SIAM J. MATRIX ANAL. APPL.
Vol. 32, No. 4, pp. 1166–1192

FAST ALGORITHMS FOR THE APPROXIMATION OF THE PSEUDOSPECTRAL ABSCISSA AND PSEUDOSPECTRAL RADIUS OF A MATRIX*

NICOLA GUGLIELMI† AND MICHAEL L. OVERTON‡

Abstract. The ε-pseudospectral abscissa and radius of an $n \times n$ matrix are, respectively, the maximal real part and the maximal modulus of points in its ε-pseudospectrum, defined using the spectral norm. Existing techniques compute these quantities accurately, but the cost is multiple singular value decompositions and eigenvalue decompositions of order n, making them impractical when n is large. We present new algorithms based on computing only the spectral abscissa or radius of a sequence of matrices, generating a sequence of lower bounds for the pseudospectral abscissa or radius. We characterize fixed points of the iterations, and we discuss conditions under which the sequence of lower bounds converges to local maximizers of the real part or modulus over the pseudospectrum, proving a locally linear rate of convergence for ε sufficiently small. The convergence results depend on a perturbation theorem for the normalized eigenprojection of a matrix as well as a characterization of the group inverse (reduced resolvent) of a singular matrix defined by a rank-one perturbation. The total cost of the algorithms is typically only a constant times the cost of computing the spectral abscissa or radius, where the value of this constant usually increases with ε, and may be less than 10 in many practical cases of interest.

Key words. pseudospectrum, eigenvalue, spectral abscissa, spectral radius, stability radius, robustness of linear systems, group inverse, reduced resolvent, sparse matrix

AMS subject classifications. 15A18, 65K05

DOI. 10.1137/100817048

1. Introduction.

The linear dynamical system $\dot{x} = Ax$ is asymptotically stable, that is, solutions $x(t)$ converge to zero as $t \to \infty$ for all initial states, if and only if all eigenvalues of A lie strictly in the left half of the complex plane. Equivalently, the spectral abscissa of A (the maximum of the real parts of the eigenvalues) is negative. Likewise, the discrete-time system $x_{k+1} = Ax_k$ is asymptotically stable when all eigenvalues of A lie inside the unit circle, so that the spectral radius of A is less than one. However, as is well known, the spectral abscissa and spectral radius are not robust measures of stability, as small perturbations to the matrix can result in large changes to the spectrum. Furthermore, the quantities $\|\exp(tA)\|$ and $\|A^k\|$ can be arbitrarily large even if the system is asymptotically stable. It has long been recognized that pseudospectra provide a more robust measure of system behavior [TE05].

Let $\Lambda(A)$ denote the spectrum (set of eigenvalues) of an $n \times n$ real or complex matrix A. The ε-pseudospectrum of A is

$$\Lambda_\varepsilon(A) = \left\{ z \in \mathbb{C} \ : z \in \Lambda(B) \text{ for some } B \in \mathbb{C}^{n\times n} \text{ with } \|A - B\| \le \varepsilon \right\}. \tag{1.1}$$

Although the pseudospectrum can be defined for any matrix norm, we are concerned

*Received by the editors December 6, 2010; accepted for publication (in revised form) by F. Tisseur June 24, 2011; published electronically November 1, 2011.
http://www.siam.org/journals/simax/32-4/81704.html

†Dipartimento di Matematica Pura e Applicata, Università dell'Aquila, I-67010 L'Aquila, Italy (guglielm@univaq.it). This author's work was supported in part by the Italian Ministry of Education, Universities and Research (M.I.U.R.).

‡Courant Institute of Mathematical Sciences, New York University, New York, NY 10012 (overton@cs.nyu.edu). This author's work was supported in part by National Science Foundation grant DMS-1016325 and in part by the Institute for Mathematical Research (F.I.M.), ETH Zurich.

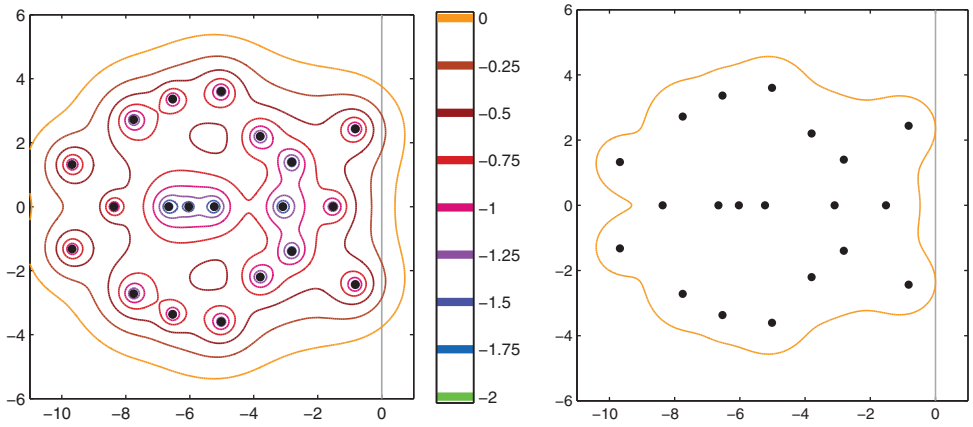

FIG. 1.1. *On the left, EigTool displays the boundaries of the ε-pseudospectra of a randomly generated matrix A_0 for $\varepsilon = 10^j$, $j = -2, -1.75, \ldots, 0$; the solid dots are the eigenvalues. The right panel shows the boundary of $\Lambda_\varepsilon(A_0)$ for the critical ε (the distance to instability) for which $\Lambda_\varepsilon(A_0)$ is tangent to, but does not cross, the imaginary axis.*

exclusively with the 2-norm (spectral norm) in this paper. Then, the following fundamental result, which is implicit in [TE05, Wil86], characterizes $\Lambda_\varepsilon(A)$ in terms of singular values. Let I be the $n \times n$ identity matrix. The singular value decomposition (SVD) of a matrix C is $C = U\Sigma V^*$, with $U^*U = I$, $V^*V = I$ and real $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$ with $\sigma_1 \geq \cdots \geq \sigma_n$. We use the notation $\sigma_k(C)$ for the kth largest singular value of C.

LEMMA 1.1. *Let $z \in \mathbb{C}$ be given, and let $\varepsilon = \sigma_n(A - zI)$. Let u and v be unit vectors, that is, $\|u\| = \|v\| = 1$. Then the following are equivalent:*

$$(A - zI)v = -\varepsilon u, \quad u^*(A - zI) = -\varepsilon v^* \tag{1.2}$$

and

$$(A + \varepsilon uv^* - zI)v = 0, \quad u^*(A + \varepsilon uv^* - zI) = 0. \tag{1.3}$$

The first condition states that v and $-u$ are right and left singular vectors of $A - zI$ corresponding to its smallest singular value, and the second condition says that v and u are, respectively, right and left eigenvectors of $B = A + \varepsilon uv^*$ corresponding to the eigenvalue z. Hence, the set $\Lambda_\varepsilon(A)$ is given by

$$\Lambda_\varepsilon(A) = \{z \in \mathbb{C} \ : \sigma_n(A - zI) \leq \varepsilon\} \tag{1.4}$$

with boundary

$$\partial\Lambda_\varepsilon(A) = \{z \in \mathbb{C} \ : \sigma_n(A - zI) = \varepsilon\}.$$

The proof is straightforward.

The left panel of Figure 1.1 shows, for $\varepsilon = 10^j$, $j = -2, -1.75, \ldots, 0$, the boundaries of the ε-pseudospectra of a randomly generated matrix A_0 with $n = 20$ (the matrix was shifted left so that its eigenvalues, marked as solid dots, are in the left half-plane). Note that $\Lambda_\varepsilon(A_0)$ is in the left half-plane for $\varepsilon \leq 10^{-0.5}$ but not for $\varepsilon \geq 10^{-0.25}$. The figure was produced by EigTool [Wri02, WT01], which uses the

characterization (1.4) to compute the pseudospectral boundaries. For each ε, the pseudospectrum is a compact set with up to n connected components, each containing at least one eigenvalue and not necessarily simply connected. For example, for $\varepsilon = 10^{-0.5}$, the pseudospectrum $\Lambda_\varepsilon(A_0)$ has three connected components, the largest of which is not simply connected.

The *ε-pseudospectral abscissa* of A is the largest of the real parts of the elements of the pseudospectrum,

$$\alpha_\varepsilon(A) = \max\{\text{Re } z : z \in \Lambda_\varepsilon(A)\}.$$

Equivalently, from Lemma 1.1,

$$\alpha_\varepsilon(A) = \max\{\text{Re } z : \sigma_n(A - zI) \le \varepsilon\}. \tag{1.5}$$

The case $\varepsilon = 0$ reduces to the spectral abscissa, which we denote by $\alpha(A)$. For the system $\dot{x} = Ax$, the pseudospectral abscissa α_ε ranges, as ε is varied from 0 to ∞, from measuring asymptotic growth/decay to initial growth/decay, and also measures robust stability with respect to perturbations bounded in norm by ε.

The pseudospectral abscissa $\alpha_\varepsilon(A)$ is negative exactly when all matrices within a distance ε of A have negative spectral abscissa, that is, if the *distance to instability* for A [VL85] is larger than ε. The right panel of Figure 1.1 shows the boundary of $\Lambda_\varepsilon(A_0)$ for ε equal to the distance to instability for A_0, approximately $10^{-0.32}$. This quantity is also known as the *complex stability radius* (radius in the sense of the norm of allowable perturbations to A) [HP86].

Analogous quantities for discrete-time systems are defined in a similar fashion. The *ε-pseudospectral radius* of A is

$$\rho_\varepsilon(A) = \max\{|z| : z \in \Lambda_\varepsilon(A)\}, \tag{1.6}$$

reducing to the spectral radius $\rho(A)$ when $\varepsilon = 0$. This quantity is less than one when all matrices within a distance ε of A have spectral radius less than one, that is, if the discrete-time distance to instability is larger than ε. For simplicity, we shall focus in this paper on the pseudospectral abscissa, but all the ideas extend to the pseudospectral radius, as explained in section 7.

The extent to which pseudospectral measures can be used to estimate transient system behavior is a topic of current research. On the one hand, it is well known that using the Kreiss matrix theorem [Kre62, TE05], it is possible to obtain bounds on the quantities $\|\exp(tA)\|$ and $\|A^k\|$ in terms of $\alpha_\varepsilon(A)$ and $\rho_\varepsilon(A)$, respectively, for example,

$$\sup_{\varepsilon>0} \frac{\rho_\varepsilon(A) - 1}{\varepsilon} \le \sup_{\ell \ge 0} \|A^\ell\| \le en \sup_{\varepsilon>0} \frac{\rho_\varepsilon(A) - 1}{\varepsilon}.$$

On the other hand, recent work of Ransford [Ran07] discusses limitations of pseudospectral measures in estimating such quantities.

The pseudospectral abscissa can be computed to arbitrary accuracy using the "criss-cross" algorithm of Burke, Lewis, and Overton [BLO03]. This uses a sequence of vertical and horizontal searches in the complex plane to identify the intersection of a given line with $\partial\Lambda_\varepsilon(A)$. Horizontal searches give estimates of $\alpha_\varepsilon(A)$, while vertical searches identify favorable locations for the horizontal searches. A related method for the pseudospectral radius was given by Mengi and Overton [MO05], using radial

and circular searches instead of horizontal and vertical searches. These algorithms are implemented in EigTool.

The criss-cross algorithm is based on earlier work of Byers for computing the distance to instability [Bye88]. Each vertical search requires the computation of all eigenvalues of a $2n \times 2n$ Hamiltonian matrix

$$H(r) = \begin{pmatrix} rI - A^* & -\varepsilon I \\ \varepsilon I & A - rI \end{pmatrix}, \tag{1.7}$$

where the real number r defines the vertical line $\{\text{Re } z = r\}$. This is because each imaginary eigenvalue $\mathbf{i}s$ of $H(r)$ identifies a complex number $z = r + \mathbf{i}s$ for which $A - zI$ has a singular value equal to ε. To see this, denote the eigenvector by $[y^T \; x^T]^T$ and compare the eigenvector equation for $H(r)$ to (1.2). Because such singular values are not necessarily the smallest, each vertical step also requires an SVD for each such imaginary eigenvalue to determine whether or not $z \in \partial\Lambda_\varepsilon(A)$. Thus each iteration effectively costs $O(n^3)$ operations. The algorithm converges quadratically, so it is fast when n is small. However, it is impractical when n is large. The purpose of this paper is to present new algorithms for approximating the pseudospectral abscissa and radius that are practical when the matrix A is large and sparse.

The paper is organized as follows. We give some definitions, basic lemmas, and an assumption in the next section. In section 3 we present a new iterative method that generates a sequence of lower bounds for the pseudospectral abscissa. In section 4 we precisely characterize the fixed points of the map associated with the iterative method and argue that, at least in practice, the only attractive fixed points correspond to local maximizers of (1.5). Then in section 5 we derive a local convergence analysis which depends on a perturbation theorem for the normalized eigenprojection of a matrix and on a characterization of the group inverse (reduced resolvent) of a singular matrix defined by a rank-one perturbation, both of which seem to be new. The results establish that the algorithm is linearly convergent to local maximizers of (1.5) for sufficiently small ε, but in practice we find that convergence takes place for much larger values of ε. In section 6 we give a simple modification to the algorithm which guarantees that it generates a monotonically increasing sequence of lower bounds for the spectral abscissa. This is the version that we recommend in practice. In section 7 we extend the ideas discussed for the pseudospectral abscissa to the computation of the pseudospectral radius.

In sections 8 and 9 we illustrate the behavior of the pseudospectral abscissa and radius algorithms on some examples from the literature, both with dense and sparse structure. We also discuss the computational complexity of the new algorithms. Here the most important point is that, for large sparse matrices, the cost of the algorithm to compute the pseudospectral abscissa or radius is typically only a constant times the cost of computing the spectral abscissa or radius, where the value of this constant may be less than 10 in many cases of practical interest.

2. Basic results, definitions, and an assumption. We will need the following standard perturbation result for eigenvalues [HJ90, Lemma 6.3.10 and Theorem 6.3.12], [Kat95, section II.1.1].

LEMMA 2.1. *Let $t \in \mathbb{R}$, and consider the $n \times n$ matrix family $C(t) = C_0 + tC_1$. Let $\lambda(t)$ be an eigenvalue of $C(t)$ converging to a simple eigenvalue λ_0 of C_0 as $t \to 0$. Then $y_0^* x_0 \neq 0$ and $\lambda(t)$ is analytic near $t = 0$ with*

$$\left.\frac{d\lambda(t)}{dt}\right|_{t=0} = \frac{y_0^* C_1 x_0}{y_0^* x_0},$$

where x_0 and y_0 are, respectively, right and left eigenvectors of C_0 corresponding to λ_0, that is, $(C_0 - \lambda_0 I)x = 0$ and $y^*(C_0 - \lambda_0 I) = 0$.

Note that x_0 and y_0 can be independently scaled by any complex number without changing the equations above. Thus, any pair of right and left eigenvectors x and y corresponding to a simple eigenvalue can be scaled to have the following property.

DEFINITION 2.2. *A pair of complex vectors x and y is called* RP-compatible *if $\|x\| = \|y\| = 1$ and y^*x is real and positive, and therefore in the interval $(0, 1]$.*

Clearly, a pair of RP-compatible right and left eigenvectors x and y may be replaced by μx and μy for any unimodular μ (with $|\mu| = 1$) without changing the property of RP-compatibility.

Next we turn to singular values. The following well-known result can be obtained from Lemma 2.1 by using the standard equivalence between singular values of A and eigenvalues of $[0\ A; A^*\ 0]$ [HJ90, Theorem 7.3.7]. The minus signs are nonstandard, but they will be convenient.

LEMMA 2.3. *Let $t \in \mathbb{R}$, and consider the $n \times n$ matrix family $C(t) = C_0 + tC_1$. Let $\sigma(t)$ be a singular value of $C(t)$ converging to a simple nonzero singular value σ_0 of C_0 as $t \to 0$. Then $\sigma(t)$ is analytic near $t = 0$ with*

$$\left.\frac{d\sigma(t)}{dt}\right|_{t=0} = -u_0^* C_1 v_0,$$

where v_0 and $-u_0$ are, respectively, right and left singular vectors of C_0 corresponding to σ_0, that is, $\|v_0\| = \|u_0\| = 1$, $C_0 v_0 = -\sigma_0 u_0$ and $u_0^* C_0 = -\sigma_0 v_0^*$.

Note that as with RP-compatible eigenvectors, v_0 and $-u_0$ may be replaced by μv_0 and $-\mu u_0$ for any unimodular μ. That eigenvectors need to be chosen to be RP-compatible to reduce the degree of freedom to a single unimodular scalar (assuming the eigenvalue is simple) is a key difference between eigenvector and singular vector pairs.

DEFINITION 2.4. *By a* rightmost *eigenvalue of A or point in $\Lambda_\varepsilon(A)$ we mean one with largest real part. By a* locally rightmost *point in $\Lambda_\varepsilon(A)$ we mean a point z which is the rightmost point in $\Lambda_\varepsilon(A) \cap \mathcal{N}$, where $\mathcal{N}$ is a neighborhood of z.*

LEMMA 2.5. *Locally rightmost points in $\Lambda_\varepsilon(A)$ are isolated.*

Proof. If not, then $\partial\Lambda_\varepsilon(A)$ would need to contain a nontrivial line segment all of whose points have the same real part, say r, but this would imply that the Hamiltonian matrix $H(r)$ defined in (1.7) has an infinite number of eigenvalues, which is impossible. ∎

Now we give the main hypothesis of the paper.

Assumption 2.1. Let $\varepsilon > 0$, and let z be a local maximizer of (1.5), that is, a locally rightmost point in the ε-pseudospectrum of A. Then the smallest singular value of $A - zI$ is simple.

We shall assume throughout the paper that Assumption 2.1 holds. It is known that generically, that is, for almost all A, the smallest singular value of $A - zI$ is simple for all $z \in \mathbb{C}$ [BLO03, section 2]. If the rightmost eigenvalues λ of A are all nonderogatory, that is, their geometric multiplicity (the multiplicity of the smallest singular value of $A - \lambda I$) is one, it clearly follows by continuity that Assumption 2.1 holds for all sufficiently small ε, and it seems to be an open question as to whether in fact this is true for all $\varepsilon > 0$.

If A is normal with distinct eigenvalues, then $\Lambda_\varepsilon(A)$ is the union of n disks with radius ε centered at the eigenvalues of A, and it is easy to see that the perpendicular bisector of the line segment joining the two eigenvalues closest to each other contains

an interval of points z for which $\sigma_n(A-zI)$ is a double singular value, but these points are not maximizers of (1.5). See [ABBO11] for some examples of nonnormal matrices for which double singular values of $A-zI$ occur, but again such points z are not local maximizers. See also [LP08] for an extensive study of mathematical properties of the pseudospectrum.

Under Assumption 2.1, we have the following results.

LEMMA 2.6. *Let $\varepsilon > 0$, and let z be a local maximizer of* (1.5)*, that is, a locally rightmost point in the ε-pseudospectrum of A. Then the right and left singular vectors v and $-u$ corresponding to the smallest singular value of $A - zI$ are not mutually orthogonal, that is, $u^*v \neq 0$.*

Proof. By Lemma 1.1, v and $-u$ are, respectively, right and left eigenvectors of $B = A + \varepsilon uv^*$ corresponding to the eigenvalue z. If $u^*v = 0$, then by Lemma 2.1, the eigenvalue z must have algebraic multiplicity m greater than one. It follows from [ABBO11, Theorem 9] that z cannot be a local maximizer of (1.5). ∎

LEMMA 2.7. *A necessary condition for z to be a local maximizer of the optimization problem* (1.5) *defining the pseudospectral abscissa is*

$$\sigma_n(A - zI) = \varepsilon \quad \textit{and} \quad u^*v > 0, \tag{2.1}$$

where v and $-u$ are, respectively, right and left singular vectors corresponding to $\sigma_n(A - zI)$.

Proof. Since Λ_ε is compact, at least one local maximizer must exist, and given the maximization objective, all local maximizers must lie on $\partial\Lambda_\varepsilon$. The standard first-order necessary condition for x to be a local maximizer of an optimization problem

$$\max\{f(\xi) : g(\xi) \leq 0,\ \xi \in \mathbb{R}^2\},$$

when f, g are continuously differentiable and $g(\xi) = 0$, $\nabla g(\xi) \neq 0$, is the existence of a Lagrange multiplier $\nu \geq 0$ such that $\nabla f(\xi) = \nu \nabla g(\xi)$. In our case, identifying $\mathbb{C}$ with $\mathbb{R}^2$, the gradient of the maximization objective is the real number 1 and the gradient of the constraint $\sigma_n(A - zI) - \varepsilon$ is, by Lemma 2.3 and the chain rule, v^*u. By Lemma 2.6, $v^*u \neq 0$. Thus, $v^*u = u^*v = \frac{1}{\nu}$, a positive real number. ∎

We also have $u^*v \leq 1$ if u and v are unit vectors.

LEMMA 2.8. *Let λ be a simple eigenvalue of A. Then for ε sufficiently small, the component of Λ_ε containing λ contains exactly one local maximizer and no local minimizers of* (1.5).

Proof. This follows from the fact that, since λ is simple, the component of Λ_ε containing λ is strictly convex for sufficiently small ε [BLO07, Corollary 4.2]. ∎

3. A new algorithm. We now introduce a new algorithm for approximating the pseudospectral abscissa, mainly intended for large sparse matrices.

Let z be a maximizer of (1.5), that is, a rightmost point in the ε-pseudospectrum of A. Lemma 1.1 states that, given z, we can use the SVD of $A-zI$ to construct a rank-one perturbation of A whose norm is ε and which has an eigenvalue z. Let v and $-u$, respectively, denote right and left singular vectors corresponding to $\varepsilon = \sigma_n(A - zI)$. Then $A-zI+\varepsilon uv^*$ is singular, so $B = A+\varepsilon uv^*$ is an ε-norm rank-one perturbation of A with eigenvalue z. The idea of the new algorithm is to generate a sequence of such rank-one perturbations $\varepsilon L_k = \varepsilon u_k v_k^*$, with $\|u_k\| = \|v_k\| = 1$ and hence $\|\varepsilon L_k\| = \varepsilon$, ideally converging to an optimal rank-one perturbation $\varepsilon L = \varepsilon uv^*$, without making

any Hamiltonian eigenvalue decompositions or SVDs. (It may help to think of the letter L as a mnemonic for low-rank.) The only allowable matrix operations are the computation of eigenvalues with largest real part and their corresponding right and left eigenvectors, which can be done efficiently using an iterative method.

The first step of the iteration computes L_0. We start by computing z_0, an eigenvalue of A with largest real part, along with its right and left eigenvectors x_0 and y_0, respectively, normalized so that x_0 and y_0 are RP-compatible. Assume that the eigenvalue z_0 is simple and consider the matrix-valued function $A(t) = A + tuv^*$, with $\|u\| = \|v\| = 1$ and with $\lambda(t)$ denoting the eigenvalue of $A(t)$ that converges to z_0 as $t \to 0$. Then from Lemma 2.1, we have

$$\operatorname{Re} \left.\frac{d\lambda(t)}{dt}\right|_{t=0} = \frac{\operatorname{Re}(y_0^*\,(uv^*)\,x_0)}{y_0^*x_0} = \frac{\operatorname{Re}((y_0^*u)(v^*x_0))}{y_0^*x_0}) \le \frac{1}{y_0^*x_0}.$$

Equality is achieved if $u = y_0$ and $v = x_0$, so this choice for u, v gives Re $\lambda(t)$ with locally maximal growth as t increases from 0. If $\alpha(A + \varepsilon y_0 x_0^*) \ge \alpha(A) + \varepsilon y_0^* x_0$, set $L_0 = y_0 x_0^*$ so that the matrix $A(\varepsilon) = A + \varepsilon L_0$ is an ε-norm rank-one perturbation of A with an eigenvalue at least a distance of $\varepsilon y^* x$ to the right of z_0. If $\alpha(A + \varepsilon y_0 x_0^*) < \alpha(A) + \varepsilon y_0^* x_0$ (which could happen if ε is not sufficiently small), then it is convenient to instead set $u = x$ and $v = y$ and hence $L_0 = x_0 y_0^*$, which implies $\alpha(A + \delta L_0) = \alpha(A) + \delta y_0^* x_0$ for all δ, including $\delta = \varepsilon$. In this way, after the first step we always have $\alpha(A + \varepsilon L_0) \ge \alpha(A) + \varepsilon y_0^* x_0$.

We now consider subsequent steps of the algorithm. Let $\widehat{B} = A + \varepsilon \widehat{L}$, where $\widehat{L} = \widehat{u}\widehat{v}^*$ with $\widehat{v}$ and $\widehat{u}$ an RP-compatible pair of vectors. Consider the matrix-valued function

$$B(t) = \widehat{B} + t\,(uv^* - \widehat{u}\widehat{v}^*) = A + \varepsilon\widehat{L} + t\left(uv^* - \widehat{L}\right),$$

which is chosen so that $B(t)$ is an ε-norm rank-one perturbation of A for $t = \varepsilon$ as well as for $t = 0$. Let $\widehat{z}$ be a rightmost eigenvalue of $\widehat{B}$. Assume $\widehat{z}$ is simple, and let $\widehat{x}, \widehat{y}$ be RP-compatible corresponding right and left eigenvectors of $\widehat{B}$. Define $\lambda(t)$ to be the eigenvalue of $B(t)$ converging to $\widehat{z}$ as $t \to 0$. Then

$$\operatorname{Re} \left.\frac{d\lambda(t)}{dt}\right|_{t=0} = \operatorname{Re}\frac{\widehat{y}^*\,(uv^* - \widehat{u}\widehat{v}^*)\,\widehat{x}}{\widehat{y}^*\widehat{x}}.$$

If we set $u = \widehat{y}$ and $v = \widehat{x}$, we obtain

$$\operatorname{Re} \left.\frac{d\lambda(t)}{dt}\right|_{t=0} = \frac{1 - \operatorname{Re}\,((\widehat{y}^*\widehat{u})(\widehat{v}^*\widehat{x}))}{\widehat{y}^*\widehat{x}} > 0$$

unless $\widehat{y} = \mu\widehat{u}$, $\widehat{x} = \mu\widehat{v}$ for some unimodular μ. This motivates setting $u = \widehat{y}$, $v = \widehat{x}$, so the next rank-one perturbation is $\varepsilon uv^* = \varepsilon\widehat{y}\widehat{x}^*$.

We summarize the algorithm below.

Algorithm PSA0. Let z_0 be a rightmost eigenvalue of A, with corresponding right and left eigenvectors x_0 and y_0 normalized so that they are RP-compatible. If $\alpha(A + \varepsilon y_0 x_0^*) \ge \operatorname{Re}(z_0) + \varepsilon y_0^* x_0$, set $L_0 = y_0 x_0^*$; otherwise, set $L_0 = x_0 y_0^*$. Set $B_1 = A + \varepsilon L_0$.

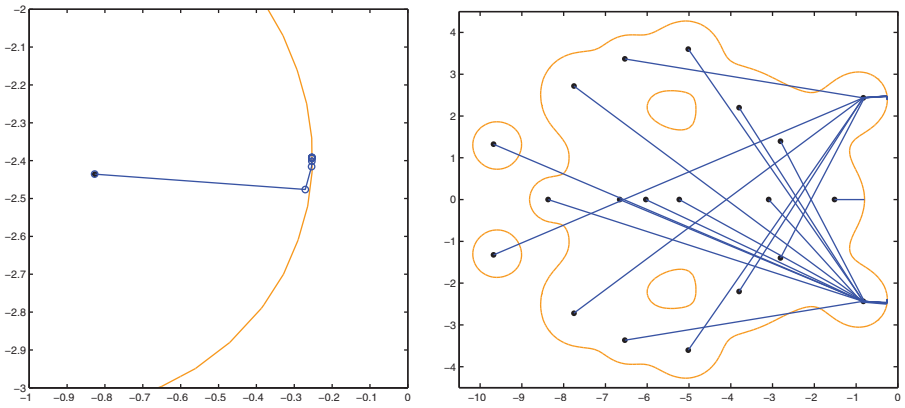

FIG. 3.1. *Illustration of the convergence of Algorithm PSA0 to approximate the pseudospectral abscissa* $\alpha_\varepsilon(A_0)$ *for* $\varepsilon = 10^{-0.5}$. *On the left, the iteration starts from the rightmost eigenvalue and converges to the rightmost point in* $\Lambda_\varepsilon(A_0)$. *On the right, the iterates are shown starting from* each *of the* 20 *eigenvalues.*

For $k = 1, 2, \ldots$:

Let z_k be the rightmost eigenvalue of B_k (if there is more than one, choose the one closest to z_{k-1}). Let x_k and y_k be corresponding right and left eigenvectors normalized so that they are RP-compatible. Set $L_k = y_k x_k^*$ and $B_{k+1} = A + \varepsilon L_k$.

We say that the algorithm breaks down if it generates an eigenvalue z_k which is not simple: in this case x_k and y_k are not defined. It is clear from continuity that this cannot happen for sufficiently small ε if the rightmost eigenvalues of A are all simple, and in practice, it never happens regardless of the value of ε. Usually, the sequence $\{\mathrm{Re}\ z_k\}$ increases monotonically, but this is not always the case. In section 6, we discuss a modified algorithm which guarantees that $\{\mathrm{Re}\ z_k\}$ increases monotonically. We defer a discussion of a stopping criterion to section 8.

The behavior of the algorithm is illustrated in Figure 3.1 for the same matrix A_0 whose pseudospectra were plotted in Figure 1.1, with $\varepsilon = 10^{-0.5}$. The iterates z_k of the algorithm are plotted as circles connected by line segments; by construction, all z_k lie in $\Lambda_\varepsilon(A_0)$. In the left panel, a close-up view shows the iteration starting from a rightmost eigenvalue and converging to a rightmost point in $\Lambda_\varepsilon(A_0)$. In the right panel, we show the iterates that are obtained when the algorithm is initialized at *each* of the 20 eigenvalues. In all but one case, convergence takes place to a globally rightmost point in $\Lambda_\varepsilon(A_0)$, but in one case, convergence takes place along the real axis to a boundary point that is only a local maximizer.

As two more examples, consider

$$A_1 = \begin{pmatrix} -\frac{1}{2} - \mathbf{i} & \mathbf{i} \\ -2 + \mathbf{i} & \frac{1}{2} \end{pmatrix} \quad \text{and} \quad A_2 = \begin{pmatrix} -1 - \mathbf{i} & \mathbf{i} & 0 \\ -2 + \mathbf{i} & \frac{1}{2} & 1 + \mathbf{i} \\ 0 & -\mathbf{i} & \frac{1}{2} + 2\mathbf{i} \end{pmatrix}. \tag{3.1}$$

The behavior of Algorithm PSA0 applied to A_1 and A_2 is shown on the left and right sides of Figure 3.2, respectively. For A_1, with $\varepsilon = 10^{-0.4}$, the algorithm converges to the global maximizer of (1.5) regardless of whether it starts at the leftmost or rightmost eigenvalue, but for $\varepsilon = 10^{-0.1}$, convergence takes place to the global maximizer if z_0 is the rightmost eigenvalue and to a local maximizer if z_0 is the leftmost eigen-

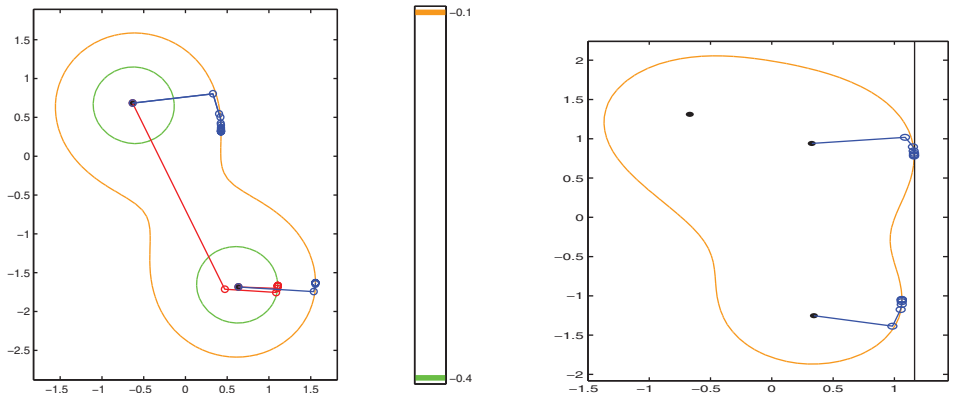

FIG. 3.2. *Behavior of Algorithm PSA0 applied to two matrices defined in* (3.1). *Left panel: the matrix* A_1 *with* $\varepsilon = 10^{-0.1}$ *and* $\varepsilon = 10^{-0.4}$, *starting from both eigenvalues. For the larger value of* ε *(outer contour), one sequence converges to a locally rightmost point and the other to the globally rightmost point, while for the smaller value of* ε *(inner contour), both converge to the globally rightmost point. Right panel: the matrix* A_2, *with* $\varepsilon = 10^{-0.4}$, *starting from two of the eigenvalues. The sequence initialized at the rightmost eigenvalue converges to a locally (but not globally) rightmost point.*

value. For A_2, with $\varepsilon = 10^{-0.4}$, starting from the rightmost eigenvalue, the algorithm converges to a local maximizer, but starting from another eigenvalue, it converges to the global maximizer.

These two examples show clearly that convergence of Algorithm PSA0 to a global maximizer of (1.5) cannot be guaranteed, even by starting at a rightmost eigenvalue. However, we shall see in section 8 that, for examples of practical interest, convergence to a global maximizer is normal behavior. Furthermore, global optimality can be checked by a single step of the criss-cross algorithm when this is computationally feasible (see section 1).

A second point of particular interest in Figures 3.1 and 3.2 is the *vertical dynamics*: the iterates typically approach the boundary of Λ_ε rapidly and then converge to a global or local maximizer nearly tangentially. We will see why this is the case in section 5.

We are, of course, interested only in nonnormal matrices. For normal matrices, the following property is immediate.

LEMMA 3.1. *If* A *is normal and* z_0 *is a rightmost eigenvalue of* A, *then the iteration converges in one step, with* $\mathrm{Re}\ z_1 = \alpha_\varepsilon(A)$.

Proof. When A is normal the pseudospectrum is the union of disks and the pseudospectral abscissa is

$$\alpha_\varepsilon(A) = \alpha(A) + \varepsilon = \max\{\mathrm{Re}\ z : z \in \Lambda(A + \varepsilon x x^*) \text{ for } \|x\| = 1\}.$$

The maximum is attained when $x = x_0$, the eigenvector corresponding to z_0. Since $B_1 = A + \varepsilon x_0 x_0^*$ is the first matrix computed by Algorithm PSA0, the method converges in one step. ∎

Another easy but powerful observation is the following.

LEMMA 3.2. *Suppose* A *is nonnegative, that is,* $A_{ij} \geq 0$ *for all* i, j, *and irreducible* [HJ90, Definition 6.2.22]. *Then Algorithm PSA0 does not break down and it generates a sequence of positive matrices* B_k.

Proof. By the Perron–Frobenius theorem [HJ90, Theorems 8.4.4 and 8.4.6], the rightmost eigenvalue z_0 of A is simple, real, and positive with positive right and left eigenvectors x_0 and y_0. Clearly, x_0 and y_0 are RP-compatible. It follows immediately that B_1 is nonnegative and irreducible, in fact strictly positive, and by induction, so is B_k for all k. ∎

We discuss convergence properties of the algorithm in the sections that follow. By construction, the sequence $\{\mathrm{Re}(z_k)\}$ is bounded above by $\alpha_\varepsilon(A)$, the global maximum of (1.5).

4. Fixed points of the iteration. Let us denote by $\mathcal{T}_\varepsilon$ the map that generates the pair (x_{k+1}, y_{k+1}) from the pair (x_k, y_k) as defined by Algorithm PSA0. Thus, $\mathcal{T}_\varepsilon$ maps the RP-compatible pair of vectors (x_k, y_k) to an RP-compatible pair of right and left eigenvectors (x_{k+1}, y_{k+1}) for the rightmost eigenvalue z_{k+1} of $B_{k+1} = A + \varepsilon y_k x_k^*$ (if there is more than one, then z_{k+1} is defined to be the one closest to z_k). Equivalently, $\mathcal{T}_\varepsilon$ maps a rank-one matrix $L_k = y_k x_k^*$ with norm one and a real positive eigenvalue $y_k^* x_k$ to a rank-one matrix L_{k+1} with norm one and a real positive eigenvalue $y_{k+1}^* x_{k+1}$.

DEFINITION 4.1. *The pair (x_k, y_k) is a* fixed point *of the map $\mathcal{T}_\varepsilon$ if $x_{k+1} = \mu x_k$, $y_{k+1} = \mu y_k$ for some unimodular μ. Thus, (x_k, y_k) is an RP-compatible pair of right and left eigenvectors of A corresponding to a rightmost eigenvalue of $A + \varepsilon y_k x_k^*$. Equivalently, L_k is a fixed point of the iteration if $L_{k+1} = L_k$.*

THEOREM 4.2. *Suppose (x, y) is a fixed point of the map $\mathcal{T}_\varepsilon$ corresponding to a rightmost eigenvalue λ of $A + \varepsilon y x^*$. Then $A - \lambda I$ has a singular value equal to ε, and furthermore, if it is the least singular value $\sigma_n(A - \lambda I)$, then λ satisfies the first-order necessary condition for a local maximizer of* (1.5) *given in* (2.1). *Conversely, suppose that λ satisfies* (2.1), *and let v and $-u$ denote unit right and left singular vectors v corresponding to $\sigma_n(A - \lambda I)$. Then λ is an eigenvalue of $A + \varepsilon y x^*$, and if it is a rightmost eigenvalue, then (x, y) is a fixed point of $\mathcal{T}_\varepsilon$.*

Proof. We have

$$(A + \varepsilon y x^* - \lambda I)x = 0, \quad y^*(A + \varepsilon y x^* - \lambda I) = 0$$

with λ a rightmost eigenvalue of $A + \varepsilon y x^*$. Therefore,

$$(A - \lambda I)x = -\varepsilon y, \quad y^*(A - \lambda I) = -\varepsilon x^*, \tag{4.1}$$

so $A - \lambda I$ has a singular value equal to ε with x and $-y$ corresponding right and left singular vectors, with $y^* x$ real and positive since x, y are RP-compatible. Suppose it is the *least* singular value, $\sigma_n(A - \lambda I)$. Then by Lemma 2.7, λ satisfies the first-order necessary condition for a local maximizer of (1.5).

Conversely, suppose $\sigma_n(A - \lambda I) = \varepsilon$ and the corresponding unit right and left singular vectors v and $-u$ satisfy $u^* v > 0$. It follows from the equivalence of (1.2) and (1.3) that $x = v$ and $y = -u$ are an RP-compatible pair of right and left eigenvectors of $A + \varepsilon y x^*$ corresponding to the eigenvalue λ, and so if λ is a rightmost eigenvalue of $A + \varepsilon y x^*$, then (x, y) is a fixed point of $\mathcal{T}_\varepsilon$. ∎

Theorem 4.2 is illustrated in Figure 4.1, which plots the level curves

$$\{z : \sigma_{n-1}(A - zI) = \varepsilon\} \quad \text{and} \quad \{z : \sigma_n(A - zI) = \varepsilon\}$$

for a randomly generated matrix A with $n = 6$ and with $\varepsilon = 1$. The outer curve is $\partial\Lambda_\varepsilon(A)$, the ε-level set for the smallest singular value, and the inner curve is the ε-level set for the second smallest. There are eight local extrema of the real part

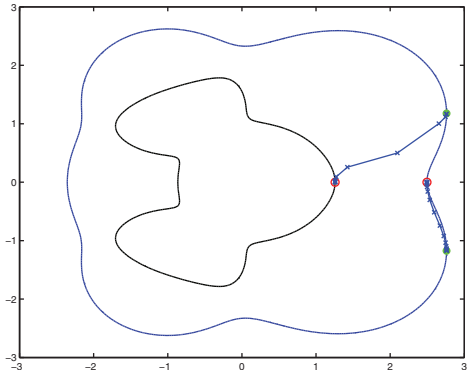

FIG. 4.1. *Eigenvalues λ corresponding to fixed points (x, y) of $\mathcal{T}_\varepsilon$ for a randomly generated matrix A with $n = 6$ and with $\varepsilon = 1$. The outer curve is the boundary of the pseudospectrum $\Lambda_\varepsilon(A)$ and the inner curve is the ε-level set of $\sigma_{n-1}(A - zI)$. There are four fixed points, but only those for which λ is a maximizer of* (1.5) *are attractive (green solid dots); the other two are repulsive (red circles).*

function on the set $\{z : \sigma_n(A - zI) \le \varepsilon\}$, five on the left and three on the right, but only the three on the right satisfy the first-order necessary condition for a maximizer. Of these, two are global maximizers (marked as green solid dots) and one is a local *minimizer* (red circle). There is also a second point marked as a red circle: the global maximizer of the real part over the set $\{z : \sigma_{n-1}(A - zI) \le \varepsilon\}$. Each of these four marked points λ is a rightmost eigenvalue of a corresponding matrix $A + \varepsilon yx^*$ for which the pair (x, y) is a fixed point.

A particularly interesting observation is that *only the two maximizers (green solid dots) are attractive points* for Algorithm PSA0. If the algorithm is initialized near λ, setting B_1 to $A+\varepsilon\widetilde{y}\widetilde{x}^*$, where $(\widetilde{x}, \widetilde{y})$ is a small perturbation of the corresponding fixed point (x, y), the iterates z_k generated by Algorithm PSA0 move away from λ, towards one of the maximizers, as shown by the crosses plotted in Figure 4.1. This clearly indicates that the fixed points corresponding to the local minimizer and the internal point are *repulsive*. We will comment further on this phenomenon in section 5.

We conjecture that *the only attractive fixed points for Algorithm PSA*0 *correspond to points λ that are local maximizers of* (1.5). Furthermore, these are the only possible kinds of fixed points when ε is sufficiently small. Local minimizers cannot occur for small ε (see Lemma 2.8), and internal fixed points cannot occur because the ε-level set of $\sigma_k(A - zI)$ is empty for $k < n$ when ε is sufficiently small, a multiple smallest singular value being ruled out at local maximizers by Assumption 2.1.

Since we are not interested in the internal fixed points, we make the following definition.

DEFINITION 4.3. *A fixed point (x, y) of $\mathcal{T}_\varepsilon$ corresponding to λ for which $\sigma_n(A - \lambda I) = \varepsilon$ is called a* boundary *fixed point. A fixed point that is not a boundary fixed point is an* internal *fixed point.*

5. Local error analysis. In this section, we need to use perturbation theory for *normalized eigenprojections*. Although we need the following result only for a linear matrix family, we state it in more generality since it may be of independent interest. First we need a definition: see [MS88] as well as [Kat95, section I.5.3] for more details. Note that the group inverse is not the same as the more well known Moore–Penrose pseudoinverse.

DEFINITION 5.1. *The* group inverse *(reduced resolvent) of a matrix M, denoted by $M^{\#}$, is the unique matrix G satisfying $MG = GM$, $GMG = G$, and $MGM = M$.*

THEOREM 5.2. *Consider an $n \times n$ complex analytic matrix family $C(\tau) = C_0 + \tau E + O(\tau^2)$, where $\tau \in \mathbb{C}$. Let $\lambda(\tau)$ be a simple eigenvalue of $C(\tau)$ in a neighborhood $\mathcal{N}$ of $\tau = 0$, with corresponding right and left eigenvectors $x(\tau)$ and $y(\tau)$, normalized to be RP-compatible; that is, $\|x(\tau)\| = \|y(\tau)\| = 1$ and $y(\tau)^* x(\tau)$ is real and positive on $\mathcal{N}$. Define $w(\tau) = x(\tau)/(y(\tau)^* x(\tau))$, so that $y(\tau)^* w(\tau) = 1$ and hence $P(\tau) = w(\tau)y(\tau)^*$ is the classical eigenprojection for $\lambda(\tau)$, satisfying*

$$C(\tau)P(\tau) = P(\tau)C(\tau) = P(\tau)C(\tau)P(\tau) = \lambda(\tau)P(\tau).$$

Define

$$Q(\tau) = \frac{P(\tau)}{\|P(\tau)\|} = x(\tau)y(\tau)^*,$$

the normalized eigenprojection for $\lambda(\tau)$. Finally let

$$G = (C_0 - \lambda I)^{\#}, \quad \beta = x^* G E x, \quad \gamma = y^* E G y,$$

where $\lambda = \lambda(0)$, $x = x(0), y = y(0)$. Then $P(\tau)$ is analytic and $Q(\tau)$ is $\mathcal{C}^\infty$ in $\mathcal{N}$, and their derivatives at $\tau = 0$ are

$$P' = -GEP - PEG, \quad Q' = \mathrm{Re}(\beta + \gamma)Q - GEQ - QEG,$$

where $P = P(0)$, $Q = Q(0)$.

Proof. The fact that P is analytic in $\mathcal{N}$ and the formula for P' are well known [Kat95, section II.2.1, eq. (2.14)] (we include the result here for comparative purposes). It follows that Q is $\mathcal{C}^\infty$ in $\mathcal{N}$, since $y^* x$ cannot vanish in $\mathcal{N}$ under the assumption that $\lambda(\tau)$ is simple. To find the derivative of Q we make use of the eigenvector perturbation theory of Meyer and Stewart [MS88]. Note that although the normalizations described above do not uniquely define the eigenvectors x, y, and w, because of an unspecified unimodular factor μ, [Kat95, section II.4.1] discusses how eigenvectors may be defined as smooth functions of a parametrization. In [MS88, Corollary 3, p. 684] it is shown that, given a smooth parametrization, the derivatives of w and y at $\tau = 0$ are

$$w' = -\gamma w - GEw, \quad y' = \overline{\gamma} y - G^* E^* y.$$

Define $f = \|w\| = (w^* w)^{1/2}$, so the normalized right eigenvector is $x = w/f$. Then

$$f' = \frac{\mathrm{Re}(w^* w')}{f},$$

and

$$\begin{aligned} x' = \frac{f w' - w f'}{f^2} &= \frac{(-\gamma w - GEw)f - w\mathrm{Re}(w^*(-\gamma w - GEw))/f}{f^2} \\ &= -\gamma x - GEx - x\mathrm{Re}\left(x^*(-\gamma x - GEx)\right) \\ &= -\gamma x - GEx + \mathrm{Re}(\gamma + \beta)x \\ &= (\mathrm{Re}\beta - \mathbf{i}\mathrm{Im}\gamma)\, x - GEx. \end{aligned}$$

Therefore

$$\begin{aligned} Q' &= x\left(y'\right)^* + x' y^* = x\left(\overline{\gamma} y - G^* E^* y\right)^* + \left((\mathrm{Re}\beta - \mathbf{i}\mathrm{Im}\gamma)x - GEx\right) y^* \\ &= \mathrm{Re}(\beta + \gamma)xy^* - GExy^* - xy^* EG. \qquad \square \end{aligned}$$

Now we apply this result to the error analysis of Algorithm PSA0. For convenience, assume that $L_0 = y_0 x_0^*$ (as opposed to $x_0 y_0^*$).

THEOREM 5.3. *Suppose that (x, y) is a boundary fixed point of $\mathcal{T}_\varepsilon$ corresponding to a rightmost eigenvalue λ of $B = A + \varepsilon y x^*$. Let the sequence B_k and $L_k = y_k x_k^*$ be defined as in Algorithm PSA0, and set, for $k = 1, 2, \ldots$,*

$$E_k = B_k - B = (A + \varepsilon L_{k-1}) - (A + \varepsilon L) = \varepsilon\,(L_{k-1} - L)\,. \tag{5.1}$$

Let $\delta_k = \|E_k\|$. Then, if δ_k is sufficiently small, we have

$$E_{k+1} = \varepsilon\Big(\mathrm{Re}(\beta_k + \gamma_k)L - LE_k^* G^* - G^* E_k^* L\Big) + \mathcal{O}(\delta_k^2), \tag{5.2}$$

where

$$G = (B - \lambda I)^{\#}, \quad \beta_k = x^* G E_k x, \quad \gamma_k = y^* E_k G y. \tag{5.3}$$

Proof. We have from (5.1) that

$$E_{k+1} = \varepsilon(L_k - L), \tag{5.4}$$

so we need an estimate of the difference between L_k and L. By its definition in Algorithm PSA0, L_k is the conjugate transpose of the normalized eigenprojection $x_k y_k^*$ for a rightmost eigenvalue z_k of B_k, while L is the conjugate transpose of the normalized eigenprojection xy^* for the eigenvalue λ of B. Let us define the matrix family

$$C(t) = B + t(B_k - B) = B + tE_k,$$

so $C(0) = B$ and $C(1) = B_k$. From Theorem 5.2, substituting $C_0 = B$, $E = E_k$, $\beta = \beta_k$, $\gamma = \gamma_k$, and $Q = L^*$, the derivative of the conjugated normalized eigenprojection of $C(t)$ at $t = 0$ is

$$\mathrm{Re}(\beta_k + \gamma_k)L - LE_k^* G^* - G^* E_k^* L.$$

Therefore

$$L_k = L + \mathrm{Re}(\beta_k + \gamma_k)L - LE_k^* G^* - G^* E_k^* L + \mathcal{O}(\delta_k^2). \tag{5.5}$$

The result now follows from combining (5.4) and (5.5). ∎

A corollary addresses the convergence of z_k to the rightmost eigenvalue λ of $B = A + \varepsilon y x^*$.

COROLLARY 5.4. *Under the assumptions of Theorem 5.3, the rightmost eigenvalue z_{k+1} of B_{k+1} satisfies*

$$z_{k+1} - \lambda = \frac{\varepsilon}{y^* x}\mathrm{Im}(\beta_k + \gamma_k)\mathbf{i} + \mathcal{O}(\delta_k^2).$$

Proof. Since $B_{k+1} = B + E_{k+1}$, we have from Lemma 2.1 that

$$z_{k+1} = \lambda + \frac{y^* E_{k+1} x}{y^* x} + \mathcal{O}(\delta_{k+1}^2).$$

Using (5.2), (5.3), the fact that $L = yx^*$, and the fact that $\delta_{k+1} = \mathcal{O}(\delta_k)$ by (5.2), this gives

$$\begin{aligned} z_{k+1} &= \lambda + \frac{\varepsilon}{y^*x} y^* \Big(\mathrm{Re}(\beta_k + \gamma_k) L - L E_k^* G^* - G^* E_k^* L \Big) x + \mathcal{O}(\delta_k^2) \\ &= \lambda + \frac{\varepsilon}{y^*x} \Big(\mathrm{Re}(\beta_k + \gamma_k) - \overline{\gamma}_k - \overline{\beta}_k \Big) + \mathcal{O}(\delta_k^2), \end{aligned}$$

proving the result. ∎

This explains the observed behavior of the sequence of rightmost eigenvalues $\{z_k\}$ of the matrices $\{B_k\}$, that is, the typical vertical dynamics of the iteration close to fixed points (see Figures 3.1 and 3.2).

Formulas for the group inverse of a matrix M may be found in [MS88] (using the SVD of M) and in [Kat95, section II.2.2, Remark 2.2] (using the spectral decomposition of M; see also Stewart [Ste01, p. 46], where essentially the same formula appears in a more general form with a sign difference). Neither of these is convenient for our purpose. The following theorem establishes a useful formula for the group inverse of the singular matrix $B - \lambda I = A + \varepsilon yx^* - \lambda I$.

THEOREM 5.5. *Suppose that (x, y) is a boundary fixed point of $\mathcal{T}_\varepsilon$ corresponding to a rightmost eigenvalue λ of $B = A + \varepsilon yx^*$. Then the group inverse of $B - \lambda I$ is given by*

$$G = (A + \varepsilon yx^* - \lambda I)^{\#} = (I - wy^*)\,(A - \lambda I)^{-1}\,(I - wy^*)\,, \tag{5.6}$$

where $w = \varrho x$ and $\varrho = \frac{1}{y^*x}$, so $y^*w = 1$. Furthermore,

$$Gx = 0, \quad y^*G = 0, \quad \text{and} \quad \|G\| \le \frac{\varrho^2}{\sigma_{n-1}\,(A - \lambda I)}. \tag{5.7}$$

Proof. We need to prove that the matrix G given in (5.6) satisfies Definition 5.1, that is, the following hold:

(i) $(A + \varepsilon yx^* - \lambda I)G = G(A + \varepsilon yx^* - \lambda I)$;
(ii) $G(A + \varepsilon yx^* - \lambda I)G = G$;
(iii) $(A + \varepsilon yx^* - \lambda I)G(A + \varepsilon yx^* - \lambda I) = (A + \varepsilon yx^* - \lambda I)$.

We will repeatedly make use of (4.1), which holds since (x, y) is a fixed point of $\mathcal{T}_\varepsilon$.

(i) First compute $(A + \varepsilon yx^* - \lambda I)G$. Using (4.1), we obtain

$$\begin{aligned} (A + \varepsilon yx^* - \lambda I)\,(I - wy^*) &= A + \varepsilon yx^* - \lambda I - \varrho(A - \lambda I)xy^* - \varrho\varepsilon yy^* \\ &= A + \varepsilon yx^* - \lambda I. \end{aligned}$$

Then, still making use of (4.1), we get

$$(A + \varepsilon yx^* - \lambda I)\,(A - \lambda I)^{-1} = I + \varepsilon yx^*\,(A - \lambda I)^{-1} = I - yy^*.$$

In conclusion

$$(A + \varepsilon yx^* - \lambda I)\,G = (I - yy^*)\,(I - wy^*) = I - wy^*. \tag{5.8}$$

Similarly, compute $G(A + \varepsilon yx^* - \lambda I)$. We have

$$\begin{aligned} (I - wy^*)\,(A + \varepsilon yx^* - \lambda I) &= A + \varepsilon yx^* - \lambda I - \varrho xy^*(A - \lambda I) - \varepsilon\varrho xx^* \\ &= A + \varepsilon yx^* - \lambda I. \end{aligned}$$

Then we have

$$(A-\lambda I)^{-1}(A+\varepsilon yx^*-\lambda I) = I+\varepsilon(A-\lambda I)^{-1}yx^* = I-xx^*.$$

Finally

$$G(A+\varepsilon yx^*-\lambda I) = (I-wy^*)(I-xx^*) = I-wy^*. \tag{5.9}$$

Comparing (5.8) and (5.9) proves (i).

(ii) Exploiting (5.9) we derive

$$G(A+\varepsilon yx^*-\lambda I)G = (I-wy^*)G = G,$$

which is a consequence of the property $y^*G = y^*(I-wy^*) = 0$. This proves (ii).

(iii) By (5.8) it follows that

$$\begin{aligned}(A+\varepsilon yx^*-\lambda I)G(A+\varepsilon yx^*-\lambda I) &= (I-wy^*)(A+\varepsilon yx^*-\lambda I)\\ &= (A+\varepsilon yx^*-\lambda I) - \varrho xy^*(A-\lambda I) - \varepsilon\varrho xx^* = A+\varepsilon yx^*-\lambda I,\end{aligned}$$

which proves (iii).

The middle statement in (5.7) has already been proved, and the first follows similarly. For the inequality, consider the SVD $A-\lambda I = USV^*$ with $S = \operatorname{diag}(s_1,\dots,s_n)$. Because (x,y) is a boundary fixed point associated with λ, we have $s_n = \sigma_n(A-\lambda I) = \varepsilon$, and the corresponding left and right singular vectors are $u_n = -y$ and $v_n = x$. By (5.6) we obtain

$$G = (I-wy^*)VS^{-1}U^*(I-wy^*) = (I-wy^*)\sum_{i=1}^{n} s_i^{-1}v_iu_i^*(I-wy^*). \tag{5.10}$$

Since $(I-wy^*)v_n = 0$, the last term in the sum disappears so that

$$G = (I-wy^*)V\Xi U^*(I-wy^*), \tag{5.11}$$

where Ξ is the diagonal matrix $\operatorname{diag}(s_1^{-1},\dots,s_{n-1}^{-1},0)$. Now, using submultiplicativity of the spectral norm we get

$$\|G\| \le \|I-wy^*\|^2\|\Xi\|. \tag{5.12}$$

The quantity $\|I-wy^*\|$ is the square root of the largest eigenvalue of the matrix

$$C = (I-yw^*)(I-wy^*) = I+ww^*-(wy^*+yw^*).$$

Observe that C has $n-2$ eigenvalues equal to 1 and, since it is singular, one eigenvalue equal to zero. The remaining eigenvalue is

$$\operatorname{tr}(C)-(n-2) = \varrho^2 > 1.$$

As a consequence

$$\|I-wy^*\| = \varrho.$$

The inequality in (5.7) follows using $\|\Xi\| = 1/s_{n-1} = 1/\sigma_{n-1}(A-\lambda I)$. ∎

We can now establish a sufficient condition for local convergence.

THEOREM 5.6. *Suppose that (x, y) is a boundary fixed point of $\mathcal{T}_\varepsilon$ corresponding to a rightmost eigenvalue λ of $B = A + \varepsilon yx^*$. Define*

$$r = \frac{4\varrho^2\varepsilon}{\sigma_{n-1}(A - \lambda I)}, \tag{5.13}$$

where $\varrho = \frac{1}{y^*x}$. Then if $r < 1$, and if $\delta_k = \|E_k\|$ is sufficiently small, the sequence $z_{k+1}, z_{k+2}, \ldots$ generated by Algorithm PSA0 converges linearly to λ with rate r.

Proof. Assume δ_k is sufficiently small. According to Theorem 5.3, in order to analyze local convergence of Algorithm PSA0 we need to consider whether the linear map $\mathcal{L}_\varepsilon$ defined by

$$\mathcal{L}_\varepsilon(\mathcal{E}_k) = \varepsilon\Big(\mathrm{Re}(\beta_k + \gamma_k)L - L\mathcal{E}_k^*G^* - G^*\mathcal{E}_k^*L\Big),$$

with $\beta_k = x^*G\mathcal{E}_kx$ and $\gamma_k = y^*\mathcal{E}_kGy$, is a contraction. By Theorem 5.5 and $\|L\| = 1$ we have

$$\|\mathcal{L}_\varepsilon(\mathcal{E}_k)\| \le \frac{4\varrho^2\varepsilon}{\sigma_{n-1}(A - \lambda I)}\|\mathcal{E}_k\| = r\|\mathcal{E}_k\|.$$

So, if $r < 1$, the map $\mathcal{L}_\varepsilon$ is a contraction, and $z_{k+1}, z_{k+2}, \ldots$ converges to λ with linear rate r. ∎

Because $\sigma_{n-1}(A - \lambda I) > \varepsilon$, we always have $r < 4\varrho^2$, but to show that $r < 1$ we need to assume that ε is sufficiently small. In the next theorem, we make the dependence on ε explicit.

THEOREM 5.7. *Let λ_0 be a simple rightmost eigenvalue of A, and let $\lambda(\varepsilon)$ be the local maximizer of* (1.5) *in the component of $\Lambda_\varepsilon(A)$ that contains λ_0, which is unique for ε sufficiently small (see Lemma* 2.8*). Let $x(\varepsilon)$ and $y(\varepsilon)$ be corresponding RP-compatible right and left eigenvectors, and suppose that $\lambda(\varepsilon)$ is the rightmost eigenvalue of $A + \varepsilon y(\varepsilon)x(\varepsilon)^*$, so that $(x(\varepsilon), y(\varepsilon))$ is a fixed point of $\mathcal{T}_\varepsilon$. Define the corresponding convergence factor $r(\varepsilon)$ as in* (5.13)*. Then as $\varepsilon \to 0$, $r(\varepsilon) = \mathcal{O}(\varepsilon)$. In other words, for sufficiently small ε, the linear rate of convergence of Algorithm PSA0 is arbitrarily fast.*

Proof. Let x_0 and y_0 be RP-compatible right and left eigenvectors for the eigenvalue λ_0 of A. Then $\varrho(\varepsilon) \to 1/(y_0^*x_0)$ as $\varepsilon \to 0$ and $\sigma_{n-1}(A - \lambda(\varepsilon)I) \to \sigma_{n-1}(A - \lambda_0 I)$ as $\varepsilon \to 0$. Both these limits are positive. The result follows immediately. ∎

Note, however, that if λ_0 is a very sensitive eigenvalue of A, then $\varrho(0) = 1/(y_0^*x_0)$ is large, and hence ε may need to be quite small to ensure that $r < 1$.

It is of particular interest to see the ratio of the two smallest singular values, $\varepsilon/\sigma_{n-1}(A - \lambda I)$, as a key factor in the convergence rate r. This is reminiscent of well-known results for the power and inverse power methods for computing eigenvalues. Note that if we had used (5.10) instead of (5.11) in Theorem 5.5, we would have obtained the factor one in place of this ratio.

All of Theorems 5.3, 5.5, and 5.6 apply to any boundary fixed point, so the corresponding point λ could be any local maximizer or minimizer satisfying the first-order condition (2.1) (see Figure 4.1). However, we have never observed convergence of Algorithm PSA0 to a fixed point for which λ is a local minimizer, because even if δ_k should happen to be small, r is typically greater than one and convergence does not take place. For ε sufficiently small λ cannot be a local minimizer because of the convexity of the pseudospectral components (see Lemma 2.8).

Although the theorems of this section assume that the fixed point (x, y) is a boundary fixed point, they could easily be extended to include internal fixed points too, provided we make the assumption that the corresponding points λ satisfy the condition that the singular value satisfying $\sigma_k(A - \lambda I) = \varepsilon$ (with $k < n$) be simple. We then find, however, that the ratio $\varepsilon/\sigma_{n-1}(A - \lambda I) = \sigma_n(A - \lambda I)/\sigma_{n-1}(A - \lambda I)$ in (5.13), which is less than one, is replaced by $\varepsilon/\sigma_n(A - \lambda I) = \sigma_k(A - \lambda I)/\sigma_n(A - \lambda I)$, which is greater than one. This likely explains the observation that such fixed points are typically repulsive (see Figure 4.1).

We conclude this section by noting that even if ε is relatively large and hence $r > 1$, convergence to local maximizers is usually observed, but the rate of convergence typically deteriorates as ε is increased.

6. A variant that guarantees monotonicity. Although Algorithm PSA0 usually generates a monotonically increasing sequence $\{\text{Re } z_k\}$, this is not guaranteed, and there are pathological examples on which the algorithm cycles [Gur10]. For this reason we introduce a modified version of the algorithm that guarantees monotonicity independent of ε.

Let x_k, y_k be an RP-compatible pair of right and left eigenvectors corresponding to the rightmost eigenvalue z_k of $B_k = A + \varepsilon y_{k-1} x_{k-1}^*$. Consider the one-parameter family of matrices

$$B(t) = A + \varepsilon y(t) x(t)^*,$$

with

$$x(t) = \frac{t x_k + (1-t) x_{k-1}}{\|t x_k + (1-t) x_{k-1}\|}, \quad y(t) = \frac{t y_k + (1-t) y_{k-1}}{\|t y_k + (1-t) y_{k-1}\|}.$$

Thus, $B(t) - A$ has rank 1 and spectral norm equal to ε for all t and $B(0) = A + \varepsilon x_{k-1} y_{k-1}^*$, $B(1) = A + \varepsilon y_k x_k^*$.

Let $n_x(t) = \|t x_k + (1-t) x_{k-1}\|$ and $n_y(t) = \|t y_k + (1-t) y_{k-1}\|$. Using $'$ to denote differentiation with respect to t, we have

$$B'(t) = \varepsilon \left[\frac{(y_k - y_{k-1}) - n_y'(t) y(t)}{n_y(t)} x(t)^* + y(t) \frac{(x_k - x_{k-1})^* - n_x'(t) x(t)^*}{n_x(t)} \right].$$

Exploiting

$$n_x'(t) = \text{Re}\left(x(t)^* (x_k - x_{k-1})\right), \quad n_y'(t) = \text{Re}\left(y(t)^* (y_k - y_{k-1})\right)$$

and $x(0) = x_{k-1}, y(0) = y_{k-1}, n_x(0) = n_y(0) = 1$ we obtain

$$n_x'(0) = \text{Re}(x_k^* x_{k-1}) - 1, \quad n_y'(0) = \text{Re}(y_k^* y_{k-1}) - 1$$

and

$$B'(0) = \varepsilon \left[\Big(y_k - \text{Re}\left(y_k^* y_{k-1}\right) y_{k-1} \Big) x_{k-1}^* + y_{k-1} \Big(x_k - \text{Re}\left(x_k^* x_{k-1}\right) x_{k-1} \Big)^* \right].$$

Now let $\lambda(t)$ denote the eigenvalue of $B(t)$ that converges to z_k at $t \to 0$. Using Lemma 2.1, we have

$$\lambda'(0) = \frac{y_k^* B'(0) x_k}{y_k^* x_k} = \frac{\varepsilon \psi_k}{y_k^* x_k}, \tag{6.1}$$

where

$$\psi_k = \Big(1 - y_k^* y_{k-1} \mathrm{Re}\,(y_k^* y_{k-1})\Big) x_{k-1}^* x_k + \Big(1 - x_{k-1}^* x_k \mathrm{Re}\,(x_k^* x_{k-1})\Big) y_k^* y_{k-1}. \tag{6.2}$$

As before, we choose x_k and y_k to be RP-compatible, so the denominator of (6.1) is real and positive. Furthermore, if Re $\psi_k < 0$, we change the sign of both x_k and y_k so that Re $\psi_k > 0$. Excluding the unlikely event that Re $\psi_k = 0$, defining (x_k, y_k) in this way guarantees that Re $\lambda(t) >$ Re $\lambda(0)$ for sufficiently small t, so the following algorithm is guaranteed to generate monotonically increasing $\{\mathrm{Re}\ z_k\}$.

ALGORITHM PSA1. Let z_0 be a rightmost eigenvalue of A, with corresponding right and left eigenvectors x_0 and y_0 normalized so that they are RP-compatible. Set $L_0 = y_0 x_0^*$ and $B_1 = A + \varepsilon L_0$.

For $k = 1, 2, \ldots$:

1. Let z_k be the rightmost eigenvalue of B_k (if there is more than one, choose the one closest to z_{k-1}). Let x_k and y_k be corresponding right and left eigenvectors normalized so that they are RP-compatible. Furthermore, set

   $$\psi_k = \Big(1 - y_k^* y_{k-1} \mathrm{Re}\,(y_k^* y_{k-1})\Big) x_{k-1}^* x_k + \Big(1 - x_{k-1}^* x_k \mathrm{Re}\,(x_k^* x_{k-1})\Big) y_k^* y_{k-1}.$$

   If Re $\psi_k < 0$, then replace x_k by $-x_k$ and y_k by $-y_k$. Set $t = 1$ and $z = z_k$, $x = x_k$, $y = y_k$.
2. Repeat the following zero or more times until Re $z >$ Re z_{k-1}: replace t by $t/2$, set

   $$x = \frac{t x_k + (1-t) x_{k-1}}{\| t x_k + (1-t) x_{k-1} \|}, \quad y = \frac{t y_k + (1-t) y_{k-1}}{\| t y_k + (1-t) y_{k-1} \|},$$

   and set z to the rightmost eigenvalue of $A + \varepsilon y x^*$.
3. Set $z_k = z$, $x_k = x$, $y_k = y$, $L_k = y_k x_k^*$, and $B_{k+1} = A + \varepsilon L_k$.

Note that if t is always 1, then Algorithm PSA1 generates the same iterates as Algorithm PSA0, and if we simply omit step 2, Algorithm PSA1 reduces to Algorithm PSA0.

Clearly, the monotonically increasing sequence $\{\mathrm{Re}\ z_k\}$ generated by Algorithm PSA1 must converge by compactness. This does not necessarily imply that z_k converges to an eigenvalue associated with a fixed point of Algorithm PSA0, but we have not encountered any counterexample, and indeed in all our experiments, we have always observed convergence of z_k to a local maximizer of (1.5).

7. The pseudospectral radius. An algorithm for the pseudospectral radius $\rho_\varepsilon(A)$ is obtained by a simple variant of Algorithm PSA1.

Let z_k be an eigenvalue of $B_k = A + \varepsilon y_{k-1} x_{k-1}^*$ with largest modulus, and let x_k, y_k be corresponding right and left eigenvectors with unit norm. Define $B(t)$ as in the previous section, and let $\lambda(t)$ denote the eigenvalue of $B(t)$ converging to z_k as $t \to 0$. We have

$$\left.\frac{d\left(|\lambda(t)|^2\right)}{dt}\right|_{t=0} = 2\,\mathrm{Re}\,(\overline{z}_k \lambda'(0)) \tag{7.1}$$

with $\lambda'(0)$ defined in (6.1), (6.2). This leads us to generalize the notion of RP-compatibility for the pseudospectral radius case as follows.

DEFINITION 7.1. *Let nonzero $z \in C$ be given. A pair of complex vectors x and y is called* RP(z)-compatible *if $\|x\| = \|y\| = 1$ and $y^* x$ is a real positive multiple of z.*

Thus, if x_k and y_k are RP($\overline{z}_k$)-compatible (note the conjugate!), then the denominator of (6.1) is a positive real multiple of $\overline{z}_k$, and hence by choosing the real part of the numerator to be positive as before, we can ensure that (7.1) is positive. This yields the following algorithm.

ALGORITHM PSR1. Let z_0 be an eigenvalue of A with largest modulus, with corresponding right and left eigenvectors x_0 and y_0 normalized so that they are RP($\overline{z}_0$)-compatible. Set $L_0 = y_0 x_0^*$ and $B_1 = A + \varepsilon L_0$.

For $k = 1, 2, \ldots$:

1. Let z_k be the eigenvalue of B_k with largest modulus (if there is more than one, choose the one closest to z_{k-1}). Let x_k and y_k be corresponding right and left eigenvectors normalized so that they are RP($\overline{z}_k$)-compatible. Furthermore, set

$$\psi_k = \Big(1 - y_k^* y_{k-1} \mathrm{Re}\,(y_k^* y_{k-1})\Big) x_{k-1}^* x_k + \Big(1 - x_{k-1}^* x_k \mathrm{Re}\,(x_k^* x_{k-1})\Big) y_k^* y_{k-1}.$$

   If Re $\psi_k < 0$, then replace x_k by $-x_k$ and y_k by $-y_k$. Set $t = 1$ and $z = z_k$, $x = x_k$, $y = y_k$.
2. Repeat the following zero or more times until $|z| > |z_{k-1}|$: replace t by $t/2$, set

$$x = \frac{t x_k + (1-t) x_{k-1}}{\| t x_k + (1-t) x_{k-1} \|}, \quad y = \frac{t y_k + (1-t) y_{k-1}}{\| t y_k + (1-t) y_{k-1} \|},$$

   and set z to the eigenvalue of $A + \varepsilon y x^*$ with largest modulus.
3. Set $z_k = z$, $x_k = x$, $y_k = y$, $L_k = y_k x_k^*$, and $B_{k+1} = A + \varepsilon L_k$.

Let us also define Algorithm PSR0, a variant of Algorithm PSA0 for the pseudospectral radius, as Algorithm PSR1 with step 2 omitted. We briefly summarize the theoretical results for the pseudospectral radius algorithms without giving proofs; they are obtained in complete analogy to the proofs given in sections 4 and 5.

The fixed points (x, y) of Algorithm PSR0 correspond to points λ for which ε is a singular value of $A - \lambda I$, and, if this is the smallest singular value, λ satisfies the first-order necessary condition for a maximizer of (1.6); namely, the corresponding right and left singular vectors v and $-u$ are RP($\overline{\lambda}$)-compatible. We conjecture that only fixed points corresponding to local maximizers are attractive. The rate of linear convergence of z_k to local maximizers is governed by the same ratio r as given in (5.13), except that $\varrho = 1/y^*x$ must be replaced by $\varrho = 1/|y^*x|$. For ε sufficiently small, the rate of convergence to local maximizers in a component of the pseudospectrum containing a simple maximum-modulus eigenvalue is arbitrarily fast. Algorithm PSR1 generates a monotonically increasing sequence $\{|z_k|\}$.

In fact, we can easily generalize Algorithms PSR0 and PSR1 to maximize *any* desired smooth convex function of $z \in \Lambda_\varepsilon(A)$, where we interpret "smooth" and "convex" by identifying $\mathbb{C}$ with $\mathbb{R}^2$, simply by generalizing the notion of RP-compatibility accordingly.

8. Examples: Dense matrices. A MATLAB code implementing Algorithms PSA1 and PSR1 is freely available.[1] In this section we present numerical results for the suite of dense matrix examples included in EigTool, some of which are very challenging with highly ill-conditioned eigenvalues.[2] We insert the following termination condition

[1] http://www.cs.nyu.edu/overton/software/psapsr/.

[2] The codes used to generate the results are also available at the same website.

TABLE 8.1

Results of Algorithm PSA1 on dense problems from EigTool for $\varepsilon = 10^{-4}$. The last four columns, respectively, show the error with respect to the criss-cross algorithm implemented in EigTool, the number of iterates, the maximum number of bisection steps needed, and r, the upper bound on the local rate of convergence.

Call	n	α	α_ε	Error	Iters	nb	r
`airy(100)`	99	-0.0782603	-0.0780263	$4.5e-014$	2	0	$1.9e-002$
`basor(100)`	100	6.10735	6.10748	$2.7e-015$	2	0	$1.6e-003$
`boeing('0')`	55	0.1015	0.232649	$4.2e-010$	7	0	$1.3e+006$
`boeing('S')`	55	-0.0787714	2.10578	$5.8e-009$	16	0	$3.6e+007$
`chebspec(100)`	99	220.368	336.188	$1.1e-005$	49	0	$9.2e+004$
`companion(10)`	10	3.37487	16.0431	$2.1e-008$	13	0	$4.2e+005$
`convdiff(100)`	99	-7.58225	-4.77608	$1.6e-010$	4	0	$9.5e+002$
`davies(100)`	99	58507.3	58507.3	$7.3e-011$	2	0	$2.7e-005$
`demmel(10)`	10	-1	-0.451107	$5.5e-007$	506	0	$2.8e+002$
`frank(100)`	100	361.465	431.807	$2.0e-012$	3	0	$2.8e+004$
`gallery3(100)`	3	3	3.02208	$3.2e-012$	3	0	$3.5e+001$
`gallery5(100)`	5	0.0320395	1.3298	$2.4e-008$	15	0	$1.7e+004$
`gaussseidel({100, 'C'})`	100	0.999033	0.999133	$2.4e-013$	2	0	$1.3e-001$
`gaussseidel({100, 'D'})`	100	0.442161	0.696024	$3.9e-008$	54	0	$6.5e+001$
`gaussseidel({100, 'U'})`	100	0.437068	0.941755	$4.0e-007$	288	0	$5.2e+001$
`godunov(100)`	7	4.29582	136.594	$3.0e-006$	43	0	$6.1e+004$
`grcar(100)`	100	1.68447	2.41276	$6.6e-007$	262	0	$2.6e+002$
`hatano(100)`	100	3.06266	3.06292	$4.9e-015$	2	0	$8.9e-003$
`kahan(100)`	100	1	1.00879	$5.8e-012$	3	0	$1.9e+001$
`landau(100)`	100	0.998463	0.998564	$6.7e-016$	2	0	$9.2e-003$
`orrsommerfeld(100)`	99	$-7.81914e-005$	0.00391589	$1.1e-012$	4	0	$9.0e+001$
`random(100)`	100	0.99703	0.997427	$8.9e-015$	2	0	$9.9e-002$
`randomtri(100)`	100	0.229298	0.287963	$2.1e-010$	5	0	$1.9e+004$
`riffle(100)`	100	0.5	0.509954	$1.2e-011$	4	0	$1.3e+003$
`transient(100)`	100	-0.0972178	0.138158	$4.9e-012$	5	0	$7.4e+001$
`twisted(100)`	100	1.95582	1.95594	$4.7e-015$	2	0	$6.0e-003$

in step 1 of Algorithms PSA1 and PSR1: the iteration stops if $k > 1$ and

$$|f(z_k) - f(z_{k-1})| < \eta \max\left(1, |f(z_{k-1})|\right),$$

where f is the real part or modulus function, respectively, and we use $\eta = 10^{-8}$.

Since the matrices A and therefore B_k are dense, all eigenvalues and right eigenvectors of A and of B_k are computed by calling the standard eigenvalue routine `eig`. In principle, we could compute the left eigenvectors by inverting a matrix of right eigenvectors after one call to `eig`, or alternatively we could compute the relevant left eigenvector by one step of inverse iteration applied to A^T, but because the relevant eigenvalues and eigenvectors are often ill conditioned, we instead compute the left eigenvectors explicitly by making a second call to `eig` to compute the right eigenvectors of A^T. Regardless of which way a left eigenvector is computed, it must be subsequently conjugated to be compatible with the standard definition in Lemma 2.1.

Table 8.1 shows results for the pseudospectral abscissa when $\varepsilon = 10^{-4}$. The first column shows the call to the EigTool code generating the matrix A (omitting the trailing string `_demo` in the name of the function), and the second shows its dimension. The `boeing`, `gallery3`, `gallery5`, and `godunov` examples have fixed dimension regardless of the input. The largest entry of the $n \times n$ `companion` matrix is of order $n!$, so we use $n = 10$ in this case. We set $n = 10$ for the `demmel` example too, because this matrix has just one eigenvalue with multiplicity n, as discussed further below. The next two columns of the table show the computed spectral abscissa $\alpha(A)$ and pseudospectral abscissa $\alpha_\varepsilon(A)$, respectively. The column headed "Error" shows

TABLE 8.2
Results of Algorithm PSA1 on dense problems from EigTool for $\varepsilon = 10^{-2}$.

Call	n	α	α_ε	Error	Iters	nb	r
`airy(100)`	99	-0.0782603	-0.0577769	$2.9e-010$	8	0	$1.1e+000$
`basor(100)`	100	6.10735	6.11958	$6.4e-012$	3	0	$1.6e-001$
`boeing('D')`	55	0.1015	53.9798	$2.7e-004$	69	2	$4.0e+008$
`boeing('S')`	55	-0.0787714	54.1494	$7.2e-006$	45	2	$3.0e+008$
`chebspec(100)`	99	220.368	474.537	$8.2e-006$	40	0	$2.3e+003$
`companion(10)`	10	3.37487	229.283	$4.4e-007$	14	0	$2.4e+004$
`convdiff(100)`	99	-7.58225	-2.91953	$2.0e-012$	4	0	$2.7e+001$
`davies(100)`	99	58507.3	58507.3	$0.0e+000$	2	0	$2.7e-003$
`demmel(10)`	10	-1	4.38931	$8.9e-011$	8	0	$1.7e+003$
`frank(100)`	100	361.465	531.948	$1.1e-012$	4	0	$8.7e+002$
`gallery3(100)`	3	3	4.79265	$2.4e-012$	4	0	$3.1e+002$
`gallery5(100)`	5	0.0320395	29.6715	$6.0e-008$	14	0	$3.6e+003$
`gaussseidel({100, 'C'})`	100	0.999033	1.00903	$1.5e-008$	13	0	$3.1e+000$
`gaussseidel({100, 'D'})`	100	0.442161	0.842735	$2.6e-008$	43	0	$3.8e+000$
`gaussseidel({100, 'U'})`	100	0.437068	0.994281	$3.0e-007$	224	0	$3.3e+000$
`godunov(100)`	7	4.29582	282.767	$4.9e-006$	41	0	$2.0e+003$
`grcar(100)`	100	1.68447	2.73991	$6.1e-007$	217	0	$7.6e+000$
`hatano(100)`	100	2.95843	2.97154	$2.2e-010$	4	0	$1.3e+000$
`kahan(100)`	100	1	1.05746	$1.7e-010$	6	0	$3.2e+000$
`landau(100)`	100	0.998463	1.00851	$2.1e-011$	3	0	$8.7e-001$
`orrsommerfeld(100)`	99	$-7.81914e-005$	0.134554	$1.8e-008$	34	0	$6.8e+001$
`random(100)`	100	0.985354	1.00215	$1.2e-011$	4	0	$1.0e+000$
`randomtri(100)`	100	0.254198	0.484744	$5.6e-009$	25	1	$1.1e+002$
`riffle(100)`	100	0.5	0.668439	$4.9e-009$	21	0	$1.3e+002$
`transient(100)`	100	-0.0972178	0.233235	$3.3e-011$	6	0	$3.6e+000$
`twisted(100)`	100	1.95582	1.96761	$6.7e-013$	4	0	$5.3e-001$

the absolute difference between the value of $\alpha_\varepsilon(A)$ computed by the new algorithm and that obtained using the criss-cross algorithm [BLO03] which is implemented as part of EigTool. The small values shown in this column demonstrate that Algorithm PSA1 converges to a global maximizer of (1.5) for every one of these examples, although we know that it is possible to construct simple examples for which the algorithm converges only to a local maximizer.

The column headed "Iters" shows the number of iterates generated by the new algorithm, that is, the number of times that step 1 of Algorithm PSA1 is executed. The column headed "nb" shows the maximum, over all iterates, of the number of bisections of t in step 2. When this is zero, as is it is for all the examples in Table 8.1, Algorithm PSA1 reduces to Algorithm PSA0. The final column, headed r, shows the upper bound on the local convergence rate defined in (5.13). For some examples this is less than one, but for others it is enormous, indicating that the left and right eigenvector are close to orthogonal at the maximizer and therefore correspond to an ill-conditioned final eigenvalue z. Nonetheless, as already noted, the iteration converges to a global maximizer even in these cases. Thus, it is clear that r is far from a tight upper bound on the convergence rate.

Table 8.2 shows the same results for the pseudospectral abscissa when $\varepsilon = 10^{-2}$. Note that some bisections of t take place for two of these examples.

Tables 8.3 and 8.4 show similar results for the pseudospectral radius. In this case, the column headed "Error" shows the absolute difference between the result returned by Algorithm PSR1 and that returned by the radial-circular search algorithm of [MO05], also implemented as a part of EigTool. There is a significant discrepancy for `kahan(100)` with $\varepsilon = 10^{-2}$. This is an interesting example, because the spectrum is positive real and Algorithm PSR1 finds a local maximizer on the positive real axis,

TABLE 8.3
*Results of Algorithm PSR*1 *on dense problems from EigTool for* $\varepsilon = 10^{-4}$.

Call	n	ρ	ρ_ε	Error	Iters	nb	r
airy(100)	99	1421.53	1421.53	$6.8e-012$	2	0	$2.3e-004$
basor(100)	100	6.12272	6.12284	$1.2e-014$	2	0	$1.6e-003$
boeing('O')	55	1000	1001.7	$3.5e-002$	2	0	$1.3e+009$
boeing('S')	55	1000.25	1002.01	$1.4e-007$	3	0	$2.8e+007$
chebspec(100)	99	869.338	869.339	$6.6e-012$	2	0	$8.0e-006$
companion(10)	10	6.56063	27.1478	$3.5e-006$	57	0	$2.2e+005$
convdiff(100)	99	158598	158598	$5.8e-011$	2	0	$3.3e-007$
davies(100)	99	58507.3	58507.3	$2.2e-011$	2	0	$2.7e-005$
demmel(10)	10	1	4.14044	$9.1e-008$	47	0	$6.7e+003$
frank(100)	100	361.465	431.807	$1.3e-008$	3	0	$2.8e+004$
gallery3(100)	3	3	3.02208	$2.9e-013$	3	0	$3.5e+001$
gallery5(100)	5	0.0405204	4.65026	$3.0e-007$	64	0	$2.1e+004$
gaussseidel({100, 'C'})	100	0.999033	0.999133	$2.4e-013$	2	0	$1.3e-001$
gaussseidel({100, 'D'})	100	0.442161	0.696024	$6.2e-008$	75	0	$6.5e+001$
gaussseidel({100, 'U'})	100	0.437068	0.941754	$2.0e-006$	770	0	$5.2e+001$
godunov(100)	7	4.29582	136.593	$8.5e-005$	515	0	$6.1e+004$
grcar(100)	100	2.26293	2.85216	$9.2e-007$	238	0	$1.7e+002$
hatano(100)	100	2.88253	2.88333	$3.6e-014$	2	0	$7.0e-001$
kahan(100)	100	1	1.00879	$2.9e-014$	3	0	$1.9e+001$
landau(100)	100	0.998573	0.998674	$2.2e-016$	2	0	$9.2e-003$
orrsommerfeld(100)	99	2959.97	2959.97	$1.4e-011$	2	0	$2.6e-002$
random(100)	100	1.04721	1.04748	$1.0e-014$	2	0	$2.2e-002$
randomtri(100)	100	0.28979	0.314459	$1.9e-010$	6	0	$1.4e+004$
riffle(100)	100	0.5	0.509954	$1.2e-011$	4	0	$1.3e+003$
transient(100)	100	0.903776	1.13816	$5.3e-007$	331	0	$7.4e+001$
twisted(100)	100	2.76594	2.76606	$3.4e-014$	2	0	$5.9e-003$

TABLE 8.4
*Results of Algorithm PSR*1 *on dense problems from EigTool for* $\varepsilon = 10^{-2}$.

Call	n	ρ	ρ_ε	Error	Iters	nb	r
airy(100)	99	1421.53	1421.54	$4.1e-012$	2	0	$2.3e-002$
basor(100)	100	6.12272	6.13495	$7.2e-012$	3	0	$1.6e-001$
boeing('O')	55	1000	1146.9	$2.6e+000$	1000	0	$7.4e+008$
boeing('S')	55	1000.25	1150.15	$6.1e-006$	18	0	$4.8e+008$
chebspec(100)	99	869.338	869.35	$4.2e-012$	2	0	$8.0e-004$
companion(10)	10	6.56063	238.597	$1.3e-004$	107	0	$2.3e+004$
convdiff(100)	99	158598	158598	$2.0e-010$	2	0	$3.3e-005$
davies(100)	99	58507.3	58507.4	$1.0e-010$	2	0	$2.7e-003$
demmel(10)	10	1	14.9909	$5.5e-007$	72	0	$8.4e+002$
frank(100)	100	361.465	531.948	$2.2e-010$	4	0	$8.7e+002$
gallery3(100)	3	3	4.79265	$3.8e-012$	4	0	$3.1e+002$
gallery5(100)	5	0.0405204	33.693	$6.2e-006$	156	0	$2.9e+003$
gaussseidel({100, 'C'})	100	0.999033	1.00903	$1.5e-008$	13	0	$3.1e+000$
gaussseidel({100, 'D'})	100	0.442161	0.842735	$4.7e-008$	65	0	$3.8e+000$
gaussseidel({100, 'U'})	100	0.437068	0.994279	$1.6e-006$	649	0	$3.3e+000$
godunov(100)	7	4.29582	282.767	$1.2e-004$	395	0	$2.0e+003$
grcar(100)	100	2.26293	3.07351	$1.1e-006$	185	0	$5.9e+000$
hatano(100)	100	2.88007	2.89905	$8.1e-012$	4	0	$4.1e+000$
kahan(100)	100	1	1.05746	$8.1e-002$	6	0	$3.2e+000$
landau(100)	100	0.998573	1.00862	$2.1e-011$	3	0	$8.8e-001$
orrsommerfeld(100)	99	2959.97	2960.03	$1.1e-011$	2	0	$2.6e+000$
random(100)	100	1.07362	1.09612	$1.8e-011$	5	0	$1.7e+000$
randomtri(100)	100	0.262564	0.443161	$4.3e-002$	15	0	$1.0e+002$
riffle(100)	100	0.5	0.668439	$4.9e-009$	21	0	$1.3e+002$
transient(100)	100	0.903776	1.23323	$4.1e-007$	248	0	$3.6e+000$
twisted(100)	100	2.76594	2.77768	$4.0e-011$	3	0	$5.0e-001$

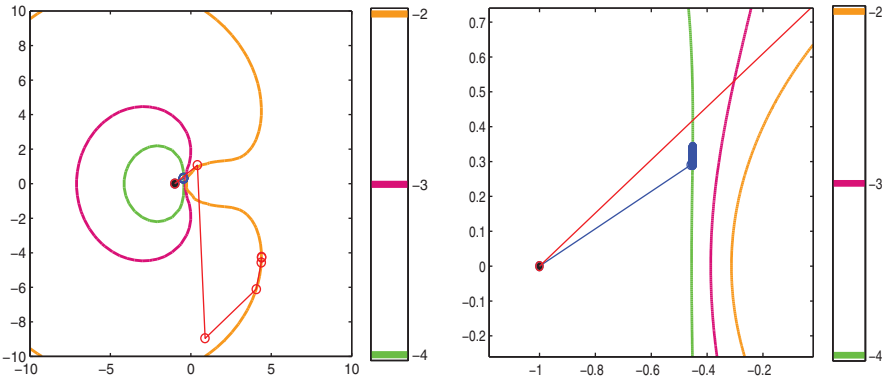

FIG. 8.1. *Iterates of Algorithm PSA*1 *on the "demmel" example with* $n = 10$.

but the global maximizer of (1.6) is actually on the negative real axis. However, Algorithm PSA1 applied to $-A$ finds the correct value $\alpha_\varepsilon(-A) = \rho_\varepsilon(A) = 1.13797$. A similar observation applies to some instances of `randomtri(100)`, for which the spectrum is also real, though not positive. The large error for `boeing('O')` is due to very slow convergence arising from a double eigenvalue of A at -1000, violating the simple maximum-modulus eigenvalue assumption.

Figure 8.1 shows iterates of Algorithm PSA1 for computing the pseudospectral abscissa of the `demmel` matrix with $n = 10$. Again, the eigenvalue simplicity assumption is violated: in fact, A has just one eigenvalue with algebraic multiplicity n and geometric multiplicity one and therefore with left and right eigenvectors satisfying $y_0^* x_0 = 0$. Nonetheless, in the left panel we see that the iterates for $\varepsilon = 10^{-2}$ converge rapidly to a maximizer of (1.5). A close-up view is shown in the right panel, where we see that the convergence for $\varepsilon = 10^{-4}$ is much slower. Note that Theorem 5.7 does not apply to this example because $y_0^* x_0 = 0$, and hence $r(\varepsilon)$ does not converge to 0 as $\varepsilon \to 0$.

Figure 8.2 shows the iterates of Algorithm PSR1 for computing the pseudospectral radius of the `grcar` matrix for $n = 100$. The left panel shows the characteristic shape of the spectrum and pseudospectra; note the sensitivity to perturbation as evidenced by the distance from the spectrum to the pseudospectral boundary for $\varepsilon = 10^{-4}$. The iterates for $\varepsilon = 10^{-4}$ and $\varepsilon = 10^{-2}$ are shown, with the right panel giving a close-up view.

Figure 8.3 shows the iterates of Algorithm PSA1 for computing the pseudospectral abscissa of the `orrsommerfeld` matrix for $n = 100$. In the left panel one can see the well-known Y-shaped spectrum extending almost to the imaginary axis, with the iterates for both $\varepsilon = 10^{-4}$ and $\varepsilon = 10^{-2}$ extending into the right half-plane, while the right panel again offers a close-up view. The matrix A is fully dense, arising from a spectral discretization of the Orr–Sommerfeld operator, which arises in fluid dynamics of laminar flows.

The Orr–Sommerfeld matrix has the following interesting property. Let $z \in \mathbb{C}$ be arbitrary, and let $\widetilde{U}\widetilde{\Sigma}\widetilde{V}^*$ be the SVD of $A - zI$. Then the left and right singular vectors satisfy

$$\widetilde{u}_\ell^* \widetilde{v}_m = 0 \qquad \text{if } |\ell - m| \text{ is odd.}$$

This property allows us to improve the upper bound r for the local convergence rate, as we now show.

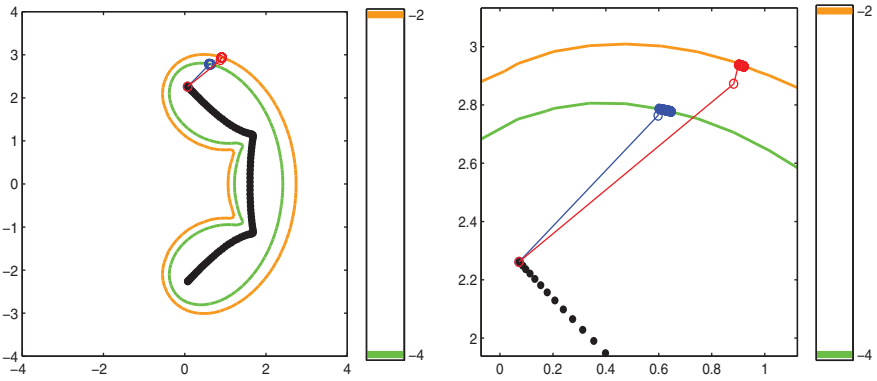

FIG. 8.2. *Iterates of Algorithm PSR*1 *on the "grcar" example with* $n = 100$.

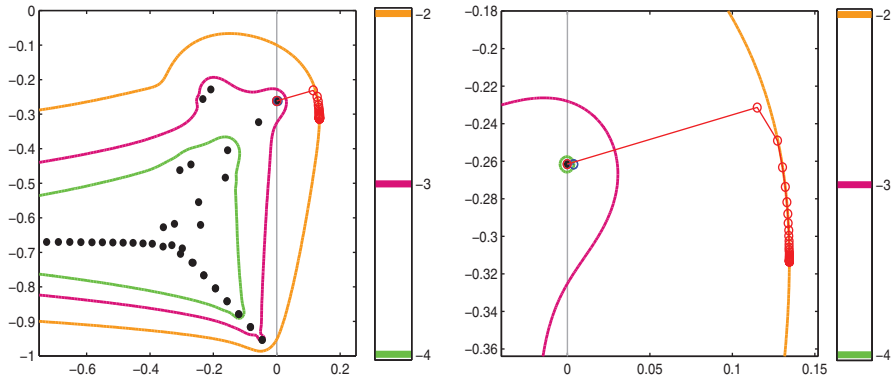

FIG. 8.3. *Iterates of Algorithm PSA*1 *on the "orrsommerfeld" example with* $n = 100$.

Let λ be a local maximizer of (1.5) associated with a fixed point (x, y), and let USV^* be the SVD of $A - \lambda I$. Then, for the group inverse $G = (A + \varepsilon yx^* - \lambda I)^{\#}$ we have the following improvements to (5.10) and (5.12):

$$G = s_{n-1}^{-1} v_{n-1} u_{n-1}^* + (I - wy^*) \sum_{i=1}^{n-2} s_i^{-1} v_i u_i^* (I - wy^*),$$

where we have used the property

$$y^* v_{n-1} = -u_n^* v_{n-1} = 0,$$

and hence

$$\|G\| \le \frac{1}{\sigma_{n-1}(A - \lambda I)} + \frac{\varrho^2}{\sigma_{n-2}(A - \lambda I)}, \qquad \text{where } \varrho = \frac{1}{y^* x}.$$

As a consequence, the upper bound for the convergence rate r may be replaced by

$$4\varepsilon \left(\frac{\varrho^2}{\sigma_{n-2}(A - \lambda I)} + \frac{1}{\sigma_{n-1}(A - \lambda I)} \right). \tag{8.1}$$

Since ϱ^2 turns out to be much larger than $\frac{\sigma_{n-2}(A-\lambda I)}{\sigma_{n-1}(A-\lambda I)}$, the bound (8.1) is a significant improvement over (5.13).

TABLE 9.1
*Results of Algorithm PSA*1 *on sparse problems from EigTool for* $\varepsilon = 10^{-4}$. *NaN means eigs failed.*

Call	n	α	α_ε	Iters	nb
`dwave(2048)`	2048	0.978802	0.978902	2	0
`convdiff_fd(10)`	400	NaN	NaN	NaN	NaN
`markov(100)`	5050	1	1.00024	3	0
`olmstead(500)`	500	4.51018	4.51029	2	0
`pde(2961)`	2961	9.90714	9.90769	2	0
`rdbrusselator(3200)`	3200	0.106623	0.106871	2	0
`sparserandom(10000)`	10000	2.99998	3.00027	2	0
`skewlap3d(30)`	24389	−749.08	−518.171	4	0
`supg(20)`	400	0.0541251	0.0838952	9	0
`tolosa(4000)`	4000	NaN	NaN	NaN	NaN

TABLE 9.2
*Results of Algorithm PSA*1 *on sparse problems from EigTool for* $\varepsilon = 10^{-2}$.

Call	n	α	α_ε	Iters	nb
`dwave(2048)`	2048	0.978802	0.988803	3	0
`convdiff_fd(10)`	400	NaN	NaN	NaN	NaN
`markov(100)`	5050	1	1.01634	38	0
`olmstead(500)`	500	4.51018	4.52058	2	0
`pde(2961)`	2961	9.90714	9.95362	7	0
`rdbrusselator(3200)`	3200	0.106623	0.131476	3	0
`sparserandom(10000)`	10000	2.99999	3.02588	3	0
`skewlap3d(30)`	24389	−749.081	−404.348	4	0
`supg(20)`	400	0.0544674	0.102749	55	0
`tolosa(4000)`	4000	NaN	NaN	NaN	NaN

9. Examples: Sparse matrices. Our MATLAB implementation supports three kinds of matrix input: dense matrices, as in the previous section; sparse matrices, to be illustrated in this section; and function handles, which specify the name of a MATLAB file implementing matrix-vector products. In the last two cases, we use the MATLAB routine `eigs`, which is an interface for ARPACK, a well-known code implementing the implicitly restarted Arnoldi method [LS96, LSY98]. Since `eigs` does not require B_k explicitly, but needs only the ability to do matrix-vector products with B_k, it also accepts as input either a sparse matrix or a function handle. The latter functionality is critical for us, because it is essential to avoid computing the dense matrix $B_k = A + \varepsilon y_k x_k^*$ explicitly. In contrast, writing an efficient function to compute matrix-vector products with B_k is straightforward, and it is a handle for this function that we pass to `eigs`. We instruct `eigs` to compute the largest eigenvalue with respect to real part or modulus, respectively, by passing the string "LR" (largest real) or "LM" (largest magnitude) as an option. In the former case particularly, the implicit restarting is an essential aspect of the method which allows it to deliver the correct eigenvalue and eigenvector.

As in the dense case, we compute the corresponding left eigenvector of B_k by a second call to `eigs`, to find the right eigenvector of the transposed matrix B_k^T. Thus, when the input to our implementation is a function handle, it must implement matrix-vector products with A^T as well as with A. Except for $k = 0$, when computing the eigenvector z_k (y_k) of B_k, we provide the previous right (left) eigenvector z_{k-1} (y_{k-1}) to `eigs` as an initial guess. The appearance of NaN in Tables 9.1–9.4 means that `eigs` failed to compute the desired eigenvalue to the default required accuracy; when this occurred it was always during the initialization step ($k = 0$) and happened more frequently when computing the spectral abscissa than the spectral radius.

TABLE 9.3
*Results of Algorithm PSR*1 *on sparse problems from EigTool for* $\varepsilon = 10^{-4}$.

Call	n	ρ	ρ_ε	Iters	nb
`dwave(2048)`	2048	0.978802	0.978902	2	0
`convdiff_fd(10)`	400	NaN	NaN	NaN	NaN
`markov(100)`	5050	1	1.00024	4	0
`olmstead(500)`	500	2544.02	2544.02	2	0
`pde(2961)`	2961	9.91937	9.91992	2	0
`rdbrusselator(3200)`	3200	111.073	111.074	2	0
`sparserandom(10000)`	10000	2.99999	3.00027	2	0
`skewlap3d(30)`	24389	10050.9	10281.8	4	0
`supg(20)`	400	NaN	NaN	NaN	NaN
`tolosa(4000)`	4000	4842	4842.25	2	0

TABLE 9.4
*Results of Algorithm PSR*1 *on sparse problems from EigTool for* $\varepsilon = 10^{-2}$.

call	n	ρ	ρ_ε	Iters	nb
`dwave(2048)`	2048	0.978802	0.988803	3	0
`convdiff_fd(10)`	400	NaN	NaN	NaN	NaN
`markov(100)`	5050	1	1.01634	38	0
`olmstead(500)`	500	2544.02	2544.11	2	0
`pde(2961)`	2961	9.91937	9.96546	7	0
`rdbrusselator(3200)`	3200	111.073	111.084	2	0
`sparserandom(10000)`	10000	3.00008	3.02605	3	0
`skewlap3d(30)`	24389	10051	10395.7	4	0
`supg(20)`	400	NaN	NaN	NaN	NaN
`tolosa(4000)`	4000	4842	4867.31	2	0

As in the previous section, we used a relative stopping tolerance $\eta = 10^{-8}$. Although we are not able to check global optimality as in the previous section, it seems likely that globally optimal values were again computed in most cases. Under this assumption, we see from the tables that for most of the problems for which `eigs` is able to compute the spectral abscissa or spectral radius to the default accuracy, Algorithms PSA1 and PSR1 approximate the pseudospectral abscissa or pseudospectral radius to about 8 digits of relative precision with only about 10 times as much work as the computation of the spectral abscissa or spectral radius alone.

10. Conclusions and future work. We have presented, analyzed, and implemented efficient algorithms for computing the pseudospectral abscissa α_ε and pseudospectral radius ρ_ε of a large sparse matrix. We believe that the potential impact on practical applications is significant. We also expect further algorithmic development to follow. In particular, together with M. Gürbüzbalaban we are developing related algorithms for computing the distance to instability (or complex stability radius; see section 1) and an important generalization, the H_∞ norm of the transfer function for a linear dynamical system with input and output, for which the current standard method [BB90] is based on computing eigenvalues of Hamiltonian matrices like (1.7). Finally, the new algorithms may provide a useful approach to solving *structured* problems. A well-known challenging problem is computation of the *real* stability radius, that is, the distance to instability when perturbations to a real matrix are required to be real [QBR+95]. Even if A is real, Algorithms PSA1 and PSR1 do not usually generate real matrices B_k because x_k and y_k are typically complex, but real perturbations may be enforced by projection, replacing the rank one complex matrix $L_k = y_k x_k^*$ by its real part, which has rank two.

Acknowledgments. The first author thanks the faculty and staff of the Courant Institute for their hospitality during several visits to New York, and both authors thank Daniel Kressner for arranging a stimulating visit to ETH Zürich, where some of this work was conducted. Finally, the authors thank the referees for carefully reading the paper.

REFERENCES

[ABBO11] R. Alam, S. Bora, R. Byers, and M. L. Overton, *Characterization and construction of the nearest defective matrix via coalescence of pseudospectral components*, Linear Algebra Appl., 435 (2011), pp. 494–513.

[BB90] S. Boyd and V. Balakrishnan, *A regularity result for the singular values of a transfer matrix and a quadratically convergent algorithm for computing its* $\mathbf{L}_\infty$*-norm*, Systems Control Lett., 15 (1990), pp. 1–7.

[BLO03] J. V. Burke, A. S. Lewis, and M. L. Overton, *Robust stability and a criss-cross algorithm for pseudospectra*, IMA J. Numer. Anal., 23 (2003), pp. 359–375.

[BLO07] J. V. Burke, A. S. Lewis, and M. L. Overton, *Convexity and Lipschitz behavior of small pseudospectra*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 586–595.

[Bye88] R. Byers, *A bisection method for measuring the distance of a stable matrix to the unstable matrices*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 875–881.

[HJ90] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 1990. Corrected reprint of the 1985 original.

[Gur10] M. Gürbüzbalaban, *private communication*, 2010.

[HP86] D. Hinrichsen and A. J. Pritchard, *Stability radii of linear systems*, Systems Control Lett., 7 (1986), pp. 1–10.

[Kat95] T. Kato, *Perturbation Theory for Linear Operators*, Classics Math., Springer-Verlag, Berlin, 1995. Reprint of the 1980 edition.

[Kre62] H.-O. Kreiss, *Über die Stabilitätsdefinition für Differenzengleichungen die partielle Differentialgleichungen approximieren*, Nordisk Tidskr. Informations-Behandling, 2 (1962), pp. 153–181.

[LP08] A. S. Lewis and C. H. J. Pang, *Variational analysis of pseudospectra*, SIAM J. Optim., 19 (2008), pp. 1048–1072.

[LS96] R. B. Lehoucq and D. C. Sorensen, *Deflation techniques for an implicitly restarted Arnoldi iteration*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 789–821.

[LSY98] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*, Software, Environ. Tools 6, SIAM, Philadelphia, 1998.

[MO05] E. Mengi and M. L. Overton, *Algorithms for the computation of the pseudospectral radius and the numerical radius of a matrix*, IMA J. Numer. Anal., 25 (2005), pp. 648–669.

[MS88] C. D. Meyer and G. W. Stewart, *Derivatives and perturbations of eigenvectors*, SIAM J. Numer. Anal., 25 (1988), pp. 679–691.

[QBR+95] L. Qiu, B. Bernhardsson, A. Rantzer, E. J. Davison, P. M. Young, and J. C. Doyle, *A formula for computation of the real stability radius*, Automatica J. IFAC, 31 (1995), pp. 879–890.

[Ran07] T. Ransford, *On pseudospectra and power growth*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 699–711.

[Ste01] G. W. Stewart, *Matrix Algorithms: Volume* II: *Eigensystems*, SIAM, Philadelphia, 2001.

[TE05] L. N. Trefethen and M. Embree, *Spectra and Pseudospectra: The Behavior of Nonnormal Matrices and Operators*, Princeton University Press, Princeton, NJ, 2005.

[VL85] C. Van Loan, *How near is a stable matrix to an unstable matrix?*, in Linear Algebra and Its Role in Systems Theory (Brunswick, Maine, 1984), Contemp. Math. 47, AMS, Providence, RI, 1985, pp. 465–478.

[Wil86] J. H. Wilkinson, *Sensitivity of eigenvalues* II, Util. Math., 30 (1986), pp. 243–286.

[Wri02] T. G. Wright, *EigTool: A graphical tool for nonsymmetric eigenproblems*, Oxford University Computing Laboratory, Oxford, UK; available online from http://www.comlab.ox.ac.uk/pseudospectra/eigtool/ (2002).

[WT01] T. G. Wright and L. N. Trefethen, *Large-scale computation of pseudospectra using ARPACK and eigs*, SIAM J. Sci. Comput., 23 (2001), pp. 591–605.