A NEWTON-TYPE METHOD WITH NON-EQUIVALENCE DEFLATION FOR NONLINEAR EIGENVALUE PROBLEMS ARISING IN PHOTONIC CRYSTAL MODELING

TSUNG-MING HUANG*AND WEN-WEI LIN†AND VOLKER MEHRMANN‡

Abstract. The numerical simulation of the band structure of three-dimensional dispersive metallic photonic crystals with face-centered cubic lattices leads to large-scale nonlinear eigenvalue problems, which are very challenging due to a high dimensional subspace associated with the eigenvalue zero and the fact that the desired eigenvalues (with smallest real part) cluster near the zero eigenvalues. For the solution of the eigenvalue problem, a Newton-type iterative method is proposed and the nullspace-free method is applied to exclude the zero eigenvalues from the associated generalized eigenvalue problem. To find the successive eigenvalue/eigenvector pairs, we propose a new non-equivalence deflation method to transform converged eigenvalues to infinity, while all other eigenvalues remain unchanged. The deflated problem is then solved by the same Newton-type method, which uses a hybrid method that combines the Jacobi-Davidson, the shift-invert residual Arnoldi and nonlinear Arnoldi methods to compute the clustered eigenvalues. Numerical results illustrate that the method is robust even for the case of computing many eigenvalues in very large problems.

Key words. Maxwell equation, dispersive metallic photonic crystals, nonlinear eigenvalue problem, Newton-type method, non-equivalence deflation, Jacobi-Davidson method, shift-invert residual Arnoldi method, nonlinear Arnoldi method

AMS subject classifications. 15A18, 15A90, 65F15

1. Introduction.

The electromagnetic wave propagation through dispersive metallic photonic crystals (PCs) has been extensively studied over the past few decades [8, 9, 10, 34, 38, 46]. A standard model to study the electromagnetic effects in periodic structures and dispersive isotropic materials is the three-dimensional (3D) Maxwell equation

$$\nabla \times \nabla \times E(\mathbf{r}) = \omega^2 \varepsilon(\mathbf{r}, \omega) E(\mathbf{r}), \tag{1.1}$$

where $E(\mathbf{r})$ denotes the electric field at position $\mathbf{r} \in \mathbb{R}^3$ and $\varepsilon(\mathbf{r}, \omega)$ denotes the permittivity, which is dependent on the position $\mathbf{r}$ and the frequency ω.

In the Drude model of a dispersive material [34, 38, 51, 52], the permittivity $\varepsilon(\mathbf{r}, \omega)$ is modeled as

$$\varepsilon(\mathbf{r}, \omega) = 1 - \frac{\omega_p^2}{\omega^2 + \imath \Gamma_p \omega}, \tag{1.2}$$

for $\mathbf{r}$ in the material domain, and $\varepsilon(\mathbf{r}, \omega) = \varepsilon_n$ (constant) otherwise, where $\imath = \sqrt{-1}$, ω_p is a plasma frequency, and Γ_p is the corresponding damping frequency. The more involved Drude-Lorentz model [10, 11, 34, 46] uses the permittivity model

$$\varepsilon(\mathbf{r}, \omega) = \varepsilon_\infty - \frac{\omega_p^2}{\omega^2 + \imath \Gamma_p \omega} + \sum_{j=1}^{2} \Omega_j A_j \left(\frac{e^{\imath \phi_j}}{\Omega_j - \omega - \imath \Gamma_j} + \frac{e^{-\imath \phi_j}}{\Omega_j + \omega + \imath \Gamma_j} \right), \tag{1.3}$$

[1]Department of Mathematics, National Taiwan Normal University, Taipei 116, Taiwan. E-mail: min@ntnu.edu.tw

[2]Department of Applied Mathematics, National Chiao Tung University, Hsinchu 300, Taiwan. E-mail: wwlin@math.nctu.edu.tw

[3]Institut für Mathematik, MA 4-5, TU Berlin, Straße des 17. Juni 136, D-10623 Berlin, Germany. E-mail: mehrmann@math.tu-berlin.de

for $\mathbf{r}$ in the material domain and $\varepsilon(\mathbf{r},\omega) = \varepsilon_n$ (constant) otherwise, where ε_∞ is a high-frequency limit dielectric constant, ϕ_j, Ω_j, A_j and Γ_j are the parameters for the two pairs of poles in the Lorentz model. The first and second term in (1.3) correspond to the contributions of a Drude model with $\varepsilon_\infty = 1$. The third term arises from the interband transitions.

There are many approaches for the numerical simulation of wave propagation in photonic crystals. Very often, time-domain simulation methods, see e.g. [12], are used to calculate the band structures of 3D metallic PCs, but in order to obtain reasonable results extremely long simulation times are required. As an alternative, often Fourier transform and the discretization of the time-invariant system (1.1) is considered, which leads to a nonlinear eigenvalue problem (NLEVP) [8, 9, 34], which is rational in the frequency ω. To solve large scale NLEVPs, however, is also a non-trivial task [37, 45], and is particularly challenging for 3D metallic PCs. In [8, 9, 34], the rational eigenvalue problem is reformulated as a polynomial eigenvalue problems (PEPs) by multiplying with the common denominator. The PEP is then reformulated (linearized) [13, 35] as a generalized eigenvalue problem to which standard eigenvalue methods [2, 32, 39] can be applied. However, in this approach the order of the problem is highly enlarged and the sensitivity of the eigenvalues and eigenvectors may increase considerably, since the set of admissible perturbations for the linearized eigenvalue problem is larger than that of the PEP [44] and, even more of the rational problem.

A completely different eigenvalue method is to work directly with the NLEVP. One can use the polynomial Jacobi-Davidson method [21, 24, 41, 42, 49] for the PEPs or the rational Krylov method [26, 40], nonlinear Arnoldi method [47], contour integrals [1, 4], and other methods in [7, 25, 29, 33, 41, 48] to solve the general NLEVP. In [5, 7] a very promising new method for PEPs is suggested that computes full invariant pairs but in Jacobi-Davidson or Newton type methods, often only one eigenvalue/eigenvector pair is determined at a time, e.g. when the algorithms in [21, 24, 26, 33, 40, 42, 47, 47, 48, 49] are applied to solve the NLEVP. One way then to compute several successive eigenpairs is to deflate converged eigenvalue/eigenvectors from the NLEVP. In [15], an explicit non-equivalence low-rank deflation method is proposed for computing the smallest real eigenvalues of a special quadratic eigenvalue problem. Once the smallest positive eigenvalue is obtained, it is then transformed to zero by the deflation scheme, while all other eigenvalues remain unchanged. The next successive eigenvalue thus becomes the smallest positive eigenvalue of the transformed problem, which is then again solved by the proposed method. The concept of non-equivalence deflation is also applied to solve special quadratic problems in [20, 23], and cubic polynomial eigenvalue problems [22, 49]. One of the differences of the non-equivalence deflation scheme in [15] and [20, 22, 23, 49] is that the convergent eigenvalue is transformed to infinity rather than 0.

Almost all currently available eigenvalue methods have large difficulties when multiple eigenvalues occur, because then the convergence of Newton-type methods deteriorates and in this case also the condition number of the problem typically is extremely large, so that small perturbations lead to large errors. In this case it is necessary to consider either block oriented methods [3, 5, 7, 36] which currently are designed for quadratic and polynomial problems, or in the more general nonlinear case, one needs to employ specially designed deflation techniques as the one we will discuss in this paper.

We consider large, sparse NLEVPs of the form

$$A\mathbf{x} = \omega^2 B(\omega)\mathbf{x}, \tag{1.4}$$

arising from a 3D dispersive metallic PC with a face-centered cubic lattice, where A, $B(\omega)$ are large, sparse matrices. This problem has a large number of zero eigenvalues and most of the desired eigenvalues are clustered near the zero eigenvalues so that it becomes an extremely challenging problem. To deal with this challenge, we propose a Newton-type iterative method with a special non-equivalence deflation scheme to solve (1.4).

Starting from a sample frequency ω_k, we solve the generalized eigenvalue problem (GEP)

$$\beta A\mathbf{x} = (\omega_k B(\omega_k))\,\mathbf{x}.$$

We apply the nullspace-free method of [17, 19], and transform the GEP to a standard eigenvalue problem (SEP). Both, the SEP and the GEP have the same nonzero finite eigenvalues. Using a computed eigenvalue/eigenvector pair of the SEP, a novel Newton-type scheme is presented to update the starting frequency ω_k. The quadratic convergence of the method is illustrated via numerical examples. We also propose a novel non-equivalence low-rank deflation scheme to transform a convergent eigenvalue to infinity, while the other eigenvalues remain unchanged. The deflated NLEVP is then again solved by the same method. To make this method practical, we discuss some strategies for determining ω_0, initial vectors and stopping tolerances for the solution of the SEP, which reduce the computational cost and accelerate the convergence. Furthermore, a hybrid eigenvalue method is proposed to compute the desired eigenvalue clusters.

This paper is organized as follows. In Section 2, we briefly derive the NLEVP and the new non-equivalence low-rank deflation method. In Section 3, we introduce the Newton-type method. The Jacobi-Davidson, the shift-invert residual Arnoldi and the nonlinear Arnoldi methods to solve the SEP and NLEVP, respectively, are reviewed in Section 4. Some practical implementations to reduce the computational cost are proposed in Section 5. Numerical experiments to validate the robustness of the proposed schemes are demonstrated in Section 6. We conclude the paper in Section 7.

2. Nonlinear eigenvalue problems. In this section, we first introduce the resulting NLEVP by using the Yee scheme [50] for the discretization of the Maxwell equation (1.1). Then we propose a non-equivalence deflation scheme which allows to transform the resulting NLEVP with eigenvalue/eigenvector pair $(\mu, \mathbf{x})$ to a new NLEVP with the same eigenvalues except that μ is replaced by infinity. Both the original and the new NLEVP have the form

$$A\mathbf{x} = \omega\widehat{B}(\omega)\mathbf{x} \tag{2.1}$$

In the next section, we will then develop a Newton-type method to solve (2.1) so that we can intertwine this method and the non-equivalence deflation scheme to compute the desired eigenvalue/eigenvector pairs.

Based on the Bloch Theorem [28], we aim to find the Bloch eigenfunctions $E(\mathbf{r})$ for (1.1) satisfying the following quasi-periodicity condition

$$E(\mathbf{r} + \mathbf{a}_\ell) = e^{\imath 2\pi\mathbf{k}\cdot\mathbf{a}_\ell} E(\mathbf{r}),$$

for $\ell = 1, 2, 3$. Here, $2\pi\mathbf{k}$ is the Bloch wave vector in the first Brillouin zone [27] and the vectors $\mathbf{a}_\ell$ are the lattice translation vectors that span the primitive cell, which

are extended periodically to form the dispersive metallic photonic crystal. In this paper, we focus on the face-centered cubic lattice (FCC) vectors, i.e.,

$$\mathbf{a}_1 = \frac{a}{\sqrt{2}}[1,0,0]^T,\ \mathbf{a}_2 = \frac{a}{\sqrt{2}}\left[\frac{1}{2},\frac{\sqrt{3}}{2},0\right]^T,\ \mathbf{a}_3 = \frac{a}{\sqrt{2}}\left[\frac{1}{2},\frac{1}{2\sqrt{3}},\sqrt{\frac{2}{3}}\right]^T$$

with lattice constant a.

Let n_1, n_2, and n_3 with $n = n_1 n_2 n_3$ be the number of grid points in x, y, and z direction, respectively, and let δ_x, δ_y, and δ_z denote the associated grid lengths in x, y, and z axial direction, respectively.

The resulting matrix A arising from the discretized double-curl operator using the Yee scheme [50] on a primitive cell is then of the form [16, 17, 18]

$$A = C^*C \in \mathbb{C}^{3n\times 3n}, \tag{2.2}$$

where

$$C = \begin{bmatrix} 0 & -C_3 & C_2 \\ C_3 & 0 & -C_1 \\ -C_2 & C_1 & 0 \end{bmatrix} \in \mathbb{C}^{3n\times 3n},$$

with

$$C_1 = I_{n_2 n_3} \otimes K_1 \in \mathbb{C}^{n\times n},\ C_2 = I_{n_3} \otimes K_2 \in \mathbb{C}^{n\times n},\ C_3 = K_3 \in \mathbb{C}^{n\times n} \tag{2.3}$$

where $\otimes$ denotes the Kronecker product, see [17] for the detailed definition of the pseudo periodical matrices K_1, K_2, and K_3. The resulting NLEVP then has the form

$$F(\omega)\mathbf{x} \equiv \left(A - \omega^2 B(\omega)\right)\mathbf{x} = 0, \tag{2.4}$$

where A is defined in (2.2) and $B(\omega)$ is a diagonal matrix associated with the permittivity that can be split into the sum of two diagonal matrices

$$B(\omega) = B_n + \varepsilon(\mathbf{r},\omega)B_d, \tag{2.5}$$

where the subscripts n and d indicate the non-dispersive and the dispersive materials, respectively [34], and $\varepsilon(\mathbf{r},\omega)$ is the related permittivity of the dispersive material given in (1.2) and (1.3). The goal of the eigenvalue computation is to compute the eigenvalues of smallest real part and associated eigenvectors.

To conduct the analysis of the NLEVP (2.4) we have the following lemma.

LEMMA 2.1. *Let $\varepsilon(\mathbf{r},\omega)$ be defined as in* (1.2) *or* (1.3). *Then $\overline{\varepsilon(\mathbf{r},\omega)} = \varepsilon(\mathbf{r},-\bar{\omega})$.*

Proof. From (1.2) and (1.3), we directly have

$$\overline{\varepsilon(\mathbf{r},\omega)} = 1 - \frac{\omega_p^2}{\bar{\omega}^2 - \imath\Gamma_p\bar{\omega}} = 1 - \frac{\omega_p^2}{(-\bar{\omega})^2 + \imath\Gamma_p(-\bar{\omega})} = \varepsilon(\mathbf{r},-\bar{\omega})$$

and

$$\begin{aligned}\overline{\varepsilon(\mathbf{r},\omega)} &= \varepsilon_\infty - \frac{\omega_p^2}{(\bar{\omega})^2 - \imath\Gamma_p\bar{\omega}} + \sum_{j=1}^{2}\Omega_j A_j\left(\frac{e^{-\imath\phi_j}}{\Omega_j - \bar{\omega} + \imath\Gamma_j} + \frac{e^{\imath\phi_j}}{\Omega_j + \bar{\omega} - \imath\Gamma_j}\right)\\ &= \varepsilon(\mathbf{r},-\bar{\omega}),\end{aligned}$$

respectively. ∎

With the help of Lemma 2.1 we have the following Theorem.

THEOREM 2.2. *The NLEVP* (2.4) *has the following properties.*

1. *$F(\omega)$ has n zero eigenvalues.*
2. *$F^*(\omega) = F(-\bar{\omega})$, i.e., ω and $-\bar{\omega}$ are eigenvalues of $F(\omega)$.*
3. *If $\mathbf{y}$ is a left eigenvector of $F(\omega)$ associated with the eigenvalue ω, then $\mathbf{y}$ is a right eigenvector associated with the eigenvalue $-\bar{\omega}$.*

Proof. Rewrite (2.4) as

$$A\mathbf{x} = \omega B_1(\omega)\mathbf{x} \quad \text{with} \quad B_1(\omega) = \omega B(\omega). \tag{2.6}$$

Then from (2.5) and the definition of $\varepsilon(\mathbf{r},\omega)$ in (1.2) or (1.3), it follows that $B_1(0) = -\frac{\omega_p^2}{\imath\Gamma_p}B_d$. This implies that the eigenvectors of A associated to the eigenvalue 0 are also the eigenvectors of $F(\omega)$ associated to the eigenvalue 0. Therefore, by Theorem 3.7 in [17], $F(\omega)$ has an n-fold eigenvalue 0.

By the definition of $B(\omega)$ in (2.5) and Lemma 2.1, it follows that

$$\begin{aligned} F^*(\omega) &= A^* - (\bar{\omega})^2\left(B_n + \overline{\varepsilon(\mathbf{r},\omega)}B_d\right) \\ &= A - (-\bar{\omega})^2\left(B_n + \varepsilon(\mathbf{r},-\bar{\omega})B_d\right) \\ &= F(-\bar{\omega}). \end{aligned}$$

This means that if $\mathbf{y}$ is a left eigenvector of $F(\omega)$ associated to ω, then

$$0 = F^*(\omega)\mathbf{y} = F(-\bar{\omega})\mathbf{y},$$

and, therefore, $\mathbf{y}$ is a right eigenvector associated with $-\bar{\omega}$. ∎

Let us assume that each of the pairwise different eigenvalues μ_i has equal algebraic and geometric multiplicity m_i, $i = 1,\ldots,\ell$, and let X_i be a basis of the corresponding nullspace of $F(\mu_i)$. The following non-equivalence deflation allows us to transform the original problem (2.4) to a new NLEVP with the same eigenvalues, except that μ_i is replaced by infinity with multiplicity m_i for $i = 1,\ldots,\ell$. With

$$\widetilde{F}(\omega)\tilde{\mathbf{x}} := \left(F(\omega)\prod_{j=1}^{\ell}\left(I - \frac{\omega}{\omega-\mu_j}X_jX_j^*\right)\right)\tilde{\mathbf{x}}, \tag{2.7}$$

then we have the following theorem.

THEOREM 2.3. *Let $F(\omega)$ and $\widetilde{F}(\omega)$ be defined as in* (2.4) *and* (2.7)*, respectively. Then,*

$$\begin{aligned} &\left\{\omega|\ \widetilde{F}(\omega)\tilde{\mathbf{x}} = 0,\ \tilde{\mathbf{x}} \neq 0\right\} \\ &\quad = \{\omega|\ F(\omega)\mathbf{x} = 0, \mathbf{x} \neq 0\} \setminus \{\mu_1,\cdots,\mu_1,\cdots,\mu_\ell,\cdots,\mu_\ell\} \cup \{\infty\}. \end{aligned}$$

Furthermore, if $(\mu,\tilde{\mathbf{x}})$ is an eigenvalue/eigenvector pair of $\widetilde{F}(\omega)$, then $(\mu,\mathbf{x})$ is an eigenvalue/eigenvector pair of $F(\omega)$ with

$$\mathbf{x} = \prod_{j=1}^{\ell}\left(I - \frac{\mu}{\mu-\mu_j}X_jX_j^*\right)\tilde{\mathbf{x}}. \tag{2.8}$$

Proof. Using the determinant identity $\det(I_n + RS) = \det(I_m + SR)$, where R and S^* are $n \times m$ matrices, we get

$$\begin{aligned}\det\left(\widetilde{F}(\omega)\right) &= \det\left(F(\omega)\right) \prod_{j=1}^{\ell} \det\left(I - \frac{\omega}{\omega - \mu_j} X_j X_j^*\right) \\ &= \det\left(F(\omega)\right) \prod_{j=1}^{\ell} \left(1 - \frac{\omega}{\omega - \mu_j}\right)^{m_j} \\ &= \det\left(F(\omega)\right) \prod_{j=1}^{\ell} \left(\frac{-\mu_j}{\omega - \mu_j}\right)^{m_j}.\end{aligned}$$

Hence, the nonlinear eigenvalue problem (2.7) has the same eigenvalues as (2.4) except that m_j copies of the eigenvalue μ_j are replaced by the eigenvalue infinity. □

For the convenience of computation in our proposed method, we assume that the columns of $X_1, X_2, \ldots, X_\ell$ are forming an orthonormal basis for the corresponding space $\text{span}\{X_1, X_2, \ldots, X_\ell\}$, which can always be assumed using re-orthogonalization. In the following, we use $X_1, X_2, \ldots, X_\ell$ to denote this orthonormal basis. With

$$X = \begin{bmatrix} X_1 & X_2 & \cdots & X_\ell \end{bmatrix},$$

it holds that $X^*X = I_m$ with $m = m_1 + \cdots + m_\ell$. Then, we have

$$\prod_{j=1}^{\ell} \left(I - \frac{\omega}{\omega - \mu_j} X_j X_j^*\right) = I - \sum_{j=1}^{\ell} \frac{\omega}{\omega - \mu_j} X_j X_j^* = I - \omega X D(\omega) X^*, \tag{2.9}$$

where

$$D(\omega) = \text{diag}\left(\frac{1}{\omega - \mu_1} I_{m_1}, \frac{1}{\omega - \mu_2} I_{m_2}, \cdots, \frac{1}{\omega - \mu_\ell} I_{m_\ell}\right).$$

Plugging $F(\omega)$ in (2.4) and (2.9) into (2.7), $\widetilde{F}(\omega)$ can be reformulated in the following simple form

$$\begin{aligned}\widetilde{F}(\omega) &= \left(A - \omega B_1(\omega)\right)\left(I - \omega X D(\omega) X^*\right) \\ &= A - \omega\left[B_1(\omega) + \left(A - \omega B_1(\omega)\right) X D(\omega) X^*\right] \\ &\equiv A - \omega \widetilde{B}(\omega),\end{aligned} \tag{2.10}$$

where $B_1(\omega) = \omega B(\omega)$ is as in (2.6). It follows that the NLEVPs (2.4) and (2.7) can both be represented in the form

$$A\mathbf{x} = \omega \widehat{B}(\omega)\mathbf{x}, \tag{2.11}$$

where $\widehat{B}(\omega)$ is either equal to $\omega B(\omega)$ (for (2.4)) or $\widetilde{B}(\omega)$ (for (2.10)).

In the next subsection, we will develop a Newton-type method for the computation of the desired eigenvalue/eigenvector pairs for general NLEVPs of the form (2.11).

3. Newton-type methods. Based on the Newton-type method suggested in [15], in this section we propose a Newton-type method for computing eigenvalues of NLEVPs of the form (2.11). We rewrite (2.11) as

$$\frac{1}{\omega} A\mathbf{x} = \widehat{B}(\omega)\mathbf{x},$$

and then, for a given ω, we consider the GEP

$$\beta A\mathbf{x} = \widehat{B}(\omega)\mathbf{x}, \tag{3.1}$$

where the eigenvalues β depend on the chosen value of ω. To determine an eigenvalue of (2.11), it is sufficient to find a value ω_* such that the eigenvalue $\beta(\omega_*)$ of (3.1) satisfies the condition $\beta(\omega_*) = \omega_*^{-1}$, which is equivalent to determine a root of the nonlinear equation

$$\beta(\omega) = \omega^{-1}. \tag{3.2}$$

The simplest method to solve this equation is to use a fix-point iteration $\omega_{k+1} = \beta(\omega_k)^{-1}$, so that when it has converged to a value ω_*, then an eigenvalue $\beta(\omega_k)$ of (3.1) with $\omega = \omega_*$ has been computed. But, since the convergence of fix-point iterations is typically linear if it converges, we apply the Newton method

$$\omega_{k+1} = \omega_k - \left(\beta'(\omega_k) + \omega_k^{-2}\right)^{-1}\left(\beta(\omega_k) - \omega_k^{-1}\right) \tag{3.3}$$

to (3.2) to accelerate the convergence.

Before discussing the computation of the derivative $\beta'(\omega)$, we first reduce the GEP (3.1) to a SEP. Suppose that $\widehat{B}(\omega)$ in (3.1) is invertible. In Theorem 3.7 of [17] it has been shown that with $n = n_1n_2n_3$ in the discretization (2.2), the GEP (3.1) has n eigenvalues at infinity. Since we are interested in finding the eigenvalues of (2.11) with smallest positive real part, it means that the eigenvalues β of (3.1) with largest positive real part are of interest. In this respect, the large dimension of the invariant space associated with the eigenvalue infinity in (3.1) leads to several numerical difficulties, see [17]. To address this problem, we apply the nullspace-free method of [17] to solve (3.1). For this we make use of the following theorems.

THEOREM 3.1 ([17]). *Let C_ℓ ($\ell = 1, 2, 3$) be as in (2.3). Then, $C_i^*C_j = C_jC_i^*$, $C_iC_j = C_jC_i$, for $i, j = 1, 2, 3$ and all three matrices C_ℓ can be diagonalized by the same unitary matrix T, i.e.,*

$$C_1T = T\Lambda_x,\ C_2T = T\Lambda_y, \text{ and } C_3T = T\Lambda_z, \tag{3.4}$$

where Λ_x, Λ_y, and Λ_z are diagonal matrices.

THEOREM 3.2 ([17]). *Let A and $(\Lambda_x, \Lambda_y, \Lambda_z, T)$ be defined as in (2.2) and (3.4), respectively. Then there exists a unitary matrix*

$$\begin{bmatrix} Q_0 & Q \end{bmatrix} := (I_3 \otimes T)\begin{bmatrix} \Lambda_0 & \Lambda \end{bmatrix} \equiv (I_3 \otimes T)\left[\begin{array}{c|cc} \ddots & \ddots & \ddots \\ \ddots & \ddots & \ddots \\ \ddots & \ddots & \ddots \end{array}\right],$$

where Λ_0 is a 3×1 block diagonal matrix, and Λ is a 3×2 block diagonal matrix such that

$$\begin{bmatrix} Q_0 & Q \end{bmatrix}^* A \begin{bmatrix} Q_0 & Q \end{bmatrix} = \operatorname{diag}(0, \Lambda_q, \Lambda_q), \tag{3.5}$$

where $\Lambda_q = \Lambda_x^*\Lambda_x + \Lambda_y^*\Lambda_y + \Lambda_z^*\Lambda_z$.

THEOREM 3.3 ([19]). *Let A be as in (2.2), let (Q, Λ) be as in Theorem 3.2, and let ω be such that $\widehat{B}(\omega)$ in (3.1) is nonsingular. Then (denoting by span of a matrix the span of its columns),*

$$\operatorname{span}\widehat{B}(\omega)^{-1}Q\Lambda^{1/2} = \operatorname{span}\left\{\mathbf{x} | A\mathbf{x} = \lambda\widehat{B}(\omega)\mathbf{x},\ \lambda \neq 0\right\},$$

and furthermore,

$$\left\{\lambda \neq 0 \,|\, A\mathbf{x} = \lambda \widehat{B}(\omega)\mathbf{x}\right\} = \left\{\lambda \,|\, \Lambda^{1/2} Q^* \widehat{B}(\omega)^{-1} Q \Lambda^{1/2}\mathbf{u} = \lambda \mathbf{u}\right\}.$$

Using Theorem 3.3, we can transform the GEP (3.1) to the SEP

$$K(\omega)^{-1}\mathbf{u} = \beta \mathbf{u}, \tag{3.6}$$

where

$$K(\omega) = \Lambda^{1/2} Q^* \widehat{B}(\omega)^{-1} Q \Lambda^{1/2},$$

and the GEP (3.1) and the SEP (3.6) have the same nonzero eigenvalues.

To evaluate the derivative $\beta'(\omega)$ in (3.3) we can use the following method. Let $\mathbf{u}(\omega)$ and $\mathbf{v}(\omega)$ with $\mathbf{v}^*(\omega)\mathbf{u}(\omega) = 1$ be the right and the left eigenvectors of $K(\omega)^{-1}$, respectively, corresponding to the eigenvalue $\beta(\omega)$, i.e., $K(\omega)^{-1}\mathbf{u}(\omega) = \beta(\omega)\mathbf{u}(\omega)$ and $\mathbf{v}^*(\omega)K(\omega)^{-1} = \beta(\omega)\mathbf{v}^*(\omega)$. Then

$$\beta(\omega) = \mathbf{v}^*(\omega)K(\omega)^{-1}\mathbf{u}(\omega) \tag{3.7}$$

and

$$\mathbf{v}^*(\omega)'\mathbf{u}(\omega) + \mathbf{v}^*(\omega)\mathbf{u}(\omega)' = 0. \tag{3.8}$$

Using (3.7) and (3.8), and the fact that $(K(\omega)^{-1})' = -K(\omega)^{-1}K(\omega)'K(\omega)^{-1}$, we obtain the following method to compute $\beta'(\omega)$,

$$\begin{aligned}
\beta'(\omega) &= \mathbf{v}^*(\omega)\left(K(\omega)^{-1}\right)'\mathbf{u}(\omega) + \mathbf{v}^*(\omega)'K(\omega)^{-1}\mathbf{u}(\omega) + \mathbf{v}^*(\omega)K(\omega)^{-1}\mathbf{u}(\omega)' \\
&= \mathbf{v}^*(\omega)\left(K(\omega)^{-1}\right)'\mathbf{u}(\omega) + \beta(\omega)\mathbf{v}^*(\omega)'\mathbf{u}(\omega) + \beta(\omega)\mathbf{v}^*(\omega)\mathbf{u}(\omega)' \\
&= \mathbf{v}^*(\omega)\left(K(\omega)^{-1}\right)'\mathbf{u}(\omega) \\
&= -\mathbf{v}^*(\omega)K(\omega)^{-1}K(\omega)'K(\omega)^{-1}\mathbf{u}(\omega) \\
&= -\beta(\omega)^2\mathbf{v}^*(\omega)K(\omega)'\mathbf{u}(\omega) \\
&= -\beta(\omega)^2\mathbf{v}^*(\omega)\Lambda^{1/2}Q^*\left[\widehat{B}(\omega)^{-1}\right]'Q\Lambda^{1/2}\mathbf{u}(\omega) \\
&= \beta(\omega)^2\mathbf{v}^*(\omega)\Lambda^{1/2}Q^*\widehat{B}(\omega)^{-1}\widehat{B}(\omega)'\widehat{B}(\omega)^{-1}Q\Lambda^{1/2}\mathbf{u}(\omega).
\end{aligned} \tag{3.9}$$

We summarize the Newton-type method in Algorithm 1. In the calculation of $\beta'(\omega_k)$ in (3.9), the schemes proposed in [17] that are based on the fast Fourier transform allow to compute products $Q^*\mathbf{p}$ and $Q\mathbf{q}$ efficiently.

In Algorithm 1 for the solution of (2.11), it is necessary to compute $\widehat{B}(\omega_k)^{-1}\mathbf{d}$ for a given vector $\mathbf{d}$. If $\widehat{B}(\omega_k)$ is as in (2.10), then $\widehat{B}(\omega_k)$ can be represented as

$$\widehat{B}(\omega_k) = B_1(\omega_k) + Y(\omega_k)X^*,$$

where

$$Y(\omega_k) = (A - \omega_k B_1(\omega_k))XD(\omega_k).$$

Using the Sherman-Morrison-Woodbury formula [14] we get

$$\widehat{B}(\omega_k)^{-1} = B_1(\omega_k)^{-1}\left\{I - Y(\omega_k)\left(I + X^*B_1(\omega_k)^{-1}Y(\omega_k)\right)^{-1}X^*B_1(\omega_k)^{-1}\right\}, \tag{3.11}$$

Algorithm 1 Newton-type method for computing eigenvalues of $A\mathbf{x} = \omega\widehat{B}(\omega)\mathbf{x}$.

Input: Coefficient matrices A (Hermitian), $\widehat{B}(\omega)$, an initial value ω_0 and a stopping tolerance tol.

Output: An eigenvalue/eigenvector pair $(\mu, \mathbf{x})$.

1: Set $k = 0$.
2: **repeat**
3: Compute the target eigenvalue/eigenvector pair $(\beta_k, \mathbf{u}_k)$ of

$$K(\omega_k)^{-1}\mathbf{u} \equiv \left(\Lambda^{1/2}Q^*\widehat{B}(\omega_k)^{-1}Q\Lambda^{1/2}\right)^{-1}\mathbf{u} = \beta\mathbf{u}, \tag{3.10}$$

by algorithm JD or SIRA (see Section 4 for details);
4: Use inverse iteration to compute the left eigenvector $\mathbf{v}_k$ of (3.10) associated with β_k;
5: Compute $\beta'(\omega_k)$ by

$$\beta'(\omega_k) = \beta_k^2\mathbf{v}_k^*\Lambda^{1/2}Q^*\widehat{B}(\omega_k)^{-1}\widehat{B}(\omega_k)'\widehat{B}(\omega_k)^{-1}Q\Lambda^{1/2}\mathbf{u}_k;$$

6: Compute ω_{k+1} by

$$\omega_{k+1} = \omega_k - \left(\beta'(\omega_k) + \omega_k^{-2}\right)^{-1}\left(\beta_k - \omega_k^{-1}\right);$$

7: Set $k = k + 1$;
8: **until** $|\omega_k - \omega_{k-1}| < tol$.
9: Set $\mu = \omega_k$;
10: Compute the eigenvector $\mathbf{x} = \widehat{B}(\omega_k)^{-1}Q\Lambda^{1/2}\mathbf{u}_k$.

which means that $\widehat{B}(\omega_k)^{-1}\mathbf{d}$ can be computed within a reasonable cost.

The desired eigenvalue/eigenvector pairs of $F(\omega)$, i.e. the ones with smallest real part, can be found by repeatedly applying Algorithm 1 to $A\mathbf{x} = \omega\widehat{B}(\omega)\mathbf{x}$ and using (2.8) in Theorem 2.3 to recover the eigenvector of $F(\omega)$. We summarize the computational process in Algorithm 2.

After proposing the Newton-type method it remains to describe the used eigenvalue methods used in Algorithm 1. This is done in the next section.

4. Eigenvalue Solvers. In this section, we recall different eigenvalue methods for the solution of the NLEVP (2.4) and the SEP (3.10) in Algorithm 1.

4.1. Nonlinear Arnoldi Method for the NLEVP (2.4)**.** The NLEVP (2.4) can be solved by the nonlinear Arnoldi method (NAr) [47] directly. For a given search subspace V, let $(\tilde{\omega}, \tilde{\mathbf{z}})$ be an eigenvalue/eigenvector pair of the projected problem $V^*(A - \omega^2 B(\omega))V\mathbf{z} = 0$ and let $\tilde{\mathbf{x}} = V\tilde{\mathbf{z}}$ be a corresponding Ritz vector, i.e. the eigenvector lifted to the large space. The new search direction in the NAr method is chosen as

$$\mathbf{v} = \left(A - \sigma^2 B(\sigma)\right)^{-1}\mathbf{r} \tag{4.1}$$

where $\mathbf{r} = (A - \tilde{\omega}^2 B(\tilde{\omega}))\tilde{\mathbf{x}}$ is the residual vector for $(\tilde{\omega}, \tilde{\mathbf{x}})$ and σ is a given shift value. After re-orthogonalizing $\mathbf{v}$ against V, the vector is appended to V and one repeats this process until $(\tilde{\omega}, \tilde{\mathbf{x}})$ has converged to the desired eigenvalue/eigenvector pair.

Algorithm 2 Deflated iterative method for solving $A\mathbf{x} = \omega^2 B(\omega)\mathbf{x}$.

Input: Coefficient matrices A (Hermitian) and $B_1(\omega) \equiv \omega B(\omega)$.
Output: The desired eigenvalue/eigenvector pair $(\mu_i, \mathbf{x}_i)$ for $i = 1, \ldots, \ell$.

1: Set $X = [\,]$ and $\widehat{B}(\omega) = B_1(\omega)$.
2: **for** $i = 1, \ldots, \ell$ **do**
3: Apply Algorithm 1 to compute the desired eigenvalue/eigenvector pair $(\mu_i, \mathbf{x}_i)$ of $A\mathbf{x} = \omega\widehat{B}(\omega)\mathbf{x}$;
4: **for** $j = 1, \ldots, i-1$ **do**
5: Compute $\mathbf{x}_i = \left(I - \frac{\mu_i}{\mu_i - \mu_j}\tilde{\mathbf{x}}_j\tilde{\mathbf{x}}_j^*\right)\mathbf{x}_i$;
6: **end for**
7: Set $\tilde{\mathbf{x}}_i = \mathbf{x}_i$; Orthogonalize $\tilde{\mathbf{x}}_i$ against X and normalize $\tilde{\mathbf{x}}_i$;
8: Expand $X = [X, \tilde{\mathbf{x}}_i]$;
9: Set $\widehat{B}(\omega) = B_1(\omega) + (A - \omega B_1(\omega))XD(\omega)X^*$, where $D(\omega) = \mathrm{diag}\left((\omega - \mu_1)^{-1}, \cdots, (\omega - \mu_i)^{-1}\right)$;
10: **end for**

The major computational cost in the NAr method arises in the solution of (4.1). This cost can be significantly reduced by using a technique suggested in [19]. Since $B(\sigma)$ in (2.4) is diagonal, we employ a preconditioner

$$M = A - \sigma^2\alpha_\sigma I$$

for the solution of (4.1), where α_σ is the average of the diagonal elements of $B(\sigma)$. Furthermore, we can rewrite (4.1) as

$$\left[I + \sigma^2 M^{-1}\left(\alpha_\sigma I - B(\sigma)\right)\right]\mathbf{v} = M^{-1}\mathbf{r},$$

which only requires to compute $\mathbf{d} + \sigma^2 M^{-1}\left(\alpha_\sigma I - B(\sigma)\right)\mathbf{d}$ in each iteration of an iterative solver for the computation of a vector $\mathbf{d}$. There is no need to compute a matrix-vector multiplication with A. The associated preconditioned linear system with coefficient matrix M can be efficiently solved by applying the spectral decompositions of the matrices C_ℓ, $\ell = 1, 2, 3$, see [19].

4.2. Jacobi-Davidson Method for the SEP (3.10)**.** The Jacobi-Davidson method (JD) [43] is an inexact eigenvalue solver for solving the SEP. In each iteration of JD, the correction equation

$$\left(I - \mathbf{u}\mathbf{u}^*\right)\left(K(\omega_k) - \theta I\right)\left(I - \mathbf{u}\mathbf{u}^*\right)\mathbf{t} = -\mathbf{r}, \quad \mathbf{t}\perp\mathbf{u} \tag{4.2}$$

is solved approximately by an iterative solver, where $K(\omega_k)$ is defined as in (3.10), $(\theta, \mathbf{u})$ is the Ritz pair of $K(\omega_k)$ and $\mathbf{r} = (K(\omega_k) - \theta I)\mathbf{u}$. Here $\mathbf{t}\perp\mathbf{u}$ means that $\mathbf{t}$ is orhogonal to $\mathbf{u}$. In each iteration of (4.2), we need to solve a linear system of the form

$$M_p\mathbf{z} = \mathbf{d}, \quad \mathbf{z}\perp\mathbf{u}, \tag{4.3}$$

where $\mathbf{d}$ is a given vector and

$$M_p \equiv \left(I - \mathbf{u}\mathbf{u}^*\right)M_J\left(I - \mathbf{u}\mathbf{u}^*\right),$$

with M_J being the preconditioner of $K(\omega_k) - \theta I$.

For the deflated NLEVP (2.7), i.e., $\widehat{B}(\omega) = \widetilde{B}(\omega)$ as in (3.11), one has that

$$K(\omega_k) - \theta I = \left(\Lambda^{1/2} Q^* B_1(\omega_k)^{-1} Q \Lambda^{1/2} - \theta I\right) - U(\omega_k)\Psi(\omega_k)^{-1} V(\omega_k)^*,$$

where

$$\begin{aligned} U(\omega_k) &= \Lambda^{1/2} Q^* B_1(\omega_k)^{-1}(A - \omega_k B_1(\omega_k)) X D(\omega_k), \\ V(\omega_k) &= \left[X^* B_1(\omega_k)^{-1} Q \Lambda^{1/2}\right]^*, \\ \Psi(\omega_k) &= I + X^* B_1(\omega_k)^{-1}(A - \omega_k B_1(\omega_k)) X D(\omega_k). \end{aligned}$$

Therefore, we take as preconditioner

$$\begin{aligned} M_J &= \left(\Lambda^{1/2} Q^* \alpha_{a,k} I Q \Lambda^{1/2} - \theta I\right) - U(\omega_k)\Psi(\omega_k)^{-1} V(\omega_k)^* \\ &= (\alpha_{a,k}\Lambda - \theta I) - U(\omega_k)\Psi(\omega_k)^{-1} V(\omega_k)^* \qquad (4.4) \\ &:= \Omega_k - U(\omega_k)\Psi(\omega_k)^{-1} V(\omega_k)^*, \qquad (4.5) \end{aligned}$$

where $\alpha_{a,k}$ is the average of the diagonal elements of $B_1(\omega_k)^{-1}$. By the Sherman-Morrison-Woodbury formula, we get

$$\begin{aligned} & M_J^{-1} \\ &= \Omega_k^{-1}\left\{I + U(\omega_k)\left(I - \Psi(\omega_k)^{-1} V(\omega_k)^* \Omega_k^{-1} U(\omega_k)\right)^{-1} \Psi(\omega_k)^{-1} V(\omega_k)^* \Omega_k^{-1}\right\} \\ &= \Omega_k^{-1}\left\{I + U(\omega_k)\left(\Psi(\omega_k) - V(\omega_k)^* \Omega_k^{-1} U(\omega_k)\right)^{-1} V(\omega_k)^* \Omega_k^{-1}\right\} \qquad (4.6) \end{aligned}$$

and the linear system (4.3) can be easily solved via

$$\mathbf{z} = M_J^{-1}\mathbf{d} + \eta M_J^{-1}\mathbf{u} \quad \text{with} \quad \eta = -\frac{\mathbf{u}^* M_J^{-1}\mathbf{d}}{\mathbf{u}^* M_J^{-1}\mathbf{u}}.$$

4.3. Shift-Invert Residual Arnoldi Method for the SEP (3.10)**.** The shift-and-invert residual Arnoldi method (SIRA), see [19, 30, 31] is another inexact eigenvalue solver. In each iteration of SIRA, the linear residual system

$$\left(K(\omega_k) - \sigma I\right)\mathbf{t} = \mathbf{r} \qquad (4.7)$$

is solved approximately by an iterative solver, where $\mathbf{r} = K(\omega_k)\mathbf{u} - \theta\mathbf{u}$ is the residual vector and σ is a fixed shift value. We express (4.7) in the form

$$\left(Q^* \widehat{B}(\omega_k)^{-1} Q - \sigma\Lambda^{-1}\right)\left(\Lambda^{1/2}\mathbf{t}\right) = \Lambda^{-1/2}\mathbf{r}. \qquad (4.8)$$

Using the construction of the preconditioner M_J in (4.5), we take

$$M_S = \left(\alpha_{a,k} I - \sigma\Lambda^{-1}\right) - \Lambda^{-1/2} U(\omega_k)\Psi(\omega_k)^{-1} V(\omega_k)^* \Lambda^{-1/2}$$

as a preconditioner and rewrite the linear system (4.8) as

$$A_M\left(\Lambda^{1/2}\mathbf{t}\right) = M_S^{-1}\Lambda^{-1/2}\mathbf{r}, \qquad (4.9)$$

where

$$A_M = M_S^{-1}\left[Q^* \widehat{B}(\omega_k)^{-1} Q - \sigma\Lambda^{-1}\right] = I + M_S^{-1} Q^* \left[B_1(\omega_k)^{-1} - \alpha_{a,k} I\right] Q.$$

Each of the three methods has its advantages and disadvantages, see [37]. We discuss a hybrid method using all three techniques in the next section.

5. Practical implementation. In this section we discuss some practical implementation issues and a new hybrid technique.

5.1. Initial value ω_0 and initial vector for Algorithm 1. In Algorithm 1, JD or SIRA is applied to solve (3.10). In the j-th iteration of JD or SIRA, usually more than one Ritz pair is computed. Suppose that $(\beta_{1,j}, \mathbf{u}_{1,j})$, ..., $(\beta_{m,j}, \mathbf{u}_{m,j})$ with $|\beta_{1,j}^{-1} - \sigma| \leq \cdots \leq |\beta_{m,j}^{-1} - \sigma|$ have been computed. When $\{\beta_{1,j}\}$ is converging to the desired eigenvalue, then the subspace $\text{span}\{\mathbf{u}_{2,j}, \ldots, \mathbf{u}_{m,j}\}$ is used as initial subspace in JD or SIRA for the solution of the new nullspace-free eigenvalue problem (3.10).

In Algorithm 2, Algorithm 1 is applied to compute the i-th eigenvalue/eigenvector pair of (2.4). The m Ritz pairs $(\beta_{1,j}, \mathbf{u}_{1,j})$, ..., $(\beta_{m,j}, \mathbf{u}_{m,j})$ of (3.10) are computed in Algorithm 1. Therefore, when Algorithm 1 converges, we can use $\omega_0 \equiv \beta_{2,j}^{-1}$ and the subspace generated from $\{\mathbf{u}_{2,j}, \ldots, \mathbf{u}_{m,j}\}$ as initial value and initial subspace, respectively, in computing the $(i+1)$-th eigenvalue/eigenvector pair of (2.4).

Let $(\mu_k, \mathbf{x}_k)$ for $k = 1, \ldots, i$ be the converged eigenvalue/eigenvectors pairs of (2.4) and let $\{\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_i\}$ be an orthonormal basis of $\text{span}\{\mathbf{x}_1, \ldots, \mathbf{x}_i\}$. The initial subspace generated from $\{\mathbf{u}_{2,j}, \ldots, \mathbf{u}_{m,j}\}$ is then constructed as follows. For convenience, we use $\mathbf{u}_{2,j}$ as an example. From Theorems 2.3 and 3.3, we set

$$\mathbf{x}_0 := \prod_{k=1}^{i-1} \left(I - \frac{\mu_i}{\mu_i - \mu_k} \widetilde{\mathbf{x}}_k \widetilde{\mathbf{x}}_k^* \right) \widetilde{B}(\mu_i)^{-1} Q \Lambda^{1/2} \mathbf{u}_{2,j}.$$

Then, the pair $(\omega_0, \mathbf{x}_0)$ can be used as an approximate eigenvalue/eigenvector pair of NLEVP (2.4) and the associated residual vector $\mathbf{r}_2$ is equal to

$$\begin{aligned} \mathbf{r}_2 &= A\mathbf{x}_0 - \omega_0^2 B(\omega_0)\mathbf{x}_0 = F(\omega_0)\mathbf{x}_0 = \widetilde{F}(\omega_0) \prod_{j=1}^{i} \left(I - \frac{\omega_0}{\omega_0 - \mu_j} \widetilde{\mathbf{x}}_j \widetilde{\mathbf{x}}_j^* \right)^{-1} \mathbf{x}_0 \\ &:= \widetilde{F}(\omega_0)\tilde{\mathbf{x}}_0, \end{aligned} \tag{5.1}$$

where

$$\begin{aligned} \tilde{\mathbf{x}}_0 &= \prod_{j=1}^{i} \left(I - \frac{\omega_0}{\omega_0 - \mu_j} \widetilde{\mathbf{x}}_j \widetilde{\mathbf{x}}_j^* \right)^{-1} \mathbf{x}_0 = \prod_{j=1}^{i} \left(I - \frac{\omega_0}{\mu_j} \widetilde{\mathbf{x}}_j \widetilde{\mathbf{x}}_j^* \right) \mathbf{x}_0 \\ &= \mathbf{x}_0 - \sum_{j=1}^{i} \frac{\omega_0}{\mu_j} \left(\widetilde{\mathbf{x}}_j^* \mathbf{x}_0 \right) \widetilde{\mathbf{x}}_j. \end{aligned}$$

This shows that $(\omega_0, \widetilde{\mathbf{x}}_0)$ can be used as an approximate eigenvalue/eigenvector pair of the deflated NLEVP. Therefore, using Theorem 3.3, we may take

$$\mathbf{u}_0 = \Lambda^{-1/2} Q^* \widetilde{B}(\omega_0) \widetilde{\mathbf{x}}_0$$

as initial vector for the solution of (3.10) in the first iteration of Algorithm 1.

5.2. Stopping tolerance τ_k in JD or SIRA. In Algorithm 1, the sequence $\{\omega_k\}$ is constructed by a sequence of eigenvalues of the SEPs (3.10). This allows us to adaptively control the accuracy of the eigenvalue/eigenvector pairs to reduce the computational costs. Here, we propose a heuristic strategy for the construction of the

stopping tolerance $\tau_k,\ k > 0$, in (3.10) as

$$\tau_k = \max\left\{\max\left\{5\cdot 10^{-12}, \frac{10^4\text{eps}}{2\sqrt{\delta_x^{-2}+\delta_y^{-2}+\delta_z^{-2}}}\right\}, \min\left\{5\cdot 10^{-4}, 0.1\cdot|\omega_k-\omega_{k-1}|^2\right\}\right\}. \tag{5.2}$$

The stopping tolerance τ_0 for ω_0 is set to

$$\tau_0 = \begin{cases} 10^{-3}, & \text{for the first eigenvalue/eigenvector pair of (2.4);} \\ \min(10^{-3}, 0.1\cdot\|\mathbf{r}_2\|_2), & \text{otherwise,} \end{cases} \tag{5.3}$$

where $\mathbf{r}_2$ is the residual vector in (5.1).

Algorithm 3 $[\lambda, \mathbf{x}, \mathbf{r}_k, \mathbb{V}_k]$ = NFEP(K,$\mathbb{V}_1$, σ, τ, maxit)

Input: Hermitian matrix K, an initial matrix $\mathbb{V}_1$, shift σ, tolerance τ and maximal number of iterations maxit.

Output: The desired eigenvalue/eigenvector pair $(\lambda, \mathbf{x})$.

Set $k = 1$, $\mathbf{r}_0 = \mathbf{e}_1$ and 'solver' = 'JD'.
Compute $\mathbb{W}_k = K\mathbb{V}_k$ and $\mathbb{M}_k = \mathbb{V}_k^*\mathbb{W}_k$.
while ($k \le$ maxit and $\|\mathbf{r}_{k-1}\|_2 \ge \tau$) **do**
  Compute the eigenvalue/eigenvector pairs $(\theta_i, \mathbf{s}_i)$ of $\mathbb{M}_k\mathbf{s} = \theta\mathbf{s}$ with $\|\mathbf{s}_i\|_2 = 1$ and $\sigma < \theta_1 \le \theta_2 \le \cdots$.
  Compute $\mathbf{u}_k = \mathbb{V}_k\mathbf{s}_1$ and $\mathbf{r}_k = (K - \theta_1 I)\mathbf{u}_k$.
  if ($\|\mathbf{r}_k\|_2 \ge \epsilon$) **then**
    if ('solver' = 'SIRA') **then**
      Compute (approximate) solution $\mathbf{t}_k$ for

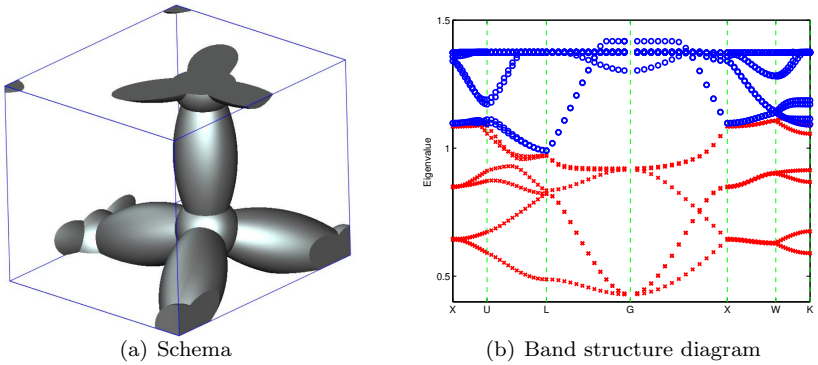
$$(K - \sigma I)\mathbf{t}_k = \mathbf{r}_k.$$

    else if ('solver' = 'JD') **then**
      Compute (approximate) solution $\mathbf{t}_k \perp \mathbf{u}_k$ for

$$(I - \mathbf{u}_k\mathbf{u}_k^*)(K - \theta_1 I)(I - \mathbf{u}_k\mathbf{u}_k^*)\mathbf{t}_k = -\mathbf{r}_k.$$

      if (the approximate solution $\mathbf{t}_k$ is difficult to compute) **then**
        Set 'solver' = 'SIRA';
      end if
    end if
    Orthogonalize $\mathbf{t}_k$ against $\mathbb{V}_k$; set $\mathbf{v}_{k+1} = \mathbf{t}_k/\|\mathbf{t}_k\|$.
    Compute $\mathbf{w}_{k+1} = K\mathbf{v}_{k+1}$, $\mathbb{M}_{k+1} = \begin{bmatrix} \mathbb{M}_k & \mathbb{V}_k^*\mathbf{w}_{k+1} \\ \mathbf{v}_{k+1}^*\mathbb{W}_k & \mathbf{v}_{k+1}^*\mathbf{w}_{k+1} \end{bmatrix}$.
    Expand $\mathbb{V}_{k+1} = [\mathbb{V}_k, \mathbf{v}_{k+1}]$ and $\mathbb{W}_{k+1} = [\mathbb{W}_k, \mathbf{w}_{k+1}]$. Set $k := k + 1$.
  end if
end while
Set $\lambda = \theta_1$, $\mathbf{x} = \mathbf{u}_k$ and $\mathbb{V}_1 = \mathbb{V}_k[\mathbf{s}_1, \cdots, \mathbf{s}_p]$.

5.3. A Hybrid eigensolver with non-equivalence deflation. We use the heuristic strategies in [16] to determine the maximal iteration number for solving the



(a) Schema (b) Band structure diagram

FIG. 5.1. *(a) A schematic view of a dispersive metallic photonic crystal structure with a FCC lattice within a single primitive cell. (b) The computed band structure diagram of the Drude model with matrix dimension* $3 \cdot 48^3 = 331,776$. *There are* 16 *eigenvalues in the diagram. The six smallest real part nonzero eigenvalues* $\mu_1, \ldots, \mu_6$ *are denoted by (red)* $\times$.

correction equation (4.2) in JD. As the results in [19] suggest, we set the stopping tolerance 5×10^{-4} for (4.7) in SIRA. The numerical results presented in Subsection 6.1 show that JD outperforms SIRA for most benchmark problems. However, sometimes the iterative solver for solving the correction equation cannot get a better approximate solution. Therefore, a hybrid of JD and SIRA in Algorithm 3 is used.

The Newton-type method in Algorithm 1 needs a good initial value ω_0 of ω to guarantee convergence. If the NLEVP (2.4) has clustered eigenvalues, then we need to provide a method to compute ω_0. For this we use NAr and the stopping tolerance for NAr is taken as

$$\tau_a = \max\left\{\min\left\{\|\mathbf{r}_h\|, 10^{-3}\right\}, 5 \times 10^{-8}\right\}, \tag{5.4}$$

where $\mathbf{r}_h$ is the residual vector of the approximate eigenvalue/eigenvector produced by Algorithm 3. The stopping tolerance for solving the linear system (4.1) in NAr is set to be $\max(10^{-12}, \tau_a^2)$. A heuristic hybrid method with JD, SIRA and NAr for computing one eigenvalue/eigenvector pair of (2.11) is summarized in Algorithm 4. We replace Algorithm 1 with this heuristic hybrid method to solve $A\mathbf{x} = \omega\widehat{B}(\omega)\mathbf{x}$ in Algorithm 2.

6. Numerical Results. To study the numerical performance of the described method for the solution of the nonlinear eigenvalue problem (2.4) arising in the 3D dispersive metallic PCs, we consider the setup described in [6, 17, 18]. The lattice in Figure 5.1(a) consists of spheres with a connecting spheroid. The radius r of the spheres is $r = 0.08a$ and the connecting spheroid has a minor axis length $s = 0.06a$ with $a = 2\pi$. The perimeter of the irreducible Brillouin zone for the lattice is formed by the corners $X = \frac{2\pi}{a}\Omega[0,1,0]^\top$, $U = \frac{2\pi}{a}\Omega\left[\frac{1}{4},1,\frac{1}{4}\right]^\top$, $L = \frac{2\pi}{a}\Omega\left[\frac{1}{2},\frac{1}{2},\frac{1}{2}\right]^\top$, $G = [0,0,0]^\top$, $W = \frac{2\pi}{a}\Omega\left[\frac{1}{2},1,0\right]^\top$, and $K = \frac{2\pi}{a}\Omega\left[\frac{3}{4},\frac{3}{4},0\right]^\top$, where

$$\Omega = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 & 0 \\ -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{2}{\sqrt{3}} \\ \frac{2}{\sqrt{6}} & -\frac{2}{\sqrt{6}} & \frac{2}{\sqrt{6}} \end{bmatrix}.$$

Algorithm 4 Hybrid eigensolver for computing smallest real part nonzero eigenvalues of $A\mathbf{x} = \omega\widehat{B}(\omega)\mathbf{x}$.

Input: Coefficient matrices A (Hermitian), $\widehat{B}(\omega)$, initial value ω_0, initial vector $\mathbb{V}_1$, maximal iteration number m, the stopping tolerance tol, converged eigenvalues $\{\mu_1, \ldots, \mu_\ell\}$ and orthonormal basis $\widetilde{\mathbf{x}}_1, \ldots, \widetilde{\mathbf{x}}_\ell$ of the associated eigenspace.

Output: The target eigenvalue/eigenvector pairs $(\mu, \mathbf{x})$.

1: Set $k = 0$ and τ_0 as in (5.3).
2: **repeat**
3: **while** ($\|\mathbf{r}_h\| \geq \tau_k$) **do**
4: Use

$$[\beta_k^{-1}, \mathbf{u}_k, \mathbf{r}_h, \mathbb{V}_1] = \text{NFEP}(K(\omega_k), \mathbb{V}_1, \sigma, \tau_k, m)$$

in Algorithm 3 to compute the target eigenvalue/eigenvector pair $(\beta_k, \mathbf{u}_k)$ of (3.10);
5: **if** $(\|\mathbf{r}_h\| \geq \epsilon_k)$ **then**
6: Compute

$$\mathbf{x}_0 = \prod_{j=1}^{\ell} \left(I - \frac{\omega_k}{\omega_k - \mu_j} \widetilde{\mathbf{x}}_j \widetilde{\mathbf{x}}_j^* \right) \widehat{B}(\omega_k)^{-1} Q \Lambda^{1/2} \mathbf{u}_k.$$

7: Use NAr with initial vector $\mathbf{x}_0$ and stopping tolerance τ_a in (5.4) to compute the approximate eigenvalue/eigenvector pair $(\omega_a, \mathbf{x}_a)$ of the NLEVP (2.4), where ω_a is the closest eigenvalue to σ.
8: Set $\omega_k = \omega_a$ and compute the initial vector $\mathbb{V}_1$ for (3.10) as

$$\mathbb{V}_1 = \Lambda^{-1/2} Q^* \widehat{B}(\omega_k) \left\{ \mathbf{x}_a - \sum_{j=1}^{\ell} \frac{\omega_k}{\mu_j} \left(\widetilde{\mathbf{x}}_j^* \mathbf{x}_a \right) \widetilde{\mathbf{x}}_j \right\}.$$

9: **end if**
10: **end while**
11: Use one step of inverse iteration to compute the left eigenvector $\mathbf{v}_k$ of (3.10) associated with β_k;
12: Compute $\beta'(\omega_k)$ via

$$\beta'(\omega_k) = \beta_k^2 \mathbf{v}_k^* \Lambda^{1/2} Q^* \widehat{B}(\omega_k)^{-1} \widehat{B}(\omega_k)' \widehat{B}(\omega_k)^{-1} Q \Lambda^{1/2} \mathbf{u}_k;$$

13: Compute ω_{k+1} by

$$\omega_{k+1} = \omega_k - \left(\beta'(\omega_k) + \omega_k^{-2} \right)^{-1} \left(\beta_k - \omega_k^{-1} \right);$$

14: Set $k = k + 1$ and ϵ_k as in (5.2);
15: **until** $|\omega_k - \omega_{k-1}| < tol$.
16: Set $\mu = \omega_k$;
17: Compute the eigenvector $\mathbf{x} = \widehat{B}(\omega_k)^{-1} Q \Lambda^{1/2} \mathbf{u}_k$.

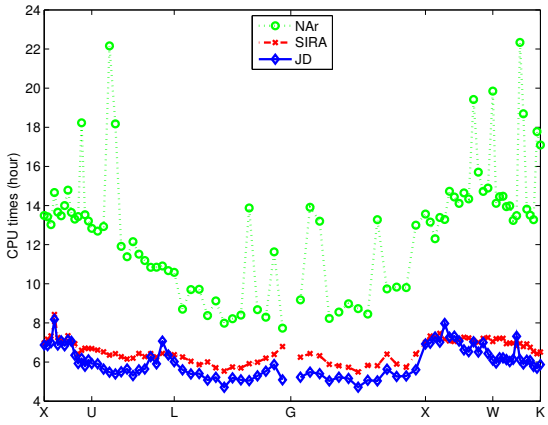


FIG. 6.1. *CPU times for computing the six smallest real part nonzero eigenvalues (and associated eigenvectors) denoted by (red) $\times$ in Figure 5.1(b). The matrix dimension is* $2,654,208$.

The permittivity ε_n of the non-dispersive material is set to be 1. Parameters in the Drude and Drude-Lorentz models are $\omega_p = \frac{10\pi}{a}$, $\Gamma_p = \frac{2\pi}{14500}$, $\Omega_1 = \frac{2\pi}{470}$, $\Omega_2 = \frac{2\pi}{325}$, $\Gamma_1 = \frac{2\pi}{1900}$, $\Gamma_2 = \frac{2\pi}{1060}$, $\varepsilon_\infty = 1.54$, $A_1 = 1.27$, $A_2 = 1.1$, $\phi_1 = -\frac{\pi}{4}$ and $\phi_2 = -\frac{\pi}{4}$ [34]. The associated band structure for the Drude model is shown in Figure 5.1(b). The band structure for the Drude-Lorentz model is similar to Figure 5.1(b).

All computations in this section are carried out in MATLAB 2013b, and some implementation details are addressed as follows. The MATLAB functions `bicg` and `bicgstab` are used to solve the linear systems in Algorithm 3 and NAr, respectively. On the other hand, the MATLAB functions `fft` and `ifft` are applied to compute the products $Q^*\mathbf{p}$ and $Q\mathbf{q}$, respectively.

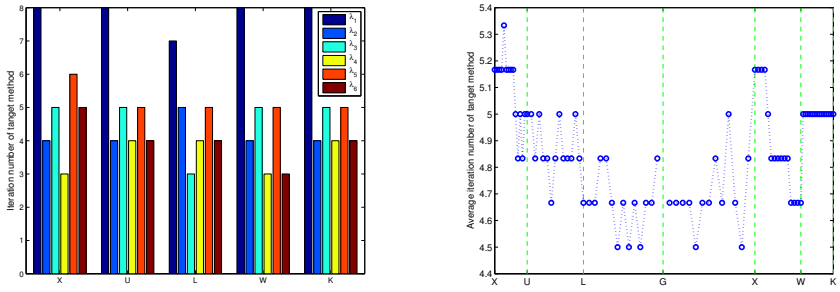
For the hardware configuration, we use a HP workstation that is equipped with two Intel Quad-Core Xeon E5-2643 3.33GHz CPUs, 96 GB of main memory, and the RedHat Linux operating system.

6.1. Comparison between Newton-type method and nonlinear Arnoldi method. We demonstrate the efficiency of the new Newton-type method (Algorithm 1 and Algorithm 2) and NAr in computing the six smallest real part eigenvalues $\mu_1, \ldots, \mu_6$ of the Drude model (1.2). The real parts of these eigenvalues are denoted by red $\times$ in Figure 5.1(b). As shown in the figure, $\mu_1, \ldots, \mu_6$ are well separated so that they can be computed by Algorithm 2 without using NAr to get initial data. The CPU times for computing these six eigenvalue/eigenvector pairs by

- JD: Algorithm 2 + Algorithm 1 with solving (3.10) by JD
- SIRA: Algorithm 2 + Algorithm 1 with solving (3.10) by SIRA
- NAr: nonlinear Arnoldi method for solving (2.4)

are depicted in Figure 6.1, which shows that JD and SIRA obviously outperform NAr. Furthermore, in the Newton-type method, JD outperforms SIRA for most benchmark problems.

6.2. Convergence of the Newton-type method. In this subsection, we illustrate the convergence of the Newton-type method in Algorithm 1 for computing



(a) Number of iterations k of Newton-type method for computing each eigenvalue at wave vectors X, U, L, W and K.

(b) Average number of iterations of Newton-type method for computing six eigenvalues.

FIG. 6.2. *Number of iterations of Newton-type method with JD eigensolver to compute the six smallest real part nonzero eigenvalues denoted by (red) $\times$ in Figure 5.1(b). The matrix dimension is* $2,654,208$.

	Drude model	Drude-Lorentz model
μ_7	$1.352760915 - 2.15717754 \times 10^{-4}\imath$	$1.326911260 - 2.11350594 \times 10^{-3}\imath$
μ_8	$1.352771023 - 2.15790978 \times 10^{-4}\imath$	$1.326915939 - 2.11375183 \times 10^{-3}\imath$
μ_9	$1.352771589 - 2.15790991 \times 10^{-4}\imath$	$1.326916471 - 2.11375357 \times 10^{-3}\imath$
μ_{10}	$1.352774278 - 2.15790186 \times 10^{-4}\imath$	$1.326919090 - 2.11375510 \times 10^{-3}\imath$
μ_{11}	$1.354710739 - 2.15785421 \times 10^{-4}\imath$	$1.328746727 - 2.11897302 \times 10^{-3}\imath$
μ_{12}	$1.354711852 - 2.15790561 \times 10^{-4}\imath$	$1.328747433 - 2.11899196 \times 10^{-3}\imath$
μ_{13}	$1.354711871 - 2.15790691 \times 10^{-4}\imath$	$1.328747439 - 2.11899260 \times 10^{-3}\imath$
μ_{14}	$1.354711899 - 2.15790684 \times 10^{-4}\imath$	$1.328747467 - 2.11899263 \times 10^{-3}\imath$

TABLE 6.1

Eigenvalues μ_7, ..., μ_{14} of (2.4) with wave vector $\frac{3}{7}X$ and matrix dimension $2,654,208$.

μ_1, ..., μ_6 for the problem discussed in the previous subsection. Using JD to solve the eigenvalue problem (3.10), the number of iterations k for computing each μ_i at wave vectors X, U, L, W and K in the FCC lattice is depicted in Figure 6.2(a). Since random initial data are used to compute the first eigenvalue μ_1, typically about $k = 8$ iterations are needed for most benchmark wave vectors. For other eigenvalues, using the initial data proposed in Subsection 5.1, 3 to 6 iterations are needed. Figure 6.2(b) shows the average number of iterations for computing μ_1, ..., μ_6 with various wave vectors $\mathbf{k}$. The numerical experience indicates that the average ranges from 4.5 to 5.2 for all the benchmark problems with the same matrix dimension $3 \times 96^3 = 2,654,208$. This convergence behaviour coincides with that of Newton's method for solving general nonlinear equations.

6.3. Clustered eigenvalues. From the band structure diagram in Figure 5.1(b) we see that the eigenvalues are clustered near 1.35 (1.32) for the Drude (Drude-Lorentz) model. Table 6.1 shows the values of the clustered eigenvalues μ_7, ..., μ_{14} of (2.4) (with the wave vector $\frac{3}{7}X$) for the Drude and Drude-Lorentz model. These clustered eigenvalues not only significantly increase the number of iterations for the NAr as shown in Figure 6.3, but also lead to a challenge for the Newton-type method: how to detect the clustered eigenvalues. In Figure 6.4, we depict the number

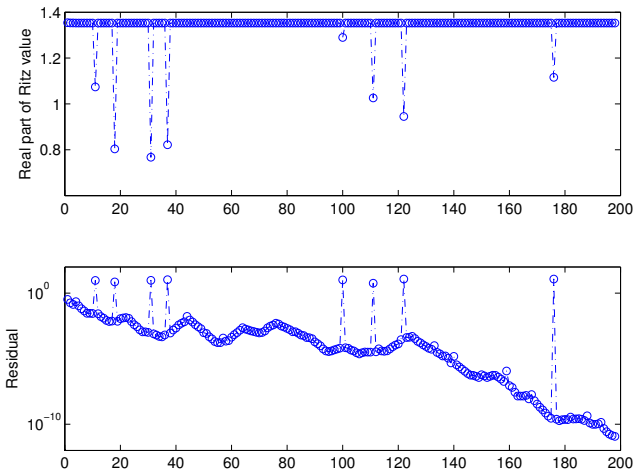
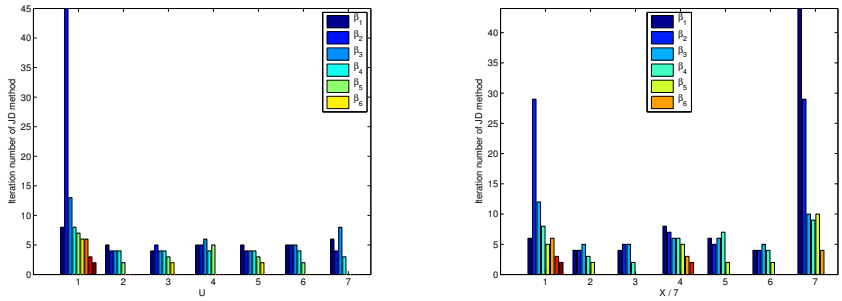


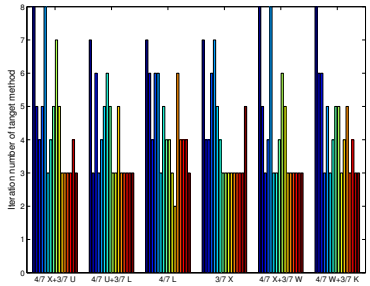
FIG. 6.3. *Convergence behaviour of the Ritz values (vs number of iterations), produced by NAr, for computing* μ_7 *(at the wave vector* $\frac{3}{7}X$*) for the Drude model. The matrix dimension is* $2,654,208$.



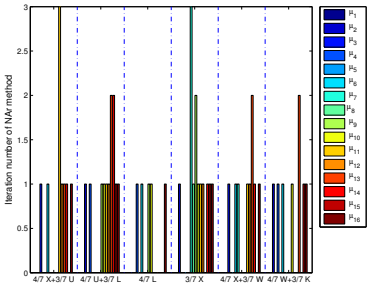
(a) $\mathbf{k} = U$ (b) $\mathbf{k} = \frac{1}{7}X$

FIG. 6.4. *Number of iterations of JD for computing eigenvalue* β_k *of* (3.10) *in Algorithm 1 at the wave vector* $\mathbf{k}$. *The matrix dimension is* $2,654,208$.

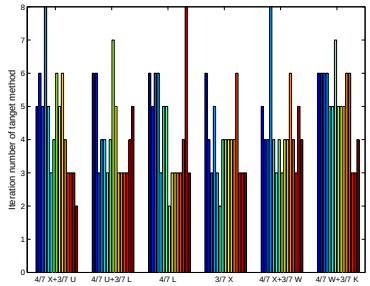
of iterations for JD in computing eigenvalue/eigenvector pairs $(\beta_k, \mathbf{u}_k)$ of (3.10) in computing $\mu_1, \ldots, \mu_7$ for the Drude model. For one specific wave vector, where $\mu_1, \ldots, \mu_7$ are well separated, Figure 6.4(a) shows that in each case all the iteration numbers of the JD method are less than 10. However, μ_7 is close to the clustered eigenvalues for wave vector $\frac{1}{7}X$. The results in Figure 6.4(b) indicate that the number of iterations for computing β_1 and β_2 are obviously larger than that for other eigenvalues. This means that the number of iterations of the JD method are an important indicator for detecting clustered eigenvalues. In this example, we set the maximal number of iterations to $m = 35$ in Algorithm 4 and if JD is not converged in this many steps, then we regard the eigenvalues to be clustered and use NAr to provide good initial data.



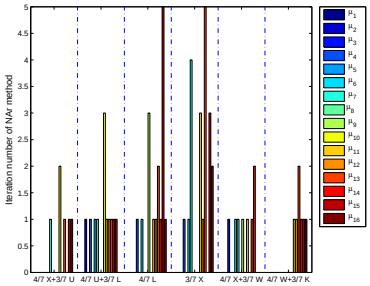
(a) Number of iterations of Newton-type method.



(b) Number of iterations of NAr.



(c) Number of iterations of Newton-type method.



(d) Number of iterations of NAr.

FIG. 6.5. *Number of iterations of Newton-type method for computing eigenvalues μ_1, ..., μ_{16}, and number of steps of the NAr method for computing the initial value and vector for each eigenvalue/eigenvector pair at wave vectors $\frac{4}{7}X+\frac{3}{7}U$, $\frac{4}{7}U+\frac{3}{7}L$, $\frac{4}{7}L$, $\frac{3}{7}X$, $\frac{4}{7}X+\frac{3}{7}W$, and $\frac{4}{7}W+\frac{3}{7}K$. The matrix dimension is $3\times 96^3 = 2,654,208$. (a) and (b) are the results for the Drude model, (c) and (d) are the results for the Drude-Lorentz model.*

6.4. Efficiency of Algorithm 2 with Algorithm 4. Combining Algorithm 2 with the hybrid eigensolver in Algorithm 4 with $m = 35$, the band structure diagram can be produced as in Figure 5.1(b). We consider six different wave vectors $\frac{4}{7}X+\frac{3}{7}U$, $\frac{4}{7}U+\frac{3}{7}L$, $\frac{4}{7}L$, $\frac{3}{7}X$, $\frac{4}{7}X+\frac{3}{7}W$, and $\frac{4}{7}W+\frac{3}{7}K$, and take $n_1 = n_2 = n_3 = 96$, i.e., the matrix dimension is $2,654,208$. For each wave vector, 16 eigenvalues μ_1, ..., μ_{16} are considered. For each eigenvalue μ_i, we depict the associated number of iterations k_i of the Newton-type method and the number p_i of the NAr to get initial data. The values of (k_i, p_i) for the Drude and the Drude-Lorentz models are shown in Figures (6.5(a), 6.5(b)) and (6.5(c), 6.5(d)), respectively. In our proposed method, we not only use NAr to provide an initial value and vector for the eigenvalue/eigenvector pair, but also use the strategies in Subsection 5.1 to confirm convergence. Therefore, even if the eigenvalues are strongly clustered as shown in Table 6.1, we can compute the desired eigenvalue/eigenvector pair within a reasonable (k_i, p_i) as shown in Figure 6.5.

Except for our proposed method, NAr can also be applied to compute the desired eigenvalue/eigenvector pairs directly. For this reason, we also compared the CPU times for computing the desired 16 eigenvalue/eigenvectors by our proposed method (T_H) and NAr (T_A). The numerical results for the Drude model show that the ratio T_A/T_H is 2.425, 2.534, 2.002, 2.019, 2.712, and 4.071, for wave vectors $\frac{4}{7}X + \frac{3}{7}U$,

$\frac{4}{7}U + \frac{3}{7}L$, $\frac{4}{7}L$, $\frac{3}{7}X$, $\frac{4}{7}X + \frac{3}{7}W$, and $\frac{4}{7}W + \frac{3}{7}K$, respectively. All the results which demonstrates that the proposed method is robust and outperforms NAr.

7. Conclusions. Solving the nonlinear eigenvalue problem (NLEVP) arising from Yee's discretization of a three-dimensional dispersive metallic photonic crystal is a computational challenge. We have proposed a Newton-type method to compute one desired eigenvalue/eigenvector pair of the NLEVP at a time. Once the desired eigenvalue is converged, it is then transformed to infinity by the proposed non-equivalence deflation scheme, while all other eigenvalues remain unchanged. The next successive eigenvalue thus becomes the smallest nonzero real part eigenvalue of the transformed NLEVP. which is then again solved by the Newton-type method. Furthermore, some heuristic strategies for the determination of initial data and stopping tolerances of the iterative eigenvalue methods are introduced to accelerate the convergence. In order to compute the clustered eigenvalues of the NLEVP, we propose a hybrid method which uses the Jacobi-Davidson or shift-invert residual Arnoldi method to solve the standard eigenvalue problems in the Newton-type method and the nonlinear Arnoldi method to compute the initial vectors. The numerical results demonstrate that our proposed method is robust and outperforms the nonlinear Arnoldi methods for the direct solution of the NLEVP.

Acknowledgments. The first author's work was partially supported by the Ministry of Science and Technology (MST) in Taiwan. The second author's work was partially supported by MST in Taiwan and the DAAD "Deutscher Akademischer Austausch Dienst" in Germany. The research of the third author was carried out in the framework of MATHEON project *D-OT3, Adaptive finite element methods for nonlinear parameter-dependent eigenvalue problems in photonic crystals* supported by Einstein Foundation Berlin.

REFERENCES

[1] J. Asakura, T. Sakurai, H. Tadano, T. Ikegami, and K. Kimura. A numerical method for nonlinear eigenvalue problems using contour integrals. *JSIAM Lett.*, 1:52–55, 2009.

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

[3] C. G. Baker, U. L. Hetmaniuk, R. B. Lehoucq, and H. K. Thornquis. Anasazi software for the numerical solution of large-scale eigenvalue problems. *ACM Trans. Math. Software*, 36:Article 13, 2009.

[4] W.-J. Beyn. An integral method for solving nonlinear eigenvalue problems. *Linear Alg. Appl.*, 436:3839–3863, 2012.

[5] W.-J. Beyn, C. Effenberger, and D. Kressner. Continuation of eigenvalues and invariant pairs for parameterized nonlinear eigenvalue problems. *Numer. Math.*, 119:489–516, 2011.

[6] R. Chern, C. Chang, C.-C. Chang, and R. Hwang. Numerical study of three-dimensional photonic crystals with large band gaps. *J. Phys. Soc. Japan*, 73:727–737, 2004.

[7] C. Effenberger. *Robust solution methods for nonlinear eigenvalue problems*. PhD thesis, Ecole polytechnique federale de Lausanne, Lausanne, CH, 2013.

[8] C. Engström, C. Hafner, and K. Schmidt. Computations of lossy Bloch waves in two-dimensional photonic crystals. *J. Comput. Theor. Nanosci.*, 6:1–9, 2009.

[9] C. Engström and M. Wang. Complex dispersion relation calculations with the symmetric interior penalty method. *Int. J. Numer. Meth. Engng.*, 84:849–863, 2010.

[10] P. G. Etchegoin, E. C. Le Ru, and M. Meyer. An analytic model for the optical properties of gold. *J. Chem. Phys.*, 125:164705, 2006.

[11] P. G. Etchegoin, E. C. Le Ru, and M. Meyer. Erratum: "An analytic model for the optical properties of gold" [J. Chem. Phys. 125, 164705 (2006)]. *J. Chem. Phys.*, 127:189901, 2007.

[12] S. Fan, P. R. Villeneuve, and J. D. Joannopoulos. Large omnidirectional band gaps in metal-lodielectric photonic crystals. *Phys. Rev. B*, 54:11245–11251, 1996.

[13] I. Gohberg, P. Lancaster, and L. Rodman. *Matrix Polynomials*. Academic Press, New York, 1982.
[14] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins Univ. Pr., 1996.
[15] J.-S. Guo, W.-W. Lin, and C.-S. Wang. Numerical solutions for large sparse quadratic eigenvalue problems. *Linear Alg. Appl.*, 225:57–89, 1995.
[16] T.-M. Huang, W.-J. Chang, Y.-L. Huang, W.-W. Lin, W.-C. Wang, and W. Wang. Preconditioning bandgap eigenvalue problems in three dimensional photonic crystals simulations. *J. Comput. Phys.*, 229:8684–8703, 2010.
[17] T.-M. Huang, H.-E. Hsieh, W.-W. Lin, and W. Wang. Eigendecomposition of the discrete double-curl operator with application to fast eigensolver for three dimensional photonic crystals. *SIAM J. Matrix Anal. Appl.*, 34:369–391, 2013.
[18] T.-M. Huang, H.-E. Hsieh, W.-W. Lin, and W. Wang. Matrix representation of the double-curl operator for simulating three dimensional photonic crystals. *Math. Comput. Model.*, 58:379–392, 2013.
[19] T.-M. Huang, H.-E. Hsieh, W.-W. Lin, and W. Wang. Eigenvalue solvers for three dimensional photonic crystals with face-centered cubic lattice. *J. Comput. Appl. Math.*, 272:350–361, 2014.
[20] T.-M. Huang and W.-W. Lin. A novel deflation technique for solving quadratice eigenvalue problems. *Bulletin of the Institute of Mathematics, Academia Sinica (New Series)*, 9:57–84, 2014.
[21] F. Hwang, Z. Wei, T.-M. Huang, and W. Wang. A parallel additive Schwarz preconditioned Jacobi-Davidson algorithm for polynomial eigenvalue problems in quantum dot simulation. *J. Comput. Phys.*, 229:2932–2947, 2010.
[22] T.-M. Hwang, W.-W. Lin, J.-L. Liu, and W. Wang. Jacobi–Davidson methods for cubic eigenvalue problems. *Numer. Linear Algebr. Appl.*, 12:605–624, 2005.
[23] T.-M. Hwang, W.-W. Lin, and V. Mehrmann. Numerical solution of quadratic eigenvalue problems with structure-preserving methods. *SIAM J. Sci. Comput.*, 24:1283–1302, 2003.
[24] T.-M. Hwang, W.-W. Lin, W.-C. Wang, and W. Wang. Numerical simulation of three dimensional pyramid quantum dot. *J. Comput. Phys.*, 196:208–232, 2004.
[25] E. Jarlebring, W. Michiels, and K. Meerbergen. A linear eigenvalue algorithm for the nonlinear eigenvalue problem. *Numer. Math.*, 122:169–195, 2012.
[26] E. Jarlebring and H. Voss. Rational Krylov for nonlinear eigenvalues, an iterative projectioon method. *Appl. Math.*, 50:543–554, 2005.
[27] J. D. Joannopoulos, S. G. Johnson, J. N. Winn, and R. D. Meade. *Photonic Crystals: Molding the Flow of Light*. Princeton University Press, 2008.
[28] C. Kittel. *Introduction to solid state physics*. Wiley, New York, 2005.
[29] D. Kressner. A block Newton method for nonlinear eigenvalue problems. *Numer. Math.*, 114:355–372, 2009.
[30] C. Lee. *Residual Arnoldi method: theory, package and experiments*. PhD thesis, TR-4515, Department of Computer Science, University of Maryland at College Park, 2007.
[31] C. Lee and G. W. Stewart. Analysis of the residual Arnoldi method. Technical report, TR-4890, Department of Computer Science, University of Maryland at College Park, 2007.
[32] R. Lehoucq, D. Sorensen, and C. Yang. *ARPACK USERS GUIDE: Solution of large scale eigenvalue problems with implicitely restarted Arnoldi methods*. SIAM, Philadelphia, 1998.
[33] B.-S. Liao, Z. Bai, L.-Q. Lee, and K. Ko. Nonlinear Rayleigh-Ritz iterative method for solving large scale nonlinear eigenvalue problems. *Taiwan. J. Math.*, 14:869–883, 2010.
[34] M. Luo and Q. H. Liu. Three-dimensional dispersive metallic photonic crystals with a bandgap and a high cutoff frequency. *J. Opt. Soc. Am. A Opt. Image Sci. Vis.*, 27:1878–1884, 2010.
[35] D. Mackey, N. Mackey, C. Mehl, and V. Mehrmann. Vector spaces of linearizations for matrix polynomials. *SIAM J. Matrix Anal. Appl.*, 28:971–1004, 2006.
[36] V. Mehrmann and C. Schröder. Nonlinear eigenvalue and frequency response problems in industrial practice. *J. Math. in Industry*, 1:7, 2011. http://www.mathematicsinindustry.com/.
[37] V. Mehrmann and H. Voss. Nonlinear eigenvalue problems: A challenge for modern eigenvalue methods. *GAMM Mitt. Ges. Angew. Math. Mech.*, 27:121–152, 2004.
[38] E. Moreno, D. Erni, and C. Hafner. Band structure computations of metallic photonic crystals with the multiple multipole method. *Phys. Rev. B*, 65:155120, 2002.
[39] B. N. Parlett. *The Symmetric Eigenvalue Problem*. Englewood Cliffs, NJ, 1980.
[40] A. Ruhe. The rational Krylov algorithm for nonlinear matrix eigenvalue problems. *Zap. Nauchn. Semin. POMI*, 268:176–180, 2000.
[41] K. Schreiber. *Nonlinear eigenvalue problems: Newton type methods and nonlinear Rayleigh functionals*. PhD thesis, Inst. f. Mathematik, TU Berlin, Berlin, FRG, 2008.
[42] G. L. G. Sleijpen, A. G. L. Booten, D. R. Fokkema, and H. A. van der Vorst. Jacobi-Davidson

type methods for generalized eigenproblems and polynomial eigenproblems. *BIT*, 36:595–633, 1996.

[43] G. L. G. Sleijpen and H. A. van der Vorst. A Jacobi-Davidson iteration method for linear eigenvalue problems. *SIAM J. Matrix Anal. Appl.*, 17:401–425, 1996.

[44] F. Tisseur. Backward error and condition of polynomial eigenvalue problems. *Linear Alg. Appl.*, 309:339–361, 2000.

[45] F. Tisseur and K. Meerbergen. The quadratic eigenvalue problem. *SIAM Rev.*, 43:235–286, 2001.

[46] A. Vial. Implementation of the critical points model in the recursive convolution method for modelling dispersive media with the finite-difference time domain method. *J. Opt. A: Pure Appl. Opt.*, 9:745–748, 2007.

[47] H. Voss. An Arnoldi method for nonlinear eigenvalue problems. *BIT*, 44:387–401, 2004.

[48] H. Voss. A Jacobi-Davidson method for nonlinear and nonsymmetric eigenproblems. *Comput. Struct.*, 85:1284–1292, 2007.

[49] W. Wang, T.-M. Hwang, W.-W. Lin, and J.-L. Liu. Numerical methods for semiconductor heterostructures with band nonparabolicity. *J. Comput. Phys.*, 190:141–158, 2003.

[50] K. Yee. Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media. *IEEE Trans. Antennas and Propagation*, 14:302–307, 1966.

[51] Y. Zhao, C. Argyropoulos, and Y. Hao. Full-wave finite-difference time-domain simulation of electromagnetic cloaking structures. *Opt. Express*, 16:6717–6730, 2008.

[52] R. Ziolkowski. Pulsed and CW Gaussian beam interactions with double negative metamaterial slabs. *Opt. Express*, 11:662–681, 2003.