(19) \quad \frac{d}{dt} \|E(t)\|_F^2 = 2 \left\langle E(t), \dot{E}(t) \right\rangle = 0 \quad \forall t.$$

Our aim, given  $t$ , is to choose  $\dot{E}(t)$  to maximize (18) subject to the constraint (19), leading to an optimization problem whose solution is given by the following lemma. The proof is a straightforward application of first-order optimality conditions; see also [GL13, Lemma 2.4].

LEMMA 3.4. *Let  $E \in \mathbb{R}^{p \times m}$  have unit Frobenius norm, and let  $u \in \mathbb{C}^p, v \in \mathbb{C}^m$  be given complex vectors such that  $\operatorname{Re}(uv^*) \neq 0$ . Under the assumption that  $mp > 1$  (see the end of section 2.1), a solution to the optimization problem*

$$(20) \quad \tilde{Z} = \arg \max_{Z \in \Omega} \operatorname{Re}(u^* Z v), \quad \Omega = \{Z \in \mathbb{R}^{p \times m} : \|Z\|_F = 1, \langle E, Z \rangle = 0\}$$

exists, and it satisfies

$$(21) \quad \tau \tilde{Z} = \left( \operatorname{Re}(uv^*) - \langle E, \operatorname{Re}(uv^*) \rangle E \right),$$

where  $\tau$  is the Frobenius norm of the matrix on the right-hand side in (21).

This suggests consideration of the following ODE:

$$(22) \quad \dot{E}(t) = \operatorname{Re}(u(t)v(t)^*) - \langle E(t), \operatorname{Re}(u(t)v(t)^*) \rangle E(t),$$

with  $u(t)$  and  $v(t)$  defined by

$$(23) \quad u(t) = (I - \varepsilon E(t)D)^{-\top} B^\top y(t), \quad v(t) = (I - \varepsilon DE(t))^{-1} Cx(t),$$

where  $(\lambda(t), x(t), y(t))$  is a rightmost RP-compatible eigentriple for the matrix  $M(\varepsilon E(t))$  (see Definition 3.2). For the remainder of this paper, we make use of the following assumption, which we note is generically true.

*Assumption 3.5.* Let  $E_0 \in \mathbb{R}^{p \times m}$  satisfy  $\|E_0\|_F = 1$ . We assume that the rightmost eigenvalue of  $M(\varepsilon E_0)$  in  $\overline{\mathbb{H}}$  is simple and unique.

**3.2. Equilibrium points of the ODE.** We now focus on the properties of the ODE (22), in particular characterizing equilibrium points.

**THEOREM 3.6.** *Consider the differential equation (22) with initial condition  $E(0) = E_0$ . There exists  $t_{\max} \in (0, \infty]$  such that, for all  $t \in [0, t_{\max})$ , the solution  $E(t)$  exists and is unique and the following all hold:*

- (I) *The rightmost eigenvalue  $\lambda(t)$  of  $M(\varepsilon E(t))$  in  $\overline{\mathbb{H}}$  is simple and unique.*
- (II)  $\|E(t)\|_F = 1$ .
- (III)  $\operatorname{Re} \lambda(t) \geq 0$ .

Furthermore, at a given value  $t \in [0, t_{\max})$ , the following three conditions are equivalent:

- (i)  $\operatorname{Re} \dot{\lambda}(t) = 0$ .
- (ii) *One of the following two mutually exclusive conditions holds:*

$$(24) \quad \operatorname{Re}(u(t)v(t)^*) = 0 \quad \text{or} \quad E(t) = \frac{\operatorname{Re}(u(t)v(t)^*)}{\|\operatorname{Re}(u(t)v(t)^*)\|_F}.$$

- (iii)  $\dot{E}(t) = 0$ .

*Proof.* (I) Because the rightmost eigenvalue in  $\overline{\mathbb{H}}$  is simple and unique for  $t = 0$  (by Assumption 3.5), the ODE is well defined and the same properties must hold for sufficiently small positive  $t$ . (II) Taking the trace inner product of  $E(t)$  with the ODE (22), we find that (19) holds when  $\|E(t)\|_F = 1$ , so the norm is preserved by the ODE. (III) Substituting the ODE (22) into (18), we obtain from Cauchy–Schwarz that

$$(25) \quad \operatorname{Re} \dot{\lambda}(t) = \frac{\varepsilon}{y(t)^* x(t)} \left[ \|\operatorname{Re}(u(t)v(t)^*)\|_F^2 - \langle \operatorname{Re}(u(t)v(t)^*), E(t) \rangle^2 \right] \geq 0,$$

establishing (III). Clearly, equality holds in (25) at  $t$  if and only if one of the two alternatives in (24) holds, so (i) and (ii) are equivalent. Furthermore, from the definition of the ODE (22), it is clear that (ii) implies (iii), while the only way the right-hand side of (22) can be zero is if one of the two alternatives in (ii) holds. Hence, (ii) and (iii) are equivalent.  $\square$

Recalling that  $\Delta(t) = \varepsilon E(t)$ , we know that the eigenvalues of  $M(\Delta(t))$  all lie in the spectral value set  $\sigma_{\varepsilon}^{\mathbb{R}, \|\cdot\|_F}(A, B, C, D)$ , with the rightmost eigenvalue  $\lambda(t)$  moving monotonically to the right with respect to  $t$ . In particular, if  $E_0$  has unit Frobenius norm and its rightmost eigenvalue  $\lambda(0)$  is simple and unique and is already a locally rightmost point of the spectral value set  $\sigma_{\varepsilon}^{\mathbb{R}, \|\cdot\|_F}(A, B, C, D)$ , it must be the

case that  $E_0$  is an equilibrium point of the ODE with  $\operatorname{Re} \dot{\lambda}(0) = 0$ , since  $\operatorname{Re} \dot{\lambda}(0) > 0$  would violate the definition of a point being locally rightmost and  $\operatorname{Re} \dot{\lambda}(0) < 0$  would violate (3) in Theorem 3.6. The hope is that for  $E_0$  with  $\lambda(0)$  in the interior of  $\sigma_\varepsilon^{\mathbb{R}, \|\cdot\|_F}(A, B, C, D)$ , the solution  $E(t)$  of the ODE will have its rightmost eigenvalue  $\lambda(t)$  converging to a locally rightmost point of  $\sigma_\varepsilon^{\mathbb{R}, \|\cdot\|_F}(A, B, C, D)$ , although this cannot be guaranteed, just as was the case with the algorithm of [GO11] for approximating the rightmost points of pseudospectra, and algorithms developed in subsequent papers (see section 1). A new consideration here is the possibility that  $\operatorname{Re}(u(t)v(t)^*) = 0$  at an equilibrium point. We do not know whether convergence to such an equilibrium point is possible or not. In the special case of pseudospectra, that is, with  $B = C = I$  and  $D = 0$ , the outer product  $u(t)v(t)^*$  reduces to  $y(t)x(t)^*$ , whose real part *cannot* be zero, as was shown in [GL13, sec. 2.1.6]; in fact, the proof given there shows that  $\operatorname{Re}(y(t)x(t)^*)$  has rank one when  $\lambda(t)$  is real and rank two when it is complex. Note that the condition  $\operatorname{Re}(u(t)v(t)^*) = 0$  generalizes the notions of uncontrollability and unobservability, because if  $\lambda(t)$  is unobservable, then  $B^\top y(t) = 0$ , implying  $u(t) = 0$ , while if it is uncontrollable, then  $Cx(t) = 0$ , implying  $v(t) = 0$ .

The next lemma explicitly states formulas for  $\operatorname{Re}(u(t)v(t)^*)$  and bounds on its rank.

LEMMA 3.7. *Fix  $t < t_{\max}$ , and let  $u \in \mathbb{C}^p, v \in \mathbb{C}^m$  be defined by (23) for some vectors  $y(t) = y$  and  $x(t) = x$ . If  $\lambda = \lambda(t) \in \mathbb{R}$ , then we can choose  $y, x, u$ , and  $v$  to be real, with  $\operatorname{Re}(uv^*) = uv^\top$  having rank one. If  $\lambda \notin \mathbb{R}$ , set  $X = (\operatorname{Re} x, \operatorname{Im} x) \in \mathbb{R}^{n \times 2}$ ,  $Y = (\operatorname{Re} y, \operatorname{Im} y) \in \mathbb{R}^{n \times 2}$ , so  $\operatorname{Re}(yx^*) = YX^\top$ . Then*

$$\operatorname{Re}(uv^*) = (I - \varepsilon ED)^{-\top} B^\top YX^\top C^\top (I - \varepsilon DE)^{-\top}$$

with

$$\operatorname{rank}(\operatorname{Re}(uv^*)) = \operatorname{rank}(B^\top YX^\top C^\top) \leq 2.$$

Furthermore, if  $\min(p, m) = 1$ , then  $\operatorname{rank}(\operatorname{Re}(uv^*)) \leq 1$ .

*Proof.* The first statement follows from the definition (23), noting that  $E$  and  $D$  are real. The rank results follow from submultiplicativity.  $\square$

As already mentioned, the argument given in [GL13, sec. 2.1.6] shows that when  $\lambda$  is not real, the matrix  $YX^\top$  has rank two, so when  $\min(p, m) \geq 2$ , we can expect that  $UV^\top$  will also have rank two for generic  $B$  and  $C$ .

A natural idea would be to attempt to approximate  $\alpha_\varepsilon^{\mathbb{R}, \|\cdot\|_F}(A, B, C, D)$  by integrating the ODE (22) numerically to determine its equilibria, guaranteeing monotonicity by step-size control. However, a serious drawback of this approach is the fact that the solution  $E(t)$  (and hence most likely its discretization) does not preserve the low-rank structure even if both the initial point  $E_0$  and the limit of  $E(t)$  as  $t \rightarrow \infty$  both have rank two. Although it is possible to consider an ODE defined on the manifold of rank-two matrices, as done in [GL13] and [GM15] for the special case  $B = C = I, D = 0$ , we instead develop an efficient discrete iteration that is nonetheless based on the ODE (22).

**4. An iterative method to approximate the Frobenius-norm real spectral value set abscissa.** As in the previous section, assume  $\varepsilon$  is fixed with  $\varepsilon \|D\|_2 < 1$ . Following an idea briefly mentioned in [GM15], we consider the following implicit-explicit Euler discretization of (22) with a variable step-size  $h_k$ :

$$(26) \quad E_{k+1} = E_k + h_{k+1} \left( \operatorname{Re}(u_{k+1}v_{k+1}^*) - \langle E_{k+1}, \operatorname{Re}(u_{k+1}v_{k+1}^*) \rangle E_k \right),$$

where

$$u_{k+1} = (I - \varepsilon E_k D)^{-T} B^T y_k, \quad v_{k+1} = (I - \varepsilon D E_k)^{-1} C x_k,$$

and  $(\lambda_k, x_k, y_k)$  is a rightmost RP-compatible eigentriple of  $M(\varepsilon E_k)$ . The method is clearly consistent and converges with order 1 with respect to  $h_k$ .

LEMMA 4.1. *Let  $u_0, v_0$  be given complex vectors with  $\operatorname{Re}(u_0 v_0^*) \neq 0$ , and set  $E_0 = \operatorname{Re}(u_0 v_0^*) / \|\operatorname{Re}(u_0 v_0^*)\|_F$ . Let  $h_k = 1 / \|\operatorname{Re}(u_k v_k^*)\|_F$ . Then the difference equation (26) has the solution*

$$(27) \quad E_{k+1} = \frac{\operatorname{Re}(u_{k+1} v_{k+1}^*)}{\|\operatorname{Re}(u_{k+1} v_{k+1}^*)\|_F}$$

as long as the rightmost eigenvalue of  $M(\varepsilon E_k)$  is simple and unique (considering only those in  $\overline{\mathbb{H}}$ ) and as long as  $\operatorname{Re}(u_{k+1} v_{k+1}^*) \neq 0$  for all  $k = 0, 1, \dots$

*Proof.* The result is easily verified by substituting (27) into (26). The assumptions ensure that the difference equation is well defined.  $\square$

Equivalently, let  $E_k = U_k V_k^T$  be the current perturbation, with  $\|U_k V_k^T\|_F = 1$ , and let  $(\lambda_k, x_k, y_k)$  be a rightmost eigenvalue of  $M(\varepsilon U_k V_k^T)$ . Then by setting

$$X_k = (\operatorname{Re} x_k, \operatorname{Im} x_k), \quad Y_k = (\operatorname{Re} y_k, \operatorname{Im} y_k)$$

we can write (27) in the form

$$(28) \quad E_{k+1} = U_{k+1} V_{k+1}^T \text{ with } \|U_{k+1} V_{k+1}^T\|_F = 1,$$

where

$$(29) \quad \widehat{U}_{k+1} = (I - \varepsilon U_k V_k^T D)^{-T} B^T Y_k,$$

$$(30) \quad \widehat{V}_{k+1} = (I - \varepsilon D U_k V_k^T)^{-1} C X_k,$$

$$(31) \quad \beta_{k+1} = \|\widehat{U}_{k+1} \widehat{V}_{k+1}^T\|_F^{-1},$$

$$(32) \quad U_{k+1} = \sqrt{\beta_{k+1}} \widehat{U}_{k+1},$$

$$(33) \quad V_{k+1} = \sqrt{\beta_{k+1}} \widehat{V}_{k+1}.$$

Since  $E_k = U_k V_k^T$  has rank at most two, we can simplify these expressions using the Sherman–Morrison–Woodbury formula [GV83] as follows:

$$(34) \quad (I - \varepsilon U_k V_k^T D)^{-1} = I + \varepsilon U_k (I - \varepsilon V_k^T D U_k)^{-1} V_k^T D,$$

$$(35) \quad (I - \varepsilon D U_k V_k^T)^{-1} = I + \varepsilon D U_k (I - \varepsilon V_k^T D U_k)^{-1} V_k^T.$$

Note that  $I - \varepsilon V_k^T D U_k \in \mathbb{R}^{2 \times 2}$  and is invertible since  $\varepsilon \|D\|_2 < 1$  by assumption. The second formula (35) can also be used to simplify the definition of the perturbed system matrix in (3) as follows:

$$(36) \quad \begin{aligned} M(\Delta_k) &= M(\varepsilon E_k) = M(\varepsilon U_k V_k^T) \\ &= A + \varepsilon B U_k V_k^T (I - \varepsilon D U_k V_k^T)^{-1} C \\ &= A + \varepsilon B U_k V_k^T (I + \varepsilon D U_k (I - \varepsilon V_k^T D U_k)^{-1} V_k^T) C \\ &= A + (\varepsilon B U_k) \left[ I + \varepsilon (V_k^T D U_k) (I - \varepsilon V_k^T D U_k)^{-1} \right] (V_k^T C). \end{aligned}$$

The product  $U_k V_k^T$  is never computed explicitly, but is retained in factored form, so that the eigenvalues of  $M(\varepsilon_k U_k V_k^T)$  with largest real part can be computed efficiently by an iterative method. The Frobenius norm of the product can be obtained using the following equivalence:

$$\|UV^T\|_F = [\text{Tr}(VU^TUV^T)]^{\frac{1}{2}} = [\text{Tr}((U^T U)(V^T V))]^{\frac{1}{2}},$$

which requires only inner products to compute the  $2 \times 2$  matrices  $U^T U$  and  $V^T V$ .

As with the spectral value set abscissa (SVSA) iteration for complex spectral values sets given in [GGO13], there is no guarantee that the full update step for the real Frobenius-norm bounded case will satisfy monotonicity; that is,  $\text{Re } \lambda_{k+1} > \text{Re } \lambda_k$  may or may not hold, where  $\lambda_k$  is a rightmost eigenvalue of (36). However, the line search approach to make a monotonic variant [GGO13, sec. 3.5] does extend to the real rank-two iteration described above. Given the current perturbation  $U_k V_k^T$ , with  $\|U_k V_k^T\|_F = 1$ , and  $U_{k+1} V_{k+1}^T$  the updated perturbation described above, also normalized so that  $\|U_{k+1} V_{k+1}^T\|_F = 1$ , let

$$U_{\text{LS}}(t) := tU_{k+1} + (1-t)U_k \text{ and } V_{\text{LS}}(t) := tV_{k+1} + (1-t)V_k.$$

Consider the evolution of a continuously varying simple rightmost eigenvalue  $\lambda_{\text{LS}}(t)$  defined on  $t \in [0, 1]$  with RP-compatible right and left eigenvectors  $x(t)$  and  $y(t)$  of

$$(37) \quad M(\Delta_{\text{LS}}(t)) = M_{\text{LS}}(t) = A + B\Delta_{\text{LS}}(t)(I - D\Delta_{\text{LS}}(t))^{-1}C,$$

where

$$(38) \quad \Delta_{\text{LS}}(t) = \frac{\varepsilon U_{\text{LS}}(t)V_{\text{LS}}(t)^T}{\|U_{\text{LS}}(t)V_{\text{LS}}(t)^T\|}$$

is a normalized interpolation between the current perturbation and the full step update. Then by eigenvalue perturbation theory we have that

$$(39) \quad \text{Re } \dot{\lambda}_{\text{LS}}(t) = \frac{y(t)^* \dot{M}_{\text{LS}}(t)x(t)}{y(t)^* x(t)}.$$

Though the derivation is quite lengthy [Mit14, sec. 6.3.3], it holds that if (39) is negative, then flipping the signs of both  $U_{k+1}$  and  $V_{k+1}$  will also flip the sign of (39), thus making it positive. Hence, a line search can be employed to find a  $t \in (0, 1)$  such that  $\text{Re } \lambda_{k+1} > \text{Re } \lambda_k$  is guaranteed to hold provided that (39) is not zero.

Algorithm SVSA-RF summarizes our algorithm to approximate  $\alpha_\varepsilon^{\mathbb{R}, \|\cdot\|_F}(A, B, C, D)$ , the real structured Frobenius-norm bounded spectral value set abscissa, which by construction is guaranteed to find a lower bound.

**5. Approximating the real stability radius by hybrid expansion-contraction.** Recall the relationship between the stability radius  $r_{\mathbb{R}}^{\|\cdot\|}(A, B, C, D)$  and the spectral value set abscissa  $\alpha_\varepsilon^{\mathbb{R}, \|\cdot\|}(A, B, C, D)$  given in (12), which we write here for the real Frobenius-norm case:

$$(40) \quad r_{\mathbb{R}}^{\|\cdot\|_F}(A, B, C, D) = \min \left( \|D\|_2^{-1}, \inf \left\{ \varepsilon : \alpha_\varepsilon^{\mathbb{R}, \|\cdot\|_F}(A, B, C, D) \geq 0 \right\} \right).$$

The interesting case is when the second term is the lesser of these two terms, and for the remainder of the paper we assume this is the case. It follows that  $r_{\mathbb{R}}^{\|\cdot\|_F}(A, B, C, D)$

---

**Algorithm SVSA-RF:** (Spectral value set abscissa: Real Frobenius-norm)

---

**Purpose:** to approximate  $\alpha_{\varepsilon}^{\mathbb{R}, \|\cdot\|_F}(A, B, C, D)$

**Input:**  $\varepsilon \in (0, \|D\|_2^{-1})$ ,  $U_0 \in \mathbb{R}^{p \times 2}$ , and  $V_0 \in \mathbb{R}^{m \times 2}$ , such that  $\|U_0 V_0^T\|_F = 1$ , along with eigentriple  $(\lambda_0, x_0, y_0)$ , with  $\lambda_0$  a rightmost eigenvalue of  $M(\varepsilon U_0 V_0^T)$

**Output:** final iterates  $U_k, V_k$  with  $\|U_k V_k^T\|_F = 1$  along with  $\lambda_k$ , a rightmost eigenvalue of  $M(\varepsilon U_k V_k^T)$ , certifying that  $\text{Re } \lambda_k \leq \alpha_{\varepsilon}^{\mathbb{R}, \|\cdot\|_F}(A, B, C, D)$

- 1: **for**  $k = 0, 1, 2, \dots$  **do**
  - 2:    // Compute the new perturbation
  - 3:     $X_k := (\text{Re } x_k, \text{Im } x_k)$
  - 4:     $Y_k := (\text{Re } y_k, \text{Im } y_k)$
  - 5:     $\widehat{U}_{k+1} := \left( I + \varepsilon U_k (I - \varepsilon V_k^T D U_k)^{-1} V_k^T D \right)^T B^T Y_k$
  - 6:     $\widehat{V}_{k+1} := \left( I + \varepsilon D U_k (I - \varepsilon V_k^T D U_k)^{-1} V_k^T \right) C X_k$
  - 7:    // Normalize the new perturbation
  - 8:     $\beta_{k+1} := \left[ \text{Tr} \left( \left( \widehat{U}_{k+1}^T \widehat{U}_{k+1} \right) \left( \widehat{V}_{k+1}^T \widehat{V}_{k+1} \right) \right) \right]^{-\frac{1}{2}}$
  - 9:     $U_{k+1} := \sqrt{\beta_{k+1}} \widehat{U}_{k+1}$
  - 10:    $V_{k+1} := \sqrt{\beta_{k+1}} \widehat{V}_{k+1}$
  - 11:    // Attempt the full update step and, if necessary, do a line search
  - 12:     $(\lambda_{k+1}, x_{k+1}, y_{k+1}) :=$  a rightmost eigentriple of  $M(\varepsilon U_{k+1} V_{k+1}^T)$  using (36)
  - 13:    **if**  $\text{Re } \lambda_{k+1} \leq \text{Re } \lambda_k$  **then**
  - 14:      Find new  $\lambda_{k+1}$  via line search using (39) to ensure  $\text{Re } \lambda_{k+1} > \text{Re } \lambda_k$
  - 15:    **end if**
  - 16: **end for**
- 

NOTE: The  $k$ th step of the iteration is well defined if  $U_k V_k^T$  is nonzero and the rightmost eigenvalue of (36) in  $\mathbb{H}$  is simple and unique.

equals the infimum in (40) and from Remark 2.7 that this infimum is attained. Hence  $\alpha_{\varepsilon}^{\mathbb{R}, \|\cdot\|_F}(A, B, C, D) = 0$  for  $\varepsilon = r_{\mathbb{R}, \|\cdot\|_F}^{\|\cdot\|_F}(A, B, C, D)$ . For brevity, we henceforth use  $\varepsilon_{\star}$  to denote the real stability radius  $r_{\mathbb{R}, \|\cdot\|_F}^{\|\cdot\|_F}(A, B, C, D)$ .

Let

$$(41) \quad g(\varepsilon) = \alpha_{\varepsilon}^{\mathbb{R}, \|\cdot\|_F}(A, B, C, D).$$

We wish to find  $\varepsilon_{\star}$ , the root (zero) of the monotonically increasing continuous function  $g$ . However, we do not have a reliable way to evaluate  $g$ : all we have is Algorithm SVSA-RF, which is guaranteed to return a lower bound on the true value. Consequently, if the value returned is negative, we have no assurance that its sign is correct. On the other hand, if the value returned is positive, we are assured that the sign is correct. This observation underlies the hybrid expansion-contraction (HEC) algorithm recently introduced in [MO16] for approximating the complex stability radius, which we now extend to the real Frobenius-norm case.

**5.1. Hybrid expansion-contraction.** For any value of  $\varepsilon$  satisfying  $\varepsilon_{\star} < \varepsilon < \|D\|_2^{-1}$ , there exists a real perturbation matrix  $E$  with  $\|E\|_F = 1$  such that  $M(\varepsilon E)$  has an eigenvalue in the right half-plane. We assume that  $E$  has rank at most two (see the

discussion at the end of section 2.1). See section 7.3 for how an initial destabilizing perturbation  $\varepsilon UV^\top$  can be found.

Let  $U \in \mathbb{R}^{p \times 2}$  and  $V \in \mathbb{R}^{m \times 2}$  be two matrices such that  $\|UV^\top\|_F = 1$ . Consider the following matrix family, where  $U$  and  $V$  are fixed and  $0 < \varepsilon < \|D\|_2^{-1}$ :

$$M_{UV}(\varepsilon) := M(\varepsilon UV^\top) = A + B\varepsilon UV^\top(I - D\varepsilon UV^\top)^{-1}C,$$

and define the function

$$(42) \quad g_{UV}(\varepsilon) := \alpha(M_{UV}(\varepsilon)),$$

the spectral abscissa of  $M_{UV}(\varepsilon)$ . Unlike  $g$ , this function is relatively easy to evaluate at a given  $\varepsilon$ , since all that is required is to compute the rightmost eigenvalue of the matrix  $M_{UV}(\varepsilon)$ , something that we assume can be done efficiently by an iterative method such as the MATLAB function `eigs`, exploiting the equivalence (36). Now, as discussed above, suppose that  $\varepsilon_{UB}$  is known with  $M_{UV}(\varepsilon_{UB})$  having an eigenvalue in the right half-plane. There exists  $\varepsilon_c \in (0, \varepsilon_{UB})$  such that  $g_{UV}(\varepsilon_c) = 0$  because  $g_{UV}$  is continuous,  $g_{UV}(\varepsilon_{UB}) > 0$ , and  $g_{UV}(0) < 0$  (as  $A$  is stable). The contraction phase of the HEC algorithm finds such an  $\varepsilon_c$  by a simple Newton-bisection method, using the derivative of  $g_{UV}(\varepsilon)$  given in section 5.2 below. Note that by definition of  $\varepsilon_*$ , it must be the case that  $\varepsilon_* \leq \varepsilon_c$ .

Once the contraction phase delivers  $\varepsilon_c$  with the rightmost eigenvalue of  $M_{UV}(\varepsilon_c)$  on the imaginary axis, the expansion phase then “pushes” the rightmost eigenvalue of  $M(\varepsilon UV^\top)$  back into the right half-plane using Algorithm SVSA-RF, with  $\varepsilon = \varepsilon_c$  fixed and updating only the perturbation matrices  $U$  and  $V$ . The algorithm repeats this expansion-contraction process in a loop until SVSA-RF can no longer find a new perturbation that moves an eigenvalue off the imaginary axis into the right half-plane. Following [MO16], the method is formally defined in Algorithm HEC-RF. For an illustration of the main idea in the context of the complex stability radius, see [MO16, Fig. 4.1].

Convergence results for the original HEC algorithm developed for the complex stability radius were given in [MO16, Theorem 4.3]. The basic convergence result, that, under suitable assumptions, the sequence  $\{\varepsilon_k\}$  converges to some  $\tilde{\varepsilon} \geq \varepsilon_*$  and the sequence  $\{\operatorname{Re} \lambda_k\}$  converges to zero, can be extended to the real Frobenius-norm case without difficulty. However, the part that characterizes limit points of the sequence  $\{\operatorname{Re} \lambda_k\}$  as stationary points or local maxima of the norm of the transfer function on the stability boundary does not immediately extend to the real Frobenius-norm case, because instead of  $\|G(\mathbf{i}\omega)\|$ , we would have to consider the potentially discontinuous function  $\mu_{\mathbb{R}}^{\|\cdot\|_F}(G(\mathbf{i}\omega))$ . (See the discussion following Definition 2.2, as well as [QBR<sup>+</sup>95, sec. 3].)

**5.2. The derivatives of  $g_{UV}$  and  $g$ .** The contraction phase of the algorithm needs the derivative of  $g_{UV}$  defined in (42) to implement the Newton-bisection method to find a root of  $g_{UV}$ . As we shall see, it is also of interest to relate this to the derivative of  $g$  defined in (41), although this is not actually used in the algorithm. The key tool for obtaining both is Lemma 3.1, which presented the derivative of  $\Delta(t)(I - D\Delta(t))^{-1}$  with respect to  $t$ . Here, the same matrix function depends on  $\varepsilon$ . We denote differentiation with respect to  $\varepsilon$  by  $'$ .

**THEOREM 5.1.** *Let  $O \subset (0, \|D\|_2^{-1})$  be open, and suppose that, for all  $\varepsilon \in O$ , the rightmost eigenvalue  $\lambda_{UV}(\varepsilon)$  of  $M_{UV}(\varepsilon)$  in  $\mathbb{H}$  is simple and unique. Then, for all*

---

**Algorithm HEC-RF:** (Hybrid expansion-contraction: Real Frobenius-norm)

---

**Purpose:** to approximate  $r_{\mathbb{R}}^{\|\cdot\|_F}(A, B, C, D)$

**Input:**  $\varepsilon_0 \in (0, \|D\|_2^{-1})$  and matrices  $U \in \mathbb{R}^{p \times 2}$  and  $V \in \mathbb{R}^{m \times 2}$  with  $\|UV^T\|_F = 1$  and  $g_{UV}(\varepsilon_0) > 0$ , along with  $\lambda_0$ , a rightmost eigenvalue of  $M_{UV}(\varepsilon_0)$  in the right half-plane

**Output:** final value of sequence  $\{\varepsilon_k\}$  such that  $\lambda_k$  is a rightmost eigenvalue of  $M(\varepsilon_k UV^T)$  sufficiently close to the imaginary axis but in the closed right half-plane, certifying that  $\varepsilon_k \geq r_{\mathbb{R}}^{\|\cdot\|_F}(A, B, C, D)$

- 1: **for**  $k = 0, 1, 2, \dots$  **do**
  - 2:   **Contraction:** call a Newton-bisection zero-finding algorithm to compute  $\varepsilon_c \in (0, \varepsilon_k]$  so that  $g_{UV}(\varepsilon_c) = 0$ , along with  $\lambda_c$ , a rightmost eigenvalue of  $M_{UV}(\varepsilon_c)$  on the imaginary axis.
  - 3:   **Expansion:** call Algorithm SVSA-RF with input  $\varepsilon_c, U, V$  to compute  $U_e, V_e$  with  $\|U_e V_e^T\|_F = 1$  and  $\lambda_{k+1}$ , a rightmost eigenvalue of  $M(\varepsilon_c U_e V_e^T)$ , satisfying  $\text{Re } \lambda_{k+1} \geq \text{Re } \lambda_c = 0$ .
  - 4:   Set  $\varepsilon_{k+1} := \varepsilon_c, U := U_e,$  and  $V := V_e$ .
  - 5: **end for**
- 

NOTE: *In practice, we pass eigentriples computed by the contraction phase into the expansion phase and vice versa.*

$\varepsilon \in O, g_{UV}$  is differentiable at  $\varepsilon$  with

$$(43) \quad g'_{UV}(\varepsilon) = \frac{\text{Re} \left( (y_{UV}(\varepsilon)^* B U) \left[ I + \varepsilon (V^T D U) (I - \varepsilon V^T D U)^{-1} \right]^2 (V^T C x_{UV}(\varepsilon)) \right)}{y_{UV}(\varepsilon)^* x_{UV}(\varepsilon)},$$

where  $(\lambda_{UV}(\varepsilon), x_{UV}(\varepsilon), y_{UV}(\varepsilon))$  is a rightmost RP-compatible eigentriple of  $M_{UV}(\varepsilon)$ .

*Proof.* Since  $\lambda_{UV}(\varepsilon)$  is simple and unique,  $x_{UV}(\varepsilon)$  and  $y_{UV}(\varepsilon)$  are well defined (up to a unimodular scalar). Applying Lemma 3.1 with  $\Delta(\varepsilon) \equiv \varepsilon UV^T$ , and using (34)–(35) and  $\Xi := I - \varepsilon V^T D U$ , we have

$$(44) \quad \begin{aligned} M'_{UV}(\varepsilon) &= B(I - \varepsilon UV^T D)^{-1} UV^T (I - \varepsilon DUV^T)^{-1} C \\ &= B(I + \varepsilon U \Xi^{-1} V^T D) UV^T (I + \varepsilon D U \Xi^{-1} V^T) C \\ &= B U (I + \varepsilon \Xi^{-1} V^T D U) (I + \varepsilon V^T D U \Xi^{-1}) V^T C \\ &= B U \left[ I + \varepsilon (V^T D U) (I - \varepsilon V^T D U)^{-1} \right]^2 V^T C, \end{aligned}$$

noting that  $\Xi^{-1}$  and  $V^T D U$  commute, since

$$V^T D U (I - \varepsilon V^T D U)^{-1} = V^T D U \sum_{k=0}^{\infty} (\varepsilon V^T D U)^k = (I - \varepsilon V^T D U)^{-1} V^T D U.$$

Using standard eigenvalue perturbation theory, as in the proof of Lemma 3.3, we have

$$(45) \quad g'_{UV}(\varepsilon) = \text{Re } \lambda'_{UV}(\varepsilon) = \frac{\text{Re} (y_{UV}(\varepsilon)^* M'_{UV}(\varepsilon) x_{UV}(\varepsilon))}{y_{UV}(\varepsilon)^* x_{UV}(\varepsilon)},$$

from which the result follows. □

Now we obtain the derivative of the function  $g$  defined in (41).

**THEOREM 5.2.** *Let  $O \subset (0, \|D\|_2^{-1})$  be open. Suppose the following for all  $\varepsilon \in O$ :*

- (a)  $\lambda(\varepsilon)$  is the unique rightmost point in  $\overline{\mathbb{H}}$  of  $\sigma_{\varepsilon}^{\mathbb{R}, \|\cdot\|_{\mathbb{F}}}(A, B, C, D)$ .
- (b)  $E(\varepsilon)$ , with  $\|E(\varepsilon)\|_{\mathbb{F}} = 1$ , is a smooth matrix function of  $\varepsilon$  such that  $\lambda(\varepsilon)$  from (a) is a simple eigenvalue of  $M(\varepsilon E(\varepsilon))$ . (It must also be a unique rightmost eigenvalue of  $M(\varepsilon E(\varepsilon))$  in  $\overline{\mathbb{H}}$  since otherwise (a) would not hold.)
- (c)  $\operatorname{Re}(u(\varepsilon)v(\varepsilon)^*) \neq 0$ , where

$$(46) \quad u(\varepsilon) = (I - \varepsilon E(\varepsilon)D)^{-\top} B^{\top} y(\varepsilon), \quad v(\varepsilon) = (I - \varepsilon DE(\varepsilon))^{-1} C^{\top} x(\varepsilon),$$

and  $(\lambda(\varepsilon), x(\varepsilon), y(\varepsilon))$  is an RP-compatible eigentriple of  $M(\varepsilon E(\varepsilon))$ .

Then, for any  $\varepsilon \in O$ ,

$$(47) \quad g'(\varepsilon) = \frac{\|\operatorname{Re}(u(\varepsilon)v(\varepsilon)^*)\|_{\mathbb{F}}}{y(\varepsilon)^* x(\varepsilon)}.$$

*Proof.* In this proof, we again apply Lemma 3.1 with  $\Delta(t)$  now replaced by  $\varepsilon E(\varepsilon)$ . Let  $N(\varepsilon) = M(\varepsilon E(\varepsilon))$ . We obtain

$$N'(\varepsilon) = B(I - \varepsilon E(\varepsilon)D)^{-1}(E(\varepsilon) + \varepsilon E'(\varepsilon))(I - \varepsilon DE(\varepsilon))^{-1}C.$$

Again using standard eigenvalue perturbation theory, we have

$$(48) \quad \begin{aligned} g'(\varepsilon) = \operatorname{Re} \lambda'(\varepsilon) &= \frac{\operatorname{Re}(y(\varepsilon)^* N'(\varepsilon)x(\varepsilon))}{y(\varepsilon)^* x(\varepsilon)} \\ &= \frac{\operatorname{Re}(u(\varepsilon)^* E(\varepsilon)v(\varepsilon)) + \varepsilon \operatorname{Re}(u(\varepsilon)^* E'(\varepsilon)v(\varepsilon))}{y(\varepsilon)^* x(\varepsilon)} \\ &= \frac{\langle E(\varepsilon), \operatorname{Re}(u(\varepsilon)v(\varepsilon)^*) \rangle + \varepsilon \langle E'(\varepsilon), \operatorname{Re}(u(\varepsilon)v(\varepsilon)^*) \rangle}{y(\varepsilon)^* x(\varepsilon)}. \end{aligned}$$

Now consider the solution of the differential equation (22) with initial condition  $E_0 = E(\varepsilon)$ . Because  $\lambda(0)$  is the rightmost point of  $\sigma_{\varepsilon}^{\mathbb{R}, \|\cdot\|_{\mathbb{F}}}(A, B, C, D)$ ,  $E_0$  must be an equilibrium point (see the discussion after Theorem 3.6). Therefore, by Theorem 3.6, as the case  $\operatorname{Re}(u(\varepsilon)v(\varepsilon)^*) = 0$  is ruled out by the assumptions, we have the identity

$$E(\varepsilon) = \frac{\operatorname{Re}(u(\varepsilon)v(\varepsilon)^*)}{\|\operatorname{Re}(u(\varepsilon)v(\varepsilon)^*)\|_{\mathbb{F}}}.$$

Plugging this identity into (48) and using the fact that  $\langle E'(\varepsilon), E(\varepsilon) \rangle = \frac{1}{2} \frac{d\|E(\varepsilon)\|_{\mathbb{F}}^2}{d\varepsilon} = 0$ , we conclude that (47) holds. □

We now relate  $g'_{UV}(\varepsilon)$  to  $g'(\varepsilon)$ .

**THEOREM 5.3.** *Using the notation established above, suppose the assumptions of the two previous theorems apply for the same open interval  $O$  and that for some specific  $\varepsilon \in O$ ,*

$$(49) \quad UV^{\top} = E(\varepsilon) = \frac{\operatorname{Re}(u(\varepsilon)v(\varepsilon)^*)}{\|\operatorname{Re}(u(\varepsilon)v(\varepsilon)^*)\|_{\mathbb{F}}},$$

so that the matrices  $M_{UV}(\varepsilon)$  and  $M(\varepsilon E(\varepsilon))$  are the same and the eigentriples  $(\lambda_{UV}(\varepsilon), x_{UV}(\varepsilon), y_{UV}(\varepsilon))$  and  $(\lambda(\varepsilon), x(\varepsilon), y(\varepsilon))$  coincide, with  $g_{UV}(\varepsilon) = g(\varepsilon)$ . Then

$$g'_{UV}(\varepsilon) = g'(\varepsilon).$$

*Proof.* Using (45), (44), and (49), we have

$$g'_{UV}(\varepsilon) = \frac{\operatorname{Re}(y(\varepsilon)^* (B(I - \varepsilon E(\varepsilon)^\top D)^{-1} E(\varepsilon)(I - \varepsilon DE(\varepsilon))^{-1} C) x(\varepsilon))}{y(\varepsilon)^* x(\varepsilon)}.$$

So, using (46) and (47), we obtain

$$g'_{UV}(\varepsilon) = \frac{\operatorname{Re}(u(\varepsilon)^* E(\varepsilon)v(\varepsilon))}{y(\varepsilon)^* x(\varepsilon)} = \frac{\|\operatorname{Re}(u(\varepsilon)v(\varepsilon)^*)\|_F}{y(\varepsilon)^* x(\varepsilon)} = g'(\varepsilon). \quad \square$$

This result is important, because at the start of the contraction phase of Algorithm HEC-RF, assuming that the expansion phase has returned a locally rightmost point of  $\sigma_{\varepsilon}^{\mathbb{R}, \|\cdot\|_F}(A, B, C, D)$ , we have that (49) holds. Hence, the first Newton step of the contraction phase, namely a Newton step for finding a zero of  $g_{UV}$ , is *equivalent* to a Newton step for finding a zero of  $g$ , which is the ultimate goal. For this reason, under a suitable regularity condition, Algorithm HEC-RF is actually quadratically convergent. We omit the details here, but a convergence rate analysis similar to that given in [MO16, Theorem 4.4] for the complex stability radius hybrid expansion-contraction algorithm holds for Algorithm HEC-RF too.

**6. Discrete-time systems.** We now briefly summarize the changes to our results and algorithms that are needed to handle, instead of (1)–(2), the discrete-time system

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

where  $k = 1, 2, \dots$ . The definitions of the transfer matrix function, spectral value sets, and real- $\mu$  functions in section 2.1 remain unchanged. In section 2.2, the stability region is the open unit disk  $\mathbb{D}_-$  instead of the open left half-plane  $\mathbb{C}_-$ , and the definition of the stability radius must be adjusted accordingly. In section 2.3, instead of the spectral abscissa  $\alpha$  and spectral value set abscissa  $\alpha_\varepsilon$ , we require the spectral radius  $\rho$  and spectral value set radius  $\rho_\varepsilon^{\mathbb{K}, \|\cdot\|}$ , which are defined by maximization of  $|\lambda|$  instead of  $\operatorname{Re} \lambda$  over the spectral value set.<sup>2</sup> Now, instead of “rightmost” points, we search for “outermost” points.

In section 3, it is convenient to extend Definition 3.2 as follows:  $(\lambda, x, y)$  is an  $\operatorname{RP}(z)$ -compatible eigentriple of  $M$  if  $\lambda$  is a simple eigenvalue of  $M$ ,  $x$  and  $y$  are corresponding normalized right and left eigenvectors, and  $y^*x$  is a real positive multiple of  $z$ . Then, instead of (16), we have, taking  $(\lambda(t), x(t), y(t))$  to be an  $\operatorname{RP}(\bar{\lambda}(t))$ -compatible eigentriple,

$$\frac{d}{dt} |\lambda(t)| = \frac{\operatorname{Re}(\bar{\lambda}(t)\dot{\lambda}(t))}{|\lambda(t)|} = \frac{1}{|y(t)^*x(t)|} \operatorname{Re}(u(t)^* \dot{\Delta}(t)v(t)).$$

The ODE (22) then remains unchanged, except that the eigentriple  $(\lambda(t), x(t), y(t))$  is an *outermost*  $\operatorname{RP}(\bar{\lambda}(t))$ -compatible eigentriple of  $M(\varepsilon E(t))$  instead of a rightmost  $\operatorname{RP}$ -compatible eigentriple. Theorem 3.6 also holds as before, with the same change. In section 4, we replace Algorithm SVSA-RF by Algorithm SVSR-RF (spectral value set

<sup>2</sup>Recall again the completely different usage of “radius” in these names; the stability radius refers to the data space and the spectral radius to the complex plane.

radius: real Frobenius-norm), whose purpose is to approximate  $\rho_\varepsilon^{\mathbb{R}, \|\cdot\|_F}(A, B, C, D)$ . The only change that is needed is to replace rightmost RP-compatible eigentriple  $(\lambda_k, x_k, y_k)$  by outermost RP( $\bar{\lambda}_k$ )-compatible eigentriple  $(\lambda_k, x_k, y_k)$ . To ensure that  $|\lambda_{k+1}| \geq |\lambda_k|$ , a line search can again be used, as explained in [Mit14, sec. 6.3.3]. The algorithm produces  $\lambda_k$ , certifying that  $|\lambda_k| \leq \rho_\varepsilon^{\mathbb{R}, \|\cdot\|_F}(A, B, C, D)$ .

In section 5, since the stability region is now the open unit disk  $\mathbb{D}_-$ , instead of (41) we have  $g(\varepsilon) = \rho_\varepsilon^{\mathbb{R}, \|\cdot\|_F}(A, B, C, D) - 1$ , and instead of (42) we have  $g_{UV}(\varepsilon) := \rho(M_{UV}(\varepsilon)) - 1$ . The derivatives of  $g_{UV}$  in (45) and  $g$  in (47) remain unchanged except for the RP-compatibility change and the replacement of  $y^*x$  by  $|y^*x|$  in both denominators. In addition to this alternate RP-compatibility definition, Algorithm HEC-RF is changed as follows: rightmost, right half-plane, and imaginary axis are changed to outermost,  $\mathbb{C} \setminus \mathbb{D}_-$ , and unit circle, respectively.

**7. Implementation and experiments.** We implemented Algorithm HEC-RF by extending the open-source MATLAB code `getStabRadBound` [MO16, sec. 7], which is the implementation of the original HEC algorithm for approximating the complex stability radius. Our new code supports approximating both the complex and real stability radii, for both continuous-time and discrete-time systems, although for brevity, we continue to refer primarily only to the continuous-time case. We similarly adapted the related fast upper bound algorithm [MO16, sec. 4.4], which aims to quickly find a destabilizing perturbation necessary for initializing Algorithm HEC-RF. This “greedy” strategy aims to take steps as large as possible toward a destabilizing perturbation by alternating between increasing  $\varepsilon$  and taking a single SVSA-RF update step of the perturbation matrices  $U$  and  $V$ . In the course of this work, we also significantly improved the convergence criteria of `getStabRadBound` and made several improvements to help accelerate the algorithm. All experiments were performed using MATLAB R2015a running on a Macbook Pro with an Intel i7-5557U dual-core CPU and 16GB of RAM, running Mac OS X v10.11.5.

### 7.1. Algorithm HEC-RF and its subroutine SVSA-RF in practice.

**7.1.1. Convergence criteria and tolerances.** In theory, Algorithm HEC-RF has converged once it has found a perturbation  $\varepsilon UV^T$  such that  $\lambda \in \sigma(M(\varepsilon UV^T))$  is a locally rightmost point of  $\sigma_\varepsilon^{\mathbb{R}, \|\cdot\|_F}(A, B, C, D)$  and  $\text{Re } \lambda = 0$ . However, in practice it is not so straightforward, and we have found that the original convergence criteria outlined in [GGO13] and [MO16], respectively, for Algorithms SVSA and HEC for complex spectral value sets, can sometimes be inadequate. While we will continue to refer to HEC-RF and SVSA-RF, the following discussion of improved convergence criteria also applies directly to the complex spectral value set versions of these algorithms. Describing the convergence criteria in full is a somewhat lengthy and technical discussion, so we instead present a higher-level overview here and then use Appendix A to further discuss these crucial matters for implementing a practicable and reliable version of the HEC-RF algorithm, along with pseudocode.

In order to ensure that the iterates of HEC-RF always remain in the closed right half-plane (a necessary condition for HEC-RF’s convergence theory), the Newton-bisection-based contraction phase actually solves a slightly shifted version of the root problem given by (42), that is, so it converges to a point on the vertical line  $x = 0.5\tau_\varepsilon$ , instead of on the imaginary axis. We consider the contraction phase converged if it finds a point  $\lambda_c$  such that  $\text{Re } \lambda_c \in [0, \tau_\varepsilon]$ . The shift allows acceptable iterates to be obtained when the contraction phase converges from either the right or left side; for the unshifted problem, convergence from the left would not yield an acceptable

iterate. For more details on this shift, see the discussion of  $\tau_\varepsilon$  in [MO16, sec. 7.1].

Detecting whether or not the subroutine SVSA-RF has indeed converged to a locally rightmost point is less straightforward, as even in the case of the complex spectral value sets, computing the first-order necessary conditions for locally rightmost points is too expensive (under the assumption that linear solves involving matrix  $A$  are prohibitively costly). Instead, the implementation of SVSA-RF terminates when it detects that its progress has essentially come to a halt, specifically when the following condition holds:

$$(50) \quad \frac{|\lambda_{k+1} - \lambda_k|}{|\lambda_k|} \leq \tau_{uv},$$

where  $\lambda_k \in \mathbb{C}$  in this context is an iterate of SVSA-RF and  $\tau_{uv} \in \mathbb{R}^+$  is the expansion phase tolerance. Note that it is important that the rate of progress be measured both as a relative change and as a distance in the complex plane; for more details, see Appendix A. Of course, if either  $\lambda_{k+1}$  or  $\lambda_k$  is zero, our proposed measure is not meaningful. In this case, we simply skip the convergence check, noting that it can, at most, only cause two additional steps to be taken. Furthermore, since in the context of HEC-RF the iterates of SVSA-RF are always in the closed right half-plane and typically never exactly on the imaginary axis, this case almost never occurs. Finally, if matrices  $A$ ,  $B$ ,  $C$ , and  $D$  are all real (which is assumed for the real stability radius but not for the complex stability radius), we must ensure that distance due to complex conjugacy is not measured, which is done by flipping signs of the imaginary parts as necessary.

Given that SVSA-RF is initialized at a point  $\lambda_c$  with  $\text{Re } \lambda_c \in [0, \tau_\varepsilon]$ , we then say that Algorithm HEC-RF has converged if subroutine SVSA-RF satisfies its convergence criteria (50) and the total amount of rightward expansion from  $\lambda_c$  is at most  $\tau_\varepsilon$ , that is, Algorithm HEC-RF has converged if it finds a locally rightmost point  $\lambda$  such that  $\text{Re } \lambda \in [0, 2\tau_\varepsilon]$ .

While HEC-RF is generically quadratically convergent, its subroutine SVSA-RF typically only exhibits linear convergence. To mitigate incurring the slow convergence of SVSA-RF, particularly as HEC-RF may call it multiples times, an early contraction strategy was proposed in [MO16, sec. 4.3] so that initial expansion phases will generally terminate early, before locally rightmost points have been attained, since it is only necessary to find a locally rightmost point to high accuracy on the very last expansion. This behavior is governed by the expansion relative-step-size tolerance  $\tau_{uv}^{\text{rss}} \in [0, 1)$ , which says that SVSA-RF should terminate early if the size of the current step falls below some fraction of the largest step taken so far, that is,

$$(51) \quad |\lambda_{k+1} - \lambda_k| < \tau_{uv}^{\text{rss}} \max_{0 \leq j < k} |\lambda_{j+1} - \lambda_j|,$$

where in this context  $\lambda_k \in \mathbb{C}$  is again an iterate of SVSA-RF. This measure is slightly different from the one used in [MO16], since our improved measure checks distances in the complex plane modulo complex conjugacy. Although enabling early contraction technically reduces HEC-RF's theoretical rate of convergence from quadratic to superlinear, it was found in the complex stability radius case that the number of SVSA iterates incurred was generally reduced so significantly that HEC was usually much faster with early contraction enabled, with respect to wall-clock time.

Similarly, the contraction phase can also optionally terminate early when it has at least made significant progress, as accurate contraction back to a point  $\lambda_c$  such that

$\operatorname{Re} \lambda_c \in [0, \tau_\varepsilon]$  is also only necessary on the very last contraction phase. Specifically, the contraction phase can terminate early once one of its iterates  $\lambda_k$  satisfies

$$(52) \quad \operatorname{Re} \lambda_k \in [0, \tau_\varepsilon^{\operatorname{rss}} \operatorname{Re} \lambda_0],$$

where  $\tau_\varepsilon^{\operatorname{rss}} \in [0, 1)$  and  $\lambda_k \in \mathbb{C}$  in this context is now an iterate of the Newton-bisection contraction phase.

As the contraction and expansion phases can terminate in multiple ways, detecting whether or not HEC-RF has actually converged is somewhat involved, and we leave the remaining discussion of how to properly implement it to Appendix A. For the experiments in this paper, we set  $\tau_\varepsilon = \tau_{uv} = 10^{-12}$ . We tested the code in two configurations, once with early expansion and contraction disabled, and again with them both enabled; in the latter case, we set  $\tau_\varepsilon^{\operatorname{rss}} = \tau_{uv}^{\operatorname{rss}} = 10^{-2}$ .

**7.1.2. Extrapolating the iterates of Algorithm SVSA-RF.** Although enabling early contraction can help to reduce the number of iterates incurred by the linearly convergent SVSA-RF when used as a subroutine of HEC-RF, the number of SVSA-RF iterates may still sometimes be large, and, of course, early contraction is not relevant if one wishes to approximate the spectral value set abscissa. For complex spectral value sets, an implicit extrapolation procedure was proposed in [MO16] and [Mit14, sec. 4.2] to accelerate the rank-one perturbation matrix sequence  $\{u_k v_k^*\}$  of Algorithm SVSA, where  $u_k \in \mathbb{C}^p$  and  $v_k \in \mathbb{C}^m$  have unit norm. As extrapolating the entire matrix sequence  $\{u_k v_k^*\}$  directly could be costly, since it has  $mp$  complex-valued entries, the procedure instead applies vector extrapolation to a chosen row and column of  $\{u_k v_k^*\}$ , using, say, the last five iterates, to produce extrapolated row  $r_\star$  and column  $c_\star$ , for a total of only  $m + p$  extrapolated values. Note that it is generally not possible to extrapolate vector sequences  $\{u_k\}$  and  $\{v_k\}$  directly, since even though  $u_k v_k^*$  is unique,  $u_k$  and  $v_k$  are only unique up to a unimodular scalar, and so  $\{u_k\}$  and  $\{v_k\}$  may not actually be converging individually. Finally, after computing extrapolations  $r_\star$  and  $c_\star$ , it is possible to cheaply construct a pair of unit norm vectors  $u_\star$  and  $v_\star$  such that  $u_\star v_\star^*$ , we hope, best recovers row  $r_\star$  and column  $c_\star$ . For more details, see [Mit14, sec. 4.2].

For the case of real spectral value sets, we extended this implicit extrapolation strategy so that it can also handle matrix sequences  $\{U_k V_k^T\}$  where each  $U_k V_k^T$  has rank at most two. As describing this would also be a rather lengthy and technical digression, we refer the reader to [Mit14, sec. 6.3.5] for full details and only briefly sketch the main differences here. To implicitly extrapolate sequences of up-to-rank-two matrices, we perform vector extrapolation four times, on two chosen rows and two chosen columns of the evolution  $\{U_k V_k^T\}^T$ . Having obtained extrapolated rows  $r_1$  and  $r_2$  and extrapolated columns  $c_1$  and  $c_2$ , it is also possible to cheaply construct a new pair  $U_\star \in \mathbb{R}^{p \times 2}$  and  $V_\star \in \mathbb{R}^{m \times 2}$  such that  $\|U_\star V_\star^T\|_F = 1$  and  $U_\star V_\star^T$ , we hope, best recovers the pair of extrapolated rows and the pair of extrapolated columns. If the rightmost eigenvalue of  $M(\varepsilon U_\star V_\star^T)$  does not satisfy monotonicity, that is, it is not to the right of the current iterate, then the extrapolation is rejected. Given extrapolations  $r_1, r_2, c_1,$  and  $c_2$ , we also investigated construction of  $U_\star$  and  $V_\star$  iteratively by specifying it as a constrained optimization problem, where the objective is to best recover the extrapolated rows and columns such that  $\|U_\star V_\star^T\|_F = 1$ . However, this was much more expensive than our direct procedure, and we did not observe any improved extrapolation performance; hence, we only used our direct procedure for `getStabRadBound`. For more details on the optimization-based alternative, we refer the reader to the supplementary material.

**7.1.3. Interpolating iterates of Algorithm SVSA-RF.** In the case that the full SVSA-RF update step satisfies monotonicity, that is,  $\operatorname{Re} \lambda_{\text{LS}}(1) - \operatorname{Re} \lambda_{\text{LS}}(0) > 0$  holds for the rightmost eigenvalue of (37) at  $t = 0$  and  $t = 1$ , it may still happen that there exists an even better step for some  $\hat{t} \in (0, 1)$ . Since we have the values of  $\operatorname{Re} \lambda_{\text{LS}}(t)$  at  $t = 0$  and  $t = 1$  and  $\dot{\operatorname{Re}} \lambda_{\text{LS}}(0)$ , we can fit an interpolating quadratic polynomial to predict if an intermediary value  $t = \hat{t}$  would yield an even better step than the full  $t = 1$  step. If this quadratic interpolation-derived step is predicted to be at least 1.5 times larger than  $\operatorname{Re} \lambda_{\text{LS}}(1) - \operatorname{Re} \lambda_{\text{LS}}(0)$ , then the rightmost eigenvalue of (37) at  $t = \hat{t}$  is computed, and if  $\operatorname{Re} \lambda_{\text{LS}}(\hat{t}) > \operatorname{Re} \lambda_{\text{LS}}(1)$  does hold, then the interpolated step is taken in lieu of the full step. Note that we do not use cubic interpolation because we would need to first compute  $\operatorname{Re} \dot{\lambda}_{\text{LS}}(1)$ ; as doing so requires an additional eigensolve, the predictions would no longer be negligibly cheap to compute.

**7.1.4. Code optimizations.** We improved the `getStabRadBound` code so that it now only computes left eigenvectors on demand, instead of always computing eigen-triples, as was done in previous versions of the code. If the resulting rightmost eigenvalue of  $M(\Delta)$  for some perturbation  $\Delta$  encountered by the algorithm does not satisfy monotonicity, then computing the corresponding left eigenvector is unnecessary; left eigenvectors need only be computed at points accepted by the algorithm, since it is only at such points that derivatives of eigenvalues are used. This optimization essentially halves the cost of all incurred line searches in both Algorithm SVSA-RF and step 3 of the fast upper bound procedure, while it also halves the cost of computing extrapolation and interpolation steps that end up being rejected.

**7.2. Plotting complex and real spectral value sets.** In order to verify that Algorithm SVSA-RF does converge to locally rightmost points in practice, we attempted to visualize Frobenius-norm bounded real spectral value sets. For complex spectral value sets, it is straightforward to compute the boundary of  $\sigma_{\varepsilon}^{\mathbb{C}, \|\cdot\|_2}(A, B, C, D)$  for a given  $\varepsilon$ ; by Theorem 2.5, one needs to just evaluate  $\|G(s)\|_2^{-1}$  on a grid for a desired range of  $s \in \mathbb{C}$  and use the MATLAB function `contour`. However, in the real Frobenius-norm case, one would have to evaluate  $\mu_{\mathbb{R}}^{\|\cdot\|_F}(G(s))^{-1}$ , but we lack a method for doing that. Instead, our Frobenius-norm bounded real spectral value set visualizations rely on a variety of sampling techniques.

For a given value of  $\varepsilon$ , we may easily produce sample points of  $\sigma_{\varepsilon}^{\mathbb{R}, \|\cdot\|_F}(A, B, C, D)$  by randomly generating a *sample perturbation matrix*  $\Delta_s$  (using `randn()`) such that  $\|\Delta_s\|_F = \varepsilon$  and then plotting the spectrum of  $M(\Delta_s)$  in the complex plane. However, randomly generated perturbations generally only produce eigenvalues in the interior of  $\sigma_{\varepsilon}^{\mathbb{R}, \|\cdot\|_F}(A, B, C, D)$  and rarely on the boundary, let alone near locally rightmost points, even for a large number of samples matrices. As such, we also relied on using perturbation matrices generated via quasi-random Sobol sequences; while Sobol-derived samples did better capture the eigenvalues of  $\sigma_{\varepsilon}^{\mathbb{R}, \|\cdot\|_F}(A, B, C, D)$  away from the interior, the combination of both random and quasi-random sampling was often still not effective for capturing the regions explored by Algorithm SVSA-RF. Thus, we additionally considered sampling the real spectral value sets by considering small perturbations to the matrix sequence  $\{U_k V_k^T\}$  produced by Algorithm SVSA-RF, which we collected for all our experiments. If Algorithm SVSA-RF has indeed converged to a locally rightmost point  $\lambda$  of  $\sigma_{\varepsilon}^{\mathbb{R}, \|\cdot\|_F}(A, B, C, D)$ , then small perturbations to its matrix sequence cannot produce a nearby eigenvalue to the right of  $\lambda$ . Furthermore, if the iterates  $\lambda_k$  of Algorithm SVSA-RF are on or near the boundary of  $\sigma_{\varepsilon}^{\mathbb{R}, \|\cdot\|_F}(A, B, C, D)$ , then small perturbations to the matrices of  $\{U_k V_k^T\}$  will produce

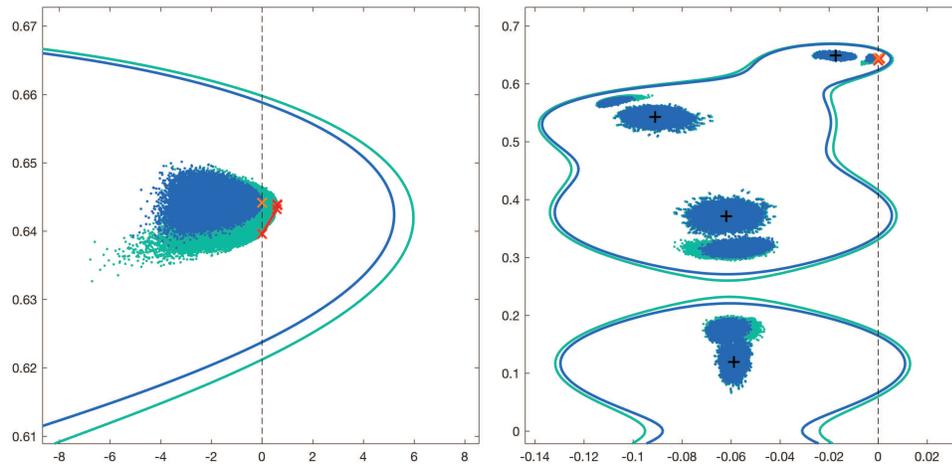


FIG. 1. Test problem ROC3. Selected iterates of Algorithm HEC-RF, namely the first and last expansion phases ( $\varepsilon_1$  and  $\varepsilon_4$ ), are depicted as two sequences of  $x$ 's connected by line segments, respectively in red and orange, in a close-up view (left) and in a wide view (right). The corresponding sets of point clouds for  $\sigma_{\varepsilon}^{\mathbb{R}, \|\cdot\|_{\mathbb{F}}}(A, B, C, D)$ , in green for  $\varepsilon = \varepsilon_1$  and in blue for  $\varepsilon = \varepsilon_4$ , were visualized by the techniques described in section 7.2, with each using 100,000 randomly generated matrices, another 100,000 quasi-random matrices (generated via Sobol sequences), and finally 100,000 randomly perturbed versions of the expansion phases' sequences of matrices. The 200,000 random and quasi-random samples were unable to capture the region near the locally rightmost point found by Algorithm HEC-RF; the points from these samples only appear in the wider view on the right, in small regions about the eigenvalues of  $A$  (represented by the black '+'s). The sample points shown in the close-up view on the left are all from the randomly perturbed versions of the expansion phases' matrix iterates, demonstrating Algorithm HEC-RF's ability to efficiently find extremal rightmost values in real spectral value sets. The solid curves depict the boundaries of the corresponding sets  $\sigma_{\varepsilon}^{\mathbb{C}, \|\cdot\|_2}(A, B, C, D)$ . As can be readily seen, the iterates of Algorithm HEC-RF converged to a locally rightmost point of  $\sigma_{\varepsilon}^{\mathbb{R}, \|\cdot\|_{\mathbb{F}}}(A, B, C, D)$ ,  $\varepsilon = \varepsilon_4$ , close to the imaginary axis (represented by the dashed vertical line) and in the interior of  $\sigma_{\varepsilon}^{\mathbb{C}, \|\cdot\|_2}(A, B, C, D)$ .

eigenvalue samples that highlight the spectral value set boundary in this region.

**7.3. Numerical evaluation of Algorithm HEC-RF.** We tested our new version of `getStabRadBound` on the 34 small-scale and 14 large-scale linear dynamical systems used in the numerical experiments of [GGO13] and [MO16], noting that the system matrices  $(A, B, C, D)$  for these problems are all real-valued. The dimensions of the  $A$  matrices range from 8 to 351 for the small-scale examples and 1006 to 24398 for the large-scale examples; for more details, see [GGO13, Tables 5.1–5.4]. We ran the code in two different configurations for each problem: once in its “pure” HEC form, which we call “v1,” and a second time using an “accelerated” configuration, which we call “v2,” where extrapolation, interpolation, and both early contraction and expansion features were all enabled (all of which are described in section 7.1). The “v2” configuration was set to attempt extrapolation every fifth iteration, from the previous five iterates. For all other parameters, we used `getStabRadBound`'s default user options. That Algorithm SVSA-RF does converge to locally rightmost points (or outermost for SVSR-RF) was supported by examining our real spectral value set visualizations (described in section 7.2) for all 34 problems in our small-scale test set. See Figures 1 and 2 for two such plots.

As Algorithm HEC-RF is, to the best of our knowledge, the only available method to approximate the Frobenius-norm bounded real stability radius, we simply report

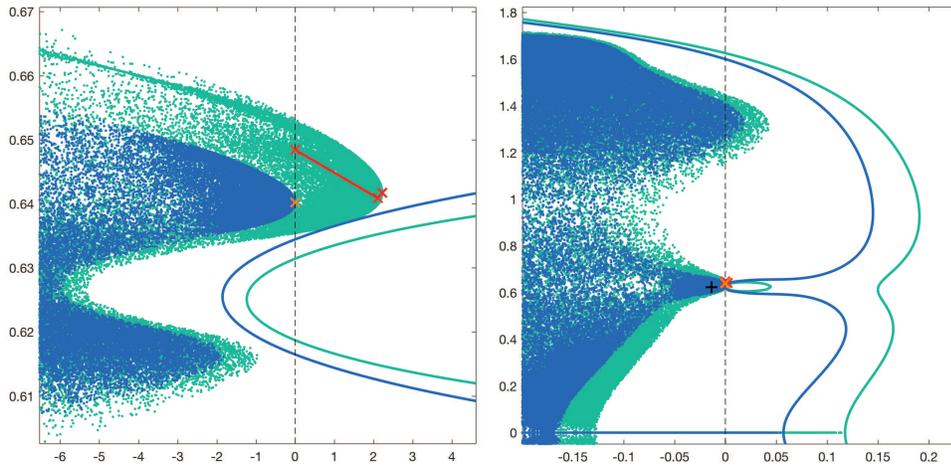


FIG. 2. Test problem ROC9. Selected iterates of Algorithm HEC-RF, namely the first and last expansion phases ( $\varepsilon_1$  and  $\varepsilon_5$ ), are depicted as two sequences of  $x$ 's connected by line segments, respectively in red and orange, in a close-up view (left) and in a wide view (right). The corresponding sets of point clouds for  $\sigma_\varepsilon^{\mathbb{R}, \|\cdot\|_F}(A, B, C, D)$  and set boundaries of  $\sigma_\varepsilon^{\mathbb{C}, \|\cdot\|_2}(A, B, C, D)$ , in green for  $\varepsilon = \varepsilon_1$  and in blue for  $\varepsilon = \varepsilon_5$ , were plotted in a manner similar to that described in Figure 1 and section 7.2. The black + represents an eigenvalue of  $A$ . As can be seen, the iterates of Algorithm HEC-RF converged to a locally rightmost point of  $\sigma_\varepsilon^{\mathbb{R}, \|\cdot\|_F}(A, B, C, D)$ , with  $\varepsilon = \varepsilon_5$ , close to the imaginary axis (represented by the dashed vertical line), though in this case, it is clear that this is not a globally rightmost point. Interestingly, the corresponding sets  $\sigma_\varepsilon^{\mathbb{C}, \|\cdot\|_2}(A, B, C, D)$  have no locally rightmost points near the sequence of locally rightmost points of  $\sigma_\varepsilon^{\mathbb{R}, \|\cdot\|_F}(A, B, C, D)$  found by Algorithm HEC-RF, highlighting the striking difference between complex and real spectral value sets. In fact, for  $\varepsilon = \varepsilon_1$ , it is seen that  $\sigma_\varepsilon^{\mathbb{C}, \|\cdot\|_2}(A, B, C, D)$  actually has a hole below and to the right of the locally rightmost point of  $\sigma_\varepsilon^{\mathbb{R}, \|\cdot\|_F}(A, B, C, D)$  found by Algorithm HEC-RF; the hole is depicted by the small green ellipse in the right plot, a portion of which can be seen in the left plot.

the better of the two upper bounds  $\varepsilon_{v1}$  and  $\varepsilon_{v2}$ , respectively produced by “v1” and “v2” variants of our method, to 12 digits for each test problem, along with their relative difference and statistics on the computational cost, in Tables 1–3. We observe that both variants of the code tend to produce approximations with high agreement, showing that there seems to be little to no numerical penalty for enabling the acceleration features. In fact, on two examples (AC6 and ROC6, both discrete-time systems), we see that the accelerated version of the code actually produced substantially better approximations, with improvement to their respective second-most significant digits. Furthermore, the accelerated “v2” configuration does appear to be effective in reducing the number of eigensolves incurred on most problems, though there are two notable exceptions to this: ROC5 (continuous-time) and ROC6 (discrete-time). It is worth noting that many of the small-scale test problems have such tiny dimensions that a reduction in eigensolves does not always correspond with a speedup in terms of wall-clock time (and can sometimes seemingly paradoxically have increased running times due to the inherent variability in collecting timing data). However, on only moderate-sized problems, such as CBM, CSE2, and CM4 (all continuous-time), we start to see that the running time is dominated by the number of eigensolves. This correspondence is readily apparent in the large and sparse examples in Table 3.

Though it is difficult to tease out the effects of the different acceleration options, since they interact with each other, we were able to determine that the early expan-

TABLE 1

The “Iters” columns show the number of HEC-RF iterations until termination for the “v1” and “v2” configurations of `getStabRadBound`; note that these can be fractional since the method may quit after either a contraction or an expansion phase. The “# Eig” columns show the total number of eigensolves (the sum of the number of right and left eigenvectors computed) incurred while the “Time (secs)” columns show the elapsed wall-clock time in seconds per problem for both code variants. The left column under the “RSR approximation” heading shows the better (smaller) of the two real stability radius approximations  $\varepsilon_{v1}$  and  $\varepsilon_{v2}$ , respectively computed by “v1” and “v2” versions of the code. The rightmost column shows the relative difference between these two approximations, with positive values indicating that the “v2” code produced a better approximation. Relative differences below the  $10^{-12}$  optimality tolerances used for the code are not shown.

Small dense problems: Continuous-time								
Problem	Iters		# Eig		Time (secs)		RSR approximation	
	v1	v2	v1	v2	v1	v2	$\min\{\varepsilon_{v1}, \varepsilon_{v2}\}$	$(\varepsilon_{v1} - \varepsilon_{v2})/\varepsilon_{v1}$
CBM	3	3	122	79	8.646	5.577	$4.46769464697 \times 10^0$	$-8.5 \times 10^{-12}$
CSE2	7	7	223	117	0.643	0.386	$4.91783643704 \times 10^1$	-
CM1	2	3	91	81	0.198	0.203	$1.22474487041 \times 10^0$	-
CM3	3	4	126	108	1.063	0.952	$1.22290355805 \times 10^0$	-
CM4	3	4	222	181	8.181	6.680	$6.30978638860 \times 10^{-1}$	-
HE6	11	9	20852	9972	13.828	8.305	$2.02865555290 \times 10^{-3}$	$+1.5 \times 10^{-10}$
HE7	4	6	492	248	0.406	0.322	$2.88575420548 \times 10^{-3}$	$-3.2 \times 10^{-12}$
ROC1	2	3	93	78	0.127	0.150	$9.11416570667 \times 10^{-1}$	$+2.3 \times 10^{-12}$
ROC2	3	3	98	83	0.136	0.161	$7.49812117968 \times 10^0$	$+1.0 \times 10^{-10}$
ROC3	4	4	204	117	0.211	0.209	$7.68846259016 \times 10^{-5}$	$-3.5 \times 10^{-11}$
ROC4	1	1	40	40	0.084	0.134	$3.47486815789 \times 10^{-3}$	-
ROC5	5.5	11	263	426	0.226	0.390	$1.02041223979 \times 10^2$	$-8.0 \times 10^{-9}$
ROC6	4	4	149	80	0.174	0.182	$3.88148973329 \times 10^{-2}$	-
ROC7	3	3	142	107	0.165	0.163	$8.96564880558 \times 10^{-1}$	-
ROC8	3	4	160	114	0.183	0.194	$2.08497314619 \times 10^{-1}$	$+4.7 \times 10^{-7}$
ROC9	5	8	235	173	0.223	0.297	$4.20965764059 \times 10^{-1}$	-
ROC10	1	1	26	26	0.079	0.096	$1.01878607021 \times 10^1$	-

TABLE 2

For a description of the columns, see the caption of Table 1.

Small dense problems: Discrete-time								
Problem	Iters		# Eig		Time (secs)		RSR approximation	
	v1	v2	v1	v2	v1	v2	$\min\{\varepsilon_{v1}, \varepsilon_{v2}\}$	$(\varepsilon_{v1} - \varepsilon_{v2})/\varepsilon_{v1}$
AC5	3	4	244	197	0.250	0.234	$2.01380141605 \times 10^{-2}$	$+6.3 \times 10^{-12}$
AC12	2	2	35	51	0.098	0.131	$9.33096040564 \times 10^{-2}$	-
AC15	5	6	143	94	0.178	0.179	$4.22159665084 \times 10^{-2}$	-
AC16	4	5	119	78	0.157	0.181	$7.75365184115 \times 10^{-2}$	-
AC17	5	5	222	150	0.201	0.219	$3.35508111043 \times 10^{-6}$	$+4.5 \times 10^{-10}$
REA1	2	2	77	66	0.116	0.167	$1.37498793652 \times 10^{-3}$	-
AC1	4	5	325	230	0.267	0.253	$7.99003318082 \times 10^0$	-
AC2	3	4	61	57	0.111	0.171	$3.36705685350 \times 10^0$	-
AC3	4	4	427	297	0.305	0.329	$7.43718998002 \times 10^{-2}$	-
AC6	5	9.5	253	366	0.215	0.356	$2.32030683553 \times 10^{-8}$	$+2.6 \times 10^{-1}$
AC11	5	3	198	113	0.213	0.167	$5.21908412146 \times 10^{-8}$	$-2.5 \times 10^{-8}$
ROC3	4	5	204	187	0.209	0.264	$5.30806020326 \times 10^{-2}$	-
ROC5	6	5	280	176	0.246	0.306	$2.85628817204 \times 10^{-4}$	$-1.7 \times 10^{-10}$
ROC6	5	7	324	111	0.269	0.239	$5.81391974240 \times 10^{-2}$	$+1.8 \times 10^{-1}$
ROC7	4	4	68	55	0.115	0.161	$9.01354011348 \times 10^{-1}$	-
ROC8	3	6	134	119	0.163	0.217	$2.08192687301 \times 10^{-5}$	$+1.6 \times 10^{-10}$
ROC9	3	4	137	101	0.160	0.177	$4.07812890254 \times 10^{-2}$	-

TABLE 3  
 For a description of the columns, see the caption of Table 1.

Large sparse problems: Continuous-time (top); discrete-time (bottom)								
Problem	Iters		# Eig		Time (secs)		RSR approximation	
	v1	v2	v1	v2	v1	v2	$\min\{\varepsilon_{v1}, \varepsilon_{v2}\}$	$(\varepsilon_{v1} - \varepsilon_{v2})/\varepsilon_{v1}$
NN18	1	2	27	37	1.833	2.430	$9.77424680376 \times 10^{-1}$	-
dwave	2	4	72	59	32.484	28.794	$2.63019715625 \times 10^{-5}$	-
markov	2	3	61	56	12.581	10.479	$1.61146532880 \times 10^{-4}$	-
pde	4	5	128	79	4.670	3.011	$2.71186478815 \times 10^{-3}$	-
rdbrusseletator	2	3	50	45	4.517	4.402	$5.47132014748 \times 10^{-4}$	-
skewlap3d	2	2	103	78	115.004	90.605	$4.59992022215 \times 10^{-3}$	-
sparserandom	2	2	90	74	3.056	2.655	$7.04698184529 \times 10^{-6}$	$-3.5 \times 10^{-10}$
dwave	2	4	34	33	14.776	14.397	$2.56235064981 \times 10^{-5}$	-
markov	3	3	73	64	15.740	14.056	$2.43146945130 \times 10^{-4}$	-
pde	2	2	46	35	1.713	1.450	$2.77295935785 \times 10^{-4}$	-
rdbrusseletator	3	5	76	60	5.926	5.041	$2.56948942080 \times 10^{-4}$	-
skewlap3d	2	3	50	52	53.297	50.526	$3.40623440406 \times 10^{-5}$	-
sparserandom	2	2	21	19	1.035	0.880	$2.53298721605 \times 10^{-7}$	$-3.5 \times 10^{-8}$
tolosa	3	3	98	53	10.097	6.251	$2.14966549184 \times 10^{-7}$	$-6.5 \times 10^{-12}$

sion termination feature was usually the dominant factor in reducing the number of eigensolves. However, extrapolation was crucial for the large gains observed on HE6 (continuous-time) and ROC6 (discrete-time). By comparison, in [MO16], when using HEC to approximate the complex stability radius, extrapolation tended to be much more frequently beneficial while usually providing greater gains as well. Part of this disparity may be because of the greatly increased number of eigensolves we observed when running `getStabRadBound` to approximate the complex stability radius as opposed to the real stability radius; on the 34 small-scale problems, the complex stability radius variant incurred 1226 more eigensolves per problem on average, with the median being 233 more. In our real stability radius experiments, HE6 notwithstanding, Algorithm SVSA-RF simply did not seem to incur slow convergence as often nor to the same severity as its rank-one counterpart for complex spectral value sets, and hence, there was less need for extrapolation. We note that our ODE-based approach for updating real rank-two Frobenius-norm bounded perturbations underlying Algorithm SVSA-RF also provides a new expansion iteration for complex spectral value sets; we evaluate the performance of this new variant when approximating the complex stability radius in section 1 of the supplementary material.

**7.4. Finding an initial upper bound can be challenging.** We noticed that the fast upper bound procedure had some difficulty before it was able to find a destabilizing perturbation for problem CM4 (continuous-time). Generally, we have found that the upper bound procedure can find a destabilizing perturbation within a handful of iterations, but on CM4, it took 23 iterations. In Figure 3, we show plots of  $\sigma_\varepsilon^{\mathbb{R}, \|\cdot\|_F}(A, B, C, D)$  for the largest value of  $\varepsilon$  obtained in the upper bound procedure, along with selected iterates of the routine corresponding to that value of  $\varepsilon$ . As is apparent from the plots, part of the difficulty in finding an upper bound is due to the highly nonconvex “horseshoe” shapes that create locally rightmost points in the left half-plane for values of  $\varepsilon$  near its upper bound  $\|D\|^{-1}$ . The expansion routine had converged to such a point and then iteratively increased  $\varepsilon$  to be near its upper bound in vain. However, and surprisingly, on the 23rd iteration of the upper bound procedure, the expansion phase was actually able to jump out of this region and land in the right half-plane to find a destabilizing perturbation and thus an upper bound. Even

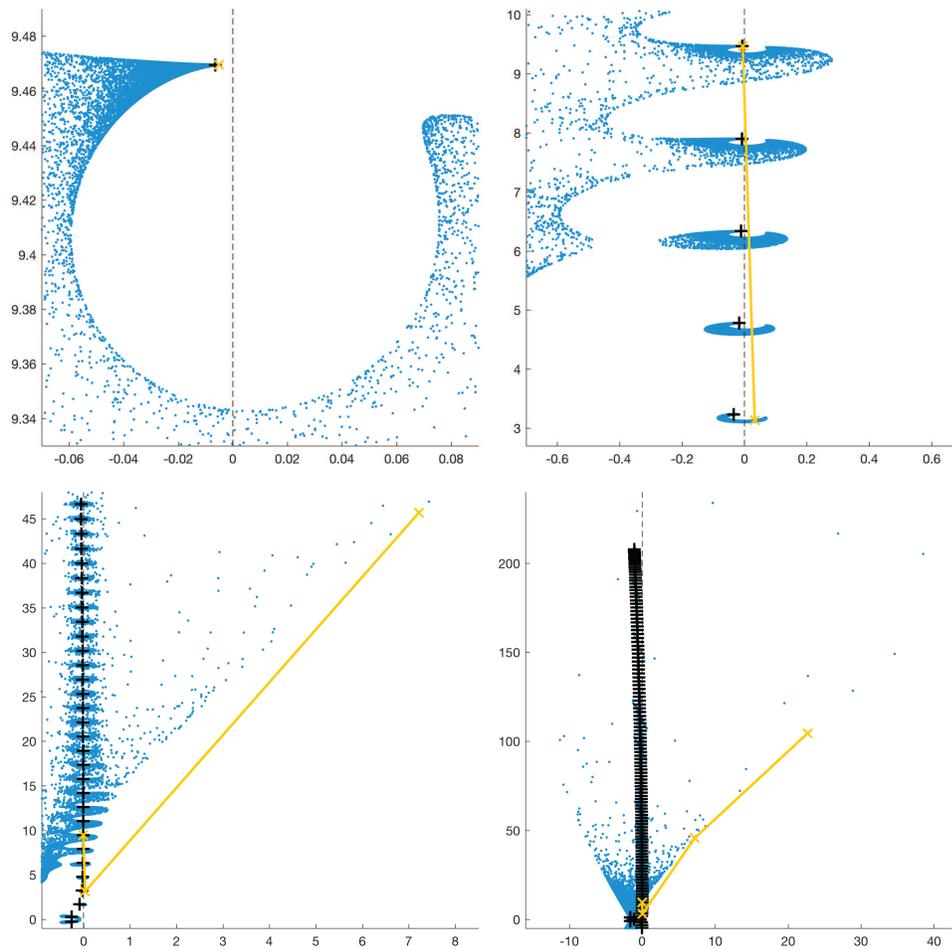


FIG. 3. Test problem CM4. Successively wider views (left-right, top-to-bottom) of  $\sigma_{\varepsilon}^{\mathbb{R}, \|\cdot\|_{\mathbb{F}}}(A, B, C, D)$  (realized by the blue dot samples and generated in a manner similar to that described in Figure 1 and section 7.2) showing selected iterates of the upper bound procedure (yellow  $x$ 's connected by line segments) with  $\varepsilon$  near its limit of  $\|D\|_2^{-1}$ . The black  $+$ 's are eigenvalues of  $A$  while the black dashed line is the imaginary axis. Top left: The expansion phase of the upper bound procedure has nearly converged to a locally rightmost point that is just to the right of an eigenvalue of  $A$ , but this region of  $\sigma_{\varepsilon}^{\mathbb{R}, \|\cdot\|_{\mathbb{F}}}(A, B, C, D)$  is always contained in the left half-plane due to the limit  $\varepsilon < \|D\|_2^{-1}$ . Top right: This highly nonconvex “horseshoe” structure is repeated in multiple places; on the next expansion step, the routine was able to “jump” to a different region of  $\sigma_{\varepsilon}^{\mathbb{R}, \|\cdot\|_{\mathbb{F}}}(A, B, C, D)$  that is in the right half-plane. Bottom left: On the next step after that, the expansion phase again jumps out of one region and into another, this time significantly farther to the right; though technically an upper bound had already been found, the routine continued to iterate to better locate where a minimal destabilizing perturbation may lie. Bottom right: Zooming out further, we see that  $\sigma_{\varepsilon}^{\mathbb{R}, \|\cdot\|_{\mathbb{F}}}(A, B, C, D)$  is much larger than was initially apparent.

though the routine had essentially already converged to a perturbation corresponding to this locally rightmost point in the left half-plane, the routine still produced another update step to try, but this update step was nearly identical to the current perturbation, because further rightward continuous progress was not possible. The full update step failed to satisfy monotonicity, so the line search was invoked with

an initial interpolation of  $t = 0.5$ , and as a result, the unnormalized interpolation of the current perturbation and the full update step perturbation nearly annihilated each other. When that interpolation was renormalized back to have unit Frobenius norm, as is necessary, the resulting perturbation was very different from the current perturbation (as well as the full update step), which thus allowed the algorithm to jump to an entirely different disconnected region of the spectral value set. We note that Algorithm SVSA-RF can also “jump” when a new perturbation just happens to result in a second eigenvalue of  $A$  being taken farther to the right than the eigenvalue it had intended to push rightward. In practice, if the fast upper bound procedure fails, it can always be restarted from a randomly generated perturbation.

**8. Conclusion.** We have presented an algorithm that, to the best of our knowledge, is the first method available to approximate the real stability radius of a linear dynamical system with inputs and outputs defined using Frobenius-norm bounded perturbations. It is efficient even in the large-scale case, and since it generates destabilizing perturbations explicitly, it produces guaranteed upper bounds on the real stability radius. The hybrid expansion-contraction method works by alternating between (a) iterating over a sequence of destabilizing perturbations of fixed norm  $\varepsilon$  to push an eigenvalue of the corresponding perturbed system matrix as far to the right in the complex plane as possible and (b) contracting  $\varepsilon$  to bring the rightmost eigenvalue back to the imaginary axis. The final computed eigenvalue is very close to the imaginary axis and is typically at least a locally rightmost point of the corresponding  $\varepsilon$ -spectral value set. The method is supported by our theoretical results for the underlying ODE that motivates the method, and our computational results are validated by extensive random sampling techniques. The method has been implemented in our open-source MATLAB code `getStabRadBound`.

**Appendix A. Notes on implementing HEC.** We use SVSA and HEC here to refer to both the original SVSA [GGO13] and HEC [MO16] algorithms, as well as their respective real Frobenius-norm variants, SVSA-RF and HEC-RF, since the issues outlined in this section apply to all equally. As mentioned in section 7.1.1, the original convergence criteria for both can sometimes be inadequate. We begin by discussing the limitations of SVSA’s original convergence criteria.

In [GGO13, sec. 5], it was proposed that the expansion phase (SVSA) should be halted once

$$(53) \quad \operatorname{Re} \lambda_{k+1} - \operatorname{Re} \lambda_k < \tau_{uv} \cdot \max(1, \operatorname{Re} \lambda_k)$$

is satisfied.<sup>3</sup> However, this condition can cause the expansion to halt prematurely, which in turn can cause HEC to return an unnecessarily higher value of  $\varepsilon$ , that is, a worse approximation. Part of this is due to the fact that when the expansion iterates have real part less than one, the condition only measures an absolute difference between consecutive steps, which is often a rather poor optimality measure. Furthermore, by only measuring the difference between the real parts, (53) fails to capture any change in the imaginary parts, which if present, would strongly indicate that a locally rightmost point has not yet been reached. For example, an oscillation in the imaginary part (modulo complex conjugacy) is often a sign that a locally rightmost point has not been attained, even when the change in the real part is small, and this is not an exceptional occurrence. These shortcomings have motivated our new SVSA

<sup>3</sup>Compared to [GGO13], we have dropped the absolute value signs here, since in the context of HEC, all points lie in the right half-plane.

TABLE 4

For both phases, return code 1 is used to indicate that the respective desired convergence has been achieved, while return code 2 indicates that the early termination features were invoked. For the contraction phase, return code 3 indicates that the precision limits of the hardware preclude satisfying the convergence criteria; i.e., it is a hard failure. For the expansion phase, return code 3 can be interpreted as a sign of convergence, as further rightward progress is apparently no longer possible.

Simplified list of expansion/contraction termination possibilities	
Contraction phase:	
0:	maxit reached though some contraction achieved
1:	point with real part in $[0, \tau_\varepsilon)$ attained
2:	significant contraction achieved in right half-plane, halted early
3:	desired point is bracketed by two consecutive floating point numbers (no convergence criteria satisfied)
Expansion phase:	
0:	maxit reached
1:	relative difference, in $\mathbb{C}$ , between $\lambda_{k+1}$ and $\lambda_k$ is less than $\tau_{uv}$
2:	step length $ \lambda_{k+1} - \lambda_k $ has significantly shortened, halted early
3:	line search failed to produce a monotonic step

convergence criteria using a relative difference of distance in the complex plane, as described in section 7.1.1.

Previously, in [MO16, sec. 7.1], HEC was said to have converged if either (a) the expansion phase returned a point  $\lambda$  such that  $\operatorname{Re} \lambda < \tau_\varepsilon + \tau_{uv}$  or (b) the expansion and contraction phases failed consecutively, in either order. However, we have found that this too can be inadequate, even with our improvements to convergence criteria for SVSA and our new condition that says that HEC has converged if it has found a locally rightmost point  $\lambda$  such that  $\operatorname{Re} \lambda \in [0, 2\tau_\varepsilon]$ . The main issue is that, in practice, both the contraction and expansion phases may terminate in a multitude of ways without satisfying either of their respective convergence criteria. They may terminate early, using their respective early termination conditions described in section 7.1.1 to help accelerate HEC, or they may hit their maximum number of allowed iterations. They may also simply fail to make further progress, for a variety of reasons. In Table 4, we give a simplified list of these termination possibilities. We now describe how to interpret the combination of these possibilities that can occur and how their consequences should be handled, and we present concise pseudocode for the resulting practical implementation of HEC.

Although initially counterintuitive, we have redesigned the contraction procedure so that reaching its maximum allowed iteration count will only cause it to actually halt iterating if it has also achieved some amount of contraction, that is, it has encountered at least one point  $\tilde{\lambda}$  such that  $0 \leq \operatorname{Re} \tilde{\lambda} \leq \operatorname{Re} \lambda$ , where  $\lambda$  is the initial point. This seemingly unconventional behavior has the two benefits that (a) the only case when it does not make any progress is when it is impossible to do so (i.e., when it exhausts the machine's precision) and (b) a sufficiently large maximum iteration limit to find a first contraction step no longer needs to be known a priori, which is generally not possible. If the contraction routine has made progress (no matter the termination condition), then additional expansion is always potentially possible, and so the next expansion phase must be attempted. Furthermore, even if the contraction phase failed to make any progress, but the previous expansion did not converge, then the next expansion phase should attempt to *resume* it since further expansion is apparently possible and may enable the subsequent contraction phase to finally make progress. The only case

---

**Pertinent pseudocode for a practical HEC implementation**

---

**Input:**  $\varepsilon_0 > 0$ , rightmost  $\lambda_0 \in \sigma(M(\varepsilon_0 UVV^*))$  such that  $\operatorname{Re} \lambda_0 > 0$  with  $\|UVV^*\| = 1$  for the chosen norm and  $U \in \mathbb{C}^p$ ,  $V \in \mathbb{C}^m$  or  $U \in \mathbb{R}^{p \times 2}$ ,  $V \in \mathbb{R}^{m \times 2}$  for respectively approximating the complex or real stability radius, initial contraction bracket given by  $\varepsilon_{\text{LB}} := 0$  and  $\varepsilon_{\text{UB}} := \varepsilon_0$ , boolean variable `expand_converged` := `false`,  
 ...

**Output:** final value of sequence  $\{\varepsilon_k\}$

```

1: for  $k = 0, 1, 2, \dots$  do
2:    $[\varepsilon_c, \lambda_c, \varepsilon_{\text{LB}}, \varepsilon_{\text{UB}}, \text{ret\_con}] = \text{contract}(\varepsilon_{\text{LB}}, \varepsilon_{\text{UB}}, \lambda_k, \dots)$  // where:
3:   //  $\varepsilon_c$  is the possibly contracted value of  $\varepsilon$  such that  $0 < \varepsilon_c \leq \varepsilon$ 
4:   //  $\lambda_c$  is the possibly contracted eigenvalue such that  $0 \leq \operatorname{Re} \lambda_c \leq \operatorname{Re} \lambda_k$ 
5:   //  $\varepsilon_{\text{LB}}$  and  $\varepsilon_{\text{UB}}$  are possibly updated, to the tightest bracket encountered
6:   // ret_con is the contraction's return code from Table 4
7:   // Check if no contraction was possible (precision of hardware exhausted)
8:   if ret_con == 3 and expand_converged then
9:     if  $\operatorname{Re} \lambda_c \leq 2 \cdot \tau_\varepsilon$  then
10:      return // HEC converged to tolerance
11:     else
12:       return // HEC stagnated
13:     end if
14:   end if
15:    $\varepsilon_{k+1} := \varepsilon_c$ 
16:    $[\lambda_{k+1}, \text{ret\_exp}] = \text{expand}(\varepsilon_{k+1}, \lambda_c, \dots)$  // where:
17:   //  $\operatorname{Re} \lambda_{k+1} \geq \operatorname{Re} \lambda_c$ 
18:   // ret_exp is the expansion's return code from Table 4
19:   expand_converged := (ret_exp == 1 or ret_exp == 3)
20:   if expand_converged and  $\operatorname{Re} \lambda_{k+1} \leq 2 \cdot \tau_\varepsilon$  then
21:     return // HEC converged to tolerance
22:   else if  $\operatorname{Re} \lambda_{k+1} - \operatorname{Re} \lambda_c > 0$  then
23:      $\varepsilon_{\text{LB}} := 0, \varepsilon_{\text{UB}} := \varepsilon_{k+1}$  // Expansion made some progress; do new contraction
24:   else if ret_con == 3 then
25:     return // HEC stagnated
26:   end if // Else contraction will be resumed/restarted from where it last left off
27: end for

```

---

remaining is when the contraction phase failed to make any progress (by reaching the limits of the hardware) *after* having had the preceding expansion phase converge (meaning it would not be able to make further progress if it were to be rerun with the same value of  $\varepsilon$ ). In this situation, HEC can no longer make any progress and must quit. However, even though the contraction phase failed to meet its convergence criteria,  $\operatorname{Re} \lambda_c \in [0, 2\tau_\varepsilon]$  may still hold, so HEC may sometimes terminate successfully in this case. If not, HEC has stagnated, which is likely an indication that tolerances are too tight for the problem in question (or possibly that a subroutine has failed in practice).

For each expansion phase, we consider it to have converged once it can no longer make any meaningful rightward progress. Our new stopping criteria described in section 7.1.1 attempt to capture precisely that, and do so more accurately than the

previous scheme. Furthermore, if the line search fails to produce a monotonic step, then the expansion routine is, by default, unable to make further progress. We have observed that the line search failing is generally a good sign, often implying that a stationary point has already been found. We thus consider the expansion phase to have converged if either our new stopping condition is met or the line search fails. Otherwise, further expansion is potentially possible. After an expansion phase, HEC should first check if the expansion phase converged and whether  $\operatorname{Re} \lambda \in [0, 2\tau_\varepsilon]$  holds, as the two conditions together indicate that HEC has converged and can halt with success. However, if the expansion phase has made progress, then, since it has not converged, HEC should continue by starting a new contraction phase. Otherwise, we know that the expansion phase has not made any progress and is thus considered converged. If the previous contraction phase exhausted the precision of the machine, then the HEC iteration can no longer continue, and it has stagnated before meeting its convergence criteria for tolerances that are likely too tight. The only remaining possibility is that the contraction phase achieved some amount of contraction but did not yet converge. In this last case, the contraction phase should be restarted from its most recent bracket to see if it can make further progress, which might enable a subsequent expansion to succeed.

The above design also causes the respective maximum iteration limits of the expansion and contraction phases to act as additional early termination features within the HEC iteration, without ever compromising the final numerical accuracy.

## 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*, Systems Control Lett., 15 (1990), pp. 1–7.
- [BB01] N.A. BOBYLEV AND A.V. BULATOV, *A bound on the real stability radius of continuous-time linear infinite-dimensional systems*, Comput. Math. Model., 12 (2001), pp. 359–368.
- [BBD01] N.A. BOBYLEV, A.V. BULATOV, AND PH. DIAMOND, *Estimates of the real structured radius of stability of linear dynamic systems*, Autom. Remote Control, 62 (2001), pp. 505–512.
- [Bob99] N.A. BOBYLEV, *An easily computable estimate for the real unstructured  $F$ -stability radius*, Internat. J. Control, 72 (1999), pp. 493–500.
- [BS90] N.A. BRUINSMAN AND M. STEINBUCH, *A fast algorithm to compute the  $H_\infty$ -norm of a transfer function matrix*, Systems Control Lett., 14 (1990), pp. 287–293.
- [BS98] E.K. BOUKAS AND P. SHI, *Stochastic stability and guaranteed cost control of discrete-time uncertain systems with Markovian jumping parameters*, Internat. J. Robust Nonlinear Control, 8 (1998), pp. 1155–1167.
- [BS99] E.K. BOUKAS AND P. SHI,  *$H_\infty$  control for discrete-time linear systems with Frobenius norm-bounded uncertainties*, Automatica J. IFAC, 35 (1999), pp. 1625–1631.
- [BV14] P. BENNER AND M. VOIGT, *A structured pseudospectral method for  $H_\infty$ -norm computation of large-scale descriptor systems*, Math. Control Signals Systems, 26 (2014), pp. 303–338.
- [FS14] M.A. FREITAG AND A. SPENCE, *A new approach for calculating the real stability radius*, BIT, 54 (2014), pp. 381–400.
- [FSVD14] M.A. FREITAG, A. SPENCE, AND P. VAN DOOREN, *Calculating the  $H_\infty$ -norm using the implicit determinant method*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 619–635, <https://doi.org/10.1137/130933228>.
- [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 (2013), pp. 709–737, <https://doi.org/10.1137/120875752>.
- [GL13] N. GUGLIELMI AND C. LUBICH, *Low-rank dynamics for computing extremal points of real pseudospectra*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 40–66, <https://doi.org/10.1137/120862399>.

- [GM15] N. GUGLIELMI AND M. MANETTA, *Approximating real stability radii*, IMA J. Numer. Anal., 35 (2015), pp. 1402–1425.
- [GO11] N. GUGLIELMI AND M.L. OVERTON, *Fast algorithms for the approximation of the pseudo-spectral abscissa and pseudospectral radius of a matrix*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 1166–1192, <https://doi.org/10.1137/100817048>.
- [Gug16] N. GUGLIELMI, *On the method by Rostami for computing the real stability radius of large and sparse matrices*, SIAM J. Sci. Comput., 38 (2016), pp. A1662–A1681, <https://doi.org/10.1137/15M1029709>.
- [GV83] G.H. GOLUB AND C. VAN LOAN, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, 1983.
- [HJ90] R.A. HORN AND C.R. JOHNSON, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1990; corrected reprint of the 1985 original.
- [HK94] D. HINRICHSEN AND B. KELB, *Stability radii and spectral value sets for real matrix perturbations*, in Systems and Networks: Mathematical Theory and Applications, Vol. II, Akademie-Verlag, Berlin, 1994, pp. 217–220.
- [HP90a] D. HINRICHSEN AND A.J. PRITCHARD, *A note on some differences between real and complex stability radii*, Systems Control Lett., 14 (1990), pp. 401–408.
- [HP90b] D. HINRICHSEN AND A.J. PRITCHARD, *Real and complex stability radii: A survey*, in Control of Uncertain Systems (Bremen, 1989), Progr. Systems Control Theory 6, Birkhäuser Boston, Boston, MA, 1990, pp. 119–162.
- [HP05] D. HINRICHSEN AND A.J. PRITCHARD, *Mathematical Systems Theory I: Modelling, State Space Analysis, Stability and Robustness*, Springer, Berlin, Heidelberg, New York, 2005.
- [Kar03] M. KAROW, *Geometry of Spectral Value Sets*, Ph.D. thesis, Universität Bremen, Bremen, Germany, 2003, [http://page.math.tu-berlin.de/~karow/papers/diss\\_karow.pdf](http://page.math.tu-berlin.de/~karow/papers/diss_karow.pdf).
- [KV14] D. KRESSNER AND B. VANDEREYCKEN, *Subspace methods for computing the pseudo-spectral abscissa and the stability radius*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 292–313, <https://doi.org/10.1137/120869432>.
- [LKL96] J.H. LEE, W.H. KWON, AND J.W. LEE, *Quadratic stability and stabilization of linear systems with Frobenius norm-bounded uncertainties*, IEEE Trans. Automat. Control, 41 (1996), pp. 453–456.
- [Mit14] T. MITCHELL, *Robust and Efficient Methods for Approximation and Optimization of Stability Measures*, Ph.D. thesis, New York University, New York, NY, 2014.
- [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 (2016), pp. 985–1014.
- [QBR<sup>+</sup>95] L. QIU, B. BERNHARDSSON, A. RANTZER, E.J. DAVISON, P.M. YOUNG, AND J.C. DOYLE, *A formula for computation of the real stability radius*, Automatica J. IFAC, 31 (1995), pp. 879–890.
- [Ros15] M.W. ROSTAMI, *New algorithms for computing the real structured pseudospectral abscissa and the real stability radius of large and sparse matrices*, SIAM J. Sci. Comput., 37 (2015), pp. S447–S471, <https://doi.org/10.1137/140975413>.
- [SVDT96] J. SREEDHAR, P. VAN DOOREN, AND A.L. TITS, *A fast algorithm to compute the real structured stability radius*, in Stability Theory, Springer, New York, 1996, pp. 219–230.
- [ZGD95] K. ZHOU, K. GLOVER, AND J. DOYLE, *Robust and Optimal Control*, Prentice Hall, Englewood Cliffs, NJ, 1995.