SIAM J. MATRIX ANAL. APPL.
Vol. 38, No. 3, pp. 891–923

CRISS-CROSS TYPE ALGORITHMS FOR COMPUTING THE REAL PSEUDOSPECTRAL ABSCISSA*

DING LU† AND BART VANDEREYCKEN†

Abstract. The real ε-pseudospectrum of a real matrix A consists of the eigenvalues of all real matrices that are ε-close in spectral norm to A. The closeness is commonly measured in spectral or Frobenius norm. The real pseudospectral ε-pseudospectral abscissa, which is the largest real part of these eigenvalues for a prescribed value ε, measures the structured robust stability of A w.r.t. real perturbations. In this paper, we introduce a criss-cross type algorithm to compute the real pseudospectral ε-pseudospectral abscissa for the spectral norm. Our algorithm is based on a superset characterization of the real pseudospectrum where each criss and cross search involves solving linear eigenvalue problems and singular value optimization problems. The new algorithm is proved to be globally convergent, and observed to be locally linearly convergent. Moreover, we propose a subspace projection framework in which we combine the criss-cross algorithm with subspace projection techniques to solve large-scale problems. The subspace acceleration is proved to be locally superlinearly convergent. The robustness and efficiency of the proposed algorithms are demonstrated on numerical examples.

Key words. eigenvalue problem, real pseudospectra, spectral abscissa, subspace methods, criss-cross methods, robust stability

AMS subject classifications. 15A18, 93B35, 30E10, 65F15

DOI. 10.1137/16M107952X

1. Introduction.

The *real ε-pseudospectrum* of a matrix $A \in \mathbb{R}^{n\times n}$ is defined as

$$\Lambda_\varepsilon^{\mathbb{R}}(A) = \{\lambda \in \mathbb{C} : \lambda \in \Lambda(A+E) \text{ with } E \in \mathbb{R}^{n\times n}, \|E\| \le \varepsilon\}, \tag{1}$$

where $\Lambda(A)$ denotes the eigenvalues of a matrix A and $\|\cdot\|$ is the spectral norm. It is also known as the real unstructured spectral value set (see, e.g., [9, 13, 11]) and it can be regarded as a generalization of the more standard complex pseudospectrum (see, e.g., [23, 24]). Real pseudospectra find their use in the *robust stability* of linear dynamical systems subject to real perturbations. Since the linear system $dx/dt = Ax$ is stable when all eigenvalues A have negative real part, the perturbed system $dx/dt = (A+E)x$ is stable for real $\|E\| \le \varepsilon$ when $\Lambda_\varepsilon^{\mathbb{R}}(A)$ remains in the left-hand side of the complex plane. We are thus interested in computing the *real ε-pseudospectral abscissa* defined as

$$\alpha_\varepsilon^{\mathbb{R}}(A) = \max\{\mathrm{Re}(\lambda) \colon \lambda \in \Lambda_\varepsilon^{\mathbb{R}}(A)\} \tag{2}$$

since it provides a robust indication of the stability of A under ε-bounded real perturbations.

The notion of real pseudospectra (1) has been adopted in several works (see, e.g., [10, 9, 1, 20, 14, 8]). It has been shown in [1] that it admits the sublevel sets characterization

$$\Lambda_\varepsilon^{\mathbb{R}}(A) = \{\lambda = \alpha + \jmath\beta \in \mathbb{C} \colon \mu(\alpha,\beta) \le \varepsilon\}, \tag{3}$$

*Received by the editors June 13, 2016; accepted for publication (in revised form) by M. L. Overton May 18, 2017; published electronically August 24, 2017.
http://www.siam.org/journals/simax/38-3/M107952.html
†Section of Mathematics, University of Geneva, 1211 Geneva, Switzerland (Ding.Lu@unige.ch, Bart.Vandereycken@unige.ch).

where the μ *function*,

$$\mu(\alpha,\beta) = \max_{\gamma\in(0,1]} g(\alpha,\beta,\gamma), \tag{4}$$

is defined using

$$g(\alpha,\beta,\gamma) = \sigma_{-2}\big(G(\alpha,\beta,\gamma)\big), \qquad G(\alpha,\beta,\gamma) = \begin{bmatrix} A-\alpha I & -\beta\gamma I \\ \beta\gamma^{-1} I & A-\alpha I \end{bmatrix}. \tag{5}$$

Here, $\sigma_{-2}(\cdot)$ denotes the second smallest (in this case, the $(2n-1)$st) singular value of the corresponding matrix. The μ function, also known as the real perturbation value or real structured singular value, has been extensively studied (see, e.g., [18, 1, 13]). A well-known result in [18, p. 887] shows that the objective function $g(\alpha,\beta,\gamma)$ is *unimodal* in γ, that is, it has at most one local maximum in $\gamma\in(0,1]$. Therefore, the $\mu(\alpha,\beta)$ value can be computed by golden section search for $g(\alpha,\beta,\gamma)$ over $\gamma\in(0,1]$, or by a root finding algorithm applied to $d\sigma_{-2}/d\gamma = 0$ since $g(\alpha,\beta,\gamma)$ is also piecewise analytic for given (α,β) (see, e.g., [14]).

1.1. Existing algorithms. The real pseudospectral abscissa $\alpha_\varepsilon^{\mathbb{R}}(A)$ can be computed by the low-rank dynamical algorithms from Guglielmi and Lubich [8] and Rostami [19]. These algorithms are based on locally improving an approximation of $\alpha_\varepsilon^{\mathbb{R}}(A)$ in the direction of steepest ascent of the function $E\mapsto \operatorname{Re}\lambda(A+E)$ where the real matrix E can be restricted to be of rank two. However, their convergence is sometimes slow and they have a tendency to get stuck in locally optimal points. To the best of our knowledge, there is so far no algorithm that reliably computes $\alpha_\varepsilon^{\mathbb{R}}(A)$ to high accuracy. This is in stark contrast to the complex case where reliable algorithms like the criss-cross method from Burke, Lewis, and Overton [2] are routinely applied. In this paper, we show that a similar algorithm can in fact also be derived for the real case.

Since the boundary of $\Lambda_\varepsilon^{\mathbb{R}}(A)$ is not easily computable due to the complicated definition of the μ function (4), directly applying the criss-cross method for the computation of $\alpha_\varepsilon^{\mathbb{R}}(A)$ is, however, prohibitively expensive. Our technique to overcome this difficulty is a certain *superset characterization* of $\Lambda_\varepsilon^{\mathbb{R}}(A)$ that represents $\Lambda_\varepsilon^{\mathbb{R}}(A)$ as the intersection of infinitely many supersets but with easy to compute boundaries. Based on this characterization, we develop a superset based criss-cross algorithm to find the rightmost point. Similar to the original criss-cross method, our algorithm also relies heavily on linear eigenvalue computations. Since dense eigensolvers are used, it is intended for small and medium sized problems only (say, $n<1000$). We, therefore, propose combining it with a subspace projection scheme similar to that of [15] for the complex case. It will allow us to use the criss-cross method also for large scale matrices.

1.2. Outline. This paper is outlined as follows. In section 2, we discuss our superset characterization of $\Lambda_\varepsilon^{\mathbb{R}}(A)$. In section 3, we present the horizontal search algorithm to solve for the rightmost point of the horizontal cross-section with $\Lambda_\varepsilon^{\mathbb{R}}(A)$. The criss-cross type algorithm to solve for the rightmost point in $\Lambda_\varepsilon^{\mathbb{R}}(A)$ is finally introduced and analyzed in section 4. Numerical experiments follow in section 5.

In section 6, we introduce the reduced problems $\Lambda_\varepsilon^{\mathbb{R}}(AV,V)$ obtained by one-sided projection and study their approximation to the original problem $\Lambda_\varepsilon^{\mathbb{R}}(A)$. A subspace method for large-scale problems is then introduced and analyzed in section 7, which computes $\alpha_\varepsilon^{\mathbb{R}}(A)$ by applying the criss-cross algorithm described in section 4 to a

sequence of reduced problems. Numerical examples for the subspace acceleration are presented in section 8, and we conclude in section 9 with a summary of the contributions and open problems.

2. Superset characterization. The real pseudospectrum can in principle be directly computed from the μ function using (3). In particular, we can define the real ε-pseudospectral abscissa as

$$\alpha_\varepsilon^{\mathbb{R}}(A) = \max\big\{\alpha\colon \mu(\alpha,\beta) \le \varepsilon\big\} = \max\big\{\alpha\colon \mu(\alpha,\beta) = \varepsilon\big\}. \tag{6}$$

The second equality is due to continuity of $\mu(\alpha,\beta)$ in α as shown in the following theorem. This result is well known but we state it explicitly since we will use it quite frequently in the rest of this paper.

THEOREM 2.1. *The function $g(\alpha,\beta,\gamma)$ as defined in* (5) *is continuous on $\mathbb{R}^2\times(0,1]$, and it is smooth in a neighborhood of (α,β,γ) whenever the corresponding singular value is simple. In addition, the function $\mu(\alpha,\beta)$ as defined in* (4) *is continuous on $\mathbb{R}\times\{0\}$ and on $\mathbb{R}\times\mathbb{R}_*$, where $\mathbb{R}_* = \mathbb{R}\setminus\{0\}$.*

Proof. The continuity of $g(\alpha,\beta,\gamma) = \sigma_{-2}\big(G(\alpha,\beta,\gamma)\big)$ is a consequence of the singular value perturbation bound $\big|\sigma_{-2}(G+E) - \sigma_{-2}(G)\big| \le \|E\|_2$ (see, e.g., [12, Cor. 7.3.5]). The local smoothness of $\sigma_{-2}\big(G(\alpha,\beta,\gamma)\big)$ follows directly from smoothness of simple singular values (see, e.g., [22, Chap. IV, Theorem 2.3]). The identity

$$\mu(\alpha,0) = g(\alpha,0,\gamma) \quad \text{for all} \quad \gamma\in(0,1], \tag{7}$$

together with the continuity of g, shows the continuity of μ on the real line $\mathbb{R}\times\{0\}$. Finally, the continuity of μ on $\mathbb{R}\times\mathbb{R}_*$ is shown in [11, Theorem 4.4.48]). ∎

The idea of the original criss-cross algorithm—and to certain extent also our generalization—requires finding the intersections of horizontal and vertical lines in the complex plane with the level set $\{\alpha+\jmath\beta\colon \mu(\alpha,\beta)=\varepsilon\}$. These intersections are, however, not easily characterized since the μ function is defined as the maximum of another function involving singular values; see (4)–(5). To avoid this inconvenience, we first reformulate the expression of $\Lambda_\varepsilon^{\mathbb{R}}(A)$ into an equivalent form which will allow for cheaper computations.

From definition (4), we see that

$$\mu(\alpha,\beta)\le\varepsilon \quad\Longleftrightarrow\quad g(\alpha,\beta,\gamma)\le\varepsilon \quad \text{for all} \quad \gamma\in(0,1].$$

The left-hand side defines the real pseudospectrum. The right-hand side defines the intersection of the sets

$$\Lambda_{\varepsilon,\gamma}(A) = \{\alpha+\jmath\beta\colon\ g(\alpha,\beta,\gamma)\le\varepsilon\}, \tag{8}$$

indexed by γ over the interval $\gamma\in(0,1]$. In other words, we can write

$$\Lambda_\varepsilon^{\mathbb{R}}(A) = \bigcap_{\gamma\in(0,1]} \Lambda_{\varepsilon,\gamma}(A). \tag{9}$$

For each value of γ, the set $\Lambda_{\varepsilon,\gamma}(A)$ is a superset of $\Lambda_\varepsilon^{\mathbb{R}}(A)$ in the complex plane. We will refer to it as the *γ-superset* of $\Lambda_\varepsilon^{\mathbb{R}}(A)$. To illustrate the idea, we have plotted in

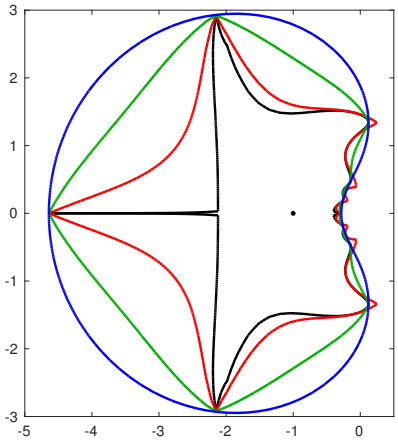

FIG. 1. *Contour plots of the real pseudospectrum of the Demmel matrix $D(5,5)$ with perturbation level $\varepsilon = 0.01$. The thick black line denotes the boundary of $\Lambda_\varepsilon^{\mathbb{R}}(A)$. The colored lines denote the boundaries of the γ-supersets $\Lambda_{\varepsilon,\gamma}(A)$ with $\gamma \in \{0.2, 0.6, 1.0\}$ (red, green, blue, from inside to outside). (Figure is in color online.)*

Figure 1 a few supersets[1] of the square Demmel matrix [4]

$$D(n,\beta) = -1 \times \begin{bmatrix} 1 & \beta & \beta^2 & \cdots & \beta^{n-1} \\ & 1 & \ddots & \ddots & \vdots \\ & & \ddots & \ddots & \beta^2 \\ & & & 1 & \beta \\ & & & & 1 \end{bmatrix}, \tag{10}$$

for $n = 5$ and $\beta = 5$. See also section 5 for more examples. We remark that the superset with $\gamma = 1$ coincides with the complex pseudospectrum; this follows from (33) which we will show later.

The benefit of (9) over (4) is that the intersections of a γ-superset $\Lambda_{\varepsilon,\gamma}(A)$ with a horizontal or vertical line turn out to be efficiently computable as the solution of certain eigenvalue problems. To this end, let us introduce the shorthand notation

$$\gamma_{\max}(\alpha,\beta) = \arg\max_{\gamma\in(0,1]} g(\alpha,\beta,\gamma). \tag{11}$$

Recall that due to the unimodality of $g(\alpha,\beta,\gamma)$ for $\gamma \in (0,1]$, we can compute $\gamma_{\max}(\alpha,\beta)$ by golden section search. While the maximum is unique, the maximizer need not be; e.g., if $\beta = 0$, then any $\gamma \in (0,1]$ is a maximizer due to (7). In that case, $\gamma_{\max}(\alpha,\beta)$ denotes any maximizer. Before we derive how to solve for the intersections, we have the following two useful properties about $\Lambda_{\varepsilon,\gamma}(A)$.

LEMMA 2.2.

(a) *For any $\gamma \in (0,1]$, the γ-superset $\Lambda_{\varepsilon,\gamma}(A)$ is bounded and closed, and symmetric w.r.t. the real axis.*

[1]The level set definition (8) can be directly used to visualize $\Lambda_{\varepsilon,\gamma}(A)$ for fixed γ. For example, by sampling $g(\alpha,\beta,\gamma)$ on a two-dimensional grid in the (α,β) plane, we have plotted the boundary of $\Lambda_{\varepsilon,\gamma}(A)$ for given values of ε using the `contour` function in MATLAB.

(b) *Each point* $\alpha_* + \jmath\beta_*$ *on the level set* $\{\alpha + \jmath\beta\colon \mu(\alpha,\beta) = \varepsilon\}$ *lies on the level set* $\{\alpha + \jmath\beta\colon g(\alpha,\beta,\gamma_*) = \varepsilon\}$ *of some* γ_**-superset that satisfies* $\gamma_* = \gamma_{\max}(\alpha_*,\beta_*)$.

Proof. (a) The level set $\Lambda_{\varepsilon,\gamma}(A)$ is closed by continuity of g (see Theorem 2.1). Boundedness follows from coercivity: For $|\alpha| \to \infty$ and/or $|\beta| \to \infty$, we have

$$\begin{aligned} g(\alpha,\beta,\gamma) &\geq \sigma_{-2}\left(\begin{bmatrix} -\alpha I & -\beta\gamma I \\ \beta\gamma^{-1} I & -\alpha I \end{bmatrix}\right) - \|A\|_2 \geq \sigma_{\min}\left(\begin{bmatrix} -\alpha & -\beta\gamma \\ \beta\gamma^{-1} & -\alpha \end{bmatrix}\right) - \|A\|_2 \\ &\geq \gamma\sigma_{\min}\left(\begin{bmatrix} -\alpha & -\beta \\ \beta & -\alpha \end{bmatrix}\right) - \|A\|_2 = \gamma\sqrt{\alpha^2+\beta^2} - \|A\|_2 \to \infty, \end{aligned}$$

where in the first inequality we used the singular value perturbation bound $\sigma_{-2}(X + E) \leq \sigma_{-2}(X) + \|E\|_2$ (see, e.g., [12, Cor. 7.3.5]), in the second inequality we exploited the block structure of the matrix which implies that only two singular values exist and each with multiplicity n, and in the third inequality we used the relation

$$\begin{bmatrix} -\alpha & -\beta\gamma \\ \beta\gamma^{-1} & -\alpha \end{bmatrix} = \begin{bmatrix} 1 & \\ & \gamma^{-1} \end{bmatrix} \begin{bmatrix} -\alpha & -\beta \\ \beta & -\alpha \end{bmatrix} \begin{bmatrix} 1 & \\ & \gamma \end{bmatrix}$$

together with $\sigma_{\min}(P^{-1}XP) = 1/\|P^{-1}X^{-1}P\|_2 \geq \sigma_{\min}(X)/(\|P\|_2\|P^{-1}\|_2)$.

The symmetry w.r.t. the real axis follows directly from the identity $g(\alpha,\beta,\gamma) = g(\alpha,-\beta,\gamma)$, since multiplying $\mathrm{diag}(I,-I)$ to both sides of $G(\alpha,\beta,\gamma)$ preserves singular values.

(b) From (11), we have $\varepsilon = \mu(\alpha_*,\beta_*) = \max_{\gamma\in(0,1]} g(\alpha_*,\beta_*,\gamma) = g(\alpha_*,\beta_*,\gamma_*)$. ∎

From the continuity of $g(\alpha,\beta,\gamma)$, it follows that the topological boundary $\partial\Lambda_{\varepsilon,\gamma}(A)$ is a subset of the level set $\{\alpha + \jmath\beta\colon g(\alpha,\beta,\gamma) = \varepsilon\}$. Geometrically, property (b) in Lemma 2.2 means then that for each point $\alpha+\jmath\beta$ in the level set $\mu(\alpha,\beta) = \varepsilon$ of $\Lambda_\varepsilon^{\mathbb{R}}(A)$, there is a γ-superset that "touches" $\Lambda_\varepsilon^{\mathbb{R}}(A)$ at $\alpha + \jmath\beta$. See again Figure 1 for a few examples. We will refer to such a superset as a *touching superset* of the point $\alpha + \jmath\beta$. We note that for a point on the real line (i.e., $\beta = 0$), any $\Lambda_{\varepsilon,\gamma}(A)$ for $\gamma \in (0,1]$ is a touching superset due to (7).

We are now ready to state how to compute the intersections. The idea is well known and is the same as in [3, 2] for complex pseudospectra. For a fixed α', Lemma 2.2 gives that the intersections $\alpha' + \jmath\beta_*$ of the level set defining $\Lambda_{\varepsilon,\gamma}(A)$ with the vertical line $\{\alpha'+\jmath\beta\colon \beta\in\mathbb{R}\}$ are obtained as the roots of $\beta \mapsto g(\alpha',\beta,\gamma) - \varepsilon$. These roots β_* have to satisfy the following eigenvalue problem.

LEMMA 2.3. *Suppose* $g(\alpha,\beta,\gamma) = \varepsilon$. *Then* $\jmath\beta$ *is a purely imaginary eigenvalue of the* $4n \times 4n$ *Hamiltonian matrix*

$$H(\alpha,\gamma) = \begin{bmatrix} 0 & A^{\mathrm{T}} - \alpha I & \gamma^{-1}\varepsilon I & 0 \\ A^{\mathrm{T}} - \alpha I & 0 & 0 & \gamma\varepsilon I \\ -\gamma^{-1}\varepsilon I & 0 & 0 & -A + \alpha I \\ 0 & -\gamma\varepsilon I & -A + \alpha I & 0 \end{bmatrix}. \tag{12}$$

Proof. Since

$$\begin{bmatrix} A - \alpha I & \jmath\beta\gamma I \\ \jmath\beta\gamma^{-1} I & A - \alpha I \end{bmatrix} = \begin{bmatrix} I & \\ & \jmath I \end{bmatrix} \begin{bmatrix} A - \alpha I & -\beta\gamma I \\ \beta\gamma^{-1} I & A - \alpha I \end{bmatrix} \begin{bmatrix} I & \\ & -\jmath I \end{bmatrix},$$

we have

$$\varepsilon = \sigma_{-2}\left(\begin{bmatrix} A - \alpha I & -\beta\gamma I \\ \beta\gamma^{-1} I & A - \alpha I \end{bmatrix}\right) = \sigma_{-2}\left(\begin{bmatrix} A - \alpha I & \jmath\beta\gamma I \\ \jmath\beta\gamma^{-1} I & A - \alpha I \end{bmatrix}\right).$$

Since ε is a singular value of $G(\alpha,\beta,\gamma)$, it is also an eigenvalue of the matrix $S = \left[\begin{smallmatrix} & G \\ G^{\mathrm{T}} & \end{smallmatrix}\right]$. Hence, $\det(S-\varepsilon I_{4n})=0$, which gives after some manipulation

$$\det\left(\begin{bmatrix} \varepsilon I & 0 & A-\alpha I & 0 \\ 0 & \varepsilon I & 0 & A-\alpha I \\ A^{\mathrm{T}}-\alpha I & 0 & \varepsilon I & 0 \\ 0 & A^{\mathrm{T}}-\alpha I & 0 & \varepsilon I \end{bmatrix} - \jmath\beta \begin{bmatrix} 0 & 0 & 0 & -\gamma I \\ 0 & 0 & -\gamma^{-1} I & 0 \\ 0 & \gamma^{-1} I & 0 & 0 \\ \gamma I & 0 & 0 & 0 \end{bmatrix}\right) = 0.$$

After left and right multiplication by $\operatorname{diag}(\gamma^{-1/2}I_n, \gamma^{1/2}I_{2n}, \gamma^{-1/2}I_n)$, permuting row and column blocks $(3,4)$, and left multiplying both matrices by the inverse of the second matrix, we obtain $\det(H(\alpha,\gamma)-(\jmath\beta)I_{4n})=0$. ∎

The above lemma states that if $\alpha'+\jmath\beta_*$ is an intersection point of $\partial\Lambda_{\varepsilon,\gamma}(A)$ and the vertical line $\{\alpha'+\jmath\beta\colon \beta\in\mathbb{R}\}$, then $\jmath\beta_*$ is an eigenvalue of $H(\alpha',\gamma)$. To compute all such intersection points we therefore first compute all the eigenvalues of $H(\alpha',\gamma)$ and then retain the imaginary ones $\jmath\beta$ for which $g(\alpha',\beta,\gamma)=\varepsilon$.

The intersections $\alpha_*+\jmath\beta'$ with the horizontal line $\{\alpha+\jmath\beta'\colon \alpha\in\mathbb{R}\}$ are obtained similarly as the roots of $\alpha\mapsto g(\alpha,\beta',\gamma)-\varepsilon$, where the roots α_* again satisfy an eigenvalue problem. We skip the proof since it is an easier version of the one for Lemma 2.3 by directly writing out the eigenvalue / singular value equivalency of the $G(\alpha,\beta,\gamma)$ matrix.

LEMMA 2.4. *Suppose $g(\alpha,\beta,\gamma)=\varepsilon$. Then α is a real eigenvalue of the $4n\times 4n$ matrix*

$$P(\beta,\gamma) = \begin{bmatrix} A^{\mathrm{T}} & \beta\gamma^{-1}I & \varepsilon I & 0 \\ -\beta\gamma I & A^{\mathrm{T}} & 0 & \varepsilon I \\ \varepsilon I & 0 & A & -\beta\gamma I \\ 0 & \varepsilon I & \beta\gamma^{-1}I & A \end{bmatrix}. \tag{13}$$

3. Horizontal search. As explained before, our algorithm consists of finding (approximate) intersections of horizontal and vertical lines with $\Lambda_\varepsilon^{\mathbb{R}}(A)$ which we will accomplish by intersecting supersets instead. While the vertical search may be performed approximately, the *horizontal search* needs to be more accurate since our final aim is an accurate computation of the rightmost point of $\Lambda_\varepsilon^{\mathbb{R}}(A)$.

To this end, fix the imaginary part β' of a horizontal line and denote its rightmost intersection with $\Lambda_\varepsilon^{\mathbb{R}}(A)$ as $\lambda'=\alpha'+\jmath\beta'$. We shall explain in this section how to compute

$$\alpha' = \max\{\alpha:\ \alpha+\jmath\beta'\in\Lambda_\varepsilon^{\mathbb{R}}(A)\} = \max\{\alpha:\ \mu(\alpha,\beta')=\varepsilon\}. \tag{14}$$

For the special case of $\beta'=0$ (i.e, intersecting with the real axis), we can simplify the computation to $\alpha'=\max\{\alpha\colon g(\alpha,0,\gamma)=\varepsilon\}$ using (7). Hence, α' is obtained as one of the eigenvalues of $P(0,\gamma)$ in Lemma 2.4 for any $\gamma\in(0,1]$. For the general case of $\beta'\neq 0$, we present a *monotonic reduction* algorithm whereby an upper bound $\alpha_0\geq\alpha'$ is monotonically reduced to α'. We shall also prove locally quadratic convergence under generic conditions.

3.1. Monotonic reduction. Thanks to the superset property, $\Lambda_\varepsilon^{\mathbb{R}}(A)\subset\Lambda_{\varepsilon,\gamma_0}(A)$ for any $\gamma_0\in(0,1]$. Hence, an upper bound $\alpha_0\geq\alpha'$ is computed from $\Lambda_{\varepsilon,\gamma_0}(A)$ as

$$\begin{aligned} \alpha' &= \max\{\alpha:\ \alpha+\jmath\beta'\in\Lambda_\varepsilon^{\mathbb{R}}(A)\} \\ &\leq \max\{\alpha:\ \alpha+\jmath\beta'\in\Lambda_{\varepsilon,\gamma_0}(A)\} = \max\{\alpha:\ g(\alpha,\beta',\gamma_0)=\varepsilon\} \ \equiv\ \alpha_0. \end{aligned} \tag{15}$$

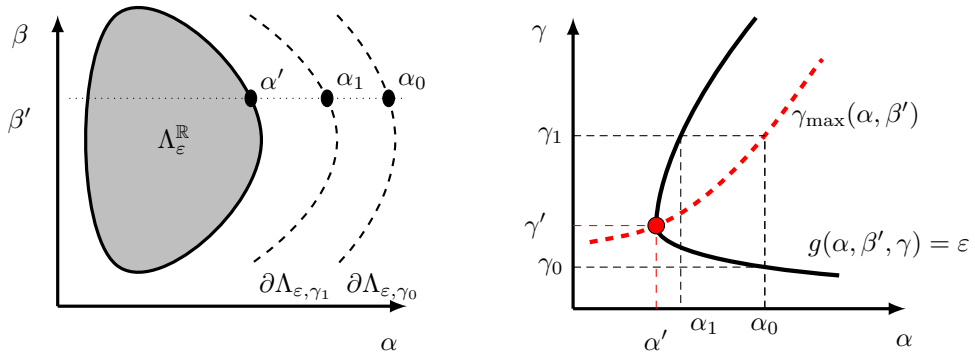

FIG. 2. *Left panel: Illustration of one step of monotonic reduction as explained in section* 3.1. *Right panel: Illustration of the horizontal search with fixed* β' *in the* (α,γ) *plane. The contour curve* $g(\alpha,\beta',\gamma)=\varepsilon$ *is marked with black solid, and the optimal parameter curve* $(\alpha,\gamma_{\max}(\alpha,\beta'))$ *is marked with red dashed curve. Starting from* γ_0*, we find* α_0 *as the horizontal intersection with the contour curve* $g(\alpha,\beta',\gamma)=\varepsilon$*. Using* α_0*, we then update* $\gamma_1=\gamma_{\max}(\alpha_0,\beta')$*. This process is updated until convergence. (Figure is in color online.)*

In turn, $\alpha_0+\jmath\beta$ is the rightmost intersection point of $\partial\Lambda_{\varepsilon,\gamma_0}$ and the horizontal line $\{\alpha+\jmath\beta'\}$ and therefore can be found by computing the algebraically largest real eigenvalue of $P(\beta',\gamma_0)$.

Unless $\alpha_0=\alpha'$ (in which case we are done with the current horizontal search), the point $\alpha_0+\jmath\beta'$ will be outside $\Lambda_\varepsilon^{\mathbb{R}}(A)$ which can be computably verified as

$$\mu(\alpha_0,\beta')>\varepsilon. \tag{16}$$

The next step is to improve α_0 to a better upper bound α_1 such that $\alpha'\le\alpha_1<\alpha_0$. The idea (see also Figure 2) is to find a γ_1-superset that does not include $\alpha_0+\jmath\beta'$, i.e., γ_1 has to satisfy

$$\alpha_0+\jmath\beta'\notin\Lambda_{\varepsilon,\gamma_1}(A)\quad\Longleftrightarrow\quad g(\alpha_0,\beta',\gamma_1)>\varepsilon.$$

Since the inclusion property of supersets gives

$$\alpha'+\jmath\beta'\in\Lambda_\varepsilon^{\mathbb{R}}(A)\subset\Lambda_{\varepsilon,\gamma_1}(A),$$

the line segment $[\alpha'+\jmath\beta',\alpha_0+\jmath\beta']$ must intersect the level set defining $\Lambda_{\varepsilon,\gamma_1}(A)$ at some point $\alpha_1+\jmath\beta'$ satisfying $\alpha'\le\alpha_1<\alpha_0$. Hence, Lemma 2.2(a) allows us to define

$$\alpha_1=\max\{\alpha<\alpha_0\colon g(\alpha,\beta',\gamma_1)=\varepsilon\}, \tag{17}$$

and Lemma 2.4 gives that α_1 can be found from the eigenvalues of $P(\beta',\gamma_1)$.

It remains to explain how to determine γ_1. By maximizing $g(\alpha_0,\beta',\gamma)$ over $\gamma\in(0,1]$, we have

$$\gamma_1=\gamma_{\max}(\alpha_0,\beta')\quad\Longleftrightarrow\quad g(\alpha_0,\beta',\gamma_1)=\max_{\gamma\in(0,1]}g(\alpha_0,\beta',\gamma)=\mu(\alpha_0,\beta')>\varepsilon, \tag{18}$$

where the last equation is due to (16). We prove below that this choice of γ_1 is the key to the local quadratic convergence of the resulting algorithm.

We can now repeatedly apply this monotonic reduction until convergence is achieved. The final algorithm is depicted in Algorithm 1. Note that the input parameter α is used as a convenient way to determine γ_0.

Algorithm 1 Horizontal search by monotonic reduction: `H-search`(β, α).

Input: Horizontal search level β, and an initial value α.

Output: Rightmost intersection α' defined as (14), and $\gamma' = \gamma_{\max}(\alpha', \beta)$.

1: Set $\gamma_0 = \gamma_{\max}(\alpha, \beta)$ and $\alpha_{-1} = \infty$.
2: **for** $i = 0, 1, \ldots$ until convergence **do**
3: Update α_i as the largest zero of $g(\cdot, \beta, \gamma_i) = \varepsilon$ that is less than α_{i-1}, computed based on the eigenvalues of $P(\beta, \gamma_i)$ in Lemma 2.4.
4: Stopping criteria: (1) α_i has no solution, then return $\alpha' = -\infty$; (2) $\mu(\alpha_i, \beta) \leq \varepsilon(1 + \texttt{tol}_h)$, then return $\alpha' = \alpha_i$.
5: Update $\gamma_{i+1} = \gamma_{\max}(\alpha_i, \beta)$.
6: **end for**

Note: $\gamma_{\max}(\alpha, \beta)$ defined in (11) can be evaluated by golden section search.

3.2. Convergence analysis. Since Algorithm 1 updates the parameters α_i and γ_i, the iteration can be visualized in the (α, γ) plane as done schematically in Figure 2. In particular, from lines 3 and 5 it follows that

$$g(\alpha_i, \beta', \gamma_i) = \varepsilon \quad \text{and} \quad g(\alpha_i, \beta', \gamma_{i+1}) = g(\alpha_i, \beta', \gamma_{\max}(\alpha_i, \beta')) = \mu(\alpha_i, \beta'), \tag{19}$$

where, like above, we have chosen the horizontal line at β'. Hence, the points (α_i, γ_i) fall on the contour curve

$$g(\alpha, \beta', \gamma) = \varepsilon,$$

while the points (α_i, γ_{i+1}) lie on the optimal parameter curve

$$(\alpha, \gamma_{\max}(\alpha, \beta')).$$

The rightmost intersection of these two curves gives the stationary point of the iteration, i.e., the optimal solution (α', γ').

From the dynamics of updating α_i and γ_i in Figure 2, we expect fast convergence if the function $\gamma_{\max}(\alpha, \beta')$ is not very sensitive in α (i.e., the red dashed line is nearly flat), since then α_1 will be extremely close to α'. Hence, using an input parameter α that is close to α' in Algorithm 1 will lead to fast convergence already from the first step. In particular, when the red line is exactly horizontal, α_1 is the exact solution α'.

We now prove that Algorithm 1 is globally convergent and that the convergence is locally quadratic under mild assumptions.

THEOREM 3.1 (global convergence). *Algorithm* 1 *generates a sequence* $\alpha_0, \alpha_1, \ldots$ *that monotonically decreases to* α' *which corresponds to a rightmost point in* $\Lambda_\varepsilon^{\mathbb{R}}(A)$ *restricted to the specified horizontal search level.*

Proof. Let us denote the horizontal line again by β'. As explained above, we need to consider only the case $\beta' \neq 0$. The monotonicity property $\alpha_{i+1} \leq \alpha_i$ is immediate from (17). Since α_i is also bounded below by α' due to the superset property, it has a limiting point $\alpha_\infty \geq \alpha'$. We will prove $\alpha_\infty = \alpha'$ by showing that

$$\alpha_\infty + \jmath\beta' \notin \Lambda_\varepsilon^{\mathbb{R}}(A) \quad \Longleftrightarrow \quad \mu(\alpha_\infty, \beta') > \varepsilon \tag{20}$$

leads to a contradiction.

Since the sequence $\{\gamma_i\} \in (0, 1]$ is bounded, it has a convergent subsequence. Without loss of generality we now assume that the whole sequence $\{\gamma_i\}$ converges to

a limiting point $\gamma_\infty \in [0,1]$. It holds that $\gamma_\infty \neq 0$ since otherwise we arrive at the contradiction

$$\varepsilon = \lim_{i\to\infty} g(\alpha_i,\beta',\gamma_i) \leq \lim_{i\to\infty} \sigma_{-2}\left(\begin{bmatrix} A-\alpha_\infty I & -\beta'\gamma_i I \\ \beta'\gamma_i^{-1} I & A-\alpha_\infty I \end{bmatrix}\right) + |\alpha_i - \alpha_\infty| \to 0,$$

where we used $\sigma_{-2}(X+E) \leq \sigma_{-2}(X) + \|E\|_2$ and exploited that $g(\alpha_\infty,\beta',\gamma_i) \to 0$ as $\gamma_i \to 0$ by [18, Lemma 5].

Since $\beta' \neq 0$ and by continuity of g and μ (see Theorem 2.1), taking the limit $i \to \infty$ in each equation in (19) gives

$$g(\alpha_\infty,\beta',\gamma_\infty) = \lim_{i\to\infty} g(\alpha_{i+1},\beta',\gamma_{i+1}) = \varepsilon,$$
$$g(\alpha_\infty,\beta',\gamma_\infty) = \lim_{i\to\infty} g(\alpha_i,\beta',\gamma_{i+1}) = \lim_{i\to\infty} \mu(\alpha_i,\beta') = \mu(\alpha_\infty,\beta').$$

The two equations above imply $\mu(\alpha_\infty,\beta') = \varepsilon$, thereby contradicting (20). ∎

In order to show the locally quadratic convergence, we require some regularity assumptions in the limit point $\alpha' + \jmath\beta'$.

DEFINITION 3.2 (regularity). *Assume $\beta' \neq 0$ and denote $\gamma' = \gamma_{\max}(\alpha',\beta')$. We call $\alpha'+\jmath\beta'$ a* regular point *for the function g if the following conditions are satisfied.*
(a) *The singular value $\sigma_{-2}(G(\alpha',\beta',\gamma')) = g(\alpha',\beta',\gamma') = \mu(\alpha',\beta')$ is simple.*
(b) *The derivatives of g satisfy*

$$g_{\gamma\gamma}(\alpha',\beta',\gamma') \neq 0 \qquad \textit{and} \qquad g_\alpha(\alpha',\beta',\gamma') \neq 0. \tag{21}$$

According to Theorem 2.1, the simplicity condition (a) guarantees that $g(\alpha,\beta',\gamma)$ is indeed a smooth function in (α,γ) in a neighborhood of (α',γ'). A similar singular value condition also appears in [2, Definition 4.4] to establish quadratic convergence of the standard criss-cross algorithm, and seems to be true generically.

Recall from the introduction that the function $\gamma \mapsto g(\alpha,\beta',\gamma)$ has only one local maximum on $(0,1]$. Hence, the first equation in (21) requires that this maximum is strict or, equivalently, that $\gamma' = \gamma_{\max}(\alpha',\beta')$ is the unique maximizer. In addition, the second equation states that $\Lambda_{\varepsilon,\gamma'}(A)$ does not have a *horizontal* tangent to the boundary at $\alpha'+\jmath\beta'$. This is a reasonable assumption given that we are looking for the rightmost intersection in the horizontal search which generically leads to a vertical tangent.

THEOREM 3.3 (local quadratic convergence). *Let $\alpha'+\jmath\beta'$ be a regular point, then Algorithm* 1 *converges at least locally quadratically to α'.*

Proof. As explained above, $\gamma' = \gamma_{\max}(\alpha',\beta')$ is the unique maximizer of $\gamma \mapsto g(\alpha',\beta',\gamma)$. Hence, using local smoothness of g, it satisfies for $\gamma = \gamma'$ the equation

$$g_\gamma(\alpha',\beta',\gamma) = 0. \tag{22}$$

Applying the implicit function theorem to this equation and using the first equation in (21), we show that the unique maximizer $\gamma_{\max}(\alpha,\beta')$ is a real analytic function in α near α'. Since $\gamma_{i+1} = \gamma_{\max}(\alpha_i,\beta')$ by line 5 of Algorithm 1 and $\alpha_i \to \alpha'$ by Theorem 3.1, we therefore obtain for all $i \geq I$ with I sufficiently large that

$$\gamma_{i+1} = \gamma' + O(|\alpha_i - \alpha'|). \tag{23}$$

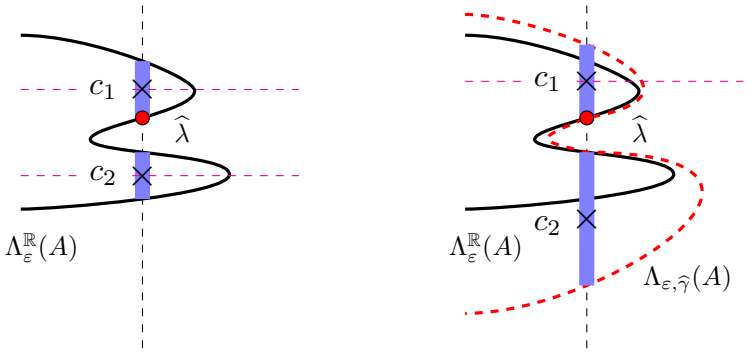

FIG. 3. *Exact criss-cross search (left panel) and criss-cross search by touching supersets (right panel). The solid black and dashed red line represent the boundary of* $\Lambda_\varepsilon^{\mathbb{R}}(A)$ *and the touching superset* $\Lambda_{\varepsilon,\widehat{\gamma}}(A)$ *at* $\widehat{\lambda}$*, respectively. The intersection point* $\widehat{\lambda}$ *is marked with a red dot. In both panels, the vertical cross-sections each contain two intervals, marked with a thick blue line, that lead to two candidate search point* c_1 *and* c_2*, marked with crosses. (Figure is in color online.)*

Line 3 of Algorithm 1 determines α_{i+1} such that $g(\alpha_{i+1}, \beta', \gamma_{i+1}) = \varepsilon$. Another application of the implicit function theorem, together with the second equation in (21), shows that there exists a real analytic function $\alpha(\gamma)$ near γ' such that

$$g(\alpha(\gamma), \beta', \gamma) = \varepsilon, \qquad \alpha(\gamma') = \alpha'.$$

By Definition 3.2(a) and continuity, $\sigma_{-2}(G(\alpha(\gamma), \beta', \gamma)) = \varepsilon$ remains simple for γ sufficiently close to γ'. We therefore get for all $i \geq I' \geq I$ with I' sufficiently large

$$\begin{aligned}\alpha_{i+1} &= \alpha' - \frac{g_\gamma(\alpha', \beta', \gamma')}{g_\alpha(\alpha', \beta', \gamma')}(\gamma_{i+1} - \gamma') + O(|\gamma_{i+1} - \gamma'|^2) \\ &= \alpha' + O(|\alpha_i - \alpha'|^2),\end{aligned}$$

where we used (22) and (23). Consequently, $\alpha_{i+1} - \alpha' = O(|\alpha_i - \alpha'|^2)$ and the local quadratic convergence is proved. ∎

4. The criss-cross algorithm by touching supersets. Having explained the horizontal search in detail, we can now present our algorithm that computes a rightmost point in $\Lambda_\varepsilon^{\mathbb{R}}(A)$. The idea is similar as in the original criss-cross method [2] for complex pseudospectra but it will make use of the superset characterization (9).

We start by performing an H-search (Algorithm 1) with fixed imaginary part $\widehat{\beta}$ and obtain the rightmost point $\widehat{\lambda} = \widehat{\alpha} + \jmath\widehat{\beta}$ that lies on the level set defined by $\Lambda_\varepsilon^{\mathbb{R}}(A)$. Next, by suitably modifying Algorithm 1 we *could* compute all the intersections of the vertical line at $\widehat{\alpha}$ with $\Lambda_\varepsilon^{\mathbb{R}}(A)$, resulting in the vertical cross-section

$$\Omega^* = \{\beta : \ \mu(\widehat{\alpha}, \beta) \leq \varepsilon\}. \tag{24}$$

See also the left panel of Figure 3 for a graphical illustration. Taking some $\beta_+ \in \Omega^*$ as a new value for a horizontal search (c_1 or c_2 in Figure 3), we then obtain an improved α_+. This method could be called the *exact criss-cross method* for $\Lambda_\varepsilon^{\mathbb{R}}(A)$ since Ω^* is the exact intersection with $\Lambda_\varepsilon^{\mathbb{R}}(A)$. Unfortunately, it is also a very expensive method since Ω^* may consist of several intersection points, each of which is relatively expensive to compute.

Instead, we replace Ω^* by the intersection of the vertical line at $\widehat{\alpha}$ with the touching superset $\Lambda_{\varepsilon,\widehat{\gamma}}(A)$ at $\widehat{\lambda}$; see the right panel of Figure 3. Hence, $\widehat{\gamma} = \gamma_{\max}(\widehat{\alpha}, \widehat{\beta})$. This results in the vertical cross-section

$$\Omega = \{\beta : \ g(\widehat{\alpha}, \beta, \widehat{\gamma}) \le \varepsilon\} = \left(\bigcup_{i=1}^{m} [\, -u_i, -\ell_i \,] \right) \bigcup \left(\bigcup_{i=1}^{m} [\, \ell_i, u_i \,] \right) \bigcup [\, -u_0, u_0 \,], \tag{25}$$

which is symmetric with respect to $\beta = 0$ due to the symmetry of the superset $\Lambda_{\varepsilon,\gamma}(A)$ (see Lemma 2.2(a)). The interval $[-u_0, u_0]$ will be deliberately set to empty if such a type of intersection does not exist, otherwise we define $\ell_0 = -u_0$. Assume now that $[\ell_0, u_0] \neq \emptyset$. The intersections $0 \le u_0 \le \ell_1 \le u_1 \le \cdots \le \ell_m \le u_m$ of the vertical line with the level set defining $\Lambda_{\varepsilon,\widehat{\gamma}}(A)$ can be computed from the eigenvalues of $H(\alpha, \gamma)$ in Lemma 2.3 using the ordering strategy in [2]. Now we use the nonnegative midpoints of the intervals

$$c_i = \tfrac{1}{2}(\ell_i + u_i) \in \Omega \quad \text{for} \quad i = 0, 1, \ldots, m,$$

as the horizontal levels for a new H-search. If this results in a larger $\widehat{\alpha}$, we repeat the procedure at the newly obtained intersection point. This method is called the *criss-cross method by touching supersets* and it is outlined in Algorithm 2. Except for lines 9 and 12, which will be explained next, the algorithm is the direct implementation of the iteration explained above.

Remark 4.1. Note that similar algorithmic ideas have also occurred in [21, 16] for computing the real stability radius, where a search based on parametrized "superfunctions" is used to solve for $\max_{x\in\mathbb{R}} \inf_{\gamma\in(0,1]} f(x, \gamma)$. Our extension to the real pseudospectral abscissa is similar since we need to find the rightmost point of a level set that is defined in a parametrized way. However, in our case $f(x, \gamma)$ would be multivalued which does not allow for the same solution strategy as in [21, 16].

Algorithm 2 Criss-cross search by touching supersets.

Input: Matrix $A \in \mathbb{R}^{n\times n}$, perturbation level ε, rightmost eigenvalue $\alpha_0 + \jmath\beta_0$ of A.
Output: A rightmost point $\lambda' = \alpha' + \jmath\beta' \in \Lambda_\varepsilon^{\mathbb{R}}(A)$.

1: *Initialize*: Apply `H-search(`β_0, α_0`)` to get α_1 and γ_1.
2: **for** $i = 1, 2, \ldots$ until convergence **do**
3: *Vertical search*: Compute intervals $\{[\ell_j, u_j]\}_{j=0}^{m_i}$ from eigenvalues of $H(\alpha_i, \gamma_i)$ in Lemma 2.3.
4: **for** $j = 0, \ldots, m_i$ **do** {*Horizontal search*}
5: Set midpoint $c_j = (\ell_j + u_j)/2$.
6: Apply `H-search(`c_j, α_i`)` to get α'_j and γ'_j.
7: **end for**
8: *Update*: $\alpha_{i+1} = \alpha'_\mu = \max_{j=1}^{m_i}\{\alpha'_j\}$ and $\beta_i = c_\mu$, $\gamma_{i+1} = \gamma'_\mu$.
9: *Safeguard*: If $\alpha_{i+1} < \alpha_i$, apply `V-reduce(`α_i, ℓ_j, u_j`)` to reduce intervals $\{[\ell_j, u_j]\}_{j=0}^{m_i}$ and repeat horizontal search (line 4) for the nonempty reduced intervals.
10: *Stopping criteria*: If $\alpha_{i+1} - \alpha_i \le$ `tol`, set $\alpha' = \alpha_{i+1}$ and $\beta' = \beta_i$, and break.
11: **end for**
12: *Global verification*: If the vertical line of $\alpha' +$ `tol`$_g$ intersects $\Lambda_\varepsilon^{\mathbb{R}}(A)$ at a point $(\alpha' +$ `tol`$_g, \beta')$, then goto line 2 and restart with $\alpha_1 = \alpha' +$ `tol`$_g$ and $\gamma_1 = \gamma_{\max}(\alpha_1, \beta')$. Otherwise, return $\lambda' = \alpha' + \jmath\beta'$.

4.1. Vertical interval reduction. We now explain line 9 in Algorithm 2 that addresses an obvious problem in the procedure explained above. When a middle point c_j obtained from the vertical cross-section Ω falls outside of $\Lambda_\varepsilon^{\mathbb{R}}(A)$—as is the case for c_2 in the right panel of Figure 3—its H-search may fail to find an intersection to the right of the current approximation α_i. If this is the case for *all* middle points c_j, we reduce the intervals in Ω such that their endpoints lie on the level set of $\Lambda_\varepsilon^{\mathbb{R}}(A)$.

The algorithm to compute these intersections is a straightforward modification of Algorithm 1 to the vertical case. In particular, let $[\ell, u]$ be one of the intervals in Ω. By the superset property and Ω^* from (24), we know that either there exists a nonempty interval

$$[\ell', u'] \subset [\ell, u] \quad \text{such that} \quad \ell', u' \in \partial\Omega^* \tag{26}$$

or $[\ell', u']$ is empty. Hence, we can monotonically reduce u to u' and increase ℓ to ℓ' using the same ideas behind the monotonic reduction algorithm from section 3.1 but applied vertically. The result is Algorithm 3, which we call *V-reduce*.

We stress that V-reduce is more expensive than the vertical search from line 3 in Algorithm 2. In the numerical experiments, we observed that it is almost never needed and at most a few times for more difficult problems.

Remark 4.2. It may happen that even after applying V-reduce, the middle point $c = (\ell' + u')/2$ still falls outside of Ω^*, which would lead to a breakdown in the criss-cross. In that case, we could again call V-reduce on each of the bisected intervals $[\ell', c]$ and $[c, u']$, and possibly continue recursively, until we reach a middle point in Ω^*. Since the breakdown very rarely happens in practice, and also for simplicity of presentation, we leave out such recursion in Algorithm 2. Instead, convergence to the globally rightmost point is guaranteed by the global verification test in line 12.

Algorithm 3 Vertical interval reduction by monotonic reduction: `V-reduce`(α, ℓ, u).

Input: Vertical search level α, and an initial interval $[\ell, u]$.
Output: Reduced interval bound ℓ' and u' satisfying (26).

1: $u_{-1} = u$, and $\gamma_0 = \gamma_{\max}(\alpha, u)$
2: **for** $i = 0, 1, \ldots$ until convergence **do**
3: Update u_i as the largest zero of $g(\alpha, \cdot, \gamma_i) = \varepsilon$ within $[\ell, u_{i-1}]$, computed by solving the eigenvalue problem of $H(\alpha, \gamma_i)$.
4: Stopping criteria: (1) u_i has no solution, then return $u' = -\infty$; (2) $\mu(\alpha, u_i) \le \varepsilon(1 + \texttt{tol}_h)$, then return $u' = u_i$.
5: Update $\gamma_{i+1} = \gamma_{\max}(\alpha, u_i)$.
6: **end for**
7: If $u' \neq -\infty$, then repeat lines 1–6 for the upper bounds of $[-u', -\ell]$ to get $-\ell'$.
8: If $u' = -\infty$ or $\ell' = \infty$, set $[\ell', u']$ to be an empty set.

4.2. Verification of global optimality. We stop the criss-cross search in line 10 when the increment in the approximate pseudospectral abscissa is small enough. Although this is a very practical stopping criterion, it does not ensure the global optimality of the computed solution. To solve this problem we add a *verification* step 12 at the end of the algorithm to check the global optimality. Then, upon convergence,

$$\alpha_\varepsilon^{\mathbb{R}}(A) - \alpha' \le \texttt{tol}_g, \tag{27}$$

where $\texttt{tol}_g$ is a user provided tolerance. The verification step is to check whether $\Lambda_\varepsilon^{\mathbb{R}}(A)$ has any intersection points with the vertical line of real part $\alpha' + \texttt{tol}_g$. This

is done by V-reduce (Algorithm 3) where the initial interval $[\ell, u]$ is determined from the smallest and largest vertical intersection, respectively, with an arbitrary superset (say $\gamma = 0.5$). If this process fails to find an intersection, then (27) is guaranteed to hold.

THEOREM 4.3 (global convergence). *The criss-cross algorithm (Algorithm 2), is globally convergent to a point $\lambda' = \alpha' + \jmath\beta'$ for which its real part α' satisfies $\alpha_\varepsilon^{\mathbb{R}}(A) - \alpha' \leq \texttt{tol}_g$, where $\texttt{tol}_g$ is a user provided tolerance parameter.*

Proof. The convergence to α' follows directly due to the monotonicity of α_i in the inner loop of Algorithm 2, the finiteness of $\alpha_\varepsilon^{\mathbb{R}}(A)$, and the fixed length improvement of α' after each outer loop iteration in Algorithm 2 with the global verification.

The condition $\alpha_\varepsilon^{\mathbb{R}}(A) - \texttt{tol}_g \leq \alpha'$ is due to the fact that Algorithm 2 is started at a rightmost eigenvalue, so $\alpha_\varepsilon^{\mathbb{R}}(A)$ cannot exceed $\alpha' + \texttt{tol}_g$ if the verification fails. Otherwise, the vertical line $\alpha' + \texttt{tol}_g$ lies between two disjoint components of $\Lambda_\varepsilon^{\mathbb{R}}(A)$ (since it does not intersect $\Lambda_\varepsilon^{\mathbb{R}}(A)$), with the rightmost point on the right side, and all the eigenvalues of A on the left (since $\max \mathrm{Re}(\Lambda(A)) = \alpha_0 \leq \alpha'$). This is impossible as the rightmost point must belong to a connected component of $\Lambda_\varepsilon^{\mathbb{R}}(A)$ that contains an eigenvalue of A; see, e.g., [11, Proposition 5.1.8]. ∎

5. Numerical experiments for criss-cross method. In this section, we report on several numerical examples that show the effectiveness of our criss-cross method in Algorithm 2, which we implemented in MATLAB version 2016a. We use the `seigtool` [14] toolbox[2] to display the real ε-pseudospectrum in the figures. The μ-function is evaluated with `fminbnd` in MATLAB (which uses golden section search accelerated with parabolic interpolation) and tolerance 1.0×10^{-6}. The overall stopping criteria of Algorithm 2 is set to $\texttt{tol} = 100\mathbf{u}$ with $\mathbf{u}$ to be the machine precision ($\approx 2.2 \times 10^{-16}$), and the global verification tolerance to $\texttt{tol}_g = 10^{-3}$. Unless otherwise stated, the H-search (Algorithm 1) and V-reduce tolerance $\texttt{tol}_h$ and $\texttt{tol}_v$ are set to $\texttt{tol}$, and due to numerical errors, we will also stop H-search and V-reduce when the iterates stop improving.

All examples were chosen so that their rightmost points do not lie on the real line. Otherwise, Algorithm 2 converges in one step (after the horizontal search). When reporting errors, we take the last iterate as the exact solution.

An illustrative example. We consider the random 6-by-6 matrix from [7, eq. (3.1)] with perturbation level $\varepsilon = 10^{-0.25}$. For illustration purposes, we take the eigenvalue $\lambda_0 = 0.1869$ of A as the starting point for the criss-cross search (if the rightmost eigenvalue $1.4950 \pm 1.2663\jmath$ is used, the algorithm converges too rapidly to the rightmost point to see anything useful). Figure 4 shows the first two criss-cross iterations of Algorithm 2. Observe that the touching supersets provide a good approximation to $\Lambda_\varepsilon^{\mathbb{R}}(A)$. The convergence of the approximation α_i to the real pseudospectral abscissa is reported in Table 1, which suggests locally linear convergence. Note that no safeguard interval reduction is necessary in this example.

Now we illustrate the convergence of the H-search (Algorithm 1) during the criss-cross iterations. In particular, we summarize the convergence history starting at $\lambda_0 = (\alpha_0, \beta_0) = (1.5, 1.1698)$ and $\lambda_0 = (0.5, 0.4)$, respectively, in Table 2. We use the starting point to evaluate the initial parameter $\gamma_0 = \gamma_{\max}(\alpha_0, \beta_0)$ in Algorithm 1, which leads to a tight upper bound α_1 in both cases, although the initial point λ_0 is still far away from the target intersection. This is due to the insensitivity of the optimal parameter $\gamma_{\max}(\alpha, \beta_0)$ w.r.t. α, as discussed in section 3.2; see also Figure 5.

[2]Obtained from http://anchp.epfl.ch/seigtool and modified to evaluate (5) directly.

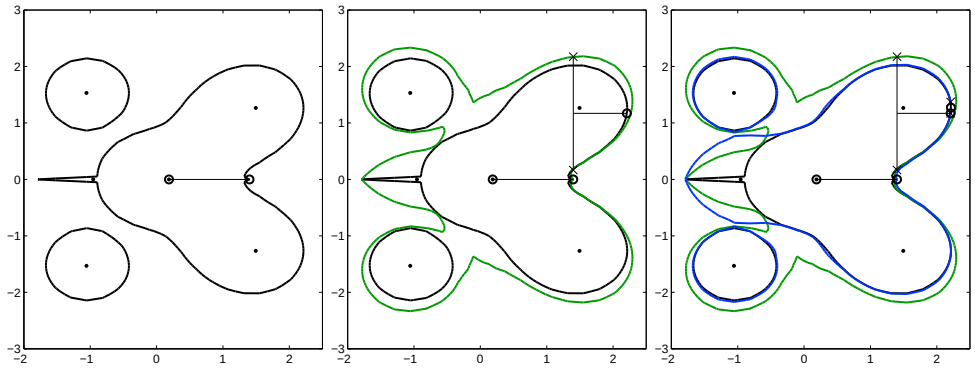

FIG. 4. *From left to right: the initial and first two iterations of Algorithm* 2 *for the matrix from the illustrative example. Eigenvalues of* A *are marked with black dots, and the boundary of* $\Lambda_\varepsilon^{\mathbb{R}}(A)$ *is the solid black line. The black circles represent the iterates and the crosses returned by the vertical searches. The solid green and blue lines are the first and second touching supersets, respectively. (Figure is in color online.)*

TABLE 1
Convergence of Algorithm 2 *for the random matrix from the illustrative example.*

i	α_i	Absolute error
0	0.186937821448622	2.02×10^{-00}
1	1.401216397097001	8.12×10^{-01}
2	2.208323300596337	5.63×10^{-03}
3	2.213936057960352	2.26×10^{-05}
4	2.213958583061330	9.67×10^{-08}
5	2.213958679389527	4.12×10^{-10}
6	2.213958679799917	1.76×10^{-13}
7	2.213958679801668	6.21×10^{-15}
8	2.213958679801672	–

In Table 2 we observe that the local convergence is at least quadratic. Especially in the first case where the horizontal search level 1.1698 is close to the global rightmost point (see Figure 4), it suggests cubic convergence. This can be explained by the fact that the optimal $\gamma_{\max}$ parameter curve is almost horizontal (see Figure 5).

More challenging examples. In this example, we consider two real pseudospectra with less smooth boundaries. In the first experiment, we consider the Demmel matrix $D(5,5)$ defined in (10). The eigenvalues are all -1, with algebraic multiplicity 5. We use perturbation level $\varepsilon = 10^{-2}$. The first three criss-cross searches of Algorithm 2, and the supersets used are depicted in Figure 6. Observe that the real pseudospectrum has a very wavy boundary on the right side. Nevertheless, the touching supersets still provide a reasonably good approximation, leading to fast convergence even from the first iteration.

As the second experiment, we consider a more challenging example, the Demmel matrix $D(3,100)$. The real pseudospectrum with perturbation level $\varepsilon = 10^{-3.2}$ has two scimitar-shaped branches pointing to the right; see Figure 7. For this problem, Algorithm 2 successfully finds the rightmost point. The convergence of the iterates is visible in Figure 7 and from the inspection of the numerical errors (not shown) it seems superlinear.

From the left panel of Figure 7, we see that the first superset (corresponding to $\gamma = 0.5$) does not provide a good approximation for the narrow branch (scimitar)

TABLE 2
Convergence of H-search (Algorithm 1) starting with $(1.5, 1.1698)$ *and* $(0.5, 0.4)$*, respectively, for the illustrative example.*

	(1.5, 1.1698)		(0.5, 0.4)	
i	α_i	Absolute error	α_i	Absolute error
0	1.500000000000000	7.08×10^{-01}	0.500000000000000	1.18×10^{-00}
1	2.209947297845896	1.62×10^{-03}	1.696034217786731	8.34×10^{-03}
2	2.208323302217018	1.62×10^{-09}	1.687689711893155	3.39×10^{-06}
3	2.208323300596337	4.44×10^{-16}	1.687686313452227	5.95×10^{-13}
4	2.208323300596337	–	1.687686313451632	–

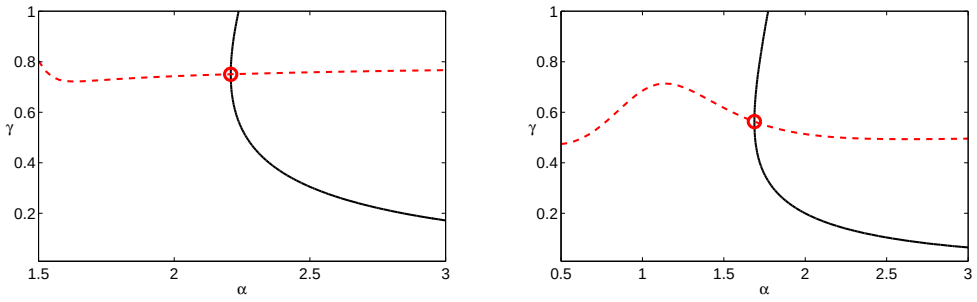

FIG. 5. *The optimal* $\gamma_{\max}(\alpha, \beta_0)$ *curve (dashed red line) for the random matrix in the illustrative example with fixed* $\beta_0 = 1.1698$ *(left) and* 0.4 *(right), respectively. The solid black line represents the contour curve of* $g(\alpha, \beta_0, \gamma) = \varepsilon$ *with* $\varepsilon = 10^{-0.25}$*, and the red circle is* (α', γ') *where* α' *is the real part of the target intersection,* $\gamma' = \gamma_{\max}(\alpha', \beta_0)$*. (Figure is in color online.)*

containing the rightmost point of $\Lambda_\varepsilon^{\mathbb{R}}(A)$. As a result, the midpoint of an interval on the vertical cross-section of the superset falls outside the pseudospectrum showing the necessity of V-reduce (Algorithm 3) to obtain global convergence. The convergence history of V-reduce for this interval is visible in Table. 3. As Algorithm 2 proceeds, further V-reduce steps are applied in iterations $i = \{22, 23, 24\}$ but they are caused by numerical rounding errors and can be omitted without destroying the convergence.

For comparison, we applied the algorithm from [19] to solve for the rightmost point.[3] The algorithm converges to $-0.14094 + 0.50607 \cdot \jmath$ for the 5×5 Demmel matrix, and -0.11074 for the 3×3 Demmel matrix. Both are only locally optimal.

6. Reduced real pseudospectrum. Our criss-cross method by touching supersets (Algorithm 2) uses dense linear algebra routines and is thus intended only for small to medium sized matrices A. In this and the next sections, we will show how to extend it for large-scale problems.

As the first step, we reduce the dimension of the original problem by subspace projection so that it becomes amenable again for dense linear algebra routines and, in particular, a slightly modified version of Algorithm 2. Given an orthonormal matrix $V \in \mathbb{R}^{n \times k}$ with $n \geq k$, the matrix pencil $A - \lambda I$ is reduced by projection on the right to $AV - \lambda V = (A - \lambda I)V$. In analogy to (3), we therefore define the *reduced real* ε*-pseudospectrum* for a rectangular matrix pair (AV, V) as

$$\Lambda_\varepsilon^{\mathbb{R}}(AV, V) = \{\lambda = \alpha + \jmath\beta \colon \widehat{\mu}(\alpha, \beta\,;\, V) \leq \varepsilon\}, \tag{28}$$

[3] The code is available from: http://users.wpi.edu/~mwu/.

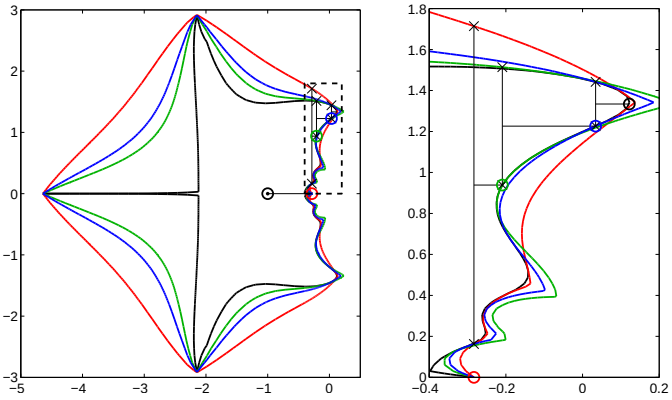

FIG. 6. *The convergence history of Algorithm* 2 *(left) to the Demmel matrix* $D(5,5)$ *with* $\varepsilon = 10^{-2}$*, and a zoomed in plot (right) of the dashed rectangle region. The boundary of* $\Lambda_\varepsilon^{\mathbb{R}}(A)$ *is marked with the solid black line. The red, green, and blue lines represent the touching supersets of the first three iterates, marked with circles. (Figure is in color online.)*

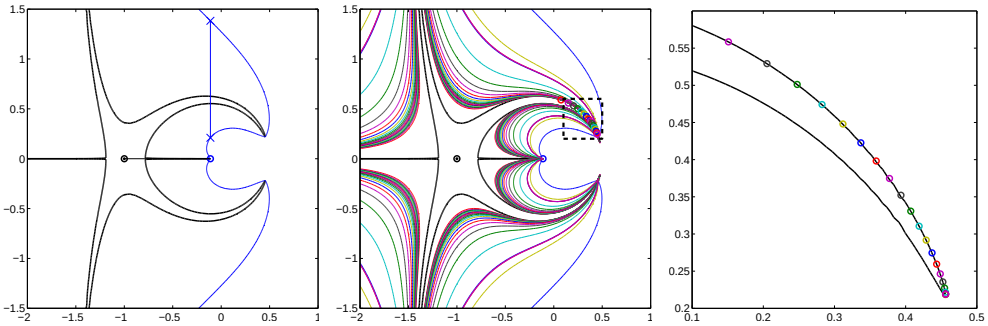

FIG. 7. *The convergence history of Algorithm* 2 *for the Demmel matrix* $D(3,100)$ *with* $\varepsilon = 10^{-3.2}$*. The boundary of* $\Lambda_\varepsilon^{\mathbb{R}}(A)$ *is marked with the solid black line. The colored contour curves represent the touching supersets of each iterate, marked with circles with the same color. Left: The first vertical search indicating the need for V-reduce. Middle: All* 25 *supersets used in the computation. Right: Zoomed in plot of dashed region. (Figure is in color online.)*

where the *reduced* $\widehat{\mu}$ *function*,

$$\widehat{\mu}(\alpha,\beta\,;\,V) = \sup_{\gamma\in(0,1]} \widehat{g}(\alpha,\beta,\gamma\,;\,V), \tag{29}$$

is defined using

$$\widehat{g}(\alpha,\beta,\gamma\,;\,V) = \sigma_{-2}\left(\begin{bmatrix} AV-\alpha V & -\beta\gamma V \\ \beta\gamma^{-1}V & AV-\alpha V \end{bmatrix}\right). \tag{30}$$

Here, $\sigma_{-2}(\cdot)$ denotes the $(2k-1)$st largest singular value since $V \in \mathbb{R}^{n\times k}$ with $n \geq k$.

6.1. Unimodality. When $V = I_n$, we have $\widehat{g}(\alpha,\beta,\gamma\,;\,I_n) = g(\alpha,\beta,\gamma)$ since (AV,V) equals (A,I_n)—and thus also $\widehat{\mu}(\alpha,\beta\,;\,I_n) = \mu(\alpha,\beta)$. Hence, the reduced functions $\widehat{g}$ and $\widehat{\mu}$ can be seen as generalizations of g and μ.

More importantly, when the (real) subspace $\mathcal{V} = \operatorname{range} V$ contains at least the real and imaginary parts of a complex eigenvector or two real eigenvectors of the matrix A, Theorem 6.1 below shows that $\widehat{g}$ also preserves the important unimodality

TABLE 3
Convergence of V-reduce (Algorithm 3) starting with interval $[\ell, u] = [0.209881, 1.382812]$ *and the horizontal level* $\alpha = -0.1107411$.

	Lower bounds		Upper bounds	
i	ℓ_i	Absolute error	u_i	Absolute error
-1	0.209883893721293	3.43×10^{-01}	1.382812400302739	7.57×10^{-01}
0	0.552585447434263	4.27×10^{-04}	0.626019636720839	3.33×10^{-05}
1	0.553011926012991	2.53×10^{-08}	0.625986364777608	1.56×10^{-10}
2	0.553011951349838	1.11×10^{-16}	0.625986364621430	7.70×10^{-14}
3	0.553011951349839	–	0.625986364621353	4.44×10^{-16}
4			0.625986364621353	–

property in γ of the original function g, as mentioned in section 1. It implies that we can simply evaluate $\widehat{\mu}(\alpha, \beta; V) = \sup_{\gamma \in (0,1]} \widehat{g}(\alpha, \beta, \gamma; V)$ for a given α, β by golden section search.

THEOREM 6.1 (reduced unimodality). *Denote the set of all eigenvectors of* A *by* $EigV(A)$ *and suppose* $\mathcal{V} \subset \mathbb{R}^n$ *satisfies*

$$\text{(31)} \qquad \begin{cases} x_0, y_0 \in \mathcal{V}, \\ \operatorname{rank}([\ x_0 \quad y_0\ ]) = 2, \\ x_0 + \jmath y_0 \in EigV(A) \end{cases} \quad or \quad \begin{cases} x_1, x_2 \in \mathcal{V}, \\ \operatorname{rank}([\ x_1 \quad x_2\ ]) = 2, \\ x_1, x_2 \in EigV(A). \end{cases}$$

Then, the following holds:

(a) *For all* $\alpha + \jmath\beta \in \mathbb{C}$ *with* $\beta \neq 0$, *the function* $\widehat{g}(\alpha, \beta, \gamma; V)$ *defined in* (30) *is either unimodal in* $\gamma \in (0, 1]$, *or otherwise it is constant in* γ.

(b) *The set* $\Lambda_\varepsilon^{\mathbb{R}}(AV, V)$ *defined in* (28) *is nonempty for each* $\varepsilon \geq 0$. *In particular, it contains the eigenvalues belonging to the eigenvectors in* (31).

Since A is real, observe that the condition $\operatorname{rank}(\left[\ x_0 \quad y_0\ \right]) = 2$ in (31) is equivalent to $x_0 \neq 0, y_0 \neq 0$.

The proof of this result is postponed to Appendix A.1 since it is quite technical. In a nutshell, it modifies the essential parts of the (long) proof of [18] that shows the unimodality of the original function g. To this end, we exploit that the matrix used in (30) to define $\widehat{g}$ can be written as a projected version of the matrix $G(\alpha, \beta, \gamma)$ from (5):

$$\text{(32)} \qquad \begin{bmatrix} AV - \alpha V & -\beta\gamma V \\ \beta\gamma^{-1} V & AV - \alpha V \end{bmatrix} = G(\alpha, \beta, \gamma) V_{[2]}, \qquad V_{[2]} = \begin{bmatrix} V & \\ & V \end{bmatrix}.$$

Such block diagonal projections preserve the 2-by-2 block structure in $G(\alpha, \beta, \gamma)$. In particular, by direct verification, we have

$$\text{(33)} \qquad \begin{bmatrix} AV - \alpha V & -\beta\gamma V \\ \beta\gamma^{-1} V & AV - \alpha V \end{bmatrix} = T_{n,\gamma} \begin{bmatrix} AV - \lambda V & \\ & AV - \overline{\lambda} V \end{bmatrix} T_{k,\gamma}^{-1},$$

with $\lambda = \alpha + \jmath\beta$, and the invertible matrices

$$T_{p,\gamma} = \begin{bmatrix} I_p & \\ & \jmath\gamma^{-1} I_p \end{bmatrix} \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} I_p & I_p \\ I_p & -I_p \end{bmatrix}, \qquad \gamma \in (0, 1].$$

This identity allows us to relate the rank of the augmented *real* matrix $G(\alpha, \beta, \gamma) V_{[2]}$ to that of the reduced *complex* matrices $AV - \lambda V$ and $AV - \overline{\lambda} V$.

6.2. Monotonicity and quadratic approximation. In this section, we investigate theoretically how well the reduced $\widehat{\mu}$ function approximates the original μ function in terms of the subspace used for the projection. We present two theorems that prove to be useful for the final subspace method to be developed in section 7. In particular, we show that the error behaves quadratically w.r.t. the distance of the subspace to a certain singular vector.

The first result states that $\widehat{\mu}$ is an upper bound for μ and monotonic w.r.t. subspace inclusion.

THEOREM 6.2 (monotonicity). *Let U and V be orthonormal matrices for the subspaces $\mathcal{U} \subset \mathcal{V}$, respectively, then*

$$\mu(\alpha,\beta) \le \widehat{\mu}(\alpha,\beta\,;\,V) \le \widehat{\mu}(\alpha,\beta\,;\,U). \tag{34}$$

In particular,

$$\Lambda_\varepsilon^{\mathbb{R}}(AU,U) \subset \Lambda_\varepsilon^{\mathbb{R}}(AV,V) \subset \Lambda_\varepsilon^{\mathbb{R}}(A). \tag{35}$$

Proof. For $V_{[2]}$ and $U_{[2]}$ defined according to (32), we have

$$\widehat{g}(\alpha,\beta,\gamma\,;\,V) = \sigma_{-2}\left(\,G(\alpha,\beta,\gamma)V_{[2]}\,\right), \quad \widehat{g}(\alpha,\beta,\gamma\,;\,U) = \sigma_{-2}\left(\,G(\alpha,\beta,\gamma)U_{[2]}\,\right).$$

By the min-max characterization of singular values (see, e.g., [12, Theorem 7.3.8]), we can derive (we omit the arguments in $G(\alpha,\beta,\gamma)$)

$$\begin{aligned}\sigma_{-2}(\,GV_{[2]}\,) = \min_{\dim \mathcal{S}=2} \max_{\substack{s\in\mathcal{S}\\ \|s\|_2=1}} \|GV_{[2]}s\|_2 &= \min_{\substack{\dim \mathcal{W}=2\\ \mathcal{W}\subset \mathrm{range}(V_{[2]})}} \max_{\substack{w\in\mathcal{W}\\ \|w\|_2=1}} \|Gw\|_2 \\ &\le \min_{\substack{\dim \mathcal{W}=2\\ \mathcal{W}\subset \mathrm{range}(U_{[2]})}} \max_{\substack{w\in\mathcal{W}\\ \|w\|_2=1}} \|Gw\|_2 = \sigma_{-2}(\,GU_{[2]}\,),\end{aligned}$$

where the third inequality is due to the inclusion $\mathrm{range}(U_{[2]}) \subset \mathrm{range}(V_{[2]})$. Therefore,

$$\widehat{g}(\alpha,\beta,\gamma\,;\,V) \le \widehat{g}(\alpha,\beta,\gamma\,;\,U) \tag{36}$$

and taking the supremum over $\gamma \in (0,1]$, we obtain the second inequality in (34). The first one is obtained by observing that $\mu(\alpha,\beta) = \widehat{\mu}(\alpha,\beta\,;\,I_n)$ and $\mathcal{V} \subset \mathrm{range}\,(I_n)$. The relation (35) is a direct consequence of an application of (34) to (28). ∎

The next result concerns the error $\widehat{\mu}(\alpha,\beta\,;\,U) - \mu(\alpha,\beta)$ for an arbitrary subspace $\mathcal{U} = \mathrm{range}(U) \subset \mathbb{R}^n$. In analogy to $\gamma_{\max}(\alpha,\beta)$ in (11), we define the reduced optimizing parameter

$$\widehat{\gamma}_{\max}(\alpha,\beta\,;\,U) = \arg\max_{\gamma\in(0,1]} \widehat{g}(\alpha,\beta,\gamma\,;\,U). \tag{37}$$

This definition will only be used if the supremum is uniquely attained on $(0,1]$.

We will express the error in terms of the following subspace distance:

$$\delta(x,U) = \min\{\|x-z\|_2 \colon z \in \mathrm{range}(U_{[2]})\}. \tag{38}$$

Here, $x \in \mathbb{R}^{2n}$ is the right singular vector that corresponds to the evaluation of the μ function in (α,β); see also its definition (4)–(5). More concretely, we define $x = [\ x_1^{\mathrm{T}}\ x_2^{\mathrm{T}}\ ]^{\mathrm{T}} \in \mathbb{R}^{2n}$ a right singular vector satisfying

$$\mu(\alpha,\beta) = \sigma_{-2}(G(\alpha,\beta,\gamma_*)) = \|G(\alpha,\beta,\gamma_*)x\|_2, \qquad \gamma_* = \gamma_{\max}(\alpha,\beta) \in (0,1]. \tag{39}$$

Observe that x depends on (α, β), but we have suppressed this in the notation.

Since the singular value above need not be simple, its right singular subspace may have dimension greater than one. For simplicity, we state and prove the approximation result for $\mathcal{V}$ only for the case of simple singular values. The notation $V = \mathrm{orth}([x_1, x_2])$ means that the columns of V are an orthonormal basis for $\mathrm{range}\{x_1, x_2\}$.

THEOREM 6.3 (quadratic approximation). *Assume $\beta \neq 0$. Suppose $\mu(\alpha, \beta) = \sigma_{-2}(G(\alpha, \beta, \gamma_*))$ with $\gamma_* = \gamma_{\max}(\alpha, \beta)$ is a simple singular value with right singular vector $x = [x_1^{\mathrm{T}}, x_2^{\mathrm{T}}]^{\mathrm{T}}$. If $\widehat{g}_{\gamma\gamma}(\alpha, \beta, \gamma_*\,; V) \neq 0$ with $V = orth([x_1, x_2])$, then for any orthonormal matrix U,*

$$\mu(\alpha, \beta) \leq \widehat{\mu}(\alpha, \beta\,; U) \leq \mu(\alpha, \beta) + O(\delta(x, U)^2). \tag{40}$$

To prove this result, we expand $\widehat{g}(\alpha, \beta, \gamma; U)$ around $\gamma = \gamma_*$ and $U = V$. While it is not difficult to show an error bound of $O(\delta)$, proving the quadratic dependence requires some technical regularity results about $\widehat{g}$. We therefore defer the proof of this theorem to Appendix A.2.

We assumed in Theorem 6.3 that $\beta \neq 0$. In the special case of $\beta = 0$, however,

$$\mu(\alpha, 0) = \sigma_{\min}(A - \alpha I) \qquad \text{and} \qquad \widehat{\mu}(\alpha, 0) = \sigma_{\min}(AV - \alpha V)$$

coincide with the standard *complex* pseudospectra of the pencil $AV - \lambda V$. A quadratic approximation property, closely related to the one from Theorem 6.3, now follows directly from Theorem 3.4 in [15].

7. Subspace methods. In this section we propose a subspace method to compute $\alpha_\varepsilon^{\mathbb{R}}(A)$ for large matrices A based on the one-sided projection that was introduced above.

7.1. Basic algorithm. Defining the *reduced real ε-pseudospectral abscissa* as

$$\alpha_\varepsilon^{\mathbb{R}}(AV, V) = \max\{\, \mathrm{Re}(\lambda) \colon \lambda \in \Lambda_\varepsilon^{\mathbb{R}}(AV, V)\,\}, \tag{41}$$

the idea is to construct a series of subspaces $\mathrm{range}(V_k) \subset \mathrm{range}(V_{k+1}) \subset \cdots$ for which the $\alpha_\varepsilon^{\mathbb{R}}(AV_k, V_k), \alpha_\varepsilon^{\mathbb{R}}(AV_{k+1}, V_{k+1}), \ldots$ provide successively better approximations of $\alpha_\varepsilon^{\mathbb{R}}(A)$. By the monotonicity result in Theorem 6.2, we have immediately that these approximations are monotonic lower bounds; that is,

$$\alpha_\varepsilon^{\mathbb{R}}(AV_k, V_k) \leq \alpha_\varepsilon^{\mathbb{R}}(AV_{k+1}, V_{k+1}) \leq \alpha_\varepsilon^{\mathbb{R}}(A). \tag{42}$$

In addition, from (29), we can define $\Lambda_\varepsilon^{\mathbb{R}}(AV_k, V_k)$ as the intersection of an infinite number of supersets, as we did for $\Lambda_\varepsilon^{\mathbb{R}}(A)$ in (9). This allows us to compute $\alpha_\varepsilon^{\mathbb{R}}(AV_k, V_k)$ using a criss-cross algorithm by touching supersets as explained in the next section.

To build the subspaces V_k we proceed as explained in Algorithm 4. In particular, for the rightmost point $\alpha_k + \jmath\beta_k$ of $\Lambda_\varepsilon^{\mathbb{R}}(AV_k, V_k)$, we expand V_k with the right singular subspace belonging to $\sigma_{-2}(G(\alpha_k, \beta_k, \gamma_{\max}(\alpha_k, \beta_k)))$. If the corresponding singular value is simple, we can prove asymptotically superlinear convergence of the subspace algorithm.

THEOREM 7.1. *Let $\lambda_* = \alpha_* + \jmath\beta_*$ be the unique rightmost point in $\Lambda_\varepsilon^{\mathbb{R}}(A) \cap \{\alpha + \jmath\beta \colon \beta > 0\}$, and let $\lambda_k = \alpha_k + \jmath\beta_k$ be produced by Algorithm 4 using the exact parameter $\gamma_k = \gamma_{\max}(\alpha_k, \beta_k)$. Suppose λ_* is a regular point (see Definition 3.2) for*

Algorithm 4 Subspace method for computing real pseudospectral abscissa.

Input: Matrix $A \in \mathbb{R}^{n\times n}$, perturbation level ε
Output: An approximation to the real pseudospectral abscissa $\alpha_\varepsilon^{\mathbb{R}}(A)$.
1: Set V_1 to be the real basis matrix of the rightmost eigenvector(s) of A.
2: **for** $k = 1, 2, \ldots$ **do**
3: Compute by Algorithm 2 the rightmost point $\alpha_k + \jmath\beta_k \in \Lambda_\varepsilon^{\mathbb{R}}(AV_k, V_k)$, and

$$\gamma_k = \begin{cases} \gamma_{\max}(\alpha_k, \beta_k) & \text{(exact version)}, \\ \widehat{\gamma}_{\max}(\alpha_k, \beta_k\,;\, V_k) & \text{(approximate version)}. \end{cases}$$

4: Compute $\varepsilon_k = \sigma_{-2}(G(\alpha_k, \beta_k, \gamma_k))$ with right singular subspace range$([X_1^{\mathrm{T}}, X_2^{\mathrm{T}}]^{\mathrm{T}})$.
5: Update the subspace $V_{k+1} = \mathrm{orth}(\,[V_k, X_1, X_2]\,)$.
6: **end for**

the functions $g(\alpha, \beta, \gamma)$ and $\widehat{g}(\alpha, \beta, \gamma; V_k)$ for $k \geq 1$. If the curvature of $\partial\Lambda_\varepsilon^{\mathbb{R}}(A)$ at λ_* is not zero, then

$$\alpha_* - \alpha_k = O(|\alpha_* - \alpha_{k-2}|^2), \tag{43}$$

provided that λ_k is linearly convergent, i.e., $|\lambda_* - \lambda_k| \leq c|\lambda_* - \lambda_{k-1}|$ for some constant $c \in [0, 1)$.

Proof. Like in the proof of Theorem 3.3, the regularity of $\lambda_* = \alpha_* + \jmath\beta_*$ and the implicit function theorem allow us to conclude that $\gamma_{\max}(\alpha, \beta)$ is smooth near (α_*, β_*). Consequently, the same holds for $\mu(\alpha, \beta) = g(\alpha, \beta, \gamma_{\max}(\alpha, \beta))$ and the chain rule together with $g_\gamma(\alpha_*, \beta_*, \gamma_{\max}(\alpha_*, \beta_*)) = 0$ gives

$$\mu_\alpha(\alpha_*, \beta_*) = g_\alpha(\alpha_*, \beta_*, \gamma_{\max}(\alpha_*, \beta_*)) \neq 0.$$

Therefore, the level set $\mu(\alpha, \beta) = \varepsilon$ near λ_* can be written as the graph of a smooth function $\alpha(\beta)$. Since λ_* is the rightmost point on the boundary $\partial\Lambda_\varepsilon^{\mathbb{R}}(A)$, we get that $\alpha(\beta)$ is maximized at β_*, that is, $\alpha'(\beta_*) = 0$. The nonzero curvature condition implies in addition that $\alpha''(\beta_*) \neq 0$. Whence, with $C \neq 0$,

$$\alpha(\beta) = \alpha_* + C|\beta - \beta_*|^2 + O(|\beta - \beta_*|^3).$$

Let $\widehat{\alpha} + \jmath\widehat{\beta}$ be any point inside the pseudospectrum $\Lambda_\varepsilon^{\mathbb{R}}(A)$ and close to λ_*. Then, for $\widehat{\alpha} + \jmath\beta$ with $\widehat{\alpha} = \alpha(\beta)$ a point on the boundary, we obtain

$$|\widehat{\beta} - \beta_*|^2 \leq |\beta - \beta_*|^2 = O(|\widehat{\alpha} - \alpha_*|).$$

From the above result and the monotonicity of the subspace method, we have for λ_{k-2} close to λ_* that $\alpha_{k-2} \leq \alpha_{k-1} \leq \alpha_*$ and

$$|\beta_{k-2} - \beta_*|^2 = O(|\alpha_{k-2} - \alpha_*|), \qquad |\beta_{k-1} - \beta_*|^2 = O(|\alpha_{k-1} - \alpha_*|). \tag{44}$$

Next, since $\mu(\alpha, \beta) = \sigma_{-2}(G(\alpha, \beta, \gamma_{\max}(\alpha, \beta))$ is smooth its singular value remains simple near λ_*. Hence, its unique right singular vector $x(\alpha, \beta)$ is also smooth. Using (44) and expanding, we get

$$\begin{aligned} x^{(k-2)} &= x(\alpha_{k-2}, \beta_{k-2}) = x(\alpha_*, \beta_*) + x_\beta(\alpha_*, \beta_*) \cdot (\beta_{k-2} - \beta_*) + O(|\alpha_{k-2} - \alpha_*|), \\ x^{(k-1)} &= x(\alpha_{k-1}, \beta_{k-1}) = x(\alpha_*, \beta_*) + x_\beta(\alpha_*, \beta_*) \cdot (\beta_{k-1} - \beta_*) + O(|\alpha_{k-1} - \alpha_*|), \end{aligned}$$

where x_β denotes the partial derivative in β.

The idea is now to use $x^{(k-2)}$ and $x^{(k-1)}$ to expand projection subspace. Following the same arguments as in [15], we take a linear combination of these two vectors to eliminate the first order term in $\beta - \beta_*$. We obtain

$$x = c_1 x^{(k-1)} + c_2 x^{(k-2)} = x(\alpha_*, \beta_*) + O(|\alpha_{k-2} - \alpha_*|),$$

where c_1 and c_2 are bounded coefficients due to the linear convergence of λ_k (see, e.g., [15, eq. (4.3)] for details). This implies $\delta(x(\alpha_*, \beta_*), V_k) = O(|\alpha_{k-2} - \alpha_*|)$, and by Theorem 6.3 we have

$$\varepsilon \le \widehat{\mu}(\alpha_*, \beta_*\,; V_k) \le \varepsilon + O(|\alpha_{k-2} - \alpha_*|^2). \tag{45}$$

The regularity of $\widehat{\mu}$ at λ_* implies the analyticity of $\widehat{\gamma}_{\max}(\alpha, \beta\,; V_k)$, and consequently, the analyticity of $\widehat{\mu}(\alpha, \beta) = \widehat{g}(\alpha, \beta, \widehat{\gamma}_{\max}(\alpha, \beta))$ close to λ_* (we omit the fixed V_k argument in $\widehat{\mu}$, $\widehat{g}$ and $\widehat{\gamma}_{\max}$). Moreover, by the chain rule

$$\frac{\partial \widehat{\mu}}{\partial \alpha}(\alpha_*, \beta_*) = \widehat{g}_\alpha(\alpha_*, \beta_*, \widehat{\gamma}_*) + \widehat{g}_\gamma(\alpha_*, \beta_*, \widehat{\gamma}_*) \frac{\partial \widehat{\gamma}_{\max}}{\partial \alpha}(\alpha_*, \beta_*) = \widehat{g}_\alpha(\alpha_*, \beta_*, \widehat{\gamma}_*) \neq 0,$$

where $\widehat{\gamma}_* = \widehat{\gamma}_{\max}(\alpha_*, \beta_*)$, and in the second equation we exploited $\widehat{g}_\gamma = 0$ due to the maximization condition, and in the last equation the regularity condition $\widehat{g}_\alpha \neq 0$ (see Definition 3.2). Therefore, the function $\rho(\alpha) := \widehat{\mu}(\alpha, \beta_*)$, with fixed β_*, is locally invertible close to α_*, with differentiable inverse function ρ^{-1}. So we have

$$\alpha' = \rho^{-1}(\varepsilon) = \rho^{-1}\big(\rho(\alpha_*) + O(|\alpha_{k-2} - \alpha_*|^2)\big) = \alpha_* + O(|\alpha_{k-2} - \alpha_*|^2),$$

where the second equality is due to $\rho(\alpha_*) = \widehat{\mu}(\alpha_*, \beta_*)$ and (45). In other words,

$$\varepsilon = \rho(\alpha') = \widehat{\mu}(\alpha', \beta_*) \qquad \text{and} \qquad |\alpha' - \alpha_*| \le O(|\alpha_{k-2} - \alpha_*|^2).$$

We complete the proof by noting that $\alpha_* \ge \alpha_k = \alpha_\varepsilon^{\mathbb{R}}(AV_k, V_k) > \alpha'$. ∎

7.2. Implementation details. The computation of the optimizing right singular vector x requires evaluating the exact maximizer $\gamma_* = \gamma_{\max}(\alpha_k, \beta_k)$, which can be expensive for large-scale problems. We therefore propose in Algorithm 4 to use instead the reduced maximizer $\gamma_k = \widehat{\gamma}_{\max}(\alpha_k, \beta_k\,; V_k)$, which is computable without much effort. When V_k approximates x well, we see from Lemma A.4(c) in the appendix that γ_k will indeed be a good approximation of the exact value γ_*.

The initial subspace V_1 is determined to satisfy (31), which ensures that the next subspaces V_k share the same property. In particular, let $v_0 = x_0 + \mathrm{i} y_0$ be a complex eigenvector belonging to a rightmost eigenvalue λ_0 of A. We then take V_1 from the truncated QR decomposition $\begin{bmatrix} x_0 & y_0 \end{bmatrix} = V_1 R_1$. If $v_0 = x_0$ is real, we add another eigenvector of A. Due to Theorem 6.3(b), this construction guarantees that the reduced real pseudospectra $\Lambda_\varepsilon^{\mathbb{R}}(AV_k, V_k)$ are never empty since $\lambda_0 \in \Lambda_\varepsilon^{\mathbb{R}}(AV_1, V_1) \subset \Lambda_\varepsilon^{\mathbb{R}}(AV_k, V_k)$.

To compute the rightmost point in $\Lambda_\varepsilon^{\mathbb{R}}(AV_k, V_k)$, we first reduce its dimension when $2k \le n$. In particular, from the truncated QR decomposition $\begin{bmatrix} V_k & AV_k \end{bmatrix} = \widehat{Q} \begin{bmatrix} \widehat{B} & \widehat{A} \end{bmatrix}$, we obtain the pencil

$$\widehat{A} - z\widehat{B} = \begin{bmatrix} \widehat{A}_1 \\ \widehat{A}_2 \end{bmatrix} - z \begin{bmatrix} \widehat{B}_1 \\ 0 \end{bmatrix} = \begin{bmatrix} \square \\ \diagdown \end{bmatrix} - z \begin{bmatrix} \diagdown \\ \end{bmatrix} \in \mathbb{R}^{2k \times k}.$$

We then have $\Lambda_\varepsilon^{\mathbb{R}}(AV_k, V_k) = \Lambda_\varepsilon^{\mathbb{R}}(\widehat{A}, \widehat{B})$, since the QR decomposition from above gives

$$\sigma_{-2}\left(\begin{bmatrix} AV_k - \alpha V_k & -\beta\gamma V_k \\ \beta\gamma^{-1}V_k & AV_k - \alpha V_k \end{bmatrix}\right) = \sigma_{-2}\left(\begin{bmatrix} \widehat{A} - \alpha\widehat{B} & -\beta\gamma\widehat{B} \\ \beta\gamma^{-1}\widehat{B} & \widehat{A} - \alpha\widehat{B} \end{bmatrix}\right). \tag{46}$$

Next, we formulate a criss-cross algorithm by touching supersets, similar to Algorithm 2 explained in section 4. Most of the algorithm stays the same after we use the following generalization of Lemmas 2.3–2.4 for determining the horizontal and vertical intersections of the supersets for $\Lambda_\varepsilon^{\mathbb{R}}(\widehat{A}, \widehat{B})$. The proof is similar to the ones for Lemmas 2.3–2.4, and can be found in Appendix A.3.

LEMMA 7.2. *Let α, β satisfy $\widehat{g}(\alpha, \beta, \gamma\,; V_k) = \varepsilon$ defined in* (29)*. Then, with $\widehat{A}$ and $\widehat{B}$ as defined above, the following holds:*

(a) *α is a real generalized eigenvalue satisfying $\widehat{P}(\beta, \gamma)x = \alpha\widehat{G}x$ with*

$$\widehat{P}(\beta,\gamma) = \begin{bmatrix} -\varepsilon I_{2k} & 0 & \widehat{A} & -\beta\gamma\widehat{B} \\ 0 & -\varepsilon I_{2k} & \beta\gamma^{-1}\widehat{B} & \widehat{A} \\ \widehat{A}^{\mathrm{T}} & \beta\gamma^{-1}\widehat{B}^{\mathrm{T}} & -\varepsilon I_k & 0 \\ -\beta\gamma\widehat{B}^{\mathrm{T}} & \widehat{A}^{\mathrm{T}} & 0 & -\varepsilon I_k \end{bmatrix}, \quad \widehat{G} = \begin{bmatrix} 0 & 0 & \widehat{B} & 0 \\ 0 & 0 & 0 & \widehat{B} \\ \widehat{B}^{\mathrm{T}} & 0 & 0 & 0 \\ 0 & \widehat{B}^{\mathrm{T}} & 0 & 0 \end{bmatrix}; \tag{47}$$

(b) *$\jmath\beta$ is an imaginary generalized eigenvalue satisfying $\widehat{H}(\alpha, \gamma)x = \jmath\beta\widehat{G}x$ with*

$$\widehat{H}(\alpha,\gamma) = \begin{bmatrix} -\gamma^{-1}\varepsilon I_{2k} & 0 & 0 & -\widehat{A} + \alpha\widehat{B} \\ 0 & -\gamma\varepsilon I_{2k} & -\widehat{A} + \alpha\widehat{B} & 0 \\ 0 & \widehat{A}^{\mathrm{T}} - \alpha\widehat{B}^{\mathrm{T}} & \gamma^{-1}\varepsilon I_k & 0 \\ \widehat{A}^{\mathrm{T}} - \alpha V^{\mathrm{T}} & 0 & 0 & \gamma\varepsilon I_k \end{bmatrix}. \tag{48}$$

In addition, we start the criss-cross search in the rightmost point of

$$\{\alpha + \jmath\beta \in \Lambda(\widehat{A}_1, \widehat{B}_1)\colon \widehat{\mu}(\alpha, \beta) \le \varepsilon\} \subset \Lambda_\varepsilon^{\mathbb{R}}(\widehat{A}, \widehat{B}), \tag{49}$$

which involves a square generalized eigenvalue problem. This strategy is also proposed in [15] for the complex pseudospectrum and usually leads to an initial point in the rightmost component of $\Lambda_\varepsilon^{\mathbb{R}}(\widehat{A}, \widehat{B})$.

7.3. Extensions. Similar to the algorithms from [15], it is relatively straightforward to incorporate the low-rank dynamic algorithms from [8, 19] in the subspace expansion phase of Algorithm 4. For example, [19] constructs a series of rank 2 (or rank 1) perturbation matrices

$$\Delta_k \in \mathbb{R}^{n\times n} \text{ with } \|\Delta_k\|_2{=}1 \quad \text{for } k = 1, 2, \ldots,$$

such that the rightmost eigenvalue $\lambda_k \in \Lambda(A + \varepsilon\Delta_k)$ is moved to a rightmost point of $\Lambda_\varepsilon^{\mathbb{R}}(A)$. In particular, $\Delta_{k+1} = L_k R_k^{\mathrm{T}}$ where $L_k\Sigma_k R_k^{\mathrm{T}} = \mathrm{Re}(z_k)\,\mathrm{Re}(w_k)^{\mathrm{T}} + \mathrm{Im}(z_k)\,\mathrm{Im}(w_k)^{\mathrm{T}}$ is the truncated singular value decomposition, and w_k and z_k are the right and left eigenvectors of $A + \varepsilon\Delta_k$ for the eigenvalue λ_k.

During the expansion, the idea is to extend the subspace with the right eigenvector w_k from above. The result is Algorithm 5 where, given the current estimate $\alpha_k + \jmath\beta_k$ of the rightmost point in $\Lambda_\varepsilon^{\mathbb{R}}(A)$, we first determine the perturbation matrix Δ_k such that

$$\alpha_k + \jmath\beta_k \in \Lambda(A + \widehat{\varepsilon}\Delta_k) \quad \text{and} \quad \widehat{\varepsilon} = \mu(\alpha_k, \beta_k) \le \varepsilon, \tag{50}$$

and then we compute w_k and add it to V_k. From [18, Construction of a worst Δ, p. 886], one can show that the formula in line 4b indeed satisfies (50) assuming that the singular value is simple.

As we will show in the numerical experiments, Algorithm 5 has a potential advantage over Algorithm 4 for achieving faster initial convergence. However, it requires solving an additional eigenvalue problem in each iteration, which is in general more expensive compared to the basic subspace method of Algorithm 4.

Algorithm 5 Subspace projection method (low-rank dynamics).

Replace lines 4 *and* 5 *in Algorithm* 4 *by the following lines.*

4a: Compute $\varepsilon_k = \sigma_{-2}(G(\alpha_k, \beta_k, \gamma_k))$ with left / right singular vectors $[y_1^{\mathrm{T}}, y_2^{\mathrm{T}}]^{\mathrm{T}}$ and $[x_1^{\mathrm{T}}, x_2^{\mathrm{T}}]^{\mathrm{T}}$.

4b: Set $\Delta_k = L_k R_k^{\mathrm{T}}$ with $L_k \Sigma_k R_k^{\mathrm{T}} = \mathrm{svd}([y_1, y_2][x_1, x_2]^{\mathrm{T}})$ being the truncated SVD.

4c: Compute the rightmost eigenvalue λ_k and eigenvector w_k of $A + \varepsilon\Delta_k$.

5: Update the subspace $V_{k+1} = \mathrm{orth}([V_k, \mathrm{Re}(w_k), \mathrm{Im}(w_k)])$.

8. Numerical experiments for subspace projection. In this section we report on numerical experiments for the subspace methods, Algorithms 4 and 5, where in both cases γ_k is computed as the approximate version. We compare with Rostami's algorithm[3] from [19], which is a method for large-scale problems based on low-rank dynamics. Its performance is comparable to the algorithm in [8] for the complex pseudospectral abscissa.

A small illustrative example. We take A to be the 100×100 Grcar matrix [5] and choose perturbation level $\varepsilon = 0.3$. The rightmost point in $\Lambda_\varepsilon^{\mathbb{R}}(A)$, as computed by Algorithm 2, lies on the real axis and satisfies $\lambda = 3.242289581449518$. Since this is a small-scale example, we use `eig` and `svd` in MATLAB to solve the necessary singular value and eigenvalue problems. The convergence of the subspace methods is reported in Table 4. We observe that their local convergence behavior is similar, but the method using low-rank dynamics converges faster at the beginning. The convergence seems superlinear, possibly quadratic.

The convergence of Algorithm 5 is also depicted in Figure 8. In the first two plots, as the subspace expands, a disconnected component appears as a narrow strip in the reduced pseudospectrum, and the rightmost eigenvalue jumps into this component. This jumping behavior was also observed in [15] for complex pseudospectra. It has the benefit of accelerating the global convergence and reducing the chance getting stuck in locally optimal points; see also the examples below. Rostami's algorithm, for example, converges into 27 iterations to the locally optimal solution $\lambda \approx 3.22 + 0.295\jmath$.

As the second experiment, we take the matrix $-A$ with A the same Grcar matrix from above and we choose $\varepsilon = 0.2$. For this problem, $\alpha_\varepsilon^{\mathbb{R}}(A) = 0.808921287786494$ (computed by Algorithm 2), and the rightmost point is not on the real axis. The convergence history of the subspace methods is reported in Table 5 and Figure 9. We observe similar performance as in the first experiment.

Large-scale problems. We now consider the following $n \times n$ matrices: `pde` ($n = 2961$), `rdrusselator` ($n = 3200$), `tolsa` ($n = 4000$), and `tubular` ($n = 1000$). These matrices are available from `EigTool` [25] and are used in [8, 19, 6] when computing the real pseudospectral abscissa and the real stability radius.

We test Algorithm 4 and Rostami's algorithm [19]. Like in [19], the stopping criterion of the algorithms is

$$|\alpha_{k+1} - \alpha_k| \leq 10^{-8} \max(1, |\alpha_k|),$$

TABLE 4
Absolute error of the approximate real pseudospectral abscissa α_k computed by Algorithms 4 and 5 applied to the Grcar matrix A with $\varepsilon = 0.3$.

	Algorithm 5 (EIG expansion; 0.37 sec)		Algorithm 4 (SVD expansion; 0.88 sec)	
k	α_k	Abs error	α_k	Abs error
1	1.984474363959230	1.25×10^{-00}	1.984474363959230	1.25×10^{-00}
2	3.175882920816043	6.64×10^{-02}	2.392609752195605	8.49×10^{-01}
3	3.242082760262466	2.06×10^{-04}	2.886063103282154	3.56×10^{-01}
4	3.242289581449020	4.98×10^{-13}	3.088313024442922	1.53×10^{-01}
5	3.242289581449493	2.57×10^{-14}	3.144674015336815	9.76×10^{-02}
6			3.191944599525125	5.03×10^{-02}
7			3.232519204607793	9.77×10^{-03}
8			3.242289574610471	6.83×10^{-09}
9			3.242289581449495	2.35×10^{-14}

TABLE 5
Same setting as in Table 4 but with $\varepsilon = 0.2$ and $-A$.

	Algorithm 5 (EIG expansion; 0.58 sec)		Algorithm 4 (SVD expansion; 0.77 sec)	
k	α_k	Abs error	α_k	Abs error
1	0.138765495369257	6.70×10^{-01}	0.138765495369257	6.70×10^{-01}
2	0.787056279869948	2.18×10^{-02}	0.510430912109342	2.98×10^{-01}
3	0.807163564395887	1.75×10^{-03}	0.764383520621478	4.45×10^{-02}
4	0.808787650422084	1.33×10^{-04}	0.801683260139172	7.23×10^{-03}
5	0.808921188593893	9.91×10^{-07}	0.807732143164990	1.18×10^{-03}
6	0.808921287786466	2.75×10^{-13}	0.808909603550515	1.16×10^{-05}
7			0.808921287748526	3.79×10^{-11}
8			0.808921287786473	2.08×10^{-14}

and the maximun iteration number is 100. In both algorithms, we use `eigs` and `svds` in MATLAB to compute the necessary eigenvalue and singular value problems. Both routines are suitable for large-scale problems and rely on the implicitly restarted Arnoldi method [17] to compute only a few eigenvectors / singular vectors. As suggested in the software provided by Rostami (also indicated in [19]), we call `eigs` with option `''lr''` and number of required eigenvalues $k = 20$ (and $k = 150$ for `tolosa` in order to obtain convergence) to compute the rightmost eigenvalue. For `svds`, we use the default parameter provided by MATLAB for smallest singular values.

The computation for $\varepsilon = 0.01$ is summarized in Table 6. Both algorithms have a comparable number of iterations, but the subspace method is faster. The difference in time is mainly attributed to the more expensive computation of the rightmost eigenvalue, needed in Rostami's algorithm. On the other hand, Algorithm 4 spends most of its time computing the second smallest singular value, which is considerably cheaper. We remark that for the same reason, Algorithm 4 is faster than Algorithm 5 for this problem even if the latter takes fewer iterations to converge.

We can also observe that both algorithms converge to the same solution, except for `tolosa` where Algorithm 4 obtained a slightly better solution corresponding to the rightmost point

$$\lambda_r = \alpha_r + \jmath\beta_r = -0.1341888118283331 + 156.0048827992994 \cdot \jmath.$$

Since the corresponding perturbation value verifies $\mu(\alpha_r, \beta_r) = 0.00999996 < \varepsilon = 0.01$, λ_r is inside but very close to the boundary of $\Lambda_\varepsilon^{\mathbb{R}}(A)$. Hence, α_r is indeed a valid

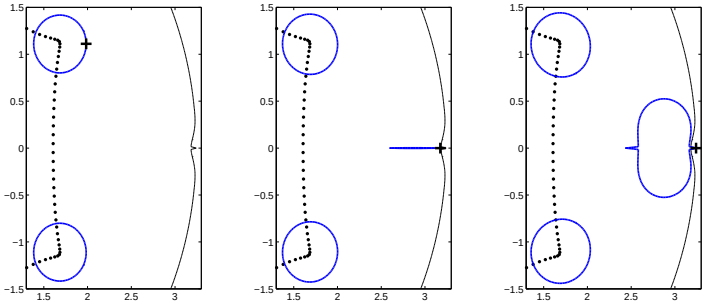

FIG. 8. *Convergence of Algorithm* 5 *applied to the Grcar matrix with* $\varepsilon = 0.3$. *The black dots represent the eigenvalues of* A. *The thin black line represents the boundary of* $\Lambda_\varepsilon^{\mathbb{R}}(A)$. *The thick blue lines represent the boundary of* $\Lambda_\varepsilon^{\mathbb{R}}(AV_k, V_k)$ *with the computed rightmost point marked with* +. *The iteration numbers are* $k = 1, 2, 4$, *from left to right. (Figure is in color online.)*

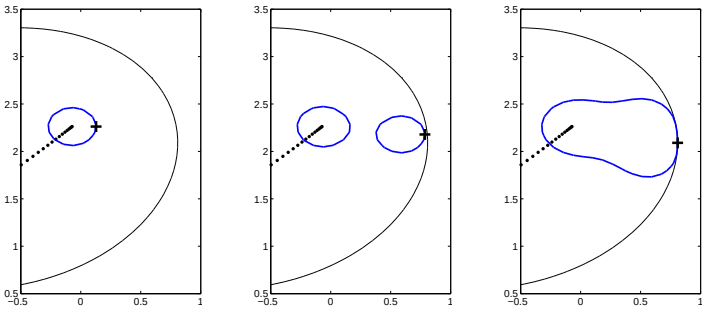

FIG. 9. *Same setting as in Figure* 8 *but with* $\varepsilon = 0.2$ *and* $-A$.

lower bound for the true real pseudospectral abscissa.

Now we repeat the experiment for $\varepsilon = 0.1$. The computed results are reported in Table 7. The subspace method is always faster and Rostami's algorithm reached the maximum number of iterations allowed for the `tolosa` problem.

We also observe that, except for `tubular`, both algorithms produce quite different solutions. The subspace method usually provides a larger approximation for $\alpha_\varepsilon^{\mathbb{R}}(A)$. In addition, Rostami's algorithm is very sensitive to the initial guess in case of `tolosa`. To verify that we have obtained a point on the boundary of $\Lambda_\varepsilon^{\mathbb{R}}(A)$, we list in Table 8 the corresponding rightmost points and values for μ. We see that for the subspace algorithm, these points are indeed on the boundary of $\Lambda_\varepsilon^{\mathbb{R}}(A)$, whereas for Rostami's algorithm the solution is still far away from the boundary for `rdrusselator`. A similar situation occurs for a particular initial guess with `tolosa` due to the nonmonotonic convergence of the error. While this might be alleviated by the monotonic algorithm from [8], Rostami's algorithm takes considerably more time even if it converges (monotonically) in 24 steps. For `pde`, Rostami's algorithm converges to the boundary but it is not a global solution as already indicated in [19]. This is also clearly visible in Figure 10.

9. Summary and conclusions. We presented a robust criss-cross type algorithm for computing the real pseudospectral abscissa. Our algorithm exploits the superset characterization of the real pseudospectrum, and performs criss-cross searches on selective touching supersets. Global convergence of the algorithm is proved. Due to use of dense eigenvalue and singular value calculations, the criss-cross algorithm

TABLE 6
Computational results for $\varepsilon = 0.01$.

	Rostami's algorithm [19]			Subspace Algorithm 4 (SVD expansion)		
Case	$\alpha_\varepsilon^{\mathbb{R}}(A)$	# Its.	time (s.)	$\alpha_\varepsilon^{\mathbb{R}}(A)$	# Its.	time (s.)
`pde`	9.95239250	7	2.0	9.95239251	5	2.3
`rdrusselator`	0.11662268	2	15.2	0.11662268	2	3.5
`tolosa`	−0.14119780	3	154.0	−0.13418881	5	15.0
`tubular`	−0.80500864	4	6.6	−0.80500864	5	0.9

TABLE 7
Computational results for $\varepsilon = 0.1$. The additional row for `tolosa` corresponds to an initial guess for `eigs` that leads to nonmonotonic behavior of the error.

	Rostami's algorithm [19]			Subspace Algorithm 4 (SVD expansion)		
Case	$\alpha_\varepsilon^{\mathbb{R}}(A)$	# Its.	time (s.)	$\alpha_\varepsilon^{\mathbb{R}}(A)$	# Its.	time (s.)
`pde`	10.1758394	14	4.0	10.2037672	5	1.6
`rdrusselator`	0.20662268	2	14.9	0.28535238	3	7.0
`tolosa`	7.17495157	24	2891.1	7.17495157	6	19.0
	(1.57926302)	(100)	(3993.4)			
`tubular`	−0.59396162	10	14.6	−0.59396162	5	1.0

is suitable only for small to medium sized problems. We therefore developed a subspace version of our algorithm that is suitable for large-scale problems. Under generic assumptions, this algorithm is proved to be superlinearly convergent. Numerical examples show the effectiveness and robustness of the original algorithm and its subspace accelerated version. In particular, the algorithms are more reliable in finding the globally rightmost points compared to approaches based on dynamical low-rank iterations. A MATLAB implementation of our proposed algorithms is available from http://www.unige.ch/math/vandereycken.

As future work, we expect that our approach also applies to other structured pseudospectra problems that allow for a superset characterization. For example, skew-symmetric, Hermitian, and Hamiltonian perturbations have a μ function with an optimization formulation similar to (4); see, e.g., [14].

Appendix A. Technical results.

A.1. Proof of the reduced unimodality (Theorem 6.1). In this section, we prove the unimodality of the reduced function $\widehat{g}$ as stated in Theorem 6.1. To this end, we need the following two technical lemmas. The first one is a generalization of a similar result in [18] for the unreduced function g.

LEMMA A.1. *Let $X, Y \in \mathbb{R}^{n\times k}$ with $n \geq k \geq 1$.*

(a) *The function $g(\gamma) = \sigma_{-2}\left(\left[\begin{smallmatrix} X & -\gamma Y \\ \gamma^{-1}Y & X \end{smallmatrix}\right]\right)$ has at most one local maximum on $(0,1)$.*

(b) *Suppose $k \geq 2$ and*

$$\operatorname{rank}\left(\begin{bmatrix} X & Y \end{bmatrix}\right) + \operatorname{rank}\left(\begin{bmatrix} X^{\mathrm{T}} & Y^{\mathrm{T}} \end{bmatrix}\right) - \operatorname{rank}(Y) < 2k - 1, \tag{51}$$

then the supremum of $g(\gamma)$ on $(0,1]$ is attained.

Proof. (a) We shall prove that any local extremum (either maximum or minimum) of $g(\gamma)$ on $(0,1)$ must be a global maximum. Let $M = X + \jmath Y$. We claim

$$\sup_{\gamma\in(0,1)} g(\gamma) \leq \inf\{\|\Delta\|_2 : \sigma_{\min}(\Delta - M) = 0,\ \Delta \in \mathbb{R}^{n\times k}\}. \tag{52}$$

TABLE 8
Corresponding rightmost points and values for μ with $\varepsilon = 0.1$.

Case	Rostami's algorithm [19] $\lambda_r = \alpha_r + \jmath\beta_r$	$\mu(\alpha_r, \beta_r)$
`pde`	$10.17583935249123 + 0.411740915433475 \cdot \jmath$	0.100
`rdrusselator`	$0.206622682956816 + 1.901154527116052 \cdot \jmath$	0.068
`tolosa`	$7.174951581710051 + 158.7304383782220 \cdot \jmath$	0.100
	$(1.579263021428338 + 153.4647745341267 \cdot \jmath)$	(0.080)
Case	Subspace Algorithm 4 (SVD expansion) $\lambda_r = \alpha_r + \jmath\beta_r$	$\mu(\alpha_r, \beta_r)$
`pde`	$10.20376723848194 + 0 \cdot \jmath$	0.100
`rdrusselator`	$0.285352383762668 + 1.847271019773242 \cdot \jmath$	0.100
`tolosa`	$7.174951567538598 + 158.7301860895381 \cdot \jmath$	0.100

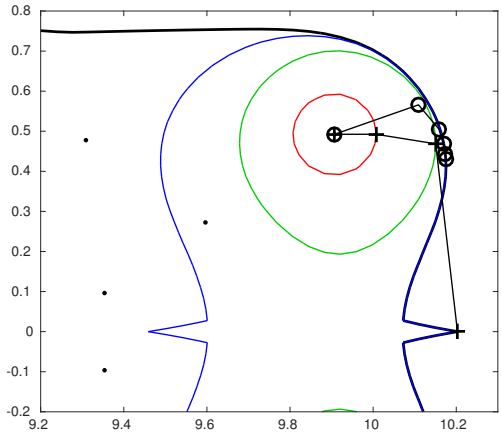

FIG. 10. *Convergence of Algorithm* 4 *and Rostami's algorithm* [19] *applied to* `pde` *with* $\varepsilon = 0.1$. *The black dots are the eigenvalues of* A. *The convergence history of the rightmost points computed by Algorithm* 4 *and Rostami's algorithm are marked with* $+$ *and* $\circ$, *respectively. The boundaries of* $\Lambda_\varepsilon^{\mathbb{R}}(A)$ *and* $\Lambda_\varepsilon^{\mathbb{R}}(AV_k, V_k)$ *are marked with black and colored lines. The iteration numbers are* $k = 1, 2, 3$, *from inside to outside. (Figure is in color online.)*

Using the identity (33), we have

$$T_{n,\gamma} \begin{bmatrix} \Delta - M & \\ & \Delta - \overline{M} \end{bmatrix} T_{k,\gamma}^{-1} = \begin{bmatrix} \Delta & \\ & \Delta \end{bmatrix} - \begin{bmatrix} X & -\gamma Y \\ \gamma^{-1} Y & X \end{bmatrix}$$

since Δ is a real matrix. Hence, $\sigma_{\min}(\Delta - M) = 0$ leads to

$$2k - 2 \geq \operatorname{rank}\left(\begin{bmatrix} \Delta & \\ & \Delta \end{bmatrix} - \begin{bmatrix} X & -\gamma Y \\ \gamma^{-1} Y & X \end{bmatrix} \right).$$

Applying the Weyl's inequalities for singular values (see, e.g., [12, Cor. 7.3.5]) then establishes (52) since

$$\|\Delta\|_2 = \left\| \begin{bmatrix} \Delta & \\ & \Delta \end{bmatrix} \right\|_2 \geq \sigma_{-2}\left(\begin{bmatrix} X & -\gamma Y \\ \gamma^{-1} Y & X \end{bmatrix} \right) = g(\gamma). \tag{53}$$

Now suppose that $g(\gamma_*) > 0$ is a local extremum satisfying $\gamma_* \in (0,1)$. We shall construct Δ_* such that $\sigma_{\min}(\Delta_* - M) = 0$ and $g(\gamma_*) = \|\Delta_*\|_2$, thereby proving that any local maximum on $(0,1)$ has to be a global one by virtue of (52).

By [18, Lemma 4, and Claims 1 and 2], there exist $u = [u_1^{\mathrm{T}}, u_2^{\mathrm{T}}] \in \mathbb{R}^{n+n}$ and $v = [v_1^{\mathrm{T}}, v_2^{\mathrm{T}}] \in \mathbb{R}^{k+k}$ such that

$$\begin{bmatrix} X & -\gamma_* Y \\ \gamma_*^{-1} Y & X \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = g(\gamma_*) \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \tag{54}$$

(i.e., they are the left and right singular vector for the singular value $g(\gamma_*)$), and due to the local optimality condition of γ_*, the singular vectors satisfy

$$u_1^{\mathrm{T}} u_1 = v_1^{\mathrm{T}} v_1, \quad u_2^{\mathrm{T}} u_2 = v_2^{\mathrm{T}} v_2, \quad \text{and} \quad u_1^{\mathrm{T}} u_2 = v_1^{\mathrm{T}} v_2.$$

Let $V^\dagger$ be the Moore–Penrose pseudoinverse of V. Then, [18, Lemma 2] shows

$$\|UV^\dagger\|_2 = 1, \quad UV^\dagger V = U \quad \text{for} \quad U = \begin{bmatrix} u_1 & u_2 \end{bmatrix}, \quad \text{and} \quad V = \begin{bmatrix} v_1 & v_2 \end{bmatrix}. \tag{55}$$

Define now $\Delta_* = g(\gamma_*) \cdot UV^\dagger$. Recalling $M = X + \jmath Y$ and using (54)–(55), we obtain

$$\begin{aligned} (\Delta_* - M)\,(v_1 + \jmath\gamma_* v_2) &= g(\gamma_*) UV^\dagger (v_1 + \jmath\gamma_* v_2) - (Xv_1 - \gamma_* Y v_2) - \jmath(\gamma_* X v_2 + Y v_1) \\ &= g(\gamma_*)(u_1 + \jmath\gamma_* u_2) - g(\gamma_*) u_1 - \jmath\gamma_* g(\gamma_*) u_2 = 0. \end{aligned}$$

Hence, we have shown that $\Delta_* - M$ has nontrivial kernel and $g(\gamma_*) = \|\Delta_*\|_2$.

(b) Under condition (51), we claim $\lim_{\gamma \to 0} g(\gamma) = 0$. Since $g(\gamma) \geq 0$, the result then follows immediately. To prove the claim, we have from [18, Lemma 5] that

$$\lim_{\gamma \to 0} \sigma_{-2}\left(\begin{bmatrix} X & -\gamma Y \\ \gamma^{-1} Y & X \end{bmatrix} \right) = \sigma_{-2}\left(\begin{bmatrix} 0_{n-r,k-r} & U_2^{\mathrm{T}} X \\ X V_2 & 0_{n \times k} \end{bmatrix} \right), \tag{56}$$

where $r = \operatorname{rank}(Y)$, and the orthonormal columns of $U_2 \in \mathbb{R}^{n \times (n-r)}$ and $V_2 \in \mathbb{R}^{k \times (k-r)}$ are bases for $\ker(Y^{\mathrm{T}})$ and $\ker(Y)$, respectively.

On the other hand, let the orthonormal columns of $U_1 \in \mathbb{R}^{n \times (n-r)}$ be a basis for $\operatorname{span}(Y)$. Then $\begin{bmatrix} U_1 & U_2 \end{bmatrix} \in \mathbb{R}^{n \times n}$ is orthogonal. Hence,

$$\begin{bmatrix} U_1 & U_2 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} Y & X \end{bmatrix} = \begin{bmatrix} U_1^{\mathrm{T}} Y & U_1^{\mathrm{T}} X \\ 0 & U_2^{\mathrm{T}} X \end{bmatrix},$$

which implies

$$\operatorname{rank}(\begin{bmatrix} Y & X \end{bmatrix}) \geq \operatorname{rank}(U_2^{\mathrm{T}} X) + \operatorname{rank}(Y).$$

Analogously, we have

$$\operatorname{rank}(\begin{bmatrix} Y^{\mathrm{T}} & X^{\mathrm{T}} \end{bmatrix}) \geq \operatorname{rank}(V_2^{\mathrm{T}} X^{\mathrm{T}}) + \operatorname{rank}(Y^{\mathrm{T}}).$$

Therefore, from the assumption (51),

$$\begin{aligned} \operatorname{rank}\left(\begin{bmatrix} 0_{n-r,k-r} & U_2^{\mathrm{T}} X \\ X V_2 & 0_{n \times k} \end{bmatrix} \right) &= \operatorname{rank}(U_2^{\mathrm{T}} X) + \operatorname{rank}(X V_2) \\ &\leq \operatorname{rank}(\begin{bmatrix} X & Y \end{bmatrix}) + \operatorname{rank}(\begin{bmatrix} X^{\mathrm{T}} & Y^{\mathrm{T}} \end{bmatrix}) - 2\operatorname{rank}(Y) \\ &< (2k-1) - r. \end{aligned}$$

This implies that the $(2k-r-1)$st singular value in (56) vanishes, which completes the proof. ∎

Next, we connect the previous lemma to our condition on the subspace $\mathcal{V}$ in Theorem 6.1.

LEMMA A.2. *Suppose the projection subspace satisfies* (31) *with* $\mathcal{V} = \operatorname{range} V$. *Then condition* (51) *holds for* $X = AV - \alpha V$ *and* $Y = \beta V$ *for all* $\alpha + \jmath\beta \in \mathbb{C}$ *with* $\beta \neq 0$.

Proof. We only consider the case of complex eigenvectors, the real case is proved similarly. By separating the real and imaginary part, we see that $A(x_0 + \jmath y_0) = (\alpha_0 + \jmath\beta_0)(x_0 + \jmath y_0)$ implies $A \begin{bmatrix} x_0 & y_0 \end{bmatrix} \subset \operatorname{range} \begin{bmatrix} x_0 & y_0 \end{bmatrix}$. Therefore, $\mathcal{V}$ contains an invariant subspace of A with dimension 2. Since $\operatorname{rank}(Y) = \operatorname{rank}(V) = k$, condition (51) follows from

$$2k - 2 \geq \operatorname{rank}\left(\begin{bmatrix} AV & V \end{bmatrix}\right) = \operatorname{rank}\left(\begin{bmatrix} AV - \alpha V & \beta V \end{bmatrix}\right),$$

and $\operatorname{rank}\left(\begin{bmatrix} (AV - \alpha V)^{\mathrm{T}} & \beta V^{\mathrm{T}} \end{bmatrix}\right) = k$. ∎

We are now ready to prove the final result.

Proof of Theorem 6.1. (a) The function $g(\gamma)$ in Lemma A.1 equals $\widehat{g}(\alpha, \beta, \gamma; V)$ by taking $X = AV - \alpha V$ and $Y = \beta V$. Thanks to Lemma A.2, the statement of the theorem now follows directly.

(b) By construction, both $AV - \lambda V$ and $AV - \overline{\lambda} V$ are rank deficient. The identity (33) gives

$$\sigma_{-2}\left(\begin{bmatrix} AV - \lambda V & \\ & AV - \overline{\lambda} V \end{bmatrix}\right) = \sigma_{-2}\left(\begin{bmatrix} AV - \alpha V & -\beta\gamma V \\ \beta\gamma^{-1} V & AV - \alpha V \end{bmatrix}\right) = 0.$$

Hence, $\widehat{g}(\alpha, \beta, \gamma) \equiv 0$ for all $\gamma \in (0, 1]$ which implies $\widehat{\mu}(\alpha, \beta) = 0$ and $\lambda = \alpha + \jmath\beta \in \Lambda_\varepsilon^{\mathbb{R}}(AV, V)$ for all $\varepsilon \geq 0$. ∎

A.2. Proof of the quadratic approximation (Theorem 6.3). In this section, we show the quadratic approximation property w.r.t. the subspace distance of the reduced $\widehat{\mu}$ function as stated in Theorem 6.3. We start with an exactness result that builds again on [18].

LEMMA A.3 (exactness). *Suppose* V *is an orthonormal matrix such that* $\mathcal{V} = \operatorname{range}(V) \subset \mathbb{R}^n$ *contains all vectors* $x_1, x_2 \in \mathbb{R}^n$ *that satisfy* (39). *Then,*

$$\mu(\alpha, \beta) = \widehat{\mu}(\alpha, \beta\,;\, V) = \widehat{g}(\alpha, \beta, \gamma_*\,;\, V), \tag{57}$$

with γ_* *the maximizer of* $g(\alpha, \beta, \gamma)$.

Proof. Due to the monotonicity from Theorem 6.2, it suffices to show $\mu(\alpha, \beta) \geq \widehat{\mu}(\alpha, \beta\,;\, V)$. A long argument in [18, Construction of a worst Δ, p. 886] proves that there exists a real perturbation matrix Δ such that[4]

$$(A - \lambda I - \Delta)v = 0 \quad \text{with} \quad \Delta \in \mathbb{R}^{n\times n}, \quad \|\Delta\|_2 \leq \mu(\alpha, \beta), \tag{58}$$

where $\lambda = \alpha + \jmath\beta$, and $v = x_1 + \jmath\gamma_* x_2$ with x_1 and x_2 being real vectors satisfying (39). Since $\operatorname{range}(V)$ contains x_1, x_2, there exists a nonzero (complex) vector s such that $v = Vs$. Therefore,

$$0 = \sigma_{\min}\big((A - \lambda I - \Delta)V\big) = \sigma_{\min}(AV - \lambda V - \widehat{\Delta}),$$

[4]When λ is not an eigenvalue of A, (58) follows from the equation below [18, eq. (10)] with $M = (A - \lambda I)^{-1}$ and $P(\gamma) = G(\alpha, \beta, \gamma)^{-1}$. Otherwise, take $\Delta = 0$ and use (33).

and by exploiting the identity (33) we have

$$0 = \sigma_{-2}\left(\begin{bmatrix} AV - \alpha V & -\beta\gamma V \\ \beta\gamma^{-1}V & AV - \alpha V \end{bmatrix} - \begin{bmatrix} \widehat{\Delta} & \\ & \widehat{\Delta} \end{bmatrix}\right) \qquad \text{for} \quad \gamma \in (0, 1].$$

Then, by standard perturbation of singular values (see, e.g., [12, Cor. 7.3.5]) we obtain

$$\widehat{g}(\alpha, \beta, \gamma\,;\, V) = \sigma_{-2}\left(\begin{bmatrix} AV - \alpha V & -\beta\gamma V \\ \beta\gamma^{-1}V & AV - \alpha V \end{bmatrix}\right) \le \|\widehat{\Delta}\|_2 \le \mu(\alpha, \beta) \le \widehat{g}(\alpha, \beta, \gamma_*\,;\, V),$$

where the last inequality is due to (39) and the monotonicity in $\widehat{g}$; see (36). The proof is completed by taking supremum w.r.t. γ on the left. ∎

Next, we state some regularity properties of $\widehat{g}$ assuming it is smooth and strongly concave in γ.

LEMMA A.4. *Assume $\beta \neq 0$. Let $\mu(\alpha, \beta) = \sigma_{-2}(G(\alpha, \beta, \gamma_*))$ with $\gamma_* = \gamma_{\max}(\alpha, \beta)$ be a simple singular value with right singular vector $x = [x_1^{\mathrm{T}}, x_2^{\mathrm{T}}]^{\mathrm{T}}$. Then, for $V = orth([x_1, x_2])$ the following hold:*
(a) *x_1 and x_2 are linearly independent, hence $V \in \mathbb{R}^{n\times 2}$;*
(b) *$\widehat{g}(\alpha, \beta, \gamma_*\,;\, V) = \sigma_{-2}\big(G(\alpha, \beta, \gamma_*)V_{[2]}\big)$ is a simple singular value.*
If, in addition, $\widehat{g}_{\gamma\gamma}(\alpha, \beta, \gamma_*\,;\, V) \neq 0$, then for any orthonormal matrix $U \in \mathbb{R}^{n\times 2}$ the following hold:
(c) *$\widehat{\gamma}_{\max}(\alpha, \beta\,;\, U) = \gamma_* + O(\|U - V\|_2)$;*
(d) *$\widehat{g}(\alpha, \beta, \gamma_*\,;\, U) = \mu(\alpha, \beta) + O(\|U - V\|_2^2)$.*

Proof. (a) Assume $x_2 = cx_1$, then (58) implies $(A - \lambda I - \Delta)x_1 = 0$. The imaginary part then gives $\jmath\beta x_1 = 0$. So we have $x_1 = x_2 = 0$, which leads to a contradiction.

(b) We first show that $x' \in \mathrm{range}(V_{[2]})$, where x' is the right singular vector of the smallest singular value $\sigma_{-1}(G)$ (we omit the argument in $G(\alpha, \beta, \gamma_*)$). According to [18, Construction of a worst Δ, p. 886], there exists Δ with

$$\|\Delta\|_2 = \sigma_{-2}(G) \quad \text{and} \quad \mathrm{range}(\Delta^{\mathrm{T}}) \subset \mathrm{range}(V),$$

such that $\det(A - \lambda I - \Delta) = 0$. This implies $\mathrm{rank}(G - \Delta_{[2]}) \le 2n - 2$ due to the augmentation relation (33) with $A - \Delta$ and I taken as A and B. Consequently, the matrix $M = G - \Delta_{[2]}$ is a best rank $2n-2$ approximation to the matrix G in spectral norm since since $\|G - M\|_2 = \|\Delta\|_2 = \sigma_{-2}(G)$. Therefore, by parametrizing all optimal rank $2n - 2$ solutions, it holds that

$$\Delta_{[2]} = \sigma_{-1}(G) \cdot y' x'^{\mathrm{T}} + \sigma_{-2}(G) \cdot y x^{\mathrm{T}} + \sigma_{-2}(G) \cdot Y \Theta X^{\mathrm{T}},$$

where $[y', y, Y]$ and $[x', x, X]$ are the left and right singular vectors of G, and Θ such that $\|\Theta\|_2 \le 1$. So we obtain $x, x' \in \mathrm{range}(\Delta_{[2]}^{\mathrm{T}}) \subset \mathrm{range}(V_{[2]})$.

Then (b) follows immediately from

$$\sigma_{-1}(G) = \sigma_{-1}(GV_{[2]}) < \sigma_{-2}(G) = \sigma_{-2}(GV_{[2]}) < \sigma_{-3}(G) \le \sigma_{-3}(GV_{[2]}),$$

where we used $\sigma_{-j}(G) \le \sigma_{-j}(GV_{[2]})$ due to the the min-max characterization of singular values [12, Theorem 7.3.8], and the equalities hold due to both $x, x' \in \mathrm{range}(V_{[2]})$.

(c) Fix (α, β). By standard results, the simple singular value

$$\widehat{g}(\alpha, \beta, \gamma\,;\, V) = \sigma_{-2}\left(\begin{bmatrix} AV - \alpha V & -\beta\gamma V \\ \beta\gamma^{-1}V & AV - \alpha V \end{bmatrix}\right)$$

is analytic at (γ_*, V). For all U in a neighborhood of V, the maximizer $\widehat{\gamma}_{\max}(\alpha, \beta\,;\, U)$ is determined by $\widehat{g}_\gamma(\alpha, \beta, \gamma\,;\, U) = 0$. Since γ_* is also the maximizer of $\widehat{g}(\alpha, \beta, \gamma\,;\, V)$ by (57), the continuity of $\widehat{\gamma}_{\max}(\alpha, \beta\,;\, U)$ at $U = V$ follows immediately from the implicit function theorem and $\widehat{g}_{\gamma\gamma}(\alpha, \beta, \gamma_*\,;\, V) \neq 0$.

(d) Let z and s be the left and right singular vectors of $\sigma_{-2}\big(GV_{[2]}\big)$, respectively. Then, denoting $U = V + R$, we have

$$\sigma_{-2}\big(GU_{[2]}\big) = \sigma_{-2}\big(GV_{[2]} + GR_{[2]}\big) = \sigma_{-2}\big(GV_{[2]}\big) + z^{\mathrm{T}} G R_{[2]} s + O(\|R\|_2^2), \tag{59}$$

where the second equation is derived by eigenvalue expansion [22, Chap. IV, Theorem 2.3]. Since the right singular vector $x \in \mathrm{range}(V_{[2]})$ belongs to the simple singular value $\sigma_{-2}(GV_{[2]})$, it is easy to see that $x = V_{[2]} s$. We therefore get

$$G \cdot V_{[2]} s = \mu \cdot z \quad \text{and} \quad G^{\mathrm{T}} z = \mu \cdot V_{[2]} s \quad \text{with} \quad \mu = \sigma_{-2}(G).$$

Consequently,

$$z^{\mathrm{T}} G R_{[2]} s = \mu \cdot (V_{[2]} s)^{\mathrm{T}} R_{[2]} s = \frac{\mu}{2} \cdot s^{\mathrm{T}} R_{[2]}^{\mathrm{T}} R_{[2]} s = O(\|R\|_2^2),$$

where in the second equation we exploited $V^{\mathrm{T}} R + R^{\mathrm{T}} V + R^{\mathrm{T}} R = 0$ due to $U^{\mathrm{T}} U = I$ and $V^{\mathrm{T}} V = I$. Plugging the equation above into (59) and noting that $\sigma_{-2}(GV_{[2]}) = \mu(\alpha, \beta)$ completes the proof. ∎

We are now ready to prove the final result.

Proof of Theorem 6.3. The first inequality is a direct consequence of Theorem 6.2. To prove the second, let $\delta = \delta(x, U)$. Then there exists E such that $\|E\|_F = O(\delta)$ and $[x_1, x_2] + E \in \mathrm{range}(U)$. Define

$$\widetilde{U} = \mathrm{orth}([x_1, x_2] + E) = \mathrm{orth}([x_1, x_2]) + \widetilde{E} = V + \widetilde{E}, \qquad \|\widetilde{E}\|_F = O(\delta), \tag{60}$$

where the second equation holds since $\mathrm{rank}([x_1, x_2]) = 2$ by Lemma A.4(a). Observe that $\mathrm{range}(\widetilde{U}) \subset \mathrm{range}(U)$. The min-max characterization of singular values (cf. the proof of Theorem 6.2) implies for all $\gamma \in (0, 1]$,

$$\sigma_{-2}\big(G(\alpha, \beta, \gamma) U_{[2]}\big) \leq \sigma_{-2}\big(G(\alpha, \beta, \gamma) \widetilde{U}_{[2]}\big) \leq \widehat{g}\big(\alpha, \beta, \widehat{\gamma}_*\,;\, \widetilde{U}\big), \qquad \widehat{\gamma}_* = \widehat{\gamma}_{\max}(\alpha, \beta\,;\, \widetilde{U}).$$

where the function $\widehat{\gamma}_{\max}$ is defined by (37) analogous to $\gamma_{\max}$. Lemma A.4(c) and (60) imply $\widehat{\gamma}_* = \gamma_* + O(\delta)$.

Since $\widehat{g}(\alpha, \beta, \gamma_*;\, V\,)$ is a simple singular value, $\widehat{g}(\alpha, \beta, \widehat{\gamma}_*;\, \widetilde{U}\,)$ remains simple for $(\widehat{\gamma}_*, \widetilde{U})$ close to (γ_*, V). Therefore, expanding $\widehat{g}(\alpha, \beta, \gamma\,;\, \widetilde{U})$ at $\gamma = \widehat{\gamma}_*$ gives

$$\widehat{g}(\alpha, \beta, \widehat{\gamma}_*\,;\, \widetilde{U}) = \widehat{g}(\alpha, \beta, \gamma_*\,;\, \widetilde{U}) + O(\delta^2) = \mu(\alpha, \beta) + O(\delta^2),$$

where we first used that $\widehat{g}_\gamma(\alpha, \beta, \widehat{\gamma}_*\,;\, \widetilde{U}) = 0$ due to $\widehat{\gamma}_*$ being a maximizer, and then Lemma A.4(d) with (60). ∎

A.3. Proof of Lemma 7.2.

Proof. Let us first prove (b). Due to $\widehat{g}(\alpha, \beta, \gamma\,;\, V_k) = \varepsilon$ and (46), we have

$$\varepsilon = \sigma_{-2}\left(\begin{bmatrix} \widehat{A} - \alpha\widehat{B} & -\beta\gamma\widehat{B} \\ \beta\gamma^{-1}\widehat{B} & \widehat{A} - \alpha\widehat{B} \end{bmatrix}\right) = \sigma_{-2}\left(\begin{bmatrix} \widehat{A} - \alpha\widehat{B} & \jmath\beta\gamma\widehat{B} \\ \jmath\beta\gamma^{-1}\widehat{B} & \widehat{A} - \alpha\widehat{B} \end{bmatrix}\right), \tag{61}$$

where in the last equality we applied the same block scaling idea as in the proof of Lemma 2.3. Since ε is a singular value, we can again exploit the augmented matrix eigenvalue characterization to obtain

$$\det\left(\begin{bmatrix} \varepsilon I_{2k} & 0 & \widehat{A}-\alpha\widehat{B} & 0 \\ 0 & \varepsilon I_{2k} & 0 & \widehat{A}-\alpha\widehat{B} \\ \widehat{A}^{\mathrm{T}}-\alpha\widehat{B}^{\mathrm{T}} & 0 & \varepsilon I_k & 0 \\ 0 & \widehat{A}^{\mathrm{T}}-\alpha\widehat{B}^{\mathrm{T}} & 0 & \varepsilon I_k \end{bmatrix} - \jmath\beta \begin{bmatrix} 0 & 0 & 0 & -\gamma\widehat{B} \\ 0 & 0 & -\gamma^{-1}\widehat{B} & 0 \\ 0 & \gamma^{-1}\widehat{B}^{\mathrm{T}} & 0 & 0 \\ \gamma\widehat{B}^{\mathrm{T}} & 0 & 0 & 0 \end{bmatrix}\right) = 0.$$

After left and right multiplication by $\mathrm{diag}(\gamma^{-1/2}I_{2k}, \gamma^{1/2}I_{3k}, \gamma^{-1/2}I_k)$, permuting row and column blocks $(3,4)$, and left multiplying $\mathrm{diag}(-I_{4k}, I_{2k})$, we obtain $\det(\widehat{H}(\alpha,\gamma) - \jmath\beta\widehat{G}) = 0$.

Result (a) follows immediately from the augmented eigenvalue characterization of singular values, applied to the first equation in (61). ∎

Acknowledgments. The authors are grateful for the very helpful and insightful remarks by the referees, in particular for their suggestions on presentation and for a question that led to Remark 4.2.

REFERENCES

[1] B. Bernhardsson, A. Rantzer, and L. Qiu, *Real perturbation values and real quadratic forms in a complex vector space*, Linear Algebra Appl., 270 (1998), pp. 131–154, https://doi.org/10.1016/S0024-3795(97)00232-2.

[2] 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, https://doi.org/10.1093/imanum/23.3.359.

[3] R. Byers, *A bisection method for measuring the distance of a stable matrix to the unstable matrices*, SIAM J. Sci. Comput., 9 (1988), pp. 875–881, https://doi.org/10.1137/0909059.

[4] J. W. Demmel, *A counterexample for two conjectures about stability*, IEEE Trans. Automat. Control, 32 (1987), pp. 340–342, https://doi.org/10.1109/TAC.1987.1104595.

[5] J. F. Grcar, *Operator coefficient methods for linear equations*, Technical Report SAND898691, Sandia National Laboratories, Albuquerque, NM, 1989.

[6] N. Guglielmi, *On the method by Rostami for computing the real stability radius of large and sparse matrices*, SIAM J. Sci. Comput., 38 (2016), pp. A1662–A1681, https://doi.org/10.1137/15M1029709

[7] N. Guglielmi and C. Lubich, *Differential equations for roaming pseudospectra: Paths to extremal points and boundary tracking*, SIAM J. Numer. Anal., 49 (2011), pp. 1194–1209, https://doi.org/10.1137/100817851.

[8] N. Guglielmi and C. Lubich, *Low-rank dynamics for computing extremal points of real pseudospectra*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 40–66, https://doi.org/10.1137/120862399.

[9] D. Hinrichsen and A. J. Pritchard, *On spectral variations under bounded real matrix perturbations*, Numer. Math., 60 (1991), pp. 509–524, https://doi.org/10.1007/BF01385734.

[10] D. Hinrichsen and A. J. Pritchard, *On the robustness of stable discrete time linear systems*, in Progr. Systems Control Theory, 7, Birkhäuser Boston, Boston, 1991, pp. 393–400, https://doi.org/10.1007/978-1-4612-0439-8_49.

[11] D. Hinrichsen and A. J. Pritchard, *Mathematical Systems Theory* I, Springer, Heidelberg, 2010.

[12] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd ed., Cambridge University Press, Cambridge, 2013.

[13] M. Karow, *Geometry of Spectral Value Sets*, Ph.D. thesis, TU Berlin, 2003.

[14] M. Karow, E. Kokiopoulou, and D. Kressner, *On the computation of structured singular values and pseudospectra*, Systems Control Lett., 59 (2010), pp. 122–129, https://doi.org/10.1016/j.sysconle.2009.12.007.

[15] D. Kressner and B. Vandereycken, *Subspace methods for computing the pseudospectral abscissa and the stability radius*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 292–313, https://doi.org/10.1137/120869432.

[16] C. T. Lawrence, A. L. Tits, and P. Van Dooren, *A fast algorithm for the computation of an upper bound on the μ-norm*, Automatica J. IFAC, 36 (2000), pp. 449–456, https://doi.org/10.1016/S0005-1098(99)00165-X.
[17] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*, SIAM, Philadelphia, 1998, https://doi.org/10.1137/1.9780898719628.
[18] L. Qiu, B. Bernhardsson, A. Rantzer, E. J. Davison, P. M. Young, and J. C. Doyle, *A formula for computation of the real stability radius*, Automatica J. IFAC, 31 (1995), pp. 879–890, https://doi.org/10.1016/0005-1098(95)00024-Q.
[19] M. W. Rostami, *New algorithms for computing the real structured pseudospectral abscissa and the real stability radius of large and sparse matrices*, SIAM J. Sci. Comput., 37 (2015), pp. S447–S471, https://doi.org/10.1137/140975413.
[20] S. M. Rump, *Eigenvalues, pseudospectrum and structured perturbations*, Linear Algebra Appl., 413 (2006), pp. 567–593, https://doi.org/10.1016/j.laa.2005.06.009.
[21] J. Sreedhar, P. Van Dooren, and A. Tits, *A fast algorithm to compute the real structured stability radius*, in Stability Theory, Birkhäuser, Basel, 1996, pp. 219–230, https://doi.org/10.1007/978-3-0348-9208-7_23.
[22] G. W. Stewart and J. Sun, *Matrix Perturbation Theory*, Academic Press, Boston, 1990.
[23] L. N. Trefethen, *Computation of pseudospectra*, Acta Numer., 8 (1999), pp. 247–295, https://doi.org/10.1017/S0962492900002932.
[24] L. N. Trefethen and M. Embree, *Spectra and Pseudospectra: the Behavior of Nonnormal Matrices and Operators*, Princeton University Press, Princeton, NJ, 2005.
[25] T. G. Wright, *Eigtool*, http://www.comlab.ox.ac.uk/pseudospectra/eigtool/, 2002.