A NULL SPACE FREE JACOBI-DAVIDSON ITERATION FOR MAXWELL’S OPERATOR

TSUNG-MING HUANG †, YIN-LIANG 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 the degenerate elliptic operator arising from Maxwell’s equations. We consider spatial compatible discretizations such as Yee’s scheme which guarantee the existence of a discrete vector potential. During the Jacobi-Davidson iteration, the correction process is applied to the vector potential instead. The correction equation is solved approximately as in original 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 new method is much more efficient than the original Jacobi-Davidson method.

Key words. Maxwell’s equations, Yee’s scheme, generalized eigenvalue problem, discrete vector potential, Poincaré Lemma, Jacobi-Davidson method

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

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 past few decades, photonic crystals with specific band structures are of practical interest and have been extensively studied. The governing equation for three-dimensional photonic crystals is the time harmonic Maxwell’s equations:

\begin{align}
\nabla \times \mathbf{H} &= i\omega \varepsilon \mathbf{E} \\
\n\nabla \times \mathbf{E} &= -i\omega \mu_0 \mathbf{H} \\
\n\nabla \cdot (\varepsilon \mathbf{E}) &= 0 \\
\n\n\nabla \cdot (\mathbf{H}) &= 0
\end{align}

or equivalently

\begin{align}
\nabla \times \nabla \times \mathbf{E} &= \lambda \varepsilon \mathbf{E} \\
\n\nabla \cdot (\varepsilon \mathbf{E}) &= 0
\end{align}

Here it is written as equations for the electric field \( \mathbf{E} \) and the unknown eigenvalue \( \lambda = \mu_0 \omega^2 \) where \( \omega \) is the frequency. The degenerate elliptic operator \( \nabla \times \nabla \times \) is self-adjoint and non-negative. The permittivity \( \varepsilon \) is a material dependent constant and therefore (1.5) is an elliptic interface problem.

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 calculus identity

\begin{align}
\nabla \cdot \nabla \times \equiv 0.
\end{align}

A traditional wisdom to reflect this fact is to adopt spatial discretizations that admit discrete analogue of (1.7). Namely, those satisfying

\begin{align}
\nabla_h \cdot \nabla_h \times \equiv 0
\end{align}

This work is partially supported by the National Science Council, the National Center for Theoretical Sciences, and the Center of Mathematical Modeling and Scientific Computing (National Chiao Tung University) in Taiwan.

†Department of Mathematics, National Taiwan Normal University, Taipei, 116, Taiwan (min@math.ntnu.edu.tw).
‡Department of Mathematics, National Tsing Hua University, HsinChu, 300, Taiwan. (yin1108@gmail.com)
§Department of Applied Mathematics, National Chiao Tung University, and Division of Mathematics, National Center for Theoretical Science, National Tsing Hua University, Hsinchu 300, Taiwan (wwlin@math.nctu.edu.tw).
¶Department of Mathematics, National Tsing Hua University, HsinChu, 300, Taiwan. (wangwc@math.nthu.edu.tw)
where $\nabla_h$ and $\nabla_h \times$ denote corresponding discrete divergence and discrete curl operator, respectively.

This class of spatial discretizations includes the Yee’s scheme [21], the Whitney form [20, 4], the co-volume discretization [14], the mimetic discretization [10] and the edge element [12, 13, 15]. With this approach, the divergence free constrained (1.6) is ignored and the resulting discretized system is a generalized eigenvalue problem

$$(1.9) \quad A \vec{e} = \lambda B \vec{e}$$

where $A$ is the matrix representation for the discrete operator $\nabla_h \times \nabla_h \times$ and $B$ is a mass matrix.

In the unconstrained formulation (1.9), the matrix $A$ is symmetric and non-negative definite.

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

$$(1.10) \quad \nabla \times \nabla \phi \equiv 0.$$  

In other words, the null space of $\nabla \times \nabla \times$ contains all the gradient functions. The discrete counterpart of (1.10) holds true for the generalized eigenvalue problem (1.9) 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 iteration. The Jacobi-Davidson method [2, 6, 17, 18] is a well established, efficient eigen-solver based on expanding subspace iteration. Contrast to classical eigen-solvers such as the inverse power method and various Lanczos [1, 16] or Arnoldi methods [2] which require the shift-and-invert technique to compute interior eigen-pairs, the linear system for the corrector in the Jacobi-Davidson iteration only needs to be solved approximately. This is the key to efficiency of the Jacobi-Davidson method. On the other hand, since the corrector is only solved approximately, direct application of the Jacobi-Davidson method inevitably brings null vectors into the expanding subspace. As a result, the efficiency of the Jacobi-Davidson method is significantly deteriorated. The pollution of the spurious null space is common to other eigen-solvers as well. The standard remedy is to project the approximate eigenvector back to the orthogonal complement of the null space. For example, in Chern et al. [5], the inverse power iteration is applied together with a CG method for the projection step.

If we adopt the Jacobi-Davidson method together with the null space projection, the projection step needs to be solved accurately to prevent the null vectors from entering the approximating subspace. This is potentially a bottleneck for computations cost and an impairment to the merit of Jacobi-Davidson method.

In contrast, we propose a null space free Jacobi-Davidson iteration with the aid of Poincaré Lemma, whose discrete analogue remains valid for the class of spatial discretizations under consideration:

**Poincaré Lemma:** If $\nabla \cdot v = 0$ on a contractible domain, then $v = \nabla \times \tilde{v}$ for some vector potential $\tilde{v}$.

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 standard 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 constrain (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 drastically improves the slow convergence caused by the spurious null space in original Jacobi-Davidson iteration.
The rest of the paper is organized as follows. In Section 2, we present detailed mathematical formulations for the eigenvalue problem (1.9), including Theorem 2.2 which characterizes the subspace spanned by the eigenvectors perpendicular to the null space. This is the foundation of our new scheme. In Section 3, we derive our null space free version of Jacobi-Davidson method by direct application of Theorem 2.2. In Section 4, we report the numerical tests to support our theory. We close the paper by summarizing our results in Section 5.

2. Background and Mathematical Formulation. The photonic crystals consist of dielectric materials fabricated in periodic structure. The permittivity $\varepsilon(x)$, as a material dependent constant, is therefore periodic and piecewise constant. In other words,

$$\varepsilon(x + a\ell) = \varepsilon(x), \quad \ell = 1, 2, 3.$$  

and

$$\varepsilon(x) = \begin{cases} \varepsilon_1, & \text{in material 1}, \\ \varepsilon_2, & \text{in material 2}, \end{cases}$$

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

From Bloch’s Theorem [11], the eigenfunctions of (1.5) can be written as

$$E(x) = e^{ik \cdot x} e(x)$$

for some vector $k$ in the first Brillouin zone and $e$ satisfies the periodic condition

$$e(x + a\ell) = e(x), \quad \ell = 1, 2, 3.$$  

To apply our scheme, we will only consider a class of spatially compatible discretizations satisfying (1.8). Such discretizations include Yee’s scheme, the co-volume discretization and the edge elements. In this paper, we only elaborate our scheme for Yee’s discretization due to its simplicity in implementation.

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

$$a_1 = (1, 0, 0), \quad a_2 = (0, 1, 0), \quad a_3 = (0, 0, 1).$$

The corresponding first Brillouin zone is given by

$$\{k = (k_1, k_2, k_3) \in \mathbb{R}^3 \mid -\pi \leq k_j \leq \pi, j = 1, 2, 3\}$$

In Yee’s discretization for (1.1-1.4), the magnetic field and electric field are defined on centers of cell edges and cell faces which we denote by

$$E = E_1 \cup E_2 \cup E_3$$

$$E_1 = \{(x_i - \frac{1}{2}, y_j, z_k)\}, \quad E_2 = \{(x_i, y_j - \frac{1}{2}, z_k)\}, \quad E_3 = \{(x_i, y_j, z_k - \frac{1}{2})\}$$

$$F = F_1 \cup F_2 \cup F_3$$

$$F_1 = \{(x_i - \frac{1}{2}, y_j, z_k - \frac{1}{2})\}, \quad F_2 = \{(x_i, y_j - \frac{1}{2}, z_k - \frac{1}{2})\}, \quad F_3 = \{(x_i - \frac{1}{2}, y_j, z_k - \frac{1}{2})\}$$

$$V' = \{(x_i, y_j, z_k)\}$$
We denote the complex valued functions on \( \mathcal{E} \) and \( \mathcal{F} \) by

\[
L^2(\mathcal{E}) = \{ E_1(i - \frac{1}{2}, j, k), E_2(i, j - \frac{1}{2}, k), E_3(i, j, k - \frac{1}{2}) \} \\
L^2(\mathcal{F}) = \{ F_1(i, j - \frac{1}{2}, k - \frac{1}{2}), F_2(i - \frac{1}{2}, j, k - \frac{1}{2}), F_3(i - \frac{1}{2}, j - \frac{1}{2}, k) \}
\]

the periodic functions by:

\[
L^p_n(\mathcal{E}) = \{ e \in L^2(\mathcal{E}) \mid e_j(x + a_\ell) = e_j(x), x \in \mathcal{E}_j, j, \ell = 1, 2, 3 \} \\
L^p_n(\mathcal{F}) = \{ f \in L^2(\mathcal{F}) \mid f_j(x + a_\ell) = f_j(x), x \in \mathcal{F}_j, j, \ell = 1, 2, 3 \}
\]

and the \( k \)-periodic functions by:

\[
L^p_{n,k}(\mathcal{E}) = \{ E \in L^2(\mathcal{E}) \mid E_j(x + a_\ell) = e^{ik_\ell x} E_j(x), x \in \mathcal{E}_j; j, \ell = 1, 2, 3 \} = \{ e^{ik_\ell x} \mid e \in L^p_n(\mathcal{E}) \} \\
L^p_{n,k}(\mathcal{F}) = \{ F \in L^2(\mathcal{F}) \mid F_j(x + a_\ell) = e^{ik_\ell x} F_j(x), x \in \mathcal{F}_j; j, \ell = 1, 2, 3 \} = \{ e^{ik_\ell x} \mid f \in L^p_n(\mathcal{F}) \}
\]

Therefore the curl operator can be discretized naturally using standard centered differencing:

\[
\nabla_h \times : L^2(\mathcal{F}) \mapsto L^2(\mathcal{E}).
\]

For example, if \( H \in L^2(\mathcal{F}) \), then the first component of \( \nabla_h \times H \) is given by

\[
(\nabla_h \times H)_1(x_i, y_j, z_k) = \frac{H_3(x_i, y_j, z_{k+\frac{1}{2}}) - H_3(x_i, y_j, z_{k-\frac{1}{2}})}{h_2} - \frac{H_2(x_i, y_j, z_{k+\frac{1}{2}}) - H_2(x_i, y_j, z_{k-\frac{1}{2}})}{h_3}
\]

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.

With the same discretization, one can define the discrete curl operator on \( L^2(\mathcal{E}) \):

\[
\nabla^*_h \times : L^2(\mathcal{E}) \mapsto L^2(\mathcal{F}),
\]

with

\[
(\nabla^*_h \times E)_1(x_i, y_j, z_{k+\frac{1}{2}}) = \frac{E_3(x_i, y_j, z_{k+\frac{1}{2}}) - E_3(x_i, y_j, z_{k-\frac{1}{2}})}{h_2} - \frac{E_2(x_i, y_j, z_{k+\frac{1}{2}}) - E_2(x_i, y_j, z_{k-\frac{1}{2}})}{h_3}
\]

and so on.

Similarly, one can define the discrete divergence operator on \( L^2(\mathcal{E}) \):

\[
\nabla^*_h : L^2(\mathcal{E}) \mapsto L^2(\mathcal{F}),
\]

\[
(\nabla^*_h \cdot E)(x_i, y_j, z_k) = \frac{E_1(x_i, y_j, z_k) - E_1(x_{i-\frac{1}{2}}, y_j, z_k)}{h_1} + \frac{E_2(x_i, y_j, z_k) - E_2(x_i, y_{j-\frac{1}{2}}, z_k)}{h_2} + \frac{E_3(x_i, y_j, z_{k+\frac{1}{2}}) - E_3(x_i, y_j, z_{k-\frac{1}{2}})}{h_3}.
\]
The crucial identity
\begin{equation}
\nabla_h \cdot \nabla_h \times \equiv 0
\end{equation}

follows from straight forward calculation. The resulting generalized eigenvalue problem can be written as
\begin{equation}
\nabla_h \times \nabla_h^* \times \mathbf{E} = \lambda \varepsilon_h \mathbf{E}, \quad \mathbf{E} \in L^2_{p,k}(E)
\end{equation}

where \( \varepsilon_h : E \mapsto \mathbb{R} \) is a numerical approximation of \( \varepsilon \). Here in (2.25), both \( \nabla_h \times \) and \( \nabla_h^* \times \) are discrete operators, but acting on different spaces. They are both discrete approximations of the curl operator \( \nabla \times \) and are adjoint to each other. See Lemma 2.1 for details.

For \( \mathbf{E}, \mathbf{U} \in L^2(E) \) and \( \mathbf{H}, \mathbf{V} \in L^2(J) \), we denote by
\begin{equation}
\langle \mathbf{E}, \mathbf{U} \rangle_E = h_1 h_2 h_3 \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \sum_{k=1}^{N_3} \left\{ (E_1 U_1)_{i,j,k} + (E_2 U_2)_{i,j,k} + (E_3 U_3)_{i,j,k} \right\},
\end{equation}

and
\begin{equation}
\langle \mathbf{H}, \mathbf{V} \rangle_J = h_1 h_2 h_3 \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \sum_{k=1}^{N_3} \left\{ (H_1 V_1)_{i,j,k} + (H_2 V_2)_{i,j,k} + (H_3 V_3)_{i,j,k} \right\},
\end{equation}

the standard inner products on \( L^2(E) \) and \( L^2(J) \) over the unit cell respectively. The following observation explains the notations used for the two discrete curl operators in (2.18, 2.20) and is crucial to the development of our scheme. The proof follows from straight forward calculation and is omitted.

**Lemma 2.1.** If \( \mathbf{U}, \mathbf{V} \in L^2_{p,k}(E) \), then
\begin{equation}
\langle \mathbf{U}, \nabla_h \times \nabla_h^* \times \mathbf{V} \rangle_E = \langle \nabla_h \times \mathbf{U}, \nabla_h^* \times \mathbf{V} \rangle_J.
\end{equation}

From Lemma 2.1, the operator \( \nabla_h \times \nabla_h^* \times \) is self adjoint on \( L^2_{p,k}(E) \) and non-negative. In addition, the operator \( \varepsilon_h \) is positive definite. It follows that the eigenvalues in (2.25) are all real and non-negative. The eigenvectors \( \mathbf{V}_j \) form a basis which are orthogonal with respect to the following inner product on \( L^2_{p,k}(E) \) induced by \( \varepsilon_h \):
\begin{equation}
\langle \mathbf{E}, \mathbf{U} \rangle_{\varepsilon_h} = h_1 h_2 h_3 \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \sum_{k=1}^{N_3} \left\{ (E_1 U_1)_{i,j,k} \varepsilon_h + (E_2 U_2)_{i,j,k} \varepsilon_h + (E_3 U_3)_{i,j,k} \varepsilon_h \right\}.
\end{equation}

We therefore have the following eigen decomposition
\begin{equation}
L^2_{p,k}(E) = \mathcal{N}_{p,k} \oplus \varepsilon_h \mathcal{N}_{p,k}^{\perp \varepsilon_h}
\end{equation}

where
\begin{equation}
\mathcal{N}_{p,k} = \ker (\nabla_h \times \nabla_h^* \times) = \ker (\nabla_h^* \times) = \{ \mathbf{V} \in L^2_{p,k}(E), \nabla_h \times \mathbf{V} = \mathbf{0} \}
\end{equation}

and
\begin{equation}
\mathcal{N}_{p,k}^{\perp \varepsilon_h} = \{ \mathbf{U} \in L^2_{p,k}(E) | \langle \mathbf{U}, \mathbf{V} \rangle_{\varepsilon_h} = 0, \forall \mathbf{V} \in \mathcal{N}_{p,k} \} = \text{Span}\{ \mathbf{V}_j \in L^2_{p,k}(E) | \nabla_h \times \nabla_h^* \times \mathbf{V}_j = \lambda_j \varepsilon_h \mathbf{V}_j, \lambda_j > 0 \}.
\end{equation}

The following main Theorem is the foundation of our null space free algorithm.
\textbf{Theorem 2.2.} \( U \in \mathcal{N}_{p,k}^{l+\varepsilon h} \) if and only if \( \varepsilon_h U = \nabla_h \times \hat{U} \) for some discrete vector potential \( \hat{U} \in L^2_{p,k}(\mathcal{F}). \)

\textit{Proof.} Denote by \((\lambda_j, V_j)\) the eigen-pairs of the generalized eigenvalue problem (2.25).

Suppose that \( U \in \mathcal{N}_{p,k}^{l+\varepsilon h} \), then there exist constants \( a_j \) such that \( U = \sum_{\lambda_j > 0} a_j V_j \). Thus

\begin{equation}
(2.33) \varepsilon_h U = \varepsilon_h \sum_{\lambda_j > 0} a_j V_j = \sum_{\lambda_j > 0} a_j \nabla_h \times \nabla_h \times (\lambda_j^{-1} V_j) = \nabla_h \times (\sum_{\lambda_j > 0} a_j \nabla_h \times (\lambda_j^{-1} V_j)) = \nabla_h \times \hat{U}
\end{equation}

where \( \hat{U} = \sum_{\lambda_j > 0} a_j \nabla_h \times (\lambda_j^{-1} V_j) \in L^2_{p,k}(\mathcal{F}). \)

Conversely, we suppose that \( \varepsilon_h U = \nabla_h \times \hat{U} \) for some \( \hat{U} \in L^2_{p,k}(\mathcal{F}) \). Then for any \( V \in \mathcal{N}_{p,k} \), we have

\begin{equation}
(2.34) \langle U, V \rangle_{\varepsilon_h} = \langle \varepsilon_h U, V \rangle_{\varepsilon} = \langle \nabla_h \times \hat{U}, V \rangle_{\varepsilon} = \langle \hat{U}, \nabla_h \times V \rangle_{\varepsilon} = 0.
\end{equation}

Thus \( U \in \mathcal{N}_{p,k}^{l+\varepsilon h} \), completing the proof. \( \square \)

\textbf{Remark 1.} From (1.8), it is easy to see that if \( U \in \mathcal{N}_{p,k}^{l+\varepsilon h} \) then \( \nabla_h \cdot (\varepsilon_h U) = 0 \) on \( \mathcal{V}. \) Therefore Theorem 2.2 can be viewed as a weaker version of the discrete analogue of Poincaré Lemma. Nevertheless, it is sufficient for development of our scheme.

For practical purposes, it is sometimes convenient to work on the space \( L^2_p(\mathcal{E}) \) with standard periodic condition. In view of (2.3), (2.25) translates to

\begin{equation}
(2.35) e^{-ik \cdot \hat{\nabla}_h} \nabla_h \times \nabla_h^* e \times e^{-ik \cdot \hat{\nabla}_h} e = \lambda \varepsilon_h e, \quad e \in L^2_p(\mathcal{E}),
\end{equation}

while (2.28) now becomes

\begin{equation}
(2.36) \langle u, e^{-ik \cdot \hat{\nabla}_h} \nabla_h \times \nabla_h^* u \times e^{-ik \cdot \hat{\nabla}_h} u \rangle_{\varepsilon} = \langle \nabla_h^* \times e^{-ik \cdot \hat{\nabla}_h} u, \nabla_h^* \times e^{-ik \cdot \hat{\nabla}_h} u \rangle_{\varepsilon}, \quad u, v \in L^2_p(\mathcal{E}).
\end{equation}

The counter part of Theorem 2.2 is given by

\textbf{Theorem 2.3.} Denote by \( \mathcal{N}_{p}^{l+\varepsilon h} = \text{Span}\{v_j \in L^2_p(\mathcal{E}) | e^{-ik \cdot \hat{\nabla}_h} \nabla_h \times \nabla_h^* v_j = \lambda_j \varepsilon_h v_j, \lambda_j > 0\}. \)

Then \( u \in \mathcal{N}_{p}^{l+\varepsilon h} \) if and only if \( \varepsilon_h e^{-ik \cdot \hat{\nabla}_h} u = \nabla_h \times (e^{-ik \cdot \hat{\nabla}_h} u) \) for some discrete vector potential \( \hat{u} \in L^2_{p}(\mathcal{F}). \)

With standard ordering on components of \( L^2(\mathcal{E}) \) and \( L^2(\mathcal{F}) \) within the primitive cell, the matrix representation of the discrete curl operator \( \nabla_h \times : L^2_{p,k}(\mathcal{E}) \mapsto L^2_{p,k}(\mathcal{F}) \) is given by

\[
\mathbb{C} = \begin{bmatrix}
0 & -1/k_3 & \frac{1}{k_1} \text{diag}(k_2, \cdots, k_2) \\
-1/k_3 & 0 & \frac{1}{k_1} \text{diag}(k_2, \cdots, k_1) \\
\frac{1}{k_2} \text{diag}(k_3, \cdots, k_3) & \frac{1}{k_1} \text{diag}(k_2, \cdots, k_1) & 0
\end{bmatrix} \in \mathbb{C}^{3N \times 3N},
\]
where

\begin{align}
(2.37a) \quad K_1 &= \begin{bmatrix}
1 & -e^{-ik_1} \\
-1 & 1 \\
\vdots & \vdots \\
-1 & 1
\end{bmatrix} \in \mathbb{C}^{N_1 \times N_1}, \\
(2.37b) \quad K_2 &= \begin{bmatrix}
\mathbb{I}_{N_1} & -e^{-ik_2} \mathbb{II}_{N_1} \\
-\mathbb{I}_{N_1} & \mathbb{II}_{N_1} \\
\vdots & \vdots \\
-\mathbb{I}_{N_1} & \mathbb{II}_{N_1}
\end{bmatrix} \in \mathbb{C}^{(N_1, N_2) \times (N_1, N_2)}, \\
(2.37c) \quad K_3 &= \begin{bmatrix}
\mathbb{I}_{N_1} & -e^{-ik_3} \mathbb{II}_{N_1} \\
-\mathbb{I}_{N_1} \mathbb{II}_{N_1} & \mathbb{II}_{N_1} \mathbb{II}_{N_1}
\end{bmatrix} \in \mathbb{C}^{N \times N},
\end{align}

where $N = N_1N_2N_3$.

The matrix representation of the operator $e^{-ik \times \nabla_h} \times \nabla_h^* \times e^{ik \times \cdot} : L^2_p(\mathcal{E}) \mapsto L^2_p(\mathcal{E})$ is therefore given by

\begin{equation}
(2.38) \quad A = \mathbb{D}^* \mathbb{C}^* \mathbb{D},
\end{equation}

where the matrix $\mathbb{D}$, as well as the mass matrix $\mathbb{B}$ are both diagonal matrices representing multiplication by $e^{ik \cdot x}$ and $\varepsilon(x)$, respectively for $x \in \mathcal{E}$.

Before we proceed, we briefly summarize our notations for reader’s convenience.

(a) The upper/lowercase boldface characters denote elements of $L^2_p(\mathcal{E})/L^2_p(\mathcal{E})$ respectively. For example, $\mathbf{U} \in L^2_p(\mathcal{E})$ and $\mathbf{u} = e^{-ik \cdot \mathbf{x}} \mathbf{U} \in L^2_p(\mathcal{E})$.

(b) A hatted upper/lowercase boldface character denotes a discrete vector potential. For example, \( \hat{\mathbf{V}} \in L^2_p(\mathcal{F}) \) is a vector potential of $\mathbf{V} \in L^2_p(\mathcal{E})$: \( \nabla_h \times \hat{\mathbf{V}} = \varepsilon_h \mathbf{V} \). Similarly, $\hat{\mathbf{v}} \in L^2_p(\mathcal{F})$ is related to $\mathbf{v} \in L^2_p(\mathcal{E})$ by $\nabla_h \times (e^{ik \cdot \mathbf{x}} \mathbf{v}) = \varepsilon_h e^{ik \cdot \mathbf{x}} \mathbf{v}$.

(c) A vector symbol over lowercase boldface character, such as $\mathbf{t}$ or $\mathbf{T}$, denotes a $3N^3 \times 1$ column vector representation of the three components of $\mathbf{t} \in L^2_p(\mathcal{E})$ or $\mathbf{T} \in L^2_p(\mathcal{F})$, each being an $N \times N \times N$ array.

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

3. Null Space Free Jacobi-Davidson Iteration. In this section, we describe our algorithm which is a modified Jacobi-Davidson method specifically tailored for the eigenvalue problem (2.25) to handle the huge spurious null space issue. By an appropriate application of Theorem 2.2, our algorithm guarantees the approximating subspace to stay perpendicular to the null space during the iteration. Numerical evidence confirms that our approach yields better performance against original Jacobi-Davidson method. We start this section by reviewing the original Jacobi-Davidson method.

3.1. Jacobi-Davidson Method. The Jacobi-Davidson method [18] 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 [7, 8, 9, 19].

When applied to the generalized eigenvalue problem (2.35), the Jacobi-Davidson method consists of the following steps:

1. To compute $i$th eigen-pair $(\lambda_i, \mathbf{E}_i)$, one initializes a subspace $\mathcal{V}_i := \text{Span}\{\mathbf{V}_1, \cdots, \mathbf{V}_i\} \subset L^2_p(\mathcal{E})$ with $\|\mathbf{V}_1\|_{\varepsilon_h} = \cdots = \|\mathbf{V}_i\|_{\varepsilon_h} = 1$, where $\mathbf{V}_1 = \mathbf{E}_1, \cdots, \mathbf{V}_{i-1} = \mathbf{E}_{i-1}, i \geq 2$, are previously computed eigenvectors corresponding to eigenvalues $\lambda_1, \cdots, \lambda_{i-1}$. 

2. For $k = 1, 2, 3, \ldots$, do
   (i) Find $\theta \in \mathbb{R} \setminus \{\lambda_1, \cdots, \lambda_{i-1}\}$ nearest to the target value $\theta^*$ and $U \in V_k$, with $\|U\|_{\varepsilon_h} = 1$
   such that
   \begin{equation}
   \langle (\nabla_h \times \nabla_h^\times - \theta \varepsilon_h)U, V \rangle_{\varepsilon} = 0, \forall V \in V_k.
   \end{equation}
   Denote by $(\theta_k, U_k)$ the solution to (3.1).
   (ii) Set
   \begin{equation}
   R_k := (\nabla_h \times \nabla_h^\times - \theta_k \varepsilon_h)U_k.
   \end{equation}
   (iii) If $\|R_k\|_{\varepsilon} < \text{tol}$, then $(\theta_k, U_k)$ is an eigen-pair. Set $(\lambda_i, E_i) = (\theta_k, U_k)$. Stop.
   Else, solve approximately for $T$ from
   \begin{equation}
   (I - \varepsilon_h U_k \otimes U_k)(\nabla_h \times \nabla_h^\times - \theta_k \varepsilon_h)(I - U_k \otimes (\varepsilon_h U_k))T = -R_k
   \end{equation}
   with $T \perp_{\varepsilon_k} U_k$, where $(I - U \otimes V)W := W - (V, W)_{\varepsilon} U$.
   (iv) Orthonormalize $T$ against $V_k$ with respect to the inner product (2.29) to get $V_{i+k}$. Expand
   $V_{k+1} := \text{Span}\{V_k, V_{i+k}\}$.

The implementation detail is summarized as Algorithm 1 in matrix-vector notations:

```
Set $E_0 = [\ ]$, $A_0 = \varnothing$.
for $i = 1, 2, 3, \ldots, i_{\text{max}}$ do
   Initialize a vector $\vec{v}_i$ with $\|\vec{v}_i\|_B = 1$.
   Set $V_i = [E_{i-1}, \vec{v}_i]$.
   Compute $H_i = V_i^* A V_i$.
   for $k = 1, 2, 3, \ldots$ do
      (i) Compute all the Ritz-pairs of $(H_k - \theta_k I)\vec{s} = 0$.
      Select the desired Ritz value $\theta_k \notin \Lambda_{i-1}$ nearest to the target eigenvalue $\theta^*$
      and let $\vec{s}_k$ be the associated eigenvector with $\|\vec{s}_k\|_2 = 1$.
      (ii) Compute $\vec{u}_k = V_k \vec{s}_k$, $r_k = (A - \theta_k B)\vec{u}_k$.
      (iii) if $\|r_k\|_2 < \text{tol}$ then
         Set $\lambda_i = \theta_k$, $E_i = \vec{u}_k$.
         Update $E_i = [E_{i-1}, \vec{u}_i]$, $\Lambda_i = \Lambda_{i-1} \cup \{\lambda_i\}$.
         Exit $k$.
      else
         Solve (approximately)
         \begin{equation}
         (I - B \vec{u}_k \vec{u}_k^*) (A - \theta_k B) (I - \vec{u}_k \vec{u}_k^*) \vec{t} = -\vec{r}_k
         \end{equation}
         with $\vec{t} \perp_B \vec{u}_k$.
         $B$-orthonormalize $\vec{t}$ against $V_k$: $\vec{v}_{i+k} = \frac{\vec{t} - \sum_{\ell=1}^k (\vec{u}_\ell^* B) \vec{v}_\ell}{\|\vec{t} - \sum_{\ell=1}^k (\vec{u}_\ell^* B) \vec{v}_\ell\|_B}$.
      endif
   end for $k$
end for $i$
```

**Algorithm 1:** Jacobi-Davidson Method.

The main computational cost is in step (iii) of Algorithm 1, where the correction equation
\begin{equation}
(I - B \vec{u}_k \vec{u}_k^*) (A - \theta_k B) (I - \vec{u}_k \vec{u}_k^*) \vec{t} = -\vec{r}_k, \quad \vec{t} \perp_B \vec{u}_k
\end{equation}
is solved by standard iterative methods such as GMRES with a pre-conditioner
\[ M_p = (\mathbb{I} - B \hat{u}_k \hat{u}_k^*) M (\mathbb{I} - \bar{u}_k \bar{u}_k^*) \approx (\mathbb{I} - B \hat{u}_k \hat{u}_k^*) (A - \theta_k \mathbb{B}) (\mathbb{I} - \bar{u}_k \bar{u}_k^*) , \]
where \( \mathbb{M} \) is a pre-conditioner for \( (A - \theta_k \mathbb{B}) \). In \( j \)th GMRES iteration, one solves for \( z^{(j)} \) from
\[ M_p z^{(j)} = y^{(j)}, \quad z^{(j)} \perp \bar{u}_k \]
The solution to (3.6) is given by
\[ z^{(j)} = \mathbb{M}^{-1} y^{(j)} - \zeta^{(j)} \mathbb{M}^{-1} B \hat{u}_k, \]
with
\[ \zeta^{(j)} = \frac{\hat{u}_k B \mathbb{M}^{-1} y^{(j)}}{\hat{u}_k B \mathbb{M}^{-1} B \hat{u}_k}. \]

Remark 2. The correction equation (3.3) is equivalent to
\[ \left( \nabla_h \times \nabla_h^* - \theta_h \mathbb{U}_k \right) T = -R_k + \eta \mathbb{U}_k = -\left( \nabla_h \times \nabla_h^* - \theta_h \mathbb{U}_k \right) U_k + \eta \mathbb{U}_k \]
where \( \eta = \frac{\mathbb{E}_h U_k}{\mathbb{E}_h \nabla_h \times \nabla_h^* - \theta_h \mathbb{U}_k} \). If (3.3) were solved accurately, then one would have \( T = \frac{1}{\eta} \mathbb{E}_h \nabla_h \times \nabla_h^* \left( \mathbb{E}_h \mathbb{U}_k \right) \). In other words, the corrector \( T \) 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 original Jacobi-Davidson method (3.5)-(3.8). Under such circumstances, null vectors are inevitably introduced into \( \mathcal{V}_{k+1} \) to deteriorate the overall performance. In the following subsection, we will describe our version of Jacobi-Davidson method that is automatically null space free.

3.2. Null Space Free Jacobi-Davidson Method. The major advantage of the Jacobi-Davidson method for general eigenvalue problem is the correction equation (3.4) only needs to be solved approximately. This advantage becomes a drawback for Algorithm 1 as the corrector \( \hat{t} \) inevitably contains components in the spurious null space, causing slow convergence.

Our remedy to this problem is to initialize and expand the approximating subspace with the help of the discrete vector potential given in Theorem 2.2. Instead of solving the correction equation (3.4) directly, we lift the correction equation and the approximate solver (3.7, 3.8) to the vector potential level. The vector potential \( \hat{V} \) only needs to be solved approximately as original Jacobi-Davidson method.

An approximate corrector \( \hat{V} \) is then obtained by taking the discrete curl of the approximate vector potential \( \hat{V} \).

This procedure can be illustrated, for example, by the evaluation of \( \mathbb{M}^{-1} B \hat{u}_k \) in (3.7), or equivalently by getting an approximate solution of
\[ \left( \nabla_h \times \nabla_h^* - \theta_h \mathbb{U}_k \right) Q = \mathbb{E}_h \mathbb{U}_k. \]
By construction (or induction on \( i \)), see Algorithm 2 for details), \( \mathbb{E}_h U_k \in \mathcal{N}_{p,k+1} \). Therefore from Theorem 2.2, we have
\[ \mathbb{E}_h U_k = \nabla_h \times \hat{U}_k. \]
One seeks solution to (3.10) of the form
\[ \mathbb{E}_h Q = \nabla_h \times \hat{Q}. \]
Substituting (3.11, 3.12) into (3.10), one arrives at
\begin{equation}
\nabla_{\text{h}} \times (\nabla_{\text{h}}^* \times \varepsilon_{\text{h}}^{-1} \nabla_{\text{h}} \times -\theta k \mathbb{I}) \hat{Q} = \nabla_{\text{h}} \times \hat{U}_{k}.
\end{equation}

Instead of solving (3.10) directly, we propose to solve for the vector potential
\begin{equation}
(\nabla_{\text{h}}^* \times \varepsilon_{\text{h}}^{-1} \nabla_{\text{h}} \times -\theta k \mathbb{I}) \hat{Q} = \hat{U}_{k},
\end{equation}
then denote the approximate solution by \( \hat{Q}_1 \), and take \( Q_1 = \varepsilon_{\text{h}}^{-1} \nabla_{\text{h}} \times \hat{Q}_1 \) as an approximate solution of (3.10) which lies in \( \mathcal{N}_{p,k}^{1+\text{h}} \) automatically.

The same procedure can be used to derive the equation for the vector potential from the correction equation (3.3). From Theorem 2.2 and equation (3.2), we have
\begin{equation}
R_{k} = \nabla_{\text{h}} \times \hat{R}_{k},
\end{equation}
where
\begin{equation}
\hat{R}_{k} = (\nabla_{\text{h}}^* \times \varepsilon_{\text{h}}^{-1} \nabla_{\text{h}} \times -\theta k \mathbb{I}) \hat{U}_{k} \in L_{p,k}^{2}(\mathcal{F}).
\end{equation}

Following the same procedure as in (3.10-3.13), together with the ansatz
\begin{equation}
T = \varepsilon_{\text{h}}^{-1} \nabla_{\text{h}} \times \hat{T},
\end{equation}
we can recast the left hand side of (3.3) as
\begin{equation}
\nabla_{\text{h}} \times (I - \hat{U}_{k} \otimes (\nabla_{\text{h}}^* \times \varepsilon_{\text{h}}^{-1} \nabla_{\text{h}} \times \hat{U}_{k}))(\nabla_{\text{h}}^* \times \varepsilon_{\text{h}}^{-1} \nabla_{\text{h}} \times -\theta k \mathbb{I})(I - \hat{U}_{k} \otimes (\nabla_{\text{h}}^* \times \varepsilon_{\text{h}}^{-1} \nabla_{\text{h}} \times \hat{U}_{k})) \hat{T}
\end{equation}
An approximate solution to (3.3) is therefore obtained by solving
\begin{equation}
(I - \hat{U}_{k} \otimes (\nabla_{\text{h}}^* \times \varepsilon_{\text{h}}^{-1} \nabla_{\text{h}} \times \hat{U}_{k}))(\nabla_{\text{h}}^* \times \varepsilon_{\text{h}}^{-1} \nabla_{\text{h}} \times -\theta k \mathbb{I})(I - \hat{U}_{k} \otimes (\nabla_{\text{h}}^* \times \varepsilon_{\text{h}}^{-1} \nabla_{\text{h}} \times \hat{U}_{k})) \hat{T} = -\hat{R}_{k}
\end{equation}
and applying the transformation (3.17) to the approximate solution of (3.19).

Equation (3.19) remains the core of the algorithm. In matrix notations, it reads
\begin{equation}
\begin{pmatrix}
I - \hat{u}_k \hat{u}^*_k \hat{a}_e \\
\hat{a}_e - \theta k \mathbb{I}
\end{pmatrix}
\begin{pmatrix}
\hat{t} \\\n\hat{u}_k
\end{pmatrix}
= -\begin{pmatrix}
\hat{a}_e - \theta k \mathbb{I} \hat{u}_k
\end{pmatrix}
\begin{pmatrix}
\hat{t} \\\n\hat{u}_k
\end{pmatrix}
= -\hat{r}_k.
\end{equation}

where
\begin{equation}
\hat{a}_e = D^*C^*B^{-1}CD.
\end{equation}
Alternatively, (3.20) can be derived from (3.4) with
\begin{equation}
\hat{t} = D^*B^{-1}CD \hat{t}, \quad \hat{u}_k = D^*B^{-1}CD \hat{u}_k, \quad \hat{r}_k = D^*B^{-1}CD \hat{r}_k.
\end{equation}

Note that if \( \hat{u} = D^*B^{-1}CD \hat{u} \) and \( \hat{v} = D^*B^{-1}CD \hat{v} \), then
\begin{equation}
\hat{u}^* \hat{v} = \hat{u}^* \hat{a}_e \hat{v}
\end{equation}
and it is convenient to introduce the notation
\begin{equation}
|\hat{v}|_{\hat{a}_e} := (\hat{v}^* \hat{a}_e \hat{v})^{1/2}.
\end{equation}
The solution procedure for (3.20) remains the same. One solves it iteratively with preconditioner
\begin{equation}
\tilde{M}_p \equiv \left( 1 - \tilde{u}_k \tilde{u}_k^* \tilde{\Lambda}_z \right) \tilde{M} \left( 1 - \tilde{u}_k \tilde{u}_k^* \tilde{\Lambda}_z \right)
\end{equation}
where $\tilde{M}$ is an preconditioner for $(\tilde{\Lambda}_z - \theta \mathbb{I})$. The condition $\tilde{t} \perp_{\tilde{\Lambda}_z} \tilde{u}_k$ now translates to $\tilde{t} \perp_{\tilde{\Lambda}_z} \tilde{u}_k$.

In $j$th GMRES iteration, one solves for
\begin{equation}
\tilde{M}_{p(j)^*} \tilde{z}^{(j)} = \tilde{y}^{(j)}, \quad \tilde{u}_k^* \tilde{\Lambda}_z \tilde{z}^{(j)} = 0.
\end{equation}
A solution to (3.26) is therefore given by
\begin{equation}
\tilde{z}^{(j)} = \tilde{M}^{-1} \tilde{y}^{(j)} - \tilde{\zeta}^{(j)} \tilde{M}^{-1} \tilde{\varepsilon}_k^{(j)}.
\end{equation}
where
\begin{equation}
\tilde{\zeta}^{(j)} = \frac{\tilde{u}_k^* \tilde{\Lambda}_z \tilde{M}^{-1} \tilde{y}^{(j)}}{\tilde{u}_k^* \tilde{\Lambda}_z \tilde{M}^{-1} \tilde{\varepsilon}_k^{(j)}}.
\end{equation}

We summarize the null space free version as Algorithm 2. Note that the $\mathbb{B}$-orthonormalization for the corrector in step (iii) of Algorithm 1, or
\begin{equation}
\tilde{v}_{k+1} = \frac{\tilde{t} - \sum_{\ell=1}^{k} (\tilde{v}_{\ell}^* \mathbb{B} \tilde{t}) \tilde{v}_{\ell}}{\| \tilde{t} - \sum_{\ell=1}^{k} (\tilde{v}_{\ell}^* \mathbb{B} \tilde{t}) \tilde{v}_{\ell} \|}_{\mathbb{B}}
\end{equation}
can be recast to
\begin{equation}
\mathbb{D}^* \mathbb{B}^{-1} \mathbb{C} \tilde{v}_{k+1} = \mathbb{D}^* \mathbb{B}^{-1} \mathbb{C} \left( \frac{\tilde{t} - \sum_{\ell=1}^{k} (\tilde{v}_{\ell}^* \tilde{\Lambda}_z \tilde{t}) \tilde{v}_{\ell}}{\| \tilde{t} - \sum_{\ell=1}^{k} (\tilde{v}_{\ell}^* \tilde{\Lambda}_z \tilde{t}) \tilde{v}_{\ell} \|} \right),
\end{equation}
hence the $\tilde{\Lambda}_z$-orthonormalization in step (iii) of Algorithm 2.

Remark 3. Although our scheme is tested for Yee’s discretization only. Similar performance gain (see Section 4.3) should be expected for other spatially compatible discretizations provided a suitable preconditioner is available. We believe that the difficulty in preconditioning the matrices $\mathbb{A} - \theta \mathbb{B} = \mathbb{D}^* \mathbb{C}^* \mathbb{D} - \theta \mathbb{B}$ and $\tilde{\Lambda}_z - \theta \mathbb{I} = \mathbb{D}^* \mathbb{C}^* \mathbb{D}^{-1} \mathbb{C} \mathbb{D} - \theta \mathbb{I}$ are comparable. For Yee’s discretization, the FFT serves as an efficient preconditioner for both operators. Finding suitable preconditioners for other spatially compatible discretizations is a topic currently under investigation. The result will be reported elsewhere if successful.

4. Numerical results. The key to success of Jacobi-Davidson method is an efficient preconditioner for the correction equation. In some other applications, such as the quantum dot models [7, 8, 9, 19], the underlying discrete operator results from a second order elliptic system and corresponds to a diagonal dominant matrix problem. Under such circumstances, the preconditioning strategy is quite simple. The combination of SSOR as preconditioner, together with GMRES or Bi-CGSTAB as linear solver proves to be successful. In the current situation, the matrices $\mathbb{A}$ and $\tilde{\Lambda}_z$ are both degenerate and far from diagonal dominant. A simple preconditioning strategy such as SSOR may not work. Instead, the Fast Fourier Transform (FFT) turns out to be an efficient preconditioner for both (3.4) and (3.20). In other words, we take $\mathbb{M} = (\mathbb{D}^* \mathbb{C}^* \mathbb{D} - \theta \mathbb{I})$ and $\tilde{\mathbb{M}}_z = (\varepsilon^{-1} \mathbb{C}^* \mathbb{C} - \theta \mathbb{I})$ where $\mathbb{C}$ and $\varepsilon^{-1}$ are spatial averages of $\varepsilon$ and $\varepsilon^{-1}$, respectively. Both (3.6) and (3.26) can then be solved efficiently using FFT.
Set $\hat{E}_0 = [ ]$, $A_0 = \emptyset$.

for $i = 1, 2, 3, \ldots, i_{\text{max}}$ do

Initialize a vector potential $\hat{\vartheta}_i$ with $|\hat{\vartheta}_i|_{\hat{A}_e} = 1$.

Set $\hat{\vartheta}_1 = [ \hat{E}_{i-1}, \hat{\vartheta}_i ]$.

Compute $\hat{H}_1 = (D^*B^{-1}CD\hat{\vartheta}_1)^*A(D^*B^{-1}CD\hat{\vartheta}_1) = \hat{\vartheta}_1^*\hat{A}_e^*\hat{\vartheta}_1$.

for $k = 1, 2, 3, \ldots$ do

(i) Compute all the Ritz-pairs of $((\hat{H}_k - \hat{\vartheta}_k)\hat{s} = \hat{\vartheta}_k$.

Select the desired Ritz value $\theta_k \notin \Lambda_{i-1}$ nearest to the target eigenvalue $\theta^*$

and let $\hat{s}_k$ be the associated eigenvector with $|\hat{s}_k|_{\hat{A}_e} = 1$.

(ii) Compute $\hat{u}_k = \hat{\vartheta}_k \hat{s}_k$, $\hat{r}_k = (\hat{A}_e - \theta_k \hat{I})\hat{u}_k$.

(iii) if $(||\hat{r}_k||_2 < \text{tol})$ then

Set $\lambda_i = \theta_k$ and $\hat{e}_i = D^*B^{-1}CD\hat{u}_k$.

Update $\hat{E}_i = [ \hat{E}_{i-1}, \hat{u}_k ]$, $A_i = A_{i-1} \cup \{ \lambda_i \}$.

Exit $k$.

else

Solve (approximately)

\[ (\hat{I} - \hat{u}_k \hat{A}_e \hat{e}_k) (\hat{A}_e - \theta_k \hat{I}) (\hat{I} - \hat{u}_k \hat{A}_e \hat{e}_k)^T \hat{t} = -\hat{r}_k \]

with $\hat{t} \perp \hat{A}_e \hat{u}_k$.

$\hat{A}_e$-orthonormalize $\hat{t}$ against $\hat{\vartheta}_k$: $\hat{v}_{k+1} = \hat{v}_{k+1} - \frac{\hat{v}_{k+1}^* \hat{A}_e^* \hat{t}}{1 - \sum_{\ell=1}^{k} \hat{v}_{\ell}^* \hat{A}_e \hat{v}_{\ell}}$.

endif

(iv) Expand $\hat{\vartheta}_{k+1} = [ \hat{\vartheta}_k, \hat{\vartheta}_{i+k} ]$, $\hat{H}_{k+1} = \begin{bmatrix} \hat{H}_k & \hat{H}_k^* \hat{A}_e^* \hat{\vartheta}_{i+k} \\ \hat{\vartheta}_{i+k}^* \hat{A}_e^* \hat{H}_k & \hat{\vartheta}_{i+k}^* \hat{A}_e^* \hat{\vartheta}_{i+k} \end{bmatrix}$.

end for $k$
end for $i$

Algorithm 2: Null space free Jacobi-Davidson (NFJD) method

Our numerical tests are based on the benchmark example shown in Figure 4.1 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
$\varepsilon_i/\varepsilon_o = 13$.

To illustrate the accuracy and efficiency of our scheme, we have devised numerical experiments for
both Algorithm 1 and 2 in various setting. In Section 4.1, we first conduct a convergence test for NFJD
by refining the mesh. In Section 4.2, we explore the original Jacobi-Davidson with different strategy
of the target value selection, to investigate the effect of the null space to both algorithms. In Section
4.3, we compare the performance of original Jacobi-Davidson method against NFJD to demonstrate the
efficiency of NFJD.

All numerical tests are performed on a HP rx2600 workstation equipped with 1.6 GHz Intel Itanium
II processor.

Since the residuals for Algorithm 1 and 2 are related by $\hat{r}_k = D^*B^{-1}CD\hat{r}_k$ (or equivalently by
(3.15)), we expect that, in the worst scenario, $\hat{r}_k = O(h^{-1})\hat{r}_k$. Accordingly, we set the stopping
tolerance to $5 \times 10^{-8}$ for Algorithms 1 and $10^{-10}$ for Algorithm 2, respectively. Generically, the
amplification factor from $||\hat{r}_k||_2$ to $||\hat{r}_k||_2$ is significantly lower than expected.
4.1. Convergence Test. Figure 4.2 shows the plot of $\omega = a \sqrt{\lambda} / (2\pi)$ Vs. representative points $k$ in the first Brillouin zone for the benchmark problem computed using Algorithm 2 with target value $\theta^* = 0$ and $N_1 = N_2 = N_3 = 100$. The smallest nonzero eigenvalues are calculated for 40 sample points $k$ distributed along the segments connecting $\Gamma = (0, 0, 0)$, $X = (\pi, 0, 0)$, $M = (\pi, \pi, 0)$, $R = (\pi, \pi, \pi)$ and back to $\Gamma$ in the first Brillouin zone. A clear band gap lies between the 5-th and 6-th smallest positive eigenvalues.

The result with various grid resolutions is summarized in Table 4.1, where $\omega_{\text{low}}$ denotes the maximum of the 5-th eigenvalue, $\omega_{\text{up}}$ the minimum of the 6-th eigenvalue and

$$\gamma_{gm} = \frac{\omega_{\text{up}} - \omega_{\text{low}}}{(\omega_{\text{up}} + \omega_{\text{low}})/2}$$

The result shows clear convergence and agrees well with those reported in the literature [3, 5].

<table>
<thead>
<tr>
<th>grids</th>
<th>$50 \times 50 \times 50$</th>
<th>$100 \times 100 \times 100$</th>
<th>$200 \times 200 \times 200$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\omega_{\text{low}}$</td>
<td>0.41785</td>
<td>0.41789</td>
<td>0.41782</td>
</tr>
<tr>
<td>$\omega_{\text{up}}$</td>
<td>0.48023</td>
<td>0.48079</td>
<td>0.48096</td>
</tr>
<tr>
<td>$\gamma_{gm}$</td>
<td>0.1389</td>
<td>0.1400</td>
<td>0.1405</td>
</tr>
</tbody>
</table>
4.2. Effect of the Null Space on Original Jacobi-Davidson Method. In this subsection, we study the effect of the null space on original Jacobi-Davidson method, Algorithm 1. A good indication on the effect of the null space is how often and how far the computed Ritz value $\theta_k$ wanders away towards zero. We describe below two different strategies of choosing the target value $\theta^*$ and investigate the effect of the null space separately.

The smallest positive eigenvalue in our simulation occurs at $k = (0.1\pi, 0, 0)$ with $\lambda_1 \sim 0.045$. Accordingly, our first targeting strategy is to take fixed $\theta^*_f = 0.01$ and will be referred to as fixed-target Jacobi-Davidson method, or “JD$_f$”. An alternative approach is to take dynamic target values during the iteration. If $(i - 1)$th smallest eigenvalue $\lambda_{i-1}$ has been obtained, we take $\theta^*_d = 0.9\lambda_{i-1}$ during the computation of $\lambda_i$, $i \geq 2$. This dynamic targeting strategy will be referred to as “JD$_d$”.

Figure 4.3 shows the convergence history of the Ritz value and the residual for the first two eigenvalues at $k = (\pi, 0.6\pi, 0)$ computed by JD$_f$. It is clear that $\theta_j$ is constantly dragged towards zero during the subspace iteration. This effect is also reflected in the convergence history of the residual of the computed Ritz vector.

To quantitatively understand the effect of the null space, we further devise the following measurement from the computed Ritz value.

Denote by $\{\theta_{1,i}, \ldots, \theta_{m,i}\}$ the computed Ritz values in step (i) of Algorithm 1, with $\theta_{m,i} \approx \lambda_i$ being the convergent one satisfying the stopping criterion in step (iii).

**Definition 4.1.** A Ritz value $\theta_{k,i}$ is “dragged” if $\theta_{k,i} < 0.5 \theta_{m,i}$.

Denote by $N_{DRV,i}$ the numbers of dragged Ritz values $\theta_{k,i}$. The fraction $\gamma = \sum_{i=1}^{10} N_{DRV,i} / \sum_{i=1}^{10} m_i$ serves as a good measurement of inefficiency of Jacobi-Davidson method caused by the dragging effect of the null space during the computation of the first ten eigenvalues $\lambda_1, \ldots, \lambda_{10}$. The larger $\gamma$ is, the more CPU time is wasted in producing dragged Ritz values.

Figure 4.4 shows the fraction $\gamma$ over sample points in the first Brillouin zone for both JD$_f$ with $\theta^*_f = 0.01$ and JD$_d$ with $\theta^*_d = 0.9\lambda_{i-1}$. The result clearly shows that JD$_f$ produces more dragged Ritz values than JD$_d$ does. This is well expected since $\theta^*_f < \theta^*_d$, therefore JD$_f$ suffers more easily from attraction of the null space. However, it should be noted that, once an eigenvalue $\lambda_{i+1}$ emerges before $\lambda_i$ does, JD$_d$ will then take $\theta^*_d = 0.9\lambda_{i+1}$ and is at higher risk of missing $\lambda_i$ in subsequent computation.
In general, it may be difficult to choose a safe and efficient dynamic target value.

In contrast, the targeting strategy for Algorithm 2 is quite simple. We simply set $\theta^* = 0$ in our simulation and did not find any effect of dragging from the null space. As shown in Figure 4.5, the Ritz value decreases monotonically to $\lambda_1$. The residual shows similar behavior. In fact, the fraction $\gamma$ remains zero even if we raise the threshold for dragged Ritz value to $\theta_{k,i} < 0.99 \theta_{m,i}$. This is a clear indication that NFJD is indeed null space free.

4.3. Performance of Algorithm 2. In this subsection, we continue with the comparison of Algorithm 1 and Algorithm 2 in terms of computational cost. In both algorithms, the main computational cost comes from the FFT preconditioner and the matrix-vector product.

Figure 4.6 compares both JD$_f$ and JD$_d$ against NFJD in terms of numbers of FFT calls and matrix-vector products. NFJD is about three times faster than JD$_d$ and about eight times faster than JD$_f$. The efficiency of NFJD is quite obvious.

5. Conclusion. This article aims at developing an efficient, robust, and accurate numerical scheme for simulating three-dimensional photonic crystals. The novelty of our approach is the delicate combination of Jacobi-Davidson method and the discrete vector potential. 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 Jacobi-Davidson method and filters out the spurious null space.

Fig. 4.3. Figure (a), (b): convergent history of the Ritz value $\theta_k$ for $\lambda_1$ and $\lambda_2$ using Algorithm 1 with $\theta_f^* = 0.01$ at $\mathbf{k} = (\pi, 0.6\pi, 0)$. Figure (c), (d): convergent history of the the residual.
almost cost-free. Numerical evidence has confirmed the efficiency and robustness of the new scheme.

REFERENCES


[16] V. Simoncini. Algebraic formulations for the solution of the nullspace-free eigenvalue problem using the inexact
Fig. 4.6. Comparison of numbers of FFT calls and matrix-vector products between Algorithms 1 and 2.