Electronic Transactions on Numerical Analysis.
Volume 58, pp. 402–431, 2023.

ISSN 1068–9613.
DOI: 10.1553/etna_vol58s402

ETNA
Kent State University and
Johann Radon Institute (RICAM)

FAST COMPUTATION OF SEP$_\lambda$ VIA INTERPOLATION-BASED GLOBALITY CERTIFICATES*

TIM MITCHELL†

Abstract. Given two square matrices A and B, we propose a new approach for computing the smallest value $\varepsilon \geq 0$ such that $A+E$ and $A+F$ share an eigenvalue, where $\|E\| = \|F\| = \varepsilon$. In 2006, Gu and Overton proposed the first algorithm for computing this quantity, called $\operatorname{sep}_\lambda(A,B)$ ("sep-lambda"), using ideas inspired from an earlier algorithm of Gu for computing the distance to uncontrollability. However, the algorithm of Gu and Overton is extremely expensive, which limits it to the tiniest of problems, and until now, no other algorithms have been known. Our new algorithm can be orders of magnitude faster and can solve problems where A and B are of moderate size. Moreover, our method consists of many "embarrassingly parallel" computations, and so it can be further accelerated on multi-core hardware. Finally, we also propose the first algorithm to compute an earlier version of sep-lambda where $\|E\| + \|F\| = \varepsilon$.

Key words. sep-lambda, eigenvalue separation, eigenvalue perturbation, pseudospectra, Hamiltonian matrix

AMS subject classifications. 15A18, 15A22, 15A42, 65F15, 65F30

Notation. $\|\cdot\|$ denotes the spectral norm, $\sigma_{\min}(\cdot)$ the smallest singular value, $\Lambda(\cdot)$ the spectrum, $\kappa(\cdot)$ the condition number of a matrix with respect to the spectral norm, $J = \left[\begin{smallmatrix} 0 & I \\ -I & 0 \end{smallmatrix}\right]$; a matrix $A \in \mathbb{C}^{2n\times 2n}$ is Hamiltonian if $(JA)^* = JA$, $\mu(\cdot)$ denotes the Lebesgue measure on $\mathbb{R}$, and $\operatorname{bd}\mathcal{A}$, $\operatorname{int}\mathcal{A}$, and $\operatorname{cl}\mathcal{A}$, respectively, the boundary, interior, and closure of a set $\mathcal{A}$.

1. Introduction.

The quantity $\operatorname{sep}_\lambda(A,B)$ measures how close two square matrices $A \in \mathbb{C}^{m\times m}$ and $B \in \mathbb{C}^{n\times n}$ are to sharing a common eigenvalue in the sense of how much A and B must be perturbed in order to make this so. Varah first introduced $\operatorname{sep}_\lambda(A,B)$ in 1979 in [46], and it was subsequently studied by Demmel in [17, 18, 19], although Demmel used a slightly modified version, partly "because it lets us state slightly sharper results later on" [17, p. 24]. The two definitions are:

$$\operatorname{sep}_\lambda^{\mathrm{V}}(A,B) \coloneqq \min_{\substack{E\in\mathbb{C}^{m\times m}\\ F\in\mathbb{C}^{n\times n}}} \big\{\varepsilon : \Lambda(A+E)\cap\Lambda(B+F)\neq\varnothing,\ \|E\|+\|F\|\leq\varepsilon\big\},$$

$$\operatorname{sep}_\lambda^{\mathrm{D}}(A,B) \coloneqq \min_{\substack{E\in\mathbb{C}^{m\times m}\\ F\in\mathbb{C}^{n\times n}}} \big\{\varepsilon : \Lambda(A+E)\cap\Lambda(B+F)\neq\varnothing, \max(\|E\|,\|F\|)\leq\varepsilon\big\},$$

with $\operatorname{sep}_\lambda^{\mathrm{V}}(A,B)$ denoting Varah's definition and $\operatorname{sep}_\lambda^{\mathrm{D}}(A,B)$ denoting Demmel's. Obviously, they are both zero if A and B share an eigenvalue and are both positive otherwise. When it is not necessary to distinguish between the two variants, we drop the superscript and just write $\operatorname{sep}_\lambda(A,B)$. For convenience, we also assume that $m \leq n$ throughout the paper.

The two $\operatorname{sep}_\lambda(A,B)$-quantities can also be equivalently defined in terms of singular values as well as pseudospectra [24, pp. 348–349], where for some $\varepsilon \geq 0$, the *ε-pseudospectrum* of a matrix A is defined

$$\Lambda_\varepsilon(A) \coloneqq \{z\in\mathbb{C} : z\in\Lambda(A+\Delta),\ \|\Delta\|\leq\varepsilon\}, \tag{1.1a}$$

$$= \{z\in\mathbb{C} : \sigma_{\min}(A-zI)\leq\varepsilon\}. \tag{1.1b}$$

*Received May 25, 2021. Accepted January 23, 2023. Published online on April 28, 2023. Recommended by Michiel Hochstenbach.

†Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, 39106 Magdeburg, Germany. Current address: Department of Computer Science, Queens College / CUNY, 65-30 Kissena Boulevard, Flushing, NY 11367, USA (tim.mitchell@qc.cuny.edu). ORCID: 000-0002-8426-0242.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

The first definition of pseudospectra dates to at least 1967 in Varah's Ph.D. thesis [45] with his introduction of an *r-approximate eigenvalue*, while in his 1979 paper on $\mathrm{sep}_\lambda^{\mathrm{V}}(A,B)$, Varah used the term ε*-spectrum* for $\Lambda_\varepsilon(A)$. The current definitive reference on pseudospectra and their applications is certainly Trefethen and Embree's well-known book on the topic [44]. The term "pseudospectrum" was actually coined by Trefethen in 1990 [44, Ch. 6], 23 years after Varah's thesis, although it now considered the standard name.

The singular-value-based definitions of $\mathrm{sep}_\lambda(A,B)$ are

$$\mathrm{sep}_\lambda^{\mathrm{V}}(A,B) = \min_{z\in\mathbb{C}} \{\sigma_{\min}(A-zI) + \sigma_{\min}(B-zI)\} =: \min_{z\in\mathbb{C}} f^{\mathrm{V}}(z), \tag{1.2a}$$

$$\mathrm{sep}_\lambda^{\mathrm{D}}(A,B) = \min_{z\in\mathbb{C}} \max\left\{\sigma_{\min}(A-zI),\, \sigma_{\min}(B-zI)\right\} =: \min_{z\in\mathbb{C}} f^{\mathrm{D}}(z). \tag{1.2b}$$

For equivalent pseudospectral-based definitions of $\mathrm{sep}_\lambda(A,B)$, we have

$$\mathrm{sep}_\lambda^{\mathrm{V}}(A,B) = \min_{\varepsilon_1,\varepsilon_2\geq 0} \{\varepsilon_1+\varepsilon_2 : \Lambda_{\varepsilon_1}(A)\cap\Lambda_{\varepsilon_2}(B)\neq\varnothing\}, \tag{1.3a}$$

$$\mathrm{sep}_\lambda^{\mathrm{D}}(A,B) = \min_{\varepsilon\geq 0} \{\varepsilon : \Lambda_{\varepsilon}(A)\cap\Lambda_{\varepsilon}(B)\neq\varnothing\}. \tag{1.3b}$$

If $\varepsilon \geq \mathrm{sep}_\lambda^{\mathrm{D}}(A,B)$ holds, then $\operatorname{int}\Lambda_\varepsilon(A)\cap\operatorname{int}\Lambda_\varepsilon(B) = \varnothing$ is a sufficient condition for $\varepsilon = \mathrm{sep}_\lambda^{\mathrm{D}}(A,B)$. In contrast, while $\operatorname{int}\Lambda_{\varepsilon_1}(A)\cap\operatorname{int}\Lambda_{\varepsilon_2}(B) = \varnothing$ is a necessary condition for $\varepsilon_1+\varepsilon_2 = \mathrm{sep}_\lambda^{\mathrm{V}}(A,B)$ to hold, it is not a sufficient condition. This is because one can continuously adjust ε_1 and ε_2 such that the two pseudospectra always touch but never have interior points in common. For example, suppose that $\mathrm{sep}_\lambda^{\mathrm{V}}(A,B) > 0$, and let $\hat{\varepsilon}_1$ be such that $\Lambda_{\hat{\varepsilon}_1}(A)$ and $\Lambda_0(B)$ only touch, i.e., an eigenvalue of B is in $\operatorname{bd}\Lambda_{\hat{\varepsilon}_1}(A)$ but $\operatorname{int}\Lambda_{\hat{\varepsilon}_1}(A)\cap\Lambda_0(B) = \varnothing$. In the same fashion, let $\hat{\varepsilon}_2$ be such that $\Lambda_0(A)$ and $\Lambda_{\hat{\varepsilon}_2}(B)$ only touch. Then by continuity of pseudospectra, it is clear that the 2D point $(\varepsilon_1,\varepsilon_2)$ can be continuously adjusted between point $(\hat{\varepsilon}_1,0)$ and point $(0,\hat{\varepsilon}_2)$ such that $\operatorname{bd}\Lambda_{\varepsilon_1}(A)\cap\operatorname{bd}\Lambda_{\varepsilon_2}(B)\neq\varnothing$ and $\operatorname{int}\Lambda_{\varepsilon_1}(A)\cap\operatorname{int}\Lambda_{\varepsilon_2}(B) = \varnothing$ both always hold.

Varah called $\mathrm{sep}_\lambda^{\mathrm{V}}(A,B)$ the *spectrum separation* in [46, Definition 3.2] due to its pseudospectral underpinnings; in fact, in his definition, he used the form given in (1.3a), not the other two alternatives. His motivation in defining $\mathrm{sep}_\lambda^{\mathrm{V}}(A,B)$ was its connection to the sensitivity of solving the Sylvester equation:

$$AX - XB = C, \tag{1.4}$$

where $X, C \in \mathbb{C}^{m\times n}$ and (1.4) has a unique solution if and only if A and B have no common eigenvalue. As Varah noted [46, p. 216], the sensitivity of a solution to (1.4) is inversely proportional to the *separation of A and B*:

$$\mathrm{sep}(A,B) := \min_{\|X\|_{\mathrm{F}}=1} \|AX-XB\|_{\mathrm{F}} = \sigma_{\min}(I_n\otimes A - B^T\otimes I_m),$$

a quantity which Stewart had earlier introduced for studying invariant subspaces [42, Definition 4.5]. It holds that $0 \leq \mathrm{sep}(A,B) \leq \min_{\lambda\in\Lambda(A),\mu\in\Lambda(B)}|\lambda-\mu|$, and clearly, the lower bound is attained if and only if A and B have an eigenvalue in common, while the upper bound is attained if A and B are both normal. However, Varah stressed that if A or B is nonnormal, then $\mathrm{sep}(A,B)$ can be very close to zero, e.g., to machine precision, even if the eigenvalues of A and B are well separated, and that $\mathrm{sep}(A,B)$ is often orders of magnitude smaller than $\mathrm{sep}_\lambda^{\mathrm{V}}(A,B)$. In 1993, Higham's thorough error analysis for solving (1.4) numerically showed that bounding the error of a computed solution in terms of $\mathrm{sep}(A,B)^{-1}$ can sometimes "severely overestimate the effect of a perturbation on the data when only A and B are perturbed, because it does not take account of the special structure of the problem" [26, p. 133], while

ETNA
Kent State University and
Johann Radon Institute (RICAM)

simultaneously presenting an alternative error bound that remedies this deficiency. A few years later, Simoncini used $\mathrm{sep}_\lambda^{\mathrm{V}}(A,B)$ and pseudospectra in her analysis of solving (1.4) via a Galerkin method [41].

Meanwhile, Demmel's initial interest in (his version of) $\mathrm{sep}_\lambda(A,B)$ was for problem of computing stable eigendecompositions [17, 18], but in an entirely different context [19], he subsequently used $\mathrm{sep}_\lambda^{\mathrm{D}}(A,B)$ to disprove two conjectures respectively made by himself and Van Loan related to the (then unsolved) problem of computing the distance to instability of a stable matrix. Following in the spirit of using $\mathrm{sep}_\lambda(A,B)$ in the analysis of the stability of invariant subspaces of matrices [17, 18, 46], Karow and Kressner used $\mathrm{sep}_\lambda^{\mathrm{D}}(A,B)$ in 2014 as a tool in deriving improved perturbation bounds [27]. Most recently in 2021, Roy et al. [40] used $\mathrm{sep}_\lambda^{\mathrm{V}}(A,B)$ in connection with approximating pseudospectra of block triangular matrices; in this case, the value of $\mathrm{sep}_\lambda^{\mathrm{V}}(A,B)$ can be used to construct several different outer approximations to pseudospectra of these structured matrices.

In terms of computing $\mathrm{sep}_\lambda(A,B)$, to the best of our knowledge, only a single algorithm has been given so far for $\mathrm{sep}_\lambda^{\mathrm{D}}(A,B)$, due to Gu and Overton in 2006 [24], while no algorithms have appeared to date for $\mathrm{sep}_\lambda^{\mathrm{V}}(A,B)$. Nevertheless, computing $\mathrm{sep}_\lambda^{\mathrm{D}}(A,B)$ can at least approximate $\mathrm{sep}_\lambda^{\mathrm{V}}(A,B)$ to within a factor of two since

$$\frac{1}{2}\mathrm{sep}_\lambda^{\mathrm{V}}(A,B) \le \mathrm{sep}_\lambda^{\mathrm{D}}(A,B) \le \mathrm{sep}_\lambda^{\mathrm{V}}(A,B),$$

which is simply a special case of the relation $\frac{1}{n}\|x\|_1 \le \|x\|_\infty \le \|x\|_1$ for $x \in \mathbb{C}^n$ obtained by respectively identifying $\mathrm{sep}_\lambda^{\mathrm{V}}(A,B)$ and $\mathrm{sep}_\lambda^{\mathrm{D}}(A,B)$ with the 1-norm and ∞-norm of $(\|E\|,\|F\|)^T \in \mathbb{R}^2$.

It is easy to obtain upper bounds for $\mathrm{sep}_\lambda(A,B)$ by simply evaluating f^{V} and/or f^{D} defined in (1.2) at any points $z \in \mathbb{C}$, or better, by applying (nonsmooth) optimization techniques to find local minimizers of them. Due to the $\max$ function in f^{D}, it is typically nonsmooth at minimizers, while f^{V} will be nonsmooth at a minimizer if that minimizer happens to coincide with an eigenvalue of A or B, which as Gu and Overton mentioned, is often the case for $\mathrm{sep}_\lambda^{\mathrm{V}}(A,B)$. Despite the potential nonsmoothness, f^{V} and f^{D} are rather straightforward functions in just two real variables (via $z = x + \mathbf{i}y$), whose function values and gradients (assuming z is a point where they are differentiable) can be obtained via computing $\sigma_{\min}(A - zI)$ and $\sigma_{\min}(B - zI)$ and their corresponding left and right singular vectors. When A and B are large and sparse, it is often still possible to efficiently compute f^{V} and f^{D} and their gradients via sparse methods. Nevertheless, finding local minimizers of (1.2) provides no guarantees for computing $\mathrm{sep}_\lambda(A,B)$, particularly since these problems may have many different local minima and the locally optimal function values associated with these minima may be very different. Moreover, in applications that use distances measures such as $\mathrm{sep}_\lambda(A,B)$, obtaining an upper bound via local optimization is generally much less useful than either computing the actual measure or a lower bound for it. Indeed, in motivating their algorithm for $\mathrm{sep}_\lambda^{\mathrm{D}}(A,B)$, Gu and Overton aptly remarked [24, p. 350]: "the inability to verify global optimality [of minimizers of f^{D}] remains a stumbling block preventing the computation of $\mathrm{sep}_\lambda(A,B)$, or even the assessment of the quality of upper bounds, via optimization" and "in applications, lower bounds for such distance functions are more important than upper bounds, as they provide 'safety margins'."

In this paper, we propose a new and much faster method to compute $\mathrm{sep}_\lambda^{\mathrm{D}}(A,B)$ to arbitrary accuracy, using properties of pseudospectra, local optimization techniques, and a new methodology that we recently introduced in [36] for finding global optimizers of singular value functions in two real variables. This new approach, called *interpolation-based globality certificates*, can be orders of magnitude faster than existing techniques and also

ETNA
Kent State University and
Johann Radon Institute (RICAM)

avoids numerical difficulties inherent in older approaches. A modified version of our new $\mathrm{sep}_\lambda^{\mathrm{D}}(A,B)$ algorithm also produces estimates of $\mathrm{sep}_\lambda^{\mathrm{V}}(A,B)$ with stronger guarantees than those obtained by optimization; specifically, this modified method produces locally optimal upper bounds $\tilde{\varepsilon} = \varepsilon_1 + \varepsilon_2 \geq \mathrm{sep}_\lambda^{\mathrm{V}}(A,B)$ such that $\operatorname{int}\Lambda_{\varepsilon_1}(A) \cap \operatorname{int}\Lambda_{\varepsilon_2}(B) = \varnothing$, which is a necessary condition for $\tilde{\varepsilon} = \mathrm{sep}_\lambda^{\mathrm{V}}(A,B)$ to hold but which optimization alone does not guarantee. Finally, we also propose a separate algorithm that is the first to compute $\mathrm{sep}_\lambda^{\mathrm{V}}(A,B)$.

The paper is organized as follows. In Section 2, we give a brief overview of Gu and Overton's method for $\mathrm{sep}_\lambda^{\mathrm{D}}(A,B)$ [24] and explain its shortcomings. Then, in Section 3, we give a high-level description of our new *optimization-with-restarts* method and an introduction to the ideas underlying interpolation-based globality certificates. As our new globality certificate for $\mathrm{sep}_\lambda^{\mathrm{D}}(A,B)$ is quite different and significantly more complicated than those we devised for computing Kreiss constants and the distance to uncontrollability in [36], we develop the necessary theoretical statements and components over three separate stages in Section 4, Section 5, and Section 6. In Section 7, we describe how to implement our completed algorithm and give its overall work complexity. We then turn to Varah's sep-lambda in Section 8. Numerical experiments are presented in Section 9, with concluding remarks given in Section 10.

2. Gu and Overton's method to compute $\mathrm{sep}_\lambda^{\mathrm{D}}(A,B)$ and its limitations. The algorithm of Gu and Overton for computing $\mathrm{sep}_\lambda^{\mathrm{D}}(A,B)$ is particularly expensive: it requires $\mathcal{O}((m+n)m^3n^3)$ work, e.g., $\mathcal{O}(n^7)$ when $m=n$, which makes it intractable for all but the tiniest of problems. The core of their method is a pair of related tests, each of which is inspired by a novel but expensive 2D level-set test developed earlier by Gu for estimating the distance to uncontrollability [22]. The cost of each test is dominated by solving an associated generalized eigenvalue problem of order $4mn$, which requires $\mathcal{O}(m^3n^3)$ work when using standard dense eigensolvers.[1] Given some $\varepsilon \geq 0$, the first test ([24, Algorithm 1]) checks whether the ε-level sets of $\sigma_{\min}(A-zI)$ and $\sigma_{\min}(B-zI)$ have any points in common. If this is indeed the case, then clearly $\varepsilon \geq \mathrm{sep}_\lambda^{\mathrm{D}}(A,B)$ must hold. However, if there are no level-set points in common, one cannot conclude that $\varepsilon < \mathrm{sep}_\lambda^{\mathrm{D}}(A,B)$ holds. For example, having no shared level-set points may just be a consequence of $\Lambda_\varepsilon(A)$ being a subset of $\operatorname{int}\Lambda_\varepsilon(B)$ or vice versa, in which case, clearly $\varepsilon > \mathrm{sep}_\lambda^{\mathrm{D}}(A,B)$ holds. To get around this difficulty, Gu and Overton devised an initialization procedure ([24, Algorithm 2]), which invokes their second test many times in order to compute an upper bound $\varepsilon_{\mathrm{ub}}$ such that for all $\varepsilon < \varepsilon_{\mathrm{ub}}$, no connected component of $\Lambda_\varepsilon(A)$ can be strictly inside a component of $\Lambda_\varepsilon(B)$ or vice versa.[2] With this possibility excluded, i.e., $\varepsilon < \varepsilon_{\mathrm{ub}}$, the outcome of the first test then does indicate whether or not $\varepsilon < \mathrm{sep}_\lambda^{\mathrm{D}}(A,B)$ holds. Gu and Overton's overall method [24, Algorithm 3] thus first computes $\varepsilon_{\mathrm{ub}}$ via their initialization procedure and then uses their first test to power a bisection iteration that converges to $\mathrm{sep}_\lambda^{\mathrm{D}}(A,B)$. The entire bisection phase of their algorithm remains $\mathcal{O}(m^3n^3)$ work, since the number of bisection steps can be taken as a constant, but the initialization phase to compute the necessary $\varepsilon_{\mathrm{ub}}$ involves invoking the second test for $(m+n)$ different parameter values, i.e., it solves $(m+n)$ different generalized eigenvalue problems of order $4mn$. Hence, the cost of their entire method is dominated by the initialization procedure, and the total asymptotic work complexity is $\mathcal{O}((m+n)m^3n^3)$.

[1] With respect to the usual convention of treating the computation of eigenvalues as an atomic operation with cubic work complexity, which we use throughout this paper.

[2] In [24], Gu and Overton state that this "not strictly inside" property holds for $\varepsilon \leq \varepsilon_{\mathrm{ub}}$, but actually this inequality should be strict. Near the top of [24, p. 354], it is claimed that "$\varepsilon = \sigma_{\min}(A-zI) > \sigma_{\min}(B-zI)$" holds, where $z \in \operatorname{bd}\Lambda_\varepsilon(A)$ and $z \in \operatorname{int}\Lambda_\varepsilon(B)$. However, per [1, p. 31], there can exist a finite number of points $z \in \operatorname{int}\Lambda_\varepsilon(B)$ such that $\sigma_{\min}(B-zI) = \varepsilon$, and so the "not strictly inside" claim may or may not hold when $\varepsilon = \varepsilon_{\mathrm{ub}}$. Fortunately, with inexact arithmetic, there is essentially no practical consequence of this small oversight, while the theory in [24] is corrected merely by replacing $\leq$ with $<$.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

In their concluding remarks [24, p. 358], Gu and Overton noted that the faster divide-and-conquer technique of [23] for computing the distance to controllability could potentially be adapted to $\operatorname{sep}_\lambda^{\mathrm{D}}(A, B)$, writing that "Although there are some inevitable difficulties with the numerical stability of this approach, the complexity drops significantly." Indeed, when $m = n$, adapting this divide-and-conquer approach would bring down the $\mathcal{O}(n^7)$ work complexity of their algorithm for $\operatorname{sep}_\lambda^{\mathrm{D}}(A, B)$ to $\mathcal{O}(n^5)$ on average and $\mathcal{O}(n^6)$ in the worst case. However, this has not been implemented, and in our own experience of adapting this divide-and-conquer technique to other algorithms, we have observed that doing so can indeed come at the cost of significantly worse reliability due to numerical issues; see [35, Section 8].

Even with dense eigensolvers, Gu and Overton's method can be susceptible to numerical difficulties. A primary concern is that the first test (used for bisection) actually requires being able to assert whether or not two matrices have an eigenvalue in common. If eigenvalues can be computed exactly (which is possible in some cases, e.g., a diagonal matrix), then testing whether two matrices share an eigenvalue can be done without issues. However, in a practical code, computed eigenvalues will have rounding errors, and so one must generally resort to using a tolerance in order to carry out this test. But this also means that it is possible for the test to incorrectly assert that two eigenvalues are the same when they should only be considered close or vice versa. This is critical because the *binary decision* of bisection hinges upon the outcome of this numerical test. Making the wrong choice about the eigenvalues can cause bisection to erroneously update a lower or upper bound, which in turn can result in a significant or even complete loss of accuracy in the computed estimate. The distance-to-uncontrollability methods of [12, 22, 23] also have the same numerical pitfall. In the context of computing Kreiss constants via 2D level-set tests [35], we recently proposed an improved procedure that does not require checking for shared eigenvalues, and as such, it is much more reliable in practice; see [35, Key Remark 6.3]. Our improved technique can also be used to improve the reliability of the aforementioned distance-to-uncontrollability algorithms, but it does not appear to be applicable for Gu and Overton's algorithm for $\operatorname{sep}_\lambda^{\mathrm{D}}(A, B)$. The fundamental difference in the $\operatorname{sep}_\lambda^{\mathrm{D}}(A, B)$ setting is that Gu and Overton's first test is based upon checking whether or not the ε-level sets of two *different* functions, $\sigma_{\min}(A - zI)$ and $\sigma_{\min}(B - zI)$, have any points in common, whereas for the other quantities, pairs of points on a given level set of a *single* function are sought.

Finally, another way to provide some speedup to Gu and Overton's method would be to replace the bisection phase with an *optimization-with-restarts* iteration. In this case, a minimizer $\tilde{z}$ of f^{D} with $\varepsilon = f^{\mathrm{D}}(\tilde{z})$ would be found using some nonsmooth optimization solver, and then, assuming $\varepsilon < \varepsilon_{\mathrm{ub}}$, Gu and Overton's first test ([24, Algorithm 1]) would be used to assert whether or not $\tilde{z}$ is a global minimizer of f^{D}. If so, then $\operatorname{sep}_\lambda^{\mathrm{D}}(A, B) = \varepsilon$, and the computation is done. Otherwise, recalling that the first test computes the points $z \in \mathbb{C}$ such that $\sigma_{\min}(A - zI) = \sigma_{\min}(B - zI) = \varepsilon$, local optimization can be restarted from these points in order to find a better (lower) minimizer. Any such optimization-with-restarts method must monotonically converge to $\operatorname{sep}_\lambda^{\mathrm{D}}(A, B)$ within a finite number of restarts because f^{D} only has a finite number of locally minimal function values, due to f^{D} being semialgebraic. However, there are some issues with this modification. The main limitation is that it neither accelerates nor removes the need for the initialization procedure for obtaining $\varepsilon_{\mathrm{ub}}$, which needs $\mathcal{O}(n^7)$ work, while the subsequent convergent phase of either bisection or optimization-with-restarts involves $\mathcal{O}(n^6)$ work. Consequently, any speedups will be both quite small and limited to the smallest values of n, while being essentially nonexistent for larger n. Another problem is that the theory for nonsmooth optimization typically requires that solvers are initialized at points where the function is differentiable (see, e.g., [10, 16, 28]), but by its nature, the points computed by Gu and Overton's first test are all points where f^{D} will almost certainly

ETNA
Kent State University and
Johann Radon Institute (RICAM)

be nonsmooth. Hence, there may be issues in restarting optimization via these points, and depending on the exact solver and problem, we have observed that solvers can indeed stagnate at these initial points. It is not entirely clear how to best overcome this latter issue, but for us, it is not a priority. Instead, the focus of this paper is to propose an entirely different approach to computing $\operatorname{sep}_\lambda^{\mathrm{D}}(A, B)$ that allows us to use optimization-with-restarts without any of the aforementioned drawbacks of extremely high costs, expensive initialization procedures, and various numerical and technical issues.

3. A high-level overview of our new $\operatorname{sep}_\lambda^{\mathrm{D}}(A, B)$ algorithm. To find a global minimizer of f^{D}, a global optimization problem in two real variables, we will instead develop our optimization-with-restarts algorithm using *interpolation-based globality certificates* [36]. The core task in developing such a method is to devise a generally continuous function (i.e., it may have some jumps) in one real variable that, given an estimate *greater* than the globally minimal value, has an identifiable subset of its domain with *positive measure* that provides a guaranteed way of locating new starting points for another round of optimization. When an estimate is globally minimal, this function should alternatively assert this fact somehow, e.g., by determining that the aforementioned subset is either empty or has measure zero. By sufficiently well approximating this function globally via a piecewise polynomial interpolant (this interpolant may also have jumps), e.g., by using Chebfun[3] [20], it is then possible to quickly check for the existence of the aforementioned positive measure subset, whose presence indicates that the estimate is not globally optimal. In fact, Chebfun can efficiently compute the precise set of intervals corresponding to this subset. When the estimate is too large, the property that there exists a subset of positive measure associated with new starting points is crucial for two reasons. First, it means that encountering this subset during the interpolation process is not a probability zero event, and so if the function is well approximated, this subset will be detected. Second, optimization can be immediately restarted once any points in this subset are discovered, and so high-fidelity interpolants will often not be needed. As a result, restarts tend to be very inexpensive, while high-fidelity approximation is generally only needed for the final interpolant, which asserts that global convergence has indeed been obtained. Moreover, in practice only a handful of restarts are typically needed. Besides overall efficiency, interpolation-based globality certificates are inherently amenable to additional acceleration via parallel processing (see [36, Section 5.2]), while also being quite numerically robust compared to other techniques. There are several reasons for this latter property, but one is that by the nature of interpolation, global convergence is assessed as the result over many computations, whereas other approaches often rely upon a single computation that may result in an erroneous conclusion due to rounding errors; for more details, see [36, Sections 1.3 and 2.3].

Given some estimate $\varepsilon \geq \operatorname{sep}_\lambda^{\mathrm{D}}(A, B)$, in the next sections, we consider the problem of what function $d_\varepsilon : \mathbb{R} \to \mathbb{R}$ to devise for our globality certificate for either finding new points for restarting optimization or asserting whether $\varepsilon = \operatorname{sep}_\lambda^{\mathrm{D}}(A, B)$ holds. The function d_ε should be reasonably well behaved and relatively cheap to evaluate, as otherwise approximating it could be prohibitively expensive and/or difficult. But as mentioned above, Chebfun can efficiently handle nonsmooth and discontinuous functions; [39, pp. 905–906] describes the algorithm that Chebfun uses to efficiently detect discontinuities, either jumps in the function values or derivatives, which allows Chebfun to work around these difficult points during its approximation process. Consequently, we do not have to limit ourselves to smooth continuous candidates for d_ε. The function that we will propose is based on detecting whether or not $\operatorname{int} \Lambda_\varepsilon(A) \cap \operatorname{int} \Lambda_\varepsilon(B)$ is empty and asserts that $\varepsilon > \operatorname{sep}_\lambda^{\mathrm{D}}(A, B)$ if

[3] Available at https://www.chebfun.org.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

and only if $\min_{\theta\in(-\pi,\pi]} d_\varepsilon(\theta) < 0$ holds. Moreover, our certificate for detecting whether $\operatorname{int}\Lambda_\varepsilon(A) \cap \operatorname{int}\Lambda_\varepsilon(B) = \varnothing$ holds works for any value $\varepsilon > \operatorname{sep}^{\mathrm{D}}_\lambda(A,B)$. As we will explain, our d_ε-based globality certificate incurs $\mathcal{O}(kn^3)$ work (recall that we assume $m \le n$), where k is the number of function evaluations required to sufficiently approximate d_ε. Furthermore, our certificate also becomes more efficient the larger ε is, i.e., relatively few function evaluations are needed to approximate d_ε when $\varepsilon \gg \operatorname{sep}^{\mathrm{D}}_\lambda(A,B)$ as compared to when $\varepsilon \approx \operatorname{sep}^{\mathrm{D}}_\lambda(A,B)$.

REMARK 3.1. Although the first test of Gu and Overton ([24, Algorithm 1]) also detects if $\operatorname{int}\Lambda_\varepsilon(A) \cap \operatorname{int}\Lambda_\varepsilon(B) = \varnothing$ holds, note that their test is both more limited in scope and more expensive than our d_ε-based certificate. Gu and Overton's first test (a) requires that $\varepsilon < \varepsilon_{\mathrm{ub}}$ holds in order to use it, with $\varepsilon_{\mathrm{ub}}$ being very expensive to obtain, and (b) does the same amount of work regardless of the value of ε; again, when $m = n$, computing $\varepsilon_{\mathrm{ub}}$ is $\mathcal{O}(n^7)$ work, while [24, Algorithm 1] is $\mathcal{O}(n^6)$ work.

4. Locating pseudospectral components. We now work on defining d_ε and establishing its properties, which is done over three sections. This section follows similarly to [36, Sections 2–4], where we first proposed interpolation-based globality certificates to find level-set components as tools for computing Kreiss constants and the distance to uncontrollability. Here we adapt these ideas to locating pseudospectral components, and throughout this section, we provide specific references to counterparts in [36]. However, as will be seen, computing $\operatorname{sep}^{\mathrm{D}}_\lambda(A,B)$ is more complicated than computing these other quantities, and so the additional tools that we develop in Section 5 and Section 6 will also be needed.

Given a matrix $A \in \mathbb{C}^{m\times m}$, $\varepsilon \ge 0$, and some $z_0 \in \mathbb{C}$ such that ε is not a singular value of $A - z_0 I$, in this section we propose a way of determining which rays emanating from z_0 intersect with $\Lambda_\varepsilon(A)$ and which do not. We define the ray emanating from z_0 specified by the angle $\theta \in \mathbb{R}$ as

$$\mathcal{R}_\theta := \{z_0 + r\mathrm{e}^{\mathbf{i}\theta} \in \mathbb{C} : r \in \mathbb{R},\ r > 0\}. \tag{4.1}$$

As we will explain momentarily, our assumption on ε ensures that a condition needed by our method indeed holds; relatedly, our assumption also ensures that the "search point" z_0 is not on the boundary of $\Lambda_\varepsilon(A)$. Consider the following function parameterized in polar coordinates:

$$\begin{aligned} &f_A(r,\theta) = \sigma_{\min}(A - (z_0 + r\mathrm{e}^{\mathbf{i}\theta})I)) = \sigma_{\min}(F_A(r,\theta)), \quad \text{where} \\ &F_A(r,\theta) = \mathbf{i}\mathrm{e}^{-\mathbf{i}\theta}(A - z_0 I) - \mathbf{i}rI \end{aligned} \tag{4.2}$$

and the second equality above holds since multiplication by a unitary scalar does not alter the singular values of a matrix. Note that $\operatorname{bd}\Lambda_\varepsilon(A)$ is contained in the ε-level set of f_A. The following pair of results give us a way to determine whether or not $\mathcal{R}_\theta$ and $\Lambda_\varepsilon(A)$ intersect, and when they do, to also calculate all the points in $\mathcal{R}_\theta \cap \operatorname{bd}\Lambda_\varepsilon(A)$. The first of these two results is yet another variation of the 1D level-set technique that Byers introduced in order to develop the first method for computing the distance to instability in 1988 [14], a powerful tool which we and many others have adapted, extended, or used to develop 1D and 2D methods to compute various quantities. Applications include the $\mathcal{H}_\infty$- and $\mathcal{L}_\infty$-norms [3, 5, 6, 7, 9], distance to uncontrollability [15, 21, 22, 36], numerical radius [25, 32, 37], pseudospectral (or spectral-value set) abscissa and radius [4, 11, 32], Kreiss constants [35, 36], as well as the optimization of passive systems [30, 31, 38].

LEMMA 4.1 (cf. [36, Theorems 2.1, 3.1, and 4.1]). *Let $A \in \mathbb{C}^{m\times m}$, $\varepsilon \ge 0$, $z_0 \in \mathbb{C}$, and $r, \theta \in \mathbb{R}$. Then ε is a singular value of $F_A(r,\theta)$ defined in (4.2) if and only if $\mathbf{i}r$ is an eigenvalue of the Hamiltonian matrix*

$$C_\theta := \begin{bmatrix} \mathbf{i}\mathrm{e}^{-\mathbf{i}\theta}(A - z_0 I) & -\varepsilon I \\ \varepsilon I & \mathbf{i}\mathrm{e}^{\mathbf{i}\theta}(A - z_0 I)^* \end{bmatrix}. \tag{4.3}$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Proof. Suppose that ε is a singular value of $F_A(r,\theta)$ with left and right singular vectors u and v. Then

$$\varepsilon \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} F_A(r,\theta) & 0 \\ 0 & F_A(r,\theta)^* \end{bmatrix} \begin{bmatrix} v \\ u \end{bmatrix} = \begin{bmatrix} \mathbf{i}e^{-\mathbf{i}\theta}(A - z_0 I) & 0 \\ 0 & -\mathbf{i}e^{\mathbf{i}\theta}(A - z_0 I)^* \end{bmatrix} \begin{bmatrix} v \\ u \end{bmatrix} + \mathbf{i}r \begin{bmatrix} -I & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} v \\ u \end{bmatrix}.$$

Rearranging terms, using the fact that $\left[\begin{smallmatrix} u \\ v \end{smallmatrix}\right] = \left[\begin{smallmatrix} 0 & I \\ I & 0 \end{smallmatrix}\right]\left[\begin{smallmatrix} v \\ u \end{smallmatrix}\right]$, and multiplying the bottom block row by -1, we obtain $C_\theta \left[\begin{smallmatrix} v \\ u \end{smallmatrix}\right] = \mathbf{i}r \left[\begin{smallmatrix} v \\ u \end{smallmatrix}\right]$. ∎

COROLLARY 4.2. *Let $A \in \mathbb{C}^{m\times m}$, $\varepsilon \geq 0$, $z_0 \in \mathbb{C}$, $r, \theta \in \mathbb{R}$, and $\mathcal{R}_\theta$ be the ray defined by (4.1). Then $\mathcal{R}_\theta \cap \Lambda_\varepsilon(A) \neq \varnothing$ if and only if $\mathbf{i}r$ is an eigenvalue of C_θ with $r > 0$.*

Proof. Suppose that $\mathcal{R}_\theta$ and $\Lambda_\varepsilon(A)$ intersect. As $\Lambda_\varepsilon(A)$ is bounded, there exists an $r > 0$ such that the point $z_0 + re^{\mathbf{i}\theta}$ is also on the boundary of $\Lambda_\varepsilon(A)$, and so $\sigma_{\min}(F_A(r,\theta)) = \varepsilon$. Thus by Lemma 4.1, $\mathbf{i}r$ is an eigenvalue of C_θ. Now suppose C_θ has some eigenvalue $\mathbf{i}r$ with $r > 0$. Again by Lemma 4.1, ε must then be a singular value of $F_A(r,\theta)$ but not necessarily the smallest one. Thus, $\sigma_{\min}(F_A(r,\theta)) = \hat{\varepsilon} \leq \varepsilon$, and so it follows that $z_0 + re^{\mathbf{i}\theta}$ is in $\Lambda_{\hat{\varepsilon}}(A) \subseteq \Lambda_\varepsilon(A)$. ∎

For any $z_0 + re^{\mathbf{i}\theta} \in \operatorname{bd}\Lambda_\varepsilon(A)$ with $r > 0$, clearly $\sigma_{\min}(F_A(r,\theta)) = \varepsilon$, and so by Lemma 4.1, $\mathbf{i}r \in \Lambda(C_\theta)$. Hence, via computing all of the imaginary eigenvalues of C_θ, Lemma 4.1 provides a way to calculate all of the points in $\mathcal{R}_\theta \cap \operatorname{bd}\Lambda_\varepsilon(A)$. However, note that if $\mathbf{i}r \in \Lambda(C_\theta)$ with $r > 0$, then $z_0 + re^{\mathbf{i}\theta}$ may or may not be on $\operatorname{bd}\Lambda_\varepsilon(A)$. There are two reasons for this. First, per the proof of Corollary 4.2, ε may not be the smallest singular value of $F_A(r,\theta)$, in which case $z_0 + re^{\mathbf{i}\theta} \in \Lambda_{\hat{\varepsilon}}(A)$ for some $\hat{\varepsilon} < \varepsilon$. Second, there can exist a *finite number* of points $z \in \Lambda_\varepsilon(A)$ such that $z \notin \operatorname{bd}\Lambda_\varepsilon(A)$ but $\sigma_{\min}(F_A(r,\theta)) = \varepsilon$ nevertheless holds; see [1, p. 31].

Corollary 4.2 can be stated more strongly, i.e., in terms of a line intersecting $\Lambda_\varepsilon(A)$, since $\mathbf{i}r_{\text{neg}}$ with $r_{\text{neg}} < 0$ is an eigenvalue of C_θ if and only if $\mathbf{i}|r_{\text{neg}}|$ is an eigenvalue of $C_{\theta+\pi}$. However, for developing the theoretical concepts for our algorithm, it will be more intuitive and simpler to work with the notion of rays emanating from z_0 for the time being. For a code, it does make sense to take advantage of all the imaginary eigenvalues of C_θ, and we describe how this is done, along with other implementation details, in Section 7. Regarding the spectrum of C_θ, also note that since C_θ is Hamiltonian, its eigenvalues are symmetric with respect to the imaginary axis. Eigenvalues of *real* Hamiltonian matrices have additional symmetry with respect to the real axis, but this is generally not the case for the spectrum of C_θ due to C_θ being generically complex valued (even if A is real). Structure-preserving eigensolvers exist, e.g., [2], that preserve this eigenvalue symmetry numerically.

REMARK 4.3. While Lemma 4.1 pertains to the eigenvalues of a single Hamiltonian matrix, the analogous [36, Theorems 2.1, 3.1, and 4.1] used for computing Kreiss constants and the distance to uncontrollability via interpolation-based globality certificates are in terms of the eigenvalues of certain *structured matrix pencils*. For the case of Kreiss constants [36, Theorems 2.1 and 3.1], the associated matrix pencils include parametric matrices that can be singular, and so these generalized eigenvalue problems cannot be reduced to standard eigenvalue problems. However, the matrix pencil for the distance to uncontrollability does permit such a reduction, i.e., [36, Theorem 4.1] can be simplified to be in terms of the eigenvalues of the complex Hamiltonian matrix

$$\begin{bmatrix} \mathbf{i}e^{-\mathbf{i}\theta}A & \widetilde{B} \\ \gamma I & \mathbf{i}e^{\mathbf{i}\theta}A^* \end{bmatrix}, \tag{4.4}$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

where in (4.4), the matrices A and $\widetilde{B}$ and scalars γ and θ are defined in [36, Theorem 4.1].

As we will soon see, we will need to preclude the possibility of zero being an eigenvalue of C_θ. The following straightforward result shows that our assumption on ε not being a singular value of $A - z_0 I$ accomplishes this.

LEMMA 4.4 (cf. [36, Theorems 2.4, 3.3, and 4.4]). *Let $A \in \mathbb{C}^{m\times m}$, $\varepsilon \in \mathbb{R}$, $z_0 \in \mathbb{C}$, and $\theta \in \mathbb{R}$. Then the matrix C_θ defined in (4.3) has zero as an eigenvalue if and only if the matrix $(A - z_0 I)(A - z_0 I)^*$ has ε^2 as an eigenvalue.*

Proof. Since the blocks of C_θ are all square matrices of the same size and the lower two blocks, εI and $\mathbf{i}e^{\mathbf{i}\theta}(A - z_0 I)^*$, commute, we have that

$$\det(C_\theta) = \det(-(A - z_0 I)(A - z_0 I)^* - (-\varepsilon I)(\varepsilon I)) = \det((A - z_0 I)(A - z_0 I)^* - \varepsilon^2 I),$$

thus proving the if-and-only-if equivalence. ∎

We are now ready to present the first major component in our construction of d_ε. Given $\varepsilon \geq 0$ specifying the ε-pseudospectrum of A and $z_0 \in \mathbb{C}$ such that ε is not a singular value of $A - z_0 I$, we define the function $a_\varepsilon : (-\pi, \pi] \to [0, \pi^2]$ and the associated set (cf. [36, Equations (2.4), (3.4), and (4.4)]):

$$a_\varepsilon(\theta) \coloneqq \min\{\operatorname{Arg}(-\mathbf{i}\lambda)^2 : \lambda \in \Lambda(C_\theta), \operatorname{Re}\lambda \leq 0\}, \tag{4.5a}$$

$$\mathcal{A}_\varepsilon \coloneqq \{\theta : a_\varepsilon(\theta) = 0,\ \theta \in (-\pi, \pi]\}, \tag{4.5b}$$

where $\operatorname{Arg} : \mathbb{C} \setminus \{0\} \to (-\pi, \pi]$ is the principal value argument function, the matrix C_θ is defined in (4.3), and the term $\operatorname{Arg}(-\mathbf{i}\lambda)$ in (4.5a) is squared in order to smooth its value out when transitioning to/from zero. We explain this in more detail later on, but the squaring is done in order to make a_ε easier to approximate globally on its domain. Note that the definition of a_ε excludes eigenvalues in the open right half of the complex plane since the spectrum of C_θ is symmetric with respect to the imaginary axis.

THEOREM 4.5 (Properties of a_ε; cf. [36, Theorems 2.7, 3.5, and 4.6]). *Let $A \in \mathbb{C}^{m\times m}$, $\varepsilon \geq 0$, and $z_0 \in \mathbb{C}$ be such that ε is not a singular value of $A - z_0 I$. Then, the function a_ε defined in (4.5a) has the following properties:*

(i) *$a_\varepsilon(\theta) \geq 0$ on its entire domain, i.e., for all $\theta \in (-\pi, \pi]$,*
(ii) *$a_\varepsilon(\theta) = 0 \iff \exists r > 0$ such that $\mathbf{i}r \in \Lambda(C_\theta) \iff \mathcal{R}_\theta \cap \Lambda_\varepsilon(A) \neq \varnothing$,*
(iii) *a_ε is continuous on its entire domain,*
(iv) *a_ε is differentiable at a point θ if the eigenvalue $\lambda \in \Lambda(C_\theta)$ attaining the value of $a_\varepsilon(\theta)$ is unique and simple.*

Furthermore, the following properties hold for the associated set $\mathcal{A}_\varepsilon$ defined in (4.5b):

(v) *$\varepsilon = 0 \iff \mu(\mathcal{A}_\varepsilon) = 0$,*
(vi) *$\varepsilon_1 < \varepsilon_2 \iff \mu(\mathcal{A}_{\varepsilon_1}) < \mu(\mathcal{A}_{\varepsilon_2})$,*
(vii) *if $\varepsilon > f_A(0, \theta)$ for any $\theta \in \mathbb{R}$, then $\mu(\mathcal{A}_\varepsilon) = 2\pi$,*
(viii) *$\mathcal{A}_\varepsilon$ can have up to m connected components.*

Proof. Noting that $-\mathbf{i}\lambda$ in (4.5a) is always in the (closed) upper half of the complex plane, statements (i) and (ii) hold by the definition of a_ε and Corollary 4.2. Statement (iii) follows from the continuity of eigenvalues and our assumption that ε is not a singular value of $A - z_0 I$, equivalently $\varepsilon^2 \notin \Lambda((A - z_0 I)(A - z_0 I)^*)$, and thus, by Lemma 4.4, $0 \notin \Lambda(C_\theta)$ is ensured for any θ. Statement (iv) follows from standard perturbation theory for simple eigenvalues and by the definition of a_ε.

Now turning to $\mathcal{A}_\varepsilon$, either $z_0 \in \operatorname{int}\Lambda_\varepsilon(A)$ or $z_0 \notin \Lambda_\varepsilon(A)$ must hold since our assumption on ε precludes z_0 from being a boundary point. If $\varepsilon > f_A(0, \theta)$, then $z_0 \in \operatorname{int}\Lambda_\varepsilon(A)$, which in turn implies that $\mathcal{R}_\theta \cap \Lambda_\varepsilon(A) \neq \varnothing$ for all θ, thus proving (vii). Now assume $z_0 \notin \Lambda_\varepsilon(A)$. Statement (viii) is a consequence of the well-known fact that for any matrix $A \in \mathbb{C}^{m\times m}$, its

ETNA
Kent State University and
Johann Radon Institute (RICAM)

ε-pseudospectrum has at most m connected components. For any component $\mathcal{G}$ of $\Lambda_\varepsilon(A)$, by connectedness and (ii), it is clear that $\mathcal{G}$ is associated with a single interval $\mathcal{I} \subseteq (-\pi, \pi]$ such that $a_\varepsilon(\theta) = 0$ if and only if $\mathcal{R}_\theta \cap \Lambda_\varepsilon(A) \neq \varnothing$. Since $\mathcal{A}_\varepsilon$ is simply the union of those intervals associated with the components of $\Lambda_\varepsilon(A)$, of which there can be at most m, $\mathcal{A}_\varepsilon$ also has at most m components, thus proving (viii). Statement (vi) follows by noting that $\Lambda_{\varepsilon_1}(A) \subset \Lambda_{\varepsilon_2}(A)$ is equivalent to $\mathcal{A}_{\varepsilon_1} \subset \mathcal{A}_{\varepsilon_2}$. Since $\mathcal{R}_\theta \cap \Lambda_{\varepsilon_1}(A) \neq \varnothing$ implies that $\mathcal{R}_\theta \cap \Lambda_{\varepsilon_2}(A) \neq \varnothing$, it follows that $a_{\varepsilon_1}(\theta) = 0$ implies $a_{\varepsilon_2}(\theta) = 0$, and so $\mathcal{A}_{\varepsilon_1} \subset \mathcal{A}_{\varepsilon_2}$. Now suppose that $\mathcal{A}_{\varepsilon_1} \supset \mathcal{A}_{\varepsilon_2}$, and let $\theta \in \mathcal{A}_{\varepsilon_1} \setminus \mathcal{A}_{\varepsilon_2}$; hence $a_{\varepsilon_1}(\theta) = 0$ but $a_{\varepsilon_2}(\theta) > 0$. Then $\mathcal{R}_\theta$ intersects $\Lambda_{\varepsilon_1}(A)$ but not $\Lambda_{\varepsilon_2}(A)$, and so $\Lambda_{\varepsilon_1}(A) \subset \Lambda_{\varepsilon_2}(A)$ cannot hold—a contradiction. Finally, for (v), if $\varepsilon = 0$, $\Lambda_\varepsilon(A) = \Lambda(A)$, and so $\mathcal{R}_\theta \cap \Lambda_\varepsilon(A) \neq \varnothing$ for at most m different angles. As there can be at most m connected components of $\mathcal{A}_\varepsilon$, if $\mu(\mathcal{A}_\varepsilon) = 0$ holds, then $\varepsilon = 0$. ∎

Per Theorem 4.5, a_ε is a continuous function, and $a_\varepsilon(\theta) = 0$ if and only if it holds that $\mathcal{R}_\theta \cap \Lambda_\varepsilon(A) \neq \varnothing$. *Thus, by finding roots of a_ε, we find rays which intersect the ε-pseudospectrum of A*, our first step toward finding regions where $\Lambda_\varepsilon(A)$ and $\Lambda_\varepsilon(B)$ overlap. For an illustration of this correspondence, see Figure 4.1, where b_ε, the analogue of a_ε for the matrix B, is also plotted.

The properties of a_ε listed in Theorem 4.5 show that it is reasonably well behaved. Satisfying the assumption that ε is not a singular value of $A - z_0 I$ can be trivially met, e.g., just by choosing z_0 with a bit of randomness. The dominant cost of evaluating a_ε at a point θ is computing the spectrum of C_θ, i.e., $\mathcal{O}(m^3)$ work. Relative to Gu and Overton's $\mathrm{sep}^{\mathrm{D}}_\lambda(A, B)$ algorithm, this is a negligible cost. To find the roots of a_ε, we can approximate a_ε using Chebfun, which is why we defined a_ε using the squared term $\mathrm{Arg}(-\mathbf{i}\lambda)^2$ instead of just $\mathrm{Arg}(-\mathbf{i}\lambda)$. As will be made clear in Section 6, a_ε transitioning to/from zero corresponds to two (or possibly more non-generically) eigenvalues of C_θ coalescing on the positive portion of the imaginary axis. Without this squaring, a_ε would generally be non-Lipschitz at such transition points, and thus it could be difficult and/or expensive to approximate via interpolation; the squaring smooths out this high rate of change so that a_ε is easier to approximate. Although the analogues of a_ε and Theorem 4.5 that appeared in [36] for computing Kreiss constants and the distance to uncontrollability were sufficient to develop interpolation-based globality certificates for those quantities, for $\mathrm{sep}^{\mathrm{D}}_\lambda(A, B)$, a_ε and Theorem 4.5 are insufficient.

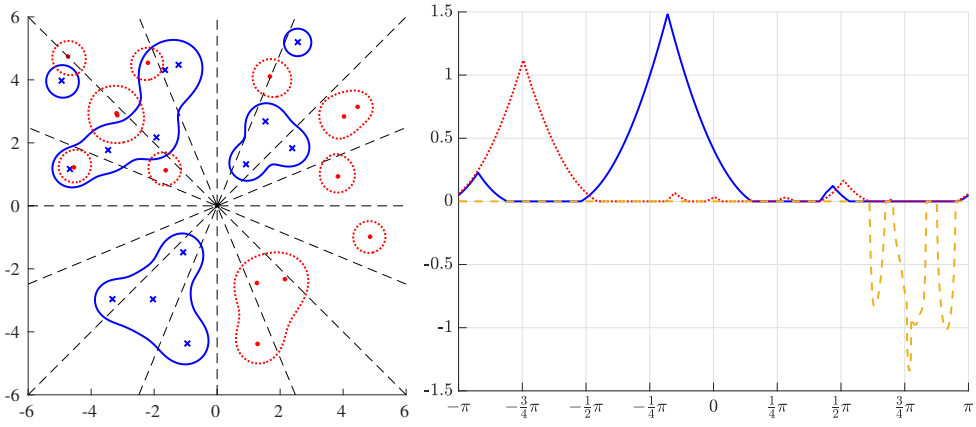
5. Locating pseudospectral overlap.

As part of locating regions where it holds that $\Lambda_\varepsilon(A) \cap \Lambda_\varepsilon(B) \neq \varnothing$, we will also need to locate the components of $\Lambda_\varepsilon(B)$ with respect to the same "search point" z_0 and given value of ε. Thus, for the matrix B, let f_B and F_B respectively denote the analogues of f_A and F_A defined in (4.2), and similarly, let b_ε and $\mathcal{B}_\varepsilon$ be the respective analogues of a_ε and $\mathcal{A}_\varepsilon$ defined in (4.5). For the matrix A, we continue to use C_θ to denote its associated Hamiltonian matrix defined in (4.3), while we use S_θ to denote the analogue Hamiltonian matrix for B, as both matrices will be needed. Per the assumption of Theorem 4.5, we now need to assume that ε is not a singular value of either $A - z_0 I$ or $B - z_0 I$, which again can be easily satisfied by choosing z_0 with some randomness. In establishing tools for locating pseudospectral overlap, we will make use of the following elementary result.

LEMMA 5.1. *Let $\mathcal{A}, \mathcal{B} \subset \mathbb{R}$ be such that $\mathcal{A}$ and $\mathcal{B}$ respectively consist of m and n connected components. Then $\mathcal{A} \cap \mathcal{B}$ can have up to $m + n - 1$ connected components.*

Proof. Let $\mathcal{A} = \mathcal{A}_1 \cup \cdots \cup \mathcal{A}_m$, where each $\mathcal{A}_j$ is a connected component of $\mathcal{A}$ and $\mathcal{A}_j \cap \mathcal{A}_k = \varnothing$ for all $j \neq k$, and in an analogous fashion, let $\mathcal{B} = \mathcal{B}_1 \cup \cdots \cup \mathcal{B}_n$. Without loss of generality, assume that $m \leq n$. If $m = 1$, suppose that the claim is not true, i.e., that $\mathcal{A} \cap \mathcal{B}$ has more than n components. Then there exists at least one pair of numbers x and y that are in different components of $\mathcal{A} \cap \mathcal{B}$ but must be in the same component $\mathcal{B}_j$ of $\mathcal{B}$. However, by connectedness of the components of $\mathcal{A}$ and $\mathcal{B}$, we have that $[x, y] \subset \mathcal{A}_1 = \mathcal{A}$ and $[x, y] \subset \mathcal{B}_j$.

ETNA
Kent State University and
Johann Radon Institute (RICAM)



(a) $\Lambda_\varepsilon(A)$ and $\Lambda_\varepsilon(B)$ (b) $a_\varepsilon(\theta)$, $b_\varepsilon(\theta)$, and $\ell_\varepsilon(\theta)$

FIG. 4.1. *For two randomly generated matrices $A, B \in \mathbb{C}^{14\times 14}$, the left pane shows their eigenvalues (respectively x's and dots) and $\Lambda_\varepsilon(A)$ and $\Lambda_\varepsilon(B)$ (respectively solid and dotted contours) for $\varepsilon = 0.3 > \operatorname{sep}^{\mathrm{D}}_\lambda(A, B)$. The search point z_0 is the origin; rays emanating from it are depicted by dashed lines. The right pane shows corresponding plots of a_ε, b_ε (respectively solid and dotted curves) and ℓ_ε (dashed), where a_ε is defined in (4.5a), b_ε is its analogue for the matrix B, and ℓ_ε is defined in (5.1a). On the left, rays in the lower left quadrant only intersect $\Lambda_\varepsilon(A)$ or neither ε-pseudospectrum, while rays in the lower right quadrant only intersect $\Lambda_\varepsilon(B)$ or neither ε-pseudospectrum. Correspondingly, for $(-\pi, -\frac{1}{2}\pi]$ on the right, we see that a_ε has zeros but b_ε is always positive and vice versa for $(-\frac{1}{2}\pi, 0]$. Meanwhile, there exist rays in the upper right quadrant that pass through both $\Lambda_\varepsilon(A)$ and $\Lambda_\varepsilon(B)$, but $\Lambda_\varepsilon(A)$ and $\Lambda_\varepsilon(B)$ do not overlap in this region; thus, on the right for $(0, \frac{1}{2}\pi]$, we see that a_ε and b_ε do have zeros in common, but ℓ_ε is equal to zero on this interval. Finally, in the upper left quadrant,* $\operatorname{int}\Lambda_\varepsilon(A)$ *and* $\operatorname{int}\Lambda_\varepsilon(B)$ *do in fact overlap, and so on the right, we see that a_ε and b_ε have zeros in common and ℓ_ε is indeed negative on a subset of $(\frac{1}{2}\pi, \pi]$ with positive measure.*

Therefore $[x, y] \subset \mathcal{A} \cap \mathcal{B}$, contradicting that x and y are in different components of $\mathcal{A} \cap \mathcal{B}$. For the inductive step, now assume that the claim holds when $\mathcal{A}$ consists of j components, for $j = 1, \ldots, m-1$ and $j < n$, and suppose that $\mathcal{A}$ has m components. Let $s = \frac{1}{2}(a_{\mathrm{L}} + a_{\mathrm{R}})$, where $a_{\mathrm{L}} = \sup_{a\in\mathcal{A}_{m-1}} a$ and $a_{\mathrm{R}} = \inf_{a\in\mathcal{A}_m} a$, and define $\mathcal{B}_{\mathrm{L}} := \{b \in \mathcal{B} : b < s\}$ and $\mathcal{B}_{\mathrm{R}} := \{b \in \mathcal{B} : b > s\}$. Clearly $\mathcal{B}_{\mathrm{L}}$ and $\mathcal{B}_{\mathrm{R}}$ are disjoint, and $\mathcal{B}_{\mathrm{L}} \cup \mathcal{B}_{\mathrm{R}} = \mathcal{B} \setminus \{s\}$. Letting n_{L} and n_{R} denote the respective number of connected components of $\mathcal{B}_{\mathrm{L}}$ and $\mathcal{B}_{\mathrm{R}}$, it follows that $n_{\mathrm{L}} + n_{\mathrm{R}} = n$ if $s \notin \operatorname{int}\mathcal{B}$ and $n_{\mathrm{L}} + n_{\mathrm{R}} = n + 1$ otherwise. Applying the inductive hypothesis, $\{\mathcal{A}_1 \cup \cdots \cup \mathcal{A}_{m-1}\} \cap \mathcal{B}_{\mathrm{L}}$ has at most $(m-1) + n_{\mathrm{L}} - 1$ connected components, while $\mathcal{A}_m \cap \mathcal{B}_{\mathrm{R}}$ has at most n_{R} connected components. Noting that $\{\mathcal{A}_1 \cup \cdots \cup \mathcal{A}_{m-1}\} \cap \mathcal{B}_{\mathrm{L}}$ and $\mathcal{A}_m \cap \mathcal{B}_{\mathrm{R}}$ are also disjoint and their union is $\mathcal{A} \cap \mathcal{B}$, since $s \notin \mathcal{A} \cap \mathcal{B}$, it follows that $\mathcal{A} \cap \mathcal{B}$ has at most $(m-1) + n_{\mathrm{L}} - 1 + n_{\mathrm{R}} \leq m + n - 1$ connected components. The bound is tight, as one can construct $\mathcal{A}$ such that $\mathcal{A}_j$ intersects both $\mathcal{B}_j$ and $\mathcal{B}_{j+1}$ for $j = 1, \ldots, m-1$, while $\mathcal{A}_m$ intersects $\mathcal{B}_j$ for $j = m-1, \ldots, n$. ∎

THEOREM 5.2 (Properties of $a_\varepsilon + b_\varepsilon$ and a necessary condition for overlap). *Let $A \in \mathbb{C}^{m\times m}$, $B \in \mathbb{C}^{n\times n}$, $\varepsilon \geq 0$, and $z_0 \in \mathbb{C}$ be such that ε is not a singular value of either $A - z_0 I$ or $B - z_0 I$, and let $\mathcal{R}_\theta$ be the ray defined in (4.1). Furthermore, let $\mathcal{Z}_\varepsilon := \{\theta \in (-\pi, \pi] : a_\varepsilon(\theta) + b_\varepsilon(\theta) = 0\}$, where a_ε is defined in (4.5a) for A and b_ε is its analogue for B. Then the following statements hold:*

(i) *if $\mathcal{R}_\theta \cap \Lambda_\varepsilon(A) \cap \Lambda_\varepsilon(B) \neq \varnothing$, then $a_\varepsilon(\theta) + b_\varepsilon(\theta) = 0$,*
(ii) *if $a_\varepsilon(\theta) + b_\varepsilon(\theta) = 0$, then $\mathcal{R}_\theta \cap \Lambda_\varepsilon(A) \neq \varnothing$ and $\mathcal{R}_\theta \cap \Lambda_\varepsilon(B) \neq \varnothing$,*
(iii) *$a_\varepsilon + b_\varepsilon$ is continuous on its entire domain $(-\pi, \pi]$,*
(iv) *$a_\varepsilon + b_\varepsilon$ is differentiable at a point θ if a_ε and b_ε are differentiable at θ,*
(v) *$\mathcal{Z}_\varepsilon$ can have up to $m + n - 1$ connected components.*

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Proof. The assumption in (i) implies that $\mathcal{R}_\theta \cap \Lambda_\varepsilon(A) \neq \varnothing$ and $\mathcal{R}_\theta \cap \Lambda_\varepsilon(B) \neq \varnothing$ hold, and so $a_\varepsilon(\theta) = 0$ and $b_\varepsilon(\theta) = 0$ by Theorem 4.5 (ii). Statements (ii)–(iv) are direct consequences of Theorem 4.5 (ii)–(iv). For (v), note that $\mathcal{Z}_\varepsilon = \mathcal{A}_\varepsilon \cap \mathcal{B}_\varepsilon$, where $\mathcal{A}_\varepsilon$ is defined in (4.5b) for A and $\mathcal{B}_\varepsilon$ is its analogue for B. As $\mathcal{A}_\varepsilon$ and $\mathcal{B}_\varepsilon$ respectively have up to m and n connected components by Theorem 4.5, statement (v) follows from Lemma 5.1. ∎

Given an angle θ, Theorem 5.2 states that $a_\varepsilon(\theta) + b_\varepsilon(\theta) = 0$ is a necessary condition for the pseudospectra $\Lambda_\varepsilon(A)$ and $\Lambda_\varepsilon(B)$ to overlap somewhere along the ray $\mathcal{R}_\theta$, but it is easy to see that this is not a sufficient condition for such overlap. To obtain such a sufficient condition, we now define the function $\ell_\varepsilon : (-\pi, \pi] \to (-\infty, 0]$ and an associated set:

$$\ell_\varepsilon(\theta) \coloneqq -\mu\left(\mathcal{R}_\theta \cap \Lambda_\varepsilon(A) \cap \Lambda_\varepsilon(B)\right), \tag{5.1a}$$

$$\mathcal{L}_\varepsilon \coloneqq \{\theta \in (-\pi, \pi] : \ell_\varepsilon(\theta) < 0\}. \tag{5.1b}$$

As $\mathcal{L}_\varepsilon$ is open, it is measurable, and via Lemma 4.1, we know that the ray $\mathcal{R}_\theta$ can intersect at most $2m$ and $2n$ boundary points, respectively, of $\Lambda_\varepsilon(A)$ and $\Lambda_\varepsilon(B)$. Thus, the number of connected components of $\mathcal{R}_\theta \cap \Lambda_\varepsilon(A)$ is finite, as is the number of connected components of $\mathcal{R}_\theta \cap \Lambda_\varepsilon(B)$; hence, the intersection in the definition of ℓ_ε is measurable. Moreover, Lemma 4.1 allows us to determine these intervals (or isolated points), and so the value of $\ell_\varepsilon(\theta)$ can be computed simply by calculating how much the intervals of $\mathcal{R}_\theta \cap \Lambda_\varepsilon(A)$ overlap those of $\mathcal{R}_\theta \cap \Lambda_\varepsilon(B)$; we explain exactly how this is done in Section 7. In addition to a_ε and b_ε, the function ℓ_ε is also plotted in Figure 4.1.

THEOREM 5.3 (Properties of ℓ_ε and a sufficient condition for overlap). *Let $A \in \mathbb{C}^{m\times m}$, $B \in \mathbb{C}^{n\times n}$, $\varepsilon \geq 0$, $z_0 \in \mathbb{C}$, $\theta \in \mathbb{R}$, and $\mathcal{R}_\theta$ be the ray defined in (4.1). Then for the function ℓ_ε defined in (5.1a), the following statements hold:*

(i) $\ell_\varepsilon(\theta) < 0 \iff \mathcal{R}_\theta \cap \operatorname{int}\Lambda_\varepsilon(A) \cap \operatorname{int}\Lambda_\varepsilon(B) \neq \varnothing$,
(ii) *if* $a_\varepsilon(\theta) + b_\varepsilon(\theta) > 0$, *then* $\ell_\varepsilon(\theta) = 0$,
(iii) ℓ_ε *is continuous on its entire domain* $(-\pi, \pi]$,
(iv) ℓ_ε *is differentiable at a point* θ *if for all* $r > 0$ *such that* $z_0 + re^{\mathbf{i}\theta} \in \operatorname{bd}\Lambda_\varepsilon(A)$, $\mathbf{i}r$ *is a simple eigenvalue of* C_θ, *and for all* $r > 0$ *such that* $z_0 + re^{\mathbf{i}\theta} \in \operatorname{bd}\Lambda_\varepsilon(B)$, $\mathbf{i}r$ *is a simple eigenvalue of* S_θ.

Furthermore, the following statements hold for the associated set $\mathcal{L}_\varepsilon$ defined in (5.1b):

(v) $\varepsilon \leq \mathrm{sep}^{\mathrm{D}}_\lambda(A, B) \iff \mu(\mathcal{L}_\varepsilon) = 0$,
(vi) $\mathrm{sep}^{\mathrm{D}}_\lambda(A, B) < \varepsilon_1 < \varepsilon_2 \iff 0 < \mu(\mathcal{L}_{\varepsilon_1}) < \mu(\mathcal{L}_{\varepsilon_2})$,
(vii) $\min_{\theta\in(-\pi,\pi]} \ell_\varepsilon(\theta) < 0 \iff 0 < \mu(\mathcal{L}_\varepsilon) \iff \mathrm{sep}^{\mathrm{D}}_\lambda(A, B) < \varepsilon$.

Proof. Statement (i) simply follows from the definition of ℓ_ε given in (5.1a) and noting that the intersection $\mathcal{R}_\theta \cap \operatorname{int}\Lambda_\varepsilon(A) \cap \operatorname{int}\Lambda_\varepsilon(B)$ is either empty or consists of a finite number of open intervals in $\mathbb{R}$. For (ii), if $a_\varepsilon(\theta) + b_\varepsilon(\theta) > 0$, then either $\mathcal{R}_\theta \cap \Lambda_\varepsilon(A) = \varnothing$ or $\mathcal{R}_\theta \cap \Lambda_\varepsilon(B) = \varnothing$ holds by Theorem 4.5 (ii), and so $\ell_\varepsilon(\theta) = 0$. Statement (iii) follows from the fact that the boundaries of ε-pseudospectra vary continuously with respect to ε, which is clear from (1.1) and, via Lemma 4.1, do not contain any straight line segments. Under the assumptions in (iv), standard perturbation theory for simple eigenvalues applies.

For $\mathcal{L}_\varepsilon$, (v) is a direct consequence of (i) and the definition of $\mathrm{sep}^{\mathrm{D}}_\lambda(A, B)$ given in (1.3b), as $\operatorname{int}\Lambda_\varepsilon(A) \cap \operatorname{int}\Lambda_\varepsilon(B) = \varnothing$ if and only if $\varepsilon \leq \mathrm{sep}^{\mathrm{D}}_\lambda(A, B)$. Statement (vi) follows by a similar argument to the proof of Theorem 4.5 (vi), with $\mu(\mathcal{L}_{\varepsilon_1}) > 0$ if and only if $\varepsilon_1 > \mathrm{sep}^{\mathrm{D}}_\lambda(A, B)$ following from (i). Statement (vii) is simply a combination of (i) and (vi). ∎

From Theorem 5.3 (vii), it is clear that if ℓ_ε can be sufficiently well approximated, then one can determine whether or not $\varepsilon > \mathrm{sep}^{\mathrm{D}}_\lambda(A, B)$ holds. Moreover, as we fully explain in Section 7, via Lemma 4.1, knowledge of such angles can be used to compute points on the ε-level set of f^{D}, points which can be used to restart optimization to find a better (lower) estimate for $\mathrm{sep}^{\mathrm{D}}_\lambda(A, B)$. Thus, one may wonder what the point was of considering $a_\varepsilon + b_\varepsilon$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

and deriving its associated necessary condition given in Theorem 5.2. There is in fact a very important reason for this.

As ℓ_ε is constant (zero) whenever it is not negative, it can, ironically, be a difficult function to approximate. The pitfall here is that regions where a function appears to be constant may be *undersampled* by interpolation software, precisely because the computed estimate of the error on such regions will generally be exactly zero, e.g., because the software initially builds a constant interpolant for the region in question. Thus, there is a concern that approximating ℓ_ε via interpolation may miss regions where $\ell_\varepsilon(\theta) < 0$ holds, particularly if these regions are small compared to the regions where $\ell_\varepsilon(\theta) = 0$. Our solution to this difficulty is to replace ℓ_ε by another non-constant function whenever $\ell_\varepsilon(\theta) = 0$ holds. We first consider the continuous function $t_\varepsilon : (-\pi, \pi] \to \mathbb{R}$

$$t_\varepsilon(\theta) := \begin{cases} a_\varepsilon(\theta) + b_\varepsilon(\theta) & \text{if } a_\varepsilon(\theta) + b_\varepsilon(\theta) > 0, \\ \ell_\varepsilon(\theta) & \text{otherwise,} \end{cases} \tag{5.2a}$$

$$\mathcal{T}_\varepsilon := \{\theta \in (-\pi, \pi] : t_\varepsilon(\theta) = 0\}, \tag{5.2b}$$

an alternative to approximating ℓ_ε; we have also defined $\mathcal{T}_\varepsilon$, the set of roots of t_ε, as this will be used later. The key point here is that t_ε tells us at which angles the sufficient condition for $\Lambda_\varepsilon(A)$ and $\Lambda_\varepsilon(B)$ to overlap is satisfied ($t_\varepsilon(\theta) < 0$), where only the necessary condition for overlap is satisfied ($t_\varepsilon(\theta) = 0$), or where neither is satisfied ($t_\varepsilon(\theta) > 0$). However, in light of Theorems 5.2 and 5.3, it is clear that t_ε could still contain (potentially large) intervals where it is zero, and generally, regions where $t_\varepsilon(\theta) < 0$ holds will often be found *in between* such regions where t_ε is the constant zero. Thus, there is still cause for concern that approximating t_ε to find regions where it is negative may be difficult. As such, in the next section we introduce an additional nonnegative function to replace the portions of t_ε where it is the constant zero.

REMARK 5.4. Recall that we added smoothing in the definitions of a_ε and b_ε by squaring the $\operatorname{Arg}(\cdot)$-terms, as they otherwise may grow like the square root function when they increase from zero (or vice versa), a behavior which can be difficult and expensive to resolve via interpolation. While ℓ_ε can also exhibit similar non-Lipschitz behavior when it transitions to being negative (and possibly elsewhere when it is already negative), we have intentionally not smoothed this term. The reason is that once an angle θ is found such that $\ell_\varepsilon(\theta) < 0$, there is no need to continue building an interpolant approximation. This angle can immediately be used to compute new level-set points to restart optimization and improve (lower) the current estimate to $\operatorname{sep}^{\mathrm{D}}_\lambda(A, B)$.

6. Locally supporting rays of pseudospectra and our certificate function d_ε. In this section, we propose a new function with which we can replace the constant-zero portions of t_ε. However, we begin with the following general definitions, which are variations of the concept of a *supporting hyperplane* in $\mathbb{R}^n$ [8, Chapter 2.5.2] specialized to $\mathbb{C}$, and a pair of related theoretical results.

DEFINITION 6.1. *Given a connected set $\mathcal{A} \subset \mathbb{C}$, a line $\mathcal{L} \subset \mathbb{C}$* supports *$\mathcal{A}$ at a point $z \in \operatorname{bd}(\mathcal{A}) \cap \mathcal{L}$ if $\mathcal{A}$ lies completely in one of the closed half-planes defined by $\mathcal{L}$.*

DEFINITION 6.2. *Given a set $\mathcal{B} \subset \mathbb{C}$, a line $\mathcal{L} \subset \mathbb{C}$* locally supports *$\mathcal{B}$ at a point $z \in \operatorname{bd}(\mathcal{B}) \cap \mathcal{L}$ if the line $\mathcal{L}$ supports $\mathcal{A} \cap \mathcal{N}$ at z for some neighborhood $\mathcal{N}$ about z, where $\mathcal{A}$ is a connected component of $\mathcal{B}$. A ray $\mathcal{R}$ locally supports $\mathcal{B}$ at $z \in \operatorname{bd}(\mathcal{B}) \cap \operatorname{int} \mathcal{R}$ if the line $\mathcal{L}$ containing $\mathcal{R}$ locally supports $\mathcal{B}$ at z.*

Note that if θ is a point where a_ε transitions from positive to zero (or vice versa), this implies that the ray $\mathcal{R}_\theta$ locally supports $\Lambda_\varepsilon(A)$. Similarly, if θ is a point where b_ε transitions from positive to zero (or vice versa), then $\mathcal{R}_\theta$ locally supports $\Lambda_\varepsilon(B)$. Thus, it follows that if θ is a point where $a_\varepsilon + b_\varepsilon$ transitions from positive to zero (or vice versa), then $\mathcal{R}_\theta$ locally

ETNA
Kent State University and
Johann Radon Institute (RICAM)

supports either $\Lambda_\varepsilon(A)$ or $\Lambda_\varepsilon(B)$ or both simultaneously (though not necessarily at the same point). Also note that if ℓ_ε transitions from zero to negative (or vice versa) at θ, then $\mathcal{R}_\theta$ locally supports $\Lambda_\varepsilon(A) \cap \Lambda_\varepsilon(B)$. We now derive necessary conditions based on the eigenvalues of C_θ and S_θ for these scenarios. We first consider the case when $\mathcal{R}_\theta$ locally supports $\Lambda_\varepsilon(A)$. Note that [11, p. 371–373] also informally touches upon this subject and related issues for the specific case of vertical lines.

LEMMA 6.3. *Let $A \in \mathbb{C}^{m\times m}$, $\varepsilon \geq 0$, $z_0 \in \mathbb{C}$, $\theta \in \mathbb{R}$, and $\mathcal{R}_\theta$ be the ray defined in (4.1). If $\mathcal{R}_\theta$ locally supports $\Lambda_\varepsilon(A)$, then the matrix C_θ defined in (4.3) has $\mathbf{i}\hat{r}$ with $\hat{r} > 0$ as a repeated eigenvalue with even algebraic multiplicity.*

Proof. Without loss of generality, assume that $z_0 = 0$ and $\theta = 0$, and suppose that $\mathcal{R}_\theta$ locally supports $\Lambda_\varepsilon(A)$ at $\hat{r} > 0$. Thus, $\hat{r} \in \operatorname{bd}\Lambda_\varepsilon(A)$, and so $\sigma_{\min}(A - \hat{r}I) = \varepsilon$ and $\mathbf{i}\hat{r} \in \Lambda(C_\theta)$ by Lemma 4.1. By Definition 6.2, there exists a neighborhood $\mathcal{N}$ (in the open right half-plane) about $\hat{r}$ such that $(\Lambda_\varepsilon(A) \cap \mathcal{N}) \setminus \mathcal{R}_\theta$ is connected. As $\mathcal{R}_\theta$ separates $\mathcal{N}$ into $\mathcal{N}_1 = \{z \in \mathcal{N} : \operatorname{Im} z > 0\}$ and $\mathcal{N}_2 = \{z \in \mathcal{N} : \operatorname{Im} z < 0\}$, either $\Lambda_\varepsilon(A) \cap \mathcal{N}_1$ or $\Lambda_\varepsilon(A) \cap \mathcal{N}_2$ must be empty. Without loss of generality, suppose that $\Lambda_\varepsilon(A) \cap \mathcal{N}_1 = \varnothing$, and now consider how the eigenvalue $\mathbf{i}\hat{r}$ evolves as θ is varied, i.e., $\lambda(\theta) \in \Lambda(C_\theta)$ with $\lambda(0) = \mathbf{i}\hat{r}$. By continuity, the eigenvalue $\lambda(\theta)$ can either move up or down on the imaginary axis or it can move off the imaginary axis as the value of θ is increased from zero. If it moves along the imaginary axis, then locally we have that $\lambda(\theta) = \mathbf{i}r(\theta)$, where $r : \mathbb{R} \to \mathbb{R}$ is continuous and $r(0) = \hat{r}$. Since $\hat{r} > 0$, there exists a $\theta_{\mathrm{p}} > 0$ such that $r(\theta) > 0$ for all $\theta \in (0, \theta_{\mathrm{p}})$. By Lemma 4.1, it thus follows that $r(\theta)\mathrm{e}^{\mathbf{i}\theta} \in \Lambda_\varepsilon(A)$ for all $\theta \in (0, \theta_{\mathrm{p}})$, but this contradicts the assumption that $\Lambda_\varepsilon(A) \cap \mathcal{N}_1$ is empty. Thus, $\lambda(\theta)$ must move off the imaginary axis as the value of θ is increased from zero. Since the eigenvalues of the Hamiltonian matrix C_θ are symmetric with respect to the imaginary axis, by continuity at least one pair of eigenvalues (or possibly more pairs non-generically) must coalesce on the imaginary axis at $\mathbf{i}\hat{r}$ as $\theta \to 0$. ∎

Now consider the case when $\mathcal{R}_\theta$ locally supports $\Lambda_\varepsilon(A) \cap \Lambda_\varepsilon(B)$, which can happen at a boundary point of either $\Lambda_\varepsilon(A)$ or $\Lambda_\varepsilon(B)$ or a shared boundary point of both. Building on Lemma 6.3, we have the following result.

LEMMA 6.4. *Let $A \in \mathbb{C}^{m\times m}$, $B \in \mathbb{C}^{n\times n}$, $\varepsilon \geq 0$, $z_0 \in \mathbb{C}$, $\theta \in \mathbb{R}$, and $\mathcal{R}_\theta$ be the ray defined in (4.1). Furthermore, for the matrix A, let C_θ be the matrix defined in (4.3), and let S_θ be its analogue for the matrix B. If $\mathcal{R}_\theta$ locally supports $\Lambda_\varepsilon(A) \cap \Lambda_\varepsilon(B)$ at a point $z \in \mathbb{C}$, then at least one, and possibly all, of the following conditions must hold:*

(i) *C_θ and/or S_θ has $\mathbf{i}\hat{r}$ with $\hat{r} > 0$ as a repeated eigenvalue with even algebraic multiplicity,*
(ii) *C_θ and S_θ have an eigenvalue $\mathbf{i}\hat{r}$ with $\hat{r} > 0$ in common.*

Proof. Without loss of generality, we can assume that $z_0 = 0$ and $\theta = 0$, and so z is on the positive part of the real axis, i.e., $z = \hat{r}$ for some $\hat{r} > 0$. If $\mathcal{R}_\theta$ locally supports $\Lambda_\varepsilon(A) \cap \Lambda_\varepsilon(B)$ at $\hat{r}$, either $\hat{r} \in \operatorname{bd}\Lambda_\varepsilon(A)$ but not $\operatorname{bd}\Lambda_\varepsilon(B)$ (or vice versa) or $\hat{r}$ is a shared boundary point of both $\Lambda_\varepsilon(A)$ and $\Lambda_\varepsilon(B)$. If $\hat{r}$ is not a shared boundary point, then $\mathcal{R}_\theta$ must locally support either $\Lambda_\varepsilon(A)$ or $\Lambda_\varepsilon(B)$ at $\hat{r}$, and so Lemma 6.3 applies, yielding the "or" part of (i). Now suppose $\hat{r}$ is a shared boundary point, and so $\sigma_{\min}(A - \hat{r}I) = \sigma_{\min}(B - \hat{r}I) = \varepsilon$. Then by Lemma 4.1, $\mathbf{i}\hat{r}$ is an eigenvalue of both C_θ and S_θ, yielding (ii). Furthermore, $\mathcal{R}_\theta$ may or may not also locally support $\Lambda_\varepsilon(A)$ and/or $\Lambda_\varepsilon(B)$ at $\hat{r}$. All four scenarios are possible, with the "and" part of (i) corresponding to when the ray simultaneously locally supports both $\Lambda_\varepsilon(A)$ and $\Lambda_\varepsilon(B)$ at $\hat{r}$. ∎

Recall the set of roots $\mathcal{T}_\varepsilon$ of t_ε, which is defined in (5.2b). If $\theta \in \mathcal{T}_\varepsilon$, then the necessary condition for overlap $a_\varepsilon(\theta) + b_\varepsilon(\theta) = 0$ is satisfied, and so $\mathcal{R}_\theta$ intersects both $\Lambda_\varepsilon(A)$ and $\Lambda_\varepsilon(B)$. However, as $\ell_\varepsilon(\theta) = 0$, the sufficient condition is not met, and via Theorem 5.3 it follows that $\Lambda_\varepsilon(A)$ and $\Lambda_\varepsilon(B)$ either have no points in common along $\mathcal{R}_\theta$ or at most only boundary points in common. For a function to replace the regions of t_ε where $t_\varepsilon(\theta) = 0$,

ETNA
Kent State University and
Johann Radon Institute (RICAM)

i.e., $\mathcal{T}_\varepsilon$, we propose a function $d_\varepsilon^{AB} : \mathcal{T}_\varepsilon \to [0, \infty)$ that is a measure of how close $\Lambda_\varepsilon(A)$ and $\Lambda_\varepsilon(B)$ are to sharing a boundary point along $\mathcal{R}_\theta$. To that end, let

$$d_\varepsilon^{AB}(\theta) := \min\{d_\varepsilon^A(\theta),\, d_\varepsilon^B(\theta)\}, \qquad \text{where} \tag{6.1a}$$

$$d_\varepsilon^A(\theta) := \min\{f_A(r,\theta) - \varepsilon : z_0 + r\mathrm{e}^{\mathbf{i}\theta} \in \mathcal{R}_\theta \cap \operatorname{bd}\Lambda_\varepsilon(B)\}, \tag{6.1b}$$

$$d_\varepsilon^B(\theta) := \min\{f_B(r,\theta) - \varepsilon : z_0 + r\mathrm{e}^{\mathbf{i}\theta} \in \mathcal{R}_\theta \cap \operatorname{bd}\Lambda_\varepsilon(A)\}, \tag{6.1c}$$

where f_A is defined in (4.2) for the matrix A and f_B is its analogue for the matrix B. Since $\theta \in \mathcal{T}_\varepsilon$, both $\mathcal{R}_\theta \cap \operatorname{bd}\Lambda_\varepsilon(A)$ and $\mathcal{R}_\theta \cap \operatorname{bd}\Lambda_\varepsilon(B)$ must be nonempty, and so the functions are well defined. The purpose of d_ε^A is to provide a nonnegative measure of how close $\Lambda_\varepsilon(B)$ is to touching $\Lambda_\varepsilon(A)$ along the given ray $\mathcal{R}_\theta$ and vice versa for d_ε^B. Note that if $\mathcal{R}_\theta \cap \operatorname{bd}\Lambda_\varepsilon(A) \cap \operatorname{bd}\Lambda_\varepsilon(B) \neq \varnothing$, then $d_\varepsilon^A(\theta) = d_\varepsilon^B(\theta) = 0$, but otherwise $d_\varepsilon^A(\theta)$ and $d_\varepsilon^B(\theta)$ are typically not the same value. While technically d_ε^A alone (or d_ε^B) would suffice as a closeness measure of the two pseudospectra along a given ray, we have observed that their pointwise minimum, i.e., d_ε^{AB}, is often cheaper to approximate. Important properties of d_ε^{AB} are summarized in the following statement.

THEOREM 6.5 (Properties of d_ε^{AB}). *Let $A \in \mathbb{C}^{m\times m}$, $B \in \mathbb{C}^{n\times n}$, $\varepsilon \geq 0$, and $z_0 \in \mathbb{C}$ be such that $\varepsilon \geq 0$ is not a singular value of either $A - z_0 I$ or $B - z_0 I$, and let $\mathcal{R}_\theta$ be the ray defined in (4.1). Furthermore, let d_ε^{AB} be as defined in (6.1) on the domain $\mathcal{T}_\varepsilon$ defined in (5.2b). Then for any point $\theta \in \mathcal{T}_\varepsilon$, the following statements hold:*

(i) $d_\varepsilon^{AB}(\theta) \geq 0$,

(ii) $d_\varepsilon^{AB}(\theta) = 0 \iff \mathcal{R}_\theta \cap \Lambda_\varepsilon(A) \cap \Lambda_\varepsilon(B) \neq \varnothing \iff \mathcal{R}_\theta \cap \operatorname{bd}\Lambda_\varepsilon(A) \cap \operatorname{bd}\Lambda_\varepsilon(B) \neq \varnothing$,

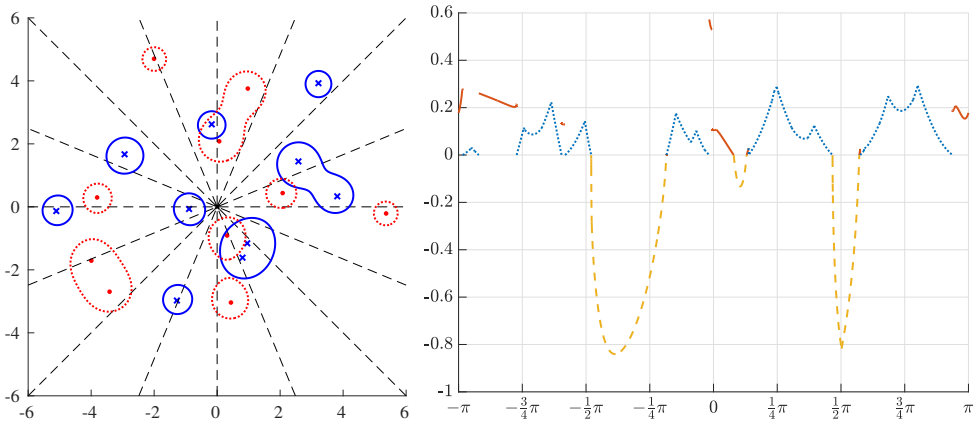
(iii) *d_ε^{AB} is continuous at θ if every eigenvalue $\mathbf{i}r$, of either C_θ or S_θ, that attains the minimum in $d_\varepsilon^{AB}(\theta)$ is simple,*

(iv) *d_ε^{AB} is differentiable at θ if there are no ties for $d_\varepsilon^{AB}(\theta)$, i.e., it is attained via $f_A(r,\theta)$ or $f_B(r,\theta)$ but not both, the corresponding minimum singular value is simple, and there is a single eigenvalue $\mathbf{i}r$, of either C_θ or S_θ as appropriate, that attains $d_\varepsilon^{AB}(\theta)$, where this eigenvalue is simple.*

Proof. Statements (i) and (ii) are simple but important direct consequences of the definition of d_ε^{AB} and the fact that its domain is restricted to $\mathcal{T}_\varepsilon$, since otherwise $d_\varepsilon^{AB}(\theta)$ could be negative (or undefined) for some θ and the equivalences in (ii) would not hold. For statement (iii), consider $d_\varepsilon^B(\theta)$ and recall that by Lemma 4.1, $z_0 + \hat{r}\mathrm{e}^{\mathbf{i}\theta} \in \mathcal{R}_\theta \cap \operatorname{bd}\Lambda_\varepsilon(A)$ is always associated with an eigenvalue $\mathbf{i}\hat{r}$ of C_θ. Since eigenvalues are continuous, the eigenvalue $\mathbf{i}\hat{r}$ can either move continuously along the positive portion of the imaginary axis or leave this region as θ is varied. Clearly, the former case cannot cause a discontinuity in d_ε^B, so consider the latter. By the assumption on ε, zero can never be an eigenvalue of C_θ for any θ, and clearly the eigenvalues of a matrix are all finite. Thus, if an eigenvalue leaves the positive portion of the imaginary axis, it cannot do by going through the origin or infinity. Since the eigenvalues of the Hamilton matrix C_θ are symmetric with respect to the imaginary axis, a simple eigenvalue cannot leave the imaginary axis, and a repeated eigenvalue is excluded by assumption; hence, d_ε^B must be continuous at θ. The same argument shows that d_ε^A is continuous at θ under the analogous assumptions for the eigenvalues of S_θ, and so d_ε^{AB} is continuous at θ. For (iv), the assumptions mean that there are no ties for the $\min$ functions, and standard perturbation theory for simple singular values and simple eigenvalues applies. ∎

While Theorem 6.5 verifies that d_ε^{AB} is reasonably well behaved, d_ε^{AB} may have jump discontinuities. However, d_ε^{AB} is discontinuous at a point $\theta \in \operatorname{int}\mathcal{T}_\varepsilon$ only if two conditions simultaneously hold: $\mathcal{R}_\theta$ locally supports $\Lambda_\varepsilon(A)$ or $\Lambda_\varepsilon(B)$ at a point $z_0 + \hat{r}\mathrm{e}^{\mathbf{i}\theta}$ with $\hat{r} > 0$, and this value $\hat{r}$ is the one that attains the value of $d_\varepsilon^{AB}(\theta)$. As a result, we expect such

ETNA
Kent State University and
Johann Radon Institute (RICAM)



(a) $\Lambda_\varepsilon(A)$ and $\Lambda_\varepsilon(B)$ (b) $d_\varepsilon(\theta)$

FIG. 6.1. *For two randomly generated matrices $A, B \in \mathbb{C}^{10\times 10}$, the left pane shows their eigenvalues (respectively x's and dots) and $\Lambda_\varepsilon(A)$ and $\Lambda_\varepsilon(B)$ (respectively solid and dotted contours) for $\varepsilon = 0.3 > \mathrm{sep}^{\mathrm{D}}_\lambda(A, B)$. The search point z_0 is the origin; rays emanating from it are depicted by dashed lines. The right pane shows a corresponding plot of d_ε, where its components are plotted as follows: $a_\varepsilon + b_\varepsilon$ (dotted), ℓ_ε (dashed), and d_ε^{AB} (solid). For $\theta = -\frac{1}{2}\pi$, it can be seen in the left pane that $\mathcal{R}_\theta$ only passes through $\Lambda_\varepsilon(B)$ and so $d_\varepsilon(\theta) = a_\varepsilon(\theta) + b_\varepsilon(\theta) > 0$ in the right pane. Meanwhile for $\theta = \frac{1}{2}\pi$, $\mathcal{R}_\theta$ passes through $\operatorname{int}\Lambda_\varepsilon(A) \cap \operatorname{int}\Lambda_\varepsilon(B)$ and so $d_\varepsilon(\theta) = \ell_\varepsilon(\theta) < 0$. Finally, for $\theta = 0$, while $\mathcal{R}_\theta$ passes through both $\Lambda_\varepsilon(A)$ and $\Lambda_\varepsilon(B)$, it never does so simultaneously, hence $a_\varepsilon(\theta) + b_\varepsilon(\theta) = \ell_\varepsilon(\theta) = 0$ and $d_\varepsilon(\theta) = d_\varepsilon^{AB}(\theta) > 0$.*

discontinuities to be relatively few, and so this should not be a problem in practice. The functions d_ε^A and d_ε^B typically do not have non-Lipschitz behavior when they transition to/from zero, and so we have not added smoothing to them when defining d_ε^{AB}. When f^{D} has a unique minimizer, d_ε^{AB} only has a single root for $\varepsilon = \mathrm{sep}^{\mathrm{D}}_\lambda(A, B)$.

Combining our three constituent pieces, we now define $d_\varepsilon : (-\pi, \pi] \to \mathbb{R}$, our key function for our interpolation-based globality certificate for $\mathrm{sep}^{\mathrm{D}}_\lambda(A, B)$:

$$d_\varepsilon(\theta) := \begin{cases} a_\varepsilon(\theta) + b_\varepsilon(\theta) & \text{if } a_\varepsilon(\theta) + b_\varepsilon(\theta) > 0, \\ \ell_\varepsilon(\theta) & \text{if } \ell_\varepsilon(\theta) < 0, \\ d_\varepsilon^{AB}(\theta) & \text{otherwise.} \end{cases} \tag{6.2}$$

In Figure 6.1, we plot d_ε for a sample problem with $\varepsilon > \mathrm{sep}^{\mathrm{D}}_\lambda(A, B)$ in order to illustrate the different components of d_ε. Recalling that $a_\varepsilon + b_\varepsilon$ is a nonnegative function and so is d_ε^{AB} on its domain, we immediately have the following global convergence conditions as a corollary of Theorems 5.2, 5.3, and 6.5.

COROLLARY 6.6 (Global convergence for $\mathrm{sep}^{\mathrm{D}}_\lambda(A, B)$ via d_ε). *Let $A \in \mathbb{C}^{m\times m}$, $B \in \mathbb{C}^{n\times n}$, $\varepsilon \geq 0$, and $z_0 \in \mathbb{C}$ be such that ε is not a singular value of either $A - z_0 I$ or $B - z_0 I$, and let d_ε be the function defined in (6.2). Then*

$$\min_{\theta\in(-\pi,\pi]} d_\varepsilon(\theta) < 0 \iff \mu(\{\theta \in (-\pi, \pi] : d_\varepsilon(\theta) < 0\}) > 0 \iff \varepsilon > \mathrm{sep}^{\mathrm{D}}_\lambda(A, B).$$

In the process of devising d_ε, we considered many different possibilities but found that these alternatives were significantly more expensive to use than d_ε, even if they had fewer jumps or even none. For example, we considered an entirely continuous alternative to d_ε that replaced its d_ε^{AB}-portions with a continuous measure of the distance to any of the necessary

ETNA
Kent State University and
Johann Radon Institute (RICAM)

conditions in Lemma 6.4 holding. However, this function often had a more complicated behavior and many more roots than d_ε because the necessary conditions in Lemma 6.4 hold for any θ such that $\mathcal{R}_\theta$ locally supports either of the two pseudospectra or their intersection and possibly at other angles as well. Even when incorporating smoothing to address non-Lipschitz behavior at roots, this alternative was still much more expensive to approximate than d_ε. We also tried replacing d_ε^{AB} with $\min\{\mu(\mathcal{R}_\theta \cap \Lambda_\varepsilon(A)), \mu(\mathcal{R}_\theta \cap \Lambda_\varepsilon(B))\}$ and other continuous alternatives, although these choices still resulted in jumps when used in conjunction with $a_\varepsilon + b_\varepsilon$ and ℓ_ε. But these choices were more expensive to approximate than d_ε^{AB} because they generally had more complicated behaviors than d_ε^{AB}, e.g., more nonsmooth points, more oscillatory behavior, etc. Finally, we considered just using the smallest pairwise distance between points in $\mathcal{R}_\theta \cap \operatorname{bd}\Lambda_\varepsilon(A)$ and $\mathcal{R}_\theta \cap \operatorname{bd}\Lambda_\varepsilon(B)$. This is quite similar to d_ε^{AB} and can have similar discontinuities, but it too ended up being more expensive to approximate than d_ε^{AB}. That all said, none of the alternatives we considered were prohibitively expensive; using any of them to compute $\operatorname{sep}_\lambda^{\mathrm{D}}(A, B)$ was still much faster than the method of Gu and Overton, even though they were generally not as fast as our ultimate choice for d_ε.

REMARK 6.7. Another approach to computing $\operatorname{sep}_\lambda^{\mathrm{D}}(A, B)$ is via

$$s(\theta) \coloneqq \min_{r\in\mathbb{R}} s_\theta(r), \quad \text{where} \quad s_\theta(r) \coloneqq \max\{\sigma_{\min}(F_A(r,\theta)),\, \sigma_{\min}(F_B(r,\theta))\},$$

i.e., $s(\theta)$ is the minimal value f^{D} takes along the line defined by θ and passing through some $z_0 \in \mathbb{C}$. It is then immediate that

$$\operatorname{sep}_\lambda^{\mathrm{D}}(A, B) = \min_{\theta\in[0,\pi)} s(\theta), \tag{6.3}$$

as this simply rewrites (1.2b) in polar coordinates about z_0. Thus, using Chebfun to approximate s and then find a global minimizer in $[0, \pi)$ provides another way to obtain $\operatorname{sep}_\lambda^{\mathrm{D}}(A, B)$. One drawback of this approach is that for any given θ, evaluating $s(\theta)$ is much more expensive than evaluating $d_\varepsilon(\theta)$. As we explain in detail in the next section, evaluating $d_\varepsilon(\theta)$ is essentially direct, since it only requires solving two eigenvalue problems of order $2m$ and $2n$ and this is generally the dominant cost. Meanwhile, computing $s(\theta)$ involves finding a global minimizer of s_θ, which requires iteration. Although we can use Lemma 4.1 to construct such an iteration, similar to the level-set methods of [6, 9] for computing the $\mathcal{H}_\infty$-norm, the resulting algorithm to compute $s(\theta)$ would generally only be linearly convergent; the key difference between here and the $\mathcal{H}_\infty$-norm setting is that s_θ, due to being a $\max$ of two $\min$ functions, will generally be nonsmooth at its minimizers. Consequently, evaluating $s(\theta)$ would require solving multiple eigenvalue problems of $2m$ and $2n$. Another issue is that although s is continuous, it is still nonsmooth, and it is generally more expensive for Chebfun to detect nonsmooth points than jumps; see [39, 43]. Finally, a third downside is that using Chebfun to *precisely* compute a (likely unique) global minimizer of some function, e.g., s, is a significantly more numerically challenging task than what we ask of Chebfun inside our algorithm using d_ε, i.e., to find *any point* where d_ε is negative, since as we have shown, the set of such points has positive measure when $\varepsilon > \operatorname{sep}_\lambda^{\mathrm{D}}(A, B)$. Thus, when attempting to compute $\operatorname{sep}_\lambda^{\mathrm{D}}(A, B)$ by applying Chebfun to (6.3), we nevertheless recommend subsequently refining its computed result by applying local optimization to f^{D} initialized from the point in the complex plane found by Chebfun.

7. Implementation and the cost of our method. We now discuss how to implement our $\operatorname{sep}_\lambda^{\mathrm{D}}(A, B)$-algorithm, which we have done in MATLAB, and describe its overall work complexity. We give detailed remarks in the following sections, while a high-level pseudocode is given in Algorithm 7.1.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Algorithm 7.1 Interpolation-based Globality Certificate Algorithm for $\mathrm{sep}_\lambda^{\mathrm{D}}(A, B)$.

Input: $A \in \mathbb{C}^{m\times m}$, $B \in \mathbb{C}^{n\times n}$, "search point" $z_0 \in \mathbb{C}$, and $z_{\mathrm{init}} \in \mathbb{C}$.
Output: $\varepsilon \approx \mathrm{sep}_\lambda^{\mathrm{D}}(A, B)$.

1: **while** true **do**
2: $\varepsilon \leftarrow$ computed locally/globally minimal value of f^{D} initialized from z_{init}
3: *// Begin approximating d_ε to assert convergence or find new starting points*
4: $p_\varepsilon \leftarrow 1$ *// Initial guess for polynomial interpolant p_ε for approximating d_ε*
5: **while** p_ε does not sufficiently approximate d_ε **do**
6: $[\theta_1, \ldots, \theta_q] \leftarrow$ new sample points from $(-\pi, \pi]$
7: *// If new starting points are detected, restart optimization to lower ε:*
8: **if** $d_\varepsilon(\theta_j) < 0$ for some $j \in \{1, \ldots, q\}$ **then**
9: $z_{\mathrm{init}} \leftarrow$ a point in $\mathrm{bd}\{\mathcal{R}_{\theta_j} \cap \Lambda_\varepsilon(A) \cap \Lambda_\varepsilon(B)\} \setminus \{z_0\}$
10: **goto line 2** *// Restart optimization from z_{init}*
11: **end if**
12: *// Otherwise, no starting points detected, keep improving p_ε:*
13: $p_\varepsilon \leftarrow$ improved polynomial interpolant of d_ε via $\theta_1, \ldots, \theta_q$
14: **end while**
15: *// p_ε approximates d_ε well and no new starting points were encountered*
16: *// However, do a final check before asserting that d_ε is nonnegative:*
17: $[\theta_1, \ldots, \theta_q] = \arg\min p_\varepsilon(\theta)$
18: **if** $d_\varepsilon(\theta_j) < 0$ for some $j \in \{1, \ldots, q\}$ **then**
19: $z_{\mathrm{init}} \leftarrow$ a point in $\mathrm{bd}\{\mathcal{R}_{\theta_j} \cap \Lambda_\varepsilon(A) \cap \Lambda_\varepsilon(B)\} \setminus \{z_0\}$
20: **goto line 2** *// Restart optimization from z_{init}*
21: **else**
22: **return** *// $p_\varepsilon \approx d_\varepsilon$ and* $\implies$ $\varepsilon \approx \mathrm{sep}_\lambda^{\mathrm{D}}(A, B)$
23: **end if**
24: **end while**

NOTE: *To keep the pseudocode a reasonable length, we make some simplifying assumptions: optimization converges to local/global minimizers exactly, z_{init} computed in lines 9 and 19 for restarting optimization is never a stationary point of f^{D}, and the "search point" z_0 is such that all encountered values of ε are not singular values of $\sigma_{\min}(A - z_0 I)$ and $\sigma_{\min}(B - z_0 I)$, per the assumptions given in Section 4 and Section 5. Lines 3–15 describe the core of the interpolation-based globality certificate, where we only give a broad outline of the interpolation process for approximating d_ε; note that for numerical reasons, each certificate should actually be done with $\tilde{\varepsilon} = (1 - \tau)\varepsilon$, where $\tau \in (0, 1)$ is some relative tolerance. See Section 7.2 and Section 7.3 for more implementation details.*

7.1. Choosing a search point. Regarding what search point z_0 to use, we recommend the average of all the distinct eigenvalues of A and B. This helps to ensure that the whole domain of d_ε is relevant. Otherwise, if for a given value of ε, z_0 is chosen far from the pseudospectra of A and B, then $a_\varepsilon(\theta) + b_\varepsilon(\theta) = 0$ would only hold on a very small subset of $(-\pi, \pi]$, which in turn would likely make it harder to find the regions where $d_\varepsilon(\theta)$ is negative. On every round, our code checks that the choice of z_0 still satisfies our needed assumptions and perturbs it slightly if it does not (in practice, we have not observed that this is necessary). Finally, if the pseudospectra of A and B both have real-axis symmetry, by choosing z_0 on the real axis, it is then only necessary to approximate d_ε on $[0, \pi]$.

7.2. Evaluating $d_\varepsilon(\theta)$ and its cost. Given some θ, evaluating $d_\varepsilon(\theta)$ proceeds as follows. First, the eigenvalues of both C_θ and S_θ are computed. For increased reliability, it is recom-

ETNA
Kent State University and
Johann Radon Institute (RICAM)

mended that this be done via a structure-preserving eigensolver such as [2]. From these spectra, it is then trivial to calculate the value of $a_\varepsilon(\theta)+b_\varepsilon(\theta)$ via (4.5a). If $a_\varepsilon(\theta)+b_\varepsilon(\theta) > 0$, then the value of $d_\varepsilon(\theta)$ has been computed. Otherwise, evaluating $d_\varepsilon(\theta)$ requires the following additional computations, which begins with obtaining the value of $\ell_\varepsilon(\theta)$. To that end, we compute $\mathcal{R}_\theta \cap \Lambda_\varepsilon(A)$ and $\mathcal{R}_\theta \cap \Lambda_\varepsilon(B)$. Considering the former, we want to determine the values $r > 0$ such that $f_A(r,\theta) = \varepsilon$, and via Lemma 4.1, we have the following sorted list of candidate values $0 = r_0 < r_1 < \ldots < r_q$ that may satisfy this equality, where $\mathbf{i}r_j$, for $j = 1,\ldots,q$, are eigenvalues of C_θ and we have added $r_0 = 0$. Then to compute $\mathcal{R}_\theta \cap \Lambda_\varepsilon(A)$, we must assert which intervals on $\mathcal{R}_\theta$, defined by $[r_{j-1}, r_j]$ for $j = 1,\ldots,q$, are also in $\Lambda_\varepsilon(A)$. There are several ways to do this, but a simple and robust way is to just evaluate $f_A(\hat{r}_j,\theta)$ for $\hat{r}_j = 0.5(r_{j-1}+r_j)$ over $j = 1,\ldots,q$; since $f_A(\hat{r}_j,\theta) \neq \varepsilon$, the corresponding interval is not in $\mathcal{R}_\theta \cap \Lambda_\varepsilon(A)$ if and only if $f_A(\hat{r}_j,\theta) > \varepsilon$. Note that it does not matter if we have two or more adjacent intervals in our computed version of $\mathcal{R}_\theta \cap \Lambda_\varepsilon(A)$. An analogous computation yields $\mathcal{R}_\theta \cap \Lambda_\varepsilon(B)$. With these two sets computed, calculating the amount of their overlap along the given ray, i.e., $-\ell_\varepsilon(\theta)$, is straightforward. If $\ell_\varepsilon(\theta) < 0$, then the evaluation of $d_\varepsilon(\theta)$ is done, and the boundary points of $\mathcal{R}_\theta \cap \Lambda_\varepsilon(A) \cap \Lambda_\varepsilon(B)$ have also been computed, which are used to restart optimization. However, if $\ell_\varepsilon(\theta) = 0$, then finally we must compute $d_\varepsilon^{AB}(\theta)$ in order to complete the computation of $d(\theta)$, though this is straightforward to do from the definition of $d_\varepsilon^{AB}(\theta)$ given in (6.1) and the previous computations.

Recalling our assumption that $m \leq n$, evaluating $d_\varepsilon(\theta)$ needs $\mathcal{O}(n^3)$ work if done in the following manner. Computing all of the eigenvalues of C_θ and S_θ requires $\mathcal{O}(n^3)$ work, and that is all there is to do when $a_\varepsilon(\theta) + b_\varepsilon(\theta) > 0$. But when $a_\varepsilon(\theta) + b_\varepsilon(\theta) = 0$, computing $d_\varepsilon(\theta)$ additionally requires computing the values of $f_A(r,\theta)$ and $f_B(r,\theta)$ for different values of r. While the number of values of r is often only a handful, in the worst case, it can be $\mathcal{O}(m+n)$. Hence, if we were to evaluate this pair of functions by computing SVDs, we would exceed the stated $\mathcal{O}(n^3)$ work complexity bound by a factor of n. Fortunately, there is a more efficient option due to Lui for fast plotting of pseudospectra [29]. Since A is square, it has a Schur decomposition $A = UTU^*$, where U is unitary and T is triangular, and moreover, since unitary transformations do not alter the pseudospectrum, $\Lambda_\varepsilon(A) = \Lambda_\varepsilon(T)$ holds. The key benefit of this transformation is that at any point $z_0 + re^{\mathbf{i}\theta} \in \mathbb{C}$, we have that $T - (z_0 + re^{\mathbf{i}\theta})I$ remains in triangular form, and so inverse iteration can be done to compute this shifted matrix's minimum singular value using backsolves that only require quadratic work as opposed to the usual cubic work for solving a linear system. We only need to compute and store Schur decompositions of A and B once in an offline phase, which is cubic work, and then we can evaluate $f_A(r,\theta)$ and $f_B(r,\theta)$ for any r and θ in a most $\mathcal{O}(n^2)$ work under the mild assumption that inverse iteration converges in relatively few steps.[4] Hence, evaluating $d_\varepsilon(\theta)$ can always be done within $\mathcal{O}(n^3)$ work. In our own experience, we have seen that ten iterations is generally more than sufficient to compute $f_A(r,\theta)$ and $f_B(r,\theta)$ accurately to the full precision of the hardware, and that this technique is already faster than computing the full SVD for matrices as small as 50×50.

7.3. Approximating d_ε and restarting. To approximate d_ε, we use Chebfun, as it is rather adept at approximating functions with nonsmooth points and/or discontinuities. As Chebfun normally provides groups of points to evaluate simultaneously (line 6 of Algorithm 7.1), these evaluations of d_ε can be done in parallel; see [36, Section 5.2] for more details. Furthermore, if $d_\varepsilon(\theta) < 0$ for any of the current group of points provided by Chebfun, then we immediately halt Chebfun and use the detected boundary points of $\mathcal{R}_\theta \cap \Lambda_\varepsilon(A) \cap \Lambda_\varepsilon(B)$

[4] For more details on the actual inverse-iteration-based algorithm, including pseudocode and code examples, see [29] and [44, Chapter 39], but note that the latter has the following typo: In "Core EigTool algorithm" [44, p. 375], the second to last line should be `sigmin(j,k) = 1/sqrt(sig);`, not `sigmin(j,k) = sqrt(sig);`.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

(except for z_0) to restart optimization (lines 7–11 of Algorithm 7.1). This is accomplished by throwing an error when a point is encountered such that $d_\varepsilon(\theta) < 0$ holds, which causes Chebfun to be aborted. By subsequently catching our own thrown error, we can resume our program to restart another round of optimization.

7.4. Finding minimizers. Like many other optimization-with-restarts algorithms, it will be necessary to use a *monotonic* optimization solver, i.e., one that always decreases the objective function in every iteration, which is the case for most unconstrained optimization solvers. Minimizers of f^{D} will almost always be nonsmooth, and at best, we can expect linear convergence from a nonsmooth optimization solver. However, since there are only two real variables, we expect the number of iterations needed to converge to be relatively small. Thus, as evaluating f^{D} and its gradient is significantly cheaper than evaluating d_ε and we expect far fewer function evaluations for the former than the latter, the cost of Algorithm 7.1 will generally not be dominated by the optimization phases.

To find minimizers of f^{D} using only gradient information, we use GRANSO: GRadient-based Algorithm for Non-Smooth Optimization [33]. GRANSO implements the BFGS-SQP nonsmooth optimization algorithm of [16], which can handle nonsmooth constraints, but for problems without constraints, it reduces to BFGS with the line search of [28], a combination which Lewis and Overton have studied and advocated as a method for nonsmooth optimization. While there are no convergence results for BFGS for general nonsmooth optimization, it nevertheless seems to reliably and accurately converge to nonsmooth stationary values. Indeed, in their concluding remarks [28, p. 160], Lewis and Overton wrote "In our experience with functions with bounded sublevel sets, BFGS essentially always generates function values converging linearly to a Clarke stationary value, with exceptions only in cases that we attribute to the limits of machine precision. We speculate that, for some broad class of reasonably well-behaved functions, this behavior is almost sure." Since f^{D} is locally Lipschitz as long as $\mathrm{sep}_\lambda^{\mathrm{D}}(A, B) > 0$ and has bounded level sets, we expect that BFGS will also be an efficient and reliable tool in our setting. For improved theoretical guarantees, one could follow up optimization via BFGS with a phase of the gradient sampling algorithm [13], which would ensure convergence to nonsmooth stationary values of f^{D} when $\mathrm{sep}_\lambda^{\mathrm{D}}(A, B) > 0$. However, for simplicity, we only use BFGS here.

When restarting optimization, our certificate may provide many new starting points. Restarting from just one would give the smallest chance of converging to a global minimizer on this round, while restarting from them all could be a waste of time, particularly if this ends up just returning the same minimizer over and over again. In practice, one could prioritize them in terms of most promising first and limit the total number used. On multi-core machines, optimization can be run from multiple starting points in parallel.

7.5. Terminating the algorithm. In addition to the convergence tests described in Algorithm 7.1, it is also necessary to terminate the algorithm if consecutive estimates for $\mathrm{sep}_\lambda^{\mathrm{D}}(A, B)$ are identical. The reason is that we cannot expect optimization solvers to find minimizers *exactly*. If a global minimizer $\tilde{z}$ is obtained only up to some rounding error, then $\mathrm{sep}_\lambda^{\mathrm{D}}(A, B)$ has essentially been computed, but our certificate may still detect that the algorithm has not *truly* converged to a global minimizer, and in this case, the algorithm may try to restart optimization (unsuccessfully). This is also part of the reason why the certificates should actually be performed with $\tilde{\varepsilon} = (1 - \tau)\varepsilon$, as described in the note under Algorithm 7.1.

7.6. The overall work complexity and using lines instead of rays. In the worst case, the overall work complexity to perform the interpolation-based globality certificates is $\mathcal{O}(kn^3)$, where k is the total number of function evaluations (over all values of ε encountered). As restarts tend to happen quickly, k is roughly equal to the number of evaluations needed to

ETNA
Kent State University and
Johann Radon Institute (RICAM)

approximate d_ε when $\varepsilon = \operatorname{sep}_\lambda^{\mathrm{D}}(A, B)$, and as we will see in the numerical experiments, k can generally be considered to be like a large constant, although it is influenced by the geometry of the two pseudospectra.

When implementing the algorithm, the definition of d_ε can be modified so that it considers lines through z_0 instead of rays emanating from z_0. This can be beneficial, since we always get information for the direction $\theta + \pi$ when considering $\mathcal{R}_\theta$, and so this modified version of d_ε needs only be interpolated on $[0, \pi]$. A function a_ε measures the minimum argument of $-\mathbf{i}\lambda$ over each eigenvalue λ of C_θ, so when using lines instead of rays, it must also consider the minimum angle with respect to the negative real axis. These additional angles are computed by simply switching the sign of the imaginary part of each eigenvalue λ. The same change is made for b_ε, while modifying ℓ_ε and d_ε^{AB} is straightforward. While using lines often results in less overall work, this is not always the case, as it can sometimes make d_ε more complicated and thus more expensive to approximate.

8. Algorithms for $\operatorname{sep}_\lambda^{\mathrm{V}}(A, B)$. We now briefly turn to the problem of computing Varah's $\operatorname{sep}_\lambda(A, B)$. We first answer whether or not Algorithm 7.1 extends to $\operatorname{sep}_\lambda^{\mathrm{V}}(A, B)$ and then propose a different algorithm to compute $\operatorname{sep}_\lambda^{\mathrm{V}}(A, B)$.

8.1. Does Algorithm 7.1 extend to $\operatorname{sep}_\lambda^{\mathrm{V}}(A, B)$? In the construction of the function d_ε for computing $\operatorname{sep}_\lambda^{\mathrm{D}}(A, B)$, nowhere have we needed that the same value of ε be used for the pseudospectra of A and B. Thus for Varah's version of $\operatorname{sep}_\lambda(A, B)$, we can analogously define

$$d_{\varepsilon_1,\varepsilon_2}(\theta) \coloneqq \begin{cases} a_{\varepsilon_1}(\theta) + b_{\varepsilon_2}(\theta) & \text{if } a_{\varepsilon_1}(\theta) + b_{\varepsilon_2}(\theta) > 0, \\ \ell_{\varepsilon_1,\varepsilon_2}(\theta) & \text{if } \ell_{\varepsilon_1,\varepsilon_2}(\theta) < 0, \\ d^{AB}_{\varepsilon_1,\varepsilon_2}(\theta) & \text{otherwise,} \end{cases} \tag{8.1}$$

where

$$\begin{aligned} \ell_{\varepsilon_1,\varepsilon_2}(\theta) &\coloneqq -\mu\left(\mathcal{R}_\theta \cap \Lambda_{\varepsilon_1}(A) \cap \Lambda_{\varepsilon_2}(B)\right), \\ d^{AB}_{\varepsilon_1,\varepsilon_2}(\theta) &\coloneqq \min\{d^{A}_{\varepsilon_1,\varepsilon_2}(\theta), d^{B}_{\varepsilon_1,\varepsilon_2}(\theta)\}, \\ d^{A}_{\varepsilon_1,\varepsilon_2}(\theta) &\coloneqq \min\{f_A(r,\theta) - \varepsilon_1 : \mathcal{R}_\theta \cap \operatorname{bd}\Lambda_{\varepsilon_2}(B)\}, \\ d^{B}_{\varepsilon_1,\varepsilon_2}(\theta) &\coloneqq \min\{f_B(r,\theta) - \varepsilon_2 : \mathcal{R}_\theta \cap \operatorname{bd}\Lambda_{\varepsilon_1}(A)\}, \end{aligned}$$

and f_A is defined in (4.2) for the matrix A, while f_B is its analogue for the matrix B. Although this will not allow us to compute $\operatorname{sep}_\lambda^{\mathrm{V}}(A, B)$ to arbitrary accuracy, we do have the following necessary condition as another corollary of Theorem 5.3.

COROLLARY 8.1 (A necessary condition for $\varepsilon_1 + \varepsilon_2 = \operatorname{sep}_\lambda^{\mathrm{V}}(A, B)$ via $d_{\varepsilon_1,\varepsilon_2}$). *Let $A \in \mathbb{C}^{m\times m}$, $B \in \mathbb{C}^{n\times n}$, $\varepsilon_1, \varepsilon_2 \geq 0$, and $z_0 \in \mathbb{C}$ be such that ε_1 and ε_2 are, respectively, not singular values of $A - z_0 I$ and $B - z_0 I$, and let $d_{\varepsilon_1,\varepsilon_2}$ be the function defined in (8.1). Then*

$$\min_{\theta\in(-\pi,\pi]} d_{\varepsilon_1,\varepsilon_2}(\theta) < 0 \iff \mu(\{\theta \in (-\pi,\pi] : d_{\varepsilon_1,\varepsilon_2}(\theta) < 0\}) > 0,$$

and

$$\varepsilon_1 + \varepsilon_2 > \operatorname{sep}_\lambda^{\mathrm{V}}(A, B) \qquad \text{if} \qquad \min_{\theta\in(-\pi,\pi]} d_{\varepsilon_1,\varepsilon_2}(\theta) < 0.$$

As the last statement in Corollary 8.1 is not if-and-only-if, $d_{\varepsilon_1,\varepsilon_2}$ does not allow us compute to $\operatorname{sep}_\lambda^{\mathrm{V}}(A, B)$ with guaranteed accuracy. However, by modifying Algorithm 7.1 to instead find minimizers of f^{V} and use $d_{\varepsilon_1,\varepsilon_2}$, we can compute locally optimal upper bounds for

ETNA
Kent State University and
Johann Radon Institute (RICAM)

$\operatorname{sep}_\lambda^{\mathrm{V}}(A,B)$ that at least guarantee that the necessary condition $\operatorname{int}\Lambda_{\varepsilon_1}(A)\cap\operatorname{int}\Lambda_{\varepsilon_2}(B)=\varnothing$ is satisfied, as this is equivalent to $\min_{\theta\in(-\pi,\pi]} d_{\varepsilon_1,\varepsilon_2}(\theta)=0$. This is notably better than just computing upper bounds via finding minimizers of f^{V}, since the corresponding values of ε_1 and ε_2 associated with minimizers are not guaranteed to satisfy this necessary condition. However, when either $\varepsilon_1=0$ or $\varepsilon_2=0$ holds at the computed minimizer, note that $\operatorname{int}\Lambda_{\varepsilon_1}(A)=\varnothing$ or $\operatorname{int}\Lambda_{\varepsilon_2}(B)=\varnothing$ holds, and so satisfying the necessary condition does not preclude the possibility that an eigenvalue of A may be in $\operatorname{int}\Lambda_{\varepsilon_2}(B)$ or vice versa. Thus, when approximating $\operatorname{sep}_\lambda^{\mathrm{V}}(A,B)$ via this extended algorithm, one should always compute

$$\tilde{\varepsilon}=\min\left\{\min_{\lambda\in\Lambda(B)}\sigma_{\min}(A-\lambda I),\ \min_{\lambda\in\Lambda(A)}\sigma_{\min}(B-\lambda I)\right\}, \tag{8.2}$$

which computes an upper bound $\tilde{\varepsilon}\geq\operatorname{sep}_\lambda^{\mathrm{V}}(A,B)$ such that no eigenvalues of A are in the interior of $\operatorname{int}\Lambda_{\tilde{\varepsilon}}(B)$ and vice versa. Nevertheless, when optimization finds minimizers where neither ε_1 nor ε_2 is zero, then our certificate can be used to restart optimization if the necessary condition does not hold and hence obtain a better estimate for $\operatorname{sep}_\lambda^{\mathrm{V}}(A,B)$.

8.2. A different Chebfun-based algorithm to compute $\operatorname{sep}_\lambda^{\mathrm{V}}(A,B)$. Given $z_0\in\mathbb{C}$, let the function $v:[0,\pi)\to\mathbb{R}$ be defined as

$$v(\theta)\coloneqq\min_{r\in\mathbb{R}} v_\theta(r),\quad\text{where}\quad v_\theta(r)\coloneqq\sigma_{\min}(F_A(r,\theta))+\sigma_{\min}(F_B(r,\theta)), \tag{8.3}$$

i.e., $v(\theta)$ is the minimal value f^{V} takes along the line defined by θ and passing through z_0. It then immediately follows that

$$\operatorname{sep}_\lambda^{\mathrm{V}}(A,B)=\min_{\theta\in[0,\pi)} v(\theta). \tag{8.4}$$

Since v is a continuous function defined on a finite interval, as in the alternative $\operatorname{sep}_\lambda^{\mathrm{D}}(A,B)$-algorithm discussed in Remark 6.7, we can consider approximating v with Chebfun in order to solve (8.4).

Unfortunately evaluating v for a given θ is quite difficult, as the level-set iteration for finding a global minimizer of s_θ described in Remark 6.7 does not extend to v_θ. However, for some $\varepsilon>\operatorname{sep}_\lambda^{\mathrm{V}}(A,B)$, say $\varepsilon=f^{\mathrm{V}}(z_0)$, we can easily calculate a finite interval $[r_1,r_2]$ such that $v_\theta(r)>\varepsilon$ must hold for all $r\notin[r_1,r_2]$. To do this, we simple apply Lemma 4.1 to obtain the two extremal points, say a_1 and a_2 with $a_1\leq a_2$, in the ε-level set of $\sigma_{\min}(F_A(r,\theta))$ with r varying and θ fixed, and then analogously also obtain the two extremal level-set points b_1 and b_2 of $\sigma_{\min}(F_B(r,\theta))$ with $b_1\leq b_2$. By taking $r_1=\max\{a_1,b_1\}$ and $r_2=\min\{a_2,b_2\}$, we have that any global minimizer of v_θ must lie in $[r_1,r_2]$, since by construction, $v_\theta(r)>\varepsilon$ outside this interval. Thus, to obtain the value of $v(\theta)$, we simply solve two eigenvalue problems to obtain $[r_1,r_2]$ and then apply Chebfun to approximate v_θ on $[r_1,r_2]$ in order to obtain its globally minimal value.

Using Chebfun to approximate v over $[0,\pi)$, where for each θ the value of $v(\theta)$ is also computed by applying Chebfun to v_θ, does lead to quite an expensive algorithm, as many evaluations of $\sigma_{\min}(F_A(r,\theta))$ and $\sigma_{\min}(F_B(r,\theta))$ for different values of θ and r are required. However, this nested Chebfun-based algorithm nevertheless has the virtue of being the very first algorithm to compute $\operatorname{sep}_\lambda^{\mathrm{V}}(A,B)$, as opposed to just approximating it, e.g., within a factor of two by instead computing $\operatorname{sep}_\lambda^{\mathrm{D}}(A,B)$.

Regarding the choice of z_0, one might be tempted to use a local minimizer of f^{V}, but there are pros and cons to doing so. On the upside, if $\varepsilon=f^{\mathrm{V}}(z_0)$ is close to $\operatorname{sep}_\lambda^{\mathrm{V}}(A,B)$, then v likely will be constant (with value ε) on a much of $[0,\pi)$ or all of it if $\varepsilon=\operatorname{sep}_\lambda^{\mathrm{V}}(A,B)$,

ETNA
Kent State University and
Johann Radon Institute (RICAM)

precisely because z_0 is a minimizer. This can greatly reduce the number of function evaluations required by Chebfun, but as discussed earlier in Section 5, functions with large constant portions can actually cause Chebfun to terminate prematurely. As such, we generally recommend that a minimizer of f^{V} not be used for z_0.

Finally, recalling our recommendation at the end of Remark 6.7, when computing $\operatorname{sep}_\lambda^{\mathrm{V}}(A, B)$ via (8.4), we also similarly recommend refining Chebfun's result via subsequently applying local optimization. The upper bound given in (8.2) should also be computed.

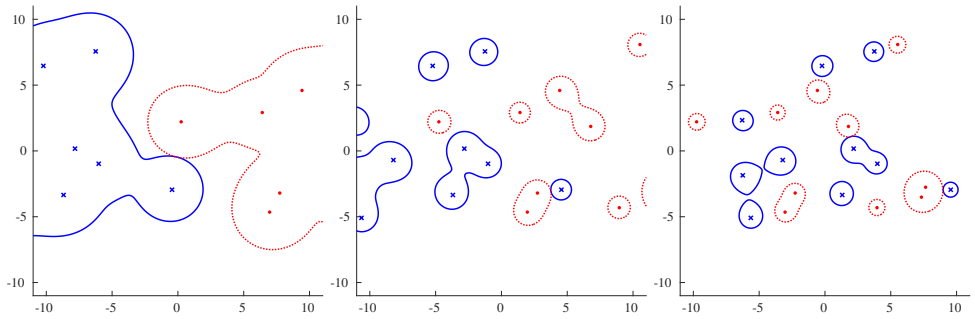
9. Numerical experiments. All experiments were done in MATLAB R2021a on a computer with two Intel Xeon Gold 6130 processors (16 cores each, 32 total) and 192GB of RAM running CentOS Linux 7. We implemented our new methods using a recent build of Chebfun (commit `119f9ad`) with `splitting` enabled and `novectorcheck`, and for simplicity, computed eigenvalues of C_θ and S_θ using `eig` in MATLAB; to account for rounding errors, the real part of any computed eigenvalue λ was set to zero if $|\operatorname{Re}\lambda| \leq 10^{-8}$. For Algorithm 7.1, we used v1.6.4 of GRANSO with `opt_tol=1e-14` to find local minimizers and used lines instead of rays for our globality certificates, as we observed that this was usually a bit faster. We forgo including any parallel processing experiments here, as we have previously validated the large benefits of using parallelism with our interpolation-based globality certificate approach in [36, Section 5.2]. The codes used to generate the results in this paper are included in the supplementary materials[5], and we plan to add robust implementations to ROSTAPACK: RObust STAbility PACKage [34].

9.1. An exploratory example. We first consider a simple example to explore the properties of our methods. We generated two different complex 10×10 matrices using `randn` and rescaled them so that the resulting matrices A and B both had spectral radii of 10. Then, for $s \in \{10, 5, 0\}$, we considered Demmel's and Varah's versions of sep-lambda for $A(s) = A - sI$ and $B(s) = B + sI$. When $s = 0$, the spectra of $A(s)$ and $B(s)$ are "centered" at the origin, but when s is increased, the centers of the two spectra, $-s$ and s, become more and more distant from each other; hence, on a macro level, increasing s generally increases the value of $\operatorname{sep}_\lambda^{\mathrm{D}}(A(s), B(s))$, and this is always true once s becomes sufficiently large. Estimates of $\operatorname{sep}_\lambda^{\mathrm{D}}(A(s), B(s))$ were computed using Algorithm 7.1, while estimates of $\operatorname{sep}_\lambda^{\mathrm{V}}(A(s), B(s))$ were computed using both of our algorithms from Section 8; for Varah's sep-lambda, the estimates for both our algorithms agreed exactly since they were obtained at an eigenvalue of $A(s)$. For Algorithm 7.1 and its extension to Varah's sep-lambda, we used $10 + 10\mathbf{i}$ as an initial point for optimization, which was chosen so that some restarts would be observed. In Figure 9.1, we show the resulting pseudospectra of $A(s)$ and $B(s)$ at the perturbation levels given by $\operatorname{sep}_\lambda^{\mathrm{D}}(A(s), B(s))$ and $\operatorname{sep}_\lambda^{\mathrm{V}}(A(s), B(s))$.

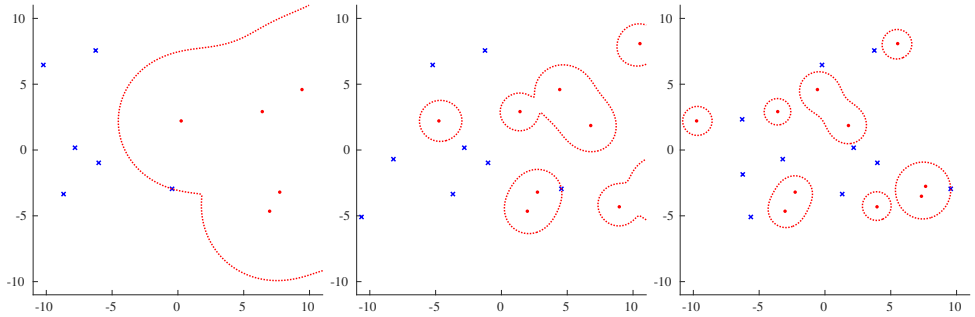
We give performance statistics of Algorithm 7.1 on our exploratory example in Table 9.1. For $s = 10$, GRANSO found a global minimizer of f^{D} from the initial point, and so only a single certificate computation was needed in this case, while two certificates were needed for the $s = 5$ and $s = 0$ instances. On both of these, the first round of optimization only found a local minimizer, and so the first certificate instead returned new points to restart optimization. But as can be seen from Table 9.1, this happened with very little effort; only 15 evaluations of d_ε were needed to find new starting points. Figure 9.2 shows that the corresponding final configurations of d_ε are all nonnegative, as they should be when $\varepsilon = \operatorname{sep}_\lambda^{\mathrm{D}}(A(s), B(s))$, per Corollary 6.6. Overall, we see that additional effort was needed to approximate d_ε as s is decreased, which is as we would expect because the behavior of d_ε generally becomes more complicated in proportion to how much the two (pseudo)spectra "intermingle", which for our test examples, is roughly controlled by s. Recalling that the search point z_0 defining d_ε is near

[5] https://etna.ricam.oeaw.ac.at/volumes/2021-2030/vol58/addition/p402.php

ETNA
Kent State University and
Johann Radon Institute (RICAM)



(a) $\Lambda_\varepsilon(A(s))$ and $\Lambda_\varepsilon(B(s))$ for $\varepsilon = \mathrm{sep}^{\mathrm{D}}_\lambda(A(s), B(s))$ and $s = 10$ (left), $s = 5$ (middle), and $s = 0$ (right).



(b) $\Lambda_{\varepsilon_1}(A(s))$ and $\Lambda_{\varepsilon_2}(B(s))$ for $\varepsilon_1 + \varepsilon_2 = \mathrm{sep}^{\mathrm{V}}_\lambda(A(s), B(s))$ and $s = 10$ (left), $s = 5$ (middle), and $s = 0$ (right).

FIG. 9.1. *For the example described in Section 9.1, pseudospectra of $A(s)$ and $B(s)$ (respectively solid and dotted contours) corresponding to Demmel's and Varah's versions of sep-lambda are shown along with the eigenvalues of $A(s)$ and $B(s)$ (respectively x's and dots) for $s \in \{10, 5, 0\}$. In the top right plot, $\Lambda_\varepsilon(A(s))$ and $\Lambda_\varepsilon(B(s))$ appear to touch at two places, but actually there is only one contact point (the one closer to the origin). In the three lower plots, Varah's sep-lambda is attained with $\varepsilon_1 = 0$.*

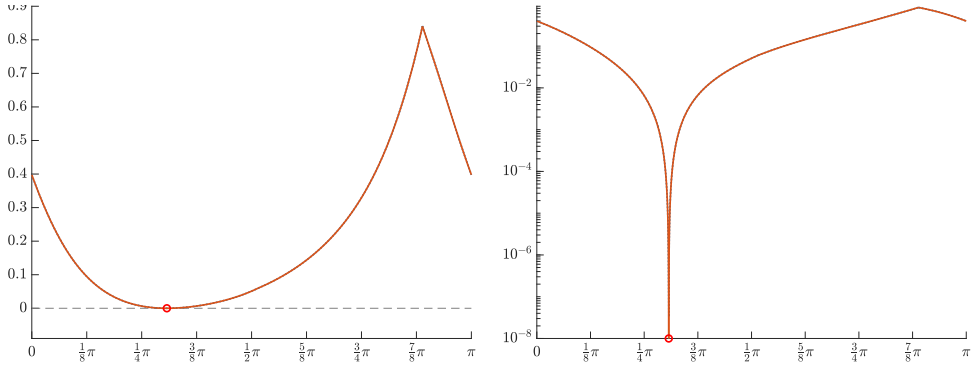
the origin for these problems, this effect can be clearly observed by looking at Figures 9.1a and 9.2 (and is also illustrated in Figure 4.1, where each quadrant of the complex plane has a different amount of pseudospectral "intermingling"). For $s = 10$, the eigenvalues of $A(s)$ and $B(s)$ are separated from each other the most, which in turn leads to the final d_ε being rather straightforward; see Figure 9.2a. However, the separation between the eigenvalues of $A(s)$ and $B(s)$ is reduced via making s smaller, and hence we see that d_ε becomes increasingly more complicated and with more discontinuities; see Figures 9.2b and 9.2c.

Performance data for our two algorithms for Varah's sep-lambda are given in Table 9.2, where we see a similar effect with respect to changing shift s. However, the main takeaway here is that, as predicted, our algorithm from Section 8.2 is indeed many times slower than our extension of Algorithm 7.1 described in Section 8.1.

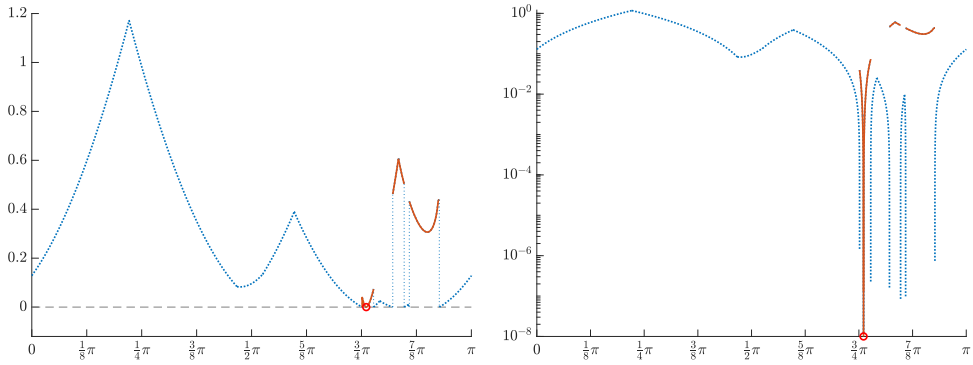
9.2. Comparing Algorithm 7.1 to the method of Gu and Overton. We now do a comparison of Algorithm 7.1 against the `seplambda` routine[6], which is Overton's MATLAB implementation of his $\mathrm{sep}^{\mathrm{D}}_\lambda(A, B)$ algorithm with Gu [24]. To do this, we generated two more examples in the manner as described in Section 9.1 but now for $m = n = 20$ and $m = n = 40$. For each, including our earlier $m = n = 10$ example, we computed $\mathrm{sep}^{\mathrm{D}}_\lambda(A(s), B(s))$ for $s = 0$ and $s = m = n$ using both our new method and `seplambda`. In order to obtain $\mathrm{sep}^{\mathrm{D}}_\lambda(A(s), B(s))$ to high precision, we set the respective tolerances for both methods to 10^{-14}.

[6] Available at https://cs.nyu.edu/faculty/overton/software/seplambda/.

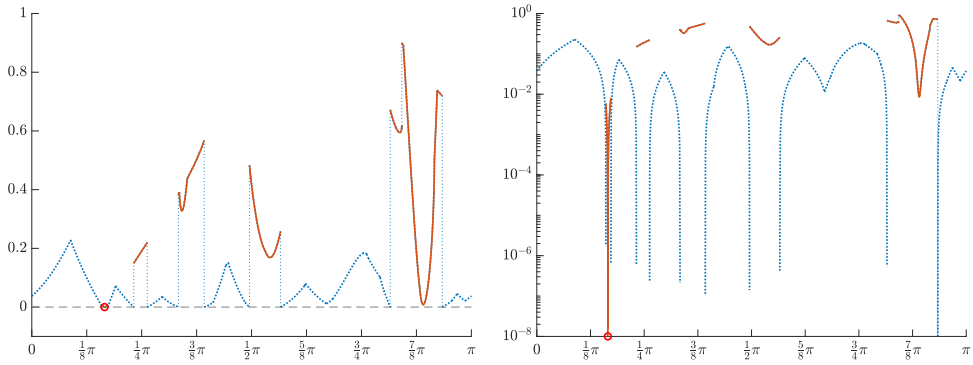
ETNA
Kent State University and
Johann Radon Institute (RICAM)



(a) $s = 10$: $d_\varepsilon(\theta)$ in linear scale (left) and in $\log_{10}$ scale (right).



(b) $s = 5$: $d_\varepsilon(\theta)$ in linear scale (left) and in $\log_{10}$ scale (right).



(c) $s = 0$: $d_\varepsilon(\theta)$ in linear scale (left) and in $\log_{10}$ scale (right).

FIG. 9.2. *Each subfigure displays the final d_ε computed by Algorithm 7.1 for the example from Section 9.1 for $s \in \{10, 5, 0\}$. The components of d_ε are plotted as follows: $a_\varepsilon + b_\varepsilon$ (dotted) and d_ε^{AB} (solid); ℓ_ε does not appear as it is never negative when $\varepsilon = \operatorname{sep}_\lambda^{\mathrm{D}}(A, B)$. The circle denotes the angle θ (with respect to z_0) associated with the best minimizer of f^{D} obtained and corresponds to the single place where $d_\varepsilon(\theta) = 0$, which is more easily seen in the $\log_{10}$ plots on the right.*

For this comparison, we always initialized the first phase of optimization for our method from the origin.

A performance overview is reported in Table 9.3. In terms of accuracy, the estimates computed by our method for $\operatorname{sep}_\lambda^{\mathrm{D}}(A(s), B(s))$ have high agreement with those computed by

ETNA
Kent State University and
Johann Radon Institute (RICAM)

TABLE 9.1
Performance data for Algorithm 7.1 for Demmel's version of sep-lambda on the example given in Section 9.1 for $s \in \{10, 5, 0\}$: "f^{D} evals." is the total number of evaluations of f^{D} during local optimization, "Certs." is the total number of certificates attempted, "All" and "Final" are the total number of evaluations of d_ε over all certificates and just the final one, respectively, and the final column is the total running time in seconds of Algorithm 7.1.

			d_ε evals.		Time (sec.)
s	f^{D} evals.	Certs.	All	Final	Alg. 7.1
10	156	1	2154	2154	4
5	150	2	8587	8572	5
0	205	2	22838	22823	13

TABLE 9.2
Performance data for our two algorithms described in Section 8 for Varah's version of sep-lambda on the example given in Section 9.1 for $s \in \{10, 5, 0\}$. For the extension of Algorithm 7.1 described in Section 8.1, "f^{V} evals." is the total number of evaluations of f^{V} during all optimization runs, "Certs." is the total number of certificates attempted, and "$d_{\varepsilon_1,\varepsilon_2}$ evals." is the total number of evaluations of $d_{\varepsilon_1,\varepsilon_2}$ over all certificates. For our algorithm described in Section 8.2, "f^{V} evals." is the total number of evaluations of f^{V}, "v evals." is the total number of evaluated of the function v defined in (8.3). Finally, we report the total running time in seconds for each method respectively under the "Section 8.1" and "Section 8.1" columns.

	Alg. from Section 8.1			Alg. from Section 8.2		Time (sec.)	
s	f^{V} evals.	Certs.	$d_{\varepsilon_1,\varepsilon_2}$ evals.	f^{V} evals.	v evals.	Sect. 8.1	Sect. 8.2
10	188	1	4494	2512417	2044	2	139
5	142	1	4583	10161448	5900	2	548
0	150	1	6859	14736950	6502	2	777

`seplambda`, though our method did return slightly better (lower) values for all the problems. On the nonshifted ($s = 0$) examples, our new method was 1.1 times faster than `seplambda` for $m = n = 10$, 18.1 times faster for $m = n = 20$, and 566.0 times faster for $m = n = 40$. Clearly, as the problems get larger, our method will be even faster relative to `seplambda`. For the shifted examples ($s = m = n$), the performance gaps are even wider: our new method was 6.2 times faster than `seplambda` for $m = n = 10$, 98.6 times faster for $m = n = 20$, and 2495.7 times faster for $m = n = 40$. The "d_ε evals." data for $s = 0$ and $s = m = n$ in Table 9.3 for these problems also indicate that d_ε is generally less complex the more the eigenvalues of A and B are separated. Meanwhile, the running times of `seplambda` were relatively unchanged by the value of s, as shifting the eigenvalues of A and B has no direct effect on its computations. In Table 9.3, we can again infer that restarts in our method, when needed, happened with relatively few evaluations of d_ε. Per [36, Section 5.2], the main work done in interpolation-based globality certificates is "embarrassingly parallel", and consequently, our method can further be accelerated by about an order of magnitude using parallel processing and substantially more if minor tweaks are made to Chebfun to make it more amenable to parallelism.

REMARK 9.1. Recalling the end of Section 2 on possibly replacing the bisection phase of Gu and Overton's with optimization-with-restarts, we now empirically validate our claim that the benefit of such a modification is indeed quite limited and diminishes as the problem dimensions increase. Besides recording the total time to run `seplambda` for each problem for Table 9.3, we also recorded the time its initialization procedure required. Then, an upper bound for the best possible speedup is simply the total time divided by the initialization time, where we idealistically assume that opimization-with-restarts has zero cost. For $m = n$ respectively equal to 10, 20, and 40, the computed ratios were approximately 3.5, 2.2, and 1.6.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

TABLE 9.3
Comparing Algorithm 7.1 and the method of Gu and Overton for Demmel's version of sep-lambda. The columns are the same as described in Table 9.1 except that we now additionally give the problem size under "$m = n$", the total running times in seconds of both methods, respectively "Alg. Algorithm 7.1" and "GO", and the relative difference between the estimates computed by both methods (Rel. Diff.), with positive indicating our method returned a better (lower) equal estimate for $\operatorname{sep}_\lambda^{\mathrm{D}}(A(s), B(s))$.

				d_ε evals.		Time (sec.)		
$m = n$	s	f^{D} evals.	Certs.	All	Final	Alg. 7.1	GO	Rel. Diff.
10	10	65	1	2154	2154	3	16	3.5×10^{-14}
10	0	75	1	23287	23287	14	15	0
20	20	97	1	4746	4746	15	1481	1.1×10^{-13}
20	0	346	3	31786	31756	78	1408	1.3×10^{-13}
40	40	128	2	5973	5910	95	237425	3.3×10^{-14}
40	0	295	3	29261	29231	407	230251	2.0×10^{-12}

Obviously, even these idealized speedups are nowhere near sufficient to overcome the very large performance gaps shown in Table 9.3 for $m = n = 20$, let alone $m = n = 40$, although such a modified version of `seplambda` would be close in performance to our method for the $m = n = 10$, $s = 10$ problem and likely pull ahead for the $m = n = 10$, $s = 0$ problem. However, if we enabled parallel processing for Algorithm 7.1, then it would again be fastest for this problem too and probably by a large margin. Finally, note that if `seplambda` were further modified by also adapting the divide-and-conquer technique of [23], it still would be significantly slower than Algorithm 7.1, except for maybe the tiniest of problems. In the context of computing the distance to uncontrollability, we compared our interpolation-based globality certificates methodology with the method of [23], which uses both optimization-with-restarts and divide-and-conquer and also does not have any expensive initialization procedure, and our approach was roughly 5 to 43 times faster depending on the dimension; see [36, Section 5.1].

9.3. Scaling performance of Algorithm 7.1. Finally, we examine the scaling performance of Algorithm 7.1 for some larger problems, which we constructed in the same fashion as before except that here we generated complex matrices A and B via `sprandn` with a density of 0.1; this change was done solely to be able to store the matrices explicitly while keeping the file sizes small for up to $m = n = 800$. For these problem sizes, it was not feasible to attempt running Gu and Overton's method, and so in Table 9.4 we only give performance data for Algorithm 7.1. The accuracy of each estimate ε for $\operatorname{sep}_\lambda^{\mathrm{D}}(A(s), B(s))$ computed by Algorithm 7.1 was verified by creating a sufficiently high resolution plot of $\Lambda_\varepsilon(A(s))$ and $\Lambda_\varepsilon(B(s))$ and inspecting it to see whether or not the interiors of the two pseudospectra overlap. This visual check suffices to confirm the high accuracy of our new method because, per Section 7.4, local minimizers discovered in every iteration of Algorithm 7.1 will be computed to high accuracy and the fact that $\varepsilon > \operatorname{sep}_\lambda^{\mathrm{D}}(A(s), B(s))$ if and only if $\operatorname{int}\Lambda_\varepsilon(A(s)) \cap \operatorname{int}\Lambda_\varepsilon(B(s)) \neq \varnothing$; hence, to assess the accuracy of a computed estimate ε, we only need to confirm whether or not Algorithm 7.1 converged to a global minimizer of f^{D} or only a local one, which is done by looking for the absence or presence, respectively, of a pseudospectral overlap. For the pair of smallest problems ($m = n = 100$), Algorithm 7.1 respectively took about 11 and 50 minutes, while on the other extreme, Algorithm 7.1 needed about 6 and 37 hours, respectively, for the two $m = n = 800$ problem instances. Again, using parallel processing can reduce these running times dramatically. Interestingly, for the intermediate sizes of $m = n = 200$ and $m = n = 400$, we actually see that Algorithm 7.1 was slightly more expensive on the

ETNA
Kent State University and
Johann Radon Institute (RICAM)

TABLE 9.4
The columns are the same as described in Table 9.3 except that here we only give running times of Algorithm 7.1 and its computed estimates of $\mathrm{sep}_\lambda^{\mathrm{D}}(A(s), B(s))$. *The running time and accuracy comparisons with Gu and Overton's method are not provided since it would have taken far too long to run their method on these larger problems.*

				d_ε evals.		Time (sec.)	
$m = n$	s	f^{D} evals.	Certs.	All	Final	Alg. 7.1	$\mathrm{sep}_\lambda^{\mathrm{D}}(A(s), B(s))$
100	100	89	1	5425	5425	704	$6.1677176880084 \times 10^{0}$
100	0	334	4	23689	23451	3045	$2.5004731832266 \times 10^{-2}$
200	200	210	2	23170	23155	11563	$6.3206868631252 \times 10^{0}$
200	0	319	2	20846	20831	10187	$4.2654521922541 \times 10^{-2}$
400	400	123	2	22456	22425	40966	$6.0394981396743 \times 10^{0}$
400	0	392	5	18237	18113	33931	$4.0258158186612 \times 10^{-2}$
800	800	153	2	3113	3098	22112	$1.0584889222355 \times 10^{1}$
800	0	383	5	21962	19436	131887	$9.9483548512835 \times 10^{-3}$

instances with nonzero s, which suggests that the spectra of $A(s)$ and $B(s)$ for these particular examples would need to be shifted even further apart in order for the complexity of d_ε to decrease. Over all the problems tested, we see that Algorithm 7.1 required at most four restarts before converging, but once again, the costs of these restarts was generally negligible with the one exception being the $m = n = 800$, $s = 0$ problem, where we can infer that the total cost of the four restarts was approximately 10% of the overall running time.

10. Concluding remarks. In this paper, we have introduced a new method to compute Demmel's version of sep-lambda that is much faster than the only previous known algorithm (due to Gu and Overton). Under our assumption that approximation of d_ε by interpolation is reliable, our method computes $\mathrm{sep}_\lambda^{\mathrm{D}}(A, B)$ to arbitrary accuracy and generally behaves like a method with cubic work complexity, albeit one with a high constant factor. Nevertheless, our new approach is so much faster that it is now possible to calculate $\mathrm{sep}_\lambda^{\mathrm{D}}(A, B)$ for moderately sized problems, e.g., for m, n in the thousands, which were simply intractable when using Gu and Overton's algorithm. We have also extended our algorithm to tackle Varah's version of sep-lambda. Although in this case global optimality cannot be guaranteed, the extension does rapidly compute locally optimal approximations which satisfy the necessary condition for global optimality. Furthermore, we have proposed a second method to actually compute $\mathrm{sep}_\lambda^{\mathrm{V}}(A, B)$, although this algorithm is significantly more expensive.

Supplementary Material. The supplementary material contains the code to generate the results in the paper:
https://etna.ricam.oeaw.ac.at/volumes/2021-2030/vol58/addition/p402.php.

To rerun all the experiments described in this paper and reproduce all the plots and table data, please see the instructions provided in `readme.txt`. Both Algorithm 7.1 for computing Demmel's sep-lambda and its modification described in Section 8.1 for approximating Varah's sep-lambda is implemented in the routine `sep-lambda\prototype\getSepLam.m`. Meanwhile, the method for computing Varah's sep-lambda described in Section 8.2 is implemented in the routine `sep-lambda\prototype\getSepLamV.m`.

Acknowledgements. The author is grateful for the referees' helpful suggestions, and particularly for the first referee's observation that the efficient pseudospectral plotting technique of [29] could be used in Algorithm 7.1 to bring down its worst-case asymptotic work complexity to cubic in n as opposed to quartic; see Section 7.2 and Section 7.6.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

REFERENCES

[1] G. ARMENTIA, J.-M. GRACIA, AND F.-E. VELASCO, *On the boundaries of strict pseudospectra*, Linear Algebra Appl., 525 (2017), pp. 20–32.
[2] P. BENNER, V. MEHRMANN, AND H. XU, *A note on the numerical solution of complex Hamiltonian and skew-Hamiltonian eigenvalue problems*, Electron. Trans. Numer. Anal., 8 (1999), pp. 115–126. https://etna.ricam.oeaw.ac.at/vol.8.1999/pp115-126.dir/pp115-126.pdf
[3] P. BENNER AND T. MITCHELL, *Faster and more accurate computation of the $\mathcal{H}_\infty$ norm via optimization*, SIAM J. Sci. Comput., 40 (2018), pp. A3609–A3635.
[4] ———, *Extended and improved criss-cross algorithms for computing the spectral value set abscissa and radius*, SIAM J. Matrix Anal. Appl., 40 (2019), pp. 1325–1352.
[5] P. BENNER, V. SIMA, AND M. VOIGT, *$\mathcal{L}_\infty$-norm computation for continuous-time descriptor systems using structured matrix pencils*, IEEE Trans. Autom. Control, 57 (2012), pp. 233–238.
[6] S. BOYD AND V. BALAKRISHNAN, *A regularity result for the singular values of a transfer matrix and a quadratically convergent algorithm for computing its $\mathbf{L}_\infty$-norm*, Systems Control Lett., 15 (1990), pp. 1–7.
[7] S. BOYD, V. BALAKRISHNAN, AND P. KABAMBA, *A bisection method for computing the H_∞ norm of a transfer matrix and related problems*, Math. Control Signals Systems, 2 (1989), pp. 207–219.
[8] S. BOYD AND L. VANDENBERGHE, *Convex Optimization*, Cambridge University Press, Cambridge, 2004.
[9] N. A. BRUINSMA AND M. STEINBUCH, *A fast algorithm to compute the H_∞-norm of a transfer function matrix*, Systems Control Lett., 14 (1990), pp. 287–293.
[10] J. V. BURKE, F. E. CURTIS, A. S. LEWIS, M. L. OVERTON, AND L. E. A. SIMÕES, *Gradient sampling methods for nonsmooth optimization*, in Numerical Nonsmooth Optimization—State of the Art Algorithms, A. M. Bagirov, M. Gaudioso, N. Karmitsa, M. M. Mäkelä, and S. Taheri, eds., Springer, Cham, 2020, pp. 201–225.
[11] J. V. BURKE, A. S. LEWIS, AND M. L. OVERTON, *Robust stability and a criss-cross algorithm for pseudospectra*, IMA J. Numer. Anal., 23 (2003), pp. 359–375.
[12] ———, *Pseudospectral components and the distance to uncontrollability*, SIAM J. Matrix Anal. Appl., 26 (2004/05), pp. 350–361.
[13] ———, *A robust gradient sampling algorithm for nonsmooth, nonconvex optimization*, SIAM J. Optim., 15 (2005), pp. 751–779.
[14] R. BYERS, *A bisection method for measuring the distance of a stable matrix to the unstable matrices*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 875–881.
[15] ———, *Detecting nearly uncontrollable pairs*, in Signal Processing, Scattering and Operator Theory, and Numerical Methods (Amsterdam, 1989), M. A. Kaashoek, J. H. van Schuppen, and A. C. M. Ran, eds., vol. 5 of Progr. Systems Control Theory, Birkhäuser, Boston, 1990, pp. 447–457.
[16] F. E. CURTIS, T. MITCHELL, AND M. L. OVERTON, *A BFGS-SQP method for nonsmooth, nonconvex, constrained optimization and its evaluation using relative minimization profiles*, Optim. Methods Softw., 32 (2017), pp. 148–181.
[17] J. W. DEMMEL, *A Numerical Analyst's Jordan Canonical Form*, Ph.D. Thesis, Center for Pure and Applied Mathematics, University of California, Berkeley, 1983.
[18] ———, *Computing stable eigendecompositions of matrices*, Linear Algebra Appl., 79 (1986), pp. 163–193.
[19] ———, *A counterexample for two conjectures about stability*, IEEE Trans. Autom. Control, 32 (1987), pp. 340–342.
[20] T. A. DRISCOLL, N. HALE, AND L. N. TREFETHEN, *Chebfun Guide*, Pafnuty Publications, Oxford, 2014.
[21] M. GAO AND M. NEUMANN, *A global minimum search algorithm for estimating the distance to uncontrollability*, Linear Algebra Appl., 188/189 (1993), pp. 305–350.
[22] M. GU, *New methods for estimating the distance to uncontrollability*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 989–1003.
[23] M. GU, E. MENG, M. L. OVERTON, J. XIA, AND J. ZHU, *Fast methods for estimating the distance to uncontrollability*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 477–502.
[24] M. GU AND M. L. OVERTON, *An algorithm to compute* Sep_λ, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 348–359.
[25] C. HE AND G. A. WATSON, *An algorithm for computing the numerical radius*, IMA J. Numer. Anal., 17 (1997), pp. 329–342.
[26] N. J. HIGHAM, *Perturbation theory and backward error for $AX - XB = C$*, BIT, 33 (1993), pp. 124–136.
[27] M. KAROW AND D. KRESSNER, *On a perturbation bound for invariant subspaces of matrices*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 599–618.
[28] A. S. LEWIS AND M. L. OVERTON, *Nonsmooth optimization via quasi-Newton methods*, Math. Program., 141 (2013), pp. 135–163.
[29] S. H. LUI, *Computation of pseudospectra by continuation*, SIAM J. Sci. Comput., 18 (1997), pp. 565–573.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

[30] V. MEHRMANN AND P. VAN DOOREN, *Optimal robustness of passive discrete-time systems*, IMA J. Math. Control Inform., 37 (2020), pp. 1248–1269.
[31] ———, *Optimal robustness of port-Hamiltonian systems*, SIAM J. Matrix Anal. Appl., 41 (2020), pp. 134–151.
[32] E. MENGI AND M. L. OVERTON, *Algorithms for the computation of the pseudospectral radius and the numerical radius of a matrix*, IMA J. Numer. Anal., 25 (2005), pp. 648–669.
[33] T. MITCHELL, *GRANSO: GRadient-based Algorithm for Non-Smooth Optimization*, Software. See also [16]. http://timmitchell.com/software/GRANSO.
[34] ———, *ROSTAPACK: RObust STAbility PACKage*, Software. http://timmitchell.com/software/ROSTAPACK.
[35] T. MITCHELL, *Computing the Kreiss constant of a matrix*, SIAM J. Matrix Anal. Appl., 41 (2020), pp. 1944–1975.
[36] ———, *Fast interpolation-based globality certificates for computing Kreiss constants and the distance to uncontrollability*, SIAM J. Matrix Anal. Appl., 42 (2021), pp. 578–607.
[37] ———, *Convergence rate analysis and improved iterations for numerical radius computation*, SIAM J. Sci. Comput., 45 (2023), pp. A753–A780.
[38] T. MITCHELL AND P. VAN DOOREN, *Root-max problems, hybrid expansion-contraction, and quadratically convergent optimization of passive systems*, to appear in SIAM J. Matrix Anal. Appl. (accepted in November, 2022).
[39] R. PACHÓN, R. B. PLATTE, AND L. N. TREFETHEN, *Piecewise-smooth chebfuns*, IMA J. Numer. Anal., 30 (2010), pp. 898–916.
[40] N. ROY, M. KAROW, S. BORA, AND G. ARMENTIA, *Approximation of pseudospectra of block triangular matrices*, Linear Algebra Appl., 623 (2021), pp. 398–419.
[41] V. SIMONCINI, *On the numerical solution of $AX - XB = C$*, BIT, 36 (1996), pp. 814–830.
[42] G. W. STEWART, *Error and perturbation bounds for subspaces associated with certain eigenvalue problems*, SIAM Rev., 15 (1973), pp. 727–764.
[43] L. N. TREFETHEN, *Personal communication*, August 2020. Recorded at https://groups.google.com/g/chebfun-users/c/9d6zzPkfwZk/m/cJJKx3C4AQAJ
[44] L. N. TREFETHEN AND M. EMBREE, *Spectra and Pseudospectra*, Princeton University Press, Princeton, 2005.
[45] J. M. VARAH, *The Computation of Bounds for the Invariant Subspaces of a General Matrix Operator*, Ph.D. Thesis, Dept. of Computer Science, Stanford University, Palo Alto, 1967.
[46] ———, *On the separation of two matrices*, SIAM J. Numer. Anal., 16 (1979), pp. 216–222.