SIAM J. MATRIX ANAL. APPL.
Vol. 45, No. 3, pp. 1455–1486

VARIATIONAL CHARACTERIZATION AND RAYLEIGH QUOTIENT ITERATION OF 2D EIGENVALUE PROBLEM WITH APPLICATIONS*

TIANYI LU†, YANGFENG SU‡, AND ZHAOJUN BAI§

Abstract. A two dimensional eigenvalue problem (2DEVP) of a Hermitian matrix pair (A, C) is introduced in this paper. The 2DEVP can be regarded as a linear algebra formulation of the well-known eigenvalue optimization problem of the parameter matrix $A - \mu C$. We first present fundamental properties of the 2DEVP, such as the existence and variational characterizations of 2D-eigenvalues, and then devise a Rayleigh quotient iteration (RQI)-like algorithm, 2DRQI in short, for computing a 2D-eigentriplet of the 2DEVP. The efficacy of the 2DRQI is demonstrated by large scale eigenvalue optimization problems arising from the minmax of Rayleigh quotients and the distance to instability of a stable matrix.

Key words. eigenvalue problem, eigenvalue optimization, variational characterization, Rayleigh quotient iteration

MSC codes. 65K10, 65F15

DOI. 10.1137/22M1472589

1. Introduction.

We consider the problem of finding scalars $\mu, \lambda \in \mathbb{R}$ and a nonzero vector $x \in \mathbb{C}^n$ to satisfy the nonlinear equations

$$(A - \mu C)x = \lambda x, \tag{1.1a}$$

$$x^H C x = 0, \tag{1.1b}$$

$$x^H x = 1, \tag{1.1c}$$

where $A, C \in \mathbb{C}^{n\times n}$ are given Hermitian matrices and C is indefinite. The pair (μ, λ) is called a *2D-eigenvalue,* x the corresponding *2D-eigenvector,* and (μ, λ, x) a *2D-eigentriplet.* We use the term "2*D*" based on the fact that an eigenvalue has two components, which is a point on the two-dimensional (μ, λ)-plane. The nonlinear equations (1.1) are called a 2*D eigenvalue problem* (2DEVP) of the matrix pair (A, C).

Our interest in studying the 2DEVP (1.1) primarily stems from eigenvalue optimization. If we regard μ as a parameter in the 2DEVP (1.1), then the equation (1.1a) is a parameter eigenvalue problem of the matrix $H(\mu) = A - \mu C$. Since A and C are Hermitian, $H(\mu)$ has n real eigenvalues $\lambda_1(\mu), \lambda_2(\mu), \ldots, \lambda_n(\mu)$ for any $\mu \in \mathbb{R}$. Suppose that these eigenvalues are sorted such that $\lambda_1(\mu) \geq \lambda_2(\mu) \geq \cdots \geq \lambda_n(\mu)$. When one wants to optimize an eigenvalue $\lambda_j(\mu)$ with respect to μ:

$$\inf_{\mu\in\mathbb{R}} \lambda_j(\mu), \tag{1.2}$$

*Received by the editors January 20, 2022; accepted for publication (in revised form) by M. A. Freitag March 20, 2024; published electronically August 6, 2024.
https://doi.org/10.1137/22M1472589

†Shanghai Center for Mathematical Sciences; School of Mathematical Sciences, Fudan University, Shanghai 200433, China (tylu17@fudan.edu.cn).

‡School of Mathematical Sciences, Fudan University, Shanghai 200433, China (yfsu@fudan.edu.cn).

§Department of Computer Science and Department of Mathematics, University of California, Davis, CA 95616 USA (zbai@ucdavis.edu).

equation (1.1b) is actually a stationary condition for (local or global) maxima or minima of $\lambda_j(\mu)$ (see section 3). This fact has been observed by Overton and Womersley [37] when $\lambda_j(\mu_*)$ is a simple eigenvalue of $H(\mu_*)$ at a stationary point μ_* of $\lambda_j(\mu)$. In general, when $\lambda_j(\mu_*)$ is a multiple eigenvalue of $H(\mu_*)$, to the best of our knowledge, the connection to the 2DEVP (1.1) as presented in this paper is new.

Different equivalent conditions of eigenvalue optimizations have been discovered in the literature, such as conditions based on the generalized gradient [42] and existence of a special positive semidefinite matrix in the context of minimizing the largest eigenvalue of a multivariable Hermitian matrix [10]. Eigenvalue optimizations in the presence of nonconvexity and multiplicity are particularly challenging [10, 18, 24, 28, 33, 35, 37, 42, 48].

Blum and Chang [1] considered the following so-called two-parameter or double eigenvalue problem arising from solving a boundary value problem of ordinary differential equations with double parameters:

$$\begin{cases} Ax = \lambda C_1 x + \mu C_2 x, \\ f(x) = 0, \\ \|x\| = 1, \end{cases} \tag{1.3}$$

where $A, C_1, C_2 \in \mathbb{R}^{n\times n}$, $\|\cdot\|$ denotes the 2-norm, and f is a real-valued function. In (1.3), $\lambda, \mu \in \mathbb{R}$ and $x \in \mathbb{R}^n$ are the eigenvalues and eigenvectors to be found. Khazanov [21] generalized the problem (1.3) to more than two parameters and derived a related eigenvalue problem. Obviously, when A and C in (1.1) are real, the 2DEVP (1.1) is a special case of (1.3). Due to the general form of the function f in (1.3), there is a lack of essential theoretical analysis of the problem (1.3) such as the existence of the solution. In addition, the restriction to the real vectors and matrices severely limits applications of the problem (1.3) such as calculating the distance to instability (see section 6). Although algorithms are proposed for solving the problem (1.3) in [1, 21], there are no convergence analyses of the proposed algorithms and no backward error analyses for a computed solution. Maybe due to these concerns, the two-parameter or double eigenvalue problems of the form (1.3) have received little attention over the years.

In this paper, we will first present theoretical results of the 2DEVP, such as the relationship between the 2DEVP and eigenvalue optimization, and variational characterization of 2D-eigenvalues. We then devise a Rayleigh quotient iteration (RQI)-like (2DRQI) algorithm for computing a 2D-eigentriplet. One of main features of the 2DRQI is that the computational kernel is a linear system of equations, similar to the classical RQI for solving a Hermitian eigenvalue problem [40, sec. 4.6]. Therefore, the 2DRQI is capable of solving large scale 2DEVP by exploiting the structure and sparsity of matrices A and C. As a part of the main contributions of this paper, the 2DEVP and the 2DRQI algorithm are exploited in depth for applications in two eigenvalue optimization problems, namely finding the minmax of two Rayleigh quotients and computing the distance to instability (DTI) of a stable matrix. We will demonstrate the theoretical and algorithmic advantages of treating these eigenvalue optimizations through the 2DEVP and 2DRQI, such as introducing the notion of the backward error of a computed DTI for the first time, and the substantial reduction in computing time compared with existing algorithms for many large scale DTI problems. A rigorous convergence analysis of the 2DRQI to show that the 2DRQI locally quadratically converges is presented in [30].

The rest of this paper is organized as follows. In section 2, we study the related parameter eigenvalue problem of the 2DEVP and introduce the notion of sorted and

analyticalized eigencurves. In section 3, we investigate the existence and variational characterizations of 2D-eigenvalues. In section 4, we introduce 2D Rayleigh quotient (2DRQ) and Jacobian of the 2DEVP and derive the 2DRQI for computing a 2D-eigentriplet. The backward error analysis of the 2DEVP is in section 5. In section 6, we discuss the applications of the 2DEVP and the 2DRQI for two eigenvalue optimization problems. In section 7, we present numerical examples to illustrate the convergence behaviors of the 2DRQI and demonstrate its efficiency in applications. Concluding remarks are in section 8. In the spirit of reproducible research, MATLAB scripts of the implementations of algorithms and data that are used to generate numerical results presented in this paper are available at https://github.com/AdrainT/2DEVP.

2. The associated parameter eigenvalue problem. If we discard (1.1b), the remaining two equations of the 2DEVP (1.1) are a parameter eigenvalue problem of the matrix $H(\mu) = A - \mu C$ with real parameter μ. For $\mu \in \mathbb{R}$, there exist n real eigenvalues $\lambda_j(\mu)$ and corresponding orthonormal eigenvectors $x_j(\mu)$ of $H(\mu)$. If $\lambda_j(\mu)$ are sorted such that $\lambda_1(\mu) \geq \cdots \geq \lambda_n(\mu)$, then we have n *sorted eigencurves* $\lambda_j(\mu)$ of $H(\mu)$ for $j = 1, 2, \ldots, n$. The sorted eigencurves $\lambda_j(\mu)$ are continuous and may be nondifferentiable at the intersections; see Figure 1a. The following theorem is a direct result of [14, Thm. S6.3] and shows that with proper reordering, the eigencurves $\lambda_j(\mu)$ can be analyticalized.

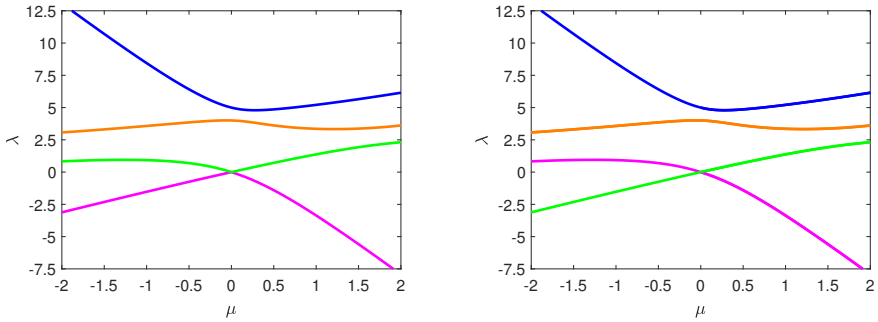
THEOREM 2.1 (see [14]). *For Hermitian matrices A and C and $\mu \in \mathbb{R}$, there exist scalar functions $\lambda_1(\mu)$, $\ldots$, $\lambda_n(\mu)$ and matrix-valued functions $X(\mu) = \big[x_1(\mu), \ldots, x_n(\mu)\big]$ such that*

$$\begin{aligned} A - \mu C &= X(\mu)\,\mathrm{diag}\,\big[\lambda_1(\mu), \quad \ldots, \quad \lambda_n(\mu)\big]\, X^H(\mu), \\ X^H(\mu)X(\mu) &= I. \end{aligned} \tag{2.1}$$

Furthermore, $\lambda_j(\mu)$ and $x_j(\mu)$ are analytic for $\mu \in \mathbb{R}$.

The analytic eigencurves $\lambda_j(\mu)$ for $\mu \in \mathbb{R}$ in Theorem 2.1 will be called *analyticalized eigencurves* of $H(\mu)$. Analyticalized eigencurves may be different from sorted eigencurves as illustrated in Figure 1b. In the rest of the paper, we will use $\lambda_j(\mu)$ and $\widetilde{\lambda}_j(\mu)$ to denote a sorted and an analyticalized eigencurve of $H(\mu)$, respectively.

A benefit of introducing analyticalized eigencurves is that we can have the notion of one-sided derivatives of the sorted eigencurves at any point $\mu \in \mathbb{R}$, even the one corresponding to the intersection of the sorted eigencurves. To that end, let us first



(a) Sorted eigencurves $\lambda_j(\mu)$ (b) Analyticalized eigencurves $\widetilde{\lambda}_j(\mu)$

FIG. 1. *Illustration of eigencurves.*

present the following theorem derived from [45, p. 45, Thm. 1] to show that the derivatives of analyticalized eigencurves can be calculated through solving an eigenvalue problem.

THEOREM 2.2 (see [45]). *Let $\widetilde{\lambda}_1(\mu),\ldots,\widetilde{\lambda}_n(\mu)$ be the analyticalized eigencurves of $H(\mu)$. Assume λ_0 is an eigenvalue of $H(\mu_0)$ with algebraic multiplicity k, i.e., $\widetilde{\lambda}_j(\mu_0)=\lambda_0$ for $p\le j\le p+k-1$ with some integer $p\ge 1$. Let X_k be an orthonormal basis of the eigensubspace corresponding to λ_0. Then by counting multiplicities, $\widetilde{\lambda}_j'(\mu_0)$ have one-to-one correspondence with the eigenvalues of $-X_k^HCX_k$ for $p\le j\le p+k-1$.*

By Theorem 2.2, we can introduce one-sided derivatives of sorted eigencurves.

THEOREM 2.3. *Assume (μ_*,λ_*) is an intersection of k sorted eigencurves, i.e., $\lambda_j(\mu_*)=\lambda_*$ for $p\le j\le p+k-1$ for some integer $p\ge 1$, and $\lambda_j(\mu_*)\ne\lambda_*$ for $j<p$ or $j\ge p+k$. Let X_k be an orthonormal basis for the eigensubspace of the eigenvalue λ_* of $A-\mu_*C$. Then for $p\le j\le p+k-1$, the one-sided derivatives*

$$\lambda_j^{'(-)}(\mu_*)\equiv\lim_{t\to 0^-}\frac{\lambda_j(\mu_*+t)-\lambda_j(\mu_*)}{t}\quad\text{and}\quad\lambda_j^{'(+)}(\mu_*)\equiv\lim_{t\to 0^+}\frac{\lambda_j(\mu_*+t)-\lambda_j(\mu_*)}{t}$$

exist. Furthermore, both multisets $\{\lambda_j^{'(-)}(\mu_*)\,|\,p\le j\le p+k-1\}$ and $\{\lambda_j^{'(+)}(\mu_*)\mid p\le j\le p+k-1\}$ have one-to-one correspondence with the multiset of eigenvalues of $-C_k\equiv -X_k^HCX_k$; i.e., if the eigenvalues of $-C_k$ are $\tau_1\ge\tau_2\ge\cdots\ge\tau_k$, then $\lambda_{p+k-j}^{'(-)}(\mu_*)=\tau_j=\lambda_{p-1+j}^{'(+)}(\mu_*)$ for $j=1,\ldots,k$.

Proof. We first prove by contradiction that there exists $r>0$, such that in the interval (μ_*,μ_*+r), for any i,j, there can be only one of the following two cases between any two analyticalized eigencurves $\widetilde{\lambda}_i(\mu)$ and $\widetilde{\lambda}_j(\mu)$ of $A-\mu C$:

$$\widetilde{\lambda}_i(\mu)=\widetilde{\lambda}_j(\mu)\quad\text{or}\quad\widetilde{\lambda}_i(\mu)\ne\widetilde{\lambda}_j(\mu)$$

for any $\mu\in(\mu_*,\mu_*+r)$. If r does not exist, then we can find a fixed pair (i,j) and a sequence $\{\mu_m\}_{m=1}^{\infty}$ such that $\widetilde{\lambda}_i(\mu_m)=\widetilde{\lambda}_j(\mu_m)$, $\mu_m\to\mu_*$, $\mu_m\ne\mu_*$, but $\widetilde{\lambda}_i(\mu)\not\equiv\widetilde{\lambda}_j(\mu)$, which contradicts the identity property of analytic functions [22, p. 87].

We next prove that in the interval $[\mu_*,\mu_*+r)$, each sorted eigencurve identically equals to an analyticalized eigencurve. In fact, we have proved that in the interval (μ_*,μ_*+r), two analyticalized eigencurves that are not equal identically will not intersect. Then by the continuity of analyticalized eigencurves, for any i,j, there are exactly the following three cases that hold for all $\mu\in(\mu_*,\mu_*+r)$:

$$\widetilde{\lambda}_i(\mu)<\widetilde{\lambda}_j(\mu)\quad\text{or}\quad\widetilde{\lambda}_i(\mu)=\widetilde{\lambda}_j(\mu)\quad\text{or}\quad\widetilde{\lambda}_i(\mu)>\widetilde{\lambda}_j(\mu).$$

This implies that in the interval (μ_*,μ_*+r), the algebraic order of the analyticalized eigencurves is preserved. Thus we can find a permutation $\{\ell_1,\ell_2,\ldots,\ell_n\}$ of $\{1,2,\ldots,n\}$, such that $\widetilde{\lambda}_{\ell_j}(\mu)=\lambda_j(\mu)$ for $\mu\in(\mu_*,\mu_*+r)$ and $j=1,\ldots,n$. By continuity, $\widetilde{\lambda}_{\ell_j}(\mu_*)=\lambda_j(\mu_*)$ for $j=1,\ldots,n$. Consequently, for $p\le j\le p+k-1$, the limit

$$\lim_{t\to 0^+}\frac{\lambda_j(\mu_*+t)-\lambda_j(\mu_*)}{t}$$

exists and equals to $\widetilde{\lambda}_{\ell_j}'(\mu_*)$. By Theorem 2.2, the multiset $\{\lambda_j^{'(+)}(\mu_*)\;\big|\;p\le j\le p+k-1\}$ has one-to-one correspondence with the multiset of eigenvalues of $-C_k$. By

a similar argument, we can show that the limit $\lambda_j^{'(-)}(\mu_*)$ exists and has one-to-one correspondence with the eigenvalues of $-C_k$, counting multiplicities.

Furthermore, note that for $t>0$ and $p\le j\le p+k-2$,

$$\frac{\lambda_j(\mu_*+t)-\lambda_j(\mu_*)}{t}\ge\frac{\lambda_{j+1}(\mu_*+t)-\lambda_{j+1}(\mu_*)}{t}$$

and

$$\frac{\lambda_j(\mu_*-t)-\lambda_j(\mu_*)}{-t}\le\frac{\lambda_{j+1}(\mu_*-t)-\lambda_{j+1}(\mu_*)}{-t}.$$

Thus for $p\le j\le p+k-2$,

$$\lambda_j^{'(+)}(\mu_*)\ge\lambda_{j+1}^{'(+)}(\mu_*)\quad\text{and}\quad\lambda_j^{'(-)}(\mu_*)\le\lambda_{j+1}^{'(-)}(\mu_*). \tag{2.2}$$

Then the equality $\lambda_{p-1+j}^{'(+)}(\mu_*)=\tau_j=\lambda_{p+k-j}^{'(-)}(\mu_*)$ follows from (2.2) and the correspondence between $\{\lambda_j^{'(+)}(\mu_*)\mid p\le j\le p+k-1\}$, $\{\lambda_j^{'(-)}(\mu_*)\mid p\le j\le p+k-1\}$, and $\{\tau_j\mid 1\le j\le k\}$. ∎

We end this section with the following corollary of Theorem 2.2. Its proof can also be drawn from the proof of Theorem 2.3 when $k=1$.

COROLLARY 2.4. *If $\lambda_p(\mu)$ is a simple eigenvalue of $A-\mu C$, then $\lambda_p(\cdot)$ is differentiable at μ and $\lambda_p'(\mu)=-x_p(\mu)^HCx_p(\mu)$, where $x_p(\mu)$ is a corresponding unit eigenvector of $\lambda_p(\mu)$.*

3. Existence and variational characterization of 2D-eigenvalues. In this section, we discuss the existence of 2D-eigenvalues and their variational characterizations to reveal intrinsic connections between the 2DEVP and eigenvalue optimization.

THEOREM 3.1. *If (μ_*,λ_*) is a local minimum or maximum of a sorted eigencurve $\lambda(\mu)$ of $A-\mu C$, then (μ_*,λ_*) must be a 2D-eigenvalue of (A,C).*

Proof. We prove for the case when (μ_*,λ_*) is a local maximum of some sorted eigencurve. The proof for the case when (μ_*,λ_*) is a local minimum is similar. Assume (μ_*,λ_*) is an intersection of k sorted eigencurves $\lambda_j(\mu)$ of $A-\mu C$ for $p\le j\le p+k-1$ with some integer $p\ge 1$. Then $\lambda_j(\mu_*)=\lambda_*$. Let X_k be an orthonormal basis for the eigensubspace of the eigenvalue λ_* of $A-\mu_*C$, and $C_k=X_k^HCX_k$. Then by Theorem 2.3 and Corollary 2.4, both multisets $\{\lambda_j^{'(-)}(\mu_*)\mid p\le j\le p+k-1\}$ and $\{\lambda_j^{'(+)}(\mu_*)\mid p\le j\le p+k-1\}$ have one-to-one correspondence with the multiset of eigenvalues of $-C_k$.

Since (μ_*,λ_*) is a local maximum, we have $\lambda_{p+k-1}^{'(+)}(\mu_*)\le 0$ and $\lambda_{p+k-1}^{'(-)}(\mu_*)\ge 0$. By the one-to-one correspondence, C_k has both nonnegative and nonpositive eigenvalues. This implies that C_k is not definite ($C_k=0$ when $k=1$). Let z be a unit vector that satisfies $z^HC_kz=0$. Then (μ_*,λ_*,X_kz) is a 2D-eigentriplet. This completes the proof. ∎

Remark 3.2. The proof of Theorem 3.1 is algebraic. An alternative proof is to use Clarke's generalized directional derivative and generalized gradient in nonsmooth optimization [6, p. 10]. Specifically, if (μ_*,λ_*) is a stationary point (locally minimum or maximum) of some sorted eigencurve $\lambda_j(\mu)$, then we have the first-order optimality condition $0\in\partial\lambda_j(\mu_*)$, where $\partial\lambda_j(\mu_*)$ is Clarke's generalized derivative $\partial\lambda_j(\mu)$ at μ_* of the eigencurve $\lambda_j(\mu)$ [6, p. 38, Prop. 2.3.2]. Based on Clarke's generalized derivatives

of spectral functions [27, p. 585] and the chain rule [6, p. 42, Thm. 2.3.9], we can derive that

(3.1)
$$\partial\lambda_j(\mu_*) \subseteq \{-x^H C x \,|\, x \text{ is a unit eigenvector corresponding to } \lambda_* \text{ of } A-\mu_* C.\}$$

Consequently, by the first-order optimal condition and (3.1), we conclude that if (μ_*, λ_*) is a stationary point of some sorted eigencurve $\lambda_j(\mu)$, then there exists a unit eigenvector x_* corresponding to the eigenvalue λ_* of $A-\mu_* C$, such that $0 = x_*^H C x_*$. Such a (μ_*, λ_*, x_*) is a 2D-eigentriplet of the 2DEVP (1.1).

Theorem 3.1 shows that if (μ_*, λ_*) is a local minimum or maximum of some sorted eigencurve, then (μ_*, λ_*) must be a 2D-eigenvalue. Conversely, a 2D-eigenvalue (μ, λ) does not necessarily correspond to a local minimum or maximum of a sorted eigencurve as shown in Example 1.

Example 1. Let

$$A = \begin{bmatrix} 2 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \quad \text{and} \quad C = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}.$$

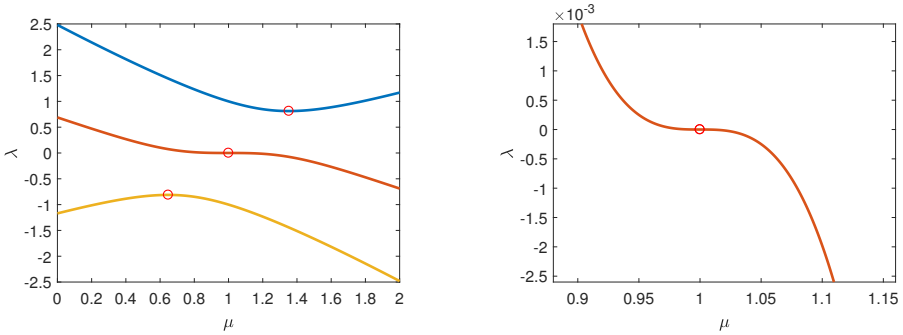
Three sorted eigencurves $\lambda_1(\mu) \geq \lambda_2(\mu) \geq \lambda_3(\mu)$ of $A-\mu C$ are depicted in blue, red, and yellow, respectively, in Figure 2a. $(\mu, \lambda, x) = (1, 0, e_3)$ is a 2D-eigentriplet. The 2D-eigenvalue $(\mu, \lambda) = (1, 0)$ is on the eigencurve $\lambda_2(\mu)$. However, it is neither a local minimum nor a local maximum of $\lambda_2(\mu)$ as shown in the close up plot in Figure 2b.

By Theorem 3.1, we immediately have the following theorem on the existence of 2D-eigenvalues.

THEOREM 3.3. *The* 2*DEVP* (1.1) *has at least one* 2*D-eigenvalue.*

Proof. We prove by construction. Let $\lambda_1(\mu)$ be the largest sorted eigencurve of $A-\mu C$. Then as $\mu \to -\infty$,

$$\lambda_1(\mu) = \max_{\|x\|=1} x^H(A-\mu C)x \geq \lambda_{\min}(A) - \mu \max_{\|x\|=1} x^H C x = \lambda_{\min}(A) - \mu\lambda_{\max}(C) \to +\infty,$$



(a) Sorted eigencurves $\lambda_1(\mu) \geq \lambda_2(\mu) \geq \lambda_3(\mu)$

(b) close up plot of eigencurve $\lambda_2(\mu)$

FIG. 2. *A* 2*D-eigenvalue can be neither minima nor maxima. (Color available online.)*

where we use the fact that $\lambda_{\max}(C) > 0$ since C is indefinite. On the other hand, as $\mu \to +\infty$,

$$\lambda_1(\mu) \geq \lambda_{\min}(A) - \mu \min_{\|x\|=1} x^H C x = \lambda_{\min}(A) - \mu\lambda_{\min}(C) \to +\infty,$$

where we use the fact that $\lambda_{\min}(C) < 0 < \lambda_{\max}(C)$ since C is indefinite. Therefore, the minimum of $\lambda_1(\mu)$ is attainable at some point μ_* by the continuity of $\lambda_1(\mu)$. By Theorem 3.1, $(\mu_*, \lambda_1(\mu_*))$ is a 2D-eigenvalue of (A, C). ∎

The following theorem reveals variational characterizations of extreme eigencurves $\lambda_1(\mu)$ and $\lambda_n(\mu)$.

THEOREM 3.4. *Let $\lambda_1(\mu) \geq \cdots \geq \lambda_n(\mu)$ be n sorted eigenvalues of $A - \mu C$. Then it holds that*

$$\min_{\mu\in\mathbb{R}} \lambda_1(\mu) = \max_{\substack{x\neq 0 \\ x^H C x = 0}} \rho_A(x) \quad \text{and} \quad \max_{\mu\in\mathbb{R}} \lambda_n(\mu) = \min_{\substack{x\neq 0 \\ x^H C x = 0}} \rho_A(x), \tag{3.2}$$

where $\rho_A(x)$ is the Rayleigh quotient of A, $\rho_A(x) = x^H A x/(x^H x)$.

Proof. We only prove the first identity in (3.2). The proof for the second identity is similar. We note that the proof of Theorem 3.3 indicates that the minimum of $\lambda_1(\mu)$ is attainable at some point μ_* and $(\mu_*, \lambda_*) = (\mu_*, \lambda_1(\mu_*))$ is a 2D-eigenvalue. Let x_* be the corresponding 2D-eigenvector of (μ_*, λ_*); then

$$x_*^H C x_* = 0 \quad \text{and} \quad \lambda_* = \rho_A(x_*) \leq \max_{\substack{x\neq 0 \\ x^H C x = 0}} \rho_A(x).$$

On the other hand,

$$\begin{aligned} \lambda_* = \lambda_1(\mu_*) = \max_{x^H x = 1} x^H (A - \mu_* C) x &\geq \max_{\substack{x^H x = 1 \\ x^H C x = 0}} x^H (A - \mu_* C) x \\ &= \max_{\substack{x^H x = 1 \\ x^H C x = 0}} x^H A x. \end{aligned}$$

This completes the proof. ∎

As a corollary of Theorem 3.4, the following result provides lower and upper bounds of the λ component of 2D-eigenvalues (μ, λ) on the (μ, λ)-plane.

COROLLARY 3.5. *Let (μ_*, λ_*) be a 2D-eigenvalue of (A, C) and $\lambda_1(\mu) \geq \cdots \geq \lambda_n(\mu)$ be n sorted eigencurves of $A - \mu C$. Then*

$$\max_{\mu\in\mathbb{R}} \lambda_n(\mu) \leq \lambda_* \leq \min_{\mu\in\mathbb{R}} \lambda_1(\mu), \tag{3.3}$$

where the first equality holds if $\lambda_n(\mu_*) = \lambda_*$, and the second equality holds if $\lambda_1(\mu_*) = \lambda_*$.

Proof. Let x_* be a 2D-eigenvector associated with (μ_*, λ_*). Then the inequalities in (3.3) hold by Theorem 3.4 and the identity $\lambda_* = \rho_A(x_*)$. If $\lambda_n(\mu_*) = \lambda_*$, we further have

$$\max_{\mu\in\mathbb{R}} \lambda_n(\mu) \leq \lambda_* = \lambda_n(\mu_*) \leq \max_{\mu\in\mathbb{R}} \lambda_n(\mu).$$

Thus the first inequality in (3.3) turns to equality. Similar arguments show the second equality holds if $\lambda_1(\mu_*) = \lambda_*$. ∎

Based on Corollary 3.5, we have the following definitions of extreme 2D-eigenvalues.

DEFINITION 3.6. *Let* $\lambda_1(\mu^*) = \min_{\mu\in\mathbb{R}} \lambda_1(\mu)$ *and* $\lambda_n(\mu_*) = \max_{\mu\in\mathbb{R}} \lambda_n(\mu)$. *Then* $(\mu^*, \lambda_1(\mu^*))$ *and* $(\mu_*, \lambda_n(\mu_*))$ *are called the* maximum *and* minimum 2D-eigenvalues *of* (A, C), *respectively.*

The following theorem provides an upper bound of $|\mu|$ of 2D-eigenvalues (μ, λ) on the (μ, λ)-plane when C is nonsingular.

THEOREM 3.7. *Assume the indefinite matrix* C *is nonsingular and* $\lambda_C^{(+)}$ *and* $\lambda_C^{(-)}$ *are the minimum positive and maximum negative eigenvalues of* C, *respectively. If* (μ_*, λ_*, x_*) *is a 2D-eigentriplet of* (A, C), *then*

$$|\mu_*| \le \|A\| / \sqrt{-\lambda_C^{(-)} \lambda_C^{(+)}}.$$

Proof. By multiplying $x_*^H C$ on the left of (1.1a), we have

$$|\mu_*| = \frac{|x_*^H C A x_*|}{\|Cx_*\|^2} \le \frac{\|A\|\|x_*\|}{\|Cx_*\|} = \frac{\|A\|}{\|Cx_*\|}.$$

Then an upper bound of $\frac{\|A\|}{\|Cx_*\|}$ can be obtained from a lower bound of $\|Cx_*\|$, which leads to computing the quantity

$$\min_{\substack{x^H x = 1 \\ x^H C x = 0}} \|Cx\|. \tag{3.4}$$

By substituting C^2 for A in the second equation of (3.2), we have

$$\begin{aligned} \min_{\substack{x^H x = 1 \\ x^H C x = 0}} \|Cx\|^2 &= \min_{\substack{x^H C x = 0 \\ x \neq 0}} \frac{x^H C^2 x}{x^H x} = \max_{\mu\in\mathbb{R}} \lambda_n(C^2 - \mu C) \\ &= \max_{\mu\in\mathbb{R}} \min\{c_i^2 - \mu c_i |\ i = 1, \cdots, n\}, \end{aligned} \tag{3.5}$$

where $c_1, c_2, \ldots, c_n$ are eigenvalues of C. Let $\lambda_C^{(-)} = c_j$ and $\lambda_C^{(+)} = c_k$ for some j and k. Then at the intersection $\widetilde{\mu}_* = c_j + c_k$ of lines $c_j^2 - \mu c_j$ and $c_k^2 - \mu c_k$, we have

$$\begin{aligned} \max_{\mu\in\mathbb{R}} \min\{c_i^2 - \mu c_i |\ i = 1, \ldots, n\} &\le \max_{\mu\in\mathbb{R}} \min\{c_i^2 - \mu c_i |\ i = j, k\} \\ &= \min\{c_i^2 - \widetilde{\mu}_* c_i |\ i = j, k\} = -\lambda_C^{(-)} \lambda_C^{(+)}. \end{aligned}$$

On the other hand, we can prove $-\lambda_C^{(-)} \lambda_C^{(+)} \le c_i^2 - \tilde{\mu}_* c_i$ for $i = 1, \ldots, n$. Without loss of generality, we only consider the case $c_i > 0$. Then

$$\begin{aligned} -\lambda_C^{(-)} \lambda_C^{(+)} - (c_i^2 - \widetilde{\mu}_* c_i) &= c_k^2 - \widetilde{\mu}_* c_k - c_i^2 + \tilde{\mu}_* c_i = (c_k - c_i)(c_k + c_i - \widetilde{\mu}_*) \\ &= (c_k - c_i)(c_i - c_j) \le 0, \end{aligned} \tag{3.6}$$

where the first equation is due to the fact $-\lambda_C^{(-)} \lambda_C^{(+)} = c_j^2 - \widetilde{\mu}_* c_j = c_k^2 - \widetilde{\mu}_* c_k$ and the last inequality results from the fact that either $c_j \le c_k \le c_i$ or $c_i \le c_j \le c_k$ holds. Hence we have

$$\max_{\mu\in\mathbb{R}} \min\{c_i^2 - \mu c_i |\ i = 1, \ldots, n\} \ge \min\{c_i^2 - \tilde{\mu}_* c_i |\ i = 1, \ldots, n\} = -\lambda_C^{(-)} \lambda_C^{(+)}.$$

This implies

$$\min_{\substack{x^H x = 1 \\ x^H C x = 0}} \|Cx\|^2 = \max_{\mu\in\mathbb{R}} \min\{c_i^2 - \mu c_i |\ i = 1, \ldots, n\} = -\lambda_C^{(-)} \lambda_C^{(+)}.$$

This completes the proof. ∎

We end this section with a well-known result on the convexity of the extreme eigencurves $\lambda_1(\mu)$ and $\lambda_n(\mu)$ of $H(\mu) = A - \mu C$, which will be used for applications such as finding the minmax of Rayleigh quotients in subsection 6.1.

THEOREM 3.8 (see [9, 27, 36]). *$\lambda_1(\mu)$ is convex and $\lambda_n(\mu)$ is concave.*

4. 2D Rayleigh quotient iteration. The Rayleigh quotient iteration (RQI) is an efficient single-vector iterative algorithm for finding an eigenpair of a Hermitian matrix; see, for example, [40, sec. 4.6], [50]. In this section, we derive an RQI-like method to solve the 2DEVP (1.1).

4.1. 2D Rayleigh quotient. Let us first introduce the concepts of Rayleigh quotients and Ritz values for the 2DEVP (1.1) and then reveal their approximation properties to 2D-eigentriplets.

DEFINITION 4.1. *Given an $n\times n$ Hermitian matrix pair (A, C) and an $n\times p$ matrix V with orthonormal columns, the $p \times p$ matrix pair (V^HAV, V^HCV) is called a* 2D Rayleigh quotient (2DRQ). *If V^HCV is indefinite and (ν, θ, z) is a 2D-eigentriplet of the 2DRQ (V^HAV, V^HCV), i.e.,*

$$\Big((V^HAV) - \nu(V^HCV)\Big)z = \theta z, \tag{4.1a}$$

$$z^H(V^HCV)z = 0, \tag{4.1b}$$

$$z^Hz = 1, \tag{4.1c}$$

then (ν, θ) is called a 2D-Ritz value*, Vz a* 2D-Ritz vector*, and (ν, θ, Vz) a* 2D-Ritz triplet*.*

The pair (V^HAV, V^HCV) is called a 2DRQ for two reasons. First it is analogous to the definition of the Rayleigh quotient for a matrix and a matrix with orthonormal columns [40, p. 288]. Second, it is to be shown in section 4.3 that when $C = 0$, the kth iterate V_k in Algorithm 4.1 degenerates to a vector parallel to $(A-\lambda_kI)^{-1}x$, and $V_k^HAV_k$ is the standard Rayleigh quotient [40, p. 75].

The 2DEVP (1.1) can be formulated as the problem of finding the root of the following system of nonlinear equations:

$$F(\mu, \lambda, x) \equiv \begin{bmatrix} Ax - \mu Cx - \lambda x \\ -x^HCx/2 \\ -(x^Hx - 1)/2 \end{bmatrix} = 0.$$

When μ, λ, and x are real, the Jacobian of the function F is well defined; see, e.g., [20, p. 65]. When x is complex, the second and third elements of F are not differentiable due to the violation of the Cauchy–Riemann conditions [22]. In this case we have the following natural extension of the Jacobian of the nonlinear function F.

DEFINITION 4.2. *The* Jacobian *of $F(\mu, \lambda, x)$ (and the 2DEVP) is defined as*

$$J(\mu, \lambda, x) = \begin{bmatrix} A - \mu C - \lambda I & -Cx & -x \\ -x^HC & 0 & 0 \\ -x^H & 0 & 0 \end{bmatrix}. \tag{4.2}$$

We note that the Jacobian $J(\mu, \lambda, x)$ has been introduced in [29] for deriving a Newton-type method to overcome the difficulties caused by the nondifferentiability of F. In [30], we have proved that for a 2D eigentriplet (μ_*, λ_*, x_*), $J(\mu_*, \lambda_*, x_*)$ is nonsingular if and only if one of the following two cases occurs:

I: The algebraic multiplicity of the eigenvalue λ_* of $A-\mu_* C$ is one (simple) and the corresponding eigencurve $\lambda(\mu)$ satisfies $\lambda''(\mu_*) \neq 0$.

II: The algebraic multiplicity of λ_* for $A-\mu_* C$ is two and the corresponding two analyticalized eigencurves $\widetilde{\lambda}_1(\mu)$ and $\widetilde{\lambda}_2(\mu)$ satisfy $\widetilde{\lambda}_1'(\mu_*)\widetilde{\lambda}_2'(\mu_*) < 0$.

Cases I and II are generic cases, namely the cases that typically arise in most applications. The Newton-type method [29] is only applicable to case I, while the algorithm derived in subsection 4.2 is applicable to both cases.

4.2. Algorithm derivation. The gist of an iterative algorithm for finding a 2D-eigentriplet (μ_*, λ_*, x_*) is how to use the kth approximation (μ_k, λ_k, x_k) of (μ_*, λ_*, x_*) to obtain a projection subspace V_k containing a vector closer to the 2D-eigenvector x_* and then define the $(k+1)$st approximation $(\mu_{k+1}, \lambda_{k+1}, x_{k+1})$ using a 2D-Ritz triplet.

To that end, assume the Jacobian $J(\mu_k, \lambda_k, x_k)$ defined in (4.2) is nonsingular. Write

$$\mu_* = \mu_k + \Delta\mu_k, \quad \lambda_* = \lambda_k + \Delta\lambda_k, \quad x_* = x_k + \Delta x_k,$$

where $|\Delta\mu_k| \leq \epsilon$, $|\Delta\lambda_k| \leq \epsilon$, and $\|\Delta x_k\| \leq \epsilon$ for some small $\epsilon > 0$. Then by (1.1a), we have

$$\widehat{J}_k \begin{bmatrix} x_* \\ \Delta\mu_k \\ \Delta\lambda_k \end{bmatrix} = O(\epsilon^2), \tag{4.3}$$

where $\widehat{J}_k = \begin{bmatrix} A - \mu_k C - \lambda_k I & -Cx_k & -x_k \end{bmatrix}$. This implies that up to the second-order approximation of ϵ, the vector $\left[\begin{smallmatrix} x_* \\ \Delta\mu_k \\ \Delta\lambda_k \end{smallmatrix}\right]$ lies in the null subspace of $\widehat{J}_k$. Since the Jacobian $J(\mu_k, \lambda_k, x_k)$ is assumed to be nonsingular, $\widehat{J}_k$ is of full rank and the dimension of the null subspace of $\widehat{J}_k$ is 2. Let $\left[\begin{smallmatrix} \widetilde{V}_k \\ R \end{smallmatrix}\right]$ be a basis matrix of the null subspace of $\widehat{J}_k$, where $\widetilde{V}_k \in \mathbb{C}^{n\times 2}$, $R \in \mathbb{C}^{2\times 2}$. Then by (4.3), up to the second-order approximation of ϵ, x_* lies approximately in $\operatorname{span}\{\widetilde{V}_k\}$. Therefore a natural idea is to use the 2D-Ritz triplet based on the Rayleigh quotient induced by $\operatorname{span}\{\widetilde{V}_k\}$ to define the next iterate $(\mu_{k+1}, \lambda_{k+1}, x_{k+1})$.

To compute $\widetilde{V}_k$, one can apply the traditional methods for computing the null space of $\widehat{J}_k$, such as the rank revealing QR decomposition [7, p. 107]. However, for exploiting the underlying structure and sparsity of (A, C), we consider the following augmented linear equation of (4.3):

$$J(\mu_k, \lambda_k, x_k) \begin{bmatrix} X_a \\ u \\ v \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{4.4}$$

By the first block row of (4.4), $\operatorname{span}\{X_a\} \subseteq \operatorname{span}\{\widetilde{V}_k\}$. Meanwhile, by the second and third block rows of (4.4), $\dim(\operatorname{span}\{X_a\}) = 2$. Since $\dim(\operatorname{span}\{\widetilde{V}_k\}) \leq 2$, we have

$$\operatorname{span}\{X_a\} = \operatorname{span}\{\widetilde{V}_k\}. \tag{4.5}$$

Once X_a is computed, an orthonormal basis of $\operatorname{span}\{\widetilde{V}_k\}$ is given by

$$V_k = \operatorname{orth}(X_a), \tag{4.6}$$

where $\operatorname{orth}(X)$ denotes an orthonormal basis for the range of the matrix X. We note that since $\widehat{J}_k$ is of full rank, V_k is well defined (up to an orthogonal transformation)

even when $A-\mu_k C-\lambda_k I$ is singular. The approach described here for computing a basis of a null space of a matrix via an augmented system is inspired by [38, 39, 47], which can be traced back to [41].

After obtaining the orthonormal basis matrix V_k of the desired projection subspace, we can define the 2DRQ:

$$(A_k, C_k) \equiv (V_k^H A V_k, V_k^H C V_k), \tag{4.7}$$

where for the sake of exposition, without loss of generality, we assume that V_k is up to another orthogonal transformation such that

$$C_k = V_k^H C V_k = \begin{bmatrix} c_{1,k} & \\ & c_{2,k} \end{bmatrix} \quad \text{is diagonal with} \quad c_{1,k} \geq c_{2,k}. \tag{4.8}$$

When C_k is indefinite, the following 2×2 2DEVP of (A_k, C_k) has explicit solutions:

$$(A_k - \nu C_k - \theta I)z = 0, \tag{4.9a}$$

$$z^H C_k z = 0, \tag{4.9b}$$

$$z^H z = 1. \tag{4.9c}$$

Specifically, we first note that up to a scaling, a nonzero vector z satisfying (4.9b) and (4.9c) is of the form

$$z(\alpha) = \frac{1}{\sqrt{c_{1,k}-c_{2,k}}} \begin{bmatrix} \sqrt{-c_{2,k}} \\ \alpha\sqrt{c_{1,k}} \end{bmatrix}, \tag{4.10}$$

where $\alpha \in \mathbb{C}$ and $|\alpha| = 1$. By multiplying $z^H(\alpha)$ and $z^H(\alpha)C$ on the left of (4.9a), we have

$$\nu(\alpha) = \frac{z^H(\alpha) C_k A_k z(\alpha)}{\|C_k z(\alpha)\|^2} \quad \text{and} \quad \theta(\alpha) = z^H(\alpha) A_k z(\alpha), \tag{4.11}$$

and the triplet $(\nu(\alpha), \theta(\alpha), z(\alpha))$ satisfies (4.9a). Since there exist infinitely many α with $|\alpha| = 1$, the 2DEVP (4.9) seems to possess an infinite number of 2D-eigenvalues. However, this does not imply that any triplet $(\nu(\alpha), \theta(\alpha), z(\alpha))$ defined in (4.10) and (4.11) is a 2D-eigentriplet of the 2DEVP (4.9) since only *real* pairs $(\nu(\alpha), \theta(\alpha))$ are of interest.

Obviously, $\theta(\alpha)$ in (4.11) is always real. By straightforward calculation, we have

$$\nu(\alpha) = \frac{a_{11,k} - a_{22,k} + (c_{1,k}\alpha a_{12,k} + c_{2,k}\overline{\alpha a_{12,k}})/\sqrt{-c_{1,k}c_{2,k}}}{c_{1,k} - c_{2,k}}, \tag{4.12}$$

where $a_{ij,k}$ is the (i,j)-element of A_k. Since $c_{1,k} > 0$ and $c_{2,k} < 0$, $\nu(\alpha)$ is real if and only if $\alpha a_{12,k}$ is real. There are two cases:

- $a_{12,k} \neq 0$. In this case, there are exactly two choices of α: $\alpha_{k,j} = \pm |a_{12,k}|/a_{12,k}$, $j = 1, 2$, such that $\alpha a_{12,k}$ is real. The 2DEVP (4.9) has exactly two 2D-eigentriplets given by

  $$(\nu(\alpha_{k,j}), \theta(\alpha_{k,j}), z(\alpha_{k,j})), \quad j = 1, 2. \tag{4.13}$$

  Furthermore, $\theta(\alpha_{k,j})$ are simple eigenvalues of $A_k - \nu(\alpha_{k,j})C_k$.
- $a_{12,k} = 0$. In this case, any α with $|\alpha| = 1$ leads to the same real $(\nu(\alpha), \theta(\alpha))$. The 2D-eigentriplets of the 2DEVP (4.9) are given by

$$(\nu(\alpha), \theta(\alpha), z(\alpha)) \tag{4.14}$$

for any $\alpha \in \mathbb{C}$ with $|\alpha| = 1$. $\theta(\alpha)$ is an eigenvalue of $A_k - \nu(\alpha)C_k$ of multiplicity 2.

By the 2D-eigentriplets (4.13) or (4.14) of (A_k, C_k), we can use the following 2D-Ritz triplets to define the $(k+1)$st iterate $(\mu_{k+1}, \lambda_{k+1}, x_{k+1})$:

$$\mu_{k+1} = \nu(\alpha_{k,j}), \quad \lambda_{k+1} = \theta(\alpha_{k,j}), \quad \text{and} \quad x_{k+1} = V_k z(\alpha_{k,j}), \tag{4.15}$$

when $a_{12,k} \neq 0$, where j is the index such that $|\mu_k - \nu(\alpha_{k,j})| + |\lambda_k - \theta(\alpha_{k,j})|$ is smaller for $j = 1, 2$. Otherwise, when $a_{12,k} = 0$, the $k+1$st iterate $(\mu_{k+1}, \lambda_{k+1}, x_{k+1})$ is given by

$$\mu_{k+1} = \nu(1), \quad \lambda_{k+1} = \theta(1) \quad \text{and} \quad x_{k+1} = V_k z(1), \tag{4.16}$$

where, for the sake of convenience, we choose $\alpha = 1$ in (4.14).

When C_k is not indefinite, as we may encounter at early stages of iterations, we propose the following strategy for determining the $(k+1)$st iterate $(\mu_{k+1}, \lambda_{k+1}, x_{k+1})$. First, since the exact 2D-eigenvector x_* satisfies $x_*^H C x_* = 0$, we choose a unit vector x_{k+1} to minimize $|x^H C x|$ for $x \in \text{span}\{V_k\}$. Specifically, when $c_{1,k} \neq c_{2,k}$, up to a scaling, x_{k+1} is uniquely determined by

$$x_{k+1} = \begin{cases} V_k e_1, & |c_{1,k}| < |c_{2,k}|, \\ V_k e_2, & |c_{1,k}| > |c_{2,k}|. \end{cases} \tag{4.17}$$

When $c_{1,k} = c_{2,k}$, we use

$$x_{k+1} = V_k w / \|V_k w\|, \tag{4.18}$$

where w is a uniformly distributed random vector on $[-1, 1]$. Once x_{k+1} is determined by (4.17) or (4.18), $(\mu_{k+1}, \lambda_{k+1})$ is obtained by solving the following least squares problem:

$$(\mu_{k+1}, \lambda_{k+1}) = \arg \min_{\nu, \theta \in \mathbb{R}} \|A x_{k+1} - \nu C x_{k+1} - \theta x_{k+1}\|. \tag{4.19}$$

4.3. Algorithm outline. Algorithm 4.1 summarizes the derivation in the previous section for an algorithm to compute a 2D-eigentriplet. It is called *2DRQI* since the algorithm is an extension of the RQI for a Hermitian matrix A. By (4.4) and (4.6), we see that when $A - \mu_k C - \lambda_k I$ is nonsingular,

$$\text{span}\{V_k\} = \text{span}\{(A - \mu_k C - \lambda_k I)^{-1} x_k, (A - \mu_k C - \lambda_k I)^{-1} C x_k\}.$$

If $C = 0$ and λ_k is the Rayleigh quotient of A and x_k, then $\text{span}\{V_k\} = \text{span}\{(A - \lambda_k I)^{-1} x_k\}$ is the one used in the classical RQI; see, e.g., [40, sec. 4.6].

A few remarks on Algorithm 4.1 are in order. (1) A proper initial (μ_0, λ_0, x_0) is critical for the rapid convergence of the algorithm. The initial pair (μ_0, λ_0) should be close to a 2D-eigenvalue of interest. For the initial vector x_0, we first compute a 2D-Ritz triplet (ν, θ, z) of 2DRQ $(X^H A X, X^H C X)$, where X consists of the two orthonormal eigenvectors corresponding to two eigenvalues of $A - \mu_0 C$ closest to λ_0, and then we set x_0 to be the 2D-Ritz vector Xz associated with the 2D-Ritz value (ν, θ) closest to (μ_0, λ_0). (2) To solve the linear system (4.4), we should exploit the structure and sparsity of matrices A and C. See numerical examples in section 7.

Algorithm 4.1 2DRQI.

Require: $n \times n$ Hermitian matrices A and C, where C is indefinite; initial (μ_0, λ_0, x_0), `tol`, `maxit`.

Ensure: An approximate 2D-eigentriplet $(\widehat{\mu}, \widehat{\lambda}, \widehat{x})$ and an estimated backward error η_1.

for $k = 0, 1, 2, \ldots,$ `maxit` **do**
  solve the linear system (4.4) for the $n \times 2$ matrix X_a.
  set $V_k = \mathrm{orth}(X_a)$ and update V_k to satisfy (4.8).
  solve the 2×2 2DEVP (4.9) of $(A_k, C_k) = (V_k^H A V_k, \mathrm{Diag}(c_{1,k}, c_{2,k}))$.
  if C_k is indefinite **then**
    determine $(\mu_{k+1}, \lambda_{k+1}, x_{k+1})$ by (4.15) or (4.16).
  else
    if $|c_{1,k}| \neq |c_{2,k}|$ **then**
      determine x_{k+1} by (4.17).
    else
      determine x_{k+1} by (4.18).
    end if
    determine $(\mu_{k+1}, \lambda_{k+1})$ by solving (4.19).
  end if
  exit for-loop if $\eta_1(\mu_{k+1}, \lambda_{k+1}, x_{k+1}) \leq$ `tol`.
end for
return $(\widehat{\mu}, \widehat{\lambda}, \widehat{x}) = (\mu_{k+1}, \lambda_{k+1}, x_{k+1})$ and $\eta_1(\widehat{\mu}, \widehat{\lambda}, \widehat{x})$.

(3) We use an estimate η_1 of the backward error of approximate 2D-eigentriplet (μ_k, λ_k, x_k) as the stopping criterion; see Theorem 5.2 in section 5. In section 7, we will provide examples to demonstrate that the 2DRQI is locally quadratically convergent. (4) Although Algorithm 4.1 is designed to compute a stationary point of the eigencurve $\lambda_j(\mu)$ for some j, there is a lack of control on j. In section 6, we will show how to address this issue through carefully choosing initial vectors and exploiting the convexity in applications to compute the desired eigencurve $\lambda_j(\mu)$ for a given j.

In [30], under the proper conditions, we prove that the 2DRQI is locally quadratically convergent for the generic cases I and II discussed in subsection 4.1.

5. Backward error analysis. It is well known that the backward error of an approximate solution is a reliable and effective stopping criterion for an iterative algorithm. In this section, we provide a backward error analysis of the 2DEVP (1.1). The resulting backward error estimate can be used as the stopping criterion of the 2DRQI (Algorithm 4.1). In subsection 6.2, the notion of the backward error analysis of the 2DEVP will be extended to the computation of the distance to instability. We start with the following theorem.

THEOREM 5.1. *Let $(\widehat{\mu}, \widehat{\lambda}, \widehat{x})$ be an approximate 2D-eigentriplet of (A, C) with $\widehat{\mu}, \widehat{\lambda} \in \mathbb{R}$ and $\|\widehat{x}\| = 1$. Then there exist Hermitian matrices δA and δC such that* (i) *$C + \delta C$ is indefinite, and* (ii) *$(\widehat{\mu}, \widehat{\lambda}, \widehat{x})$ is an exact 2D-eigentriplet of the perturbed matrix pair $(A + \delta A, C + \delta C)$:*

$$(A + \delta A - \widehat{\mu}(C + \delta C))\,\widehat{x} = \widehat{\lambda}\widehat{x}, \tag{5.1a}$$

$$\widehat{x}^H (C + \delta C)\widehat{x} = 0, \tag{5.1b}$$

$$\widehat{x}^H \widehat{x} = 1. \tag{5.1c}$$

Proof. We prove by construction. We first find the desired perturbation matrix δC to satisfy (5.1b). Define $\delta\widehat{C} = -(\widehat{x}^H C\widehat{x})I$. Then

$$\widehat{x}^H(C+\delta\widehat{C})\widehat{x} = 0. \tag{5.2}$$

If $C + \delta\widehat{C}$ is indefinite, then (5.1b) holds by taking $\delta C = \delta\widehat{C}$. If $C + \delta\widehat{C}$ is not indefinite, then $C + \delta\widehat{C}$ is positive or negative semidefinite. Equation (5.2) implies $(C + \delta\widehat{C})\widehat{x} = 0$. Let Q be an orthogonal matrix with $Qe_1 = \widehat{x}$. Then we have $Q^H(C + \delta\widehat{C})Qe_1 = 0$ and

$$Q^H(C+\delta\widehat{C})Q = \begin{bmatrix} 0 & 0 \\ 0 & \widehat{C}_1 \end{bmatrix},$$

where $\widehat{C}_1$ is an $(n-1)$-by-$(n-1)$ matrix. Define

$$\delta C = \delta\widehat{C} + Q \begin{array}{c} \\ 1 \\ 1 \\ n-2 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 1 & n-2 \end{array} \\ \begin{bmatrix} 0 & \Delta & 0 \\ \Delta & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{array} Q^H \tag{5.3}$$

with a nonzero scalar Δ. Then it can be verified that $Q^H(C+\delta C)Q$ (and thus $C+\delta C$) is indefinite and (5.1b) holds.

For finding the desired perturbation matrix δA to satisfy (5.1a), let $\delta\widehat{A}$ be a Hermitian matrix such that $\delta\widehat{A}\,\widehat{x} = h/\|h\|$, $h = -(A-\widehat{\lambda}I)\widehat{x} + \widehat{\mu}(C+\delta C)\widehat{x}$. For example, $\delta\widehat{A}$ can be a Householder matrix [17, Thm. 2.1.13]. Then it is straightforward to verify that (5.1a) holds with $\delta A = \|h\|\delta\widehat{A}$. This completes the proof. ∎

By Theorem 5.1, the backward error η of an approximate 2D-eigentriplet $(\widehat{\mu}, \widehat{\lambda}, \widehat{x})$ of (A, C) is defined as the infimum of the normwise relative perturbation of A and C such that $(\widehat{\mu}, \widehat{\lambda}, \widehat{x})$ is an exact 2D-eigentriplet of the perturbed 2DEVP (5.1):

$$\begin{aligned} \eta \equiv \inf\Big\{ \epsilon \;\Big|\; & \exists\, \delta A, \delta C \text{ s.t. } \|\delta A\| \le \epsilon\|A\|, \|\delta C\| \\ & \le \epsilon\|C\|, C+\delta C \text{ is indefinite and (5.1) holds} \Big\}. \end{aligned} \tag{5.4}$$

The following theorem provides a tight computable estimate of η.

THEOREM 5.2. *Let $(\widehat{\mu}, \widehat{\lambda}, \widehat{x})$ be an approximate 2D-eigentriplet of (A, C) with $\widehat{\mu}, \widehat{\lambda} \in \mathbb{R}$ and $\|\widehat{x}\| = 1$, and*

$$\eta_1 = \max\left\{ \frac{|\gamma_A|}{\|A\|}, \frac{|\gamma_C|}{\|C\|}, \frac{\|r\|}{\|A\| + |\widehat{\mu}|\|C\|} \right\}, \tag{5.5}$$

where $\gamma_A = \widehat{x}^H A\widehat{x} - \widehat{\lambda}$, $\gamma_C = \widehat{x}^H C\widehat{x}$, and $r = (A - \widehat{\mu}C - \widehat{\lambda}I)\widehat{x}$. Then the backward error η of $(\widehat{\mu}, \widehat{\lambda}, \widehat{x})$ defined in (5.4) satisfies

$$\eta_1 \le \eta \le \sqrt{2}\,\eta_1. \tag{5.6}$$

Proof. We first prove the lower bound $\eta \ge \eta_1$. For any δA and δC satisfying the perturbed 2DEVP (5.1), by (5.1a) and (5.1b), we have $\widehat{x}^H(A+\delta A)\widehat{x} = \widehat{\lambda}$. Hence by the definition (5.4) of η, we have

$$\eta \ge \frac{\|\delta C\|}{\|C\|} \ge \frac{|\widehat{x}^H \delta C \widehat{x}|}{\|C\|} = \frac{|\widehat{x}^H C\widehat{x}|}{\|C\|} = \frac{|\gamma_C|}{\|C\|}, \tag{5.7}$$

and

$$\eta \geq \frac{\|\delta A\|}{\|A\|} \geq \frac{|\widehat{x}^H \delta A \widehat{x}|}{\|A\|} = \frac{|\widehat{x}^H A \widehat{x} - \widehat{\lambda}|}{\|A\|} = \frac{|\gamma_A|}{\|A\|}. \tag{5.8}$$

Now by (5.1a), for the norm of the residual vector r,

$$\|r\| = \|(A - \widehat{\mu} C - \widehat{\lambda} I)\widehat{x}\| = \|(\delta A - \widehat{\mu}\delta C)\widehat{x}\| \leq \|\delta A\| + |\widehat{\mu}|\|\delta C\| \leq (\|A\| + |\widehat{\mu}|\|C\|)\epsilon.$$

Therefore, by the definition (5.4) of η, we have

$$\eta \geq \frac{\|r\|}{\|A\| + |\widehat{\mu}|\|C\|}. \tag{5.9}$$

Combining (5.7), (5.8), and (5.9), we have $\eta \geq \eta_1$.

The gist of finding the upper bound of η, namely $\eta \leq \sqrt{2}\eta_1$, is to find two particular perturbation matrices δA and δC such that

$$\delta A \widehat{x} - \widehat{\mu}\, \delta C \widehat{x} = -r, \tag{5.10a}$$

$$\widehat{x}^H \delta C \widehat{x} = -\gamma_C, \tag{5.10b}$$

and

$$C + \delta C \text{ is indefinite}, \tag{5.11}$$

and then derive the upper bound of η from the upper bound of $\max\{\frac{\|\delta A\|}{\|A\|}, \frac{\|\delta C\|}{\|C\|}\}$. We first note that we can safely discard the condition (5.11). This is due to the fact that when (5.10) holds, using the same arguments as in the proof of Theorem 5.1, we can add infinitesimal perturbation to $\delta A, \delta C$ to guarantee (5.10) and (5.11) hold. Since the backward error η takes the infimum, the quantity $\max\{\frac{\|\delta A\|}{\|A\|}, \frac{\|\delta C\|}{\|C\|}\}$ is still an upper bound.

To find δA and δC satisfying (5.10), let us define

$$\widetilde{a} \equiv -\frac{\|A\|}{\|A\| + |\widehat{\mu}|\,\|C\|}(I - \widehat{x}\widehat{x}^H)r, \quad \widetilde{c} \equiv \frac{\operatorname{sign}(\widehat{\mu})\|C\|}{\|A\| + |\widehat{\mu}|\,\|C\|}(I - \widehat{x}\widehat{x}^H)r,$$

where $\operatorname{sign}(\widehat{\mu}) = \widehat{\mu}/|\widehat{\mu}|$ if $\widehat{\mu} \neq 0$ and 1 otherwise. Then $\widetilde{a}$ and $\widetilde{c}$ are orthogonal to $\widehat{x}$ and satisfy

$$\widetilde{a} - \widehat{\mu}\,\widetilde{c} = -(I - \widehat{x}\widehat{x}^H)\, r.$$

Next, let us define $a \equiv (-\widehat{x}^H r - \widehat{\mu}\gamma_C)\widehat{x} + \widetilde{a}$ and $c \equiv -\gamma_C \widehat{x} + \widetilde{c}$. Then it holds that

$$\begin{cases} a - \widehat{\mu} c = (\widetilde{a} - \widehat{\mu}\widetilde{c}) - \widehat{x}^H r \widehat{x} = -(I - \widehat{x}\widehat{x}^H)r - \widehat{x}^H r \widehat{x} = -r, \\ \widehat{x}^H c = -\gamma_C. \end{cases}$$

From the vectors a and c, we can construct Hermitian matrices δA and δC, say real constant multiples of Householder reflections [17, Thm. 2.1.13] satisfying $\delta A \widehat{x} = a$, $\delta C \widehat{x} = c$, $\|\delta A\| = \|a\|$, and $\|\delta C\| = \|c\|$. Then δA and δC are desired matrices satisfying (5.10).

For δA, by the definition of r, we have $-\widehat{x}^H r - \widehat{\mu}\gamma_C = -\gamma_A$, and thus

$$\begin{aligned} \frac{\|\delta A\|}{\|A\|} = \frac{\|a\|}{\|A\|} = \frac{\|-\gamma_A \widehat{x} + \widetilde{a}\|}{\|A\|} = \frac{\sqrt{\gamma_A^2 + \|\widetilde{a}\|^2}}{\|A\|} = \sqrt{\left(\frac{|\gamma_A|}{\|A\|}\right)^2 + \left(\frac{\|\widetilde{a}\|}{\|A\|}\right)^2} \\ \leq \sqrt{\left(\frac{|\gamma_A|}{\|A\|}\right)^2 + \left(\frac{\|r\|}{\|A\| + |\widehat{\mu}|\,\|C\|}\right)^2} \leq \sqrt{2}\eta_1. \end{aligned} \tag{5.12}$$

By an analogous derivation, for δC, we have

$$\frac{\|\delta C\|}{\|C\|} \leq \sqrt{2}\eta_1. \tag{5.13}$$

Combining the upper bounds (5.12) and (5.13), we have $\eta \leq \sqrt{2}\eta_1$. This completes the proof. ∎

6. Applications. In this section, we discuss two eigenvalue optimization problems that can be reformulated as the 2DEVP (1.1) and then solved by the 2DRQI (Algorithm 4.1).

6.1. Minmax of Rayleigh quotients. Given Hermitian matrices $A, B \in \mathbb{C}^{n\times n}$, the minmax problem of Rayleigh quotients (RQminmax)

$$\min_{x\neq 0} \max \left\{ \frac{x^H A x}{x^H x}, \frac{x^H B x}{x^H x} \right\} \tag{6.1}$$

arises from quadratically constrained quadratic programs (QCQP) [12], the trust region methods for nonlinear equality constrained optimization [54], transmit beamforming [13, 19, 55], MIMO relay optimization [46], and cognitive radio networks [56]. The RQminmax (6.1) is closely related to the well-known S-Lemma in control theory and robust optimization [43, 53]. We have the following theorem to characterize a solution of the RQminmax (6.1).

THEOREM 6.1. *Let λ_A be the minimum eigenvalue and x_A be a corresponding unit eigenvector of A, let λ_B be the minimum eigenvalue and x_B be a corresponding unit eigenvector of B, and let $\rho_A(x) = x^H Ax/x^H x$ and $\rho_B(x) = x^H Bx/x^H x$ be the Rayleigh quotients of A and B.*

I. *If $\lambda_A \geq \rho_B(x_A)$, then x_A is a solution of the RQminmax* (6.1).
II. *If $\lambda_B \geq \rho_A(x_B)$, then x_B is a solution of the RQminmax* (6.1).
III. *Otherwise, that is, if $\lambda_A < \rho_B(x_A)$ and $\lambda_B < \rho_A(x_B)$, let μ_* be an optimizer of the eigenvalue optimization problem (EVOPT),*

$$\max_{\mu\in\mathbb{R}} \lambda_{\min}(A - \mu C), \tag{6.2}$$

and let V_{μ_*} be the set of eigenvectors x_* corresponding to $\lambda_{\min}(A-\mu_* C)$ and $x_*^H C x_* = 0$, where $C = A - B$. Then (a) $\mu_* \in [0,1]$, (b) $V_{\mu_*} \neq \emptyset$, *and* (c) *any $x_* \in V_{\mu_*}$ is a solution of the RQminmax* (6.1).

Proof. For case I, we note that for any $x \neq 0$, $\max\{\rho_A(x), \rho_B(x)\} \geq \rho_A(x) \geq \lambda_A$. On the other hand, $\max\{\rho_A(x_A), \rho_B(x_A)\} = \lambda_A$. Thus $x_* = x_A$ is a solution of the RQminmax (6.1).

Case II can be proven by exchanging the roles of A and B in the proof of case I.

The proof of case III. is divided into two subcases: Case III. (1): $\lambda_A < \theta_B$ and $\lambda_B < \theta_A$; and case III. (2): the negation of case III. (1), i.e., the inequalities $\lambda_A < \theta_B$ and $\lambda_B < \theta_A$ do not hold simultaneously, where $\theta_A = \lambda_{\min}(S_B^H A S_B)$, $\theta_B = \lambda_{\min}(S_A^H B S_A)$, and S_A and S_B are orthonormal basis of the eigensubspace of λ_A and λ_B, respectively.

Consider case III. (1). For (a), it is sufficient to prove that

$$g'^{(-)}(1) < 0 \quad \text{and} \quad g'^{(+)}(0) > 0, \tag{6.3}$$

where $g(\mu) = \lambda_{\min}(A - \mu C)$. For the first inequality in (6.3), if $g(1) = \lambda_B$ is a simple eigenvalue of $A - C = B$, then by Corollary 2.4, g is differentiable at $\mu = 1$ and

$g'(1) = -x_B^H C x_B = -x_B^H (A-B) x_B = \lambda_B - \theta_A < 0$. Thus the first inequality holds. If $g(1)$ is not simple, by Theorem 2.3, $g'^{(-)}(1)$ equals to the maximum eigenvalue of $-C_k = S_B^H (B-A) S_B = \lambda_B I - S_B^H A S_B$. Since $\lambda_B < \theta_A \equiv \lambda_{\min}(S_B^H A S_B)$, $-C_k$ is negative definite. This implies that $g'^{(-)}(1) < 0$. By an analogous argument, $g'^{(+)}(0) > 0$. Thus the result (a) holds.

For (b), note that $(\mu_*, \lambda_{\min}(A - \mu_* C))$ is a 2D-eigenvalue of (A, C) according to Theorem 3.1. Then the associated 2D-eigenvectors belong to V_{μ_*}, and thus we obtain the result (b).

For (c), we first claim that if x_* is the solution of the RQminmax (6.1), then $x_*^H C x_* = 0$. We prove by contradiction. Assume $x_*^H C x_* > 0$, i.e., $x_*^H A x_* > x_*^H B x_*$. Then x_* does not belong to S_A since otherwise $\lambda_A = \rho_A(x_*) > \rho_B(x_*) \geq \theta_B$, which contradicts the condition that $\lambda_A < \theta_B$. Consider $x(t) = x_* + t\,\mathrm{sign}(x_A^H x_*) x_A$ with $t > 0$, where by convention $\mathrm{sign}(0) = 1$. A straightforward calculation shows

$$\rho_A(x(t)) = \frac{\|x_*\|^2 \rho_A(x_*) + \left(t^2 \|x_A\|^2 + 2t |x_*^H x_A|\right) \lambda_A}{\|x_*\|^2 + t^2 \|x_A\|^2 + 2t |x_*^H x_A|} < \rho_A(x_*). \tag{6.4}$$

On the other hand, by the continuity of $\rho_A(x(t))$ and $\rho_B(x(t))$ with respect to t, $\rho_B(x(t)) < \rho_A(x(t))$ holds for a sufficiently small t. This implies that for such t we have

$$\max\{\rho_A(x(t)), \rho_B(x(t))\} = \rho_A(x(t)) < \rho_A(x_*) = \max\{\rho_A(x_*), \rho_B(x_*)\},$$

which contradicts the condition that x_* is the solution of the RQminmax (6.1). Hence $x_*^H C x_* \leq 0$. A similar argument leads to $x_*^H C x_* \geq 0$. Therefore, we conclude $x_*^H C x_* = 0$, and we have

$$\begin{aligned} \min_{x \neq 0} \max\{\rho_A(x), \rho_B(x)\} &= \min_{\substack{x^H x = 1 \\ x^H C x = 0}} \max\{x^H A x, x^H B x\} = \min_{\substack{x^H x = 1 \\ x^H C x = 0}} x^H A x \\ &= \max_{\mu \in \mathbb{R}} \lambda_{\min}(A - \mu C), \end{aligned}$$

where the last equality is from Theorem 3.4. Thus for $x_* \in V_{\mu_*}$, we have

$$\begin{aligned} \rho_A(x_*) = \rho_B(x_*) = \rho_{A-\mu_* C}(x_*) = \lambda_{\min}(A - \mu_* C) = \max_{\mu \in \mathbb{R}} \lambda_{\min}(A - \mu C) \\ = \min_{x \neq 0} \max\{\rho_A(x), \rho_B(x)\}, \end{aligned}$$

which implies the result (c). This completes the proof of case III. (1).

Case III. (2) implies that at least one of the following conditions holds: (i) $\lambda_A \geq \theta_B$; (ii) $\lambda_B \geq \theta_A$. Let us assume (i). It can be shown analogously if we assume (ii).

By the condition under case III, i.e., $\lambda_A < \rho_B(x_A)$ and $\lambda_B < \rho_A(x_B)$, we have

$$-x_A^H C x_A = -\lambda_A + \rho_B(x_A) > 0 \quad \text{and} \quad -x_B^H C x_B = -\rho_A(x_B) + \lambda_B < 0. \tag{6.5}$$

Note that x_A and x_B are also eigenvectors of $A - 0 \cdot C = A$ and $A - 1 \cdot C = B$, respectively. Thus by Theorem 2.3, the inequalities in (6.5) imply that

$$\begin{aligned} g'_-(0) = \lambda_{\max}(-S_A^H C S_A) \geq -x_A^H C x_A > 0 \quad \text{and} \\ g'_+(1) = \lambda_{\min}(-S_B^H C S_B) \leq -x_B^H C x_B < 0. \end{aligned} \tag{6.6}$$

Let μ_* be an optimizer of the EVOPT (6.2). Then by (6.6) and the concavity of $g(\mu)$, we conclude that $\mu_* \in [0, 1]$. Therefore, the result (a) holds.

The result (b) follows from the same argument as in case III. (1).

To prove (c), we first calculate the optimal value of the EVOPT (6.2). Since $\lambda_A \geq \theta_B$, by denoting z_B as the unit eigenvector of $S_A^H B S_A$ corresponding to θ_B and the definition of S_A, we have

$$\rho_A(S_A z_B) = \lambda_A \geq \theta_B = \rho_B(S_A z_B).$$

Let $\widetilde{x}_A = S_A z_B$. Then $-\widetilde{x}_A^H C \widetilde{x}_A \leq 0$ and thus by Theorem 2.3,

$$g'_+(0) = \lambda_{\min}(-S_A^H C S_A) \leq -\widetilde{x}_A^H C \widetilde{x}_A \leq 0. \tag{6.7}$$

According to (6.6), (6.7), and the concavity of $g(\mu)$ (see Theorem 3.8), 0 is an optimizer of the EVOPT (6.2) and thus

$$\max_{\mu \in \mathbb{R}} \lambda_{\min}(A - \mu C) = \lambda_A. \tag{6.8}$$

Now for any $x_* \in V_{\mu_*}$, we have

$$\begin{aligned} \rho_B(x_*) = \rho_A(x_*) = \lambda_{\min}(A - \mu_* C) &= \max_{\mu \in \mathbb{R}} \lambda_{\min}(A - \mu C) \\ = \lambda_A &= \min_{x \neq 0} \max\{\rho_A(x), \rho_B(x)\}, \end{aligned}$$

where the first equality results from $x_*^H C x_* = 0$, the second equality results from the fact that $\rho_A(x_*) = \rho_{A-\mu_* C}(x_*)$ and x_* is an eigenvector corresponding to $\lambda_{\min}(A - \mu_* C)$, the third equality comes from the fact that μ_* is an optimizer, the fourth equality results from (6.8), and the last equality holds according to

$$\begin{aligned} \lambda_A &\leq \min_{x \neq 0} \max\{\lambda_A, \rho_B(x)\} \leq \min_{x \neq 0} \max\{\rho_A(x), \rho_B(x)\} \leq \max\{\rho_A(S_A z_B), \rho_B(S_A z_B)\} \\ &= \lambda_A \end{aligned}$$

as $\theta_B \leq \lambda_A$. Thus x_* is the solution of the RQminmax (6.1) and the result (c) holds. This completes the proof of case III. (2). ∎

Remark 6.2. In [12], Gaurav and Hari considered the characterization of the solution of the RQminmax (6.1) similar to Theorem 6.1. However, it is assumed that eigenvalues λ_A, λ_B, and $\lambda_{\min}(A - \mu_* C)$ are all simple. Furthermore, there is no result (a) in case III.

By the characterization of the solution of the RQminmax (6.1) in Theorem 6.1, for cases I and II, we can obtain a solution of the RQminmax (6.1) regardless of the multiplicities of λ_A and λ_B. For case III, we know that (μ_*, λ_*) with $\lambda_* = \lambda_{\min}(A - \mu_* C)$ is the minimum 2D-eigenvalue of (A, C) (see Definition 3.6). On the other hand, by the definition of V_{μ_*}, up to a scaling, $x_* \in V_{\mu_*}$ if and only if x_* is a 2D-eigenvector associated with (μ_*, λ_*). Thus the RQminmax (6.1) in case III. turns to calculating a minimum 2D-eigenvalue and the corresponding 2D-eigenvector of (A, C).

Based on the fact that μ_* of the minimum 2D-eigenvalue (μ_*, λ_*) must be in $[0, 1]$, we can combine the bisection search and the 2DRQI (Algorithm 4.1). Starting with the initial search interval $[a, b] = [0, 1]$ of the EVOPT (6.2), let $\mu_0 = (a + b)/2$, $\lambda_0 = \lambda_{\min}(A - \mu_0 C)$, and x_0 be the one recommended for the 2DRQI (Algorithm 4.1). Then we can use the 2DRQI with the initial (μ_0, λ_0, x_0) to find a 2D-eigentriplet $(\widehat{\mu}, \widehat{\lambda}, \widehat{x})$ of (A, C).

If $\widehat{\lambda} = \lambda_{\min}(A - \widehat{\mu}C)$, then according to Corollary 3.5,

$$\widehat{\lambda} = \max_{\mu \in \mathbb{R}} \lambda_{\min}(A - \mu C).$$

Thus $\widehat{\mu}$ is an optimizer of EVOPT (6.2) and $\widehat{x}$ is the solution of RQminmax (6.1).

If $\widehat{\lambda} \neq \lambda_{\min}(A - \widehat{\mu}C)$, or the 2DRQI does not converge, then we can use the concavity of $g(\mu) = \lambda_{\min}(A - \mu C)$ (see Theorem 3.8) to bisect the interval $[a, b]$ and run the 2DRQI with a new initial (μ_0, λ_0, x_0). This bisection search strategy works under the assumption on the nonsingularity of Jacobian at the target 2D-eigentriplet due to the following facts:

- if $(x^{(n)})^H C x^{(n)} \leq 0$, where $x^{(n)}$ is an eigenvector corresponding to $\lambda_0 = \lambda_{\min}(A - \mu_0 C)$, then by Theorem 2.3, $g'_-(\mu_0) = \lambda_{\max}(-X_0(\mu_0)^H C X_0(\mu_0)) \geq 0$, where $X_0(\mu)$ is an orthonormal basis of the eigensubspace of $\lambda_{\min}(A - \mu C)$, and there is an optimizer μ_* of the EVOPT (6.2) such that $\mu_* \geq \mu_0$. Consequently, we set $a = \mu_0$ to half the search interval.
- if $(x^{(n)})^H C x^{(n)} > 0$, then by Theorem 2.3, $g'_+(\mu_0) = \lambda_{\min}(-X_0(\mu_0)^H C X_0(\mu_0)) < 0$ and there is an optimizer $\mu_*$of the EVOPT (6.2) such that $\mu_* \leq \mu_0$. Consequently, we set $b = \mu_0$ to half the search interval.

A combination of 2DRQI (Algorithm 4.1) and the bisection search described above is summarized in Algorithm 6.1 for solving the RQminmax (6.1), where in line 9 we use whether

Algorithm 6.1. Minmax of two RQs.

Require: n-by-n Hermitian matrices A and B, tolerance values `abstol`, `reltol`, and `backtol`.

Ensure: approximate solution $\widehat{x}$ and the optimal value $\widehat{\lambda}$ of RQminmax (6.1).

1: compute a minimum eigenpair (λ_A, x_A) of A. If $\lambda_A \geq \rho_B(x_A)$, then return $(\widehat{\lambda}, \widehat{x}) = (\lambda_A, x_A)$.
2: compute a minimum eigenpair (λ_B, x_B) of B. If $\lambda_B \geq \rho_A(x_B)$, then return $(\widehat{\lambda}, \widehat{x}) = (\lambda_B, x_B)$.
3: set $[a, b] = [0, 1]$.
4: **for** $k = 0, 1, 2, \ldots,$ until $b - a <$ `abstol` **do**
5: set $\mu_0 = (a + b)/2$.
6: compute two smallest eigenpairs $(\lambda_n, x^{(n)})$, $(\lambda_{n-1}, x^{(n-1)})$ of $A - \mu_0 C$.
7: compute the minimum 2D-Ritz triplet (ν, θ, z) of $(Z^H A Z, Z^H C Z)$, where $Z = \begin{bmatrix} x^{(n-1)} & x^{(n)} \end{bmatrix}$.
8: apply the 2DRQI (Algorithm 4.1) with the initial $(\mu_0, \lambda_0 = \lambda_n, x_0 = Zz)$ and the backward error tolerance `backtol`.
9: **if** 2DRQI converges to $(\widehat{\mu}, \widehat{\lambda}, \widehat{x})$ and $|\widehat{\lambda} - \lambda_{\min}(A - \widehat{\mu}C)| <$ `reltol` $\cdot\, (|1 - \widehat{\mu}|\, \|A\| + |\widehat{\mu}| \|B\|)$ **then**
10: **return** $(\widehat{\lambda}, \widehat{x})$.
11: **else**
12: **if** $(x^{(n)})^H C x^{(n)} \leq 0$ **then**
13: update $a = \mu_0$.
14: **else**
15: update $b = \mu_0$.
16: **end if**
17: **end if**
18: **end for**

$$|\widehat{\lambda} - \lambda_{\min}(A - \widehat{\mu}C)| = |\widehat{\lambda} - \lambda_{\min}\left((1-\widehat{\mu})A + \widehat{\mu}B\right)| < \texttt{reltol} \cdot (|1-\widehat{\mu}|\|A\| + |\widehat{\mu}|\|B\|)$$

to numerically check whether $\widehat{\lambda} = \lambda_{\min}(A - \widehat{\mu}C)$. Numerical examples for large scale RQminmax (6.1) arising from signal processing are presented in section 7.

6.2. The distance to instability. A basic problem in the stability analysis of dynamical systems is to compute the distance to instability (DTI); see, for example, [51, sec. 49]. In matrix notation, for a stable matrix $\widehat{A} \in \mathbb{C}^{m\times m}$, that is all eigenvalues of $\widehat{A}$ are located in the open left-half of the complex plane $\mathbb{C}$, the DTI is defined as

$$\beta(\widehat{A}) \equiv \min\left\{\|E\| \;\middle|\; \widehat{A} + E \text{ is unstable}, E \in \mathbb{C}^{m\times m}\right\}. \tag{6.9}$$

Van Loan [52] showed that $\beta(\widehat{A})$ can be recast as the singular value optimization

$$\beta(\widehat{A}) = \min_{\mu\in\mathbb{R}} \sigma_{\min}(\widehat{A} - \mu\mathtt{i}I), \tag{6.10}$$

where $\mathtt{i} = \sqrt{-1}$ and $\sigma_{\min}(X)$ refers to the smallest singular value of the matrix X. By the relation between the singular values of a matrix X and eigenvalues of Hermitian matrix $\left[\begin{smallmatrix} 0 & X^H \\ X & 0 \end{smallmatrix}\right]$ (see, e.g., [7, Thm. 3.3]), the singular value optimization (6.10) can be transformed to the eigenvalue optimization (EVOPT)

$$\beta(\widehat{A}) = \min_{\mu\in\mathbb{R}} \lambda_m(A - \mu C), \tag{6.11}$$

where A and C are $2m \times 2m$ matrices given by $A = \left[\begin{smallmatrix} & \widehat{A} \\ \widehat{A}^H & \end{smallmatrix}\right]$ and $C = \left[\begin{smallmatrix} & \mathtt{i}I \\ -\mathtt{i}I & \end{smallmatrix}\right]$, and $\lambda_m(A - \mu C)$ is the smallest positive eigenvalue of $A - \mu C$.

By Theorem 3.1, if μ_* is an optimizer of (6.11), then $(\mu_*, \beta(\widehat{A}))$ is a 2D-eigenvalue of the 2DEVP of (A, C). In addition, we have the following list of characterizations of the target 2D-eigentriplet:

- If $(\mu, \lambda, \left[\begin{smallmatrix} u \\ v \end{smallmatrix}\right])$ is a 2D-eigentriplet of (A, C), then $(\mu, -\lambda, \left[\begin{smallmatrix} -u \\ v \end{smallmatrix}\right])$ is also a 2D-eigentriplet. This implies the 2D-eigenvalues are symmetric with regard to $\lambda = 0$.
- The corresponding 2D-eigenvector $x_* = \left[\begin{smallmatrix} x_1 \\ x_2 \end{smallmatrix}\right]$ of $(\mu_*, \beta(\widehat{A}))$ must obey

$$\mathrm{Imag}(x_1^H x_2) = 0, \quad \text{and} \quad x_1^H x_1 = x_2^H x_2 = \frac{1}{2}, \tag{6.12}$$

  where, for the second identity, we use the fact that $x_1^H \widehat{A} x_2 = \overline{x_2^H \widehat{A}^H x_1}$.
- Based on the ordering of $2m$ eigenvalues of $A - \mu C$,

$$\lambda_1(\mu) \geq \lambda_2(\mu) \geq \cdots \geq \lambda_m(\mu) > 0 > \lambda_{m+1}(\mu) \geq \cdots \geq \lambda_{2m}(\mu), \tag{6.13}$$

  we have the following characterization of $\beta(\widehat{A})$:

$$\begin{aligned} \beta(\widehat{A}) &= \min\{\lambda \,|\, (\mu, \lambda) \text{ is a 2D-eigenvalue of } (A, C) \text{ and } \lambda > 0\} \\ &= -\max\{\lambda \,|\, (\mu, \lambda) \text{ is a 2D-eigenvalue of } (A, C) \text{ and } \lambda < 0\} \\ &= \min\{|\lambda| \,|\, (\mu, \lambda) \text{ is a 2D-eigenvalue of } (A, C)\}. \end{aligned} \tag{6.14}$$

- Theorem 3.7 implies the optimizer μ_* of the EVOPT (6.11) is in the interval $[-\|A\|, \|A\|]$, which is tighter than the interval $[-2\|A\|, 2\|A\|]$ derived in [52].

Algorithm 6.2 is an outline of a 2DRQI-based algorithm for computing $\beta(\widehat{A})$. A few remarks are in order.

Algorithm 6.2. DTI by 2DRQI.

Require: $m \times m$ stable matrix $\widehat{A}$, `reltol`, `tol`.

Ensure: 2D-eigentriplet $(\widehat{\mu}, \widehat{\lambda}, \widehat{x})$, where $\widehat{\lambda}$ is an estimate of the DTI $\beta(\widehat{A})$, and a backward error estimate η_2.

1: set μ_0 as the imaginary part of the rightmost eigenvalue of $\widehat{A}$.
2: compute the singular triplet (u, λ_0, v) corresponding to the smallest singular value of $\widehat{A} - \mu_0 \mathtt{i} I$.
3: apply the 2DRQI (Algorithm 4.1) with initial $(\mu_0, \lambda_0, x_0 = \frac{1}{\sqrt{2}}\left[\begin{smallmatrix} u \\ v \end{smallmatrix}\right])$ and stopping tolerance `tol` to compute an approximate 2D-eigentriplet $(\widehat{\mu}, \widehat{\lambda}, \widehat{x})$ of (A, C) and the corresponding backward error estimate η_2.
4: validate the computed DTI $\widehat{\lambda}$ with `reltol` (optional).

(1) The initial (μ_0, λ_0, x_0) (lines 1 and 2) follows the recommendation of [11] and is critical for the success of the computation.

(2) To satisfy the conditions (6.12) for the approximate 2D-eigenvector $x_k = \left[\begin{smallmatrix} x_{k,1} \\ x_{k,2} \end{smallmatrix}\right]$, we should add the following steps after line 14 in the 2DRQI (Algorithm 4.1):

$$1:\ x_{k+1,1} = \frac{\sqrt{2}}{2} x_{k+1,1} / \|x_{k+1,1}\|,$$

$$2:\ x_{k+1,2} = \frac{\sqrt{2}}{2} x_{k+1,2} / \|x_{k+1,2}\|.$$

With this normalization, we assume the computed x_k satisfies (6.12) exactly in the subsequent analysis.

(3) For the stopping criterion of the 2DRQI, we use a backward error estimate of the computed DTI. It has been a challenge to properly define the stopping criterion of iterative methods for computing DTI [11, 16, 18, 52]. A main reason is that it is meaningless to define the backward error for an estimated DTI $\widehat{\beta}$ *only.* Specifically, if a backward error $\widetilde{\eta}$ of $\widehat{\beta}$ is defined as

$$\widetilde{\eta} = \inf\left\{\epsilon \mid \exists\, \delta\widehat{A} \text{ such that } \|\delta\widehat{A}\| \le \epsilon \|\widehat{A}\| \text{ and } \beta(\widehat{A} + \delta\widehat{A}) = \widehat{\beta}\right\}, \tag{6.15}$$

then one can show that the calculation of the backward error $\widetilde{\eta}$ could be as hard as the calculation of the original $\beta(\widehat{A})$. This is analogous to the fact that for eigenvalue problems we do not define the backward error of an approximate eigenvalue only. We consider the backward error of an approximate eigenpair; see, e.g., [49, Thm. 1.3]. As an advantage of treating the DTI via the 2DEVP, we can establish the notion of the backward error for a computed DTI via an approximate 2D-eigentriplet. The resulting backward error estimation naturally leads to a reliable stopping criterion for an iterative DTI algorithm. To that end, let the approximate 2D eigentriplet $(\widehat{\mu}, \widehat{\lambda}, \widehat{x})$ of (A, C) be an exact 2D-eigentriplet of structurally perturbed 2DEVP

$$\begin{bmatrix} 0 & \widehat{A} + \delta\widehat{A} \\ \widehat{A}^H + \delta\widehat{A}^H & 0 \end{bmatrix} \widehat{x} - \widehat{\mu} C \widehat{x} = \widehat{\lambda}\widehat{x}, \tag{6.16a}$$

$$\widehat{x}^H C \widehat{x} = 0, \tag{6.16b}$$

$$\widehat{x}^H \widehat{x} = 1 \tag{6.16c}$$

for some $\delta\widehat{A}$. Then we can define a structure-preserving backward error of the 2DEVP of the DTI problem as follows:

$$\widehat{\eta}_\beta(\widehat{\mu},\widehat{\lambda},\widehat{x}) = \inf\left\{\epsilon \mid \exists\, \delta\widehat{A} \,\text{such that}\, \|\delta\widehat{A}\| \le \epsilon\|\widehat{A}\| \,\text{and}\, (6.16)\,\text{holds}\right\}. \tag{6.17}$$

We first note that the set in (6.17) is nonempty when the approximate 2D-eigenvector $\widehat{x} = \begin{bmatrix}\widehat{x}_1\\ \widehat{x}_2\end{bmatrix}$ satisfies the conditions (6.12). In fact, denote $r = \begin{bmatrix}r_1\\ r_2\end{bmatrix}$, where $r_1 = \widehat{A}\widehat{x}_2 - \widehat{\mu}\mathtt{i}\widehat{x}_2 - \widehat{\lambda}\widehat{x}_1$ and $r_2 = \widehat{A}^H\widehat{x}_1 + \widehat{\mu}\mathtt{i}\widehat{x}_1 - \widehat{\lambda}\widehat{x}_2$. Then it can be shown that the matrix

$$\delta\widehat{A}=\delta\widehat{A}_1+\delta\widehat{A}_2 \quad \text{with} \quad \delta\widehat{A}_1=-\left(I-\frac{\widehat{x}_1\widehat{x}_1^H}{\widehat{x}_1^H\widehat{x}_1}\right)\frac{r_1\widehat{x}_2^H}{\widehat{x}_2^H\widehat{x}_2} \quad \text{and} \quad \delta\widehat{A}_2=-\frac{\widehat{x}_1 r_2^H}{\widehat{x}_1^H\widehat{x}_1}$$

satisfies (6.16). Meanwhile, we have

$$\begin{aligned}\|\delta\widehat{A}\| &= \max_{\|z\|=1}\left\|(\delta\widehat{A}_1+\delta\widehat{A}_2)z\right\| = \max_{\|z\|=1}\sqrt{\left\|\delta\widehat{A}_1 z\right\|^2+\left\|\delta\widehat{A}_2 z\right\|^2}\\ &\le \sqrt{\left\|\delta\widehat{A}_1\right\|^2+\left\|\delta\widehat{A}_2\right\|^2} \le \sqrt{2\|r_1\|^2+2\|r_2\|^2} = \sqrt{2}\|r\|,\end{aligned} \tag{6.18}$$

where the second equality results from the fact that $\delta\widehat{A}_1 z$ is orthogonal to $\delta\widehat{A}_2 z$.

Next we provide an estimate of $\widehat{\eta}_\beta$. Since $\widehat{\eta}_\beta$ is the backward error of the stuctured 2DEVP (6.16), the backward error η in (5.4) of a generic (unstructured) 2DEVP is the lower bound of $\widehat{\eta}_\beta$:

$$\widehat{\eta}_\beta \ge \eta \ge \eta_1, \tag{6.19}$$

where η_1 is defined as in (5.5). On the other hand, by the definition of $\widehat{\eta}_\beta$ and (6.18), we have an upper bound of $\widehat{\eta}_\beta$:

$$\widehat{\eta}_\beta \le \eta_2 \equiv \sqrt{2}\frac{\|r\|}{\|\widehat{A}\|}. \tag{6.20}$$

By the facts that $\|\widehat{A}\| = \|A\|$ and $\|C\| = 1$, we have

$$\frac{\eta_2}{\eta_1} \le \frac{\sqrt{2}\frac{\|r\|}{\|\widehat{A}\|}}{\frac{\|r\|}{\|A\|+|\widehat{\mu}|\|C\|}} = \sqrt{2}\left(1+\frac{|\widehat{\mu}|}{\|\widehat{A}\|}\right). \tag{6.21}$$

Combining (6.19), (6.20), and (6.21), we have

$$\frac{1}{\sqrt{2}\left(1+\frac{|\widehat{\mu}|}{\|\widehat{A}\|}\right)}\eta_2 \le \widehat{\eta}_\beta \le \eta_2. \tag{6.22}$$

Therefore η_2 defined in (6.20) can be used as an estimate of $\widehat{\eta}_\beta$. Consequently, the stopping criteria (line 15) of the 2DRQI (Algorithm 4.1) should be

$$|\operatorname{Imag}(x_{k,1}^H x_{k,2})| \le \mathtt{tol} \quad \text{and} \quad \eta_2(\mu_k,\lambda_k,x_k) \le \mathtt{tol}, \tag{6.23}$$

where `tol` is a prescribed tolerance value. In addition, to handle the possible stagnation of the 2DRQI, we can also include the following test into the stopping criterion for possible stagnation:

$$\eta_2(\mu_k,\lambda_k,x_k) \ge \frac{1}{2}\Big(\eta_2(\mu_{k-2},\lambda_{k-2},x_{k-2}) + \eta_2(\mu_{k-1},\lambda_{k-1},x_{k-1})\Big). \tag{6.24}$$

(4) For the optional validation step of Algorithm 6.2, we know that if the computed $\widehat{\lambda}$ is an acceptable estimate of DTI $\beta(\widehat{A})$, it should satisfy

$$(1-\texttt{reltol})\widehat{\lambda} \le \beta(\widehat{A}) \le \widehat{\lambda} \tag{6.25}$$

for a small `reltol`, where, without loss of generality, we assume $\widehat{\lambda} > 0$. Otherwise, according to the symmetric properties of 2D-eigenvalues in DTI, we can use $-\widehat{\lambda}$ as an estimate of the DTI $\beta(\widehat{A})$.
The upper bound of (6.25) naturally holds according to (6.14) and $(\widehat{\mu}, \widehat{\lambda})$ is a 2D-eigenvalue. For the lower bound of (6.25), we just need to verify that $H((1-\texttt{reltol})\widehat{\lambda})$ has no imaginary eigenvalue. This is based on the following lemma.

LEMMA 6.3 (see [3]). *For any $\lambda > 0$, $\lambda < \beta(\widehat{A})$ if and only if $G(\lambda)$ has no pure imaginary eigenvalue, where $G(\lambda)$ is a Hamiltonian matrix of the form*

$$G(\lambda) = \begin{bmatrix} \widehat{A} & -\lambda I \\ \lambda I & -\widehat{A}^H \end{bmatrix}. \tag{6.26}$$

This validation procedure is the one proposed in [11]. However, it should be noted that checking whether $G((1-\texttt{reltol})\widehat{\lambda})$ has no imaginary eigenvalues could be prohibitively expensive for large scale problems. Therefore, the validation step is optional in all existing algorithms for computing DTI [11, 16, 18].

In section 7, we will provide a numerical example to compare the performance of the 2DRQI and a recently proposed subspace method for computing the DTI.

7. Numerical examples. In this section, we first present a numerical example to illustrate the convergence behaviors of the 2DRQI (Algorithm 4.1) and then present three examples for finding the minmax of two Rayleigh quotients (Algorithm 6.1) and for computing the DTI (Algorithm 6.2). All algorithms are implemented in MATLAB 2016b. Numerical experiments are performed on an HP computer with an Intel(R) Core(TM) 2.60 GHz i7-6700HQ CPU and 8 GB RAM.

Example 2. This example illustrates convergence behaviors of the 2DRQI (Algorithm 4.1). Let us consider the 2DEVP (1.1) of the matrices

$$A = \begin{bmatrix} -0.7 & 0.01 & 0.2 \\ 0.01 & 2 & 0 \\ 0.2 & 0 & 0 \end{bmatrix} \quad \text{and} \quad C = \begin{bmatrix} 0.3 & 0.01 & 0.2 \\ 0.01 & 1 & 0 \\ 0.2 & 0 & -1 \end{bmatrix}.$$

It can be verified that $(\mu_1, \lambda_1, x_1) = (1, 1, \begin{bmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix})$ is a 2D-eigentriplet and $\lambda_1 = 1$ is an eigenvalue of $A - \mu_1 C$ with multiplicity 2. In addition, by a brute-force bisection search following the sorted eigencurves $\lambda_1(\mu) \ge \lambda_2(\mu) \ge \lambda_3(\mu)$ of $A - \mu C$ on the interval $[-1.5, 1.5]$, we find additional two 2D-eigenvalues to machine precision:

$$(\mu_2, \lambda_2) = (-0.665101440190437, -0.239801782612878),$$
$$(\mu_3, \lambda_3) = (-0.145810069397438, -0.744080780565709).$$

Moreover, λ_2 and λ_3 are the simple eigenvalues of $A - \mu_2 C$ and $A - \mu_3 C$, respectively. The left plot of Figure 3 depicts the sorted eigencurves $\lambda_j(\mu)$ for $j = 1, 2, 3$.

The maximum 2D-eigenvalue $(\mu_1, \lambda_1) = (1, 1)$ is marked in red. The 2D-eigenvalue (μ_2, λ_2) is blue. The minimum 2D-eigenvalue (μ_3, λ_3) is green.

We use each grid point on the 100×100 mesh of the domain $(\mu, \lambda) = [-1.5, 1.5] \times [-2, 2]$ as an initial (μ_0, λ_0) and the vector x_0 is generated based on the remarks of Algorithm 4.1. If the 2DRQI with (μ_0, λ_0, x_0), $\texttt{tol} = n \cdot \texttt{macheps}$, and maxit $= 15$ converges to the ith 2D-eigenvalue (μ_i, λ_i), then we use the same color for the initial (μ_0, λ_0) and (μ_i, λ_i). The right plot of Figure 3 shows that the 2DRQI converges to a 2D-eigentriplet for all 10,000 initials (μ_0, λ_0, x_0).

Table 1 records the convergence history of a sequence $\{(\mu_{3;k}, \lambda_{3;k}, x_{3;k})\}$ to the minimum 2D-eigenvalue (μ_3, λ_3), marked in green in Figure 3. We observe that the sequence $\{(\mu_{3;k}, \lambda_{3;k})\}$ converges quadratically, the matrix C_k of the 2DRQ (A_k, C_k) remains indefinite, and $a_{12,k} \neq 0$. Table 2 shows the convergence history of a sequence $\{(\mu_{1;k}, \lambda_{1;k}, x_{1;k})\}$ to the maximum 2D-eigenvalue (μ_1, λ_1), marked in red in Figure 3. Note that λ_1 is an eigenvalue of $A - \mu_1 C$ with multiplicity 2. We observe that the sequence $\{\mu_{1;k}, \lambda_{1;k}\}$ converges quadratically and the matrix C_k of the 2DRQ (A_k, C_k) remains indefinite. However, $a_{12,k}$ approaches to 0.

For the convergence analysis of the 2DRQI presented in [30], we can see that although the algorithm and local quadratic convergence rate are the same regardless of the multiplicity of the eigenvalue λ_* of $A - \mu_* C$, the convergence analysis needs to be treated differently as indicated by whether $|a_{12,k}|$ approaches to 0.

Example 3. We use Algorithm 6.1 to solve the RQminmax (6.1) arising from a MIMO relay precoder design problem in signal communication to minimize the total

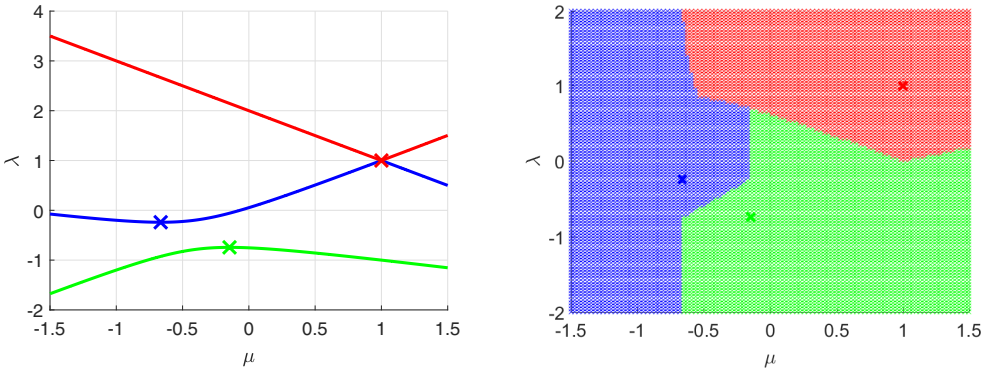


FIG. 3. *Left: Sorted eigencurves and corresponding 2D-eigenvalues of* (A, C) *in Example* 2. *Right: Computed 2D-eigenvalues with different initials. (Color available online.)*

TABLE 1
Convergence history of $\{(\mu_{3;k}, \lambda_{3;k}, x_{3;k})\}$ *to* (μ_3, λ_3, x_3).

k	$\lvert\mu_{3;k} - \mu_3\rvert$	$\lvert\lambda_{3;k} - \lambda_3\rvert$	$\eta_1(\mu_{3;k}, \lambda_{3;k}, x_{3;k})$	$(c_{1,k}, c_{2,k})$	$\lvert a_{12,k}\rvert$
0	1.6e0	8.9e-1	4.1e-1	(-1.0e0, 3.3e-1)	2.9e-1
1	2.6e-3	8.4e-3	7.1e-2	(-1.0e0, 3.3e-1)	2.9e-1
2	2.2e-5	1.2e-7	2.7e-4	(-1.0e0, 3.3e-1)	2.9e-1
3	6.5e-13	1.1e-16	2.1e-9	(-1.0e0, 3.3e-1)	2.9e-1
4	3.4e-16	2.6e-16	1.1e-16	(-1.0e0, 3.3e-1)	2.9e-1

TABLE 2
Convergence history for $\{(\mu_{1;k}, \lambda_{1;k}, x_{1;k})\}$ *to* (μ_1, λ_1, x_1).

k	$\lvert\mu_{1;k} - \mu_1\rvert$	$\lvert\lambda_{1;k} - \lambda_1\rvert$	$\eta_1(\mu_{1;k}, \lambda_{1;k}, x_{1;k})$	$(c_{1,k}, c_{2,k})$	$\lvert a_{12,k}\rvert$
0	1.0e0	1.0e0	3.2e-1	(-1.0e0, 7.9e-1)	9.3e-2
1	3.3e-1	4.6e-1	3.1e-1	(-9.6e-1, 9.6e-1)	3.2e-2
2	5.0e-2	9.0e-2	1.3e-1	(-1.0e0, 1.0e0)	2.4e-4
3	5.2e-4	3.3e-4	8.1e-3	(-1.0e0, 1.0e0)	3.2e-9
4	3.8e-10	2.2e-11	2.1e-6	(-1.0e0, 1.0e0)	8.0e-16
5	4.2e-16	2.2e-16	2.5e-16	(-1.0e0, 1.0e0)	4.6e-16

relay power subject to SINR constraints at the receivers [4]. Consider the multipoint to multipoint communication with two sources. The signals r_o after MIMO relay processing and signals y received by destinations are

$$r_o = ZH_{\text{up}}s + Zn_r \quad \text{and} \quad y = H_{\text{dl}}^H ZH_{\text{up}}x + H_{\text{dl}}^H Zn_r + n_d,$$

where s is the transmit signals of the sources, and n_r and n_d are zero-mean circularly symmetric complex Gaussian random variables with variance σ_r^2 and σ_d^2. $H_{\text{up}} = [h_1, h_2] \in \mathbb{C}^{m\times 2}$ denotes channels between two sources and antennas, $H_{\text{dl}} = [g_1, g_2] \in \mathbb{C}^{m\times 2}$ denotes channels between antennas and two destinations, and m is the number of antennas at the relay. $Z \in \mathbb{C}^{m\times m}$ is the processing matrix to be designed. Under the assumption that the source transmit signals s are zero-mean and statistically independent with the unit power, the goal of the MIMO relay precoder design is to minimize the relay power while maintaining SINR no less than a prescribed threshold γ_{th}.

After some algebraic manipulations, the MIMO precoder relay design problem becomes solving the following homogeneous quadratic constrained programming (HQCQP) problem:

$$\min_u u^H Tu \quad \text{s.t.} \quad u^H P_j u + 1 \le 0 \quad \text{for} \quad j = 1, 2, \tag{7.1}$$

where $u = \text{vec}(Z)$ is a column vector obtained by stacking the columns of Z on top of one another. Additionally, above $T = \widehat{F}_0 \otimes I$, $P_1 = \widehat{F}_1 \otimes g_1 g_1^H$ and $P_2 = \widehat{F}_2 \otimes g_2 g_2^H$ are of dimensions $n = m^2$, with $\widehat{F}_0 = \overline{h}_1 h_1^T + \overline{h}_2 h_2^T + \sigma_r^2 I$, $\widehat{F}_1 = \frac{1}{\gamma_{\text{th}}\sigma_d^2}(\gamma_{\text{th}}\overline{h}_2 h_2^T + \gamma_{\text{th}}\sigma_r^2 I - \overline{h}_1 h_1^T)/(\gamma_{\text{th}}\sigma_d^2)$, and $\widehat{F}_2 = \frac{1}{\gamma_{\text{th}}\sigma_d^2}\left(\gamma_{\text{th}}\overline{h}_1 h_1^T + \gamma_{\text{th}}\sigma_r^2 I - \overline{h}_2 h_2^T\right)/(\gamma_{\text{th}}\sigma_d^2)$. The operator $\otimes$ is the Kronecker product. Note that $\widehat{F}_0$ and $\widehat{F}_i$ are $m \times m$ Hermitian matrices with $\widehat{F}_0$ positive definite. Gaurav and Hari [12] show that the HQCQP (7.1) is equivalent to the RQminmax (6.1) of the matrices

$$A = S^H P_1 S = F_1 \otimes g_1 g_1^H, \quad B = S^H P_2 S = F_2 \otimes g_2 g_2^H, \tag{7.2}$$

where $S = T^{-\frac{1}{2}}$ is the square root of T^{-1}, $F_1 = \widehat{F}_0^{-\frac{1}{2}} \widehat{F}_1 \widehat{F}_0^{-\frac{1}{2}}$, and $F_2 = \widehat{F}_0^{-\frac{1}{2}} \widehat{F}_2 \widehat{F}_0^{-\frac{1}{2}}$. We note that by exploiting the structure of A and B, the matrix-vector multiplications Ax and Bx can be performed efficiently. For numerical experiments described in [12], H_{up} and H_{dl} are complex Gaussian random matrices. The SINR is set to 3 dB and noise variances are set to -10 dB, i.e., $\gamma_{\text{th}} = 10^{\frac{3}{10}}$, and $\sigma_d^2 = \sigma_r^2 = 10^{-1}$.

Algorithm 6.1 first checks cases I and II of the RQminmax (6.1) described in Theorem 6.1 for possible early exit. Then it uses a combination of the 2DRQI and the bisection search to find an optimizer $\mu_*^{(\text{RQI})}$ of the EVOPT (6.2) for the general case III.

A dichotomous method is proposed in [12] for solving the EVOPT (6.2). Starting from a search interval $[a, b]$ containing the global maximum of the concave function

$g(\mu)=\lambda_{\min}(A-\mu C)$, where $C=A-B$, the dichotomous method compares $g(a)$, $g(b)$, $g((a+b)/2-\epsilon_r)$, and $g((a+b)/2+\epsilon_r)$ for a small scalar ϵ_r, and then by using the concavity of $g(\mu)$, replaces a with $(a+b)/2-\epsilon_r$ or b with $(a+b)/2+\epsilon_r$ for the next iteration. When the search interval width $b-a$ is less than a prescribed tolerance `tol`, it returns an approximate optimal value $\mu_*^{(\mathrm{Dich})}=(a+b)/2$.

An alternative algorithm to solve the EVOPT (6.2) is to use the subspace method [18]. The subspace method solves an eigenvalue optimization problem by successively projecting the original problem onto a subspace. Specifically, for the EVOPT (6.2) with the prescribed interval $[0,1]$ and an initial $\mu_0=\frac{1}{2}$, the subspace method first computes $\lambda_{\min}(A-\mu_0 C)$ and the corresponding eigenvector v_0, and then sets the initial projection subspace $V_0=v_0$. At the kth iteration for $k\geq 1$, the subspace method projects EVOPT (6.2) onto the subspace V_{k-1} and solves the reduced problem:

$$\lambda_{\min}^{(k)}=\max_{\mu\in[0,1]}\lambda_{\min}\big(V_{k-1}^H A V_{k-1}-\mu V_{k-1}^H C V_{k-1}\big). \tag{7.3}$$

With a minimizer μ_k of the reduced problem (7.3), the subspace method computes the eigenvector v_k of $\lambda_{\min}(A-\mu_k C)$, and then updates the projection subspace $V_k=\mathrm{orth}(V_{k-1},v_k)$. It is proved [18] that $f_L^{(k)}=\max_{j=1,\cdots,k}\{\lambda_{\min}(A-\mu_j C)\}$ is a lower bound of the optimal value, while $f_U^{(k)}=\min_{j=1,\cdots,k}\{\lambda_{\min}^{(j)}\}$ is an upper bound, and $f_U^{(k)}-f_L^{(k)}$ will tend to 0, i.e., the iteration converges.

`leigopt` is an implementation of the subspace method in MATLAB [18].[1] It uses `eigopt`, a quadratic supporting functions based method [33], to solve the reduced problem, and terminates the iteration when $|\lambda_{\min}^{(k-1)}-\lambda_{\min}^{(k)}|<$ `tol` for a prescribed `tol` or the number of iterations exceeds $\sqrt{n}$. We note that there are two minor modifications here. First, to improve computational efficiency of `leigopt`, we set the dimension of the projection subspace `opts.p = 20` in `eigs`, instead of `round(sqrt(`n`))` used in `leigopt`. We have tried to use the interior point method to substitute for `eigopt`. Numerical experiments show the interior point method is slower on the dimensionalities shown in Table 3. Second, we observe that the stopping criterion often leads to early termination of `leigopt` and thus fails to obtain an accurate solution. To solve this problem, we use a more robust criterion, $|f_U^{(k)}-f_L^{(k)}|<$ `tol`. We denote the returned value as $\mu_*^{(\mathrm{leig})}$.

We observed that the optimizers of the EVOPT (6.2) on the interval $[0,1]$ computed by the dichotomous method (`tol = 1e-8`), `leigopt (tol = 1e-10)`, and Algorithm 6.1 (`backtol` $=n\epsilon$ and `reltol = 1e-8`) agree up to 8 significant digits for 20 runs of each of dimensions $n=10^2,100^2,200^2,400^2$.

Table 3 records the average numbers of iterations and running time (in seconds) of 100 runs of three methods. The runtime of the subspace method is written as

TABLE 3

Performance of the dichotomous method, the subspace method, and Algorithm 6.1 *for solving the EVOPT* (6.2).

	Dichotomous method		Subspace method		Algorithm 6.1	
$n=m^2$	niter	runtime	niter	runtime	niter	runtime
10^2	15	0.11	5	0.083(0.063)	3.1	0.026
100^2	15	1.2	5	0.27(0.068)	2.6	0.19
200^2	15	4.6	5	0.86(0.069)	2.4	0.57
400^2	15	29	5	5.2(0.069)	2.1	3.6

[1] http://home.ku.edu.tr/∼emengi/software/leigopt, downloaded on October 2, 2021. The codes have been migrated to https://mysite.ku.edu.tr/emengi/leigopt.

$t_{\text{all}}(t_{\text{sub}})$ with t_{all} for the total time spent on solving the EVOPT (6.2) and t_{sub} for the time spent solving the subproblems (7.3). The significant performance gain of Algorithm 6.1 in speed compared to the dichotomous method and the subspace method is due to the smaller number of iterations and the fact that each iteration of the dichotomous method needs to solve two eigenvalue problems of $A - \mu_j C$ for computing $g(\mu_j) = \lambda_{\min}(A - \mu_j C)$, where we use the sparse eigensolver `eigs`. In contrast, each iteration of Algorithm 6.1 calls the 2DRQI (Algorithm 4.1) once, which in turn only needs to solve the augmented linear system (4.4), where we use the linear solver `gmres` with the tolerance $n\epsilon$ and the maximum Krylov subspace dimension 30 and without preconditioning.

Example 4. In this example, we consider the computation of DTI for matrices from the finite difference discretization of the Orr–Sommerfeld operator for planar Poiseuille flow. An $n \times n$ Orr–Sommerfeld matrix is of the form[2]

$$\widehat{A}_n = L_n^{-1} B_n,$$

where $L_n = (1/h^2)\text{tridiag}(1, -(2+h^2), 1)$, $B_n = \frac{1}{\mathcal{R}_e} L_n^2 - \mathtt{i}(U_n L_n + 2I)$, and $U_n = \text{diag}(1-u_1^2, \ldots, 1-u_n^2)$. $h = 2/(n+1)$ is the stepsize of discretization, $u_k = -1 + kh$, $\mathcal{R}_e$ is the Reynolds number ($\mathcal{R}_e = 1000$ in numerical experiments), and $\mathtt{i} = \sqrt{-1}$. The stability of the Orr–Sommerfeld matrices has been extensively studied [8, 31, 44]. It is known that the eigenvalues of Orr–Sommerfeld matrices are highly sensitive to perturbations. The DTI is an important measure of the stability under uncertainty [11, 16, 18].

For efficiently solving the linear equation (4.4) in Algorithm 6.2, we first reorder the Jacobian $J(\mu_k, \lambda_k, x_k)$ to a banded arrow matrix [5, p. 86] and then apply a Schur complement technique [34, p. 406]. For the initial (μ_0, λ_0, x_0) of the 2DRQI, we apply the Cayley–Arnoldi algorithm with a complex shift for computing μ_0 [32] and then use the MATLAB function `svds` to compute the smallest singular triplet of $\widehat{A} - \mu_0 \mathtt{i} I$ with $\mathtt{tol} = n\epsilon$.

We also apply the subspace method and its implementation `leigopt` [18] discussed in Example 3. In this case, for computing DTI $\beta(\widehat{A}_n)$, `leigopt` solves the singular value minimization

$$\beta(\widehat{A}_n) = \min_{\mu \in \mathbb{R}} \sigma_{\min}(\widehat{A}_n - \mu \mathtt{i} I). \tag{7.4}$$

With a prescribed search interval $[a, b]$ and an initial $\mu_0 \in [a, b]$, `leigopt` first computes $\sigma_{\min}(\widehat{A}_n - \mu_0 \mathtt{i} I)$ and the corresponding right singular vector v_0 and then sets the initial projection subspace $V_0 = v_0$. At the kth iteration for $k \geq 1$, `leigopt` projects the minimization (7.4) onto the subspace V_{k-1} and solves the reduced problem:

$$\sigma_{\min}^{(k)} = \min_{\mu \in [a,b]} \sigma_{\min}(\widehat{A}_n V_{k-1} - \mu \mathtt{i} V_{k-1}). \tag{7.5}$$

With a minimizer μ_k of the reduced problem (7.5), the subspace method computes $\sigma_{\min}(\widehat{A}_n - \mu_k \mathtt{i} I)$ and the corresponding right singular vector v_k and then updates the projection subspace $V_k = \text{Orth}(v_{k-1}, v_k)$. The iteration terminates when $\sigma_{\min}^{(k-1)} - \sigma_{\min}^{(k)} < \mathtt{tol}$ for a prescribed `tol`, or the number of iterations exceeds $\sqrt{n}$. For numerical experiments, the initial $\mu_0 = 0$ and the tolerance $\mathtt{tol} = \mathtt{1e\text{-}12}$.

[2]The formulation in [16, 18] has some typos.

TABLE 4
DTI computation by the subspace method and Algorithm 6.2.

	The subspace method			Algorithm 6.2		
n	niter	runtime	$\widehat{\beta}(\widehat{A}_n)$	niter	runtime	$\widehat{\beta}(\widehat{A}_n)$
1000	9.4	0.14(0.013)	1.97789572876e-3	5.8	$0.025+0.032$	1.9778957275e-3
4000	9.4	0.42(0.029)	1.97809674700e-3	4.9	$0.062+0.095$	1.9780964583e-3
16000	8.5	1.42(0.069)	1.93794289874e-3	4.8	$0.25\ +0.38$	1.9376706543e-3

We note that to improve computational efficiency, the following minor modifications are made in `leigopt`. (1) We set the dimension of the projection subspace `opts.p = 20` in `eigs` or `svds`, instead of `round(sqrt(`n`))` used in `leigopt`. (2) `leigopt` uses `eigopt`, a quadratic supporting functions based method [33], to solve the reduced problem (7.5). For the Orr–Sommerfeld matrices, `eigopt` is too time consuming. Instead, we use a modified Boyd–Balakrishnan method [2]. As a byproduct, the search interval $[a, b]$ does not need to be prescribed with this method.[3] (3) We keep all historic right singular vectors, i.e., $v_0, v_1, \ldots, v_k$, in the projection subspace V_k. This slightly decreases the number of iterative steps and reduces the total computational time.

Table 4 shows the performance of Algorithm 6.2 and `leigopt`. The runtime of Algorithm 6.2 is written as $t_1 + t_2$ with t_1 for calculating the rightmost eigenvalue of $\widehat{A}_n$ and the singular triplet of $\widehat{A}_n - \mu_0 \mathtt{i} I$ (i.e., lines 1 and 2 of Algorithm 6.2) and t_2 for the rest of calculation. The runtime of the subspace method is written as $t_{\text{all}}(t_{\text{sub}})$ with t_{all} for the total time and t_{sub} for solving the subproblems (7.5). We observe that the $\widehat{\beta}(\widehat{A}_n)$ computed by the two algorithms agree from 4 to 8 significant digits. However, Algorithm 6.2 uses no more than half of the runtime of the subspace method. The reason for the speedup of Algorithm 6.2 is twofold. Algorithm 6.2 uses fewer iterative steps. The major cost of the subspace method is on computing the right singular vector v_k corresponding to $\sigma_{\min}(\widehat{A} - \mu_k \mathtt{i} I)$. In contrast, in Algorithm 6.2, we only need to solve a linear equation of the form (4.4) in each iteration of 2DRQI (Algorithm 4.1).

We note that the validation step for computed $\widehat{\beta}(\widehat{A}_n)$ by Algorithm 6.2 and the subspace method is not reported in Table 4. For the matrix size $n = 1000$, it is verified that both algorithms pass the validation procedure described in subsection 6.2 with reltol = 1e-9. Although there exists an algorithm [23] for checking whether $G(\lambda)$ defined in Lemma 6.3 has pure imaginary eigenvalues, it would be too expensive for large matrix sizes. As is common practice of the existing algorithms [11, 16, 18, 52], there is no validation procedure for large scale DTI calculation.

Example 5. In this example, we present the performance of the 2DRQI-based Algorithm 6.2 and the subspace method based `leigopt` on the computation of the DTI of test matrices depicted in Table 5. These matrices come from DTI related literature [15, 25, 26]. It should be noted that some of the test matrices are unstable ones. As is common practice, for those unstable matrices A, a shift $\sigma \in \mathbb{R}$ is introduced so that $\widehat{A} - \sigma i I$ is a stable matrix.

Similar to Example 4, we make the following modifications to improve the efficiency of `leigopt`. First, we set the dimension of the projection subspace `opts.p = 20` in `eigs` or `svds`, instead of `round(sqrt(`n`))` used in `leigopt`. Second, we use the

[3] This strategy is also recommended by Dr. Mengi, one of the authors of `leigopt`, in a private communication.

TABLE 5
Statistics of the test matrices.

Matrix	n	Sparsity	Shift (σ)	Frobenius norm	Source
olmstead1000	1000	0.003996	5	1.26e + 06	[15, 25]
dwave2048	2048	0.002410	1	3.02e + 01	[15, 25]
pde2961	2961	0.001660	10	3.70e + 02	[15, 25]
rdbrusselator3200	3200	0.001840	1	2.81e + 03	[15, 25]
HF2D9	3481	0.001420	1	9.51e + 03	[26]
tols4000	4000	0.000549	0	2.98e + 08	[15, 25]
HF2D_CD3	4096	0.001210	2	1.18e + 05	[26]
markov5050	5050	0.000974	2	1.48e + 02	[15, 25]
sparserandom10000	10000	0.001200	3	3.06e + 02	[15]
skewlap3d30	24389	0.000279	0	9.24e + 05	[15, 25]

TABLE 6
Performance of Algorithm 6.2 *and leigopt on the set of matrices depicted in Table* 5.

	Algorithm 6.2			leigopt		
Matrix	niter	$\beta(\widehat{A})$	Timing	niter	$\beta(\widehat{A})$	Timing
olmstead1000	0	4.740742924e-1	0.034	3	4.740742924e-1	0.053
dwave2048	0	2.119727657e-2	0.098	3	2.119727657e-2	0.122
pde2961	3	2.267878235e-2	0.58	6	2.267878235e-2	0.66
rdbrusselator3200	3	3.594599642e-1	0.72	4	3.594599642e-1	0.74
HF2D9	0	7.192219953e-1	0.20	3	7.192219953e-1	0.25
tols4000	1	1.999796888e-3	0.60	4	1.999796837e-3	0.64
HF2D_CD3	0	1.913018555e-2	0.27	3	1.913018555e-2	0.36
markov5050	1	9.263768378e-1	0.86	5	9.263768378e-1	0.59
sparserandom10000	0	1.273786238e-5	31.0	3	1.273786238e-5	60.9
skewlap3d30	0	8.729075984e + 1	9.8	3	8.729075984e + 1	13.7

modified Boyd–Balakrishnan algorithm to substitute for `eigopt`. Note that `leigopt` needs an initial μ_0 to start. In the Orr–Sommerfeld class of matrices in Example 4, μ_0 is prescribed. However, for the test matrices in Table 5, we do not have such prior information. Therefore, we calculate the rightmost eigenvalue of $\widehat{A}$ using the MATLAB `eigs` function except for the matrices `tols4000` and `olmstead1000`, and use its imaginary part as the initial μ_0. The cost for calculating the initial μ_0 is counted in the total cost of the running time. For the matrices `tols4000` and `olmstead1000`, since `eigs` failed to find the rightmost eigenvalue, we use the Cayley–Arnoldi algorithm with a complex shift. When calculating the smallest triplets, we use `svds`. We set `tol` to $n\epsilon$ in Algorithm 6.2 and 10^{-12} in `leigopt` as in [18], where ϵ is the machine precision.

Table 6 summarizes the performance of Algorithm 6.2 and `leigopt` in terms of the number of iterations, computed DTI $\beta(\widehat{A})$, and total running time. We observe that all computed DTI $\beta(\widehat{A})$ by both methods match in at least 8 significant digits. Algorithm 6.2 is generally faster except for the matrix `markov5050`. For the matrix `markov5050`, it is more expensive to solve linear systems than to calculate the singular values in Algorithm 6.2. We note that for some matrices, Algorithm 6.2 takes 0 iterations. This is due to the fact that for these matrices, the initial values already pass the backward error test, while `leigopt` does not equip such a backward error test. To end this example, we highlight that the 2DRQI is a local method and is sensitive to the choice of the initial approximation. For example, if the initial approximation μ_0 is set to 10 for the matrix `markov5050`, Algorithm 6.2 fails to converge to the correct DTI $\beta(\widehat{A})$ while `leigopt` succeeds.

8. Conclusion. We introduced the 2DEVP (1.1) of a Hermitian matrix pair (A, C). We investigated the relationships between the well-known eigenvalue optimization problem of the parameter matrix $H(\mu) = A - \mu C$ and the 2DEVP. We presented essential properties of the 2DEVP such as the existence and variational characterizations of 2D-eigenvalues. We devised an RQI-like algorithm, 2DRQI, for solving the 2DEVP. The computational kernel of the 2DRQI involves the solution of linear systems of equations. The efficiency of the 2DRQI is demonstrated for solving the large scale 2DEVP arising from the minmax problem of two Rayleigh quotients and the computation of the distance to instability of a stable matrix. A rigorous convergence analysis of the proposed 2DRQI is presented in [30].

The 2DRQI Algorithm 4.1 is designed to compute a stationary point of the eigencurve $\lambda_j(\mu)$ of $H(\mu)$ for some j. Since the 2DRQI is a local algorithm, there is a lack of control on j other than the initialization. It is a subject of further study on how to combine the 2DRQI with other global search schemes so that it is guaranteed to compute a stationary point of $\lambda_j(\mu)$ for a prescribed j.

Acknowledgments. The authors are grateful to Ding Lu for valuable discussions during the course of this work. They would like to thank the anonymous referees, whose comments helped to significantly improve the quality of the paper.

REFERENCES

[1] E. K. Blum and A. F. Chang, *A numerical method for the solution of double eigenvalue problem*, J. Inst. Math. Appl., 22 (1978), pp. 29–42, https://doi.org/10.1093/imamat/22.1.29.

[2] S. Boyd and V. Balakrishnan, *A regularity result for the singular values of a transfer matrix and a quadratically convergent algorithm for computing its L_∞ norm*, Systems Control Lett., 15 (1990), pp. 1–7, https://doi.org/10.1109/CDC.1989.70267.

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

[4] B. Chalise, L. Vandendorpe, and J. Louveaux, *MIMO relaying for multi-point to multi-point communication in wireless networks*, in IEEE CAMPSAP, St. Thomas, VI, USA, 2007, pp. 217–220, https://doi.org/10.1109/CAMSAP.2007.4498004.

[5] K. Chen, *Matrix Preconditioning Techniques and Applications*, Cambridge University Press, Cambridge, 2005, https://doi.org/10.1017/CBO9780511543258.

[6] F. H. Clarke, *Optimization and Nonsmooth Analysis*, Classics Appl. Math. 5, SIAM, Philadelphia, 1990, https://doi.org/10.1137/1.9781611971309.

[7] J. W. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997, https://doi.org/10.1137/1.9781611971446.

[8] P. G. Drazin and W. H. Reid, *Hydrodynamic Stability*, 2nd ed., Cambridge University Press, Cambridge, 2004, https://doi.org/10.1017/CBO9780511616938.

[9] K. Fan, *On a theorem of Weyl concerning eigenvalues of linear transformations* I, Proc. Natl. Acad. Sci. USA, 35 (1949), pp. 652–655, https://doi.org/10.1073/pnas.35.11.652.

[10] M. K. H. Fan and B. Nekooie, *On minimizing the largest eigenvalue of a symmetric matrix*, Linear Algebra Appl., 214 (1995), pp. 225–246, https://doi.org/10.1016/0024-3795(93)00068-B.

[11] M. A. Freitag and A. Spence, *A Newton-based method for the calculation of the distance to instability*, Linear Algebra Appl., 435 (2011), pp. 3189–3205, https://doi.org/10.1016/j.laa.2011.06.012.

[12] D. D. Gaurav and K. V. S. Hari, *A fast eigen solution for homogeneous quadratic minimization with at most three constraints*, IEEE Signal Process. Lett., 20 (2013), pp. 968–971, https://doi.org/10.1109/LSP.2013.2276791.

[13] A. B. Gershman, N. D. Sidiropoulos, S. Shahbazpanahi, M. Bengtsson, and B. Ottersten, *Convex optimization-based beamforming*, IEEE Signal Process. Mag., 27 (2010), pp. 62–75, https://doi.org/10.1109/MSP.2010.936015.

[14] I. Gohberg, P. Lancaster, and L. Rodman, *Matrix Polynomials*, Classics Appl. Math. 58, SIAM, Philadelphia, 2009, https://doi.org/10.1137/1.9780898719024.

[15] 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 (2011), pp. 1166–1192, https://doi.org/10.1137/100817048.
[16] C. He and G. A. Watson, *An algorithm for computing the distance to instability*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 101–116, https://doi.org/10.1137/S0895479897314838.
[17] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd ed., Cambridge University Press, 2012.
[18] F. Kangal, K. Meerbergen, E. Mengi, and W. Michiels, *A subspace method for large-scale eigenvalue optimization*, SIAM J. Matrix Anal. Appl., 39 (2018), pp. 48–82, https://doi.org/10.1137/16M1070025.
[19] E. Karipidis, N. D. Sidiropoulos, and Z. Luo, *Far-field multicast beamforming for uniform linear antenna arrays*, IEEE Trans. Signal Process., 55 (2007), pp. 4916–4927, https://doi.org/10.1109/TSP.2007.897903.
[20] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, Frontiers Appl. Math. 16, SIAM, Philadelphia, 1995, https://doi.org/10.1137/1.9781611970944.
[21] V. B. Khazanov, *Methods for solving spectral problems for multiparameter matrix pencils*, J. Math. Sci., 127 (2005), pp. 2033–2050, https://doi.org/10.1007/s10958-005-0161-8.
[22] K. Knopp, *Theory of Functions (Part* I*)*, Dover, New York, 1947.
[23] D. Kressner, *Finding the distance to instability of a large sparse matrix*, in Proceedings of the IEEE International Symposium on Intelligent Control, Munich, 2006, pp. 31–35, https://doi.org/10.1109/CACSD-CCA-ISIC.2006.4776620.
[24] D. Kressner, D. Lu, and B. Vandereycken, *Subspace acceleration for the Crawford number and related eigenvalue optimization problems*, SIAM J. Matrix Anal. Appl., 39 (2018), pp. 961–982, https://doi.org/10.1137/17M1127545.
[25] D. Kressner and B. Vandereycken, *Subspace methods for computing the pseudospectral abscissa and the stability radius*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 292–313, https://doi.org/10.1137/120869432.
[26] F. Leibfritz, *Compleib: Constrained Matrix Optimization Problem Library*, 2006, http://www.compleib.de/.
[27] A. S. Lewis, *Derivatives of spectral functions*, Math. Oper. Res., 21 (1996), pp. 576–588, https://doi.org/10.1287/moor.21.3.576.
[28] A. S. Lewis and M. L. Overton, *Eigenvalue optimization*, Acta Numer., 5 (1996), pp. 149–190, https://doi.org/10.1017/S0962492900002646.
[29] T. Lu and Y. Su, *A Newton-type method for two-dimensional eigenvalue problems*, Numer. Linear Algebra Appl., 29 (2022), e2430, https://doi.org/10.1002/nla.2430.
[30] T. Lu, Y. Su, and Z. Bai, *2D Eigenvalue Problem* III*: Convergence Analysis of the 2D Rayleigh Quotient Iteration*, https://doi.org/10.48550/arXiv.2303.05357, 2023.
[31] A. N. MalysheV and M. Sadkane, *On the stability of large matrices*, J. Comput. Appl. Math., 102 (1999), pp. 303–313, https://doi.org/10.1016/S0377-0427(98)00231-3.
[32] K. Meerbergen and D. Roose, *Matrix transformations for computing rightmost eigenvalues of large sparse non-symmetric eigenvalue problems*, IMA J. Numer. Anal., 16 (1996), pp. 297–346, https://doi.org/10.1093/imanum/16.3.297.
[33] E. Mengi, E. A. Yildirim, and M. Kiliç, *Numerical optimization of eigenvalues of Hermitian matrix functions*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 699–724, https://doi.org/10.1137/130933472.
[34] J. Nocedal and S. Wright, *Numerical Optimization*, Springer-Verlag, New York, 1999.
[35] M. L. Overton, *On minimizing the maximum eigenvalue of a symmetric matrix*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 256–268, https://doi.org/10.1137/0609021.
[36] M. L. Overton, *Large-scale optimization of eigenvalues*, SIAM J. Optim., 2 (1992), pp. 88–120, https://doi.org/10.1137/0802007.
[37] M. L. Overton and R. S. Womersley, *Second derivatives for optimizing eigenvalues of symmetric matrices*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 697–718, https://doi.org/10.1137/S089547989324598X.
[38] V. Pan, D. Ivolgin, B. Murphy, R. E. Rosholt, Y. Tang, and X. Yan, *Additive preconditioning for matrix computations*, Linear Algebra Appl., 432 (2010), pp. 1070–1089, https://doi.org/10.1016/j.laa.2009.10.020.
[39] V. Pan and X. Yan, *Additive preconditioning, eigenspaces, and the inverse iteration*, Linear Algebra Appl., 430 (2009), pp. 186–203, https://doi.org/10.1016/j.laa.2008.07.006.
[40] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Classics Appl. Math. 20, SIAM, Philadelphia, 1998, https://doi.org/10.1137/1.9781611971163.
[41] G. Peters and J. H. Wilkinson, *Inverse iteration, ill-conditioned equations and Newton's method*, SIAM Rev., 21 (1979), pp. 339–360, https://doi.org/10.1137/1021052.

[42] E. Polak and Y. Wardi, *Nondifferentiable optimization algorithm for designing control systems having singular value inequalities*, Automatica J. IFAC, 18 (1982), pp. 267–283, https://doi.org/10.1016/0005-1098(82)90087-5.
[43] I. Pólik and T. Terlaky, *A survey of the S-lemma*, SIAM Rev., 49 (2007), pp. 371–418, https://doi.org/10.1137/S003614450444614X.
[44] S. C. Reddy, P. J. Schmid, and D. S. Henningson, *Pseudospectra of the Orr–Sommerfeld operator*, SIAM J. Appl. Math., 53 (1993), pp. 15–47, https://doi.org/10.1137/0153002.
[45] F. Rellich, *Perturbation Theory of Eigenvalue Problems*, Gordon and Breach Science Publishers, New York, 1969.
[46] C. Roos, T. Terlaky, A. Nemirovski, and K. Roos, *On maximization of quadratic form over intersection of ellipsoids with common center*, Math. Program., 86 (1999), pp. 463–473, https://doi.org/10.1007/s101070050100.
[47] J. Sifuentes, Z. Gimbutas, and L. Greengard, *Randomized methods for rank-deficient linear systems*, Electron. Trans. Numer. Anal., 44 (2015), pp. 177–188.
[48] A. Spence and C. Poulton, *Photonic band structure calculations using nonlinear eigenvalue techniques*, J. Comput. Phys., 204 (2005), pp. 65–81, https://doi.org/10.1016/j.jcp.2004.09.016.
[49] G. W. Stewart, *Matrix Algorithms, Vol.* II*: Eigensystems*, SIAM, Philadelphia, 2001, https://doi.org/10.1137/1.9780898718058.
[50] R. A. Tapia, J. E. Dennis, Jr., and J. P. Schäfermeyer, *Inverse, shifted inverse, and Rayleigh quotient iteration as Newton's method*, SIAM Rev., 60 (2018), pp. 3–55, https://doi.org/10.1137/15M1049956.
[51] L. Trefethen and M. Embree, *Spectra and Pseudospectra: The Behavior of Nonnormal Matrices and Operators*, Princeton University Press, Princeton, NJ, 2005.
[52] C. F. Van Loan, *How near is a stable matrix to an unstable matrix?*, in Linear Algebra and Its Role in Systems Theory, Contemp. Math. 47, AMS, Providence, RI, 1985, pp. 465–478, https://doi.org/10.1090/conm/047/828319.
[53] Z.-Z. Yan and J. Guo, *Some equivalent results with Yakubovich's S-lemma*, SIAM J. Control Optim., 48 (2010), pp. 4474–4480, https://doi.org/10.1137/080744219.
[54] Y. Yuan, *On a subproblem of trust region algorithms for constrained optimization*, Math. Program., 47 (1990), pp. 53–63, https://doi.org/10.1007/BF01580852.
[55] R. Zhang, Y. Liang, C. C. Chai, and S. Cui, *Optimal beamforming for two-way multi-antenna relay channel with analogue network coding*, IEEE J. Sel. Areas Commun., 27 (2009), pp. 699–712, https://doi.org/10.1109/JSAC.2009.090611.
[56] Y. J. A. Zhang and A. M. So, *Optimal spectrum sharing in MIMO cognitive radio networks via semidefinite programming*, IEEE J. Sel. Areas Commun., 29 (2011), pp. 362–373, https://doi.org/10.1109/JSAC.2011.110209.