SIAM J. Sci. Comput.
Vol. 37, No. 1, pp. A1–A29

A NULL SPACE FREE JACOBI–DAVIDSON ITERATION FOR MAXWELL'S OPERATOR[*]

YIN-LIANG HUANG[†], TSUNG-MING HUANG[‡], WEN-WEI LIN[§], AND WEI-CHENG WANG[¶]

Abstract. We present an efficient null space free Jacobi–Davidson method to compute the positive eigenvalues of time harmonic Maxwell's equations. We focus on a class of spatial discretizations that guarantee the existence of discrete vector potentials, such as Yee's scheme and the edge elements. During the Jacobi–Davidson iteration, the correction process is applied to the vector potential instead. The correction equation is solved approximately as in the standard Jacobi–Davidson approach. The computational cost of the transformation from the vector potential to the corrector is negligible. As a consequence, the expanding subspace automatically stays out of the null space and no extra projection step is needed. Numerical evidence confirms that the proposed scheme indeed outperforms the standard and projection-based Jacobi–Davidson methods by a significant margin.

Key words. time harmonic Maxwell's equations, Yee's scheme, edge elements, generalized eigenvalue problem, discrete vector potential, discrete deRham complex, Poincaré Lemma, Jacobi–Davidson method

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

DOI. 10.1137/140954714

1. Introduction.

Photonic crystals are made of dielectric materials with periodic structure. The shape and permittivity of the dielectric material completely determines the band structure of the photonic crystal. Over the past few decades, photonic crystals with specific band structures have been of practical interest and have been extensively studied. The governing equation for three-dimensional photonic crystals is the time harmonic Maxwell's equations:

$$\nabla \times \boldsymbol{H} = i\omega\varepsilon\boldsymbol{E}, \tag{1.1}$$

$$\nabla \times \boldsymbol{E} = -i\omega\mu\boldsymbol{H}, \tag{1.2}$$

$$\nabla \cdot (\varepsilon\boldsymbol{E}) = 0, \tag{1.3}$$

$$\nabla \cdot (\mu\boldsymbol{H}) = 0, \tag{1.4}$$

where ω is the frequency and $\varepsilon = \varepsilon_r\varepsilon_0$, $\mu = \mu_r\mu_0$. The constants $\varepsilon_0 = 8.854 \times 10^{-12}$ Farad/meter and $\mu_0 = 1.257 \times 10^{-6}$ Henry/meter represent vacuum permittivity

[*] Submitted to the journal's Methods and Algorithms for Scientific Computing section January 28, 2014; accepted for publication (in revised form) August 25, 2014; published electronically January 8, 2015. This work is sponsored in part by the National Center of Theoretical Sciences of Taiwan and ST Yau Center at National Chiao Tung University.

http://www.siam.org/journals/sisc/37-1/95471.html

[†] Department of Applied Mathematics, National University of Tainan, Tainan, 700, Taiwan (liang@mail.nutn.edu.tw). This author was supported by the Ministry of Science and Technology (101-2115-M-024-002).

[‡] Department of Mathematics, National Taiwan Normal University, Taipei, 116, Taiwan (min@ntnu.edu.tw). This author was supported by the Ministry of Science and Technology (102-2115-M-003-011-MY3).

[§] Department of Applied Mathematics, National Chiao Tung University, HsinChu, 300, Taiwan (wwlin@math.nctu.edu.tw). This author was supported by the Ministry of Science and Technology (103-2628-M-009-005).

[¶] Department of Mathematics, National Tsing Hua University and National Center for Theoretical Sciences, HsinChu, 300, Taiwan (wangwc@math.nthu.edu.tw). This author was supported by the Ministry of Science and Technology (101-2115-M-007-002-MY2).

and vacuum permeability, respectively. The relative permittivity ε_r and relative permeability μ_r are dimensionless, material dependent parameters.

We can recast the Maxwell's equations in terms of the electric field $\boldsymbol{E}$ alone:

$$\nabla \times \mu_r^{-1} \nabla \times \boldsymbol{E} = \lambda\, \varepsilon_r \boldsymbol{E}, \tag{1.5}$$

$$\nabla \cdot (\varepsilon_r \boldsymbol{E}) = 0, \tag{1.6}$$

where $\lambda = \varepsilon_0 \mu_0 \omega^2$ is the eigenvalue. The degenerate elliptic operator $\nabla \times \mu_r^{-1} \nabla \times$ is self-adjoint and nonnegative. Since μ_r and ε_r are material dependent and therefore piecewise constant, (1.5) constitutes an elliptic interface problem [35, 16]. Equations (1.5) and (1.6) need to be supplied with appropriate boundary conditions which we will elaborate in the beginning of section 2.

Equation (1.6) serves as a constraint for the degenerate elliptic equation (1.5) and is redundant for the nonzero eigenvalues $\lambda \neq 0$ as a consequence of the basic identity from calculus

$$\nabla \cdot \nabla \times \equiv 0. \tag{1.7}$$

A traditional wisdom to reflect this fact is to adopt spatial discretizations that admit discrete analogue of (1.7). This class of spatial discretizations includes the Yee's scheme [37], the Whitney form [7, 36], the co-volume discretization [25], the mimetic discretization [20], and the edge element [23, 24, 28]. With this approach, the divergence free constraint (1.6) is ignored and the resulting discretized system is a generalized eigenvalue problem

$$\mathbb{A} \boldsymbol{e} = \lambda \mathbb{B} \boldsymbol{e}, \tag{1.8}$$

where $\mathbb{A}$ is the matrix representation of the discretized $\nabla \times \mu_r^{-1} \nabla \times$ and $\mathbb{B}$ is the mass matrix.

In the unconstrained formulation (1.8), the matrix $\mathbb{A}$ is symmetric and nonnegative semidefinite. The major difficulty with this approach, however, is the large null space associated with (1.5) in the absence of (1.6). This can be seen easily from the identity

$$\nabla \times \nabla \phi \equiv \mathbf{0}. \tag{1.9}$$

In other words, the null space of $\nabla \times \mu_r^{-1} \nabla \times$ contains all the gradient vectors. The discrete counterpart of (1.9) holds true for the generalized eigenvalue problem (1.8) with the class of spatial discretizations under consideration. As a result, a huge spurious null space arises from discarding the divergence free constraint (1.6). In Yee's discretization, for example, the dimension of the null space is the same as the number of cells, constituting one third of the total degrees of freedom. This causes severe numerical difficulties in numerical computation since in practice, we are mainly interested in the lowest nonzero eigenvalues of (1.5), (1.6) which are now located deep in the interior of the spectrum.

In this paper, we propose a novel numerical scheme to handle the null space issue using a modified Jacobi–Davidson (JD) iteration. The JD method [1, 2, 3, 5, 12, 26, 31, 32] is a well-established, efficient eigensolver based on expanding subspace iteration. In contrast to classical eigensolvers such as the inverse power method [9, 15] and various Lanczos [2, 30] or Arnoldi methods [5], which require the shift-and-invert technique to compute interior eigenpairs, the linear system for the corrector in the JD iteration only needs to be solved approximately. This is the key to efficiency of the JD method. On the other hand, since the corrector is only solved approximately, direct application of the JD method inevitably brings null vectors into the expanding subspace. As a result, the efficiency of the JD method is significantly deteriorated. The pollution of the spurious null space is common to other eigensolvers as well. The standard remedy is to project the approximate eigenvector back to the orthogonal complement of the null space. A well-known approach for the projection is through the Helmholtz decomposition [1, 3, 15]. See also [9] for the approach of combining the CG method with multigrid iteration.

In this paper, we take a different approach and propose a null space free Jacobi–Davidson (NFJD) iteration inspired by the Poincaré lemma [10, 11, 29], whose discrete analogue remains valid for the class of spatial discretizations under consideration:

> *Poincaré lemma.* If $\boldsymbol{v} \in C^1(\Omega)$ and $\nabla \cdot \boldsymbol{v} = 0$ on a contractible domain Ω, then $\boldsymbol{v} = \nabla \times \hat{\boldsymbol{v}}$ for some vector potential $\hat{\boldsymbol{v}}$.

The Poincaré lemma on general domains can be found in, for example, [33, Theorem 1.5]. See also [14, Theorem 2.1] for the discrete counterpart.

Motivated by the Poincaré lemma, we will show in Theorem 2.2 the existence of discrete vector potentials for the relevant vector fields. Instead of solving the corrector directly, the novelty of our approach is to derive and solve an equation for the vector potential of the corrector. The vector potential only needs to be solved approximately as in the original Jacobi–Davidson approach. By taking the discrete curl of the approximate vector potential, we annihilate completely the components in the null space and obtain a good approximation of the corrector which is null vector free. As a result, the expanding subspace automatically satisfies the divergence free constraint (1.6). The total cost to get an approximate corrector from an approximate vector potential is a single sparse matrix-vector multiplication. This is by far much cheaper than any projection. Numerical experiments have confirmed that our approach effectively resolves the slow convergence issue caused by the spurious null space in the original JD iteration. In addition, our scheme also outperforms the projection-based method by a significant margin over a wide range of parameter regime and provides a competitive alternative to existing schemes.

The rest of the paper is organized as follows. In section 2, we present detailed mathematical formulations for the eigenvalue problem (1.8), including Theorem 2.2, which characterizes the subspace spanned by the eigenvectors perpendicular to the null space in terms of discrete vector potentials. This is the foundation of our new scheme. The vector potential approach admits a magnetic field interpretation which leads to an isospectral reformulation of (1.8). We then further interplay between these equivalent formulations in section 3 and derive our NFJD method as an application of Theorem 2.2. We also present a dual version of NFJD with the roles of electric field and magnetic field interchanged during the iteration. In section 4, we conduct extensive numerical comparison among the original JD method, the Helmholtz projected Jacobi–Davidson (HPJD), and NFJD to show the robustness and efficiency of NFJD. The finite element version of our scheme is described in the appendix.

2. Background and mathematical formulation.

The photonic crystals consist of dielectric materials fabricated in periodic structure. The relative permittivity $\varepsilon_r(\mathbf{x})$ and relative permeability $\mu_r(\mathbf{x})$, as material dependent parameters, are therefore periodic and piecewise constant. In other words,

$$(2.1)\qquad \begin{gathered} (\varepsilon_r(\mathbf{x}), \mu_r(\mathbf{x})) = \begin{cases} (\varepsilon_{r,1}, \mu_{r,1}) & \text{in material 1,} \\ (\varepsilon_{r,2}, \mu_{r,2}) & \text{in material 2,} \end{cases} \\ \varepsilon_r(\mathbf{x} + \mathbf{a}_\ell) = \varepsilon_r(\mathbf{x}), \quad \mu_r(\mathbf{x} + \mathbf{a}_\ell) = \mu_r(\mathbf{x}), \quad \ell = 1, 2, 3, \end{gathered}$$

where the lattice translation vectors $\mathbf{a}_\ell$, $\ell = 1, 2, 3$, span the primitive cell which extends periodically to form the photonic crystal.

From Bloch's theorem [21], the eigenfunctions of (1.5) satisfy the $\mathbf{k}$-periodic boundary condition:

$$\boldsymbol{E}(\mathbf{x} + \mathbf{a}_\ell) = e^{\sqrt{-1}\mathbf{k}\cdot\mathbf{a}_\ell}\boldsymbol{E}(\mathbf{x}), \quad \ell = 1, 2, 3, \tag{2.2}$$

for some vector $\mathbf{k}$ in the first Brillouin zone.

To apply our scheme, we will only consider a class of spatially compatible discretizations satisfying discrete analogue of (1.7). Such discretizations include Yee's scheme, the co-volume discretization, and the edge elements. In this paper, we only report numerical results from Yee's discretization due to its simplicity in implementation. In addition, it is also easier to find an efficient preconditioner for the corresponding linear system.

For simplicity of presentation, we assume that the primitive cell is a unit cube spanned by the basis vectors

$$\mathbf{a}_1 = (1, 0, 0), \qquad \mathbf{a}_2 = (0, 1, 0), \qquad \mathbf{a}_3 = (0, 0, 1). \tag{2.3}$$

The corresponding first Brillouin zone is given by

$$\{\mathbf{k} = (\mathrm{k}_1, \mathrm{k}_2, \mathrm{k}_3) \in \mathbb{R}^3 \mid -\pi \leq \mathrm{k}_j \leq \pi,\ j = 1, 2, 3\}. \tag{2.4}$$

In Yee's discretization for (1.1)–(1.4), the magnetic field and electric field are defined on different locations. The centers of cell edges, cell faces, and cell centers are abbreviated as

$$\mathcal{E} = \mathcal{E}_1 \cup \mathcal{E}_2 \cup \mathcal{E}_3, \tag{2.5}$$

$$\mathcal{E}_1 = \{(x_{i-\frac{1}{2}}, y_j, z_k)\}, \quad \mathcal{E}_2 = \{(x_i, y_{j-\frac{1}{2}}, z_k)\}, \quad \mathcal{E}_3 = \{(x_i, y_j, z_{k-\frac{1}{2}})\}, \tag{2.6}$$

$$\mathcal{F} = \mathcal{F}_1 \cup \mathcal{F}_2 \cup \mathcal{F}_3, \tag{2.7}$$

$$\mathcal{F}_1 = \{(x_i, y_{j-\frac{1}{2}}, z_{k-\frac{1}{2}})\}, \quad \mathcal{F}_2 = \{(x_{i-\frac{1}{2}}, y_j, z_{k-\frac{1}{2}})\}, \quad \mathcal{F}_3 = \{(x_{i-\frac{1}{2}}, y_{j-\frac{1}{2}}, z_k)\}, \tag{2.8}$$

$$\mathcal{V} = \{(x_i, y_j, z_k)\}, \tag{2.9}$$

where $1 \leq i \leq N_1$, $1 \leq j \leq N_2$, $1 \leq k \leq N_3$.

We denote by $\mathcal{V}_{\mathcal{E}}$, $\mathcal{V}_{\mathcal{F}}$, and $\mathcal{V}_{\mathcal{V}}$ the spaces of complex valued, $\mathbf{k}$-periodic functions on $\mathcal{E}$, $\mathcal{F}$, and $\mathcal{V}$, respectively:

$$\begin{aligned} \mathcal{V}_{\mathcal{E}} = \big\{ & E_1(x_{i-\frac{1}{2}}, y_j, z_k),\ E_2(x_i, y_{j-\frac{1}{2}}, z_k),\ E_3(x_i, y_j, z_{k-\frac{1}{2}}) \,\big| \\ & E_m(\mathbf{x} + \mathbf{a}_\ell) = e^{\sqrt{-1}\mathbf{k}\cdot\mathbf{a}_\ell}E_m(\mathbf{x}),\ m, \ell = 1, 2, 3 \big\} \cong \mathbb{C}^{3N_1N_2N_3}, \end{aligned} \tag{2.10}$$

$$\begin{aligned} \mathcal{V}_{\mathcal{F}} = \big\{ & F_1(x_i, y_{j-\frac{1}{2}}, z_{k-\frac{1}{2}}), F_2(x_{i-\frac{1}{2}}, y_j, z_{k-\frac{1}{2}}), F_3(x_{i-\frac{1}{2}}, y_{j-\frac{1}{2}}, z_k) \,\big| \\ & F_n(\mathbf{x} + \mathbf{a}_\ell) = e^{\sqrt{-1}\mathbf{k}\cdot\mathbf{a}_\ell}F_n(\mathbf{x}),\ n, \ell = 1, 2, 3 \big\} \cong \mathbb{C}^{3N_1N_2N_3}, \end{aligned} \tag{2.11}$$

$$\mathcal{V}_{\mathcal{V}} = \big\{ \Phi(x_i, y_j, z_k) \,\big|\, \Phi(\mathbf{x} + \mathbf{a}_\ell) = e^{\sqrt{-1}\mathbf{k}\cdot\mathbf{a}_\ell}\Phi(\mathbf{x}),\ \ell = 1, 2, 3 \big\} \cong \mathbb{C}^{N_1N_2N_3}. \tag{2.12}$$

The curl operator can be discretized naturally using standard centered differencing:

$$\nabla_h \times : \mathcal{V}_{\mathcal{E}} \mapsto \mathcal{V}_{\mathcal{F}}. \tag{2.13}$$

For example, if $\boldsymbol{E} \in \mathcal{V}_{\mathcal{E}}$, then the first component of $\nabla_h \times \boldsymbol{E}$ is given by

$$\begin{aligned}(\nabla_h \times \boldsymbol{E})_1(x_i, y_{j-\frac{1}{2}}, z_{k-\frac{1}{2}}) = {} & \frac{E_3(x_i, y_j, z_{k-\frac{1}{2}}) - E_3(x_i, y_{j-1}, z_{k-\frac{1}{2}})}{h_2} \\ & - \frac{E_2(x_i, y_{j-\frac{1}{2}}, z_k) - E_2(x_i, y_{j-\frac{1}{2}}, z_{k-1})}{h_3},\end{aligned} \tag{2.14}$$

where $h_1 = 1/N_1$, $h_2 = 1/N_2$, $h_3 = 1/N_3$ are mesh sizes and N_1, N_2, N_3 are numbers of partitions in x, y, and z directions, respectively.

Similarly, one can define the discrete curl operator on $\mathcal{V}_{\mathcal{F}}$,

$$\nabla_h \times^* : \mathcal{V}_{\mathcal{F}} \mapsto \mathcal{V}_{\mathcal{E}} \tag{2.15}$$

with

$$\begin{aligned}(\nabla_h \times^* \boldsymbol{H})_1(x_{i-\frac{1}{2}}, y_j, z_k) = {} & \frac{H_3(x_{i-\frac{1}{2}}, y_{j+\frac{1}{2}}, z_k) - H_3(x_{i-\frac{1}{2}}, y_{j-\frac{1}{2}}, z_k)}{h_2} \\ & - \frac{H_2(x_{i-\frac{1}{2}}, y_j, z_{k+\frac{1}{2}}) - H_2(x_{i-\frac{1}{2}}, y_j, z_{k-\frac{1}{2}})}{h_3}\end{aligned} \tag{2.16}$$

and so on, as well as the discrete divergence operator on $\mathcal{V}_{\mathcal{E}}$,

$$-\nabla_h^* : \mathcal{V}_{\mathcal{E}} \mapsto \mathcal{V}_{\mathcal{V}}, \tag{2.17}$$

$$\begin{aligned}(-\nabla_h^* \boldsymbol{E})(x_i, y_j, z_k) = {} & \frac{E_1(x_{i+\frac{1}{2}}, y_j, z_k) - E_1(x_{i-\frac{1}{2}}, y_j, z_k)}{h_1} \\ & + \frac{E_2(x_i, y_{j+\frac{1}{2}}, z_k) - E_2(x_i, y_{j-\frac{1}{2}}, z_k)}{h_2} \\ & + \frac{E_3(x_i, y_j, z_{k+\frac{1}{2}}) - E_3(x_i, y_j, z_{k-\frac{1}{2}})}{h_3}.\end{aligned} \tag{2.18}$$

One can show that $-\nabla_h^*$ is indeed the adjoint of $-\nabla_h$, where the discrete gradient ∇_h is given by

$$\nabla_h : \mathcal{V}_{\mathcal{V}} \mapsto \mathcal{V}_{\mathcal{E}}, \tag{2.19}$$

$$(\nabla_h \Phi)_1(x_{i-\frac{1}{2}}, y_j, z_k) = \frac{\Phi(x_i, y_j, z_k) - \Phi(x_{i-1}, y_j, z_k)}{h_1}, \tag{2.20}$$

and similarly for $(\nabla_h \Phi)_2(x_i, y_{j-\frac{1}{2}}, z_k)$ and $(\nabla_h \Phi)_3(x_i, y_j, z_{k-\frac{1}{2}})$.

Even though (2.18) is a natural finite difference interpretation of discrete divergence operator on $\mathcal{V}_{\mathcal{E}}$, we have adopted $-\nabla_h^*$ instead of the usual notation $\nabla_h \cdot$ so that the notation is consistent with that of the edge element discretization detailed in Appendix A. In the latter case, the divergence free constraint can only be realized through the adjoint of the gradient operator.

The crucial identity

$$-\nabla_h^* \ \nabla_h \times^* \equiv 0, \tag{2.21}$$

a discrete analogue of the identity (1.7), follows from straightforward calculation. The resulting generalized eigenvalue problem can be written as

$$\nabla_h \times^* \mu_{r,h}^{-1} \nabla_h \times \boldsymbol{E} = \lambda \varepsilon_{r,h} \boldsymbol{E}, \quad \boldsymbol{E} \in \mathcal{V}_{\mathcal{E}}, \tag{2.22}$$

where $\varepsilon_{r,h} : \mathcal{V}_{\mathcal{E}} \mapsto \mathcal{V}_{\mathcal{E}}$ and $\mu_{r,h}^{-1} : \mathcal{V}_{\mathcal{F}} \mapsto \mathcal{V}_{\mathcal{F}}$ represent (multiplication by) numerical approximations of ε_r and μ_r^{-1}, respectively. Both $\nabla_h \times^*$ and $\nabla_h \times$ are discrete approximations of the curl operator $\nabla \times$; they act on different spaces and are adjoint to each other. See Lemma 2.1 below.

For $\boldsymbol{E}, \boldsymbol{U} \in \mathcal{V}_{\mathcal{E}}$ and $\boldsymbol{F}, \boldsymbol{H} \in \mathcal{V}_{\mathcal{F}}$, we denote by $\langle \cdot, \cdot \rangle_{\mathcal{E}}$ and $\langle \cdot, \cdot \rangle_{\mathcal{F}}$ the standard inner products on $\mathcal{V}_{\mathcal{E}}$ and $\mathcal{V}_{\mathcal{F}}$, respectively:

$$\langle \boldsymbol{E}, \boldsymbol{U} \rangle_{\mathcal{E}} = h_1 h_2 h_3 \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \sum_{k=1}^{N_3} \Big((\overline{E}_1 U_1)_{i-\frac{1}{2},j,k} + (\overline{E}_2 U_2)_{i,j-\frac{1}{2},k} + (\overline{E}_3 U_3)_{i,j,k-\frac{1}{2}} \Big), \tag{2.23}$$

$$\begin{aligned} \langle \boldsymbol{F}, \boldsymbol{H} \rangle_{\mathcal{F}} = h_1 h_2 h_3 \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \sum_{k=1}^{N_3} \Big(&(\overline{F}_1 H_1)_{i,j-\frac{1}{2},k-\frac{1}{2}} \\ &+ (\overline{F}_2 H_2)_{i-\frac{1}{2},j,k-\frac{1}{2}} + (\overline{F}_3 H_3)_{i-\frac{1}{2},j-\frac{1}{2},k} \Big). \end{aligned} \tag{2.24}$$

The following lemma explains the notation used for the two discrete curl operators in (2.16), (2.14) and is crucial to the development of our scheme.

Lemma 2.1. *If $\boldsymbol{U}$, $\boldsymbol{V} \in \mathcal{V}_{\mathcal{E}}$, then*

$$\langle \boldsymbol{U}, \nabla_h \times^* \mu_{r,h}^{-1} \nabla_h \times \boldsymbol{V} \rangle_{\mathcal{E}} = \langle \nabla_h \times \boldsymbol{U}, \mu_{r,h}^{-1} \nabla_h \times \boldsymbol{V} \rangle_{\mathcal{F}}. \tag{2.25}$$

Proof. It suffices to show that

$$\langle \boldsymbol{U}, \nabla_h \times^* \boldsymbol{H} \rangle_{\mathcal{E}} = \langle \nabla_h \times \boldsymbol{U}, \boldsymbol{H} \rangle_{\mathcal{F}} \quad \text{for all } \boldsymbol{U} \in \mathcal{V}_{\mathcal{E}}, \ \boldsymbol{H} \in \mathcal{V}_{\mathcal{F}}. \tag{2.26}$$

This follows from the summation by parts identity and the **k**-periodic boundary condition. To see this, we denote by $\boldsymbol{W} = \nabla_h \times^* \boldsymbol{H} \in \mathcal{V}_{\mathcal{E}}$. We have

$$\begin{aligned} \langle \boldsymbol{U}, \nabla_h \times^* \boldsymbol{H} \rangle_{\mathcal{E}} &= \langle \boldsymbol{U}, \boldsymbol{W} \rangle_{\mathcal{E}} \\ &= h_1 h_2 h_3 \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \sum_{k=1}^{N_3} \Big((\overline{U}_1 W_1)_{i-\frac{1}{2},j,k} + (\overline{U}_2 W_2)_{i,j-\frac{1}{2},k} + (\overline{U}_3 W_3)_{i,j,k-\frac{1}{2}} \Big) \\ &:= h_1 h_2 h_3 (I_1 + I_2 + I_3), \end{aligned} \tag{2.27}$$

where

$$\begin{aligned} I_1 = \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \sum_{k=1}^{N_3} \overline{U}_{1,i-\frac{1}{2},j,k} \Bigg(&\frac{H_{3,i-\frac{1}{2},j+\frac{1}{2},k} - H_{3,i-\frac{1}{2},j-\frac{1}{2},k}}{h_2} \\ &- \frac{H_{2,i-\frac{1}{2},j,k+\frac{1}{2}} - H_{2,i-\frac{1}{2},j,k-\frac{1}{2}}}{h_3} \Bigg), \end{aligned} \tag{2.28}$$

and I_2, I_3 are defined similarly. The first term in I_1 can be rearranged as

$$\tag{2.29}\begin{aligned}
&\sum_{i=1}^{N_1}\sum_{k=1}^{N_3}\sum_{j=1}^{N_2}\overline{U}_{1,i-\frac{1}{2},j,k}\frac{H_{3,i-\frac{1}{2},j+\frac{1}{2},k}-H_{3,i-\frac{1}{2},j-\frac{1}{2},k}}{h_2}\\
&\quad=-\sum_{i=1}^{N_1}\sum_{k=1}^{N_3}\left(\frac{\overline{U}_{1,i-\frac{1}{2},1,k}}{h_2}H_{3,i-\frac{1}{2},\frac{1}{2},k}+\sum_{j=2}^{N_2}\frac{\overline{U}_{1,i-\frac{1}{2},j,k}-\overline{U}_{1,i-\frac{1}{2},j-1,k}}{h_2}H_{3,i-\frac{1}{2},j-\frac{1}{2},k}\right.\\
&\qquad\left.-\frac{\overline{U}_{1,i-\frac{1}{2},N_2,k}}{h_2}H_{3,i-\frac{1}{2},N_2+\frac{1}{2},k}\right)\\
&\quad=-\sum_{i=1}^{N_1}\sum_{k=1}^{N_3}\sum_{j=1}^{N_2}\frac{\overline{U}_{1,i-\frac{1}{2},j,k}-\overline{U}_{1,i-\frac{1}{2},j-1,k}}{h_2}H_{3,i-\frac{1}{2},j-\frac{1}{2},k},
\end{aligned}$$

where in the last equality we have used

$$\tag{2.30} U_{1,i-\frac{1}{2},N_2,k}=e^{\sqrt{-1}k_2}U_{1,i-\frac{1}{2},0,k},\quad H_{3,i-\frac{1}{2},N_2+\frac{1}{2},k}=e^{\sqrt{-1}k_2}H_{3,i-\frac{1}{2},\frac{1}{2},k},$$

which follows from the **k**-periodic boundary conditions (2.2), (2.3) imposed on $\boldsymbol{U}$ and $\boldsymbol{H}$. The second term in I_1 can be treated similarly. Overall, we have

$$\tag{2.31}\begin{aligned}
I_1=\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}\sum_{k=1}^{N_3}&\left(-\frac{\overline{U}_{1,i-\frac{1}{2},j,k}-\overline{U}_{1,i-\frac{1}{2},j-1,k}}{h_2}H_{3,i-\frac{1}{2},j-\frac{1}{2},k}\right.\\
&\left.+\frac{\overline{U}_{1,i-\frac{1}{2},j,k}-\overline{U}_{1,i-\frac{1}{2},j,k-1}}{h_3}H_{2,i-\frac{1}{2},j,k-\frac{1}{2}}\right),
\end{aligned}$$

and similarly

$$\tag{2.32}\begin{aligned}
I_2=\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}\sum_{k=1}^{N_3}&\left(-\frac{\overline{U}_{2,i,j-\frac{1}{2},k}-\overline{U}_{2,i,j-\frac{1}{2},k-1}}{h_3}H_{1,i,j-\frac{1}{2},k-\frac{1}{2}}\right.\\
&\left.+\frac{\overline{U}_{2,i,j-\frac{1}{2},k}-\overline{U}_{2,i-1,j-\frac{1}{2},k}}{h_1}H_{3,i-\frac{1}{2},j-\frac{1}{2},k}\right),
\end{aligned}$$

$$\tag{2.33}\begin{aligned}
I_3=\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}\sum_{k=1}^{N_3}&\left(-\frac{\overline{U}_{3,i,j,k-\frac{1}{2}}-\overline{U}_{3,i-1,j,k-\frac{1}{2}}}{h_1}H_{2,i-\frac{1}{2},j,k-\frac{1}{2}}\right.\\
&\left.+\frac{\overline{U}_{3,i,j,k-\frac{1}{2}}-\overline{U}_{3,i,j-1,k-\frac{1}{2}}}{h_2}H_{1,i,j-\frac{1}{2},k-\frac{1}{2}}\right).
\end{aligned}$$

Consequently, (2.26) follows from (2.27), (2.31), (2.32), and (2.33). ▯

From Lemma 2.1, the operator $\nabla_h \times^* \mu_{r,h}^{-1}\nabla_h\times$ is self-adjoint on $\mathcal{V}_{\mathcal{E}}$ and semidefinite. In addition, the operator $\varepsilon_{r,h}\cdot$ is positive definite. It follows that the eigenvalues in (2.22) are real and nonnegative. The eigenvectors constitute a basis in $\mathcal{V}_{\mathcal{E}}$ and are orthogonal with respect to the inner product induced by $\varepsilon_{r,h}$:

$$\begin{aligned}\langle \boldsymbol{E}, \boldsymbol{U}\rangle_{\mathcal{E},\varepsilon_{r,h}} = h_1h_2h_3 \sum_{i=1}^{N_1}\sum_{j=1}^{N_2}\sum_{k=1}^{N_3} \Big(&(\overline{E}_1 \varepsilon_{r,h} U_1)_{i-\frac{1}{2},j,k} \\ &+ (\overline{E}_2 \varepsilon_{r,h} U_2)_{i,j-\frac{1}{2},k} + (\overline{E}_3 \varepsilon_{r,h} U_3)_{i,j,k-\frac{1}{2}} \Big).\end{aligned} \tag{2.34}$$

We therefore have the following eigendecomposition:

$$\mathcal{V}_{\mathcal{E}} = \ker(\nabla_h \times) \oplus \ker(\nabla_h \times)^{\perp_{\varepsilon_{r,h}}}, \tag{2.35}$$

where

$$\ker(\nabla_h \times) = \ker\left(\nabla_h \times^* \mu_{r,h}^{-1} \nabla_h \times\right) = \{\boldsymbol{V} \in \mathcal{V}_{\mathcal{E}},\ \nabla_h \times \boldsymbol{V} = \boldsymbol{0}\} \tag{2.36}$$

and

$$\begin{aligned}\ker(\nabla_h \times)^{\perp_{\varepsilon_{r,h}}} &= \left\{\boldsymbol{U} \in \mathcal{V}_{\mathcal{E}} \,|\, \langle \boldsymbol{U}, \boldsymbol{V}\rangle_{\varepsilon_{r,h}} = 0,\ \forall\ \boldsymbol{V} \in \ker(\nabla_h \times)\right\} \\ &= \operatorname{Span}\{\boldsymbol{V}_j \in \mathcal{V}_{\mathcal{E}} \,|\, \nabla_h \times^* \mu_{r,h}^{-1} \nabla_h \times \boldsymbol{V}_j = \lambda_j \varepsilon_{r,h} \boldsymbol{V}_j,\ \lambda_j > 0\}.\end{aligned} \tag{2.37}$$

The following main theorem is the foundation of our null space free algorithm.

THEOREM 2.2. *$\boldsymbol{U} \in \ker(\nabla_h \times)^{\perp_{\varepsilon_{r,h}}}$ if and only if $\varepsilon_{r,h}\boldsymbol{U} = \nabla_h \times^* \mu_{r,h}^{-1} \widehat{\boldsymbol{U}}$ for some discrete vector potential $\widehat{\boldsymbol{U}} \in \mathcal{V}_{\mathcal{F}}$.*

Proof. Denote by $(\lambda_j, \boldsymbol{V}_j)$ the eigenpairs of the generalized eigenvalue problem (2.22).

Suppose that $\boldsymbol{U} \in \ker(\nabla_h \times)^{\perp_{\varepsilon_{r,h}}}$; then there exist constants a_j such that $\boldsymbol{U} = \sum_{\lambda_j > 0} a_j \boldsymbol{V}_j$. Thus

$$\begin{aligned}\varepsilon_{r,h}\boldsymbol{U} = \varepsilon_{r,h} \sum_{\lambda_j>0} a_j \boldsymbol{V}_j &= \sum_{\lambda_j>0} a_j \nabla_h \times^* \mu_{r,h}^{-1} \nabla_h \times \lambda_j^{-1} \boldsymbol{V}_j \\ &= \nabla_h \times^* \Big(\mu_{r,h}^{-1} \sum_{\lambda_j>0} a_j \nabla_h \times \lambda_j^{-1} \boldsymbol{V}_j\Big) = \nabla_h \times^* \mu_{r,h}^{-1} \widehat{\boldsymbol{U}},\end{aligned} \tag{2.38}$$

where $\widehat{\boldsymbol{U}} = \sum\limits_{\lambda_j>0} a_j \nabla_h \times (\lambda_j^{-1} \boldsymbol{V}_j) \in \mathcal{V}_{\mathcal{F}}$.

Conversely, we suppose that $\varepsilon_{r,h}\boldsymbol{U} = \nabla_h \times^* \mu_{r,h}^{-1} \widehat{\boldsymbol{U}}$ with $\widehat{\boldsymbol{U}} \in \mathcal{V}_{\mathcal{F}}$. Then for any $\boldsymbol{V} \in \ker(\nabla_h \times)$, we have

$$\langle \boldsymbol{U}, \boldsymbol{V}\rangle_{\mathcal{E},\varepsilon_{r,h}} = \langle \varepsilon_{r,h}\boldsymbol{U}, \boldsymbol{V}\rangle_{\mathcal{E}} = \langle \nabla_h \times^* \mu_{r,h}^{-1} \widehat{\boldsymbol{U}}, \boldsymbol{V}\rangle_{\mathcal{E}} = \langle \mu_{r,h}^{-1} \widehat{\boldsymbol{U}},\, \nabla_h \times \boldsymbol{V}\rangle_{\mathcal{F}} = 0. \tag{2.39}$$

Thus $\boldsymbol{U} \in \ker(\nabla_h \times)^{\perp_{\varepsilon_{r,h}}}$, completing the proof. ∎

The matrix representation of the discrete curl operator $\nabla_h \times : \mathcal{V}_{\mathcal{E}} \mapsto \mathcal{V}_{\mathcal{F}}$ is given by

$$\mathbb{C} = \begin{bmatrix} 0 & \dfrac{-1}{h_3}\mathbb{K}_3 & \dfrac{1}{h_2}\operatorname{diag}(\mathbb{K}_2, \ldots, \mathbb{K}_2) \\ \dfrac{1}{h_3}\mathbb{K}_3 & 0 & \dfrac{-1}{h_1}\operatorname{diag}(\mathbb{K}_1, \ldots, \mathbb{K}_1) \\ \dfrac{-1}{h_2}\operatorname{diag}(\mathbb{K}_2, \ldots, \mathbb{K}_2) & \dfrac{1}{h_1}\operatorname{diag}(\mathbb{K}_1, \ldots, \mathbb{K}_1) & 0 \end{bmatrix} \in \mathbb{C}^{3N \times 3N},$$

where

$$\mathbb{K}_1 = \begin{bmatrix} 1 & & & -e^{-\sqrt{-1}\mathrm{k}_1} \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix} \in \mathbb{C}^{N_1 \times N_1}, \tag{2.40a}$$

$$\mathbb{K}_2 = \begin{bmatrix} \mathbb{I}_{N_1} & & & -e^{-\sqrt{-1}\mathrm{k}_2}\mathbb{I}_{N_1} \\ -\mathbb{I}_{N_1} & \mathbb{I}_{N_1} & & \\ & \ddots & \ddots & \\ & & -\mathbb{I}_{N_1} & \mathbb{I}_{N_1} \end{bmatrix} \in \mathbb{C}^{(N_1N_2)\times(N_1N_2)}, \tag{2.40b}$$

$$\mathbb{K}_3 = \begin{bmatrix} \mathbb{I}_{N_1\times N_2} & & & -e^{-\sqrt{-1}\mathrm{k}_3}\mathbb{I}_{N_1\times N_2} \\ -\mathbb{I}_{N_1\times N_2} & \mathbb{I}_{N_1\times N_2} & & \\ & \ddots & \ddots & \\ & & -\mathbb{I}_{N_1\times N_2} & \mathbb{I}_{N_1\times N_2} \end{bmatrix} \in \mathbb{C}^{N\times N}, \tag{2.40c}$$

and $N = N_1N_2N_3$.

The matrix representation of (2.22) is thus

$$\mathbb{A}\boldsymbol{e} = \lambda\mathbb{B}\boldsymbol{e}, \tag{2.41}$$

where $\mathbb{A} = \mathbb{C}^*\widehat{\mathbb{B}}\mathbb{C}$. The diagonal matrices $\mathbb{B}$ and $\widehat{\mathbb{B}}$ represent multiplication by $\varepsilon_{r,h}(\mathbf{x})$ for $\mathbf{x} \in \mathcal{E}$ and multiplication by $\mu_{r,h}^{-1}(\mathbf{x})$ for $\mathbf{x} \in \mathcal{F}$, respectively.

Before we proceed, we briefly summarize our notation for the reader's convenience.

(a) Hatted uppercase boldface characters denote discrete vector potentials of elements in $\mathcal{V}_{\mathcal{E}}$. The latter are denoted by (nonhatted) uppercase boldface characters. For example, $\widehat{\boldsymbol{V}} \in \mathcal{V}_{\mathcal{F}}$ is a vector potential of $\boldsymbol{V} \in \mathcal{V}_{\mathcal{E}}$ with $\nabla_h \times^* \mu_{r,h}^{-1}\widehat{\boldsymbol{V}} = \varepsilon_{r,h}\boldsymbol{V}$.

(b) Lowercase boldface characters denote the column vector representation of corresponding elements in $\mathcal{V}_{\mathcal{E}}$ or $\mathcal{V}_{\mathcal{F}}$. For example, both $\boldsymbol{T} \in \mathcal{V}_{\mathcal{E}}$ and $\widehat{\boldsymbol{T}} \in \mathcal{V}_{\mathcal{F}}$ have three components, each of them being an $N_1 \times N_2 \times N_3$ array, while $\boldsymbol{t}$ and $\widehat{\boldsymbol{t}}$ are $3N_1N_2N_3 \times 1$ column vector representation of $\boldsymbol{T}$ and $\widehat{\boldsymbol{T}}$, respectively.

(c) Uppercase blackboard bold characters such as $\mathbb{A}$, $\widehat{\mathbb{B}}$, $\mathbb{I}$ denote matrices of varying dimensions.

3. JD method, Helmholtz projection, and vector potential.

The JD method [32] is a subspace iteration algorithm for large sparse eigenvalue problems and has been proved successful in many practical applications such as various quantum dot models [17, 18, 19, 34].

The major advantage of the JD method for general eigenvalue problem is that the correction equation only needs to be solved approximately. However, this advantage becomes a drawback for (2.22) as the correcting procedure inevitably brings null space components into the expanding subspace, causing slow convergence.

A simple remedy is to project out the null space components by means of the Helmholtz decomposition. The Helmholtz projection has been applied in conjunction with both the inverse power iteration [15] and the JD iteration [3] and obtained decent convergence rate. However, there is an essential difference in the two projection-based approaches. While inverse power iteration requires accurate Helmholtz projection in order to maintain accuracy of the numerical eigenpairs, the JD method seems to be quite sensitive to inexact projections and therefore much more demanding on the accuracy of the Helmholtz projection step. We found that for the Helmholtz projected JD method to work properly, the projection step needs to be solved much more accurately than the generalized eigenvalue problem (2.22) itself. More detailed, quantitative demonstration of this assertion can be found in section 4.

In contrast, our remedy to the spurious null space components is to initialize and expand the approximating subspace in terms of the discrete vector potential given in Theorem 2.2. Instead of solving the correction equation directly, we lift the correction equation and the approximate solver to the vector potential level. The vector potential version of the correction equation is similar to the original one. As in the original JD method, the vector potential only needs to be solved approximately. An approximate corrector is then obtained by taking the discrete curl of the approximate vector potential. The null space components are then annihilated completely to machine accuracy at the expense of a single sparse matrix multiplication. As a result, in our vector potential based algorithm, the approximating subspace remains null space free throughout the iteration.

In this section, we first review the original JD method and HPJD method and then introduce our NFJD method. A detailed numerical comparison of these methods will be given in section 4. Our numerical experiment confirms that the vector potential approach indeed yields superior performance against original and projection-based JD methods.

3.1. JD method. The JD method for the generalized eigenvalue problem (2.22) consists of the following steps:

1. To compute the ith eigenpair $(\lambda_i, \boldsymbol{E}_i)$, one initializes a subspace $\mathcal{V}_1 := \mathrm{Span}\{\boldsymbol{E}_1, \ldots, \boldsymbol{E}_{i-1}, \boldsymbol{V}_1\} \subset \mathcal{V}_{\mathcal{E}}$, where $\boldsymbol{E}_1, \ldots, \boldsymbol{E}_{i-1}$, $i \geq 2$, are previously computed eigenvectors corresponding to eigenvalues $\lambda_1, \ldots, \lambda_{i-1}$.
2. For $k = 1, 2, 3, \ldots$, do

   (i) Find $\theta \in \mathbb{R} \setminus \{\lambda_1, \ldots, \lambda_{i-1}\}$ nearest to the target and $\boldsymbol{U} \in \mathcal{V}_k$, with $\|\boldsymbol{U}\|_{\varepsilon_{r,h}} = 1$ such that

   $$\left\langle (\nabla_h \times^* \mu_{r,h}^{-1} \nabla_h \times \; - \theta \varepsilon_{r,h}) \boldsymbol{U} \, , \, \boldsymbol{V} \right\rangle_{\mathcal{E}} = 0 \quad \text{for all} \quad \boldsymbol{V} \in \mathcal{V}_k. \tag{3.1}$$

   Denote by $(\theta_k, \boldsymbol{U}_k)$ the solution to (3.1).

   (ii) Set

   $$\boldsymbol{R}_k := (\nabla_h \times^* \mu_{r,h}^{-1} \nabla_h \times \; - \theta_k \varepsilon_{r,h}) \boldsymbol{U}_k. \tag{3.2}$$

   (iii) If $\|\boldsymbol{R}_k\|_{\mathcal{E}}$ is less than a prescribed tolerance, then $\big(\theta_k, \boldsymbol{U}_k\big)$ is accepted as an eigenpair. Output $(\lambda_i, \boldsymbol{E}_i) = (\theta_k, \boldsymbol{U}_k)$. Stop.
   Else, solve approximately for $\boldsymbol{T}$ from

   $$\begin{gathered} \big(\mathbb{I} - \varepsilon_{r,h} \boldsymbol{U}_k \otimes \boldsymbol{U}_k\big) \big(\nabla_h \times^* \mu_{r,h}^{-1} \nabla_h \times - \theta_k \varepsilon_{r,h}\big) \big(\mathbb{I} - \boldsymbol{U}_k \otimes (\varepsilon_{r,h} \boldsymbol{U}_k)\big) \boldsymbol{T} = -\boldsymbol{R}_k, \\ \boldsymbol{T} \perp_{\varepsilon_{r,h}} \boldsymbol{U}_k, \end{gathered} \tag{3.3}$$

   where $(\mathbb{I} - \boldsymbol{U} \otimes \boldsymbol{V}) \boldsymbol{W} := \boldsymbol{W} - \langle \boldsymbol{V}, \boldsymbol{W} \rangle_{\mathcal{E}} \boldsymbol{U}$.

Orthonormalize $\boldsymbol{T}$ against $\mathcal{V}_k$ with respect to the inner product $\langle\cdot,\cdot\rangle_{\mathcal{E},\varepsilon_{r,h}}$ to get $\boldsymbol{V}_{k+1}$.
Expand $\mathcal{V}_{k+1} := \text{Span}\{\,\mathcal{V}_k\,,\,\boldsymbol{V}_{k+1}\,\}$.

The implementation detail is summarized as Algorithm 1 in matrix-vector notation.

Algorithm 1. JD and HPJD methods. Additional projection steps in HPJD are marked by double parentheses.

```
Set $\mathbb{E}_0 = [\,]$, $\Lambda_0 = \varnothing$.
for $i = 1, 2, 3, \ldots, i_{\max}$ do
        Initialize a vector $\boldsymbol{v}_1$ with $\|\boldsymbol{v}_1\|_{\mathbb{B}} = 1$, $((\mathbb{P}_{\mathbb{B}}\boldsymbol{v}_1 = \boldsymbol{v}_1))$ and $\mathbb{E}_{i-1}^*\mathbb{B}\boldsymbol{v}_1 = \mathbf{0}$.
        Set $\mathbb{V}_1 = [\mathbb{E}_{i-1}, \boldsymbol{v}_1]$.
        Compute $\mathbb{W}_1 = \mathbb{V}_1^*\mathbb{A}\mathbb{V}_1$.
        for $k = 1, 2, 3, \ldots$ do
            (i) Compute all the eigenpairs of $(\mathbb{W}_k - \theta\,\mathbb{I})\boldsymbol{s} = \mathbf{0}$.
                Select the desired eigenpair $(\theta_k, \boldsymbol{s}_k)$ with $\theta_k \notin \Lambda_{i-1}$ nearest to the target and $\|\boldsymbol{s}_k\|_2 = 1$.
            (ii) Compute $\boldsymbol{u}_k = \mathbb{V}_k\boldsymbol{s}_k$, $\boldsymbol{r}_k = (\mathbb{A} - \theta_k\mathbb{B})\boldsymbol{u}_k$.
            (iii) if $\|\boldsymbol{r}_k\|_2 < \tau_{\text{JD}}$ then
                    Output $\lambda_i = \theta_k$, $\boldsymbol{e}_i = \boldsymbol{u}_k$.
                    Update $\mathbb{E}_i = [\mathbb{E}_{i-1}, \boldsymbol{e}_i]$, $\Lambda_i = \Lambda_{i-1} \cup \{\lambda_i\}$.
                    Exit $k$.
                else
                    Solve (approximately)
                      $(\mathbb{I} - \mathbb{B}\boldsymbol{u}_k\boldsymbol{u}_k^*)(\mathbb{A} - \theta_k\mathbb{B})(\mathbb{I} - \boldsymbol{u}_k\boldsymbol{u}_k^*\mathbb{B})\boldsymbol{t} = -\boldsymbol{r}_k, \quad \boldsymbol{t} \perp_{\mathbb{B}} \boldsymbol{u}_k$.
                    $\left(\left(\text{Apply Helmholtz projection: } \boldsymbol{t} \longleftarrow \mathbb{P}_{\mathbb{B}}\left(\frac{\boldsymbol{t}}{\|\mathbb{P}_{\mathbb{B}}\boldsymbol{t}\|}\right)\right)\right)$.
                    $\mathbb{B}$-orthonormalize $\boldsymbol{t}$ against $\mathbb{V}_k$: $\boldsymbol{v}_{k+1} = \dfrac{\boldsymbol{t} - \sum_{\ell=1}^{k}(\boldsymbol{v}_\ell^*\mathbb{B}\boldsymbol{t})\boldsymbol{v}_\ell}{\|\boldsymbol{t} - \sum_{\ell=1}^{k}(\boldsymbol{v}_\ell^*\mathbb{B}\boldsymbol{t})\boldsymbol{v}_\ell\|_{\mathbb{B}}}$.
                    Expand $\mathbb{V}_{k+1} = [\mathbb{V}_k, \boldsymbol{v}_{k+1}]$, $\mathbb{W}_{k+1} = \begin{bmatrix} \mathbb{W}_k & \mathbb{V}_k^*\mathbb{A}\boldsymbol{v}_{k+1} \\ \boldsymbol{v}_{k+1}^*\mathbb{A}\mathbb{V}_k & \boldsymbol{v}_{k+1}^*\mathbb{A}\boldsymbol{v}_{k+1} \end{bmatrix}$.
                end if
        end for $k$
end for $i$
```

The main computational cost in the original JD method is in step (iii), where the correction equation

$$(\mathbb{I} - \mathbb{B}\boldsymbol{u}_k\boldsymbol{u}_k^*)(\mathbb{A} - \theta_k\mathbb{B})(\mathbb{I} - \boldsymbol{u}_k\boldsymbol{u}_k^*\mathbb{B})\boldsymbol{t} = -\boldsymbol{r}_k, \quad \boldsymbol{t} \perp_{\mathbb{B}} \boldsymbol{u}_k, \tag{3.4}$$

is solved by standard iterative methods such as GMRES with a preconditioner [32],

$$\mathbb{M}_p := (\mathbb{I} - \mathbb{B}\boldsymbol{u}_k\boldsymbol{u}_k^*)\mathbb{M}(\mathbb{I} - \boldsymbol{u}_k\boldsymbol{u}_k^*\mathbb{B}), \tag{3.5}$$

where $\mathbb{M}$ is a preconditioner for $(\mathbb{A} - \theta_k\mathbb{B})$. In jth iteration of the linear solver, one solves for $\boldsymbol{z}^{(j)}$ from

$$\mathbb{M}_p\boldsymbol{z}^{(j)} = \boldsymbol{y}^{(j)}, \quad \boldsymbol{z}^{(j)} \perp_{\mathbb{B}} \boldsymbol{u}_k. \tag{3.6}$$

The solution to (3.6) is given by

$$\boldsymbol{z}^{(j)} = \mathbb{M}^{-1}\boldsymbol{y}^{(j)} - \zeta^{(j)}\mathbb{M}^{-1}\mathbb{B}\boldsymbol{u}_k, \quad \text{where} \quad \zeta^{(j)} = \frac{\boldsymbol{u}_k^*\mathbb{B}\mathbb{M}^{-1}\boldsymbol{y}^{(j)}}{\boldsymbol{u}_k^*\mathbb{B}\mathbb{M}^{-1}\mathbb{B}\boldsymbol{u}_k}. \tag{3.7}$$

Remark 3.1. The correction equation (3.3) is equivalent to

$$\begin{aligned}\left(\nabla_h \times^* \mu_{r,h}^{-1}\nabla_h \times\ -\ \theta_k\varepsilon_{r,h}\right)\boldsymbol{T} &= -\boldsymbol{R}_k + \eta\varepsilon_{r,h}\boldsymbol{U}_k\\ &= -\left(\nabla_h \times^* \mu_{r,h}^{-1}\nabla_h \times\ -\ \theta_k\varepsilon_{r,h}\right)\boldsymbol{U}_k + \eta\varepsilon_{r,h}\boldsymbol{U}_k,\end{aligned}\tag{3.8}$$

where $\eta = \left\langle \varepsilon_{r,h}\boldsymbol{U}_k,\ \left(\nabla_h \times^* \mu_{r,h}^{-1}\nabla_h \times\ -\ \theta_k\varepsilon_{r,h}\right)^{-1}\varepsilon_{r,h}\boldsymbol{U}_k\right\rangle_{\mathcal{E}}^{-1}$. If (3.3) were solved accurately, then one would have $\boldsymbol{T} = \frac{1}{\theta_k}\varepsilon_{r,h}^{-1}\nabla_h \times^* \left(\mu_{r,h}^{-1}\nabla_h \times (\boldsymbol{T}+\boldsymbol{U}_k)\right) - \frac{\theta_k+\eta}{\theta_k}\boldsymbol{U}_k \in \ker(\nabla_h\times)^{\perp_{\varepsilon_{r,h}}}$. In other words, the corrector $\boldsymbol{T}$ and hence the approximating subspace $\mathcal{V}_{k+1}$ would remain null vectors free. However, the same should not be expected when (3.3) is only solved approximately as in the original JD method (3.5)–(3.7). Under such circumstances, null vectors are inevitably introduced into $\mathcal{V}_{k+1}$ to deteriorate the overall performance.

3.2. HPJD method. To overcome the slowing down caused by the null space, a standard approach is to remove the null space components by means of Helmholtz decomposition. An approximate solution $\boldsymbol{t}$ of (3.4) is postprocessed with the Helmholtz projection before it is appended to the expanding subspace:

$$\boldsymbol{t} \longleftarrow \mathbb{P}_{\mathbb{B}}\left(\frac{\boldsymbol{t}}{\|\boldsymbol{t}\|_{\mathbb{B}}}\right) = \left(\mathbb{I} - \mathbb{G}(\mathbb{G}^*\mathbb{B}\mathbb{G})^{-1}\mathbb{G}^*\mathbb{B}\right)\left(\frac{\boldsymbol{t}}{\|\boldsymbol{t}\|_{\mathbb{B}}}\right).\tag{3.9}$$

Here $\mathbb{G}$ is the matrix representation of the discrete gradient $\nabla_h : \mathcal{V}_{\mathcal{V}} \mapsto \mathcal{V}_{\mathcal{E}}$, and $-\mathbb{G}^*$ is precisely the matrix representation of the discrete divergence operator (2.18).

The Helmholtz projection (3.9) requires solving an elliptic equation

$$-\mathbb{G}^*\mathbb{B}\mathbb{G}\,\phi = -\mathbb{G}^*\mathbb{B}\left(\frac{\boldsymbol{t}}{\|\boldsymbol{t}\|_{\mathbb{B}}}\right)\tag{3.10}$$

for each vector appended to the expanding subspace. The combination of the JD method and Helmholtz projection has been proposed in the literature [15]. In addition to a linear solver and preconditioner for the correction equation (3.4), the Helmholtz projection (3.9) also requires an efficient Poisson solver and preconditioner for (3.10). The overall performance of HPJD depends on the solver/preconditioner selected for the correction equation and the Poisson equation. Although solving the Poisson equation (3.10) is straightforward and much easier compared to solving the correction equation (3.4), the load balance between the Poisson equation (3.10) and the correction equation (3.4) is somewhat delicate. We will elaborate this issue in section 4. Both standard JD and HPJD methods are summarized in Algorithm 1.

3.3. Discrete vector potential and the NFJD method. Instead of the Helmholtz projection, we propose an alternative approach by vector potential formulation in order to filter out the null space components. The novelty of our scheme is to derive and solve a new correction equation satisfied by the vector potential. The vector potential $\widehat{\boldsymbol{T}}$ only needs to be solved approximately as in the original JD method. An approximate corrector for (3.3) is then obtained by taking $\boldsymbol{T} := \varepsilon_{r,h}^{-1}\nabla_h \times^* \mu_{r,h}^{-1}\widehat{\boldsymbol{T}}$.

3.3.1. Vector potential approach for the correction equation (step (iii)). We first explain how to derive a new correction equation for the vector potential. This procedure can be illustrated, for example, by the evaluation of $\mathbb{M}^{-1}\mathbb{B}\boldsymbol{u}_k$ in (3.7), or equivalently by getting an approximate solution of

$$(\nabla_h \times^* \mu_{r,h}^{-1}\nabla_h \times -\theta_k\varepsilon_{r,h})\boldsymbol{Q} = \varepsilon_{r,h}\boldsymbol{U}_k.\tag{3.11}$$

By construction, $\boldsymbol{U}_k \in \ker(\nabla_h \times)^{\perp_{\varepsilon_{r,h}}}$. Therefore from Theorem 2.2, we have

$$\varepsilon_{r,h}\boldsymbol{U}_k = \nabla_h \times^* \mu_{r,h}^{-1}\widehat{\boldsymbol{U}}_k. \tag{3.12}$$

We seek a solution to (3.11) of the form

$$\varepsilon_{r,h}\boldsymbol{Q} = \nabla_h \times^* \mu_{r,h}^{-1}\widehat{\boldsymbol{Q}}. \tag{3.13}$$

Substituting (3.12), (3.13) into (3.11), we have

$$\nabla_h \times^* \big(\mu_{r,h}^{-1}\nabla_h \times \varepsilon_{r,h}^{-1}\nabla_h \times^* \mu_{r,h}^{-1} - \theta_k\mu_{r,h}^{-1}\big)\widehat{\boldsymbol{Q}} = \nabla_h \times^* \mu_{r,h}^{-1}\widehat{\boldsymbol{U}}_k. \tag{3.14}$$

Instead of solving (3.11) directly, we propose to solve for the vector potential

$$\big(\mu_{r,h}^{-1}\nabla_h \times \varepsilon_{r,h}^{-1}\nabla_h \times^* \mu_{r,h}^{-1} - \theta_k\mu_{r,h}^{-1}\big)\widehat{\boldsymbol{Q}} = \mu_{r,h}^{-1}\widehat{\boldsymbol{U}}_k, \tag{3.15}$$

or equivalently

$$\big(\nabla_h \times \varepsilon_{r,h}^{-1}\nabla_h \times^* - \theta_k\mu_{r,h}\big)(\mu_{r,h}^{-1}\widehat{\boldsymbol{Q}}) = \widehat{\boldsymbol{U}}_k, \tag{3.16}$$

then take $\boldsymbol{Q} = \varepsilon_{r,h}^{-1}\nabla_h \times^* \mu_{r,h}^{-1}\widehat{\boldsymbol{Q}}$ as an approximate solution of (3.11) which lies in $\ker(\nabla_h\times)^{\perp_{\varepsilon_{r,h}}}$ automatically.

The same idea can be used to derive an equation for the vector potential of the corrector. We now elaborate the procedure in matrix notation. From Theorem 2.2, we have

$$\boldsymbol{u}_k = \mathbb{B}^{-1}\mathbb{C}^*\widehat{\mathbb{B}}\widehat{\boldsymbol{u}}_k, \quad \boldsymbol{t} = \mathbb{B}^{-1}\mathbb{C}^*\widehat{\mathbb{B}}\widehat{\boldsymbol{t}}. \tag{3.17}$$

We now substitute (3.17) into the correction equation

$$(\mathbb{I} - \mathbb{B}\boldsymbol{u}_k\boldsymbol{u}_k^*)(\mathbb{A} - \theta_k\mathbb{B})(\mathbb{I} - \boldsymbol{u}_k\boldsymbol{u}_k^*\mathbb{B})\boldsymbol{t} = -\boldsymbol{r}_k, \quad \boldsymbol{t} \perp_{\mathbb{B}} \boldsymbol{u}_k. \tag{3.18}$$

The left-hand side of (3.18) becomes

$$\begin{aligned}
&(\mathbb{I} - \mathbb{B}\boldsymbol{u}_k\boldsymbol{u}_k^*)(\mathbb{A} - \theta_k\mathbb{B})(\mathbb{I} - \boldsymbol{u}_k\boldsymbol{u}_k^*\mathbb{B})\boldsymbol{t}\\
&\quad = (\mathbb{I} - \mathbb{C}^*\widehat{\mathbb{B}}\widehat{\boldsymbol{u}}_k\widehat{\boldsymbol{u}}_k^*\widehat{\mathbb{B}}\mathbb{C}\mathbb{B}^{-1})(\mathbb{C}^*\widehat{\mathbb{B}}\mathbb{C} - \theta_k\mathbb{B})\mathbb{B}^{-1}\mathbb{C}^*\widehat{\mathbb{B}}(\mathbb{I} - \widehat{\boldsymbol{u}}_k\widehat{\boldsymbol{u}}_k^*\widehat{\mathbb{B}}\mathbb{C}\mathbb{B}^{-1}\mathbb{C}^*\widehat{\mathbb{B}})\widehat{\boldsymbol{t}}\\
&\quad = (\mathbb{I} - \mathbb{C}^*\widehat{\mathbb{B}}\widehat{\boldsymbol{u}}_k\widehat{\boldsymbol{u}}_k^*\widehat{\mathbb{B}}\mathbb{C}\mathbb{B}^{-1})\mathbb{C}^*(\widehat{\mathbb{B}}\mathbb{C}\mathbb{B}^{-1}\mathbb{C}^*\widehat{\mathbb{B}} - \theta_k\widehat{\mathbb{B}})(\mathbb{I} - \widehat{\boldsymbol{u}}_k\widehat{\boldsymbol{u}}_k^*\widehat{\mathbb{A}})\widehat{\boldsymbol{t}}\\
&\quad = \mathbb{C}^*(\mathbb{I} - \widehat{\mathbb{B}}\widehat{\boldsymbol{u}}_k\widehat{\boldsymbol{u}}_k^*\widehat{\mathbb{A}}\widehat{\mathbb{B}}^{-1})(\widehat{\mathbb{A}} - \theta_k\widehat{\mathbb{B}})(\mathbb{I} - \widehat{\boldsymbol{u}}_k\widehat{\boldsymbol{u}}_k^*\widehat{\mathbb{A}})\widehat{\boldsymbol{t}},
\end{aligned}$$

where $\widehat{\mathbb{A}} = \widehat{\mathbb{B}}\mathbb{C}\mathbb{B}^{-1}\mathbb{C}^*\widehat{\mathbb{B}}$. The right-hand side of (3.18) reduces to

$$\boldsymbol{r}_k = (\mathbb{A} - \theta_k\mathbb{B})\boldsymbol{u}_k = (\mathbb{C}^*\widehat{\mathbb{B}}\mathbb{C}\mathbb{B}^{-1}\mathbb{C}^*\widehat{\mathbb{B}} - \theta_k\mathbb{C}^*\widehat{\mathbb{B}})\widehat{\boldsymbol{u}}_k = \mathbb{C}^*(\widehat{\mathbb{A}} - \theta_k\widehat{\mathbb{B}})\widehat{\boldsymbol{u}}_k := \mathbb{C}^*\widehat{\boldsymbol{r}}_k, \tag{3.19}$$

which implies that

$$\mathbb{C}^*\big((\mathbb{I} - \widehat{\mathbb{B}}\widehat{\boldsymbol{u}}_k\widehat{\boldsymbol{u}}_k^*\widehat{\mathbb{A}}\widehat{\mathbb{B}}^{-1})(\widehat{\mathbb{A}} - \theta_k\widehat{\mathbb{B}})(\mathbb{I} - \widehat{\boldsymbol{u}}_k\widehat{\boldsymbol{u}}_k^*\widehat{\mathbb{A}})\widehat{\boldsymbol{t}} + \widehat{\boldsymbol{r}}_k\big) = \boldsymbol{0}, \tag{3.20}$$

where

$$\widehat{\boldsymbol{r}}_k = (\widehat{\mathbb{A}} - \theta_k\widehat{\mathbb{B}})\widehat{\boldsymbol{u}}_k. \tag{3.21}$$

It suffices to solve

$$(\mathbb{I} - \widehat{\mathbb{B}}\widehat{\boldsymbol{u}}_k\widehat{\boldsymbol{u}}_k^*\widehat{\mathbb{A}}\widehat{\mathbb{B}}^{-1})(\widehat{\mathbb{A}} - \theta_k\widehat{\mathbb{B}})(\mathbb{I} - \widehat{\boldsymbol{u}}_k\widehat{\boldsymbol{u}}_k^*\widehat{\mathbb{A}})\widehat{\boldsymbol{t}} = -\widehat{\boldsymbol{r}}_k \tag{3.22}$$

under the original constraint $\boldsymbol{t} \perp_{\mathbb{B}} \boldsymbol{u}_k$. From (3.17), it follows that

$$\boldsymbol{t}^*\mathbb{B}\boldsymbol{u}_k = \widehat{\boldsymbol{t}}^*\widehat{\mathbb{B}}\mathbb{C}\mathbb{B}^{-1}\mathbb{B}\mathbb{B}^{-1}\mathbb{C}^*\widehat{\mathbb{B}}\ \widehat{\boldsymbol{u}}_k = \widehat{\boldsymbol{t}}^*\widehat{\mathbb{A}}\widehat{\boldsymbol{u}}_k. \tag{3.23}$$

Thus the condition $\boldsymbol{t} \perp_{\mathbb{B}} \boldsymbol{u}_k$ now translates to $\widehat{\boldsymbol{t}} \perp_{\widehat{\mathbb{A}}} \widehat{\boldsymbol{u}}_k$, and the correction equation for the vector potential is given by

$$(\mathbb{I} - \widehat{\mathbb{B}}\widehat{\boldsymbol{u}}_k\widehat{\boldsymbol{u}}_k^*\widehat{\mathbb{A}}\widehat{\mathbb{B}}^{-1})(\widehat{\mathbb{A}} - \theta_k\widehat{\mathbb{B}})(\mathbb{I} - \widehat{\boldsymbol{u}}_k\widehat{\boldsymbol{u}}_k^*\widehat{\mathbb{A}})\widehat{\boldsymbol{t}} = -\widehat{\boldsymbol{r}}_k, \quad \widehat{\boldsymbol{t}} \perp_{\widehat{\mathbb{A}}} \widehat{\boldsymbol{u}}_k. \tag{3.24}$$

The solution procedure for (3.24) remains the same. One solves it iteratively with a preconditioner similar to the one given in (3.5):

$$\widehat{\mathbb{M}}_p := (\mathbb{I} - \widehat{\mathbb{B}}\widehat{\boldsymbol{u}}_k\widehat{\boldsymbol{u}}_k^*\widehat{\mathbb{A}}\widehat{\mathbb{B}}^{-1})\widehat{\mathbb{M}}(\mathbb{I} - \widehat{\boldsymbol{u}}_k\widehat{\boldsymbol{u}}_k^*\widehat{\mathbb{A}}), \tag{3.25}$$

where $\widehat{\mathbb{M}}$ is a preconditioner for $(\widehat{\mathbb{A}} - \theta_k\widehat{\mathbb{B}})$.

In jth GMRES iteration, one solves for

$$\widehat{\mathbb{M}}_p\widehat{\boldsymbol{z}}^{(j)} = \widehat{\boldsymbol{y}}^{(j)}, \quad \widehat{\boldsymbol{u}}_k^*\widehat{\mathbb{A}}\widehat{\boldsymbol{z}}^{(j)} = 0. \tag{3.26}$$

A solution to (3.26) is therefore given by

$$\widehat{\boldsymbol{z}}^{(j)} = \widehat{\mathbb{M}}^{-1}\widehat{\boldsymbol{y}}^{(j)} - \hat{\zeta}^{(j)}\widehat{\mathbb{M}}^{-1}\widehat{\mathbb{B}}\widehat{\boldsymbol{u}}_k, \quad \text{where} \quad \hat{\zeta}^{(j)} = \frac{\widehat{\boldsymbol{u}}_k^*\widehat{\mathbb{A}}\widehat{\mathbb{M}}^{-1}\widehat{\boldsymbol{y}}^{(j)}}{\widehat{\boldsymbol{u}}_k^*\widehat{\mathbb{A}}\widehat{\mathbb{M}}^{-1}\widehat{\mathbb{B}}\widehat{\boldsymbol{u}}_k}. \tag{3.27}$$

Note that if $\widehat{\boldsymbol{t}}$ is an approximate solution of (3.24), then the transformation (3.17) gives rise to an approximate solution of (3.18) which lies in $\ker(\nabla_h\times)^{\perp_{\varepsilon_{r,h}}}$ automatically. The only difference between NFJD and HPJD is how an approximate solution to (3.18) (that lies in $\ker(\nabla_h\times)^{\perp_{\varepsilon_{r,h}}}$) is obtained. A more detailed comparison between NFJD and HPJD can be found in section 4.

For the purpose of implementation, it is more convenient to work with the vector potential variables. We now express the rest of the steps in Algorithm 1 in terms of the vector potential. The $\mathbb{B}$-orthonormalization for the corrector in step (iii) of Algorithm 1,

$$\boldsymbol{v}_{k+1} = \frac{\boldsymbol{t} - \sum_{\ell=1}^{k}(\boldsymbol{v}_\ell^*\mathbb{B}\boldsymbol{t})\boldsymbol{v}_\ell}{\|\boldsymbol{t} - \sum_{\ell=1}^{k}(\boldsymbol{v}_\ell^*\mathbb{B}\boldsymbol{t})\boldsymbol{v}_\ell\|_{\mathbb{B}}}, \tag{3.28}$$

can now be recast to

$$\mathbb{B}^{-1}\mathbb{C}^*\widehat{\mathbb{B}}\ \widehat{\boldsymbol{v}}_{k+1} = \frac{\mathbb{B}^{-1}\mathbb{C}^*\widehat{\mathbb{B}}\big(\widehat{\boldsymbol{t}} - \sum_{\ell=1}^{k}(\widehat{\boldsymbol{v}}_\ell^*\widehat{\mathbb{A}}\widehat{\boldsymbol{t}})\widehat{\boldsymbol{v}}_\ell\big)}{\|\boldsymbol{t} - \sum_{\ell=1}^{k}(\boldsymbol{v}_\ell^*\mathbb{B}\boldsymbol{t})\boldsymbol{v}_\ell\|_{\mathbb{B}}}. \tag{3.29}$$

We can therefore express the orthogonalization procedure in terms of vector potentials and $\widehat{\mathbb{A}}$ as

$$\widehat{\boldsymbol{v}}_{k+1} = \frac{\widehat{\boldsymbol{t}} - \sum_{\ell=1}^{k}(\widehat{\boldsymbol{v}}_\ell^*\widehat{\mathbb{A}}\widehat{\boldsymbol{t}})\widehat{\boldsymbol{v}}_\ell}{\|\boldsymbol{t} - \sum_{\ell=1}^{k}(\boldsymbol{v}_\ell^*\mathbb{B}\boldsymbol{t})\boldsymbol{v}_\ell\|_{\mathbb{B}}} = \frac{\widehat{\boldsymbol{t}} - \sum_{\ell=1}^{k}(\widehat{\boldsymbol{v}}_\ell^*\widehat{\mathbb{A}}\widehat{\boldsymbol{t}})\widehat{\boldsymbol{v}}_\ell}{\big|\widehat{\boldsymbol{t}} - \sum_{\ell=1}^{k}(\widehat{\boldsymbol{v}}_\ell^*\widehat{\mathbb{A}}\widehat{\boldsymbol{t}})\widehat{\boldsymbol{v}}_\ell\big|_{\widehat{\mathbb{A}}}}, \tag{3.30}$$

where we have adopted the notation

$$|\widehat{\boldsymbol{z}}|_{\widehat{\mathbb{A}}} := (\widehat{\boldsymbol{z}}^*\widehat{\mathbb{A}}\widehat{\boldsymbol{z}})^{\frac{1}{2}} = (\widehat{\boldsymbol{z}}^*\widehat{\mathbb{B}}\mathbb{C}\mathbb{B}^{-1}\mathbb{C}^*\widehat{\mathbb{B}}\widehat{\boldsymbol{z}})^{\frac{1}{2}} = \|\mathbb{B}^{-1}\mathbb{C}^*\widehat{\mathbb{B}}\widehat{\boldsymbol{z}}\|_{\mathbb{B}}. \tag{3.31}$$

3.3.2. Isospectral reformulation and the subspace eigenvalue problem (step (i)). It remains to formulate the subspace eigenvalue problem in terms of vector potentials. This can be done by retaining step (i) in Algorithm 1 and substituting $\boldsymbol{v} = \mathbb{B}^{-1}\mathbb{C}^*\widehat{\mathbb{B}}\widehat{\boldsymbol{v}}$ to get

$$\begin{aligned}&(\mathbb{W}_k - \theta_k\mathbb{I})\boldsymbol{s}_k = \boldsymbol{0},\\ &(\mathbb{W}_k)_{\ell m} = \boldsymbol{v}_\ell^*\mathbb{A}\boldsymbol{v}_m = \widehat{\boldsymbol{v}}_\ell^*\widehat{\mathbb{B}}\mathbb{C}\mathbb{B}^{-1}\mathbb{C}^*\widehat{\mathbb{B}}\mathbb{C}\mathbb{B}^{-1}\mathbb{C}^*\widehat{\mathbb{B}}\widehat{\boldsymbol{v}}_m = \widehat{\boldsymbol{v}}_\ell^*\widehat{\mathbb{A}}\widehat{\mathbb{B}}^{-1}\widehat{\mathbb{A}}\widehat{\boldsymbol{v}}_m;\ \ \widehat{\boldsymbol{u}}_k = \widehat{\mathbb{V}}_k\boldsymbol{s}_k.\end{aligned} \tag{3.32}$$

The subspace eigenvalue problem (3.32) amounts to the following:

(SEP-1) Find $\widehat{\boldsymbol{u}}_k \in \mathrm{Range}(\widehat{\mathbb{V}}_k)$, $\theta_k > 0$, nearest to the target such that

$$(\mathbb{B}^{-1}\mathbb{C}^*\widehat{\mathbb{B}}\widehat{\boldsymbol{v}})^*(\mathbb{A} - \theta_k\mathbb{B})(\mathbb{B}^{-1}\mathbb{C}^*\widehat{\mathbb{B}}\widehat{\boldsymbol{u}}_k) = 0 \text{ for all } \widehat{\boldsymbol{v}} \in \mathrm{Range}(\widehat{\mathbb{V}}_k).$$

This is essentially identical to standard approximation of the original pencil $(\mathbb{A}, \mathbb{B})$ on the subspace $\mathbb{V}_k$, except the subspace now takes the particular form $\mathbb{V}_k = \mathbb{B}^{-1}\mathbb{C}^*\widehat{\mathbb{B}}\widehat{\mathbb{V}}_k$ and remains perpendicular to the null space.

There is an alternative approach to formulate the subspace eigenvalue problem in terms of the vector potential. Recall from (3.19) and (3.21) that $(\theta_k, \mathbb{B}^{-1}\mathbb{C}\widehat{\mathbb{B}}\widehat{\boldsymbol{u}}_k)$ is an approximate eigenpair of the pencil $(\mathbb{A}, \mathbb{B})$ with residual $\boldsymbol{r}_k = (\mathbb{A} - \theta_k\mathbb{B})\boldsymbol{u}_k$, provided that $(\theta_k, \widehat{\boldsymbol{u}}_k)$ is an approximate eigenpair of the pencil $(\widehat{\mathbb{A}}, \widehat{\mathbb{B}})$ with residual $\widehat{\boldsymbol{r}}_k = (\widehat{\mathbb{A}} - \theta_k\widehat{\mathbb{B}})\widehat{\boldsymbol{u}}_k$ and $\boldsymbol{r}_k = \mathbb{C}^*\widehat{\boldsymbol{r}}_k$.

In fact, it is not difficult to verify that the two pencils $(\mathbb{A}, \mathbb{B}) = (\mathbb{C}^*\widehat{\mathbb{B}}\mathbb{C}, \mathbb{B})$ and $(\widehat{\mathbb{A}}, \widehat{\mathbb{B}}) = (\widehat{\mathbb{B}}\mathbb{C}\mathbb{B}^{-1}\mathbb{C}^*\widehat{\mathbb{B}}, \widehat{\mathbb{B}})$ have identical spectrum. This is not surprising since the eigenvalue problem $\widehat{\mathbb{A}}\widehat{\boldsymbol{e}} = \lambda\widehat{\mathbb{B}}\widehat{\boldsymbol{e}}$, or equivalently

$$\mathbb{C}\mathbb{B}^{-1}\mathbb{C}^*\boldsymbol{h} = \lambda\widehat{\mathbb{B}}^{-1}\boldsymbol{h}, \tag{3.33}$$

is nothing but the matrix representation of the Maxwell's equations (1.1)–(1.4) written in terms of the magnetic field $\boldsymbol{H}$:

$$\nabla_h \times \varepsilon_{r,h}^{-1}\nabla_h \times^* \boldsymbol{H} = \lambda\mu_{r,h}\boldsymbol{H}. \tag{3.34}$$

In view of this, we now have another formulation for the subspace eigenvalue problem:

(SEP-2) Find $\widehat{\boldsymbol{u}}_k \in \mathrm{Range}(\widehat{\mathbb{V}}_k)$, $\theta_k > 0$ nearest to the target such that

$$\widehat{\boldsymbol{v}}^*(\widehat{\mathbb{A}} - \theta_k\widehat{\mathbb{B}})\widehat{\boldsymbol{u}}_k = 0, \text{ for all } \widehat{\boldsymbol{v}} \in \mathrm{Range}(\widehat{\mathbb{V}}_k).$$

Or, in matrix notation (recall the normalization $\widehat{\boldsymbol{v}}_l^*\widehat{\mathbb{A}}\widehat{\boldsymbol{v}}_m = \delta_{lm}$ in step (iii)),

$$(\mathbb{I} - \theta_k\widehat{\mathbb{Z}}_k)\boldsymbol{s}_k = \boldsymbol{0}, \quad (\widehat{\mathbb{Z}}_k)_{\ell m} = \widehat{\boldsymbol{v}}_\ell^*\widehat{\mathbb{B}}\widehat{\boldsymbol{v}}_m; \quad \widehat{\boldsymbol{u}}_k = \widehat{\mathbb{V}}_k\boldsymbol{s}_k. \tag{3.35}$$

Even though the pencils $(\mathbb{A}, \mathbb{B})$ and $(\widehat{\mathbb{A}}, \widehat{\mathbb{B}})$ are isospectral, their subspace approximations (SEP-1) and (SEP-2) are generally different and correspond to the pencils $(\widehat{\mathbb{V}}_k^*\widehat{\mathbb{A}}\widehat{\mathbb{B}}^{-1}\widehat{\mathbb{A}}\widehat{\mathbb{V}}_k, \widehat{\mathbb{V}}_k^*\widehat{\mathbb{A}}\widehat{\mathbb{V}}_k)$ and $(\widehat{\mathbb{V}}_k^*\widehat{\mathbb{A}}\widehat{\mathbb{V}}_k, \widehat{\mathbb{V}}_k^*\widehat{\mathbb{B}}\widehat{\mathbb{V}}_k)$, respectively. The $\widehat{\boldsymbol{u}}_k$ selected from (SEP-1) or (SEP-2) is implicitly processed to get an approximate eigenpair $(\theta_k, \boldsymbol{u}_k) = (\theta_k, \mathbb{B}^{-1}\mathbb{C}^*\widehat{\mathbb{B}}\widehat{\boldsymbol{u}}_k)$ for the pencil $(\mathbb{A}, \mathbb{B})$. Similar to $\mathbb{A}$, the matrix $\widehat{\mathbb{A}}$ also possesses a huge null space consisting of those $\widehat{\boldsymbol{v}}$ such that $\mathbb{C}^*\widehat{\mathbb{B}}\widehat{\boldsymbol{v}} = \boldsymbol{0}$. When the selected $\widehat{\boldsymbol{u}}_k$ lies very close to $\ker(\widehat{\mathbb{A}})$, the corresponding $(\theta_k, \boldsymbol{u}_k)$ is no longer a good approximate eigenpair for $(\mathbb{A}, \mathbb{B})$. In this situation, $\theta_k \approx 0$ and the dominant part of $\widehat{\boldsymbol{u}}_k$ (the $\ker(\widehat{\mathbb{A}})$ component) is completely wiped out upon multiplication by $\mathbb{B}^{-1}\mathbb{C}^*\widehat{\mathbb{B}}$. The remaining components in $\ker(\widehat{\mathbb{A}})^{\perp_{\widehat{\mathbb{B}}}}$ are therefore amplified to get an essentially random vector $\boldsymbol{u}_k = \mathbb{B}^{-1}\mathbb{C}^*\widehat{\mathbb{B}}\widehat{\boldsymbol{u}}_k$ in $\ker(\mathbb{A})^{\perp_{\mathbb{B}}}$. To prevent this from happening, we have set up a threshold in the selection of the nearest-to-target approximate eigenvalue for (SEP-2). More precisely, we select the smallest θ such that $\theta \geq \theta_{\rm c} > 0$ in (SEP-2). The threshold value $\theta_{\rm c}$ and other parameters are detailed in section 4. On the other hand, a threshold is not needed for (SEP-1) since it also corresponds to a subspace eigenvalue problem for the pencil $(\widehat{\mathbb{A}}\widehat{\mathbb{B}}^{-1}\widehat{\mathbb{A}}, \widehat{\mathbb{A}})$. The above mentioned scenario ($\theta_k \approx 0$, with $\widehat{\boldsymbol{u}}_k$ very close to $\ker(\mathbb{A})$) does not occur. We simply choose the smallest $\theta > 0$ for (SEP-1).

In view of (3.32) and (3.35), it is obvious that the magnetic field approach (SEP-2) requires fewer arithmetic operations in forming the subspace matrix ($\mathbb{W}_k$ vs $\widehat{\mathbb{Z}}_k$). We have implemented both versions and found that (SEP-2) with the threshold indeed prevails under otherwise identical settings and they both outperform HPJD.

Note, however, that setting up a threshold does not help in the original JD method (that is, without Helmholtz projection). In general, JD with the same threshold performs worse than the original JD and frequently fails to converge.

We summarize the NFJD method using the magnetic field subspace eigenvalue problem (SEP-2) as Algorithm 2. A detailed comparison between HPJD and NFJD will be given in section 4.

3.3.3. Stopping tolerance, error bounds, and variant of NFJD. In NFJD, even though the actual working variables are the vector potentials (or magnetic field vectors), the underlying eigenvalue problem remains the original one for the electric field, namely, $\mathbb{A}\boldsymbol{e} = \lambda\mathbb{B}\boldsymbol{e}$. An error bound for the computed eigenpair follows from the standard estimate [27]:

$$\begin{aligned} |\sin\angle_{\mathbb{B}}(\boldsymbol{e}_i, \boldsymbol{e}_i^{\rm NFJD})| &\leq \frac{1}{\gamma_i}\frac{\|\boldsymbol{r}^{\rm NFJD}\|_{\mathbb{B}^{-1}}}{\|\boldsymbol{e}_i^{\rm NFJD}\|_{\mathbb{B}}} \leq \frac{1}{\gamma_i}\frac{\|\mathbb{B}^{-1}\|_2\|\boldsymbol{r}^{\rm NFJD}\|_2}{\|\boldsymbol{e}_i^{\rm NFJD}\|_{\mathbb{B}}}, \\ |\lambda_i - \lambda_i^{\rm NFJD}| &\leq \min\left(\frac{\|\boldsymbol{r}^{\rm NFJD}\|_{\mathbb{B}^{-1}}}{\|\boldsymbol{e}_i^{\rm NFJD}\|_{\mathbb{B}}}, \frac{1}{\gamma_i}\left(\frac{\|\boldsymbol{r}^{\rm NFJD}\|_{\mathbb{B}^{-1}}}{\|\boldsymbol{e}_i^{\rm NFJD}\|_{\mathbb{B}}}\right)^2\right) \\ &\leq \min\left(\frac{\|\mathbb{B}^{-1}\|_2\|\boldsymbol{r}^{\rm NFJD}\|_2}{\|\boldsymbol{e}_i^{\rm NFJD}\|_{\mathbb{B}}}, \frac{1}{\gamma_i}\left(\frac{\|\mathbb{B}^{-1}\|_2\|\boldsymbol{r}^{\rm NFJD}\|_2}{\|\boldsymbol{e}_i^{\rm NFJD}\|_{\mathbb{B}}}\right)^2\right), \end{aligned} \tag{3.36}$$

where

$$|\sin\angle_{\mathbb{B}}(\boldsymbol{u}, \boldsymbol{v})| = \sqrt{1 - \left(\frac{\boldsymbol{u}^*\mathbb{B}\boldsymbol{v}}{\|\boldsymbol{u}\|_{\mathbb{B}}\|\boldsymbol{v}\|_{\mathbb{B}}}\right)^2}, \qquad \gamma_i = \min_{\lambda_j \neq \lambda_i}|\lambda_j - \lambda_i^{\rm NFJD}|. \tag{3.37}$$

Since $\boldsymbol{r}^{\rm NFJD} = \mathbb{C}^*\widehat{\boldsymbol{r}}$, where $\boldsymbol{r}^{\rm NFJD} = (\mathbb{A} - \theta\mathbb{B})\boldsymbol{u}$, $\widehat{\boldsymbol{r}} = (\widehat{\mathbb{A}} - \theta\widehat{\mathbb{B}})\widehat{\boldsymbol{u}}$, the stopping criterion $\|\widehat{\boldsymbol{r}}\|_2 < \tau_{\rm NFJD}$ implies

$$\|\boldsymbol{r}^{\rm NFJD}\|_2 = \|\mathbb{C}^*\widehat{\boldsymbol{r}}\| \leq \|\mathbb{C}^*\|_2\|\widehat{\boldsymbol{r}}\|_2 < \|\mathbb{C}^*\|_2\,\tau_{\rm NFJD} \tag{3.38}$$

with

$$\|\mathbb{C}^*\|_2 \leq \sqrt{\|\mathbb{C}^*\|_1\|\mathbb{C}^*\|_\infty} = 2\max\left\{\frac{1}{h_1} + \frac{1}{h_2}, \frac{1}{h_2} + \frac{1}{h_3}, \frac{1}{h_3} + \frac{1}{h_1}\right\}. \tag{3.39}$$

ALGORITHM 2. NFJD METHOD. Primary working variables (hatted vectors) are in $\mathcal{V}_{\mathcal{F}}$ and output eigenvectors are in $\mathcal{V}_{\mathcal{E}}$.

```
Set $\widehat{\mathbb{E}}_0 = [\ ]$, $\Lambda_0 = \varnothing$.
for $i = 1, 2, 3, \ldots, i_{\max}$ do
    Initialize a vector potential $\widehat{\boldsymbol{v}}_1$ with $|\widehat{\boldsymbol{v}}_1|_{\widehat{\mathbb{A}}} = 1$ and $\widehat{\mathbb{E}}_{i-1}^*\widehat{\mathbb{A}}\widehat{\boldsymbol{v}}_1 = \mathbf{0}$.
    Set $\widehat{\mathbb{V}}_1 = [\widehat{\mathbb{E}}_{i-1}, \widehat{\boldsymbol{v}}_1]$.
    Compute $\widehat{\mathbb{Z}}_1 = \widehat{\mathbb{V}}_1^*\widehat{\mathbb{B}}\widehat{\mathbb{V}}_1$.
    for $k = 1, 2, 3, \ldots$ do
      (i) Compute all the eigenpairs of $(\mathbb{I} - \theta\widehat{\mathbb{Z}}_k)\boldsymbol{s} = \mathbf{0}$.
          Select the desired eigenpair $(\theta_k, \boldsymbol{s}_k)$ with $\theta_k \notin \Lambda_{i-1}$ nearest to the
          target and $\|\boldsymbol{s}_k\|_2 = 1$.
      (ii) Compute $\widehat{\boldsymbol{u}}_k = \widehat{\mathbb{V}}_k\boldsymbol{s}_k$, $\widehat{\boldsymbol{r}}_k = (\widehat{\mathbb{A}} - \theta_k\widehat{\mathbb{B}})\widehat{\boldsymbol{u}}_k$.
      (iii) if $(\|\widehat{\boldsymbol{r}}_k\|_2 < \tau_{\mathrm{NFJD}})$ then
                Output $\lambda_i = \theta_k$ and $\boldsymbol{e}_i = \mathbb{B}^{-1}\mathbb{C}^*\widehat{\mathbb{B}}\widehat{\boldsymbol{u}}_k$.
                Update $\widehat{\mathbb{E}}_i = [\widehat{\mathbb{E}}_{i-1}, \widehat{\boldsymbol{u}}_k]$, $\Lambda_i = \Lambda_{i-1} \cup \{\lambda_i\}$.
                Exit $k$.
            else
                Solve (approximately)
                  $(\mathbb{I} - \widehat{\mathbb{B}}\widehat{\boldsymbol{u}}_k\widehat{\boldsymbol{u}}_k^*\widehat{\mathbb{A}}\widehat{\mathbb{B}}^{-1})(\widehat{\mathbb{A}} - \theta_k\widehat{\mathbb{B}})(\mathbb{I} - \widehat{\boldsymbol{u}}_k\widehat{\boldsymbol{u}}_k^*\widehat{\mathbb{A}})\widehat{\boldsymbol{t}} = -\widehat{\boldsymbol{r}}_k, \quad \widehat{\boldsymbol{t}} \perp_{\widehat{\mathbb{A}}} \widehat{\boldsymbol{u}}_k$.
                $\widehat{\mathbb{A}}$-orthonormalize $\widehat{\boldsymbol{t}}$ against $\widehat{\mathbb{V}}_k$: $\widehat{\boldsymbol{v}}_{k+1} = \frac{\widehat{\boldsymbol{t}} - \sum_{\ell=1}^{k}(\widehat{\boldsymbol{v}}_\ell^*\widehat{\mathbb{A}}\widehat{\boldsymbol{t}})\widehat{\boldsymbol{v}}_\ell}{|\widehat{\boldsymbol{t}} - \sum_{\ell=1}^{k}(\widehat{\boldsymbol{v}}_\ell^*\widehat{\mathbb{A}}\widehat{\boldsymbol{t}})\widehat{\boldsymbol{v}}_\ell|_{\widehat{\mathbb{A}}}}$.
                Expand $\widehat{\mathbb{V}}_{k+1} = [\widehat{\mathbb{V}}_k, \widehat{\boldsymbol{v}}_{k+1}]$, $\widehat{\mathbb{Z}}_{k+1} = \begin{bmatrix} \widehat{\mathbb{Z}}_k & \widehat{\mathbb{V}}_k^*\widehat{\mathbb{B}}\widehat{\boldsymbol{v}}_{k+1} \\ \widehat{\boldsymbol{v}}_{k+1}^*\widehat{\mathbb{B}}\widehat{\mathbb{V}}_k & \widehat{\boldsymbol{v}}_{k+1}^*\widehat{\mathbb{B}}\widehat{\boldsymbol{v}}_{k+1} \end{bmatrix}$.
            end if
    end for $k$
end for $i$
```

Upon convergence, an approximate eigenvector $\boldsymbol{e}$ is obtained by taking $\boldsymbol{e}^{\mathrm{NFJD}} = \boldsymbol{u} = \mathbb{B}^{-1}\mathbb{C}^*\widehat{\mathbb{B}}\widehat{\boldsymbol{u}}$, which is $\mathbb{B}$-normalized as in standard JD and HPJD computation:

$$\|\boldsymbol{e}^{\mathrm{NFJD}}\|_{\mathbb{B}} = \|\boldsymbol{u}\|_{\mathbb{B}} = \|\mathbb{B}^{-1}\mathbb{C}^*\widehat{\mathbb{B}}\widehat{\boldsymbol{u}}\|_{\mathbb{B}} = |\widehat{\boldsymbol{u}}|_{\widehat{\mathbb{A}}} = 1. \tag{3.40}$$

An error bound like (3.36) applies equally to JD and HPJD. In view of (3.38) and (3.40), it follows that for NFJD to give comparable accuracy with JD or HPJD computation, it suffices to take $\tau_{\mathrm{JD}} = \|\mathbb{C}^*\|_2\tau_{\mathrm{NFJD}}$. This is the basis of our numerical comparison. See section 4 for details.

With the magnetic field interpretation for NFJD, an alternative null space free approach emerges naturally. First observe that a slight change in step (iii) of Algorithm 2, from "Output $\lambda_i = \theta_k$ and $\boldsymbol{e}_i = \mathbb{B}^{-1}\mathbb{C}^*\widehat{\mathbb{B}}\widehat{\boldsymbol{u}}_k$" to "Output $\lambda_i = \theta_k$ and $\boldsymbol{h}_i = \widehat{\mathbb{B}}\widehat{\boldsymbol{u}}_k$," results in a (null space free) numerical scheme for the magnetic field eigenvalue problem (3.33). Alternatively, one could start with the JD method for (3.33) instead and then apply a similar derivation as in NFJD. This approach is dual to NFJD with the roles of electric field vectors and magnetic field vectors interchanged during the iteration. In this new scheme, denoted as NFJD*, both the primary working variables and output eigenvectors are the electric field vectors. The magnetic field vectors (i.e., the vector potentials) only appear as auxiliary variables in the derivation. We omit the details and summarize the result in Algorithm 3.

Algorithm 3. NFJD*: dual version of JD method for Maxwell's equations. Primary working variables and output eigenvectors are both in $\mathcal{V}_{\mathcal{E}}$.

```
Set $\mathbb{E}_0 = [\,]$, $\Lambda_0 = \varnothing$.
for $i = 1, 2, 3, \ldots, i_{\max}$ do
    Initialize a vector $\boldsymbol{v}_1$ with $|\boldsymbol{v}_1|_{\mathbb{A}} = 1$ and $\mathbb{E}_{i-1}^* \mathbb{A} \boldsymbol{v}_1 = \mathbf{0}$.
    Set $\mathbb{V}_1 = [\mathbb{E}_{i-1}, \boldsymbol{v}_1]$.
    Compute $\mathbb{Z}_1 = \mathbb{V}_1^* \mathbb{B} \mathbb{V}_1$.
    for $k = 1, 2, 3, \ldots$ do
      (i) Compute all the eigenpairs of $(\mathbb{I} - \theta \mathbb{Z}_k)\boldsymbol{s} = \mathbf{0}$.
          Select the desired eigenpair $(\theta_k, \boldsymbol{s}_k)$ with $\theta_k \notin \Lambda_{i-1}$ nearest to the
          target and $\|\boldsymbol{s}_k\|_2 = 1$.
      (ii) Compute $\boldsymbol{u}_k = \dfrac{\mathbb{V}_k \boldsymbol{s}_k}{\|\mathbb{V}_k \boldsymbol{s}_k\|_{\mathbb{B}}}$, $\boldsymbol{r}_k = (\mathbb{A} - \theta_k \mathbb{B})\boldsymbol{u}_k$.
      (iii) if $\|\boldsymbol{r}_k\|_2 < \tau_{\mathrm{NFJD}^*}$ then
              Output $\lambda_i = \theta_k$, $\boldsymbol{e}_i = \boldsymbol{u}_k$.
              Update $\mathbb{E}_i = [\mathbb{E}_{i-1}, \boldsymbol{e}_i]$, $\Lambda_i = \Lambda_{i-1} \cup \{\lambda_i\}$.
              Exit $k$.
          else
              Solve (approximately)
                 $(\mathbb{I} - \mathbb{B}\boldsymbol{u}_k \boldsymbol{u}_k^* \mathbb{A}\mathbb{B}^{-1})(\mathbb{A} - \theta_k \mathbb{B})(\mathbb{I} - \boldsymbol{u}_k \boldsymbol{u}_k^* \mathbb{A})\boldsymbol{t} = -\boldsymbol{r}_k, \quad \boldsymbol{t} \perp_{\mathbb{A}} \boldsymbol{u}_k.$
              $\mathbb{A}$-orthonormalize $\boldsymbol{t}$ against $\mathbb{V}_k$: $\boldsymbol{v}_{k+1} = \dfrac{\boldsymbol{t} - \sum_{\ell=1}^{k} (\boldsymbol{v}_\ell^* \mathbb{A} \boldsymbol{t}) \boldsymbol{v}_\ell}{|\boldsymbol{t} - \sum_{\ell=1}^{k} (\boldsymbol{v}_\ell^* \mathbb{A} \boldsymbol{t}) \boldsymbol{v}_\ell|_{\mathbb{A}}}$.
              Expand $\mathbb{V}_{k+1} = [\mathbb{V}_k, \boldsymbol{v}_{k+1}]$, $\mathbb{Z}_{k+1} = \begin{bmatrix} \mathbb{Z}_k & \mathbb{V}_k^* \mathbb{B} \boldsymbol{v}_{k+1} \\ \boldsymbol{v}_{k+1}^* \mathbb{B} \mathbb{V}_k & \boldsymbol{v}_{k+1}^* \mathbb{B} \boldsymbol{v}_{k+1} \end{bmatrix}$.
          end if
    end for $k$
end for $i$
```

The major difference between NFJD and NFJD* lies in the routine matrix-vector multiplication. Namely, $\widehat{\mathbb{A}}\widehat{\boldsymbol{v}} = \widehat{\mathbb{B}}\mathbb{C}\mathbb{B}^{-1}\mathbb{C}^*\widehat{\mathbb{B}}\widehat{\boldsymbol{v}}$ in NFJD versus $\mathbb{A}\boldsymbol{v} = \mathbb{C}^*\widehat{\mathbb{B}}\mathbb{C}\boldsymbol{v}$ in NFJD*. For Yee's discretization, both $\mathbb{B}$ and $\widehat{\mathbb{B}}$ are diagonal matrices (with $\mu_r \equiv 1$, $\widehat{\mathbb{B}} = \mathbb{I}$ in most applications). The difference between NFJD and NFJD* is insignificant. In the finite element case, the mass matrices $\mathbb{B}$ and $\widehat{\mathbb{B}}$ are sparse and banded. In addition, multiplication by $\widehat{\mathbb{A}}$ in NFJD requires solving a linear system $\mathbb{B}\boldsymbol{v} = \boldsymbol{c}$ and is more expensive than NFJD*. In Appendix A, we will give a brief derivation for the finite element version of NFJD*.

4. Numerical tests.

Our numerical tests are based on the benchmark example shown in Figure 1, where the periodic dielectric structure within a primitive cubic cell is depicted. The structure consists of dielectric spheres with radius r connected by circular cylinders with radius s. Here $r/a = 0.345$, $s/a = 0.11$, and a is the edge length of the cube. Inside the structure is the dielectric material with permittivity contrast $\varepsilon_{r,\mathrm{i}}/\varepsilon_{r,\mathrm{o}} = 13$ and $\mu_{r,\mathrm{i}} = \mu_{r,\mathrm{o}} = 1$ (corresponding to $\widehat{\mathbb{B}} = \mathbb{I}$).

Figure 2 shows the plot of $w = a\sqrt{\lambda}/(2\pi)$ versus sample points $\mathbf{k}$ in the first Brillouin zone for the benchmark problem computed using NFJD with $N_1 = N_2 = N_3 = 100$. The smallest nonzero eigenvalues are calculated for 40 sample points $\mathbf{k}$ distributed along the segments connecting $\Gamma = (0, 0, 0)$, $X = (\pi, 0, 0)$, $M = (\pi, \pi, 0)$, $R = (\pi, \pi, \pi)$, and back to Γ in the first Brillouin zone. A clear band gap lies between the fifth and sixth smallest positive eigenvalues.

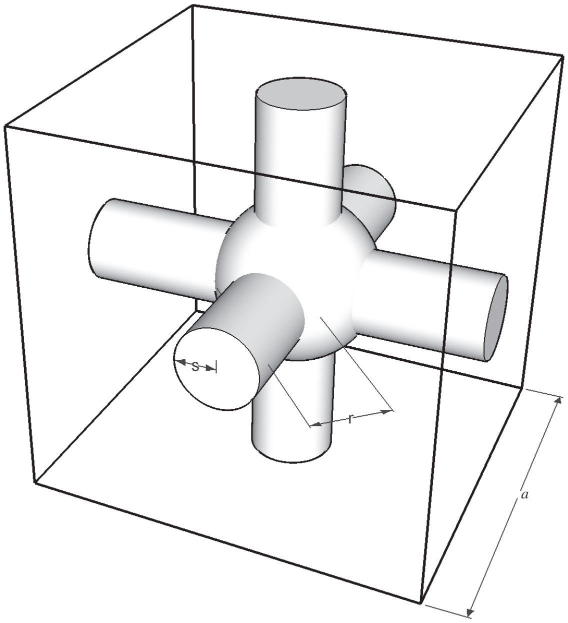

Fig. 1. *The periodic dielectric structure within a primitive cell. Inside: dielectric material. Outside: air. Here $r/a = 0.345$, $s/a = 0.11$, and $\varepsilon_{r,\mathrm{i}}/\varepsilon_{r,\mathrm{o}} = 13$.*

As a preliminary test, we summarize the result with various grid resolutions in Table 1. Here w_{low} denotes the maximum of the fifth eigenvalue, w_{up} the minimum of the sixth eigenvalue, and

$$\gamma_{\mathrm{gm}} := \frac{w_{\mathrm{up}} - w_{\mathrm{low}}}{(w_{\mathrm{up}} + w_{\mathrm{low}})/2}.$$

The result shows clear convergence and agrees well with those reported in the literature [6, 9].

To further illustrate the accuracy and efficiency of NFJD, we have devised several numerical experiments for JD, HPJD, and NFJD in various settings. In the following tests, all examples are computed using 100^3 cells with the initial vector obtained from interpolating the ground state of 50^3 calculation. All computations are conducted under identical settings for JD, HPJD, and NFJD, except the tolerances are scaled according to (3.38) and (3.39),

$$\tau_{\mathrm{NFJD}} := \tau_{\mathrm{JD}}/400. \tag{4.1}$$

The fast Fourier transform (FFT) is used as a preconditioner for (3.10), (3.18), and (3.24). More precisely, we take

$$\mathbb{M} = (\mathbb{C}^*\mathbb{C} - \theta\overline{\varepsilon_r}\mathbb{I}), \quad \widehat{\mathbb{M}} = (\overline{\varepsilon_r^{-1}}\mathbb{C}\mathbb{C}^* - \theta\mathbb{I}), \tag{4.2}$$

where $\overline{\varepsilon_r}$ and $\overline{\varepsilon_r^{-1}}$ are spatial averages of ε_r and ε_r^{-1}, respectively. Similarly, we use $\mathbb{G}^*\mathbb{G}$ as the preconditioner for the Poisson equation (3.10). The matrices $\mathbb{M}$, $\widehat{\mathbb{M}}$, and $\mathbb{G}^*\mathbb{G}$ are all FFT-invertible. Since both the correction equations and their preconditioners are highly indefinite (with about one third of eigenvalues being negative), we

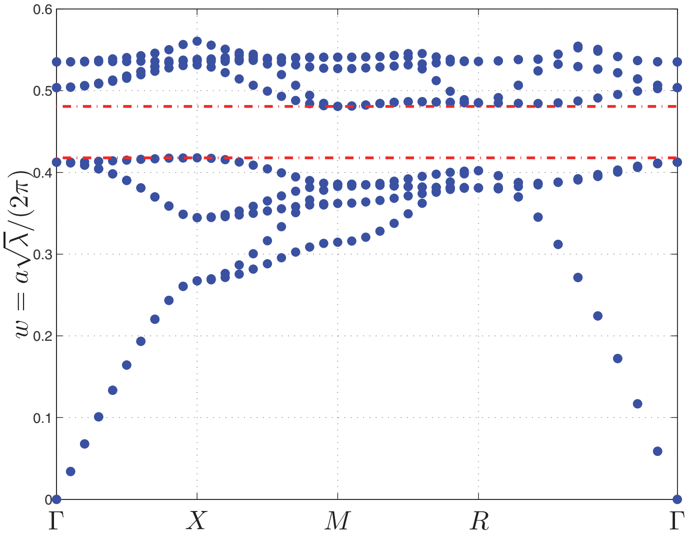

FIG. 2. *Band structure computed with* $100 \times 100 \times 100$ *grid.*

TABLE 1
The computed gap-midgap ratio with various grid sizes.

grids	$50 \times 50 \times 50$	$100 \times 100 \times 100$	$200 \times 200 \times 200$
w_{low}	0.41785	0.41789	0.41782
w_{up}	0.48023	0.48079	0.48096
γ_{gm}	0.1389	0.1400	0.1405

adopt GMRES as the linear solver for (3.18), (3.24), and PCG as the linear solver for (3.10), in conjunction with the preconditioners $\mathbb{M}$, $\widehat{\mathbb{M}}$, and $\mathbb{G}^*\mathbb{G}$. All numerical tests are performed on a PC equipped with an Intel Q9550 2.83-GHz processor and 16 GB main memory using Intel Fortran compiler version 11.1.

We start with investigating the effect of the null space on the original JD method.

Example 4.1. *Standard JD and dragging effect of the null space.* Figure 3 shows a typical convergence history of the Ritz value and the residual for the first two eigenvalues in standard JD. The computed Ritz value θ_j's are constantly dragged toward zero during the subspace iteration. This effect of dragging is also reflected in the convergence history of the residual. Without further treatment on the null space, a significant portion of the CPU time in standard JD is wasted in producing dragged Ritz values and the resulting scheme is much too slow for practical applications. In contrast, the Ritz value converges monotonically for NFJD, as shown in Figure 4.

A standard approach to accelerating convergence in JD is to apply Helmholtz projection as described in section 3.2. In addition to efficient solvers and preconditioners for the correction equation and the Helmholtz projection, the performance of HPJD also relies on proper load balance between them. Denote by τ_{HP} and τ_{JD} the stopping tolerance for (3.10) and (3.2), respectively. More precisely, an approximate eigenpair $(\theta, \boldsymbol{u})$ with $\|\boldsymbol{u}\|_{\mathbb{B}} = 1$ is accepted as a solution to (1.8) provided

$$\|(\mathbb{A} - \theta\mathbb{B})\boldsymbol{u}\|_2 < \tau_{\text{JD}}, \tag{4.3}$$

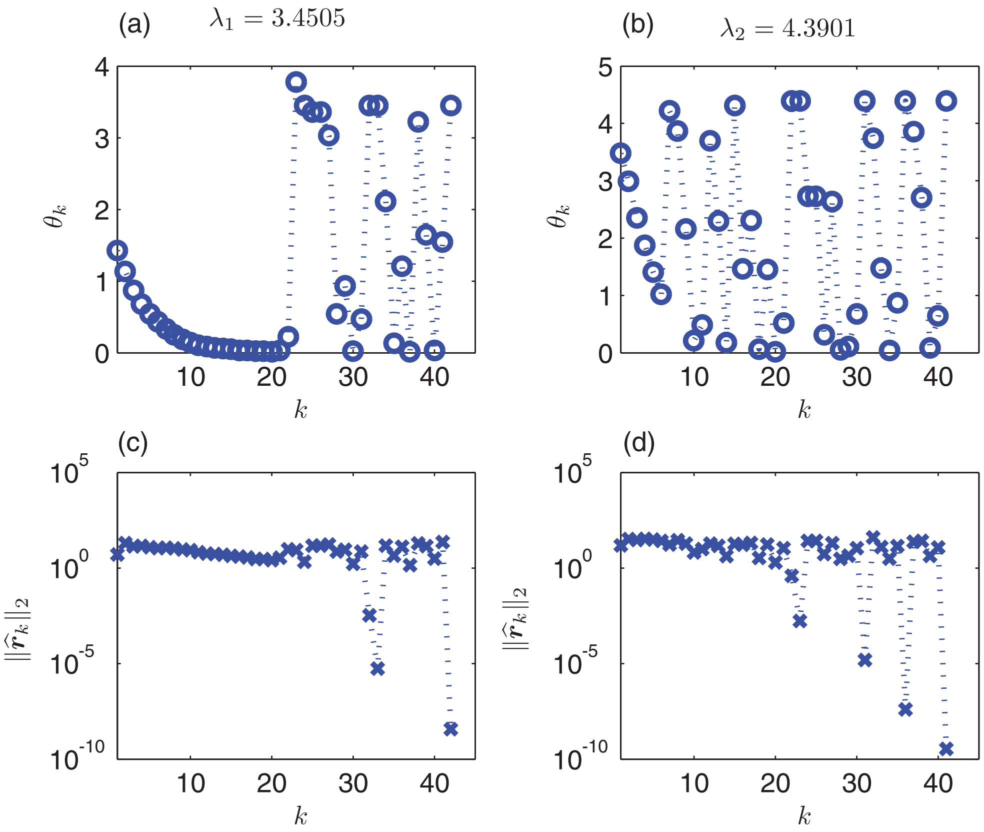

FIG. 3. (a)–(b) *Convergence history of the Ritz value* θ_k *for* λ_1 *and* λ_2 *using standard* JD *with* $\mathbf{k} = (\pi, 0.6\pi, 0)$. (c)–(d) *Convergence history of the residual.*

while an approximate solution ϕ of (3.10) is accepted if

$$\left\| -\mathbb{G}^*\mathbb{B}\mathbb{G}\,\phi + \mathbb{G}^*\mathbb{B}\left(\frac{\boldsymbol{t}}{\|\boldsymbol{t}\|_{\mathbb{B}}}\right)\right\|_2 < \tau_{\mathrm{HP}}. \tag{4.4}$$

Roughly speaking, there is a critical $\tau_{\mathrm{HP}}^{\mathrm{c}}$ (depending on τ_{JD}) such that HPJD does not converge if $\tau_{\mathrm{HP}} > \tau_{\mathrm{HP}}^{\mathrm{c}}$. Since the overall accuracy of HPJD is governed by τ_{JD} only, one should take $\tau_{\mathrm{HP}} \approx \tau_{\mathrm{HP}}^{\mathrm{c}}$ in order to minimize CPU time spent on the Helmholtz projection and get optimal performance. However, it is difficult to predict a priori what $\tau_{\mathrm{HP}}^{\mathrm{c}}$ is. This is a fairly good reason to advocate NFJD and NFJD*.

We have conducted extensive numerical experiments and concluded that

$$\tau_{\mathrm{HP}}^{\mathrm{c}} \lessapprox \tau_{\mathrm{JD}} \tag{4.5}$$

in the sense that $\tau_{\mathrm{HP}}^{\mathrm{c}}$ is generally two or three orders of magnitude smaller than τ_{JD}.

We now proceed to illustrate (4.5) with the following examples.

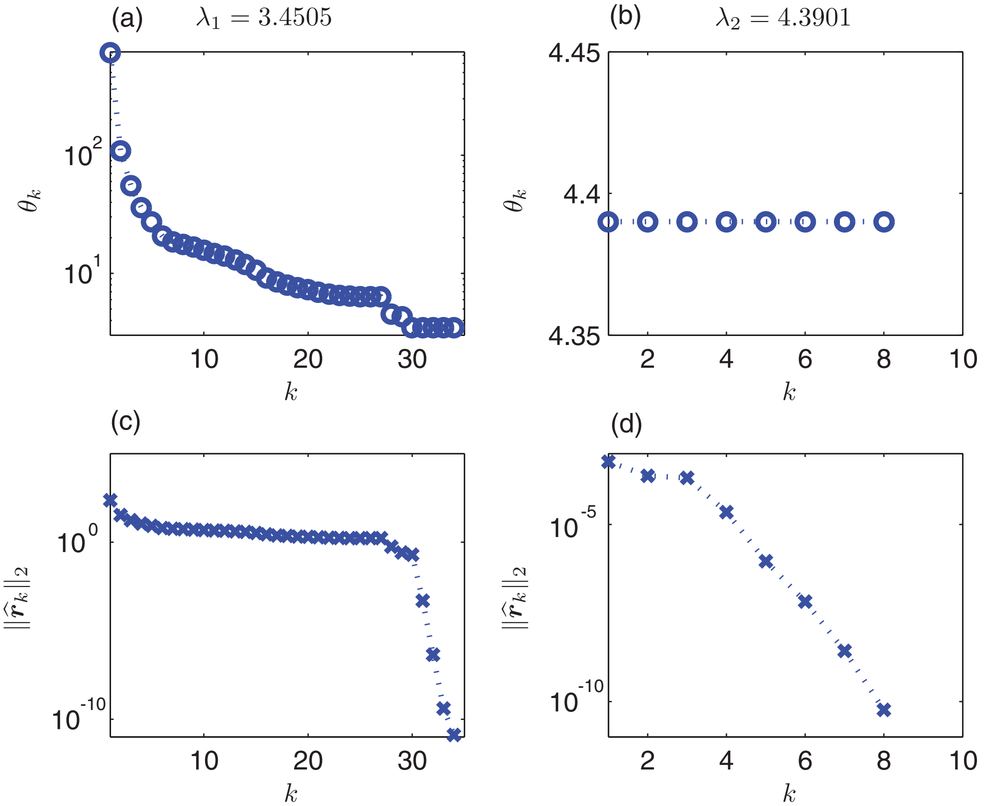

FIG. 4. (a)–(b) *Convergence history of the Ritz value for* λ_1 *and* λ_2 *using* NFJD *with* $\mathbf{k} = (\pi, 0.6\pi, 0)$. (c)–(d) *Convergence history of the residual.*

Example 4.2. HPJD *with* $(\tau_{\text{JD}}, \tau_{\text{HP}}) = (10^{-6}, 10^{-7})$. This is the case where τ_{HP} exceeds the critical value $\tau_{\text{HP}}^{\text{c}}$, resulting in an inexact Helmholtz projection. The remaining null space components accumulate gradually as the subspace $\mathbb{V}_k$ grows. As a result, HPJD fails to converge at higher eigenvalues. We have observed a similar dragging effect as shown in Figure 5,

Example 4.3. *Detailed comparison between* HPJD *and* NFJD. We now compare NFJD against HPJD at its optimal setting. We set $\tau_{\text{JD}} = 10^{-6}$ and $\tau_{\text{NFJD}} = 2.5\times10^{-9}$ according to (4.1). The critical τ_{HP} roughly corresponds to $\tau_{\text{HP}}^{\text{c}} = 10^{-9} \sim 10^{-8}$, depending on other parameters (see below). As τ_{HP} decreases, the overall CPU time for HPJD increases constantly. The smallest positive eigenvalue in our simulation occurs at $\mathbf{k} = (0.1\pi, 0, 0)$ with $\lambda_1 \sim 0.045$. Accordingly, we have set our threshold for NFJD as $\theta_{\text{c}} = 0.01$ (see section 3.3.2 for the reason for setting up the threshold). An approximate eigenvalue in step (i) of NFJD is considered admissible only if $\theta \geq \theta_{\text{c}}$.

In Table 2, we summarize the result for the computation of λ_1 through λ_{10} with different parameters. One of the varying parameters is the restart dimension, denoted by N_{S}. To accelerate convergence, we recycle a number of Ritz vectors after λ_{i-1} converges ($i \geq 2$) and after a restart. These N_{R} Ritz vectors are used to build up initial subspace for λ_i in the beginning and after a restart. This accelerated version of NFJD is summarized as Algorithm 4. The accelerated version of HPJD and NFJD* can be obtained through similar modification.

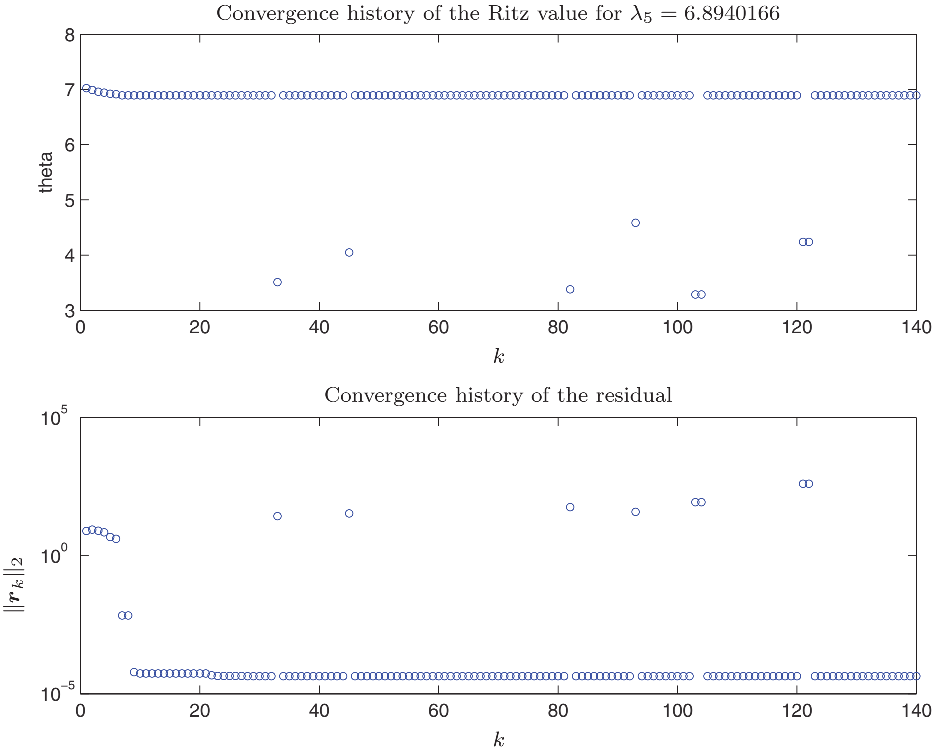

FIG. 5. *Convergence history of Ritz value θ_k and residual $\boldsymbol{r}_k$ for λ_5 at $\mathbf{k} = (0.5\pi, 0, 0)$ when using* HPJD *with $\tau_{\mathrm{JD}} = 10^{-6}$, $\tau_{\mathrm{HP}} = 10^{-7}$.*

TABLE 2
CPU time comparison for first 10 *eigenvalues (averaged over $\mathbf{k}$) with $\tau_{\mathrm{NFJD}} = 2.5 \times 10^{-9}$, $\tau_{\mathrm{JD}} = 10^{-6}$, and $\tau_{\mathrm{HP}} \approx \tau_{\mathrm{HP}}^{\mathrm{c}}$. N_{S}: restart dimension. N_{R}: number of recycled Ritz vectors from λ_{i-1}'s subspace $\mathbb{V}_k$.*

$(N_{\mathrm{R}}, N_{\mathrm{S}})$	τ_{HP}	HPJD	NFJD	Ratio
$(3, 15)$	10^{-8}	5473 sec	3789 sec	1.444
$(5, 15)$	10^{-8}	4897 sec	3536 sec	1.385
$(7, 15)$	10^{-8}	4537 sec	3442 sec	1.318
$(9, 15)$	10^{-8}	4347 sec	3528 sec	1.232

$(N_{\mathrm{R}}, N_{\mathrm{S}})$	τ_{HP}	HPJD	NFJD	Ratio
$(3, 25)$	10^{-9}	5097 sec	3695 sec	1.379
$(5, 25)$	10^{-9}	4720 sec	3412 sec	1.383
$(7, 25)$	10^{-9}	4493 sec	3308 sec	1.358
$(9, 25)$	10^{-9}	4299 sec	3307 sec	1.300

The results in Example 4.3 show that NFJD outperforms HPJD by a significant margin in all cases. In addition, HPJD is considerably slower for small N_{R}'s. On the other hand, the performance of NFJD is relatively insensitive to $(N_{\mathrm{R}}, N_{\mathrm{S}})$.

Next, in Table 3, we document in more detail the relevant components of HPJD and NFJD. One can see that, for sufficiently large N_{R}, the CPU time spent on the JD part are comparable in HPJD and NFJD. The additional CPU time spent in Helmholtz projection constitutes a significant portion in HPJD, even though such an FFT-based Helmholtz projector is indeed extremely efficient. For smaller N_{R}, NFJD is even more efficient on the JD part.

Finally, in Table 4, we record the total CPU time needed for HPJD as τ_{HP} decreases below $\tau_{\mathrm{HP}}^{\mathrm{c}}$ for various cases of $(N_{\mathrm{R}}, N_{\mathrm{S}})$. As expected, the CPU time spent in

Algorithm 4. Accelerated version of NFJD. This is modified from Algorithm 2 with N_{R} recycled Ritz vectors and restart dimension N_{S}. Accelerated versions of HPJD and NFJD* can be similarly modified from Algorithms 1 and 3.

```
Set $\widehat{\mathbb{E}}_0 = [\ ]$, $\Lambda_0 = \varnothing$.
Initialize a vector $\widehat{\boldsymbol{v}}_1$ with $|\widehat{\boldsymbol{v}}_1|_{\widehat{\mathbb{A}}} = 1$.
Set $\widehat{\mathbb{U}}_0 = [\widehat{\boldsymbol{v}}_1]$.
for $i = 1, 2, 3, \ldots, i_{\max}$ do
    Set $\widehat{\mathbb{V}}_1 = [\widehat{\mathbb{E}}_{i-1}, \widehat{\mathbb{U}}_{i-1}]$.
    Compute $\widehat{\mathbb{Z}}_1 = \widehat{\mathbb{V}}_1^* \widehat{\mathbb{B}} \widehat{\mathbb{V}}_1$.
    for $k = 1, 2, 3, \ldots,$ while $\dim(\widehat{\mathbb{V}}_k) \leq i - 1 + N_{\mathrm{S}}$, do
      (i) Compute all the eigenpairs of $(\mathbb{I} - \theta \widehat{\mathbb{Z}}_k)\boldsymbol{s} = \mathbf{0}$.
          Select the desired eigenpair $(\theta_k, \boldsymbol{s}_k)$ with $\theta_k \notin \Lambda_{i-1}$ nearest to the
          target and $\|\boldsymbol{s}_k\|_2 = 1$.
          Select next $N_{\mathrm{R}}$ nearest eigenpairs with $\widetilde{\theta}_1, \ldots, \widetilde{\theta}_{N_{\mathrm{R}}} \notin \Lambda_{i-1}$ and
          $\|\widetilde{\boldsymbol{s}}_1\|_2 = \ldots = \|\widetilde{\boldsymbol{s}}_{N_{\mathrm{R}}}\|_2 = 1$.
      (ii) Compute $\widehat{\boldsymbol{u}}_k = \widehat{\mathbb{V}}_k \boldsymbol{s}_k$, $\widehat{\boldsymbol{r}}_k = (\widehat{\mathbb{A}} - \theta_k \widehat{\mathbb{B}})\widehat{\boldsymbol{u}}_k$.
      (iii) if $\|\widehat{\boldsymbol{r}}_k\|_2 < \tau_{\mathrm{NFJD}}$ then
              Output $\lambda_i = \theta_k$, $\boldsymbol{e}_i = \mathbb{B}^{-1}\mathbb{C}^*\widehat{\mathbb{B}}\boldsymbol{u}_k$.
              Update $\widehat{\mathbb{E}}_i = [\widehat{\mathbb{E}}_{i-1}, \widehat{\boldsymbol{u}}_k]$, $\widehat{\mathbb{U}}_i = [\widehat{\mathbb{V}}_k \widetilde{\boldsymbol{s}}_1, \ldots \widehat{\mathbb{V}}_k \widetilde{\boldsymbol{s}}_{N_{\mathrm{R}}}]$,  $\Lambda_i = \Lambda_{i-1} \cup \{\lambda_i\}$.
              Exit $k$.
          else
              if $\dim(\widehat{\mathbb{V}}_k) = i - 1 + N_{\mathrm{S}}$ then
                  Restart $\widehat{\mathbb{V}}_{k+1} = [\widehat{\mathbb{E}}_{i-1}, \widehat{\mathbb{V}}_k \boldsymbol{s}_k, \widehat{\mathbb{V}}_k \widetilde{\boldsymbol{s}}_1, \ldots \widehat{\mathbb{V}}_k \widetilde{\boldsymbol{s}}_{N_{\mathrm{R}}}]$,
                  $\widehat{\mathbb{Z}}_{k+1} = \widehat{\mathbb{V}}_{k+1}^* \widehat{\mathbb{A}} \widehat{\mathbb{V}}_{k+1}$.
              else
                  Solve (approximately)
                    $(\mathbb{I} - \widehat{\mathbb{B}}\widehat{\boldsymbol{u}}_k \widehat{\boldsymbol{u}}_k^* \widehat{\mathbb{A}} \widehat{\mathbb{B}}^{-1})(\widehat{\mathbb{A}} - \theta_k \widehat{\mathbb{B}})(\mathbb{I} - \widehat{\boldsymbol{u}}_k \widehat{\boldsymbol{u}}_k^* \widehat{\mathbb{A}})\widehat{\boldsymbol{t}} = -\widehat{\boldsymbol{r}}_k, \quad \widehat{\boldsymbol{t}} \perp_{\widehat{\mathbb{A}}} \widehat{\boldsymbol{u}}_k$.
                  $\widehat{\mathbb{A}}$-orthonormalize $\widehat{\boldsymbol{t}}$ against $\widehat{\mathbb{V}}_k$: $\widehat{\boldsymbol{v}}_{k+1} = \dfrac{\widehat{\boldsymbol{t}} - \sum_{\ell=1}^{k}(\widehat{\boldsymbol{v}}_\ell^* \widehat{\mathbb{A}} \widehat{\boldsymbol{t}})\widehat{\boldsymbol{v}}_\ell}{|\widehat{\boldsymbol{t}} - \sum_{\ell=1}^{k}(\boldsymbol{v}_\ell^* \widehat{\mathbb{A}} \widehat{\boldsymbol{t}})\widehat{\boldsymbol{v}}_\ell|_{\widehat{\mathbb{A}}}}$.
                  Expand $\widehat{\mathbb{V}}_{k+1} = [\widehat{\mathbb{V}}_k, \widehat{\boldsymbol{v}}_{k+1}]$, $\widehat{\mathbb{Z}}_{k+1} = \begin{bmatrix} \widehat{\mathbb{Z}}_k & \widehat{\mathbb{V}}_k^* \widehat{\mathbb{B}} \widehat{\boldsymbol{v}}_{k+1} \\ \widehat{\boldsymbol{v}}_{k+1}^* \widehat{\mathbb{B}} \widehat{\mathbb{V}}_k & \widehat{\boldsymbol{v}}_{k+1}^* \widehat{\mathbb{B}} \widehat{\boldsymbol{v}}_{k+1} \end{bmatrix}$.
              end if
          end if
    end for $k$
end for $i$
```

the Helmholtz projection increases constantly as τ_{HP} decreases, while the CPU time for the JD part remains roughly the same (details are not shown here). The result confirms that the CPU time in Table 2 is indeed optimal for HPJD. However, we wish to reiterate that it is not easy to predict the critical $\tau_{\mathrm{HP}}^{\mathrm{c}}$ and get optimal performance for HPJD. The safe play, taking τ_{HP} to machine accuracy, may result in a significant increase in CPU time for HPJD. In contrast, our scheme NFJD is free from such consideration.

TABLE 3
Detailed comparison of HPJD *and* NFJD *with* $\tau_{\mathrm{JD}} = 10^{-6}$, $\tau_{\mathrm{NFJD}} = 2.5 \times 10^{-9}$.

	HPJD	NFJD	HPJD	NFJD
$(N_{\mathrm{R}}, N_{\mathrm{S}}, (\tau_{\mathrm{HP}}))$	$(3, 15, (10^{-8}))$	$(3, 15)$	$(9, 15, (10^{-8}))$	$(9, 15)$
avg. CPU time	5473 sec	3789 sec	4347 sec	3528 sec
avg. num. of JD iter.	161.5	143.8	149.0	142.5
avg. CPU time on solving correction eq.	3961 sec	3455 sec	2736 sec	2803 sec
avg. num. of $\mathbb{M}$ or $\widehat{\mathbb{M}}$ inversions per JD iter.	11.96	11.82	9.182	9.906
avg. CPU time on Helmholtz projection	1123 sec		959.0 sec	
avg. num. of $\mathbb{G}^*\mathbb{G}$ inversions per JD iter.	18.95		17.55	
	HPJD	NFJD	HPJD	NFJD
$(N_{\mathrm{R}}, N_{\mathrm{S}}, (\tau_{\mathrm{HP}}))$	$(3, 25, (10^{-9}))$	$(3, 25)$	$(9, 25, (10^{-9}))$	$(9, 25)$
avg. CPU time	5097 sec	3695 sec	4299 sec	3307 sec
avg. num. of JD iter.	149.8	138.9	135.8	133.2
avg. CPU time on solving correction eq.	3526 sec	3381 sec	2706 sec	2788 sec
avg. num. of $\mathbb{M}$ or $\widehat{\mathbb{M}}$ inversions per JD iter.	11.53	11.89	9.870	10.48
avg. CPU time on Helmholtz projection	1178 sec		1061 sec	
avg. num. of $\mathbb{G}^*\mathbb{G}$ inversions per JD iter.	21.53		21.28	

TABLE 4
CPU time for HPJD *on first* 10 *eigenvalues with* $\tau_{\mathrm{JD}} = 10^{-6}$ *and different* τ_{HP}*'s.*

$(N_{\mathrm{R}}, N_{\mathrm{S}})$	(3,15)	$(9, 15)$	(3,25)	$(9, 25)$
$\tau_{\mathrm{HP}} = 10^{-8}$	5473 sec	4347 sec	5056 sec	not conv.
$\tau_{\mathrm{HP}} = 10^{-9}$	5561 sec	4450 sec	5097 sec	4299 sec
$\tau_{\mathrm{HP}} = 10^{-11}$	5788 sec	4635 sec	5341 sec	4434 sec
$\tau_{\mathrm{HP}} = 10^{-13}$	5954 sec	4835 sec	5593 sec	4625 sec
$\tau_{\mathrm{HP}} = 10^{-15}$	6204 sec	5062 sec	5799 sec	4814 sec

5. Conclusion.

We have proposed an efficient numerical scheme for time harmonic Maxwell's equations. The novelty of our approach include the combination of the JD method with the discrete vector potential and delicate interplay between equivalent forms of Maxwell's equations. By lifting the correction equation up to the vector potential and mapping the (approximate) solution back to the original vector space, our scheme retains the merits of the JD method and filters out the spurious null space almost cost free. We believe these ideas may be generalized to other eigensolvers. Numerical evidence also confirms the efficiency and robustness of the new scheme.

Appendix A. Vector potential formulation for edge elements.

The combination of vector potential with JD iteration is not limited to the finite difference setting. It carries over naturally to other spatially compatible discretizations. Here we demonstrate the procedure in the finite element setting.

Let K be an interface conforming tetrahedral tessellation of Ω. We consider a family of finite element spaces

$$X^h_{\mathscr{V}} \subset H^1(\Omega), \quad X^h_{\mathscr{E}} \subset H(\mathrm{curl}; \Omega), \quad X^h_{\mathscr{F}} \subset H(\mathrm{div}; \Omega), \quad X^h_{\mathscr{C}} \subset L^2(\Omega), \tag{A.1}$$

based on the same tessellation K such that

$$\nabla X^h_{\mathscr{V}} \subset X^h_{\mathscr{E}}, \quad \nabla \times X^h_{\mathscr{E}} \subset X^h_{\mathscr{F}}, \quad \nabla \cdot X^h_{\mathscr{F}} \subset X^h_{\mathscr{C}}. \tag{A.2}$$

Here $\Omega = [0,1]^3$ with periodic extension. Functions in these finite element spaces are complex valued and satisfy the $\mathbf{k}$-periodic boundary condition. The gradient, curl, and divergence in (A.2) are realized in the sense of distribution.

Denote by $\mathcal{V}$, $\mathcal{E}$, $\mathcal{F}$, and $\mathcal{C}$ the collections of degrees of freedoms associated with $X^h_{\mathcal{V}}$, $X^h_{\mathcal{E}}$, $X^h_{\mathcal{F}}$, and $X^h_{\mathcal{C}}$, respectively. In the lowest order case, $\mathcal{V}$, $\mathcal{E}$, $\mathcal{F}$, and $\mathcal{C}$ can simply be identified with the vertices, edges, faces, and barycenters of the tetrahedrons in K.

Well-known examples in (A.1) include curl-conforming edge elements [23, 24] and div-conforming Nédélec elements [23, 24], Raviart–Thomas elements [28], and Brezzi–Douglas–Marini elements [8]. Proper combination of finite element spaces in (A.1) results in discrete de Rham subcomplexes that satisfy the inclusion relation (A.2) and the celebrated "commuting de Rham diagram" property. Examples of discrete de Rham complexes can be found in [22, pp. 150, 209] and [4, p. 60]. See also [13] for systematic construction of finite element de Rham complexes from the viewpoint of differential forms and exterior calculus. In what follows, we will give a brief derivation of finite element discretization of Maxwell's equation in $X^h_{\mathcal{E}}$ and Algorithm 3. The edge element based discretization only involves the subchain $(X^h_{\mathcal{V}}, X^h_{\mathcal{E}}, X^h_{\mathcal{F}})$.

The weak formulation of (1.5) in the curl-conforming finite element space $X^h_{\mathcal{E}}$ is given by the following: Find $\boldsymbol{E} \in X^h_{\mathcal{E}}$, $\lambda > 0$ such that

$$\langle \nabla \times \boldsymbol{\Phi}, \mu_r^{-1} \nabla \times \boldsymbol{E} \rangle_\Omega = \lambda \langle \boldsymbol{\Phi}, \varepsilon_r \boldsymbol{E} \rangle_\Omega \quad \text{for all } \boldsymbol{\Phi} \in X^h_{\mathcal{E}}. \tag{A.3}$$

Following our notational convention, we denote by $\boldsymbol{E}$ an element in $X^h_{\mathcal{E}}$ and its column vector representation by $\boldsymbol{e}$. That is, $\boldsymbol{E}(\mathbf{x}) = \sum_{j=1}^{|\mathcal{E}|} e_j \boldsymbol{\Phi}_j(\mathbf{x})$, where $\{\boldsymbol{\Phi}_j\}_{j=1}^{|\mathcal{E}|}$ is a basis for $X^h_{\mathcal{E}}$. From the second inclusion of (A.2), the curl operator induces a linear mapping

$$\nabla_h \times := \nabla \times |_{X^h_{\mathcal{E}}} : X^h_{\mathcal{E}} \mapsto X^h_{\mathcal{F}}$$

and makes it possible to generalize the vector potential formulation to (A.3). Denote by $\mathbb{C}$ the matrix representation of $\nabla_h \times$ with respect to the (real valued) basis functions $\boldsymbol{\Phi}_j \in X^h_{\mathcal{E}}$, $\boldsymbol{\Psi}_i \in X^h_{\mathcal{F}}$. That is, if $\boldsymbol{E}(\mathbf{x}) = \sum_{j=1}^{|\mathcal{E}|} e_j \boldsymbol{\Phi}_j(\mathbf{x})$, $\boldsymbol{F}(\mathbf{x}) = \sum_{i=1}^{|\mathcal{F}|} f_i \boldsymbol{\Psi}_i(\mathbf{x})$, and $\boldsymbol{F}(\mathbf{x}) = \nabla \times \boldsymbol{E}(\mathbf{x})$, then $f_i = \sum_{j=1}^{|\mathcal{E}|} \mathbb{C}_{ij} e_j$.

In matrix notation, (A.3) reads

$$\mathbb{A}\boldsymbol{e} = \lambda \mathbb{B}\boldsymbol{e}, \text{ where } \mathbb{A} = \mathbb{C}^* \mathbb{M}_{\mathcal{F}} \mathbb{C},\ \mathbb{B} = \mathbb{M}_{\mathcal{E}}, \tag{A.4}$$

and $\mathbb{M}_{\mathcal{E}}$, $\mathbb{M}_{\mathcal{F}}$ are the mass matrices generated by the basis functions in $X^h_{\mathcal{E}}$ and $X^h_{\mathcal{F}}$, respectively:

$$(\mathbb{M}_{\mathcal{E}})_{j,j'} = \langle \boldsymbol{\Phi}_j, \varepsilon_r \boldsymbol{\Phi}_{j'} \rangle_\Omega, \quad (\mathbb{M}_{\mathcal{F}})_{i,i'} = \langle \boldsymbol{\Psi}_i, \mu_r^{-1} \boldsymbol{\Psi}_{i'} \rangle_\Omega. \tag{A.5}$$

The mass matrices $\mathbb{M}_{\mathcal{E}}$ and $\mathbb{M}_{\mathcal{F}}$ induce natural inner products on $X^h_{\mathcal{E}}$ and $X^h_{\mathcal{F}}$:

$$\langle \boldsymbol{E}, \boldsymbol{U} \rangle_{\varepsilon_r} := \boldsymbol{e}^* \mathbb{M}_{\mathcal{E}} \boldsymbol{u} = \int_\Omega \bar{\boldsymbol{E}}(\mathbf{x}) \cdot \varepsilon_r(\mathbf{x}) \boldsymbol{U}(\mathbf{x}) d\mathbf{x}, \quad \boldsymbol{E}, \boldsymbol{U} \in X^h_{\mathcal{E}}, \tag{A.6}$$

$$\langle \boldsymbol{F}, \boldsymbol{H} \rangle_{\mu_r^{-1}} := \boldsymbol{f}^* \mathbb{M}_{\mathcal{F}} \boldsymbol{h} = \int_\Omega \bar{\boldsymbol{F}}(\mathbf{x}) \cdot \mu_r^{-1}(\mathbf{x}) \boldsymbol{H}(\mathbf{x}) d\mathbf{x}, \quad \boldsymbol{F}, \boldsymbol{H} \in X^h_{\mathcal{F}}. \tag{A.7}$$

Denote by $\mathbb{G}$ the matrix representation of $\nabla_h := \nabla|_{X^h_{\mathcal{V}}} : X^h_{\mathcal{V}} \mapsto X^h_{\mathcal{E}}$. Then the solutions to (A.4) with $\lambda \neq 0$ satisfy the discrete analogue of the divergence free constraint (1.6) automatically:

$$-\mathbb{G}^*\mathbb{B}\boldsymbol{e} = -\frac{1}{\lambda}\mathbb{G}^*\mathbb{C}^*\mathbb{M}_{\mathcal{F}}\mathbb{C}\boldsymbol{e} = \mathbf{0}.$$

This is a direct consequence of (1.9) and the fact that $\mathbb{C}\mathbb{G}$ is the matrix representation of $\nabla \times \nabla$ on $X^h_{\mathcal{V}}$.

The finite element version of NFJD and NFJD* are identical to Algorithm 2 and Algorithm 3, respectively. Here we give a brief derivation of NFJD* for (A.4). We start with the following reformulation of (A.4):

$$\widehat{\mathbb{A}}\widehat{\boldsymbol{e}} = \lambda\widehat{\mathbb{B}}\widehat{\boldsymbol{e}}, \text{ where } \widehat{\mathbb{B}} = \mathbb{M}_{\mathcal{F}},\ \widehat{\mathbb{A}} = \mathbb{M}_{\mathcal{F}}\mathbb{C}\mathbb{M}_{\mathcal{E}}^{-1}\mathbb{C}^*\mathbb{M}_{\mathcal{F}} = \widehat{\mathbb{B}}\mathbb{C}\mathbb{B}^{-1}\mathbb{C}^*\widehat{\mathbb{B}}. \tag{A.8}$$

Note that (A.8) is merely an equivalent formulation of (A.4) obtained through algebraic manipulation. It is not meant to be a new finite element discretization of the magnetic field equation $\mu_r^{-1}\nabla \times \varepsilon_r^{-1}\nabla \times \mu_r^{-1}\widehat{\boldsymbol{E}} = \lambda\mu_r^{-1}\widehat{\boldsymbol{E}}$ on $X^h_{\mathcal{F}}$.

In general, (A.8) is less suitable for numerical computation than (A.4). In standard numerical methods, the core matrix-vector multiplication $\widehat{\mathbb{A}}\widehat{\boldsymbol{v}}$ for (A.8) is more expensive than the corresponding operation $\mathbb{A}\boldsymbol{v}$ for (A.4). We will see, however, that the situation is reversed in their NFJD formulations Algorithm 2 (for (A.4)) and Algorithm 3 (for (A.8)).

Since both $\widehat{\mathbb{A}}$ and $\widehat{\mathbb{B}}$ are self-adjoint with $\widehat{\mathbb{B}}$ positive definite, the spectral decomposition argument used in Theorem 2.2 remains valid. In other words, we have

$$\begin{aligned}\ker(\widehat{\mathbb{A}})^{\perp_{\mathbb{M}_{\mathcal{F}}}} &= \ker(\mathbb{M}_{\mathcal{E}}^{-1}\mathbb{C}^*\mathbb{M}_{\mathcal{F}})^{\perp_{\mathbb{M}_{\mathcal{F}}}}\\ &= \operatorname{Span}\{\widehat{\boldsymbol{v}}_j \in X^h_{\mathcal{F}} \,|\, \mathbb{M}_{\mathcal{F}}\mathbb{C}\mathbb{M}_{\mathcal{E}}^{-1}\mathbb{C}^*\mathbb{M}_{\mathcal{F}}\widehat{\boldsymbol{v}}_j = \lambda_j\mathbb{M}_{\mathcal{F}}\widehat{\boldsymbol{v}}_j,\ \lambda_j > 0\},\end{aligned} \tag{A.9}$$

and the following counterpart of Theorem 2.2 holds.

Theorem A.1. *Let $\widehat{\boldsymbol{u}} \in X^h_{\mathcal{F}}$. Then $\widehat{\boldsymbol{u}} \in \ker(\widehat{\mathbb{A}})^{\perp_{\mathbb{M}_{\mathcal{F}}}}$ if and only if $\widehat{\boldsymbol{u}} = \mathbb{C}\boldsymbol{u}$ for some $\boldsymbol{u} \in X^h_{\mathcal{E}}$.*

The standard JD method applied to (A.8) results in the following correction equation:

$$(\mathbb{I} - \widehat{\mathbb{B}}\widehat{\boldsymbol{u}}_k\widehat{\boldsymbol{u}}_k^*)(\widehat{\mathbb{A}} - \theta_k\widehat{\mathbb{B}})(\mathbb{I} - \widehat{\boldsymbol{u}}_k\widehat{\boldsymbol{u}}_k^*\widehat{\mathbb{B}})\widehat{\boldsymbol{t}} = -\widehat{\boldsymbol{r}}_k, \qquad \widehat{\boldsymbol{u}}_k^*\widehat{\mathbb{B}}\widehat{\boldsymbol{t}} = 0. \tag{A.10}$$

In view of Theorem A.1, we can now substitute $\widehat{\boldsymbol{t}} = \mathbb{C}\boldsymbol{t}$, $\widehat{\boldsymbol{u}}_k = \mathbb{C}\boldsymbol{u}_k$ into (A.10) to get

$$\begin{aligned}&(\mathbb{I} - \widehat{\mathbb{B}}\widehat{\boldsymbol{u}}_k\widehat{\boldsymbol{u}}_k^*)(\widehat{\mathbb{A}} - \theta_k\widehat{\mathbb{B}})(\mathbb{I} - \widehat{\boldsymbol{u}}_k\widehat{\boldsymbol{u}}_k^*\widehat{\mathbb{B}})\widehat{\boldsymbol{t}}\\ &\quad= \big(\mathbb{I} - \widehat{\mathbb{B}}\mathbb{C}\boldsymbol{u}_k\boldsymbol{u}_k^*\mathbb{C}^*\big)\big(\widehat{\mathbb{B}}\mathbb{C}\mathbb{B}^{-1}\mathbb{C}^*\widehat{\mathbb{B}} - \theta_k\widehat{\mathbb{B}}\big)\mathbb{C}\big(\mathbb{I} - \mathbb{C}\boldsymbol{u}_k\boldsymbol{u}_k^*\mathbb{C}^*\widehat{\mathbb{B}}\mathbb{C}\big)\boldsymbol{t}\\ &\quad= \big(\mathbb{I} - \widehat{\mathbb{B}}\mathbb{C}\boldsymbol{u}_k\boldsymbol{u}_k^*\mathbb{C}^*\big)\widehat{\mathbb{B}}\mathbb{C}\mathbb{B}^{-1}\big(\mathbb{C}^*\widehat{\mathbb{B}}\mathbb{C} - \theta_k\mathbb{B}\big)\big(\mathbb{I} - \mathbb{C}\boldsymbol{u}_k\boldsymbol{u}_k^*\mathbb{A}\big)\boldsymbol{t}\\ &\quad= \widehat{\mathbb{B}}\mathbb{C}\mathbb{B}^{-1}\big(\mathbb{I} - \mathbb{B}\boldsymbol{u}_k\boldsymbol{u}_k^*\mathbb{A}\mathbb{B}^{-1}\big)\big(\mathbb{A} - \theta_k\mathbb{B}\big)\big(\mathbb{I} - \mathbb{C}\boldsymbol{u}_k\boldsymbol{u}_k^*\mathbb{A}\big)\boldsymbol{t}\end{aligned} \tag{A.11}$$

and

$$\widehat{\boldsymbol{r}}_k = (\widehat{\mathbb{A}} - \theta_k\widehat{\mathbb{B}})\widehat{\boldsymbol{u}}_k = (\widehat{\mathbb{B}}\mathbb{C}\mathbb{B}^{-1}\mathbb{C}^*\widehat{\mathbb{B}} - \theta_k\widehat{\mathbb{B}})\mathbb{C}\boldsymbol{u}_k = \widehat{\mathbb{B}}\mathbb{C}\mathbb{B}^{-1}(\mathbb{A} - \theta_k\mathbb{B})\boldsymbol{u}_k = \widehat{\mathbb{B}}\mathbb{C}\mathbb{B}^{-1}\boldsymbol{r}_k. \tag{A.12}$$

In addition, $\widehat{\boldsymbol{u}}_k^*\widehat{\mathbb{B}}\widehat{\boldsymbol{t}} = \boldsymbol{u}_k^*\mathbb{C}^*\widehat{\mathbb{B}}\mathbb{C}\boldsymbol{t} = \boldsymbol{u}_k^*\mathbb{A}\boldsymbol{t}$. Thus the correction equation can be translated to $X^h_{\mathcal{E}}$ as

$$\big(\mathbb{I} - \mathbb{B}\boldsymbol{u}_k\boldsymbol{u}_k^*\mathbb{A}\mathbb{B}^{-1}\big)\big(\mathbb{A} - \theta_k\mathbb{B}\big)\big(\mathbb{I} - \boldsymbol{u}_k\boldsymbol{u}_k^*\mathbb{A}\big)\boldsymbol{t} = -\boldsymbol{r}_k, \qquad \boldsymbol{u}_k^*\mathbb{A}\boldsymbol{t} = 0. \tag{A.13}$$

Finally, we switch back to the electric field pencil $(\mathbb{A}, \mathbb{B})$ on the subspace eigenvalue problem: Find $\boldsymbol{u}_k \in \text{Range}(\mathbb{V}_k)$, $\theta_k > 0$ nearest to the target such that

$$\boldsymbol{v}^*(\mathbb{A} - \theta_k \mathbb{B})\boldsymbol{u}_k = 0 \text{ for all } \boldsymbol{v} \in \text{Range}(\mathbb{V}_k). \tag{A.14}$$

With the orthonormalization $\boldsymbol{v}_l^* \mathbb{A} \boldsymbol{v}_m = \delta_{lm}$, (A.14) leads to the following subspace eigenvalue problem and the $\mathbb{B}$-normalized approximate eigenvector:

$$(\mathbb{I} - \theta_k \mathbb{Z}_k)\boldsymbol{s}_k = \boldsymbol{0}, \quad (\mathbb{Z}_k)_{\ell m} = \boldsymbol{v}_\ell^* \mathbb{B} \boldsymbol{v}_m; \quad \boldsymbol{u}_k = \frac{\mathbb{V}_k \boldsymbol{s}_k}{\|\mathbb{V}_k \boldsymbol{s}_k\|_{\mathbb{B}}}. \tag{A.15}$$

Acknowledgments. The authors would like to acknowledge the support from the Ministry of Science and Technology of Taiwan. In addition, W.-W. Lin would like to acknowledge the support from National Center for Theoretical Sciences in Taiwan and the ST Yau Center at National Chiao Tung University.

REFERENCES

[1] P. Arbenz, *A comparison of factorization-free eigensolvers with application to cavity resonators*, in Computational Science, Lecture Notes in Comput. Sci. 2331, Springer, Berlin, 2009, pp. 295–304.

[2] P. Arbenz and R. Geus, *A comparison of solvers for large eigenvalue problems occurring in the design of resonant cavities*, Numer. Linear Algebr. Appl., 6 (1999), pp. 3–16.

[3] P. Arbenz and R. Geus, *Multilevel preconditioned iterative eigensolvers for Maxwell eigenvalue problems*, Appl. Numer. Math., 54 (2005), pp. 107–121.

[4] D. N. Arnold, R. S. Falk, and R. Winther, *Finite element exterior calculus, homological techniques, and applications*, Acta Numer., 15 (2006), pp. 1–155.

[5] 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.

[6] R. Biswas, M. M. Sigalas, K.-M. Ho, and S.-Y. Lin. *Three-dimensional photonic band gaps in modified simple cubic lattices*, Phys. Rev. B, 65 (2002), 205121.

[7] A. Bossavit, *Mixed finite elements and the complex of Whitney forms*, in J. Whiteman, ed., The Mathematics of Finite Elements and Applications VI, Academic Press, San Diego, CA, 1988, pp. 137–144.

[8] F. Brezzi, J. Douglas, Jr., and I. D. Marini, *Two families of mixed finite elements for second order elliptic problems*, Numer. Math., 47 (1985), 217–235.

[9] R. L. Chern, C. C. Chang, C. C. Chang, and R. R. Hwang, *Numerical study of three-dimensional photonic crystals with large band gaps*, J. Phys. Soc. Jpn., 73 (2004), pp. 727–737.

[10] M. P. do Carmo, *Differential Forms and Applications*, Springer-Verlag, Berlin, 1994.

[11] H. Flanders, *Differential Forms with Applications to the Physical Sciences*, Dover Publications, New York, 1989.

[12] D. R. Fokkema, G. L. G. Sleijpen, and H. A. van der Vorst, *Jacobi-Davidson style QR and QZ algorithms for the reduction of matrix pencils*, SIAM J. Sci. Comput., 20 (1998), pp. 94–125.

[13] R. Hiptmair, *Canonical construction of finite elements*, Math. Comp., 68 (1999), pp. 1325–1346.

[14] R. Hiptmair, *Finite elements in computational electromagnetism*, Acta Numer., (2002), pp. 237–239.

[15] R. Hiptmair and K. Neymeyr, *Multilevel method for mixed eigenproblems*, SIAM J. Sci. Comput., 23 (2002), pp. 2141–2164.

[16] Y.-L. Huang and W.-C. Wang, *Adaptive computation of the corner singularity with the monotone jump condition capturing scheme*, in Recent Advances in Adaptive Computation, Contemp. Math. 383, AMS, Providence, RI, 2005, pp. 253–267.

[17] T.-M. Hwang, W.-W. Lin, J.-L. Liu, and W. Wang, *Jacobi–Davidson methods for cubic eigenvalue problems*, Numer. Linear Algebra Appl., 12 (2005), pp. 605–624.

[18] T.-M. Hwang, W.-W. Lin, W.-C. Wang, and W. Wang, *Numerical simulation of three dimensional pyramid quantum dot*, J. Comput. Phys., 196 (2004), pp. 208–232.

[19] T.-M. Hwang, W.-C. Wang, and W. Wang, *Numerical schemes for three-dimensional irregular shape quantum dots over curvilinear coordinate systems*, J. Comput. Phys., 226 (2007), pp. 754–773.

[20] J. M. Hyman and M. Shashkov, *Mimetic discretizations of Maxwell's equations*, J. Comput. Phys., 151 (1999), pp. 881–909.

[21] C. Kittel, *Introduction to Solid State Physics*, John Wiley, New York, 2005.

[22] P. Monk, *Finite Element Methods for Maxwell's Equations*, Oxford University Press, New York, 2003.

[23] J.-C. Nédélec, *Mixed finite elements in* $\mathbf{R}^3$, Numer. Math., 35 (1980), pp. 315–341.

[24] J.-C. Nédélec, *A new class of mixed finite elements in* $\mathbf{R}^3$, Numer. Math., 50 (1986), pp. 57–81.

[25] R. A. Nicolaides and D.-Q. Wang, *Convergence analysis of a covolume scheme for Maxwell's equations in three dimensions*, Math. Comp., 67 (1998), 947–963.

[26] Y. Notay, *Combination of Jacobi-Davidson and conjugate gradients for the partial symmetric eigenproblem*, Numer. Linear Algebra Appl., 9 (2002), pp. 21–44.

[27] B. N. Parlett, *The Symmetric Eigenvalue Problem*, SIAM, Philadelphia, 1998.

[28] P. A. Raviart and J. M. Thomas, *A mixed finite element method for second order elliptic problems*, in Mathematical Aspects of the Finite Element Method, Lecture Notes in Math. 606, Springer-Verlag, Berlin, 1977.

[29] W. Rudin, *Principles of Mathematical Analysis,* 3rd ed., McGraw-Hill, New York, 1976.

[30] V. Simoncini, *Algebraic formulations for the solution of the nullspace-free eigenvalue problem using the inexact shift-and-invert Lanczos method*, Numer. Linear Algebra Appl., 10 (2003), pp. 357–375.

[31] 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 (1996), pp. 595–633.

[32] 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 (1996), pp. 401–425.

[33] R. Temam, *Navier-Stokes equations: Theory and numerical analysis*, AMS, Providence, RI, 2001.

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

[35] W.-C. Wang, *A jump condition capturing finite difference scheme for elliptic interface problems*, SIAM J. Sci. Comput., 25 (2004), pp. 1479–1496.

[36] H. Whitney, *Geometric Integration Theory.* Princeton University Press, Princeton, NJ, 1957.

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