MATHICSE
Mathematics Institute of Computational Science and Engineering
School of Basic Sciences - Section of Mathematics

MATHICSE Technical Report
Nr. 13.2012
March 2012 (NEW 07.2013)

Subspace methods for computing the pseudospectral abscissa and the stability radius

Daniel Kressner, Bart Vandereycken

http://mathicse.epfl.ch

Phone: +41 21 69 37648

Address:
EPFL - SB - MATHICSE (Bâtiment MA)
Station 8 - CH-1015 - Lausanne - Switzerland

Fax: +41 21 69 32545

SUBSPACE METHODS FOR COMPUTING THE PSEUDOSPECTRAL ABSCISSA AND THE STABILITY RADIUS

DANIEL KRESSNER† AND BART VANDEREYCKEN†

Abstract. The pseudospectral abscissa and the stability radius are well-established tools for quantifying the stability of a matrix under unstructured perturbations. Based on first-order eigenvalue expansions, Guglielmi and Overton [SIAM J. Matrix Anal. Appl., 32 (2011), pp. 1166-1192] recently proposed a linearly converging iterative method for computing the pseudospectral abscissa. In this paper, we propose to combine this method and its variants with subspace acceleration. Each extraction step computes the pseudospectral abscissa of a small rectangular matrix pencil, which is comparably cheap and guarantees monotonicity. We observe local quadratic and prove local superlinear convergence of the resulting subspace methods. Moreover, these methods extend naturally to computing the stability radius. A number of numerical experiments demonstrate the robustness and efficiency of the subspace methods.

Key words. eigenvalue problem, pseudospectra, spectral abscissa, stability radius, subspace acceleration, complex approximation

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

1. Introduction.

For a square matrix $A \in \mathbb{C}^{n\times n}$, the *pseudospectrum* of A with respect to a perturbation level $\varepsilon > 0$ is defined as

$$\Lambda_\varepsilon(A) = \big\{ z \in \mathbb{C} \colon z \in \Lambda(A + \varepsilon\triangle) \text{ for some } \triangle \in \mathbb{C}^{n\times n} \text{ with } \|\triangle\| \le 1 \big\},$$

where $\|\cdot\|$ denotes the matrix 2-norm and $\Lambda(\,\cdot\,)$ denotes the set of all eigenvalues. Equivalently [36],

$$\Lambda_\varepsilon(A) = \big\{ z \in \mathbb{C} \colon \sigma_{\min}(A - zI) \le \varepsilon \big\}, \tag{1.1}$$

where $\sigma_{\min}(\,\cdot\,)$ denotes the minimal singular value. While the pseudospectrum offers a rich picture of the behavior of eigenvalues under perturbations, it is sometimes more useful to supply a single quantity, like an indication of the stability of A under perturbations. Examples of such quantities include the pseudospectral abscissa and the stability radius.

The *ε-pseudospectral abscissa* $\alpha_\varepsilon(A)$ of A is defined as the real part of the rightmost point in the ε-pseudospectrum:

$$\alpha_\varepsilon(A) := \max \big\{ \mathsf{Re}\, z \colon z \in \Lambda_\varepsilon(A) \big\} = \max \big\{ \mathsf{Re}\, z \colon \sigma_{\min}(A - zI) \le \varepsilon \big\}, \tag{1.2}$$

where the latter follows from the singular value characterization (1.1); see Figure 1.1 for an example. This and all subsequent figures have been generated by Eigtool [39].

In particular, $\alpha_\varepsilon(A) < 0$ implies that A remains stable (in the sense that all eigenvalues have negative real part) under perturbations of norm at most ε. The smallest ε for which this fails to hold is called the *stability radius* $\beta(A)$ of A:

$$\beta(A) := \min\{\varepsilon \in \mathbb{R} \colon \alpha_\varepsilon(A) \ge 0\}.$$

Of course, this definition makes only sense if A itself is stable. In this case, we obtain $\beta(A) = \min\{\varepsilon \in \mathbb{R} \colon \alpha_\varepsilon(A) = 0\}$ from the continuity of the pseudospectrum. Combined with (1.2), this yields

$$\beta(A) = \min\{\sigma_{\min}(A - zI) \colon z \in \mathrm{i}\mathbb{R}\} = 1/\max\{\|(A - zI)^{-1}\| \colon z \in \mathrm{i}\mathbb{R}\}.$$

†ANCHP, MATHICSE, École Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland. (daniel.kressner@epfl.ch, bart.vandereycken@epfl.ch)

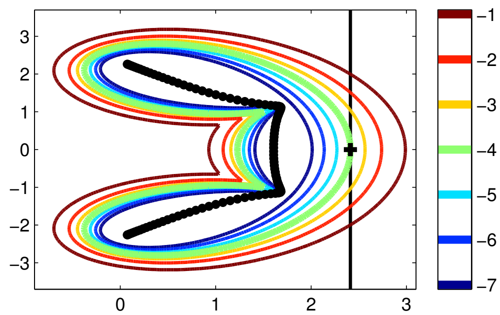

FIG. 1.1. *The pseudospectrum of A, the Grcar matrix [14] with $n = 100$, and $\alpha_\varepsilon(A)$ for $\varepsilon = 10^{-4}$. The eigenvalues A are denoted by black discs.*

In particular, the latter equality illustrates the close link between the stability radius and the H_∞ norm of a continuous linear time-invariant system [22].

Classical methods for computing the stability radius exploit that the intersection points of a vertical line with the boundary of $\Lambda_\varepsilon(A)$ can be characterized as the purely imaginary eigenvalues of a certain $2n \times 2n$ Hamiltonian matrix [10, 21]. Based on this characterization, Byers [10] proposed and analyzed a bisection method. This bisection method has been extended to H_∞ computations in [5, 20] and, subsequently, turned into quadratically convergent algorithms by exploiting higher regularity near a local optimum [4, 6]. Being very reliable, these algorithms are at the core of existing software for stability radius and H_∞ computations, such as the Control System Toolbox in MATLAB [30] and SLICOT [2]. However, the need for detecting all purely imaginary eigenvalues of a $2n \times 2n$ Hamiltonian matrix in every iteration becomes computationally challenging for a large and possibly sparse matrix A; see, e.g., [25] for a discussion. According to [19], this disadvantage can be partly avoided by using a locally convergent procedure to minimize $\sigma_{\min}(A - zI)$ on the imaginary axis. The Hamiltonian eigenvalue problem then only needs to be solved to verify that the obtained local optimum is also a global minimum and to restart the local procedure if this is not the case. Recently, Freitag and Spence [12] have proposed an approach based on a similar principle but with a different local procedure. Eigenvalue continuation is employed to find a 2×2 Jordan block corresponding to a purely imaginary eigenvalue of the parameter-dependent Hamiltonian eigenvalue problem.

Algorithms for computing the ε-pseudospectral abscissa have been proposed only relatively recently. The criss-cross algorithm by Burke, Lewis, and Overton [8] uses the Hamiltonian characterization mentioned above to traverse the pseudospectrum horizontally and vertically in an alternating way (see also Section 5.2). While this algorithm is very robust—it converges globally and locally quadratically—the need for solving Hamiltonian eigenvalue problems makes it again unsuitable for larger matrices. Guglielmi and Overton [17] have proposed an algorithm that is based on first-order eigenvalue expansions and only requires computing the right-most eigenvalue for a rank-one perturbation of A in every iteration; see Section 2 for more details. Although this algorithm is not guaranteed to converge to the globally right-most point of the pseudospectrum, such a behavior has been observed for a large number of numerical examples in [17]. However, a disadvantage of the algorithm is that its convergence is only linear and may become quite slow in certain cases.

In this paper, we propose a subspace method to overcome the potentially slow convergence of the algorithm from [17]. The basic idea is to collect the essential information from all previous iterates in a subspace and obtain a more accurate iterate by extracting a quasi-best approximation from this subspace. Such an acceleration can be found in many subspace methods in numerical linear algebra, including Krylov subspace and Jacobi-Davidson methods for solving eigenvalue problems [1]. What may be unique about our subspace approach is that it provably turns a linearly converging iteration into a superlinearly converging one. Numerically, we even observe local quadratic convergence. While this reminds of existing results for vector extrapolation techniques [13, 34], it is important to note that subspace methods can generally *not* be viewed as vector extrapolation.

The rest of this paper is organized as follows. Section 2 provides a brief summary of the algorithm by Guglielmi and Overton. In Section 3, we develop the fundamental ideas behind our subspace method and prove monotonicity as well as stability of the extraction procedure. Two different variants of the subspace are proposed and their convergence is analyzed in Section 4. How to suitably combine these two variants and other implementation details, such as the computation of the pseudospectral abscissa for rectangular matrix pencils, are discussed in Section 5, along with several numerical experiments. Section 6 is concerned with extending our methods to the computation of the stability radius.

2. The algorithm by Guglielmi and Overton. In the following, we briefly summarize the basic algorithm in [17] together with some results.

The algorithm of Guglielmi and Overton aims at constructing a sequence of unit norm perturbations $\triangle_0$, $\triangle_1$, $\triangle_2$, ... converging to $\triangle \in \mathbb{C}^{n\times n}$ such that the right-most eigenvalue of $A+\varepsilon\triangle$ coincides with a right-most point μ_α of the ε-pseudospectrum. Considering iteration k of the algorithm, suppose that the right-most eigenvalue μ_k of $A+\varepsilon\triangle_k$ is simple. Let u_k and v_k denote unit-norm normalized left and right eigenvectors belonging to μ_k. Since μ_k is simple, we have $u_k^* v_k \neq 0$ [23, Lemma 6.3.10]. In fact, by a suitable scaling of u_k or v_k, we can always assume that u_k and v_k are so-called RP-compatible.

Definition 2.1. *Two vectors $u, v \in \mathbb{C}^n$ are called* RP-compatible *if $\|u\| = \|v\| = 1$ and u^*v is real and positive.*

To determine the next perturbation $\triangle_{k+1} \in \mathbb{C}^{n\times n}$, it is desirable that the real part of the right-most eigenvalue μ_{k+1} of

$$A+\varepsilon\triangle_{k+1} = A+\varepsilon\triangle_k + \varepsilon(\triangle_{k+1}-\triangle_k)$$

is as large as possible. By a first-order eigenvalue expansion [23, Theorem 6.3.12], we have

$$\mathsf{Re}(\mu_{k+1}) = \mathsf{Re}(\mu_k) + \frac{\varepsilon}{u_k^* v_k}\mathsf{Re}(u_k^*(\triangle_{k+1}-\triangle_k)v_k) + O(\|\triangle_{k+1}-\triangle_k\|^2). \tag{2.1}$$

Since $\|\triangle_{k+1}\| = 1$, one has $|u_k^*\triangle_{k+1}v_k| \leq 1$ and therefore $\triangle_{k+1} = u_k v_k^*$ maximizes $\mathsf{Re}(u_k^*\triangle_{k+1}v_k) = 1$.

These considerations lead to Algorithm 1 below. Note that we call $\mu_\alpha \in \partial\Lambda_\varepsilon(A)$ a *locally right-most point* of the ε-pseudospectrum if μ_α is the right-most point of $U \cap \Lambda_\varepsilon(A)$ for some open set U containing μ_α. By choosing U sufficiently small, this right-most point is always unique, see [17, Lemma 2.5].

The first-order analysis based on (2.1) does not explain why one would use $\triangle_{k+1} = u_k v_k^*$ in Algorithm 1 when $\|\triangle_{k+1}-\triangle_k\|$ is not small. Alternatively, the homotopy analysis in [17] shows that $\triangle_{k+1} = u_k v_k^*$ almost surely results in an increase of the

Algorithm 1 Guglielmi and Overton [17, Algorithm PSA0]

Input: Matrix $A \in \mathbb{C}^{n\times n}$, perturbation level $\varepsilon > 0$.
Output: Approximation μ_α to a locally right-most point of $\Lambda_\varepsilon(A)$.
Set $\triangle_0 = 0$.
for $k = 0, 1, 2, \dots$ until converged **do**
Compute a right-most eigenvalue μ_k with corresponding left/right RP-compatible eigenvectors u_k, v_k of $A + \varepsilon\triangle_k$.
Set $\triangle_{k+1} = u_k v_k^*$.
end for
return $\mu_\alpha = \mu_k$

real part of the right-most eigenvalue of $A + (\varepsilon - t)\triangle_k + t\triangle_{k+1}$ when $t \to 0$. To ensure such a monotonic increase of $\mathsf{Re}(\mu_k)$ for finite t, a number of modifications to Algorithm 1 are proposed in [17]. Since our subspace variants of Algorithm 1 satisfy monotonicity automatically, we omit these discussions.

The following assumption is crucial for the analysis and success of Algorithm 1.

ASSUMPTION 1. *Let μ_α be a locally right-most point of $\Lambda_\varepsilon(A)$. Then $\sigma_{min}(A - \mu_\alpha I)$ is simple.*

Let v_α be a right singular vector belonging to $\sigma_{\min}(A - \mu_\alpha I)$, which is equal to ε. Then the corresponding left singular vector u_α is given by $u_\alpha = \frac{1}{\varepsilon}(A - \mu_\alpha I)v_\alpha$. By [17, Lemma 2.7], Assumption 1 implies that $u_\alpha^* v_\alpha$ is real and negative; in other words,

$$v_\alpha^*(A - \mu_\alpha I)v_\alpha < 0. \tag{2.2}$$

Together with

$$(A - \varepsilon u_\alpha v_\alpha^* - \mu_\alpha I)v_\alpha = 0, \qquad u_\alpha^*(A - \varepsilon u_\alpha v_\alpha^* - \mu_\alpha I) = 0,$$

this shows that $-u_\alpha$ and v_α are an RP-compatible pair of left/right eigenvectors for $A - \varepsilon u_\alpha v_\alpha^*$ belonging to the eigenvalue μ_α. Because of Assumption 1, this eigenvalue is simple.

In [17, Theorem 5.6], it has been shown that Algorithm 1 converges locally and linearly to μ_α, provided that

$$\frac{4\varepsilon}{(u_\alpha^* v_\alpha)^2\ \sigma_{n-1}(A - \mu_\alpha I)} < 1, \tag{2.3}$$

where $\sigma_{n-1}(\,\cdot\,)$ denotes the $(n-1)$th singular value, and Assumption 1 is satisfied.

Numerical experiments reported in [17] indicate a surprisingly robust global convergence behavior of Algorithm 1, even when (2.3) is not satisfied. In fact, for all practically relevant examples under consideration, Algorithm 1 converges to a point μ_α that is not only locally but also globally a right-most point of $\Lambda_\varepsilon(A)$. In particular, this yields $\alpha_\varepsilon(A) = \mathsf{Re}(\mu_\alpha)$.

3. Subspace methods. The basic idea of the approach proposed in this paper is to collect the iterates v_k produced by Algorithm 1 in a subspace $\mathcal{V}$ and locally approximate the pseudospectrum of A by its restriction to this subspace. More specifically, given an orthonormal basis $V \in \mathbb{C}^{n\times k}$ of $\mathcal{V}$ we consider the pseudospectrum of the (rectangular) matrix pencil $\hat{A} - z\hat{B} := AV - zV$. In analogy to (1.1), this can be defined as

$$\Lambda_\varepsilon(\hat{A}, \hat{B}) = \{z \in \mathbb{C} \colon \sigma_{\min}(\hat{A} - z\hat{B}) \le \varepsilon\}, \tag{3.1}$$

see also [40] and [36, Chapter 46]. In contrast to the ε-pseudospectrum of a square matrix, the ε-pseudospectrum of a rectangular matrix pencil can be empty. We will address this issue in Section 5.2.

The following lemma presents a monotonicity result, similar to [40, Theorem 2.2].

LEMMA 3.1. *Let U and V be orthonormal bases of subspaces $\mathcal{U}$ and $\mathcal{V}$ in $\mathbb{C}^{n\times n}$ with $\mathcal{U} \subset \mathcal{V}$. Then*

$$\Lambda_\varepsilon(AU, U) \subset \Lambda_\varepsilon(AV, V) \subset \Lambda_\varepsilon(A).$$

Proof. Let $z \in \Lambda_\varepsilon(AU, U)$. By definition (3.1),

$$\begin{aligned}\varepsilon \ge \sigma_{\min}(AU - zU) &= \min_{x\in\mathbb{C}^k,\, \|x\|=1} \|(AU - zU)x\| = \min_{u\in\mathcal{U},\, \|u\|=1} \|(A - zI)u\| \\ &\ge \min_{v\in\mathcal{V},\, \|v\|=1} \|(A - zI)v\| = \sigma_{\min}(AV - zV),\end{aligned}$$

which implies $z \in \Lambda_\varepsilon(AV, V)$ and thus shows the first inclusion. The second inclusion follows trivially from the first by noting that $\sigma_\varepsilon(A) = \sigma_\varepsilon(A \cdot I, I)$. □

Defining $\alpha_\varepsilon(\hat{A}, \hat{B})$ as the maximal real part of the ε-pseudospectrum of $\hat{A} - z\hat{B}$,

$$\alpha_\varepsilon(\hat{A}, \hat{B}) := \max\{\mathsf{Re}\, z \colon z \in \Lambda_\varepsilon(\hat{A}, \hat{B})\} = \max\{\mathsf{Re}\, z \colon \sigma_{\min}(\hat{A} - z\hat{B}) \le \varepsilon\}, \tag{3.2}$$

an important conclusion from Lemma 3.1 is that

$$\alpha_\varepsilon(AU, U) \le \alpha_\varepsilon(AV, V) \le \alpha_\varepsilon(A). \tag{3.3}$$

The following result characterizes when we can expect equality.

LEMMA 3.2. *Let $A \in \mathbb{C}^{n\times n}$ and V be an orthonormal basis of a subspace $\mathcal{V}$ of $\mathbb{C}^{n\times n}$. Then $\alpha_\varepsilon(AV, V) = \alpha_\varepsilon(A)$ if and only if $\mathcal{V}$ contains a vector v_α with the following property: v_α is a right singular vector belonging to $\sigma_{\min}(A - \mu_\alpha I)$ for some $\mu_\alpha \in \mathbb{C}$ with $\mathsf{Re}(\mu_\alpha) = \alpha_\varepsilon(A)$.*

Proof. The existence of a vector v_α with the described property implies that $\mu_\alpha \in \Lambda_\varepsilon(AV, V)$ and hence $\alpha_\varepsilon(AV, V) \ge \mathsf{Re}(\mu_\alpha) = \alpha_\varepsilon(A)$. Together with (3.3), this implies $\alpha_\varepsilon(AV, V) = \alpha_\varepsilon(A)$.

In the opposite direction, suppose that $\alpha_\varepsilon(AV, V) = \alpha_\varepsilon(A)$. By Lemma 3.1, this is only possible if $\Lambda_\varepsilon(AV, V)$ and $\Lambda_\varepsilon(A)$ have a common right-most point μ_α. Let x be a right singular vector belonging to $\sigma_{\min}(AV - \mu_\alpha V)$. Then $v_\alpha := Vx$ is a right singular vector belonging to $\sigma_{\min}(A - \mu_\alpha I)$. □

3.1. Stability of extraction procedure. Motivated by Lemma 3.2, we aim at constructing a subspace $\mathcal{V}$ that contains a good approximation to v_α in the sense that the distance

$$d(v_\alpha, \mathcal{V}) := \min\{\|v_\alpha - v\| \colon v \in \mathcal{V}, \|v\| = 1\} \tag{3.4}$$

is small. The distance measure (3.4) will be central in our convergence proofs and is closely related to the notion of gaps between subspaces; see [24, Section 1.2].

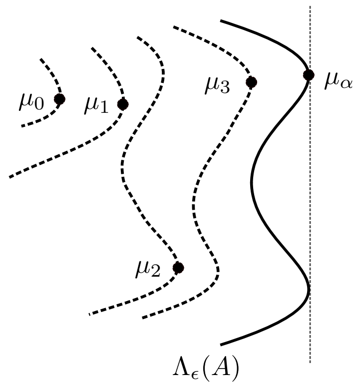

FIG. 3.1. *Jumping behavior of the right-most points μ_k for increasingly good approximations of the pseudospectrum.*

Given a subspace $\mathcal{V}$ with orthonormal basis V we extract $\alpha_\varepsilon(AV, V)$ as an approximation to $\alpha_\varepsilon(A)$. In this section, we prove the stability of this extraction procedure, that is, a small value for $d(v_\alpha, \mathcal{V})$ implies that $\alpha_\varepsilon(AV, V)$ is a good approximation to $\alpha_\varepsilon(A)$. We indeed show below that this approximation error is proportional to $d(v_\alpha, \mathcal{V})^2$, provided that Assumption 1 holds. However, for the convergence of subsequent iterates of our subspace method, it is important that not only the real parts but also the imaginary parts of the right-most points of $\Lambda_\varepsilon(AV, V)$ and $\Lambda_\varepsilon(A)$ are close. There are two obstacles to this. First, there could be several (globally) right-most points of $\Lambda_\varepsilon(A)$, and the right-most point of $\Lambda_\varepsilon(AV, V)$ could provide an approximation to a point different from the one associated with v_α as depicted in Figure 3.1. Second, even if the right-most point μ_α is unique, $\partial\Lambda_\varepsilon(A)$ may be arbitrarily flat along the imaginary axis.

To describe the influence of the shape of $\partial\Lambda_\varepsilon(A)$, recall from [8, Corollary 4.5] that under Assumption 1 we can parametrize $\partial\Lambda_\varepsilon(A)$ locally around μ_α as the curve $\mu\colon t \mapsto \mu_R(t) + \mathrm{i}\mu_I(t)$ with

$$\mu_I(t) = \mathsf{Im}(\mu_\alpha) + t, \qquad \mu_R(t) = \alpha_\varepsilon(A) - \gamma t^{2j} + O(t^{2j+1}), \quad \gamma > 0, \tag{3.5}$$

for some integer $j \geq 1$. The curvature of $\partial\Lambda_\varepsilon(A)$ in the direction of the imaginary axis is equal to γ when $j = 1$ and zero otherwise. Since the slope of $\partial\Lambda_\varepsilon(A)$ at μ_α is already vertical, a zero curvature increases the difficulty of linking the approximation of the imaginary part to the approximation of the real part. While the first obstacle from above can certainly be neglected once the approximation focuses on a particular part of the pseudospectrum, the second obstacle appears to be more severe. For convenience, we rule out both.

ASSUMPTION 2. *There is only one $\mu_\alpha \in \partial\Lambda_\varepsilon(A)$ satisfying $\mathsf{Re}(\mu_\alpha) = \alpha_\varepsilon(A)$. Moreover, the curvature of $\partial\Lambda_\varepsilon(A)$ at μ_α is not zero, that is, $j = 1$ in* (3.5).

Note that this assumption is also required by other algorithms for computing $\alpha_\varepsilon(A)$, for example, it is implicitly present in [12]. Also, there is numerical evidence that the convergence condition (2.3) for Algorithm 1 can only be satisfied if the curvature of $\partial\Lambda_\varepsilon(A)$ at μ_α is not zero. A detailed investigation of this connection is, however, beyond the scope of this paper.

THEOREM 3.3. *Let μ_α be a right-most point of $\Lambda_\varepsilon(A)$ and let v_α be a right singular vector belonging to $\sigma_{\min}(A - \mu_\alpha I)$. Moreover, consider a subspace $\mathcal{V}$ with an* *orthonormal basis V. If Assumption 1 holds then*

$$\alpha_\varepsilon(A) - \alpha_\varepsilon(AV, V) = O(d(v_\alpha, \mathcal{V})^2). \tag{3.6}$$

If in addition Assumption 2 holds then any right-most point μ_V of $\Lambda_\varepsilon(AV, V)$ satisfies

$$|\mu_\alpha - \mu_V| = O(d(v_\alpha, \mathcal{V})). \tag{3.7}$$

Proof. Because of

$$\alpha_\varepsilon(A) - \alpha_\varepsilon(AV, V) = \alpha_\varepsilon(A - \mu_\alpha I) - \alpha_\varepsilon((A - \mu_\alpha I)V, V),$$

we may assume w.l.o.g. that $\mu_\alpha = 0$ for the rest of the proof.

By compactness, there exists $v \in \mathcal{V}$ with $\|v\| = 1$ such that $d(v_\alpha, \mathcal{V}) = \|v - v_\alpha\| = \|\delta\|$, where we set $\delta := v - v_\alpha$. Inequality (3.3) and Lemma A.1 imply that

$$\alpha_\varepsilon(AV, V) \geq \alpha_\varepsilon(Av, v) = \mathsf{Re}(v^*Av) + \sqrt{\varepsilon^2 - \|Av\|^2 + |v^*Av|^2}. \tag{3.8}$$

To prove the first part of the theorem, we need to compare the quantity on the right with $\alpha_\varepsilon(A) = \alpha_\varepsilon(Av_\alpha, v_\alpha) = 0$. By a first-order expansion, it holds that

$$v^*Av = v_\alpha^*Av_\alpha + \delta^*Av_\alpha + v_\alpha^*A\delta + O(\|\delta\|^2).$$

Assumption 1 implies that $v_\alpha^*Av_\alpha$ is real and negative, see (2.2). Consequently, we also have

$$|v^*Av|^2 = (v_\alpha^*Av_\alpha)^2 + 2\, v_\alpha^*Av_\alpha \cdot \mathsf{Re}(\delta^*Av_\alpha + v_\alpha^*A\delta) + O(\|\delta\|^2).$$

Moreover,

$$\|Av\|^2 = \|Av_\alpha\|^2 + 2\,\mathsf{Re}(\delta^*A^*Av_\alpha) + O(\|\delta\|^2) = \varepsilon^2 + 2\varepsilon^2\,\mathsf{Re}(\delta^*v_\alpha) + O(\|\delta\|^2),$$

where we used $\|Av_\alpha\| = \varepsilon$ and $A^*Av_\alpha = \varepsilon^2 v_\alpha$. Using that $\|v\| = \|v_\alpha + \delta\| = 1$ implies $\mathsf{Re}(\delta^*v_\alpha) = O(\|\delta\|^2)$, we obtain

$$\|Av\|^2 = \varepsilon^2 + O(\|\delta\|^2).$$

Plugging the derived relations into (3.8) and taking a Taylor expansion of the square root gives

$$\begin{aligned}
\alpha_\varepsilon(Av, v) &= v_\alpha^*Av_\alpha + \mathsf{Re}(\delta^*Av_\alpha + v_\alpha^*A\delta) \\
&\quad + \sqrt{|v_\alpha^*Av_\alpha|^2 + 2\, v_\alpha^*Av_\alpha \cdot \mathsf{Re}(\delta^*Av_\alpha + v_\alpha^*A\delta)} + O(\|\delta\|^2) \\
&= v_\alpha^*Av_\alpha + \mathsf{Re}(\delta^*Av_\alpha + v_\alpha^*A\delta) \\
&\quad + \sqrt{|v_\alpha^*Av_\alpha|^2} + \frac{2\, v_\alpha^*Av_\alpha}{2\sqrt{|v_\alpha^*Av_\alpha|^2}}\,\mathsf{Re}(\delta^*Av_\alpha + v_\alpha^*A\delta) + O(\|\delta\|^2) = O(\|\delta\|^2),
\end{aligned}$$

which completes the proof of (3.6).

To prove the second part, we recall that Assumption 2 implies that the boundary of $\Lambda_\varepsilon(A)$ is asymptotically described by the parabola (3.5) with $j = 1$. For sufficiently small $\|\delta\|$, μ_V must be contained in this part of the ε-pseudospectrum and hence $|\mathsf{Im}(\mu_V)| \leq |\mu_I(t)| = |t|$ with t satisfying

$$\alpha_\varepsilon(AV, V) = \mu_R(t) = -\gamma t^2 + O(t^3).$$

Combined with the result from the first part, this shows $|\mathsf{Im}(\mu_V)| = O(d(v_\alpha, \mathcal{V}))$ and this completes the proof of the second part. ∎

REMARK 3.4. *If the second part of Assumption 2 is not satisfied, the parametrization (3.5) holds for some integer $j \geq 2$. A straightforward modification of the proof of Theorem 3.3 then leads to $|\mu_\alpha - \mu_V| = O\big(d(v_\alpha, \mathcal{V})^{1/j}\big)$.*

3.2. Basic methods. Algorithm 2 realizes the subspace extraction discussed above. For the subspace expansion, we essentially perform one step of Algorithm 1 and add the obtained eigenvector to the subspace.

Algorithm 2 Basic subspace method with eigenvectors

Input: Matrix $A \in \mathbb{C}^{n\times n}$, perturbation level $\varepsilon > 0$.
Output: Approximation μ_α to a locally right-most point of $\Lambda_\varepsilon(A)$.
  Compute right-most eigenvalue λ_0 and normalized right eigenvector $\hat{v}_0$ of A.
  Set $V_1 = [\hat{v}_0]$.
  for $k = 1, 2, \ldots$ until converged **do**
    Compute right-most point μ_k of $\Lambda_\varepsilon(AV_k, V_k)$.
    Compute left/right singular vectors u_k and v_k belonging to $\sigma_{\min}(A - \mu_k I)$.
    Set $\triangle_k = -u_k v_k^*$.
    Compute right-most eigenvalue λ_k and right eigenvector $\hat{v}_k$ of $A + \varepsilon\triangle_k$.
    Compute $V_{k+1} = \mathrm{orth}([V_k\ \hat{v}_k])$.
  end for
  return $\mu_\alpha = \mu_k$

Figure 3.2 illustrates the convergence of Algorithm 2 applied to the Grcar matrix for $\varepsilon = 10^{-1}$. In particular, note that the reduced pseudospectrum $\Lambda_\varepsilon(AV_k, V_k)$ provides an increasingly good approximation to the right-most part of $\Lambda_\varepsilon(A)$.

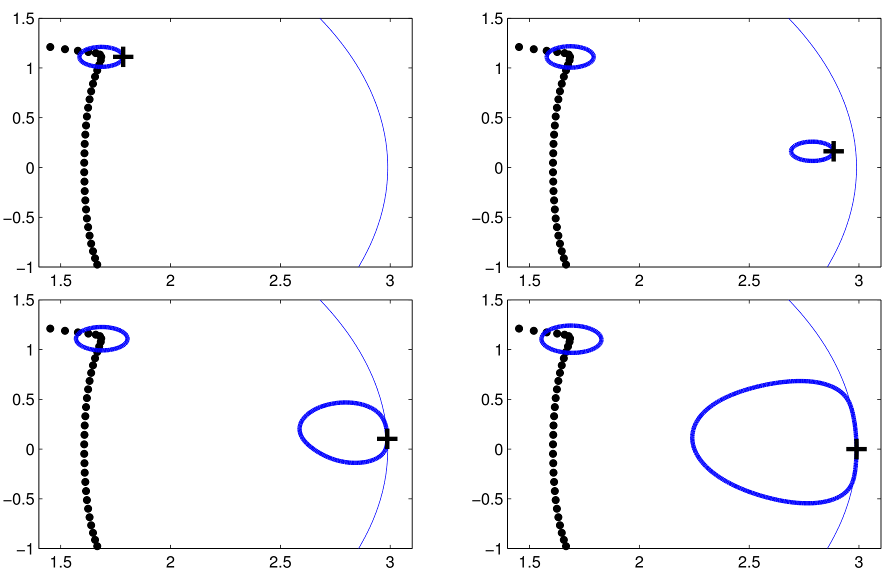

Fig. 3.2. *Behavior of Algorithm 2 applied to the* 100×100 *Grcar matrix with* $\varepsilon = 10^{-1}$. *The thick blue lines denote the boundaries of* $\Lambda_\varepsilon(AV_k, V_k)$ *for iterations* $k = \{1, 2, 4, 8\}$ *(from left to right, top to bottom) with the computed right-most point denoted by "+". The thin blue line denotes the boundary of* $\Lambda_\varepsilon(A)$.

Figure 3.3 shows a similar behavior for the Orr–Sommerfeld matrix from Section 5.3. To test the robustness of Algorithm 2, we have intentionally chosen an eigenvalue λ_0 that is not the right-most eigenvalue in the first step. As a side effect, the right-most point of $\Lambda_\varepsilon(AV_2, V_2)$ is incorrectly computed, due to difficulties explained in Section 5.2. Nevertheless, the pseudospectra of subsequent pencils $\Lambda_\varepsilon(AV_k, V_k)$ still approach the right boundary of $\Lambda_\varepsilon(A)$.

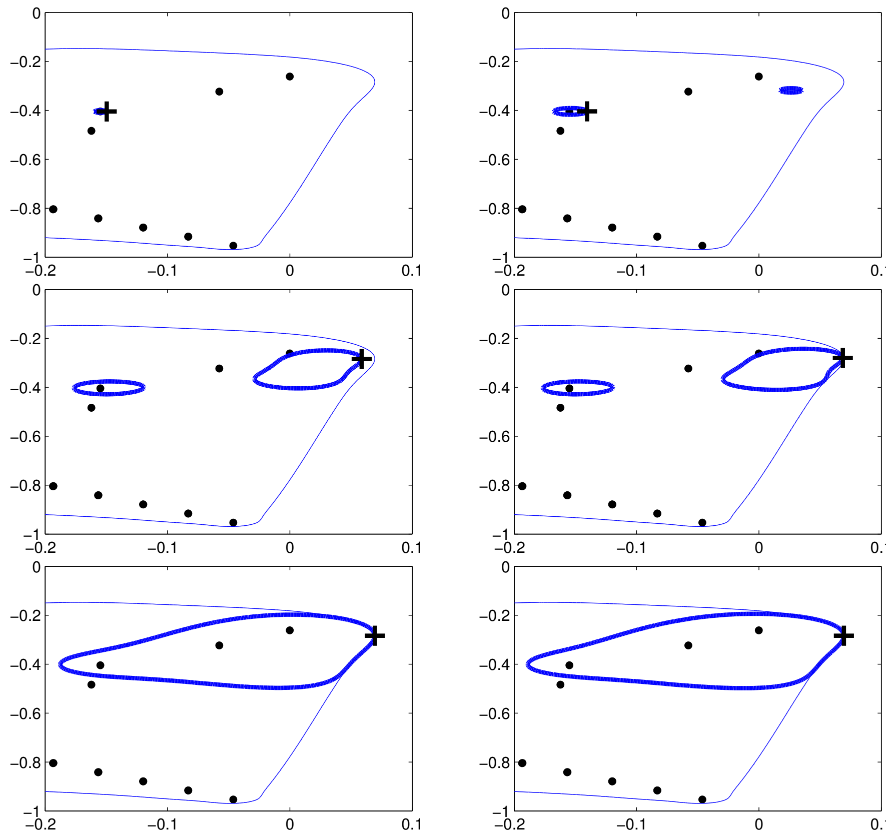

FIG. 3.3. *Same setup as in Figure 3.2 but now applied to the* 99×99 *Orr–Sommerfeld matrix with* $\varepsilon = 10^{-2.5}$ *and iterations* $k = \{1, 2, 3, 4, 6, 8\}$*. Furthermore, Algorithm 2 is started from the 6th right-most eigenvalue.*

The use of both singular vectors and eigenvectors makes Algorithm 2 harder to analyze. Moreover, computing eigenvalues and eigenvectors of a nonsymmetric matrix can be significantly more expensive than computing singular values and singular vectors. In the following, we propose another algorithm that has a similar local convergence behavior without relying on eigenvectors. To this end, suppose that Algorithm 2 is close to convergence in the sense that

$$\delta_k := d(v_\alpha, \mathsf{span}(V_k))$$

is small. Then, by Theorem 3.3, the real and imaginary parts of $\mu_k - \mu_\alpha$ are of order δ_k^2 and δ_k, respectively. Under Assumption 1, ε is a simple singular value of $A - \mu_\alpha I$ and hence a perturbation expansion [35, Theorem 3.3.4] yields

$$\begin{aligned}\tilde{\varepsilon} &:= \sigma_{\min}(A - \mu_k I) = \sigma_{\min}(A - \mu_\alpha I + (\mu_\alpha - \mu_k)I) \\ &= \varepsilon + u_\alpha^* v_\alpha \cdot \mathsf{Re}(\mu_\alpha - \mu_k) + O(|\mu_\alpha - \mu_k|^2) = \varepsilon + O(\delta_k^2).\end{aligned}$$

This implies

$$\|(A - \mu_k I - \varepsilon u_k v_k^*) v_k\| \leq \|(A - \mu_k I) v_k - \tilde{\varepsilon} u_k\| + |\tilde{\varepsilon} - \varepsilon| = |\tilde{\varepsilon} - \varepsilon| = O(\delta_k^2).$$

Table 3.1
Local convergence behavior of Algorithms 2 and 3 applied to the Grcar matrix for $\varepsilon = 10^{-2}$. Both algorithms have been initialized with $\mu_0 = \mu_\alpha - (1+\mathrm{i})/10$.

	Algorithm 2 (with EIG)		Algorithm 3 (only SVD)	
k	$\mathsf{Re}(\mu_k)$	Abs. error	$\mathsf{Re}(\mu_k)$	Abs. error
1	*2.73*6342528409585	3.57×10^{-03}	*2*.639914450044436	1.00×10^{-01}
2	*2.73*8491391291922	1.42×10^{-03}	*2*.649395239360337	9.05×10^{-02}
3	*2.73*8493713994053	1.42×10^{-03}	*2.7*14104187502880	2.58×10^{-02}
4	*2.739*123965482084	7.90×10^{-04}	*2.73*8708533085343	1.21×10^{-03}
5	*2.739*343902454546	5.71×10^{-04}	*2.739*611065118983	3.03×10^{-04}
6	*2.7399*08003532771	6.45×10^{-06}	*2.73991*2112789628	2.34×10^{-06}
7	*2.739914450004*3420	1.02×10^{-12}	*2.7399144500*39453	4.98×10^{-12}
8	*2.7399144500444*53	1.60×10^{-14}	*2.7399144500444*55	1.82×10^{-14}

In other words, (μ_k, v_k) is an approximate eigenpair of $A - \varepsilon u_k v_k^*$. Note that μ_k is simple and under Assumption 2 it will be a good approximation to the right-most eigenvalue for sufficiently small δ_k. In particular, this implies that the eigenvector $\hat{v}_k$ computed in Algorithm 2 satisfies

$$\|v_k - \hat{v}_k\| = O(\delta_k^2), \tag{3.9}$$

provided that $\hat{v}_k$ is suitably normalized. This suggests that replacing $\hat{v}_k$ by v_k in Algorithm 2 will have little effect on the local convergence. These considerations result in Algorithm 3.

Algorithm 3 Basic subspace method with singular vectors

Input: Matrix $A \in \mathbb{C}^{n\times n}$, perturbation level $\varepsilon > 0$, starting value $\mu_0 \in \mathbb{C}$.
Output: Approximation μ_α to a locally right-most point of $\Lambda_\varepsilon(A)$.
  Compute right singular vector v_0 belonging to $\sigma_{\min}(A - \mu_0 I)$.
  Set $V_1 = [v_0]$.
  for $k = 1, 2, \ldots$ until converged **do**
    Compute right-most point μ_k of $\Lambda_\varepsilon(AV_k, V_k)$.
    Compute right singular vector v_k belonging to $\sigma_{\min}(A - \mu_k I)$.
    Compute $V_{k+1} = \mathrm{orth}([V_k\ v_k])$.
  end for
  return $\mu_\alpha = \mu_k$

Table 3.1 illustrates the convergence of Algorithms 2 and 3, confirming that their behavior is quite similar. Both appear to converge locally quadratically.

3.3. A hybrid method. Computing the smallest singular value can be significantly less expensive than computing the right-most eigenvalue of a matrix, since the right-most eigenvalue may have large unknown imaginary part necessitating some sort of validation whereas for the smallest singular value simply inverse iteration suffices. This means that Algorithm 3 is most likely to be cheaper per iteration than Algorithm 2. However, the global convergence of Algorithm 3 is unfavorable. This is illustrated in Figure 3.4, where we have depicted typical paths of the iterates when applying Algorithms 2 and 3 to the Grcar matrix (for $\varepsilon = 10^{-4}$) and the Orr–Sommerfeld

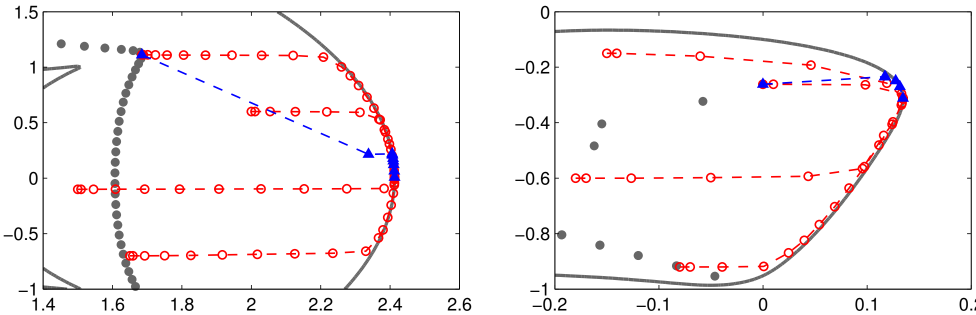

FIG. 3.4. *Iterates of Algorithms 2 (blue triangles) and 3 (red open discs) for the Grcar (left figure, $\varepsilon = 10^{-4}$) and the Orr–Sommerfeld (right figure, $\varepsilon = 10^{-2}$) matrices of size $n = 100$. The eigenvalues are indicated by grey discs and $\Lambda_\varepsilon(A)$ is visible in a grey contour line.*

matrix (for $\varepsilon = 10^{-2}$). Both algorithms are started at the right-most eigenvalue and Algorithm 3 is additionally started at three random points for each matrix. Observe that the eigenvalue-based Algorithm 2 takes larger steps in the beginning and more quickly approaches the region of local convergence. Close to the right-most point, however, both algorithms exhibit essentially the same convergence behavior.

Inspired by the good global convergence behavior of Algorithm 2 and the usually cheaper cost per iteration of Algorithm 3, we propose in Algorithm 4 a hybrid strategy: In the beginning, as long as

$$|\sigma_{\min}(A - \mu_k I) - \varepsilon| \geq \rho\,\varepsilon, \tag{3.10}$$

with $\rho > 0$ a tolerance (we use $\rho = 0.1$ by default), we know that the reduced pseudospectrum $\Lambda_\varepsilon(AV_k, V_k)$ is still far away from $\Lambda_\varepsilon(A)$. Hence, we start with Algorithm 2 and switch to Algorithm 3 when (3.10) does not hold. The numerical experiments in Sections 5.3 and 5.4 demonstrate that such a hybrid approach is indeed considerably faster than using Algorithm 2 alone.

Algorithm 4 A hybrid strategy based on Algorithms 2 and 3

Input: Matrix $A \in \mathbb{C}^{n\times n}$, perturbation level $\varepsilon > 0$.
Output: Approximation μ_α to a locally right-most point of $\Lambda_\varepsilon(A)$.
  Compute right-most eigenvalue λ_0 and right eigenvector $\hat{v}_0$ of A.
  Set $V_1 = [\hat{v}_0]$.
  for $k = 1, 2, \ldots$ until converged **do**
    Compute right-most point μ_k of $\Lambda_\varepsilon(AV_k, V_k)$.
    Compute right singular vector v_k belonging to $\sigma_{\min}(A - \mu_k I)$.
    if condition (3.10) holds **then**
      Compute the left singular vector u_k belonging to $\sigma_{\min}(A - \mu_k I)$.
      Compute right-most eigenvalue λ_k and right eigenvector $\hat{v}_k$ of $A - \varepsilon\, u_k v_k^*$.
    else
      Set $\hat{v}_k = v_k$.
    end if
    Compute $V_{k+1} = \mathrm{orth}([V_k\ \hat{v}_k])$.
  end for
  return $\mu_\alpha = \mu_k$

---

4. Convergence analysis. In this section, we study the convergence properties of Algorithms 2 and 3. Since Algorithm 4 has the same local behavior as Algorithm 3, there is no need to discuss the convergence of the hybrid strategy separately.

4.1. Stagnation. Both subspace methods converge in at most n steps: either a basis $V_n \in \mathbb{C}^{n\times n}$ of the full space $\mathbb{C}^n$ is produced and hence $\mu_n = \mu_\alpha$, or stagnation has occurred in an earlier step. Lemma 4.1 below characterizes the latter situation for Algorithm 3.

We call a boundary point $\mu \in \partial\Lambda_\varepsilon(A)$ *stationary* if $\partial\Lambda_\varepsilon(A)$ is *vertical* at μ, that is[1], locally the vertical line does not intersect $\partial\Lambda_\varepsilon(A)$ except at μ. Clearly, every locally right-most point is stationary. However, also local minima and saddle points of $t \mapsto \mathsf{Re}(\gamma(t))$, with $\gamma(t)$ a curve in $\mathbb{C}$ tracing $\partial\Lambda_\varepsilon(A)$, are stationary.

LEMMA 4.1. *Let the right-most points $\mu_1, \mu_2, \dots$ in Algorithm 3 be uniquely defined. Suppose that Algorithm 3 stagnates and let $k < n$ denote the smallest integer such that $\mu_k = \mu_{k+1}$. Then μ_k is a boundary point of $\Lambda_\varepsilon(A)$ and*

1. *μ_k is an isolated point of $\Lambda_\varepsilon(AV_k, V_k)$; or*
2. *μ_k is stationary.*

Proof. Suppose that μ_k is not a boundary point, i.e., $\sigma_{\min}(A-\mu_k I) < \varepsilon$. Then the corresponding right singular vector v_k satisfies $\|Av_k - (\mu_k+\delta)v_k\| \le \varepsilon$ for sufficiently small $\delta > 0$. Hence, the pseudospectrum $\Lambda_\varepsilon(Av_k, v_k) \subset \Lambda_\varepsilon(AV_{k+1}, V_{k+1})$ contains points with real part larger than μ_k. This contradicts stagnation.

Suppose now that $\mu_k \in \partial\Lambda_\varepsilon(A)$ but μ_k is neither isolated nor stationary. Then $\partial\Lambda_\varepsilon(AV_k, V_k)$ is not differentiable at μ_k, because otherwise $\partial\Lambda_\varepsilon(AV_k, V_k)$ would be vertical and not touch but intersect $\partial\Lambda_\varepsilon(A)$ at μ_k, contradicting that $\Lambda_\varepsilon(AV_k, V_k)$ is always included in $\Lambda_\varepsilon(A)$. This implies that the smallest singular value ε of $AV_k - \mu_k V_k$ has multiplicity at least two. By dimension counting, there is a right singular vector w_k of $AV_k - \mu_k V_k$ belonging to ε such that $V_k w_k \in \mathsf{span}(V_{k-1})$. Therefore $\mu_k \in \Lambda_\varepsilon(AV_{k-1}, V_{k-1})$. By monotonicity and the uniqueness of μ_{k-1}, this implies $\mu_{k-1} = \mu_k$ and therefore contradicts the assumption. ∎

Note that Lemma 4.1 only shows necessary conditions for stagnation. Because of the nonlocal nature of subspace methods, it seems to be difficult to provide sufficient conditions, except for trivial situations. For a similar reason, it seems to be difficult to characterize the stagnation of Algorithm 2.

4.2. Local convergence. The numerical experiments strongly suggest that Algorithms 2 and 3 converge faster than linearly, see in particular Table 3.1. The analysis below indeed shows local superlinear convergence for both algorithms, provided that certain assumptions are met. However, based on the numerical evidence, we conjecture that both algorithms actually converge locally *quadratically*. Unfortunately, our proof technique does not seem to admit such a stronger convergence result.

THEOREM 4.2. *Let Assumptions 1 and 2 hold for the rightmost point μ_α of $\Lambda_\varepsilon(A)$. Consider any $\mu_1, \mu_2 \in \mathbb{C}$ sufficiently close to μ_α such that $\mathsf{Re}(\mu_1) \le \mathsf{Re}(\mu_2) \le \mathsf{Re}(\mu_\alpha)$ and*

$$|\mu_\alpha - \mu_2| < \beta\,|\mu_\alpha - \mu_1| \qquad \textit{for some fixed} \quad 0 < \beta < 1. \tag{4.1}$$

Let $v_k \in \mathbb{C}^n$ be right singular vectors belonging to $\sigma_{\min}(A - \mu_k I)$ for $k \in \{1, 2\}$, and choose an orthonormal basis V for $\mathcal{V} = \mathrm{span}\{v_1, v_2\}$. Then

$$\alpha_\varepsilon(A) - \alpha_\varepsilon(AV, V) = O\big(|\alpha_\varepsilon(A) - \mathsf{Re}(\mu_1)|^2\big). \tag{4.2}$$

[1]This definition includes the case when $\partial\Lambda_\varepsilon(A)$ is not differentiable at μ. It generalizes directly to the rectangular pencil case $\partial\Lambda_\varepsilon(AV, V)$.

Proof. By shifting the matrix A, we may assume w.l.o.g. that $\mu_\alpha = 0$. Let us denote $\mu_1 = x_1 + \mathrm{i}y_1$ and $\mu_2 = x_2 + \mathrm{i}y_2$ with $x_1, x_2, y_1, y_2 \in \mathbb{R}$. Then, according to Assumption 2, we have $y_1^2 = O(x_1)$ and $y_2^2 = O(x_2)$.

The idea of the proof is to "remove" the influence of the imaginary parts of μ_1, μ_2 and then apply Theorem 3.3. However, we need to separately address the situation for which the imaginary parts are already sufficiently small. For this purpose, define $C = \sqrt{2}\beta/\sqrt{1-\beta^2}$. If $|y_1| \le C|x_1|$ then $|\mu_\alpha - \mu_1| = O(x_1)$. By Assumption 1, $\sigma_{\min}(A)$ is simple and a standard perturbation result for singular vectors [37] implies $d(v_\alpha, v_1) = O(x_1)$, where v_α is a right singular vector belonging to $\sigma_{\min}(A)$. Clearly, $d(v_\alpha, \mathcal{V}) \le d(v_\alpha, \mathrm{span}\{v_1\})$ and thus (4.2) follows directly from Theorem 3.3.

From now on, assume $|y_1| > C|x_1|$ and choose $\tilde{\beta}^2 = \frac{1}{2}(1+\beta^2)$ satisfying $\beta < \tilde{\beta} < 1$. Then inequality (4.1) implies

$$\begin{aligned} y_2^2 &< \tilde{\beta}^2 y_1^2 - \tfrac{1}{2}(1-\beta^2)y_1^2 + \beta^2 x_1^2 \\ &< \tilde{\beta}^2 y_1^2 - [\tfrac{1}{2}(1-\beta^2)C^2 - \beta^2]x_1^2 = \tilde{\beta}^2 y_1^2. \end{aligned}$$

Hence,

$$|y_2 - y_1| > (1-\tilde{\beta})|y_1|. \tag{4.3}$$

Let the vector-valued function $\mu \mapsto v(\mu)$ denote a right singular vector belonging to $\sigma_{\min}(A-\mu I) =: \varepsilon(\mu)$, with $v(\mu_\alpha) = v(0) = v_\alpha$. Of course, $v(\mu)$ is also an eigenvector belonging to the smallest eigenvalue $\varepsilon(\mu)^2$ of the Hermitian matrix $H(\mu) := (A - \mu I)^*(A-\mu I)$. Since this eigenvalue is simple for $\mu = 0$, it follows from [24, Sec. II.4.6] that $\varepsilon(\mu)^2$ and $v(\mu)$ are real analytic in the real and imaginary parts of μ for all μ sufficiently close to 0, provided that v is suitably normalized. In particular, we have the following perturbation expansion [32, Thm. 1]:

$$\begin{aligned} v(\mu_1) &= v(x_1) - \big(H(x_1) - \varepsilon(x_1)^2 I\big)^+ (H(\mu_1) - H(x_1))v(x_1) + O(y_1^2) \\ &= v(x_1) - \mathrm{i}y_1\big(H(x_1) - \varepsilon(x_1)^2 I\big)^+ (A - A^*)v(x_1) + O(y_1^2), \end{aligned} \tag{4.4}$$

where $(\cdot)^+$ denotes the Moore-Penrose pseudoinverse of a matrix. Since the eigenvalue zero of $H(0) - \varepsilon(0)^2 I = A^*A - \varepsilon^2 I$ is simple, the rank of $H(\mu) - \varepsilon(\mu)^2 I$ remains constant for sufficiently small μ. Consequently, Wedin's perturbation result [38, Thm. 2.1] yields $\big(H(x_1) - \varepsilon(x_1)^2 I\big)^+ = \big(A^*A - \varepsilon^2 I\big)^+ + O(x_1)$. Inserted into (4.4), and using once again a perturbation expansion of $v(x_1)$, this shows

$$v(\mu_1) = v_\alpha - \mathrm{i}y_1\big(A^*A - \varepsilon^2 I\big)^+ (A - A^*)v_\alpha + O(y_1^2) + O(x_1).$$

Analogously,

$$v(\mu_2) = v_\alpha - \mathrm{i}y_2\big(A^*A - \varepsilon^2 I\big)^+ (A - A^*)v_\alpha + O(y_2^2) + O(x_2).$$

A particular linear combination of these two expressions leads to

$$\tilde{v} := -\frac{y_2}{y_1 - y_2}v(\mu_1) + \frac{y_1}{y_1 - y_2}v(\mu_2) = v_\alpha + O(y_1^2) + O(x_1) = v_\alpha + O(x_1),$$

where we have used (4.3), $|y_2| < |y_1|$, $|x_2| \le |x_1|$, and $y_1^2 = O(x_1)$. Since $\tilde{v} \in \mathcal{V}$, we obtain $d(v_\alpha, \mathcal{V}) = O(x_1)$ and (4.2) follows once again from Theorem 3.3. ☐

Theorem 4.2 can be applied to two subsequent iterates μ_k, μ_{k+1} of Algorithm 3, provided that these iterates converge at least linearly to μ_α. By the monotonicity of the rightmost point (Lemma 3.1), it follows from (4.2) that

$$\alpha_\varepsilon(A) - \mathsf{Re}(\mu_{k+2}) = O\big(|\alpha_\varepsilon(A) - \mathsf{Re}(\mu_k)|^2\big).$$

In other words, the real parts of μ_k converge at least superlinearly with order $\sqrt{2}$ to $\alpha_\varepsilon(A)$; see [33, Thm. 2.1]. This result directly extends to Algorithm 2. Replacing the singular vector v_k by the eigenvector $\hat{v}_k$ introduces another small perturbation (3.9), which can be easily incorporated into the proof of Theorem 4.2. Thus, Algorithm 2 also converges superlinearly to $\alpha_\varepsilon(A)$ under the same assumptions.

The assumed local linear convergence of Algorithms 2 and 3 is not unreasonable: Algorithm 2 is a subspace acceleration of Algorithm 1 and for the latter algorithm local linear convergence has been shown in [17]. For Algorithm 3, we can again use a second-order perturbation argument based on (3.9).

5. Implementation details and numerical experiments. In this section, we detail some implementation aspects of the proposed subspace algoirthms and report on numerical results for a suite of dense and sparse test problems. All experiments were done with MATLAB version R2010b on an Intel Core2 2.66 GHz CPU.

5.1. Stopping condition. We use the stopping condition from [17]: the iteration is stopped when $k > 1$ and

$$|\mathsf{Re}(\mu_k) - \mathsf{Re}(\mu_{k-1})| < \eta \max(1, |\mathsf{Re}(\mu_{k-1})|), \tag{5.1}$$

where η is a user-specified tolerance. For all experiments in this section, we use $\eta = 10^{-12}$.

5.2. Pseudospectral abscissa for rectangular matrix pencils. Each iteration of our subspace methods needs to determine the right-most point of $\Lambda_\varepsilon(AV_k, V_k)$ for the rectangular matrix pencil $AV_k - zV_k \in \mathbb{C}^{n\times k}$, that is,

$$\max\{\mathsf{Re}(z) \colon z \in \Lambda_\varepsilon(AV_k, V_k)\} = \max\{\mathsf{Re}(z) \colon \sigma_{\min}(AV_k - zV_k) \le \varepsilon\}. \tag{5.2}$$

Several additional complications may occur in the case of rectangular matrix pencils, for example, the pseudospectrum could be empty. Fortunately, in Algorithm 2, this situation is avoided since the eigenvector $\hat{v}_0$ of A is contained in all subspaces. Hence, the pseudospectrum $\Lambda_\varepsilon(A\hat{v}_0, \hat{v}_0)$, which is a ball of radius ε around λ_0, is a subset of $\Lambda_\varepsilon(AV_k, V_k)$ for all k.

As a first step to address (5.2), we reduce the size of the pencil $AV_k - zV_k$ when $n \ge 2k$ by computing a reduced QR decomposition $\big[V_k,\, AV_k\big] = Q\,\big[\tilde{B},\, \tilde{A}\big]$. This results in the pencil

$$\tilde{A} - z\tilde{B} =: \begin{bmatrix} \tilde{A}_1 \\ \tilde{A}_2 \end{bmatrix} - z \begin{bmatrix} \tilde{B}_1 \\ 0 \end{bmatrix} \in \mathbb{C}^{2k\times k} \tag{5.3}$$

with the property that $\Lambda_\varepsilon(AV_k, V_k) = \Lambda_\varepsilon(\tilde{A}, \tilde{B})$; see also [40, 3]. To this reduced pencil, we apply a variant of the criss-cross algorithm from [8].

Our variant of the criss-cross algorithm repeatedly performs a series of horizontal and vertical searches in the complex plane to find intersections $z = x + \mathrm{i}y$ with the boundary of $\Lambda_\varepsilon(\tilde{A}, \tilde{B})$. By a straightforward generalization of Lemmas 2.1 and 2.5

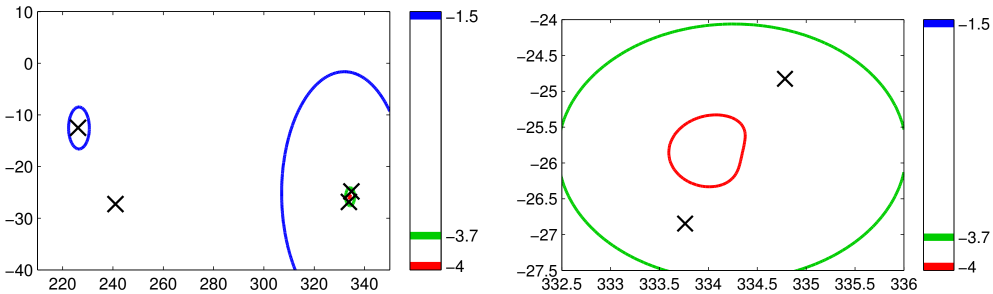

FIG. 5.1. *Left figure:* $\Lambda_\varepsilon(AV_5, V_5)$ *for Algorithm 2 applied to the matrix* `chebspec(100)` *from Section 5.3. The crosses denote* $\Lambda(\tilde{A}_1, \tilde{B}_1)$. *Right figure: Zoom of right-most component of* $\Lambda_\varepsilon(AV_5, V_5)$, *showing that it does not contain any eigenvalue of* $\tilde{A}_1 - z\tilde{B}_1$ *for* $\varepsilon = 10^{-4}$.

in [8], these intersections can be found by computing the imaginary eigenvalues of certain matrix pencils.

LEMMA 5.1. *The following equivalences hold:*

$$\varepsilon \in \sigma(\tilde{A} - (x + \mathrm{i}y)\tilde{B}) \quad \Longleftrightarrow \quad \mathrm{i}y \in \Lambda\left(\begin{bmatrix} x\tilde{B}^* - \tilde{A}^* & \varepsilon I \\ -\varepsilon I & \tilde{A} - x\tilde{B} \end{bmatrix}, \begin{bmatrix} \tilde{B}^* & 0 \\ 0 & \tilde{B} \end{bmatrix}\right)$$
$$\Longleftrightarrow \quad \mathrm{i}x \in \Lambda\left(\begin{bmatrix} -y\tilde{B}^* + \mathrm{i}\tilde{A}^* & \varepsilon I \\ -\varepsilon I & \mathrm{i}\tilde{A} + y\tilde{B} \end{bmatrix}, \begin{bmatrix} \tilde{B}^* & 0 \\ 0 & \tilde{B} \end{bmatrix}\right).$$

Based on this lemma, a criss-cross algorithm for $\Lambda_\varepsilon(\tilde{A}, \tilde{B})$ can be performed in virtually the same manner as in [8, Algorithm 3.1]. We omit the details but point out a difficulty related to rectangular matrix pencils.

To find the right-most point, the rectangular pencil variant of the criss-cross algorithm needs to be started from inside the right-most component of $\Lambda_\varepsilon(AV_k, V_k)$, in contrast to the criss-cross algorithm for $\Lambda_\varepsilon(\tilde{A}, \tilde{B})$ which has no such restriction. By the monotonicity of the pseudospectra $\Lambda_\varepsilon(AV_k, V_k)$, the previous right-most point of $\Lambda_\varepsilon(AV_{k-1}, V_{k-1})$ is guaranteed to be in $\Lambda_\varepsilon(AV_k, V_k)$. Unfortunately, this point may be in a component different from the right-most one. Motivated by the observation that for any $\lambda \in \Lambda_\varepsilon(AV_k, V_k)$, there is an ε-pseudoeigenpair (λ, v) satisfying [40]

$$\|(\tilde{A} - \lambda\tilde{B})v\|^2 = \|(A_1 - \lambda B_1)v\|^2 + \|B_2 v\|^2 \le \varepsilon^2, \tag{5.4}$$

with $\tilde{A}_1, \tilde{B}_1$ from (5.3), we also consider the points that make $\|(A_1 - \lambda B_1)v\|$ zero, that is,

$$\big\{\lambda \in \Lambda(\tilde{A}_1, \tilde{B}_1) : \ \sigma_{\min}(\tilde{A} - \lambda\tilde{B}) \le \varepsilon\big\}. \tag{5.5}$$

Among these candidate points, we take the right-most one as the initial guess. A similar heuristic is also used in [3] and it turns out to work well in our experiments too. However, it may fail on difficult problems since $\Lambda_\varepsilon(AV_k, V_k)$ may have a disconnected component that does not contain any λ satisfying (5.5). Such a situation is depicted in Figure 5.1.

5.3. Numerical experiments with small-scale problems. Our small-scale test problems essentially consist of the matrices from EigTool that are used in [17, Sec. 8]. We have omitted or modified examples of size up to 10, as using a subspace

Table 5.1
Computational results on a subset of small-scale problems for $\varepsilon = 10^{-4}$. All times in seconds.

	Algorithm 2 (always EIG)			Algorithm 4 (only EIG if (3.10))			
Problem	Rel. error	Iters	Time	Rel. error	Iters	Eig	Time
`airy(100)`	3×10^{-15}	5	0.342	2×10^{-16}	4	2	**0.121**
`boeing('S')`	4×10^{-19}	15	0.336	2×10^{-19}	14	3	**0.202**
`chebspec(100)`	8×10^{-14}	12	0.640	1×10^{-13}	13	3	**0.308**
`gaussseidel(100,'C')`	2×10^{-15}	3	0.207	1×10^{-15}	3	1	**0.049**
`grcar(100)`	1×10^{-14}	13	0.814	1×10^{-15}	13	3	**0.372**
`kahan(100)`	3×10^{-13}	5	0.115	3×10^{-13}	5	2	**0.067**
`orrsommerfeld(100)`	2×10^{-13}	9	0.398	2×10^{-13}	6	3	**0.149**
`randomtri(100)`	7×10^{-05}	(6)	(0.134)	1×10^{-12}	11	3	**0.216**

approach makes little sense for such small examples. All eigenvalue and SVD calculations were performed by calls to the Matlab commands `eig` and `svd`, respectively.

First, we compare the eigenvalue-based approach (Algorithm 2) with the hybrid approach (Algorithm 4). Table 5.1 gives a representative subset of the obtained computational results for matrices up to size 100 and for $\varepsilon = 10^{-2}$. Note that the error of the obtained approximation μ_k is reported as

$$\text{relative error} := \frac{|\mathsf{Re}(\mu_k) - \alpha_\varepsilon(A)|}{\|A\| \cdot |\alpha_\varepsilon(A)|}.$$

The reference value for $\alpha_\varepsilon(A)$ is computed by the criss-cross algorithm [7]. All problems, except **`randomtri`**, were solved up to a relative error below 10^{-12} which strongly indicates convergence to the global optimum. For **`randomtri`**, Algorithm 2 nearly stagnates but Algorithm 4 succeeds. From the table, we can clearly see that Algorithm 4 is always faster than Algorithm 2—most of the time by at least a factor of two—while still being as reliable and accurate. Usually, three eigenvalue computations are sufficient before resorting to SVDs only.

Table 5.2 provides a comprehensive comparison between Algorithm 4, the criss-cross algorithm[2] from [8], and the GO algorithm[3] from [17]. It can be seen that Algorithm 4 outperforms the two other methods for all problems except the Grcar matrix, where the criss-cross algorithm is slightly faster.

To investigate the dependence on the size of the problems, we enlarge the test set from Table 5.2 by adding matrices of size 200 and 300 for all scalable examples. Figure 5.2 displays the resulting logarithmic performance profile [11]. That is, for each solver, we plot the probability $P(\alpha)$ that a method is faster within a factor α of the best time over all methods using a $\log_{10}$ scale for α. When a method does not successfully solve a problem, its time is taken to be infinite. Algorithm 4 turns out to be the fastest for almost 90% of the test problems while being only slightly less reliable (95%) than the criss-cross algorithm (100%). A subset of these problems are reported in Table 5.3. Not surprisingly, with increasing matrix sizes, the relative performance of the criss-cross algorithm becomes worse compared to the GO algorithm and Algorithm 4.

5.4. Numerical experiments with large-scale problems. The subspace methods proposed in this paper can be applied to sparse large matrices. For this

[2]Code PSPA (March 10, 2005) from `http://www.cs.nyu.edu/mengi/robuststability.html`.
[3]Code PSAPSR (1.01) from `http://www.cs.nyu.edu/overton/software/psapsr/index.html`.

TABLE 5.2
Computational results on small-scale problems for $\varepsilon = 10^{-2}$. All times in seconds.

	Criss-cross [8]	GO [17]		Algorithm 4	
Call	Time	Rel. error	Time	Rel. error	Time
`airy(100)`	1.670	9×10^{-16}	0.785	7×10^{-17}	**0.110**
`basor(100)`	1.058	3×10^{-16}	0.415	3×10^{-16}	**0.108**
`boeing('S')`	0.395	1×10^{-19}	1.296	3×10^{-20}	**0.319**
`boeing('O')`	0.345	6×10^{-17}	3.007	2×10^{-19}	**0.234**
`chebspec(100)`	0.436	1×10^{-15}	4.596	2×10^{-15}	**0.247**
`companion(100)`	1.283	6×10^{-17}	5.653	2×10^{-17}	**0.204**
`convdiff(100)`	0.343	6×10^{-18}	0.158	9×10^{-18}	**0.149**
`davies(100)`	0.463	4×10^{-21}	0.179	4×10^{-20}	**0.048**
`demmel(100)`	1.283	6×10^{-17}	5.653	2×10^{-17}	**0.204**
`frank(100)`	0.233	2×10^{-18}	0.194	5×10^{-17}	**0.149**
`gaussseidel(100,'C')`	0.523	1×10^{-12}	1.823	9×10^{-15}	**0.049**
`gaussseidel(100,'D')`	0.277	3×10^{-12}	9.235	3×10^{-15}	**0.226**
`gaussseidel(100,'U')`	0.553	3×10^{-11}	64.744	5×10^{-15}	**0.253**
`godunov(100)`	**0.001**	1×10^{-15}	0.037	2×10^{-15}	0.029
`grcar(100)`	**0.258**	7×10^{-12}	36.049	3×10^{-15}	0.279
`hatano(100)`	0.393	2×10^{-13}	2.270	5×10^{-16}	**0.075**
`kahan(100)`	0.205	6×10^{-16}	0.320	2×10^{-16}	**0.103**
`landau(100)`	0.648	3×10^{-15}	0.250	3×10^{-13}	**0.046**
`orrsommerfeld(100)`	1.323	2×10^{-15}	3.516	1×10^{-15}	**0.220**
`random(100)`	0.536	4×10^{-15}	0.263	5×10^{-15}	**0.128**
`randomtri(100)`	0.299	2×10^{-12}	4.941	2×10^{-15}	**0.230**
`riffle(100)`	0.142	3×10^{-13}	0.601	1×10^{-15}	**0.099**
`transient(100)`	0.432	6×10^{-13}	0.815	2×10^{-15}	**0.213**
`twisted(100)`	0.891	7×10^{-16}	0.427	8×10^{-17}	**0.111**

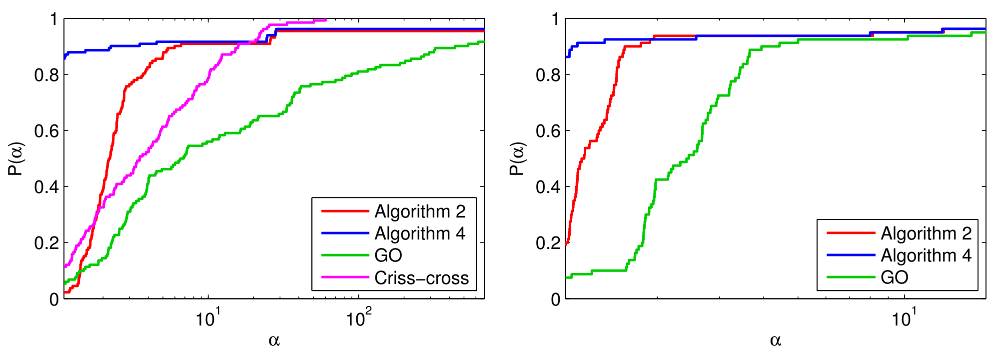

FIG. 5.2. *Performance profile for an enlarged test set of small-scale problems (left figure) and sparse, large-scale problems (right figure), both for $\varepsilon = 10^{-2}$ and $\varepsilon = 10^{-4}$. Note the logarithmic scale for the factor α.*

purpose, all eigenvalue and singular value computations need to be performed by an iterative method. For computing the right-most eigenvalue of a matrix, we use the software package ARPACK [27] available through `eigs` in MATLAB. For computing the smallest singular value $\sigma_{\min}(A - \mu I)$, we apply PROPACK[4] from [26] to $(A - \mu I)^{-1}$. The inverse is applied to a vector via a sparse LU factorization of $A - \mu I$ using `lu` in MATLAB.

Our set of large-scale test problems consists of the matrices from EigTool used

[4]Code PROPACK (1.1) from `http://soi.stanford.edu/~rmunk/PROPACK/`.

TABLE 5.3
Computational results on a subset of small-scale problems for $\varepsilon = 10^{-4}$. All times in seconds.

	Criss-cross [8]		GO [17]		Algorithm 4	
Call	Time	Iters	Time	Iters	Time	Iters
`airy(100)`	0.93	3	0.24	3	**0.10**	4
`airy(200)`	6.29	4	0.94	3	**0.60**	5
`airy(300)`	8.45	2	2.46	3	**2.23**	7
`grcar(100)`	**0.24**	2	44.5	522	0.37	13
`grcar(200)`	1.94	3	225	651	**1.13**	11
`grcar(300)`	4.11	2	861	981	**2.73**	11
`landau(100)`	0.41	3	0.13	2	**0.05**	3
`landau(200)`	1.41	2	0.65	2	**0.20**	3
`landau(300)`	9.16	4	1.53	2	**0.50**	3

in [17, Sec. 9] and all the sparse matrices of the benchmark set $COMPl_eib$ [28]. The obtained results are shown in Table 5.4. For all examples but `skewlap3d` and `sparserandom`, our subspace methods are faster compared to the GO algorithm. For some examples, this is due to less iterations. However, there are also examples with essentially the same number of iterations, for which Algorithms 2 and 4 benefit from not having to compute left eigenvectors as in the GO algorithm. For the matrices `skewlap3d` and `sparserandom` the sparse LU factorization of $A - \mu I$ is too costly and should be replaced by the use of an iterative method. Note that the failures for `tolosa` and `convdiff_fd`, indicated by — in the table, are due to failures of `eigs` to compute right-most eigenvalues.

A performance profile can be found in Figure 5.2. To assess the accuracy of the methods, checking whether the obtained points correspond to the globally right-most one is unfortunately not feasible in a large-scale setting. Instead, we started Algorithms 2 and 4 not only from the right-most eigenvalue, but also from the second, third, …, 10th most right eigenvalue. For all three methods, we observed for all examples a relative difference between 10^{-7} and 10^{-15}, strongly indicating convergence to the globally right-most point.

6. Extension to stability radius computation. The extension of the subspace methods to the computation of the stability radius $\beta(A)$ of a stable matrix A is relatively straightforward and will be outlined in this section. Recall that $\beta(A) = \min\{\sigma_{\min}(A - zI) \colon z \in \mathrm{i}\mathbb{R}\}$. An analogous quantity for a rectangular matrix pencil $\hat{A} - z\hat{B}$ can be defined as

$$\beta(\hat{A}, \hat{B}) = \min\{\sigma_{\min}(\hat{A} - z\hat{B}) \colon z \in \mathrm{i}\mathbb{R}\}.$$

From Lemma 3.1, it follows that

$$\beta(A) \le \beta(AV, V) \le \beta(AU, U) \tag{6.1}$$

for orthonormal bases U, V of subspaces $\mathcal{U}, \mathcal{V}$, respectively, such that $\mathcal{U} \subset \mathcal{V} \subset \mathbb{C}^n$.

The following lemma extends Theorem 3.3 to the stability radius.

LEMMA 6.1. *Let $\mu_\beta \in \mathrm{i}\mathbb{R}$ such that $\beta(A) = \sigma_{\min}(A - \mu_\beta I)$ and let v_β be an associated right singular vector. Then for a subspace $\mathcal{V}$ with an orthonormal basis V it holds that*

$$\beta(AV, V) - \|A - \mu_\beta I\| \cdot d(v_\beta, \mathcal{V}) \le \beta(A) \le \beta(AV, V).$$

Table 5.4
Computational results on large-scale problems from EigTool and a representative subset of $COMPl_eib$ for $\varepsilon = 10^{-2}$. All times in seconds.

		GO [17]		Algorithm 2		Algorithm 4	
Problem	n	Iters	Time	Iters	Time	Iters	Time
`dwave(2048)`	2048	7	0.597	3	0.207	3	**0.174**
`markov(100)`	5050	65	16.71	5	1.651	5	**1.080**
`convdiff_fd(10)`	400	—	(1.187)	—	(1.984)	11	**2.240**
`olmstead(500)`	500	3	1.038	3	0.387	3	**0.325**
`pde(2961)`	2961	11	1.711	5	0.819	5	**0.659**
`rdbrusselator(3200)`	3200	4	1.328	3	**0.760**	3	0.792
`sparserandom(1000)`	1000	5	**0.143**	4	1.164	4	1.144
`skewlap3d(30)`	24389	5	**27.52**	5	53.933	13	70.90
`supg(20)`	400	90	2.692	5	**0.263**	6	0.282
`tolosa(4000)`	4000	—	(3.118)	—	(5.736)	—	(5.034)
`COMPleib('HF2D9')`	3481	2	0.422	2	0.273	2	**0.269**
`COMPleib('HF2D_IS2_M529')`	529	3	0.316	3	0.133	3	**0.087**
`COMPleib('HF2D_IS4_M484')`	484	2	0.087	2	0.054	2	**0.047**
`COMPleib('HF2D_CD3_M576')`	576	3	0.275	4	0.164	4	**0.140**
`COMPleib('HF2D_IS4')`	3600	2	0.588	2	0.341	2	**0.273**
`COMPleib('HF2D_CD3')`	4096	3	1.978	4	1.077	4	**0.903**

Suppose that Assumptions 1 and 2 hold for $\varepsilon = \beta(A)$. Then any $\mu_V \in \mathrm{i}\mathbb{R}$ such that $\beta(AV, V) = \sigma_{\min}(AV - \mu_V V)$ satisfies

$$|\mu_\beta - \mu_V| = O(d(v_\beta, \mathcal{V})).$$

Proof. Let $v \in \mathcal{V}$ be with $\|v - v_\beta\| = d(v_\beta, \mathcal{V})$. Then

$$\begin{aligned}\beta(AV, V) &\leq \|Av - \mu_\beta v\| \leq \|Av_\beta - \mu_\beta v_\beta\| + \|(A - \mu_\beta I)(v - v_\beta)\| \\ &\leq \beta(A) + \|A - \mu_\beta I\| \cdot d(v_\beta, \mathcal{V}),\end{aligned}$$

which shows the first part of the theorem.

For the proof of the second part, we first note that, by continuity, Assumptions 1 and 2 remain true if we replace $\varepsilon = \beta(A)$ by $\tilde{\varepsilon} := \beta(AV, V)$ for sufficiently small $d(v_\beta, \mathcal{V})$. Let $\tilde{\mu}_\beta$ denote the right-most point of $\Lambda_{\tilde{\varepsilon}}(A)$ with associated right singular vector $\tilde{v}_\beta$. By the uniqueness and Lipschitz continuity of the right-most point with respect to ε [29, 18], there is a constant L such that

$$|\tilde{\mu}_\beta - \mu_\beta| \leq L\,|\tilde{\varepsilon} - \varepsilon| \leq L\|A - \mu_\beta I\| \cdot d(v_\beta, \mathcal{V}) = O(d(v_\beta, \mathcal{V})). \tag{6.2}$$

Applying the result (3.7) of Theorem 3.3 to $\Lambda_{\tilde{\varepsilon}}(A)$ and $\Lambda_{\tilde{\varepsilon}}(AV, V)$ yields

$$|\tilde{\mu}_\beta - \mu_V| = O(d(\tilde{v}_\beta, \mathcal{V})).$$

Consider now $v_* \in \mathcal{V}$ such that $d(\tilde{v}_\beta, \mathcal{V}) = \|v_* - \tilde{v}_\beta\|$, then

$$d(\tilde{v}_\beta, \mathcal{V}) \leq \|v_* - v_\beta\| + \|v_\beta - \tilde{v}_\beta\| \leq O(d(v_\beta, \mathcal{V})) + O(|\tilde{\mu}_\beta - \mu_\beta|) \leq O(d(v_\beta, \mathcal{V})),$$

and combined with (6.2), this completes the proof of the second part by the triangular inequality. ∎

TABLE 6.1
Convergence of Algorithm 5 applied to the 100×100 *matrix* `airy(100)`.

k	$\beta(AV_k, V_k)$	Abs. error	Rel. error
1	$7.826030706964750 \times 10^{-2}$	3.01×10^{-02}	4.31×10^{-04}
2	$\mathit{4}.874014867731968 \times 10^{-2}$	5.92×10^{-04}	4.31×10^{-04}
3	$\mathit{4.815}056990918770 \times 10^{-2}$	2.24×10^{-06}	3.21×10^{-08}
4	$\mathit{4.814833244}814336 \times 10^{-2}$	2.67×10^{-12}	3.83×10^{-14}

Algorithm 5 Stability radius computation

Input: Matrix $A \in \mathbb{C}^{n \times n}$.
Output: Approximation to the stability radius $\beta(A)$.
  Compute right-most eigenvalue λ_0 and right eigenvector $\hat{v}_0$ of A.
  Set $V_1 = [\hat{v}_0]$.
  for $k = 1, 2, \ldots$ until converged **do**
    Compute minimizer μ_k of $\min\{\sigma_{\min}(AV_k - zV_k) \colon z \in \mathrm{i}\mathbb{R}\}$.
    Compute right singular vector v_k belonging to $\sigma_{\min}(A - \mu_k I)$.
    Compute $V_{k+1} = \mathrm{orth}([V_k \ \hat{v}_k])$.
  end for
  return $\sigma_{\min}(A - \mu_k I)$

Algorithm 5 is a suitable adaption of Algorithm 3 to stability radius computation. In each step, we have to compute the minimizer μ_k corresponding to $\beta(AV_k, V_k)$. For simplicity this is done by a variant of Byers' bisection method [10], but the quadratic algorithm from [4] could be extended to the rectangular case similarly to the variants described in Section 5.2.

REMARK 6.2. *When the right-most point μ_β of $\Lambda_{\beta(A)}(A)$ is real (which happens quite frequently for real matrices), Algorithm 5 usually converges in one iteration. For $\lambda_0 \in \mathbb{R}$, it can be easily seen that $\mu_1 = 0 = \mu_\beta$. This implies $v_\beta \in \mathsf{span}(V_k)$ for all $k \geq 2$. Combined with* (6.1)*, this shows $\beta(AV_k, V_k) = \beta(A)$ for all $k \geq 2$. In contrast, when μ_β is not real, v_β will in general not be in $\mathsf{span}(V_1)$ and Algorithm 5 has not yet converged.*

We refrain from giving a detailed presentation of numerical experiments; the observed results are quite similar to the ones for the ε-pseudospectral abscissa, see Section 5. An example illustrating local quadratic convergence is given in Table 6.1.

6.1. Further extensions. Based on the algorithms from [17, 31], it is relatively straightforward to develop a subspace method for the computation of the pseudospectral radius. In contrast, it seems to be difficult to turn the method from [15] for computing the H_∞ norm into a subspace method. The major problem is to find a suitable replacement for the rectangular matrix pencils that played a major role in our developments. Similarly, the lack of efficient algorithms for computing right-most points of structured rectangular pseudospectra currently prevents the extension of our subspace methods to real or otherwise structured pseudospectra, see [9, 16] for some recent developments.

7. Conclusions. In this paper, we have proposed novel subspace methods for computing the ε-pseudospectral abscissa of a matrix. Supported by the numerical experiments, we recommend the use of these methods for matrices from size 100. A MATLAB implementation is available from `http://anchp.epfl.ch`.

Acknowledgments. The authors thank Nicola Guglielmi, Mert Gürbüzbalaban, Christian Lubich, and Michael Overton for discussions on the topic of this paper. Parts of these discussions took place during visits at FIM (Institute for Mathematical Research) at ETH Zurich and Mathematisches Forschungsinstitut Oberwolfach. The authors are grateful for the very helpful and insightful remarks by the referees, in particular for pointing out a flaw in the convergence proof in a previous version of the paper.

Appendix A. Pseudospectra of $n \times 1$ pencils.

The following lemma fully characterizes the rectangular pseudospectrum of $n \times 1$ matrix pencils.

LEMMA A.1. *Let $v, w \in \mathbb{C}^n$ with $\|v\| = 1$. If $\|w\|^2 - |v^*w|^2 > \varepsilon^2$, the ε-pseudospectrum of $w - zv$ is empty. Otherwise, the ε-pseudospectrum is given by the disc with center v^*w and radius*

$$r = \sqrt{\varepsilon^2 - \|(I - vv^*)w\|^2} = \sqrt{\varepsilon^2 - \|w\|^2 + |v^*w|^2}.$$

In the latter case, $\alpha_\varepsilon(w, v) = \mathsf{Re}(v^*w) + r$.

Proof. Consider the orthogonal decomposition $w = v(v^*w) + (I - vv^*)w$. Then for $\lambda \in \mathbb{C}$

$$\|w - \lambda v\|^2 = |v^*w - \lambda|^2 + \|(I - vv^*)w\|^2 = |v^*w - \lambda|^2 + \|w\|^2 - |v^*w|^2.$$

In particular, this implies that $\|w - \lambda v\|^2 \leq \varepsilon^2$ cannot be satisfied if the condition of the lemma is not met. Otherwise, it is easily seen that the set

$$\{\lambda \in \mathbb{C} \colon |v^*w - \lambda|^2 + \|w\|^2 - |v^*w|^2 \leq \varepsilon^2\}$$

describes a disc with center v^*w and radius r. The formula for $\alpha_\varepsilon(w, v)$ now follows immediately. □

REFERENCES

[1] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst. *Templates for the solution of Algebraic Eigenvalue Problems: A Practical Guide.* SIAM, 2000.

[2] P. Benner, V. Mehrmann, V. Sima, S. Van Huffel, and A. Varga. SLICOT—a subroutine library in systems and control theory. In *Applied and computational control, signals, and circuits, Vol. 1*, pages 499–539. Birkhäuser Boston, Boston, MA, 1999. See also `http://www.slicot.de`.

[3] G. Boutry, M. Elad, G. H. Golub, and P. Milanfar. The generalized eigenvalue problem for nonsquare pencils using a minimal perturbation approach. *SIAM J. Matrix Anal. Appl.*, 27(2):582–601, 2005.

[4] S. Boyd and V. Balakrishnan. A regularity result for the singular values of a transfer matrix and a quadratically convergent algorithm for computing its $\mathbf{L}_\infty$-norm. *Systems Control Lett.*, 15(1):1–7, 1990.

[5] S. Boyd, V. Balakrishnan, and P. Kabamba. A bisection method for computing the $\mathcal{H}_\infty$ norm of a transfer matrix and related problems. *Math. Control, Signals, Sys.*, 2:207–219, 1989.

[6] N. A. Bruinsma and M. Steinbuch. A fast algorithm to compute the H_∞-norm of a transfer function matrix. *Sys. Control Lett.*, 14(4):287–293, 1990.

[7] J. V. Burke, A. S. Lewis, E. Mengi, and M. L. Overton. `pspa.m`, 2005. See `http://www.cs.nyu.edu/mengi/robuststability.html`.

[8] J. V. Burke, A. S. Lewis, and M. L. Overton. Robust stability and a criss-cross algorithm for pseudospectra. *IMA J. Numer. Anal.*, 23(3):359–375, 2003.

[9] P. Buttà, N. Guglielmi, and S. Noschese. Computing the structured pseudospectrum of a Toeplitz matrix and its extreme points. *SIAM J. Matrix Anal. Appl.*, 33(4):1300–1319, 2012.
[10] R. Byers. A bisection method for measuring the distance of a stable matrix to the unstable matrices. *SIAM J. Sci. Statist. Comput.*, 9:875–881, 1988.
[11] E. D. Dolan and J. Moré. Benchmarking optimization software with performance profiles. *Math. Progr.*, 91(2):201–213, 2002.
[12] M. A. Freitag and A. Spence. A Newton-based method for the calculation of the distance to instability. *Linear Algebra Appl.*, 435(12):3189–3205, 2011.
[13] E. Gekeler. On the solution of systems of equations by the epsilon algorithm of Wynn. *Math. Comp.*, 26:427–436, 1972.
[14] J. F. Grcar. Operator coefficient methods for linear equations. Technical Report SAND89-8691, Sandia National Laboratories, 1989.
[15] N. Guglielmi, M. Gürbüzbalaban, and M. L. Overton. Fast Approximation of the H_∞ Norm via Optimization over Spectral Value Sets. *SIAM J. Matrix Anal. Appl.*, 34(2):709–737, 2013.
[16] N. Guglielmi and C. Lubich. Low-rank dynamics for computing extremal points of real pseudospectra. *SIAM J. Matrix Anal. Appl.*, 34(1):40–66, 2013.
[17] N. Guglielmi and M. L. Overton. Fast algorithms for the approximation of the pseudospectral abscissa and pseudospectral radius of a matrix. *SIAM J. Matrix Anal. Appl.*, 32(4):1166–1192, 2011. See also `http://www.cs.nyu.edu/overton/software/psapsr/index.html`.
[18] M. Gürbüzbalaban and M. L. Overton. Some regularity results for the pseudospectral abscissa and pseudospectral radius of a matrix. *SIAM J. Optim.*, 22(2):281–285, 2012.
[19] C. He and G. A. Watson. An algorithm for computing the distance to instability. *SIAM J. Matrix Anal. Appl.*, 20(1):101–116, 1999.
[20] D. Hinrichsen, B. Kelb, and A. Linnemann. An algorithm for the computation of the structured complex stability radius. *Automatica J. IFAC*, 25(5):771–775, 1989.
[21] D. Hinrichsen and M. Motscha. Optimization problems in the robustness analysis of linear state space systems. In *Approximation and optimization (Havana, 1987)*, volume 1354 of *Lecture Notes in Math.*, pages 54–78. Springer, Berlin, 1988.
[22] D. Hinrichsen and A. J. Pritchard. *Mathematical Systems Theory I*, volume 48 of *Texts in Applied Mathematics*. Springer-Verlag, 2005.
[23] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, 1990.
[24] T. Kato. *Perturbation Theory for Linear Operators*. Springer, second edition, 1995.
[25] D. Kressner. Finding the distance to instability of a large sparse matrix. In *IEEE International Symposium on Computer-Aided Control Systems Design, Munich*, 2006.
[26] R. M. Larsen. PROPACK—software for large and sparse SVD calculations. `http://soi.stanford.edu/~rmunk/PROPACK`, 2004.
[27] R. B. Lehoucq, D. C. Sorensen, and C. Yang. *ARPACK users' guide*. SIAM, Philadelphia, PA, 1998.
[28] F. Leibfritz. $COMPl_eib$: *CO*nstraint *M*atrix-optimization *P*roblem *lib*rary - a collection of test examples for nonlinear semidefinite programs, control system design and related problems. Technical report, University of Trier, Department of Mathematics, 2004.
[29] A. S. Lewis and C. H. J. Pang. Variational analysis of pseudospectra. *SIAM J. Optim.*, 19(3):1048–1072, 2008.
[30] The MathWorks, Inc., Natick, Mass, 01760. *The* MATLAB *Control Toolbox, Version 9.2*, 2011.
[31] E. Mengi and M. L. Overton. Algorithms for the computation of the pseudospectral radius and the numerical radius of a matrix. *IMA J. Numer. Anal.*, 25(4):648–669, 2005.
[32] C. D. Meyer and G. W. Stewart. Derivatives and perturbations of eigenvectors. *SIAM J. Numer. Anal.*, 25(3):679–691, 1988.
[33] F. A. Potra. On Q-order and R-order of convergence. *Journal of Optimization Theory and Applications*, 63(3):415–431, 1989.
[34] D. A. Smith, W. F. Ford, and A. Sidi. Extrapolation methods for vector sequences. *SIAM Rev.*, 29(199–233), 1987.
[35] G. W. Stewart. *Matrix Algorithms, Volume II: Eigensystems*. SIAM, 2001.
[36] L. N. Trefethen and M. Embree. *Spectra and Pseudospectra. The Behavior of Nonnormal Matrices and Operators*. Princeton University Press, Princeton, NJ, 2005.
[37] P.-Å. Wedin. Perturbation bounds in connection with singular value decomposition. *BIT*, 12:99–111, 1972.
[38] P.-Å. Wedin. Perturbation theory for pseudo-inverses. *BIT*, 13:217–232, 1973.
[39] T. G. Wright. EigTool, 2002. See `http://www.comlab.ox.ac.uk/pseudospectra/eigtool/`.

[40] T. G. Wright and L. N. Trefethen. Pseudospectra of rectangular matrices. *IMA J. Numer. Anal.*, 22(4):501–519, 2002.

Recent publications :

MATHEMATICS INSTITUTE OF COMPUTATIONAL SCIENCE AND ENGINEERING
Section of Mathematics
Ecole Polytechnique Fédérale
CH-1015 Lausanne

01.2012 A. ABDULLE, A. NONNENMACHER:
A posteriori error estimate in quantities of interest for the finite element heterogeneous multiscale method

02.2012 F. NOBILE, M. POZZOLI, C. VERGARA:
Time accurate partitioned algorithms for the solution of fluid-structure interaction problems in haemodynamics

03.2012 T. LASSILA, A. MANZONI, A. QUARTERONI, G. ROZZA:
Boundary control and shape optimization for the robust design of bypass anastomoses under uncertainty

04.2012 D. KRESSNER, C. TOBLER:
htucker – A Matlab toolbox for tensors in hierarchical Tucker format

05.2012 A. ABDULLE, G. VILLMART, KONSTANTINOS C. ZYGALAKIS:
Second weak order explicit stabilized methods for stiff stochastic differential equations.

06.2012 A. CABOUSSAT, S. BOYAVAL, A. MASSEREY:
Three-dimensional simulation of dam break flows.

07.2012 J BONNEMAIN, S. DEPARIS, A. QUARTERONI:
Connecting ventricular assist devices to the aorta: a numerical model.

08.2012 J BONNEMAIN, ELENA FAGGIANO, A. QUARTERONI, S. DEPARIS:
A framework for the analysis of the haemodynamics in patient with ventricular assist device.

09.2012 T. LASSILA, A. MANZONI, G. ROZZA:
Reduction strategies for shape dependent inverse problems in haemodynamics.

10.2012 C. MALOSSI, P. BLANCO, P. CROSETTO, S. DEPARIS, A. QUARTERONI:
Implicit coupling of one-dimensional and three-dimensional blood flow models with compliant vessels.

11.2012 S. FLOTRON J. RAPPAZ:
Conservation schemes for convection-diffusion equations with Robin's boundary conditions.

12.2012 A. UMSCHMAJEW, B. VANDEREYCKEN:
The geometry of algorithms using hierarchical tensors.

13.2012 (NEW) D. KRESSNER, B. VANDEREYCKEN:
Subspace methods for computing the pseudospectral abscissa and the stability radius.