Journal of Computational Mathematics
Vol.41, No.6, 2023, 1117–1136.

http://www.global-sci.org/jcm
doi:10.4208/jcm.2203-m2020-0303

IMPLICIT DETERMINANT METHOD FOR SOLVING AN HERMITIAN EIGENVALUE OPTIMIZATION PROBLEM*

Siru Gong and Yangfeng Su[1)]
School of Mathematical Sciences, Fudan University, Shanghai 200433, China
Email: srgong19@fudan.edu.cn, yfsu@fudan.edu.cn

Abstract

Implicit determinant method is an effective method for some linear eigenvalue optimization problems since it solves linear systems of equations rather than eigenpairs. In this paper, we generalize the implicit determinant method to solve an Hermitian eigenvalue optimization problem for smooth case and non-smooth case. We prove that the implicit determinant method converges locally and quadratically. Numerical experiments confirm our theoretical results and illustrate the efficiency of implicit determinant method.

Mathematics subject classification: 15A18, 65F15.
Key words: Eigenvalue optimization, Multiple eigenvalue, Non-smooth optimization, Implicit determinant method, Crawford number.

1. Introduction

Let $A(\omega) \in \mathbb{C}^{n\times n}$ be an Hermitian matrix which analytically depends on a parameter $\omega \in \mathbb{R}$. Let eigenvalues of $A(\omega)$ be sorted by $\lambda_1(\omega) \geq \lambda_2(\omega) \geq \cdots \geq \lambda_n(\omega)$. In this work, for a fixed integer l, $1 \leq l \leq n$, we restrict our attention to minimize or maximize $\varphi(\omega) = \lambda_l(\omega)$ in a bounded interval (a, b), that is

$$\min_{\omega\in(a,b)} \varphi(\omega) \quad \text{or} \quad \max_{\omega\in(a,b)} \varphi(\omega). \tag{1.1}$$

We assume that there exists a local extreme point ω^* in (a, b).

Eigenvalue optimization problem (1.1) has many applications. For examples, the computation of the stable radius [24] of a stable matrix, the computation of the H_∞ norm [13] of a linear system, the computation of the Crawford number [15] and quadratic constrained quadratic programming [25] for a frequently encountered case [8], can be converted into eigenvalue optimization problem (1.1).

Many methods solve the eigenvalue optimization problem which is more general than (1.1), say the parameter ω may be in high dimensional space. For example, Overton's method bases on successive quadratic programming [21]. The method of Mengi *et al.* bases on piecewise quadratic support functions [20]. Subspace method of Kangal *et al.* solves large-scale eigenvalue optimization problem [14]. All these methods require to solve an eigenvalue problem at each iteration step.

The implicit determinant method (IDM for short) was originally proposed by Spence and Poulton [23] for solving the nonlinear eigenvalue problem $H(\omega, \lambda)x = 0$, where $H(\omega, \lambda)$ is an

* Received December 7, 2020 / Revised version received July 13, 2021 / Accepted March 8, 2022 / Published online July 28, 2022 /
[1)] Corresponding author

Hermitian matrix function with respect to the parameter ω, and λ is the eigenvalue of this nonlinear eigenvalue problem. Later, Freitag and Spence [6] applied IDM to compute the stable radius of a stable matrix. The computation of stable radius can be transformed into eigenvalue optimization problem (1.1) which has a global minimizer ω^* existing in a bounded interval $(-2\|A\|, 2\|A\|)$, with $A(\omega) = \tilde{A} - \omega\tilde{C}$ and $l = N$, where $\tilde{A}, \tilde{C} \in \mathbb{C}^{2N\times 2N}$ are Hermitian matrices [24]. In [7], Freitag *et al.* applied IDM to compute the H_∞ norm. The computation of the H_∞ norm can also be converted into eigenvalue optimization problem (1.1) which has a global maximizer ω^* existing in a bounded interval, where $\lambda_1(\omega) \geq \lambda_2(\omega) \geq \cdots \geq \lambda_m(\omega)$ are finite generalized eigenvalues of the matrix pencil $(A(\omega), \mathrm{diag}(I, \mathbf{0}))$, $A(\omega) = \tilde{A} - \omega\tilde{C}$, $l = 1$, $\tilde{A}, \tilde{C}$ are Hermitian matrices [7]. Compared with most methods for eigenvalue optimization problem, the computation cost of IDM is to solve linear systems rather than eigenvalue problem at each iteration step. However, in [6, 7], the conditions for IDM include 1. $A(\omega)$ is a linear matrix function, that is $A(\omega) = \tilde{A} - \omega\tilde{C}$, 2. $\varphi(\omega^*)$ is a simple eigenvalue of $A(\omega^*)$.

In this paper, we first generalize IDM to solve eigenvalue optimization problem (1.1) where $A(\omega)$ is a nonlinear Hermitian matrix function of ω. The generalization is almost straightforward, and the purpose is to introduce the IDM. However, the sequence generated by IDM only converges to $(\omega^*, \varphi(\omega^*))$ if $\varphi(\omega^*)$ is a simple eigenvalue of $A(\omega^*)$, which implies $\varphi(\omega)$ is smooth at ω^*. Secondly, we generalize the IDM for the case that $\varphi(\omega^*)$ is an eigenvalue of $A(\omega^*)$ with multiplicity 2, and in this case, $\varphi(\omega)$ is usually non-smooth at ω^*. We prove that this generalized IDM converges locally quadratically. Similar to previous IDM, our generalized IDM only needs to solve linear systems at each iteration step, and in turn, IDM is more effective than other methods such as subspace method.

This paper is organized as follows. In Section 2, we apply IDM to solve eigenvalue optimization problem (1.1) for the case that $\lambda^* = \varphi(\omega^*)$ is a simple eigenvalue of $A(\omega^*)$, where $A(\omega)$ is a nonlinear matrix function of ω. Under the condition that $\lambda^* = \varphi(\omega^*)$ is an eigenvalue of $A(\omega^*)$ with multiplicity 2, we generalize IDM to solve (1.1) and prove that it converges locally quadratically in Section 3. In Section 4, numerical experiments confirm the rate of convergence established in theory and show the efficiency of generalized IDM.

2. Implicit Determinant Method for Smooth Case

We first introduce the relation between smoothness of $\varphi(\omega)$ and multiplicity of eigenvalues of $A(\omega)$ (see e.g., [9, 16, 18, 22]).

Theorem 2.1 ([9, Theorem S6.3]). *Let $A(\omega) \in \mathbb{C}^{n\times n}$ be an Hermitian matrix-valued function that depends on $\omega \in \mathbb{R}$ analytically. Then there exist scalar functions $\tilde{\lambda}_1(\omega), \ldots, \tilde{\lambda}_n(\omega)$ and a matrix-valued function $V(\omega) = [v_1(\omega), \ldots, v_n(\omega)]$, which are analytic for ω and possess the following properties for every $\omega \in \mathbb{R}$:*

$$A(\omega) = V(\omega)\,\mathrm{diag}(\tilde{\lambda}_1(\omega), \ldots, \tilde{\lambda}_n(\omega))V(\omega)^{\mathrm{H}}, \quad V(\omega)^{\mathrm{H}}V(\omega) = I.$$

The left subplot of Fig. 2.1 depicts these analytic eigenvalue curves $\tilde{\lambda}_i(\omega)$, $i = 1, \ldots, n$. Now we sort these eigenvalue curves as $\lambda_1(\omega) \geq \lambda_2(\omega) \geq \cdots \geq \lambda_n(\omega)$. The right subplot of Fig. 2.1 depicts these sorted eigenvalue curves $\lambda_i(\omega)$. From Theorem 2.1, we can see that, $\lambda_i(\omega)$, $i = 1, \ldots, n$, are continuous and piecewise analytic [20]. For a fixed point ω, if $\lambda_i(\omega)$ is a simple eigenvalue of $A(\omega)$, then $\lambda_i(\omega)$ is analytic at ω. If $\lambda_i(\omega)$ is not differentiable at ω, then $\lambda_i(\omega)$ must be a multiple eigenvalue of $A(\omega)$.

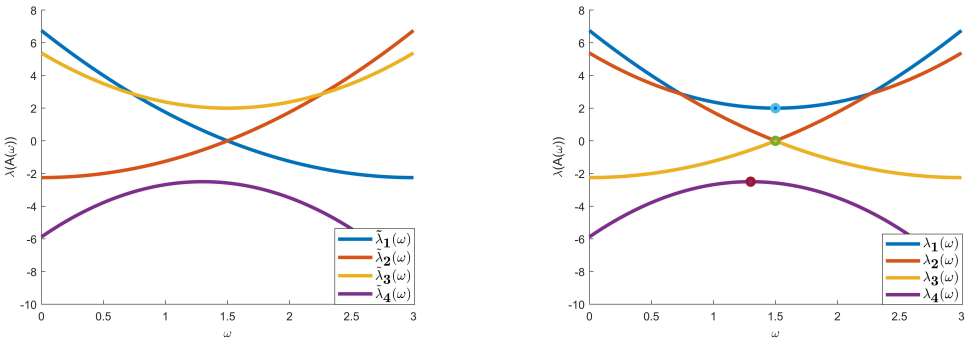

Fig. 2.1. Eigenvalue curves of $A(\omega)$. Left: Analytic eigenvalue curves $\tilde{\lambda}_i(\omega)$. Right: Sorted eigenvalue curves $\lambda_i(\omega)$ and the local extreme points marked with 'o'.

2.1. Algorithm

In this section, we apply IDM to solve eigenvalue optimization problem (1.1) where $A(\omega)$ is Hermitian and a nonlinear matrix function in ω, and $\lambda^* = \varphi(\omega^*)$ is a simple eigenvalue of $A(\omega^*)$, for example, the extreme points of eigenvalue curves $\lambda_1(\omega)$ or $\lambda_4(\omega)$ in Fig. 2.1. The generality is almost straight forward and the purposes is to introduce the IDM to readers.

Similar to the analysis of [6, section 3], we consider the following linear equation system,

$$M(\omega,\lambda)\begin{bmatrix} x(\omega,\lambda) \\ f(\omega,\lambda)\end{bmatrix} \equiv \begin{bmatrix} A(\omega)-\lambda I & c \\ c^{\mathrm{H}} & 0\end{bmatrix}\begin{bmatrix} x(\omega,\lambda) \\ f(\omega,\lambda)\end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 1\end{bmatrix}, \tag{2.1}$$

where ω, λ are two real parameters, $c \in \mathbb{C}^n$ is a constant vector, $A(\omega) \in \mathbb{C}^{n\times n}$ is an Hermitian matrix which analytically depends on ω. It is clear that $M(\omega,\lambda)$ is also an analytic matrix function with respect to ω and λ. If there exists a neighbor $\mathcal{O}$ of (ω^*,λ^*), such that $M(\omega,\lambda)$ is nonsingular when $(\omega,\lambda) \in \mathcal{O}$, then equation system (2.1) has a unique solution, say $x(\omega,\lambda)$, $f(\omega,\lambda)$, which is an analytic function of ω and λ when $(\omega,\lambda) \in \mathcal{O}$. The following proposition discuss the non-singularity of $M(\omega,\lambda)$.

Proposition 2.1. *Let ω^* solve the eigenvalue optimization problem (1.1) and $\lambda^* = \varphi(\omega^*)$. Assume that λ^* is a simple eigenvalue of $A(\omega^*)$ and x^* is the corresponding normalized eigenvector. Let $c \in \mathbb{C}^n$ be a constant vector satisfying*

$$c^{\mathrm{H}}x^* \neq 0. \tag{2.2}$$

Then $M(\omega^*,\lambda^*)$ is nonsingular and there exists a neighbor $\mathcal{O}$ of (ω^*,λ^*), such that $M(\omega,\lambda)$ is nonsingular when $(\omega,\lambda) \in \mathcal{O}$.

This proposition is a mimic of [23, Lemma 1], so we omit the proof.

If $\lambda^* = \varphi(\omega^*)$ is a simple eigenvalue of $A(\omega^*)$, then we easily deduce from Theorem 2.1 that $\varphi(\omega)$ is analytic at ω^*, and $\varphi'(\omega^*) = 0$. With the result of Proposition 2.1, the following proposition states that (ω^*,λ^*) can be computed by solving a zero of a nonlinear equation.

Proposition 2.2. *If λ^* is a simple eigenvalue of $A(\omega^*)$ and $\varphi''(\omega^*) \neq 0$, when $(\omega,\lambda) \in \mathcal{O}$, (ω^*,λ^*) can be computed by solving*

$$g(\omega,\lambda) \equiv \begin{bmatrix} f(\omega,\lambda) \\ f_\omega(\omega,\lambda)\end{bmatrix} = \mathbf{0}. \tag{2.3}$$

The proof is similar to the analysis of [6, section 3] and [23, Lemma 2]. Key points of this proof are: (1) $f(\omega,\lambda)=0$ is the necessary and sufficient condition of $A(\omega)x(\omega,\lambda)=\lambda x(\omega,\lambda)$, (2) with $f(\omega,\lambda)=0$, when $(\omega,\lambda)\in\mathcal{O}$, $f_\omega(\omega,\lambda)=0$ is the necessary and sufficient condition of $\varphi'(\omega)=0$.

By using Cramers rule to the linear equation (2.1)

$$f(\omega,\lambda)=\frac{\det(A(\omega)-\lambda I)}{\det(M(\omega,\lambda))}. \tag{2.4}$$

Thus $f(\omega,\lambda)=0$ if and only if $\det(A(\omega)-\lambda I)=0$, i.e., $\lambda(\omega)$ is an eigenvalue of $A(\omega)$. IDM solves linear system (2.1) to obtain $f(\omega,\lambda)$, i.e., compute the determinant implicitly, that is why this method is called implicit determinant method. IDM is precisely the application of Newton's method to solve a zero of nonlinear equation $g(\omega,\lambda)$ in (2.3). Now we consider the Jacobian of $g(\omega,\lambda)$.

By differentiating (2.1) with respect to ω, we get

$$\begin{bmatrix} A(\omega)-\lambda I & c \\ c^{\mathrm{H}} & 0 \end{bmatrix}\begin{bmatrix} x_\omega(\omega,\lambda) \\ f_\omega(\omega,\lambda) \end{bmatrix}=\begin{bmatrix} -A'(\omega)x(\omega,\lambda) \\ 0 \end{bmatrix}. \tag{2.5}$$

Differentiating (2.1) with respect to λ gives

$$\begin{bmatrix} A(\omega)-\lambda I & c \\ c^{\mathrm{H}} & 0 \end{bmatrix}\begin{bmatrix} x_\lambda(\omega,\lambda) \\ f_\lambda(\omega,\lambda) \end{bmatrix}=\begin{bmatrix} x(\omega,\lambda) \\ 0 \end{bmatrix}. \tag{2.6}$$

By differentiating (2.5) with respect to ω, we get

$$\begin{bmatrix} A(\omega)-\lambda I & c \\ c^{\mathrm{H}} & 0 \end{bmatrix}\begin{bmatrix} x_{\omega\omega}(\omega,\lambda) \\ f_{\omega\omega}(\omega,\lambda) \end{bmatrix}=\begin{bmatrix} -2A'(\omega)x_\omega(\omega,\lambda)-A''(\omega)x(\omega,\lambda) \\ 0 \end{bmatrix}, \tag{2.7}$$

where x_ω is obtained from (2.5). Similarly, by differentiating (2.5) with respect to λ, we get

$$\begin{bmatrix} A(\omega)-\lambda I & c \\ c^{\mathrm{H}} & 0 \end{bmatrix}\begin{bmatrix} x_{\omega\lambda}(\omega,\lambda) \\ f_{\omega\lambda}(\omega,\lambda) \end{bmatrix}=\begin{bmatrix} -A'(\omega)x_\lambda(\omega,\lambda)+x_\omega(\omega,\lambda) \\ 0 \end{bmatrix}, \tag{2.8}$$

where x_λ and x_ω is obtained from (2.6) and (2.5). By solving four Eqs. (2.5)–(2.8) with the same coefficient matrix $M(\omega,\lambda)$, we get the entries of the Jacobian of $g(\omega,\lambda)$:

$$G(\omega,\lambda)=\begin{bmatrix} f_\omega(\omega,\lambda) & f_\lambda(\omega,\lambda) \\ f_{\omega\omega}(\omega,\lambda) & f_{\omega\lambda}(\omega,\lambda) \end{bmatrix}. \tag{2.9}$$

The following proposition guarantees the non-singularity of $G(\omega,\lambda)$.

Proposition 2.3. *Assume that λ^* is a simple eigenvalue of $A(\omega^*)$ and $\varphi''(\omega^*)\neq 0$, then $G(\omega,\lambda)$ is nonsingular at (ω^*,λ^*).*

Proof. We deduce from Proposition 2.2 that $f_\omega(\omega^*,\lambda^*)=0$, thus $G(\omega^*,\lambda^*)$ is nonsingular if and only if $f_\lambda(\omega^*,\lambda^*)\neq 0$ and $f_{\omega\omega}(\omega^*,\lambda^*)\neq 0$.

By similar proof of [6, Lemma 3], we multiply the first row of (2.6) with $x^{\mathrm{H}}(\omega,\lambda)$ from the left and get

$$f_\lambda(\omega,\lambda)=\frac{\|x(\omega,\lambda)\|^2}{x^{\mathrm{H}}c}\neq 0.$$

Since $f_\lambda(\omega^*, \lambda^*) \neq 0$, by the implicit function theorem, we conclude that when (ω, λ) is in a neighbor of (ω^*, λ^*), the equation $f(\omega, \lambda) = 0$ implicitly determines λ as a function of ω, i.e., $\lambda(\omega)$. The first row of Eq. (2.1) writes $(A(\omega) - \lambda I)x(\omega, \lambda) + f(\omega, \lambda)c = \mathbf{0}$. We deduce that $f(\omega, \lambda) = 0$ if and only if $A(\omega)x(\omega, \lambda) = \lambda x(\omega, \lambda)$. Thus $\lambda(\omega) = \varphi(\omega)$. Then we conclude that

$$\lambda'(\omega^*) = 0, \quad \lambda''(\omega^*) \neq 0.$$

By differentiating $f(\omega, \lambda) = 0$, we get

$$\lambda'(\omega) = -\frac{f_\omega(\omega, \lambda)}{f_\lambda(\omega, \lambda)}. \tag{2.10}$$

By differentiating (2.10) with respect to ω, we obtain

$$\lambda''(\omega) = \frac{-f_{\omega\omega}(\omega, \lambda)f_\lambda(\omega, \lambda) + f_{\lambda\omega}(\omega, \lambda)f_\omega(\omega, \lambda)}{f_\lambda(\omega, \lambda)^2}. \tag{2.11}$$

Note that $f_\omega(\omega^*, \lambda^*) = 0$ and $f_\lambda(\omega^*, \lambda^*) \neq 0$, then at the point ω^*, Eq. (2.11) becomes

$$\lambda''(\omega^*) = -\frac{f_{\omega\omega}(\omega^*, \lambda^*)}{f_\lambda(\omega^*, \lambda^*)}.$$

Since we have $f_\lambda(\omega^*, \lambda^*) \neq 0$ and $\lambda''(\omega^*) \neq 0$, it follows that

$$f_{\omega\omega}(\omega^*, \lambda^*) \neq 0.$$

Then we have $\det(G(\omega^*, \lambda^*)) = -f_\lambda(\omega^*, \lambda^*)f_{\omega\omega}(\omega^*, \lambda^*) \neq 0$. Thus $G(\omega^*, \lambda^*)$ is nonsingular. The proof of $f_{\omega\omega}(\omega^*, \lambda^*) \neq 0$ is different from that in [6, Lemma 4] and [7, Lemma 3.1], since their proof make use of the property that $A(\omega)$ is linear with respect to ω in the calculation of stable radius and H_∞ norm. □

Here we present IDM for solving the eigenvalue optimization problem (1.1) under the condition that $\varphi(\omega^*)$ is a simple eigenvalue of $A(\omega^*)$ and $\varphi''(\omega^*) \neq 0$.

Algorithm 2.1. Implicit determinant method for smooth case

Require: initial guess ω_0, λ_0, $A(\omega)$, $A'(\omega)$, $A''(\omega)$, tolerance **tol**.
Ensure: approximate solution ω_k, λ_k of the eigenvalue optimization problem (1.1)
1: $k = 0$, $r_k = 1$.
2: **while** $r_k >$ **tol do**
3: Solve Eqs. (2.1), (2.5), (2.6), (2.7), (2.8) to get $g(\omega_k, \lambda_k)$ and $G(\omega_k, \lambda_k)$;
4: $\begin{bmatrix}\omega_{k+1}\\ \lambda_{k+1}\end{bmatrix} = \begin{bmatrix}\omega_k\\ \lambda_k\end{bmatrix} - G(\omega_k, \lambda_k)^{-1}g(\omega_k, \lambda_k)$;
5: $r_k = \|g(\omega_k, \lambda_k)\|$;
6: **end while**

Initial guess ω_0 and λ_0 are application-dependent. If only ω_0 is given, we always set $\lambda_0 = \varphi(\omega_0)$. Note that if $c \approx x^*$, then $c^{\mathrm{H}}x^* \neq 0$, and (2.1) is nonsingular by Proposition 2.1. So we let c to be the normalized eigenvector of $A(\omega_0)$ corresponding to λ_0.

Similar to the analysis of [6, section 4.1], the linear systems in step 2.1 of Algorithm 2.1 have the same coefficient matrix $M(\omega_k, \lambda_k)$, thus only one LU factorisation is needed per iteration. Since IDM is essentially Newton's method for solving a zero of the equation $g(\omega, \lambda) = 0$, it converges locally and quadratically. Generally, the computation complexity of one LU factorisation is cheaper than solving an eigenvalue problem, IDM is faster than methods which require to solve an eigenvalue problem at each iteration step, providing that we have a good initial.

3. Implicit Determinant Method for Non-smooth Case

Non-smoothness at extreme points is a common phenomenon in eigenvalue optimization problems. In Section 2, we apply IDM to solve the eigenvalue optimization problem (1.1) if λ^* is a simple eigenvalue of $A(\omega^*)$, i.e., $\varphi(\omega)$ is smooth at ω^*. However, if $\varphi(\omega)$ is not differentiable at ω^*, for example, the extreme points of $\lambda_2(\omega)$ and $\lambda_3(\omega)$ in the right part of Fig. 2.1, the assumption that λ^* is a simple eigenvalue of $A(\omega^*)$ in Proposition 2.2 is not satisfied. Thus Algorithm 2.1 is not suitable for this non-smooth case, as shown in Example 4.2, Table 4.2, the sequence generated by Algorithm 2.1 will not converge to (ω^*, λ^*).

In this section, we mainly consider the case that λ^* is an eigenvalue of $A(\omega^*)$ with multiplicity 2, possible extension to the case where multiplicity is higher than 2 is considered in Subsection 3.3.

Let (ω^*, λ^*) be the intersection of two analytic functions $\tilde{\lambda}_1(\omega)$ and $\tilde{\lambda}_2(\omega)$. Since ω^* is an extreme point of $\varphi(\omega)$, we have $\tilde{\lambda}_1'(\omega^*)\tilde{\lambda}_2'(\omega^*) \leq 0$. The following theorem shows that $\tilde{\lambda}_1'(\omega^*)$ and $\tilde{\lambda}_2'(\omega^*)$ can be obtained from $A'(\omega^*)$ and eigenvectors of $A(\omega^*)$ corresponding to λ^*.

Theorem 3.1. *Let $(\tilde{\lambda}_1(\omega), v_1(\omega))$, $(\tilde{\lambda}_2(\omega), v_2(\omega))$ be two analytic eigenpairs of $A(\omega)$. Let (ω^*, λ^*) be the intersection of $\tilde{\lambda}_1(\omega)$ and $\tilde{\lambda}_2(\omega)$. We denote*

$$X^* = [v_1(\omega^*), v_2(\omega^*)], \quad \Lambda_1(\omega) = \mathrm{diag}(\tilde{\lambda}_1(\omega), \tilde{\lambda}_2(\omega)). \tag{3.1}$$

Then

$$X^{*\mathrm{H}} A'(\omega^*) X^* = \Lambda_1'(\omega^*) = \mathrm{diag}(\tilde{\lambda}_1'(\omega^*), \tilde{\lambda}_2'(\omega^*)). \tag{3.2}$$

Proof. Letting $V_1(\omega) = [v_1(\omega), v_2(\omega)]$, we have

$$A(\omega)V_1(\omega) = V_1(\omega)\Lambda_1(\omega). \tag{3.3}$$

The derivative of (3.3) is

$$A'(\omega)V_1(\omega) + A(\omega)V_1'(\omega) = V_1'(\omega)\Lambda_1(\omega) + V_1(\omega)\Lambda_1'(\omega). \tag{3.4}$$

By left multiplying (3.4) with $V_1^{\mathrm{H}}(\omega)$, we get

$$V_1^{\mathrm{H}}(\omega)A'(\omega)V_1(\omega) + V_1^{\mathrm{H}}(\omega)A(\omega)V_1'(\omega) = V_1^{\mathrm{H}}(\omega)V_1'(\omega)\Lambda_1(\omega) + V_1^{\mathrm{H}}(\omega)V_1(\omega)\Lambda_1'(\omega).$$

Note that $\Lambda_1(\omega^*) = \lambda^* I$. We have

$$\begin{aligned} V_1^{\mathrm{H}}(\omega^*)A(\omega^*)V_1'(\omega^*) &= \Lambda_1(\omega^*)V_1^{\mathrm{H}}(\omega^*)V_1'(\omega^*) \\ &= \lambda^* V_1^{\mathrm{H}}(\omega^*)V_1'(\omega^*) \\ &= V_1^{\mathrm{H}}(\omega^*)V_1'(\omega^*)\Lambda_1(\omega^*). \end{aligned}$$

It follows that

$$V_1^{\mathrm{H}}(\omega^*)A'(\omega^*)V_1(\omega^*) = V_1^{\mathrm{H}}(\omega^*)V_1(\omega^*)\Lambda_1'(\omega^*) = \Lambda_1'(\omega^*).$$

For the case that $\lambda_i(\omega^*)$ is a simple eigenvalue of $A(\omega^*)$, it is well known that $\lambda_i'(\omega^*) = v_i^{\mathrm{H}}(\omega^*)A'(\omega^*)v_i(\omega^*)$ [18, Theorem 5]. By Theorem 3.1, for the case that $\lambda_i(\omega^*)$ is a multiple eigenvalue of $A(\omega^*)$, the conclusion is the same. This theorem plays an important role in the derivation of IDM. □

In the remainder of this section, we make the following assumption:

Assumption 3.1. *Let $\tilde{\lambda}_1(\omega)$, $\tilde{\lambda}_2(\omega)$ be defined by Theorem* 2.1. *Let (ω^*, λ^*) be the intersection of $\tilde{\lambda}_1(\omega)$ and $\tilde{\lambda}_2(\omega)$. Assume that λ^* is an eigenvalue of $A(\omega^*)$ with multiplicity* 2 *and $\tilde{\lambda}_1'(\omega^*)\tilde{\lambda}_2'(\omega^*) < 0$.*

If λ^* is an eigenvalue of $A(\omega^*)$ with multiplicity 2, there are three cases, 1: $\tilde{\lambda}_1'(\omega^*) \neq 0$, $\tilde{\lambda}_2'(\omega^*) \neq 0$. 2: $\tilde{\lambda}_1'(\omega^*) = 0$, $\tilde{\lambda}_2'(\omega^*) \neq 0$. 3: $\tilde{\lambda}_1'(\omega^*) = \tilde{\lambda}_2'(\omega^*) = 0$. See Fig. 3.1. Please note that Assumption 3.1 only excludes the case that

$$\tilde{\lambda}_1'(\omega^*)\tilde{\lambda}_2'(\omega^*) = 0,$$

i.e., middle and right figure in Fig. 3.1. In this case, the IDM can also be derived and analysed, but the process will be more tedious.

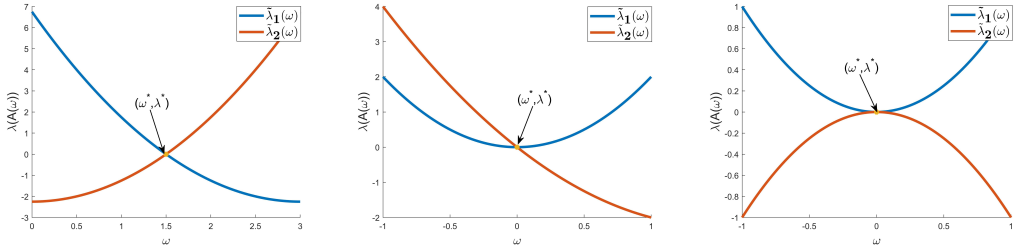

Fig. 3.1. If λ^* is an eigenvalue of $A(\omega^*)$ with multiplicity 2, there are three cases. Left: $\tilde{\lambda}_1'(\omega^*) \neq 0$, $\tilde{\lambda}_2'(\omega^*) \neq 0$. Middle: $\tilde{\lambda}_1'(\omega^*) = 0$, $\tilde{\lambda}_2'(\omega^*) \neq 0$. Right: $\tilde{\lambda}_1'(\omega^*) = \tilde{\lambda}_2'(\omega^*) = 0$.

3.1. Algorithm

In this subsection, we generalize IDM to solve eigenvalue optimization problem (1.1) under Assumption 3.1. Analogous to Section 2, we consider the following linear equation:

$$M(\omega,\lambda)\begin{bmatrix} x(\omega,\lambda) \\ \boldsymbol{f}(\omega,\lambda)\end{bmatrix} \equiv \begin{bmatrix} A(\omega)-\lambda I & C \\ C^{\mathrm{H}} & 0\end{bmatrix}\begin{bmatrix} x(\omega,\lambda) \\ \boldsymbol{f}(\omega,\lambda)\end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ d\end{bmatrix}, \tag{3.5}$$

where $d \in \mathbb{C}^2$ is a nonzero constant vector, $C \in \mathbb{C}^{n\times 2}$ is a constant matrix, $A(\omega) \in \mathbb{C}^{n\times n}$ is an Hermitian matrix which analytically depends on ω. It is clear that $M(\omega,\lambda)$ is also an analytic function with respect to ω and λ. If there exist a neighbor $\mathcal{O}$ of (ω^*,λ^*), such that $M(\omega,\lambda)$ is nonsingular when $(\omega,\lambda) \in \mathcal{O}$, then the system (3.5) has a unique solution, say $x(\omega,\lambda) \in \mathbb{C}^n$, $\boldsymbol{f}(\omega,\lambda) \in \mathbb{C}^2$, which is an analytic function of ω and λ when $(\omega,\lambda) \in \mathcal{O}$.

The following lemma discuss the non-singularity of $M(\omega^*,\lambda^*)$.

Lemma 3.1. *Let matrix X^* be defined in Theorem* 3.1. *Let $C \in \mathbb{C}^{n\times 2}$ be a constant matrix satisfying that $C^{\mathrm{H}}X^*$ is nonsingular. Under Assumption* 3.1*, there exists a rectangular neighbor $\mathcal{O} = \mathcal{O}_\omega \times \mathcal{O}_\lambda$ of (ω^*,λ^*), such that Hermitian matrix $M(\omega,\lambda)$ is nonsingular when $(\omega,\lambda) \in \mathcal{O}$.*

The proof is similar to that of [23, Lemma 1], so we omit it.

Assume that $C^{\mathrm{H}}X^*$ is nonsingular. We make the following assumption about the neighbor $\mathcal{O}$:

Assumption 3.2. *Let $\mathcal{O} = \mathcal{O}_\omega \times \mathcal{O}_\lambda$ be a rectangular neighbor of (ω^*,λ^*) satisfying the following two conditions:*

1. *If* $(\omega, \lambda) \in \mathcal{O}$, $M(\omega, \lambda)$ *is nonsingular.*
2. *If* $\omega \in \mathcal{O}_\omega$, $\varphi(\omega)$ *is a simple eigenvalue of* $A(\omega)$ *except for* ω^*.

The following theorem states that, the point (ω^*, λ^*) can be obtained by solving a zero of nonlinear equation $\boldsymbol{f}(\omega, \lambda) = \mathbf{0}$.

Theorem 3.2. *Let*

$$\mathcal{X} = \{v_i(\omega) \mid \omega \in \mathcal{O}_\omega, \quad \omega \neq \omega^*, \quad i = 1, 2\},$$

where $v_i(\omega)$, $i = 1, 2$ *are defined by Theorem* 3.1. *If for any* $x \in \mathcal{X}$, d *is not parallel to* $C^{\mathrm{H}}x$, *then* (ω^*, λ^*) *is the only point in* $\mathcal{O}$ *such that*

$$\boldsymbol{f}(\omega, \lambda) = \mathbf{0}.$$

Proof. We first prove $\boldsymbol{f}(\omega^*, \lambda^*) = \mathbf{0}$. The first row of Eq. (3.5) at (ω^*, λ^*) writes

$$(A(\omega^*) - \lambda^* I)x(\omega^*, \lambda^*) + C\boldsymbol{f}(\omega^*, \lambda^*) = \mathbf{0}. \tag{3.6}$$

Multiplying (3.6) by $X^{*\mathrm{H}}$ from the left, we obtain

$$X^{*\mathrm{H}} C\boldsymbol{f}(\omega^*, \lambda^*) = \mathbf{0}.$$

Since $C^{\mathrm{H}}X^*$ is nonsingular, thus

$$\boldsymbol{f}(\omega^*, \lambda^*) = \mathbf{0}.$$

On the other side, if $\boldsymbol{f}(\omega, \lambda) = \mathbf{0}$, Eq. (3.5) can be rewritten as

$$\begin{bmatrix} A(\omega) - \lambda I \\ C^{\mathrm{H}} \end{bmatrix} x = \begin{bmatrix} \mathbf{0} \\ d \end{bmatrix}. \tag{3.7}$$

The first row of (3.7) writes $A(\omega)x = \lambda x$. Note that $M(\omega, \lambda)$ is nonsingular when $(\omega, \lambda) \in \mathcal{O}$. It follows that, when $(\omega, \lambda) \in \mathcal{O}$,

$$\operatorname{rank} \begin{bmatrix} A(\omega) - \lambda I \\ C^{\mathrm{H}} \end{bmatrix} = n. \tag{3.8}$$

We will prove that λ is a multiple eigenvalue of $A(\omega)$ by contradiction. If λ is a simple eigenvalue of $A(\omega)$, then $x \in \mathcal{X}$, i.e., $x = \|x\| v_i(\omega)$, $i \in \{1, 2\}$. Consider a unitary matrix $U = [\frac{x}{\|x\|}, U_1]$. When $(\omega, \lambda) \in \mathcal{O}$, we have

$$\begin{aligned} \operatorname{rank} \begin{bmatrix} A(\omega) - \lambda I & \mathbf{0} \\ C^{\mathrm{H}} & d \end{bmatrix} &= \operatorname{rank} \begin{bmatrix} A(\omega) - \lambda I & \mathbf{0} \\ C^{\mathrm{H}} & d \end{bmatrix} \begin{bmatrix} U & \\ & 1 \end{bmatrix} \\ &= \operatorname{rank} \begin{bmatrix} \mathbf{0} & (A(\omega) - \lambda I)U_1 & \mathbf{0} \\ \frac{C^{\mathrm{H}}x}{\|x\|} & C^{\mathrm{H}}U_1 & d \end{bmatrix} = n + 1. \end{aligned}$$

The last equality being a consequence of d is not parallel to $C^{\mathrm{H}}x$, and by Eq. (3.8),

$$\operatorname{rank} \begin{bmatrix} A(\omega) - \lambda I \\ C^{\mathrm{H}} \end{bmatrix} U = \operatorname{rank} \begin{bmatrix} \mathbf{0} & (A(\omega) - \lambda I)U_1 \\ \frac{C^{\mathrm{H}}x}{\|x\|} & C^{\mathrm{H}}U_1 \end{bmatrix} = n.$$

For the system (3.7), the rank of augmented matrix is not equal to that of the coefficient matrix, then there does not exist a solution satisfies (3.7), which leads to a contradiction. We deduce from condition 3.2 of Assumption 3.2 that, (ω, λ) determined by $\boldsymbol{f}(\omega, \lambda) = \mathbf{0}$ is (ω^*, λ^*). $\square$

By Theorem 3.2, we only need to seek the solution of $\boldsymbol{f}(\omega, \lambda) = \mathbf{0}$. We apply Newton's method to solve it. First consider the Jacobian of $\boldsymbol{f}(\omega, \lambda)$. By differentiating (3.5) with respect to ω and λ respectively, we get

$$\begin{bmatrix} A(\omega) - \lambda I & C \\ C^{\mathrm{H}} & 0 \end{bmatrix} \begin{bmatrix} x_\omega(\omega, \lambda) \\ \boldsymbol{f}_\omega(\omega, \lambda) \end{bmatrix} = \begin{bmatrix} -A'(\omega)x(\omega, \lambda) \\ \mathbf{0} \end{bmatrix}, \tag{3.9}$$

$$\begin{bmatrix} A(\omega) - \lambda I & C \\ C^{\mathrm{H}} & 0 \end{bmatrix} \begin{bmatrix} x_\lambda(\omega, \lambda) \\ \boldsymbol{f}_\lambda(\omega, \lambda) \end{bmatrix} = \begin{bmatrix} x(\omega, \lambda) \\ \mathbf{0} \end{bmatrix}, \tag{3.10}$$

where $x(\omega, \lambda)$ can be obtained by solving the linear equation system (3.5). The Jacobian of $\boldsymbol{f}(\omega, \lambda)$ is

$$J(\omega, \lambda) = [\boldsymbol{f}_\omega(\omega, \lambda), \boldsymbol{f}_\lambda(\omega, \lambda)]. \tag{3.11}$$

Now we give the non-singularity of $J(\omega^*, \lambda^*)$.

Theorem 3.3. *Under the Assumptions* 3.1 *and* 3.2, $J(\omega^*, \lambda^*)$ *is nonsingular if and only if* d *is not parallel to* $C^{\mathrm{H}}X^*e_i$, $i = 1, 2$, *where* e_i *is the* i_{th} *column of the two by two identity matrix.*

Proof. The first rows of (3.9) and (3.10) at (ω^*, λ^*) are

$$(A(\omega^*) - \lambda^* I)x_\omega(\omega^*, \lambda^*) + C\boldsymbol{f}_\omega(\omega^*, \lambda^*) = -A'(\omega^*)x(\omega^*, \lambda^*), \tag{3.12}$$

and

$$(A(\omega^*) - \lambda^* I)x_\lambda(\omega^*, \lambda^*) + C\boldsymbol{f}_\lambda(\omega^*, \lambda^*) = x(\omega^*, \lambda^*), p \tag{3.13}$$

respectively. By multiplying (3.12) and (3.13) with $X^{*\mathrm{H}}$ from the left, we get

$$X^{*\mathrm{H}}C[\boldsymbol{f}_\omega(\omega^*, \lambda^*), \boldsymbol{f}_\lambda(\omega^*, \lambda^*)] = [-X^{*\mathrm{H}}A'(\omega^*)x(\omega^*, \lambda^*), X^{*\mathrm{H}}x(\omega^*, \lambda^*)]. \tag{3.14}$$

Note that $C^{\mathrm{H}}X^*$ is nonsingular. Then $J(\omega^*, \lambda^*)$ is nonsingular if and only if the right side of Eq. (3.14) is nonsingular.

Since $(A(\omega^*) - \lambda^* I)x(\omega^*, \lambda^*) = 0$, we deduce that $x(\omega^*, \lambda^*)$ belongs to the span of the columns of X^*. Thus there exists a nonzero vector $y \in \mathbb{C}^2$ such that

$$x(\omega^*, \lambda^*) = X^* y. \tag{3.15}$$

By Theorem 3.1 and Eq. (3.15), we conclude that

$$\begin{aligned} X^{*\mathrm{H}}CJ(\omega^*, \lambda^*) &= [-X^{*\mathrm{H}}A'(\omega^*)X^*y, X^{*\mathrm{H}}X^*y] \\ &= [-\operatorname{diag}(\tilde{\lambda}_1'(\omega^*), \tilde{\lambda}_2'(\omega^*))y, y]. \end{aligned}$$

Thus $J(\omega^*, \lambda^*)$ is nonsingular if and only if y is not the eigenvector of $\operatorname{diag}(\tilde{\lambda}_1'(\omega^*), \tilde{\lambda}_2'(\omega^*))$, i.e., y is not parallel to e_i, $i = 1, 2$. Note that $C^{\mathrm{H}}x(\omega^*, \lambda^*) = d$. Thus we have

$$C^{\mathrm{H}}X^*y = d. \tag{3.16}$$

We deduce that $J(\omega^*, \lambda^*)$ is nonsingular if and only if d is not parallel to $C^{\mathrm{H}}X^*e_i$, $i = 1, 2$. □

Here we show that a random nonzero vector d is very likely to satisfy assumptions in Theorems 3.2 and 3.3, so that we can apply Newton's method to solve a zero of $\boldsymbol{f}(\omega, \lambda) = 0$ and get (ω^*, λ^*). By Theorem 2.1, $v_i(\omega)$, $i = 1, 2$, are continuous on ω. If $x \in \mathcal{X}$, then x is close to $v_i(\omega^*)$, i.e., $C^{\mathrm{H}}x$ is close to $C^{\mathrm{H}}X^*e_i$, $i = 1$ or 2. From above analysis we conclude that, if the direction of d is not close to $C^{\mathrm{H}}X^*e_i$, $i = 1, 2$, then d satisfies assumptions in Theorems 3.2 and 3.3. A random vector d is very likely not close to $C^{\mathrm{H}}X^*e_i$, $i = 1, 2$.

Now we give IDM for non-smooth case.

Algorithm 3.1. Implicit determinant method for non-smooth case

Require: initial guess ω_0, λ_0, $A(\omega)$, $A'(\omega)$, tolerance **tol**.
Ensure: approximate solution ω_k, λ_k of eigenvalue optimization problem (1.1)
1: $k = 0$, $r_k = 1$.
2: **while** $r_k >$ **tol do**
3: Solve Eqs. (3.5), (3.9) and (3.10) to get $\boldsymbol{f}(\omega_k, \lambda_k)$ and $J(\omega_k, \lambda_k)$;
4: $\begin{bmatrix}\omega_{k+1}\\ \lambda_{k+1}\end{bmatrix} = \begin{bmatrix}\omega_k\\ \lambda_k\end{bmatrix} - J(\omega_k, \lambda_k)^{-1}\boldsymbol{f}(\omega_k, \lambda_k)$;
5: $r_k = \|\boldsymbol{f}(\omega_k, \lambda_k)\|$, $k = k + 1$;
6: **end while**

Initial guess ω_0 and λ_0 are application-dependent. If only ω_0 is given, we always set $\lambda_0 = \varphi(\omega_0)$. Note that if $C \approx X^*$, then $C^{\mathrm{H}}X^*$ is nonsingular, and Eq. (3.5) is nonsingular by Lemma 3.1. So we let

$$C = [v_1(\omega_0), v_2(\omega_0)],$$

where $v_1(\omega)$,$v_2(\omega)$ are defined by Theorem 3.1. If ω_0 is close to ω^*, then $C^{\mathrm{H}}X^*$ is close to the identity matrix I, and $C^{\mathrm{H}}X^*e_i$ is close to e_i, $i = 1, 2$. Thus we let the direction of d be away from e_i, $i = 1, 2$, such that it satisfies assumptions in Theorems 3.2 and 3.3.

Similar to the analysis of [6, section 4.1], most computational cost of Algorithm 3.1 is in step 3, i.e., solving three linear systems with the same coefficient matrix, thus only one LU factorisation is needed per iteration. Generally, the computation complexity of one LU factorisation is cheaper than solving an eigenvalue problem, thus IDM is faster than the methods which require to solve an eigenvalue problem at each iteration step. Since IDM is essentially applying Newton's method to solve the zero of $\boldsymbol{f}(\omega, \lambda) = \mathbf{0}$, it converges locally and quadratically.

3.2. A special choice of the vector d

In subsection 3.1, we show that the sequence (ω_k, λ_k) generated by Algorithm 3.1 will locally quadratically converge to (ω^*, λ^*). However, sequence (ω_k, λ_k) generated by Algorithm 3.1 may converge to a point which is not an extreme point, like point (ω_m, λ_m) in Fig. 3.2. Actually, Algorithm 3.1 only ensures the computed λ_k is an approximate eigenvalue with multiplicity 2. In this subsection, we will present an algorithm which is guaranteed to get a local extreme point.

Consider the following linear equation system:

$$\begin{bmatrix}A(\omega) - \lambda I & C\\ C^{\mathrm{H}} & \mathbf{0}\end{bmatrix}\begin{bmatrix}X(\omega, \lambda)\\ F(\omega, \lambda)\end{bmatrix} = \begin{bmatrix}\mathbf{0}\\ I\end{bmatrix}. \tag{3.17}$$

By Assumption 3.2, when $(\omega, \lambda) \in \mathcal{O}$, equation system (3.17) has a unique solution, say $X(\omega, \lambda)$, $F(\omega, \lambda)$, which is analytic functions of ω and λ when $(\omega, \lambda) \in \mathcal{O}$. It can be checked easily that

$$F(\omega^*, \lambda^*) = \mathbf{0}, \tag{3.18}$$

$$X(\omega^*, \lambda^*) = X^*(C^{\mathrm{H}}X^*)^{-1}. \tag{3.19}$$

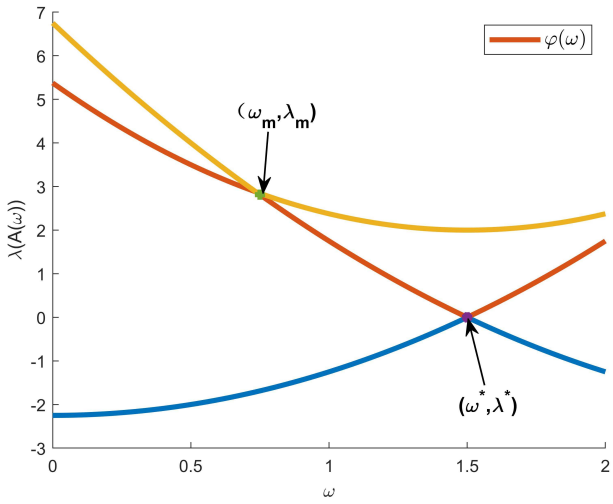

Fig. 3.2. Sequence generated by Algorithm 3.1 may converge to (ω_m, λ_m), where λ_m is a multiple eigenvalue of $A(\omega_m)$ but ω_m is not a local extreme point of $\varphi(\omega)$.

Note that the two columns of $X(\omega^*, \lambda^*)$ are two linearly independent eigenvectors of $A(\omega^*)$ corresponding to λ^*. If $C = X^*$, then $X(\omega^*, \lambda^*) = X^*$. If $C \neq X^*$, the two columns of $X(\omega^*, \lambda^*)$ are not orthogonal, while the two columns of X^* are orthogonal.

Since in practice, we choose C such that C is close to X^*, then by Eq. (3.19), $X(\omega^*, \lambda^*)$ is close to X^*. We deduce from Theorem 3.1 and Assumption 3.1 that $X^{*\mathrm{H}} A'(\omega^*) X^*$ is an indefinite matrix, say it has both positive and negative eigenvalues, then $X^{\mathrm{H}}(\omega^*, \lambda^*) A'(\omega^*) X(\omega^*, \lambda^*)$ should also be an indefinite matrix. Thus there exists a nonzero vector $d^* \in \mathbb{C}^2$ such that

$$d^{*\mathrm{H}} X^{\mathrm{H}}(\omega^*, \lambda^*) A'(\omega^*) X(\omega^*, \lambda^*) d^* = 0. \tag{3.20}$$

We restrict the first component of d^* to be nonnegative and $\|d^*\| = 1$, then d^* is unique.

If we replace d with d^* in equation (3.5), then the solution of (3.5), say $x(\omega^*, \lambda^*) = X(\omega^*, \lambda^*) d^*$, satisfies the equation

$$x(\omega^*, \lambda^*)^{\mathrm{H}} A'(\omega^*) x(\omega^*, \lambda^*) = 0. \tag{3.21}$$

Note that,

$$(C^{\mathrm{H}} X^* e_i)^{\mathrm{H}} X^{\mathrm{H}}(\omega^*, \lambda^*) A'(\omega^*) X(\omega^*, \lambda^*)(C^{\mathrm{H}} X^* e_i) = \tilde{\lambda}_i'(\omega^*) \neq 0.$$

From above, the direction of d^* is away from $C^{\mathrm{H}} X^* e_i$, $i = 1, 2$. Thus d^* satisfies assumptions in Theorems 3.2 and 3.3.

Here we describe a process which generates the sequence (ω_k, λ_k) and d_k, such that (ω_k, λ_k) converges to (ω^*, λ^*) and d_k converges to d^*, both at a quadratic rate.

Since (ω_k, λ_k) is close to (ω^*, λ^*), we assume that $X^{\mathrm{H}}(\omega_k, \lambda_k) A'(\omega_k) X(\omega_k, \lambda_k)$ is also an indefinite matrix. With a point (ω_k, λ_k) at the k_{th} step, we choose d_k such that

$$d_k^{\mathrm{H}} X(\omega_k, \lambda_k)^{\mathrm{H}} A'(\omega_k) X(\omega_k, \lambda_k) d_k = 0. \tag{3.22}$$

We restrict the first component of d_k to be nonnegative and $\|d_k\| = 1$, then d_k is unique. With d replaced by d_k in the Eq. (3.5), by doing one-step Newton's method to solve the zero of equation $\boldsymbol{f}(\omega, \lambda) = 0$, we get the updated $(\omega_{k+1}, \lambda_{k+1})$. Repeat these procedures until convergence.

If $X(\omega_k, \lambda_k)^{\mathrm{H}} A'(\omega_k) X(\omega_k, \lambda_k)$ is definite, then there does not exist a nonzero vector d_k such that $d_k^{\mathrm{H}} X(\omega_k, \lambda_k)^{\mathrm{H}} A'(\omega_k) X(\omega_k, \lambda_k) d_k = 0$. Then the process stops and returns that with this initial point, it can not converge to a local extreme point.

With the following lemma, we will show that d_k converges to d^* at the same rate as (ω_k, λ_k) converges to (ω^*, λ^*).

Lemma 3.2. *Let (ω_k, λ_k) be an $O(\epsilon)$ approximation of (ω^*, λ^*), i.e.,*

$$\|\omega_k - \omega^*\| = O(\epsilon), \quad \|\lambda_k - \lambda^*\| = O(\epsilon),$$

where ϵ is a small quantity, $O(\epsilon)$ is a quantity that has the same order as ϵ. Then

$$\|d_k - d^*\| = O(\epsilon).$$

Proof. With (ω_k, λ_k), we get the unique vector d_k which satisfies following conditions: (1) $d_k^{\mathrm{H}} X(\omega_k, \lambda_k)^{\mathrm{H}} A'(\omega_k) X(\omega_k, \lambda_k) d_k = 0$, (2) the first component of d_k is nonnegative, (3) $\|d_k\| = 1$. Note that $A(\omega)$ is analytic with respect to ω, and $X(\omega, \lambda)$ is an analytic function of ω and λ, then we have $X^{\mathrm{H}}(\omega, \lambda) A'(\omega) X(\omega, \lambda)$ is also analytic with respect to ω and λ. Since (ω_k, λ_k) is an $O(\epsilon)$ approximation of (ω^*, λ^*), it can be deduced that,

$$X^{\mathrm{H}}(\omega_k, \lambda_k) A'(\omega_k) X(\omega_k, \lambda_k) - X^{\mathrm{H}}(\omega^*, \lambda^*) A'(\omega^*) X(\omega^*, \lambda^*) = O(\epsilon).$$

Be recalled that $\|d_k\| = \|d^*\| = 1$, the first components of both d_k and d^* are nonnegative, d^* and d_k satisfy $d^{*\mathrm{H}} X^{\mathrm{H}}(\omega^*, \lambda^*) A'(\omega^*) X(\omega^*, \lambda^*) d^* = 0$ and $d_k^{\mathrm{H}} X(\omega_k, \lambda_k)^{\mathrm{H}} A'(\omega_k) X(\omega_k, \lambda_k) d_k = 0$ respectively, the rest of the proof is trivial. □

Lemma 3.2 shows if (ω_k, λ_k) is an $O(\epsilon)$ approximation of (ω^*, λ^*), then d_k is also an $O(\epsilon)$ approximation of d^*. Since d^* satisfies assumptions of Theorems 3.2 and 3.3, so does d_k. Now we give our algorithm.

Algorithm 3.2. Implicit determinant method for non-smooth case 2

Require: initial guess ω_0, λ_0, $A(\omega)$, $A'(\omega)$, tolerance **tol**.
Ensure: approximate solution ω_k, λ_k of eigenvalue optimization problem (1.1).
1: $k = 0$, $r_k = 1$.
2: **while** $r_k >$ **tol do**
3: Solve the linear system (3.17) to get $X(\omega_k, \lambda_k)$;
4: **if** $X(\omega_k, \lambda_k)^{\mathrm{H}} A'(\omega_k) X(\omega_k, \lambda_k)$ is definite **then**
5: **return** Can not converge to a local extreme point with this initial point.
6: **break**
7: **end if**
8: Choose d_k such that $d_k^{\mathrm{H}} X(\omega_k, \lambda_k)^{\mathrm{H}} A'(\omega_k) X(\omega_k, \lambda_k) d_k = 0$. Restrict the first component of d_k to be nonnegative and $\|d_k\| = 1$;
9: With d_k, solve Eqs. (3.5), (3.9) and (3.10) to get $\boldsymbol{f}(\omega_k, \lambda_k)$ and $J(\omega_k, \lambda_k)$;
10: $\begin{bmatrix} \omega_{k+1} \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} \omega_k \\ \lambda_k \end{bmatrix} - J(\omega_k, \lambda_k)^{-1} \boldsymbol{f}(\omega_k, \lambda_k)$;
11: $r_k = \|\boldsymbol{f}(\omega_k, \lambda_k)\|$, $k = k + 1$;
12: **end while**

Theorem 3.4. *Under the Assumptions* 3.1 *and* 3.2*, Algorithm* 3.2 *converges locally quadratically.*

Proof. Let (ω_k, λ_k) be an $O(\epsilon)$ approximation of (ω^*, λ^*). At the k_{th} step, we treat (ω_k, λ_k) and d_k as the initial point, then by the analysis of Algorithm 3.1, $(\omega_{k+1}, \lambda_{k+1})$ is an $O(\epsilon^2)$ approximation of (ω^*, λ^*). We can conclude from Lemma 3.2 that d_{k+1} is also an $O(\epsilon^2)$ approximation of d^*. Thus our method generates the sequence (ω_k, λ_k) and d_k, which converge to (ω^*, λ^*) and d^* both at a quadratic rate. □

3.3. Possible extension to the case where multiplicity is higher than 2

In this subsection, we will discuss some possible extensions to the case where λ^* is an eigenvalue of $A(\omega^*)$ with multiplicity $m > 2$.

In this case, Eq. (3.5) is singular at (ω^*, λ^*). A natural thought is to consider the linear equation system

$$M(\omega, \lambda) \begin{bmatrix} x(\omega, \lambda) \\ \boldsymbol{f}(\omega, \lambda) \end{bmatrix} \equiv \begin{bmatrix} A(\omega) - \lambda I & C \\ C^{\mathrm{H}} & 0 \end{bmatrix} \begin{bmatrix} x(\omega, \lambda) \\ \boldsymbol{f}(\omega, \lambda) \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ d \end{bmatrix}, \tag{3.23}$$

where $d \in \mathbb{C}^m$ is a nonzero constant vector, $C \in \mathbb{C}^{n \times m}$ is a constant matrix. Analogous to the analysis in subsection 3.1, we can choose C such that $M(\omega, \lambda)$ is nonsingular in a neighbor of (ω^*, λ^*), then we solve $\boldsymbol{f}(\omega, \lambda) = 0$ to get (ω^*, λ^*). The number of equations m are larger than the number of unknowns 2. The subsequent analysis is more complex and difficult.

Actually, in practice, Eq. (3.5) is always nonsingular when $(\omega, \lambda) \neq (\omega^*, \lambda^*)$. When (ω, λ) is very close to (ω^*, λ^*), i.e., the last one or two iteration step before convergence, though the condition number of the coefficient matrix becomes very large, we still solve Eq. (3.5) numerically. As shown in Example 4.3, Algorithm 3.2 still works and the convergence rate looks more than 2. This is probably because when Eq. (3.5) is nearly singular, the computed result is a good approximation of the null space. Strict analysis will be considered in other places.

4. Numerical Experiments

We now describe four numerical experiments to give insight into Algorithms 2.1, 3.1–3.2 and their performance. Examples 4.1 and 4.2 show our algorithms work and converge locally quadratically. Example 4.3 shows the behaviour of our algorithm for the case that λ^* is an eigenvalue of $A(\omega^*)$ with multiplicity higher than 2. Example 4.4 shows the difference between Algorithms 3.1 and 3.2. The numerical experiments were performed in MATLAB Version 9.7.0.1165820(R2019b) on a standard desktop with the following specifications: Dell DESKTOP-725VTKL with Intel(R) Core(TM) i5-9500 CPU @ 3.00GHz 8GB RAM.

The Crawford number $\gamma(B)$ of a matrix $B \in \mathbb{C}^{n \times n}$ is defined as the distance of its numerical range $\mathcal{F}(B)$ from zero

$$\gamma(B) = \min\{|z| : z \in \mathcal{F}(B)\}, \quad \mathcal{F}(B) = \{v^{\mathrm{H}} B v : v \in \mathbb{C}, \|v\| = 1\}. \tag{4.1}$$

Cheng and Higham [3] found that (4.1) is a univariate optimization problem

$$\gamma(B) = \max\Big\{ \max_{\omega \in [0, 2\pi]} \lambda_{\min}(A(\omega)), 0 \Big\}, \tag{4.2}$$

where

$$A(\omega) = S\cos(\omega) + K\sin(\omega), \tag{4.3}$$

$S = (B + B^{\mathrm{H}})/2$ and $K = (B - B^{\mathrm{H}})/2\imath$ are Hermitian matrices, $\imath$ is the imaginary unit.

Example 4.1. This example is used to demonstrate that the sequence generated by Algorithm 2.1 converges to a local extreme point locally quadratically. We consider the tridiagonal matrix B as in [15, 17], where

$$B = \mathrm{tridiag}\begin{pmatrix} \imath & \imath & \dots & \imath \\ 1 & 1 & a_3 & \dots & a_n \\ \imath & \imath & \dots & \imath \end{pmatrix} + 0.5\imath \cdot I_n \quad \text{with} \quad a_j = 2 + \frac{j}{n}, \quad j = 3, \dots, n, \tag{4.4}$$

with $n = 120$. Fig. 4.1 shows some eigenvalue curves of $A(\omega)$, where $A(\omega)$ is defined by Eq. (4.3). As shown in Fig. 4.1, λ^* is the Crawford number of matrix B. The maximum of $\lambda_n(A(\omega))$ is attained at $\omega^* = 0$, and $\lambda^* = 1$ is an eigenvalue of $A(\omega^*)$ with multiplicity 2. $\lambda_n(A(\omega))$ is non-smooth at ω^*. ω_s is a local extreme point of $\lambda_{n-1}(A(\omega))$, λ_s is a simple eigenvalue of $A(\omega_s)$, and $\lambda_{n-1}(A(\omega))$ is analytic at ω_s.

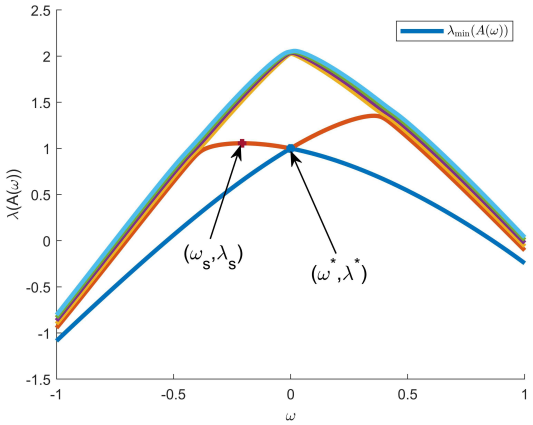

Fig. 4.1. Some eigenvalue curves of $A(\omega)$, $n = 120$.

In this example, we use Algorithm 2.1 to maximize

$$\varphi(\omega) = \lambda_{n-1}(A(\omega)), \quad \omega \in [-0.5, 0], \tag{4.5}$$

with initials

$$\omega_0 = -0.2, \quad \lambda_0 = \varphi(\omega_0).$$

We set c be the eigenvector of $A(\omega_0)$ corresponding to λ_0 in Eq. (2.1), such that Eq. (2.1) is nonsingular by Proposition 2.1. Table 4.1 shows the computed results of Example 4.1. We observed local quadratic convergence of Algorithm 2.1.

Example 4.2. In this example, we apply Algorithms 3.1 and 3.2 to compute the Crawford number of the same matrix B, i.e., (4.4), as in Example 4.1. We use Algorithms 3.1, 3.2 to maximize

$$\varphi(\omega) = \lambda_n(A(\omega)), \quad \omega \in [-1, 1], \tag{4.6}$$

Table 4.1: Results of Algorithm 2.1 for Example 4.1, $n = 120$, starting with $\omega_0 = -0.2$, $\lambda_0 = \varphi(\omega_0)$.

k	λ_k	ω_k	r_k
0	1.055691712763221	-0.200000000000000	2.3e-02
1	1.055858416183899	-0.207367720148854	3.5e-04
2	1.055774284941694	-0.207261997306516	1.1e-07
3	1.055774267042194	-0.207261963683489	1.0e-15

and $A(\omega)$ is defined by Eq. (4.3). As stated in Example 4.1, λ^* is the Crawford number of matrix B. The maximum of $\varphi(\omega) = \lambda_n(A(\omega))$ is attained at $\omega^* = 0$, and $\lambda^* = 1$ is an eigenvalue of $A(\omega^*)$ with multiplicity 2. $\varphi(\omega)$ is non-smooth at ω^*.

In Algorithms 3.1 and 3.2, we set initials as

$$\omega_0 = -0.2, \quad \lambda_0 = \varphi(\omega_0).$$

We set $C = [v_{n-1}(\omega_0), v_n(\omega_0)]$ in equation (3.5), whose columns are eigenvecotrs of $A(\omega_0)$ corresponding to $\lambda_{n-1}(\omega_0)$ and $\lambda_n(\omega_0)$ respectively, such that Eq. (3.5) is nonsingular by Lemma 3.1. In Eq. (3.5) of Algorithm 3.1, we set $d = [1, 2]^{\mathrm{T}}$.

Table 4.2 shows the computed results of Example 4.2. We observed that both Algorithms 3.1 and 3.2 converge locally quadratically. With the same starting point $\omega_0 = -0.2$, $\lambda_0 = \varphi(\omega_0)$,

Table 4.2: Results of Algorithms 3.1, 3.2 and 2.1 for Example 4.2, $n = 120$, starting with same initals $\omega_0 = -0.2$, $\lambda_0 = \varphi(\omega_0)$.

Algorithm	k	λ_k	ω_k	r_k
Algorithm 3.1	0	0.664750682012569	-0.200000000000000	7.8e-01
	1	1.066182489769177	0.043801545553132	1.8e-01
	2	1.001541295371960	0.001560381871468	4.7e-03
	3	1.000001854813342	0.000001551099389	5.3e-06
	4	1.000000000001885	0.000000000001652	5.5e-12
	5	1.000000000000000	0.000000000000000	2.5e-16
Algorithm 3.2	0	0.664750682012569	-0.200000000000000	3.9e-01
	1	1.035330809265056	0.097643817489541	1.0e-01
	2	1.006327673231500	0.003527613694800	7.2e-03
	3	1.000010556307111	0.000009977293151	1.4e-05
	4	1.000000000076651	0.000000000062189	9.6e-11
	5	1.000000000000000	0.000000000000000	1.1e-16
Algorithm 2.1	0	0.664750682012569	-0.200000000000000	4.2e-01
	1	1.040118579251858	0.041483466382649	5.8e-01
	2	1.017445576406488	-0.029097446053152	4.5e-01
	3	1.104029592625749	-0.233351595680445	1.07e-01
	4	1.056786286550737	-0.211075583121032	1.2e-02
	5	1.055797222558788	-0.207297767443690	1.2e-04
	6	1.055774269089027	-0.207261967306892	1.2e-08
	7	1.055774267042192	-0.207261963683486	1.0e-15

and let $c = v_{n-1}(\omega_0)$ in Eq. (2.1), if we apply Algorithm 2.1 to maximize $\varphi(\omega) = \lambda_n(A(\omega))$, sequence generated by Algorithm 2.1 will converge to (ω_s, λ_s) in Fig 4.1, which has no relation with (ω^*, λ^*), see Table 4.2. This example shows that the generalization in Section 3 is necessary.

Example 4.3. This example is used to show the behaviour of our algorithm in Section 3 for the case that λ^* is an eigenvalue of $A(\omega^*)$ with multiplicity higher than 2. We consider the computation of the Crawford number of matrix $B = S + \imath K$ as in [10], where (S, K) are generated by the subroutine GETMAT of [5] with parameter KPAR = 1. The size of matrix B is $n = 25$. Fig. 4.2 shows all eigenvalue curves of $A(\omega)$, where $A(\omega)$ is defined by Eq. (4.3). As shown in Fig. 4.2, λ^* is the Crawford number of matrix B. The maximum of $\varphi(\omega)$ is attained at $\omega^* = \frac{1}{4}\pi$ and the multiplicity of $\lambda^* = \sqrt{2}$ is the same with the size of B.

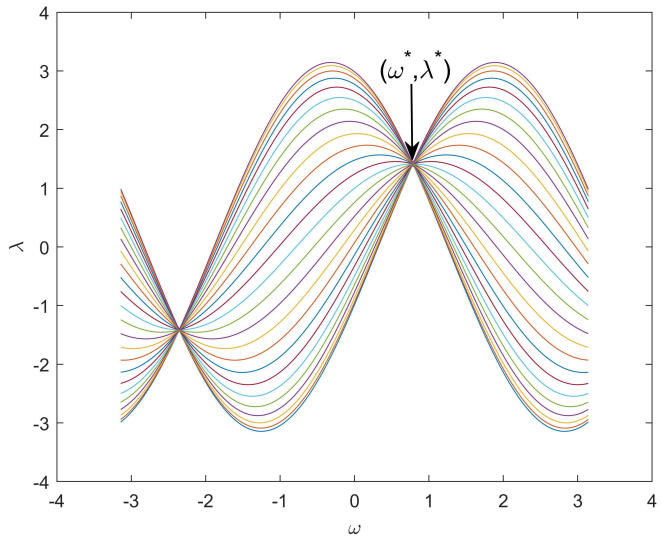

Fig. 4.2. All eigenvalue curves of $A(\omega)$ of Example 4.3, $n = 25$.

We use Algorithm 3.2 to maximize

$$\varphi(\omega) = \lambda_n(A(\omega)), \tag{4.7}$$

with initials $\omega_0 = 0,\ \lambda_0 = \varphi(\omega_0)$. Let

$$i_1 = \arg \max_{i=1,\ldots,n} v_i(\omega_0)^{\mathrm{H}} A'(\omega_0) v_i(\omega_0) \quad \text{and} \quad i_2 = \arg \min_{i=1,\ldots,n} v_i(\omega_0)^{\mathrm{H}} A'(\omega_0) v_i(\omega_0),$$

where $v_i(\omega_0)$, $i = 1, \ldots, n$, are eigenvectors of $A(\omega_0)$ corresponding to n eigenvalues. Then we set

$$C = [v_{i_1}(\omega_0), v_{i_2}(\omega_0)].$$

Table 4.3 shows the computation result of Example 4.3. The last column of Table 4.3 shows condition number of the coefficient matrix $M(\omega_k, \lambda_k)$ in Eq. (3.5). Though condition number of the coefficient matrix, i.e., $\kappa(M(\omega_k, \lambda_k))$, becomes very large when (ω, λ) is quite close to (ω^*, λ^*), numerical results in Table 4.3 shows that our algorithm also works and the convergence rate looks more than 2.

Table 4.3: Results of Algorithm 3.2 for Example 4.3, k=1, starting with $\omega_0 = 0$, $\lambda_0 = \varphi(\omega_0)$, $n = 25$.

k	λ_k	ω_k	r_k	$\kappa(M(\omega_k, \lambda_k))$
0	2.985417748196108	0	1.9e+00	96
1	2.000000000000000	1.000000000000000	0.8e+00	53
2	1.447415443253716	0.782041901539138	3.5e-02	43
3	1.414221527610335	0.785398175999702	7.9e-06	1e+05
4	1.414213562373095	0.785398163397448	0	1e+17

Example 4.4. This example is used to show that Algorithm 3.1 may converge to a point which is not an extreme point, while Algorithm 3.2 will not. We construct an Hermitian matrix $A(\omega)$ artificially as follows: Let $D(\omega) \in \mathbb{R}^{n\times n}$ be a diagonal matrix,

$$D(\omega) = \operatorname{diag}\left(\frac{\omega^2 - 1.5^2}{2}, \frac{(\omega-3)^2 - 1.5^2}{2}, 4(\omega-1.5)^2 - 2, d_4(\omega), \ldots, d_n(\omega)\right), \tag{4.8}$$

where $d_j(\omega) = -3j + 2j\sin(\omega) - 2$, $j = 4, \ldots, n$. Let

$$v(\omega) = \left(\cos(\omega), \sin(\omega)\cos(\omega), \sin^2(\omega)\cos(\omega), \ldots, \sin^{n-2}(\omega)\cos(\omega), \sin^{n-1}(\omega)\right)^{\mathrm{T}}. \tag{4.9}$$

Construct the Householder matrix

$$V(\omega) = I - 2\frac{v(\omega)v(\omega)^{\mathrm{T}}}{\|v(\omega)\|^2}. \tag{4.10}$$

Let

$$A(\omega) = V(\omega)D(\omega)V(\omega)^{\mathrm{T}}.$$

We consider minimizing

$$\varphi(\omega) = \lambda_1(\omega), \quad \omega \in [0, 3].$$

As shown in Fig. 4.3, the local minimum of $\lambda_1(A(\omega))$ is attained at $\omega^* = 1.5$, and $\lambda^* = 0$ is an eigenvalue of $A(\omega^*)$ with multiplicity 2. $\lambda_m = 2$ is an eigenvalue of $A(\omega_m)$ with multiplicity 2, but $\omega_m = 2.5$ is not an extreme point of $\lambda_1(\omega)$.

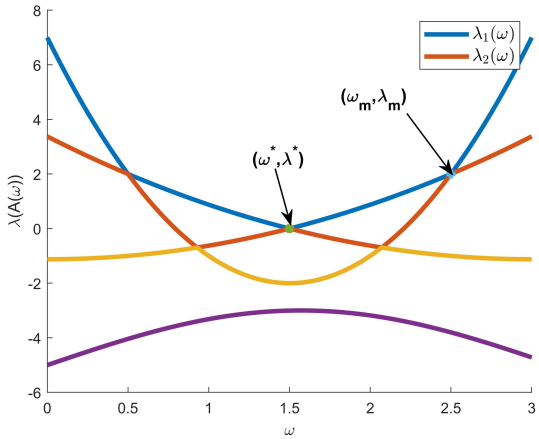

Fig. 4.3. Some eigenvalue curves of $A(\omega)$ of Example 4.4.

In Algorithm 3.1, we set the initials as

$$\omega_0 = 2.1, \quad \lambda_0 = \varphi(\omega_0).$$

We set $C = [v_1(\omega_0), v_2(\omega_0)]$, where $v_i(\omega)$, $i = 1, 2$ are eigenvectors of $A(\omega_0)$ corresponding to $\lambda_1(\omega_0)$ and $\lambda_2(\omega_0)$, such that Eq. (3.5) is nonsingular by Lemma 3.1. We set $d = [1, 1]^{\mathrm{T}}$. Table 4.4 shows that, with the initials $\omega_0 = 2.1$, $\lambda_0 = \varphi(\omega_0)$, sequence generated by Algorithm 3.1 converges to the point (ω_m, λ_m), which is not a local extreme point. While the matrix $X(\omega_0, \lambda_0)^{\mathrm{T}} A'(\omega_0) X(\omega_0, \lambda_0)$ is definite, thus Algorithm 3.2 can not go on. If we choose the initials as $\omega_0 = 2$, $\lambda_0 = \varphi(\omega_0)$, and set $C = [v_1(\omega_0), v_2(\omega_0)]$, then $X(\omega_0, \lambda_0)^{\mathrm{T}} A'(\omega_0) X(\omega_0, \lambda_0)$ is indefinite, sequence generated by Algorithm 3.2 converges to (ω^*, λ^*). Table 4.4 shows that the sequence generated by Algorithm 3.2 satisfies $x(\omega_k, \lambda_k)^{\mathrm{H}} A'(\omega_k) x(\omega_k, \lambda_k) = 0$, which ensures the computed result is a local extreme point.

For the same example, we compare the iterations and CPU time of our algorithms with the subspace method proposed by Kangal *et al.* [14] and the support based algorithm proposed by Kangal and Mengi [15] in Table 4.5. The subspace method and the support based algorithm are methods of global convergence. The codes of these two methods are available in [14] and [15] respectivly. We set the initial point $\omega_0 = 2$ in Algorithm 3.2. In subspace method, we set the initial point as $\omega_0 = 1.8$ and initial lower bound and upper bound for ω as [1.4,1.8]. In support based method we set the initial lower bound and upper bound for ω as [1.4,1.8]. All these methods get the extreme point $\omega^* = 1.5$. We have carried out dozens of runs for each of these algorithms in order to obtain average CPU time. Table 4.5 shows Algorithm 3.2 is faster than other two algorithms.

Table 4.4: Results of Algorithm 3.1 for Example 4.4, $n = 100$, starting with $\omega_0 = 2.1$, $d = [1, 1]^{\mathrm{T}}$.

Algorithm	k	λ_k	ω_k	r_k	$x_k^{\mathrm{H}} A'(\omega_k) x_k$
Algorithm 3.1	0	1.080000000000020	2.100000000000000	1.6e+01	7.52
	1	2.546568695960197	2.707407407407380	2.1e+01	29.7
	2	2.100219061263431	2.566915516731857	6.2e-01	17.6
	3	2.007935528354060	2.507030591811159	6.4e-02	14.7
	4	2.000053839808336	2.500074321596725	7.2e-04	14.4
	5	2.000000003059168	2.500000008372691	8.7e-08	14.4
	6	2.000000000000000	2.500000000000000	9.0e-16	14.4
Algorithm 3.2	0	0.874999999999886	2.000000000000000	1.2e+01	4.3e-15
	1	-0.018907286209513	1.485501574144193	1.2e-01	-4.8e-15
	2	0.001872914254212	1.500576012131182	4.9e-04	4e-16
	3	0.000000129980282	1.500000033407006	7.0e-07	0
	4	-0.000000000000001	1.500000000000000	0	-1e-16

Table 4.5: Comparison of three methods for Example 4.4.

Algorithm	iter	time(s) ($n = 10^2$)	iter	time(s) ($n = 10^3$)	iter	time(s) ($n = 2 * 10^3$)
Algorithm 3.2	5	0.007	5	0.8	5	7
Support function	6	0.02	6	3.2	6	27
Subspace method	6	0.04	6	3.3	4	15

5. Conclusions

In this paper, we generalize the IDM such that it can solve a univariant eigenvalue optimization problem (1.1). We also generalize the IDM for the non-smooth case. IDM converges quadratically locally for both smooth and non-smooth case under reasonable assumptions. Numerical experiments confirm the theoretical analysis and the efficiency of the algorithm. We give the analysis of the case where the eigenvalue is of multiplicity 2 at the extreme point. For the cases that the multiplicity is larger than 2, numerical experiments show that our method still works, and needs further study.

Acknowledgments. The work of the authors was supported by the China NSF Project (No. 11971122).

References

[1] 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**:1 (1990), 1–7.

[2] R. Byers, A bisection method for measuring the distance of a stablematrix to the unstable matrices, *SIAM J. Sci. Statist. Comput.*, **9**:5 (1988), 875–881.

[3] S.H. Cheng and N. J. Higham, The nearest definite pair for the Hermitian generalized eigenvalue problem, *Linear Algebra Appl.*, **302–303**:1 (1999), 63–76.

[4] C.R. Crawford, A stable generalized eigenvalue problem, *SIAM J. Numer. Anal.*, **13**:6 (1976), 854–860.

[5] C.R. Crawford, Algorithm 646 pdfind: A routine to find a positive definite linear combination of two real symmetric matrices, *ACM Trans. Math. Software*, **12**:3 (1986), 278–282.

[6] M.A. Freitag and A. Spence, A newton-based method for the calculation of the distance to instability, *Linear Algebra Appl.*, **435**:12 (2011), 3189–3205.

[7] M.A. Freitag, A. Spence, and P. Van Dooren, Calculating the H_∞-norm using the implicit determinant method, *SIAM J. Matrix Anal. Appl.*, **35**:2 (2014), 619–635.

[8] 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**:10 (2013), 968–971.

[9] I. Gohberg, P. Lancaster, and L. Rodman, *Matrix Polynomials*, Society for Industrial and Applied Mathematics, 2009.

[10] C.H. Guo, N.J. Higham, and F. Tisseur, An improved arc algorithm for detecting definite Hermitian pairs, *SIAM J. Matrix Anal. Appl.*, **31**:3 (2009), 1131–1151.

[11] C. Helmberg and F. Rendl, A spectral bundle method for semidefinite programming, *SIAM J. Optim.*, **10**:3 (2000), 673–696.

[12] N.J. Higham, F. Tisseur, and P. M. Van Dooren, Detecting a definite Hermitian pair and a hyperbolic or elliptic quadratic eigenvalue problem, and associated nearness problems, *Linear Algebra Appl.*, **351–352**:15 (2002) 455–474.

[13] D. Hinrichsen and A.J. Pritchard, *Mathematical Systems Theory I: Modelling, State Space Analysis, Stability and Robustness*, Springer, 2005.

[14] F. Kangal, K. Meerbergen, E. Mengi, and W. Michiels, A subspace method for large-scale eigenvalue optimization, *SIAM J. Matrix Anal. Appl.*, **39**:1 (2018), 48–82.

[15] F. Kangal and E. Mengi, Nonsmooth algorithms for minimizing the largest eigenvalue with applications to inner numerical radius, *IMA J. Numer. Anal.*, **40**:4 (2020), 2342–2376.

[16] T. Kato, *Perturbation Theory for Linear Operators*, Springer, 1995.

[17] D. Kressner, D. Lu, and B. Vandereycken, Subspace acceleration for the Crawford number and related eigenvalue optimization problems, *SIAM J. Matrix Anal. Appl.*, **39**:2 (2018), 961–982.

[18] P. Lancaster, On eigenvalues of matrices dependent on a parameter, *Numer. Math.*, **6** (1964), 377–387.
[19] A.S. Lewis and M.L. Overton, Eigenvalue optimization, *Acta Numer.* , **5** (1996), 149–190.
[20] E. Mengi, E.A. Yildrim, and M. Kilic, Numerical optimization of eigenvalues of Hermitian matrix functions, *SIAM J. Matrix Anal. Appl.*, **35**:2 (2014), 699–724.
[21] M.L. Overton, On minimizing the maximum eigenvalue of a symmetric matrix, *SIAM J. Matrix Anal. Appl.*, **9**:2 (1988), 256–268.
[22] F. Rellich, *Perturbation Theory of Eigenvalue Problems*, Gordon and Breach Science Publishers, 1969.
[23] A. Spence and C. Poulton, Photonic band structure calculations using nonlinear eigenvalue techniques, *J. Comput. Phys.*, **204**:1 (2005), 65-81.
[24] C. F. Van Loan, How near is a stable matrix to an unstable matrix?, *Contemp. Math.*, **47** (1985), 465–478.
[25] Z.Q. Luo, W.K. Ma, A.M.-C. So, Y. Ye and S. Zhang, Semidefinite relaxation of quadratic optimization problems, *IEEE Signal Process. Mag.*, **27**:3 (2010), 20–34.