Fast Eigensolvers for Three Dimensional Lossless Drude Dispersive Metallic Photonic Crystals

Tsung-Ming Huang
Department of Mathematics, National Taiwan Normal University, Taipei 116, Taiwan.
E-mail: min@ntnu.edu.tw.

Han-En Hsieh
Department of Mathematics, National Taiwan University, Taipei, 106, Taiwan.
E-mail: d99221002@ntu.edu.tw.

Wen-Wei Lin
Department of Applied Mathematics, National Chiao Tung University, Hsinchu 300, Taiwan. E-mail: wwlin@math.nctu.edu.tw.

Weichung Wang
Department of Mathematics, National Taiwan University, Taipei, 106, Taiwan.
E-mail: wwang@ntu.edu.tw.

March 23, 2013

Abstract. We develop efficient eigenvalue problem solvers to compute band structure of three-dimensional dispersive metallic photonic crystals. Based on the lossless Drude model, we consider the resulting Maxwell equations that simulate the photonic crystals with face-centered cubic lattice. The discretized Maxwell equations result in large-scale nonlinear eigenvalue problems that have clustered eigenvalues. These problems can be solved by the shift-and-invert Lanczos method, in which the most expensive computational task is a series of linear systems. It is therefore essential to solve these linear systems efficiently to accelerate the Lanczos iterations. Our numerical schemes are established via studies of the matrix properties. By exploring the properties of the coefficient matrices, we propose and analyze several preconditioning schemes to solve the linear systems. Numerical results suggest that the proposed methods can solve the target eigenvalue problems efficiently.

Keywords: Three-dimensional dispersive metallic photonic crystals, face-centered cubic lattice, eigenvalue problem solvers, preconditioner

1. Introduction

Photonic crystals have exhibited interesting electronmagnetic propagation properties that lead to various potential applications [8, 10, 16, 17]. Among various types of photonic crystals materials and periodic lattice geometric structures, dispersive metallic photonic crystals with face-centered cubic (FCC) lattice are of great interest in research and development. One important reason is that their band structures are quite different from the band structures in dielectric photonic crystals

[11, 13, 14, 15, 22]. An essential difference is the cutoff frequency. One photonic crystals example demonstrating high cutoff frequency is the FCC lattice consisting of dielectric spheres embedded in a dispersive background material [4]. Consequently, to compute the band structures of three-dimensional (3D) dispersive metallic photonic crystals with FCC lattice is important. The band structures of 3D dispersive metallic photonic crystals have been calculated by a time domain simulation method [3], frequency domain Korringa-Kohn-Rostocker method [12, 19], an iterative method [15], auxiliary field methods [2, 18], and rational polynomial method [11].

To efficiently compute the band structures of the dispersive metallic photonic crystals by solving the governing Maxwell equations numerically, however, remains a computational challenges. Generally speaking, the numerical challenges are mainly caused by (i) the large-scale nonlinear eigenvalue problem due to 3D discretizations, (ii) the complicated coefficient matrix structure due to the discretization over the FCC lattice, and (iii) the fact that the desired eigenvalues are clustered.

In this article, we tackle these computational challenges by focusing on the 3D lossless Drude model [11] with the FCC lattice for dispersive material. The discrete nonlinear eigenvalue problems due to Yee's scheme can be reformulated as generalized eigenvalue problems. We thus develop efficient numerical methods to solve the generalized eigenvalue problems by taking a new approach. The computational efficiency of our approach is achieved by exploring the matrix properties associated with the structure of the target photonic crystals. As shown in Section 2, our approach is rooted in the explicit formulation of the discrete double curl matrix. As the shift-and-invert Lanczos method is used to solve the generalized eigenvalue problem, the key component within the method is the solutions of a series of linear systems. We propose and analyze several preconditioning schemes to solve these linear systems in Section 3. Numerical efficiency of the proposed schemes are studied and verified in Section 4.

2. The Dispersive Model and the Discrete Eigenvalue Problem

The theory of electromagnetic effects with periodic structures and dispersive isotropic materials in three dimension (3D) can be modelled by the Maxwell equation

$$\nabla \times \nabla \times E = \omega^2 \varepsilon(r, \omega) E. \tag{1}$$

In the equation, the variable E stands for the electric field and ε stands for the permittivity which is depend on the frequency ω of time. Furthermore, we consider the lossless Drude model, which describes the relation between the permittivity and the frequency as

$$\varepsilon_d(r,\omega) = 1 - \frac{\omega_p^2}{\omega^2}, \tag{2}$$

where ω_p is a plasma frequency.

Based on the Bloch Theorem [9], we aim to find the Bloch eigenfunctions E for (1) that satisfy the quasi-periodic condition $E(\mathbf{x}+\mathbf{a}_\ell) = e^{\imath 2\pi \mathbf{k}\cdot\mathbf{a}_\ell} E(\mathbf{x})$, for $\ell = 1,2,3$. Here, $2\pi\mathbf{k}$ is the Bloch wave vector in the first Brillouin zone [8] and $\mathbf{a}_\ell$'s are the lattice translation vectors that span the primitive cell. Such primitive cells extend periodically to form the photonic crystals. In the FCC structure, the lattice vectors are $\mathbf{a}_1 = \frac{a}{\sqrt{2}}[1,0,0]^\top$, $\mathbf{a}_2 = \frac{a}{\sqrt{2}}\left[\frac{1}{2}, \frac{\sqrt{3}}{2}, 0\right]^\top$, $\mathbf{a}_3 = \frac{a}{\sqrt{2}}\left[\frac{1}{2}, \frac{1}{2\sqrt{3}}, \sqrt{\frac{2}{3}}\right]^\top$, in which a is a lattice constant.

Discretizing Eq. (1) by Yee's scheme [21] on a primitive cell leads to the eigenvalue problem

$$Ax = \omega^2 B(\omega) x. \tag{3}$$

Here, A is the discrete double-curl operator $\nabla\times\nabla\times$ and it is Hermitian and semi-positive definite. The explicit matrix representation of A is derived in [7]. We give a brief description of A here.

Let n_1, n_2, and n_3 be the numbers of grid points in x, y, and z directions, respectively. Let δ_x, δ_y, and δ_z denote the associated grid lengths along the x, y, and z axial directions, respectively. The resulting large-scale $(3n_1n_2n_3)\times(3n_1n_2n_3)$ Hermitian matrix A in (3) is of the form

$$A = C^*C.$$

Here,

$$C = \begin{bmatrix} 0 & -C_3 & C_2 \\ C_3 & 0 & -C_1 \\ -C_2 & C_1 & 0 \end{bmatrix}, \tag{4}$$

$$C_1 = I_{n_2n_3} \otimes K_1,\ C_2 = I_{n_3} \otimes K_2,\ C_3 = K_3, \tag{5}$$

in which

$$K_1 = \frac{1}{\delta_x}\begin{bmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \\ e^{\imath 2\pi \mathbf{k}\cdot\mathbf{a}_1} & & & -1 \end{bmatrix} \in \mathbb{C}^{n_1\times n_1},$$

$$K_2 = \frac{1}{\delta_y}\begin{bmatrix} -I_{n_1} & I_{n_1} & & \\ & \ddots & \ddots & \\ & & -I_{n_1} & I_{n_1} \\ e^{\imath 2\pi \mathbf{k}\cdot\mathbf{a}_2} J_2 & & & -I_{n_1} \end{bmatrix} \in \mathbb{C}^{(n_1n_2)\times(n_1n_2)},$$

$$K_3 = \frac{1}{\delta_z}\begin{bmatrix} -I_{n_1n_2} & I_{n_1n_2} & & \\ & \ddots & \ddots & \\ & & -I_{n_1n_2} & I_{n_1n_2} \\ e^{\imath 2\pi \mathbf{k}\cdot\mathbf{a}_3} J_3 & & & -I_{n_1n_2} \end{bmatrix} \in \mathbb{C}^{(n_1n_2n_3)\times(n_1n_2n_3)},$$

and

$$J_2 = \begin{bmatrix} 0 & e^{-\imath 2\pi \mathbf{k}\cdot\mathbf{a}_1} I_{\frac{1}{2}n_1} \\ I_{\frac{1}{2}n_1} & 0 \end{bmatrix},$$

$$J_3 = \begin{bmatrix} 0 & e^{-\imath 2\pi \mathbf{k}\cdot\mathbf{a}_2} I_{\frac{1}{3}n_2} \otimes I_{n_1} \\ I_{\frac{2}{3}n_2} \otimes J_2 & 0 \end{bmatrix}.$$

Furthermore, $B(\omega)$ is the mass matrix for the permittivity. $B(\omega)$ can be split into two matrices

$$B(\omega) = B_n + \varepsilon_d(r,\omega) B_d, \tag{6}$$

where the subscripts n and d denote non-dispersive and dispersive materials, respectively, and $\varepsilon_d(r,\omega)$ is the relative permittivity of the dispersive material defined in (2).

Substituting (2) into (6), Eq. (3) can be rewritten as

$$Ax = \omega^2 B_n x + \omega^2 \varepsilon_d(r,\omega) B_d x = \left[\omega^2 B_n + \omega^2\left(1 - \frac{\omega_p^2}{\omega^2}\right) B_d\right] x.$$

Or equivalently, the generalized eigenvalue problem (GEP)

$$\left(A + \omega_p^2 B_d\right) x = \omega^2 (B_d + B_n) x. \tag{7}$$

3. The Eigenvalue Solvers

We study how we can solve GEP (7) efficiently to obtain the smallest positive eigenvalues that are of interest in the following subsections.

3.1. Formulation of the Shifted-and-Inverted Standard Eigenvalue Problem

As the GEP has both zero and positive eigenvalues [6], the desired smallest positive eigenvalues are located in the interior of the eigenvalue spectrum. We thus use the shift-and-invert Lanczos method to solve the GEP. In the shift-and-invert Lanczos method, the GEP is rewritten as the following shifted-and-inverted standard eigenvalue problem (SI-SEP)

$$\Big(\big[A+\omega_p^2 B_d-\sigma(B_d+B_n)\big]^{-1}(B_d+B_n)\Big)\,x=\mu x, \tag{8}$$

where σ is the user-defined shift and

$$\mu=(\omega^2-\sigma)^{-1}. \tag{9}$$

The SI-SEP is then solved by the standard Lanczos method to find the eigenvalues μ of (8). The desired eigenvalues ω^2 of the GEP can be computed easily by (9).

The computational cost for solving the SI-SEP by standard Lanczos method is dominated by the matrix-vector multiplication

$$\big[A+\omega_p^2 B_d-\sigma(B_d+B_n)\big]^{-1}(B_d+B_n)\mathbf{w}$$

or equivalently

$$\big([A-\sigma B_n-(\sigma-\omega_p^2)B_d]^{-1}(B_d+B_n)\big)\,\mathbf{w} \tag{10}$$

for a certain vector $\mathbf{w}$. This matrix-vector multiplication is performed by solving the Lanczos linear system (LLS)

$$\big[A-\big(\sigma B_n+(\sigma-\omega_p^2)B_d\big)\big]\,\mathbf{z}=\mathbf{b} \tag{11}$$

by using a preconditioned Krylov space iterative method (e.g. BiCGSTAB [20]) by letting $\mathbf{b}=(B_d+B_n)\mathbf{w}$.

3.2. Preconditioning Schemes

Performance of the iterative solver for solving the LLS (11) mainly relies on (i) the choice of preconditioner and (ii) how the preconditioner is incorporated into the iterative solver.

For the choice of preconditioner, as the component $\left(\sigma B_n + (\sigma - \omega_p^2)B_d\right)$ in (11) is simply a diagonal matrix, we take τ as the average of the diagonal entries of $\left(\sigma B_n + (\sigma - \omega_p^2)B_d\right)$ and then use

$$M = A - \tau I \tag{12}$$

as the preconditioner. Note that a similar entry averaging preconditioning strategy is proposed in [5] for simple cubic photonic crystals.

Next, we consider the following two ways to incorporate the preconditioner into the iterative solver for solving the LLS (11).

– **Left Preconditioned Lanczos Linear System (LP-LLS).** We can apply the left multiplication of M^{-1} to solve the LLS (11) directly. In this case, it is equivalent to solve the LP-LLS

$$\left[M^{-1}\left(A - \sigma B_n - (\sigma - \omega_p^2)B_d\right)\right]\mathbf{z} = (M^{-1}\mathbf{b}). \tag{13}$$

In each iteration of the iterative solver, we need to perform the matrix-vector multiplication involving the matrix

$$\left[M^{-1}\left(A - \sigma B_n - (\sigma - \omega_p^2)B_d\right)\right].$$

The complexity of such matrix-vector multiplication is $87n+6O(n\log n)$, where n is the dimension of A.

– **Embedded Preconditioned Lanczos Linear System (EP-LLS).** Based on the fact that

$$\begin{aligned} & M^{-1}\left(A - \sigma B_n - (\sigma - \omega_p^2)B_d\right) \\ = \; & (A - \tau I)^{-1}\left(A - \tau I + \tau I - \sigma B_n - (\sigma - \omega_p^2)B_d\right) \\ = \; & I + M^{-1}\left(\tau I - \sigma B_n - (\sigma - \omega_p^2)B_d\right), \end{aligned}$$

we can derive the EP-LLS

$$\left[I + M^{-1}\left(\tau I - \sigma B_n - (\sigma - \omega_p^2)B_d\right)\right]\mathbf{z} = (M^{-1}\mathbf{b}). \tag{14}$$

The complexity of such matrix-vector multiplication involving the matrix $\left[I + M^{-1}\left(\tau I - \sigma B_n - (\sigma - \omega_p^2)B_d\right)\right]$ is $45n + 6O(n\log n)$.

Unlike the LP-LLS (13), the computations in the EP-LLS (14) do *not* involve the matrix A. Precisely, the EP-LLS leads to a saving of the matrix-vector multiplication involving A, which has complexity $42n$. We thus predict that EP-LLS has better timing performance. This prediction is verified in the numerical results to be shown in Section 4.

3.3. Inverting the Preconditioner

To solve the LP-LLS (13) and the EP-LLS (14) by a Krylov space iterative method, the most costly task in each iteration is the matrix-vector multiplication involving the matrix $\left[M^{-1}\left(A-\sigma B_n-(\sigma-\omega_p^2)B_d\right)\right]$ and the matrix $\left[I+M^{-1}\left(\tau I-\sigma B_n-(\sigma-\omega_p^2)B_d\right)\right]$ for LP-LLS and EP-LLS, respectively. In such matrix-vector multiplications and the computations of the right hand side $M^{-1}\mathbf{b}$, we need to compute $M^{-1}\mathbf{y}$ by solving the preconditioner linear system (PLS)

$$(A-\tau I)\mathbf{y}=\mathbf{d} \tag{15}$$

for a certain vector $\mathbf{y}$ and $\mathbf{d}$. For example, to compute $M^{-1}\mathbf{b}$, we let $\mathbf{d}=\mathbf{b}$. Taking the EP-LLS as another example, we need to compute the matrix-vector multiplication

$$\left[I+M^{-1}\left(\tau I-\sigma B_n-(\sigma-\omega_p^2)B_d\right)\right]\mathbf{t}$$

in each iteration of a Krylov space method for a given vector $\mathbf{t}$. This is equivalent to compute

$$\mathbf{t}+M^{-1}\left(\tau I-\sigma B_n-(\sigma-\omega_p^2)B_d\right)\mathbf{t}.$$

In this case, we need to solve a linear system with the form shown in (15) by letting $\mathbf{d}=\left(\tau I-\sigma B_n-(\sigma-\omega_p^2)B_d\right)\mathbf{t}$.

Next, we propose the following two ways to solve the PLS (15) by using the eigendecomposition of A and the so-called FFT based matrix-vector multiplications.

- **Direct Inverse.** As outlined in Theorem 2 in the appendix, there exists a unitary matrix Q and a diagonal matrix Σ such that $A=Q\Sigma Q^*$ (see the appendix and [6] for details). By letting $\mathbf{y}=Q\tilde{\mathbf{y}}$, the PLS can be rewritten as $(\Sigma-\tau I)\,\tilde{\mathbf{y}}=Q^*\mathbf{d}$. That is, the vector $\mathbf{y}$ can be computed by

$$\mathbf{y}=Q\left[(\Sigma-\tau I)^{-1}(Q^*\mathbf{d})\right]. \tag{16}$$

  As $(\Sigma-\tau I)$ is simply a diagonal matrix, the key computations of this approach are the matrix-vector multiplications involving Q^* and Q. Furthermore, since $Q=(I_3\otimes T)\,\Lambda$, the matrix-vector multiplications are performed with respect to T^* and T. Here, Λ and T are defined in Theorem 2 and (21a), respectively. The complexity for computing $\mathbf{y}$ in (16) is $33n$ FLOPS plus three matrix-vector multiplications involving T or T^*.

- **Block Operations.** The second scheme is derived from the viewpoint of block operations. See [6] for more details. From (4), we

have $A = I_3 \otimes (G^*G) - GG^*$, where $G = [C_1^\top, C_2^\top, C_3^\top]^\top$. Therefore, the PLS (15) can be reformulated as

$$[I_3 \otimes (G^*G) - \tau I]\,\mathbf{y} = \mathbf{d} + GG^*\mathbf{y}. \tag{17}$$

Multiplying Eq. (15) by GG^* and applying the fact $CG = 0$, it holds that $GG^*\mathbf{y} = -\tau^{-1}GG^*\mathbf{d}$. Consequently, (17) is equivalent to

$$[I_3 \otimes (G^*G) - \tau I]\,\mathbf{y} = \mathbf{d} - \tau^{-1}GG^*\mathbf{d}.$$

Applying the Schur decompositions of C_ℓ's shown in Theorem 1 of the appendix, we have

$$\mathbf{y} = (I_3 \otimes T)\,(I_3 \otimes \Lambda_q - \tau I)^{-1} \left(I - \tau^{-1} \begin{bmatrix} \Lambda_{\mathbf{x}} \\ \Lambda_{\mathbf{y}} \\ \Lambda_{\mathbf{z}} \end{bmatrix} \begin{bmatrix} \Lambda_{\mathbf{x}}^* & \Lambda_{\mathbf{y}}^* & \Lambda_{\mathbf{z}}^* \end{bmatrix} \right) (I_3 \otimes T)^*\,\mathbf{d}. \tag{18}$$

As $\Lambda_{\mathbf{x}}$, $\Lambda_{\mathbf{y}}$, and $\Lambda_{\mathbf{z}}$ are diagonal matrices, the key computations of (18) are again the matrix-vector multiplications involving T^* and T. The complexity for computing $\mathbf{y}$ in (18) is $14n$ FLOPS plus three matrix-vector multiplications involving T or T^*.

We have mentioned that the key computational cost for computing $\mathbf{y}$ by (16) and (18) is the matrix-vector products $T^*\mathbf{p}$ and $T\mathbf{q}$ for a certain vectors $\mathbf{p}$ and $\mathbf{q}$. In Algorithms 3 and 4 of appendix (Section B), we demonstrate how $T^*\mathbf{p}$ and $T\mathbf{q}$ can be computed efficiently. While these two algorithms are the same as the ones proposed in [6]. The derivation shown in the appendix is more straightforward and easier to understand.

3.4. A Short Summary

Finally, all of the schemes discussed in this section are summarized in Algorithm 1 and Algorithm 2. These conceptual algorithms emphasize the frameworks that consists of the proposed schemes. Details are omitted to increase their readabilities. As shown in line 1 of Algorithm 2, we have a choice between the LP-LLS (13) and EP-LLS (14) while solving the LLS (11). To solve the PLS (15) in line 5 of Algorithm 2, we have a choice of the Direct Inverse (16) or Block Operation (18) which can be performed efficiently by the eigendecomposition A and the FFT-based $T^*\mathbf{p}$ and $T\mathbf{q}$. In the next section, we conduct numerical experiments to decide the best scheme choice.

Algorithm 1 The Lanczos framework for solving the GEP (7)

1: Transform the GEP (7) to the SI-SEP (8)
2: Solve the SI-SEP (8) by the Lanczos method
3: <· · ·>
4: **repeat**
5: <· · ·>
6: Perform the matrix-vector multiplication (10) by Algorithm 2
7: <· · ·>
8: **until** the desired eigenpairs are converged

Algorithm 2 An iterative linear system solver for the LLS (11)

1: Choose the left preconditioning scheme (13) or the embedded preconditioning scheme (14)
2: <· · ·>
3: **repeat**
4: <· · ·>
5: Solve PLS (15) by Direct Inverse (16) or Block Operation (18)
6: <· · ·>
7: **until** the solution of (11) is converged

4. Numerical results

To study the numerical performance of the proposed schemes, we consider the setup of a 3D dispersive metallic photonic crystals with a FCC lattice described in [1, 7]. The particular lattice consists of dielectric spheres and connecting spheroids as shown in Figure 1. In particular, the radius r of the spheres is $r = 0.12a$ and the connecting spheroid has a minor axis length $s = 0.11a$, where a is the lattice constant. The plasma frequency ω_p and the permittivity ε_o of the non-dispersive material are set to be $2\pi/a$ and 1, respectively.

Implementation details of the schemes proposed in Section 3 are described as follows. The MATLAB function `eigs` is used to solve the GEP (7) with the symmetric option. The maximal number of Lanczos vectors for the restart is $3m$, where m is the number of desired eigenvalues. The MATLAB function `bicgstab` (biconjugate gradient stabilized method or BiCGSTAB) is used to solve the LP-LLS (13) with the preconditioner M. The same function `bicgstab` is used to solve the EP-LLS (14) *without* preconditioning, as the preconditioner M has been embedded in (14) already. For the forward FFT, the matrices $P_{\mathbf{x}}$, $P_{\mathbf{y}}$ and $P_{\mathbf{z}}$ in Algorithm 3 are computed by the MATLAB function `fft`. For the backward FFT, the matrices $Q_{\mathbf{z},i}$, $Q_{\mathbf{y}}$ and $Q_{\mathbf{x}}$ in

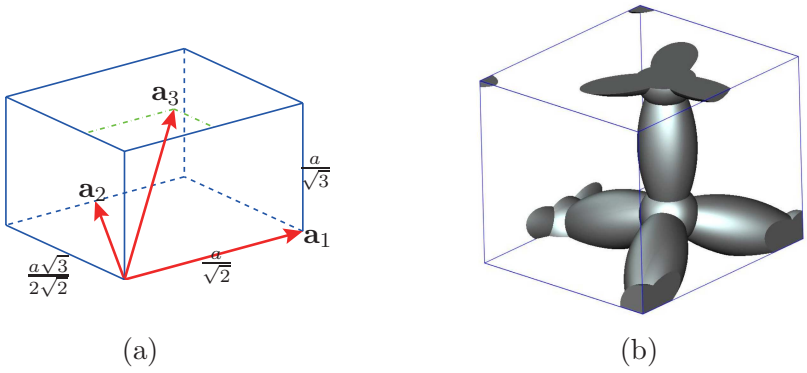


Figure 1. (a) The primitive cell and lattice translation vectors $\mathbf{a}_1$, $\mathbf{a}_2$, and $\mathbf{a}_3$. Any pair of the vectors form a 60 degree angle. The length, width and height of the primitive cell are $\frac{a}{\sqrt{2}}$, $\frac{a\sqrt{3}}{2\sqrt{2}}$ and $\frac{a}{\sqrt{3}}$, respectively.) (b) A schema of photonic crystal structure with a FCC lattice within a single primitive cell.

Algorithm 4 are computed by `ifft`. For timing results, the MATLAB commands `tic` and `toc` are used to measure the elapsed time. The IEEE double-precision floating-point arithmetic is used. The stopping criterion is $10^4\epsilon$ and ϵ/δ_x^2 for `eigs` and `bicgstab`, respectively. The constant ϵ ($\approx 2^{-52}$) is the floating-point relative accuracy in MATLAB. The numerical experiments are carried out on a HP workstation that is equipped with two Intel Quad-Core Xeon X5687 3.6GHz CPUs, 48 GB of main memory, and the RedHat Linux operating system.

The timing performance are compared for solving the PLS (15), the LLS (11), and the GEP (7). First, we study the timing performance for solving the PLS (15) by the Direct Inverse (16) and the Block Operation (18). The results are shown in Figure 2. The x-axis of the figure is the dimension of A. By taking $n_1 = n_2 = n_3 = 108+12j$ for $j = 0, 1, \ldots, 9$, the matrix dimension (i.e. $3n_1n_2n_3$) ranges from $3,779,136$ to $30,233,088$. The y-axis is the CPU time in second. The figure shows that both approaches can compute $\mathbf{y}$ in a short time even for a large A. For example, we can solve the PLS by taking around 7 seconds even for the matrix that is larger than 3 million. The figure also suggests that Block Operation (18) is slightly faster, which is mainly caused by the difference of computational complexity $(33 - 14) \cdot (n_1n_2n_3)$. Consequently, we use Block Operation (18) in the next part of the numerical experiments from now on.

Second, we study the timing performance for solving the LLS (11) by the LP-LLS (13) and the EP-LLS (14). The results are shown in Figure 3. The x-axis of the figure is the dimension of A. By taking

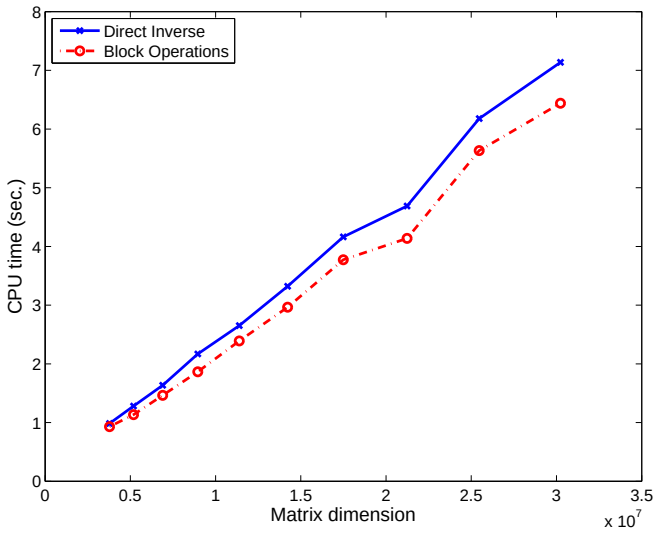


Figure 2. The time for solving the PLS (15) by the formula (16) and (18).

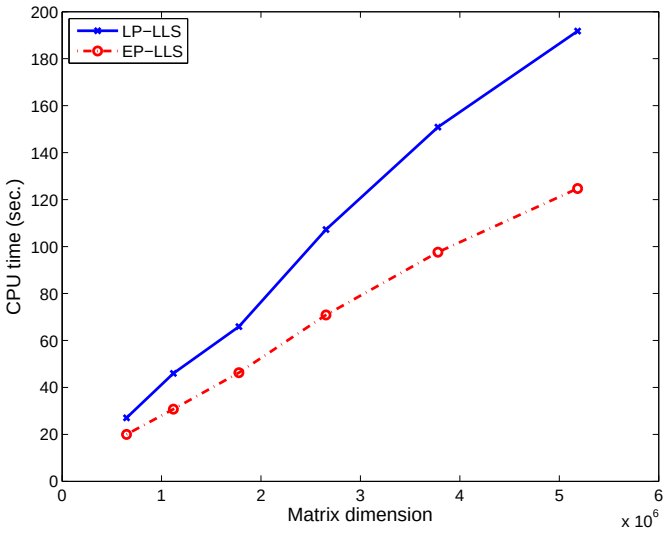
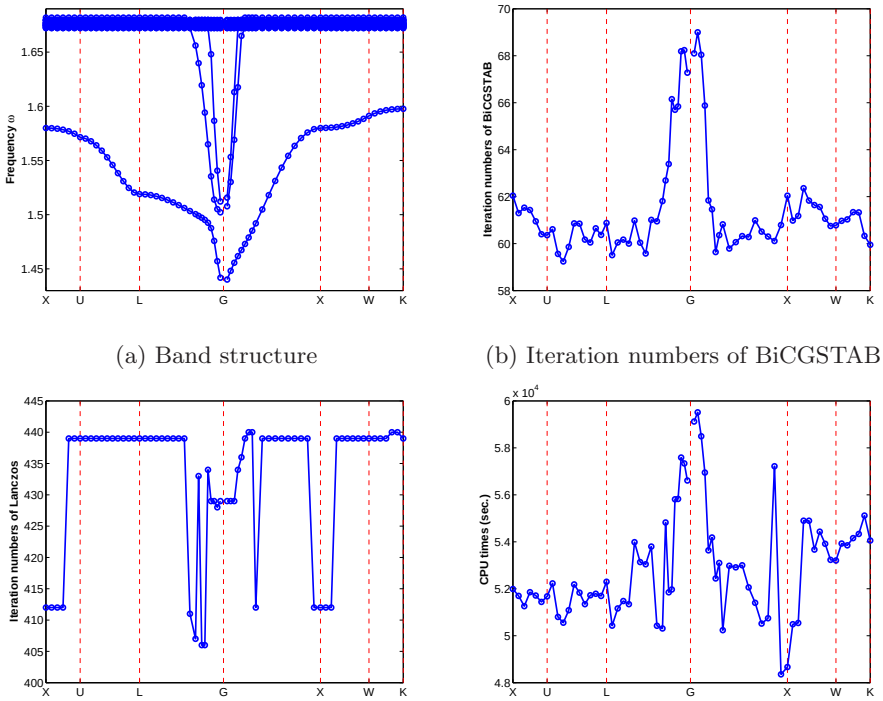


Figure 3. The time for solving the LLS (11) by the LP-LLS (13) and the EP-LLS (14).



(a) Band structure

(b) Iteration numbers of BiCGSTAB

(c) Iteration numbers of Lanczos

(d) CPU timing results

Figure 4. The numerical results for computing the band structure of the 3D dispersive metallic photonic crystal with a FCC lattice.

$n_1 = n_2 = n_3 = 60 + 12j$ for $j = 0, 1, \ldots, 5$, the matrix dimension ranges from $648,000$ to $5,184,000$. The figure suggests that the EP-LLS (14) is faster by using 26% to 35% less time.

Third, we study the performances of the proposed methods for finding the 20 smallest positive eigenvalues of the GEP (7) to plot the band structure. Based on the numerical results in the first two parts, we solve the EP-LLS (14) by the `bicgstab` and the PLS (15) by the Block Operation (18). We take $n_1 = n_2 = n_3 = 96$ so the size of the tested matrix A is $2,654,208$. The wave vectors $\mathbf{k}$'s are chosen along the segments connecting X, U, L, G, X, W, and K in the first Brillouin zone. The shift $\sigma = 1$. We highlight the following observations.

– Figure 4 (b) shows that the (average) iteration numbers for solving EP-LLS (14) in the tested eigenvalue problems ranges from 59 to 69. For the problems as large as 2.6 million, the results suggest that $M = (A - \sigma I)$ in (12) is an efficient preconditioner of the LLS (11).

– We use the shift-and-invert Lanczos method to solve the SI-SEP (8) associated with different wave vectors. As shown in Figure 4(c), the iteration numbers of the shift-and-invert Lanczos method ranges from 406 to 440 and the average is 433 iterations. The results is satisfactory as the the shift-and-invert Lanczos method takes around 433 iterations to solve the SI-SEP with matrix size as large as $2,654,208$. Note that the shift-and-invert Lanczos method spends most of its iteration numbers to find the cluster of eigenvalues near 1.67^2 (as $\lambda = \omega^2$ and $\omega = 1.67$) as shown in Figure 4 (a).

– Figure 4(d) shows that the CPU time for solving the SI-SEP ranges from 13.4 to 16.5 hours and the average is 14.7 hours. These results suggest that our approach is efficient, which mainly due to the highly efficient solver for the EP-LLS (14). From the results in parts (c) and (d) of Figure 4, each iteration of the shift-and-invert Lanczos method takes only 122.6 seconds of CPU time in average.

5. Conclusions

To compute the band structure of 3D dispersive metallic photonic crystals with a face-centered cubic lattice, we study how the GEP (7) can be solved efficiently in the form of SI-SEP (8). The GEP is due to the discretization of the governing Maxwell equations associated with the lossless Drude model. We have presented various algorithms to solve the GEP. Numerical results have verified that the GEP can be solved efficiently by the shift-and-invert Lanczos method if the EP-LLS (14) is solved by the `bicgstab` and the PLS (15) is solved by the Block Operation shown in (18).

The proposed schemes can be extended to the dispersive photonic crystals with simple cubic lattice straightforwardly. By suitable modifications (which may not be trivial), we anticipate the schemes can be extended to the rational eigenvalue problems associated with the Drude model. We believe the computational timing can be further accelerated by using parallel computers or GPU-accelerated computers.

Appendix

A. Eigendecompositions of A

In this section, we briefly summarize the explicit Schur decomposition for FCC lattices that is derived in [6]. First, we define the notations to

be used later. Let

$$\theta_{\mathbf{x},i} = \frac{\imath 2\pi i}{n_1},\ \theta_{\mathbf{y},j} = \frac{\imath 2\pi j}{n_2},\ \theta_{\mathbf{z},k} = \frac{\imath 2\pi k}{n_3};$$
$$\varepsilon_{\mathbf{x}} = \frac{\imath 2\pi \mathbf{k}\cdot\mathbf{a}_1}{n_1},\ \varepsilon_{\mathbf{y},i} = \frac{\imath 2\pi}{n_2}\left\{\mathbf{k}\cdot\left(\mathbf{a}_2 - \frac{1}{2}\mathbf{a}_1\right) - \frac{i}{2}\right\},$$
$$\varepsilon_{\mathbf{z},i+j} = \frac{\imath 2\pi}{n_3}\left\{\mathbf{k}\cdot\left(\mathbf{a}_3 - \frac{1}{3}\left(\mathbf{a}_1+\mathbf{a}_2\right)\right) - \frac{1}{3}(i+j)\right\},$$

for $i = 1,\ldots,n_1$, $j = 1,\ldots,n_2$, $k = 1,\ldots,n_3$ and define

$$\Lambda_{n_1} = \delta_x^{-1}\mathrm{diag}\left(e^{\theta_1}-1, e^{\theta_2}-1, \cdots, e^{\theta_{n_1}}-1\right), \tag{19a}$$
$$\Lambda_{i,n_2} = \delta_y^{-1}\mathrm{diag}\left(e^{\theta_{i,1}}-1, e^{\theta_{i,2}}-1, \cdots, e^{\theta_{i,n_2}}-1\right), \tag{19b}$$
$$\Lambda_{i,j,n_3} = \delta_z^{-1}\mathrm{diag}\left(e^{\theta_{i,j,1}}-1, e^{\theta_{i,j,2}}-1, \cdots, e^{\theta_{i,j,n_3}}-1\right), \tag{19c}$$

and

$$\mathbf{x}_i = E_{\mathbf{x}}\left[1\ \ e^{\theta_{\mathbf{x},i}}\ \ \cdots\ \ e^{(n_1-1)\theta_{\mathbf{x},i}}\right]^\top \equiv E_{\mathbf{x}}\mathbf{u}_{\mathbf{x},i}, \tag{20a}$$
$$\mathbf{y}_{i,j} = E_{\mathbf{y},i}\left[1\ \ e^{\theta_{\mathbf{y},j}}\ \ \cdots\ \ e^{(n_2-1)\theta_{\mathbf{y},j}}\right]^\top \equiv E_{\mathbf{y},i}\mathbf{u}_{\mathbf{y},j}, \tag{20b}$$
$$\mathbf{z}_{i,j,k} = E_{\mathbf{z},i+j}\left[1\ \ e^{\theta_{\mathbf{z},k}}\ \ \cdots\ \ e^{(n_3-1)\theta_{\mathbf{z},k}}\right]^\top \equiv E_{\mathbf{z},i+j}\mathbf{u}_{\mathbf{z},k}, \tag{20c}$$

where $\theta_i = \theta_{\mathbf{x},i} + \varepsilon_{\mathbf{x}}$, $\theta_{i,j} = \theta_{\mathbf{y},j} + \varepsilon_{\mathbf{y},i}$, $\theta_{i,j,k} = \theta_{\mathbf{z},k} + \varepsilon_{\mathbf{z},i+j}$ and

$$E_{\mathbf{x}} = \mathrm{diag}\left(1, e^{\varepsilon_{\mathbf{x}}}, \cdots, e^{(n_1-1)\varepsilon_{\mathbf{x}}}\right),$$
$$E_{\mathbf{y},i} = \mathrm{diag}\left(1, e^{\varepsilon_{\mathbf{y},i}}, \cdots, e^{(n_2-1)\varepsilon_{\mathbf{y},i}}\right),$$
$$E_{\mathbf{z},i+j} = \mathrm{diag}\left(1, e^{\varepsilon_{\mathbf{z},i+j}}, \cdots, e^{(n_3-1)\varepsilon_{\mathbf{z},i+j}}\right).$$

Furthermore, we let

$$T = \frac{1}{\sqrt{n_1 n_2 n_3}}\left[\ T_1\ \ T_2\ \ \cdots\ \ T_{n_1}\ \right] \in \mathbb{C}^{n\times n}, \tag{21a}$$

where

$$T_i = \left[\ T_{i,1}\ \ T_{i,2}\ \ \cdots\ \ T_{i,n_2}\ \right] \in \mathbb{C}^{n\times(n_2 n_3)}, \tag{21b}$$
$$T_{i,j} = \left[\mathbf{z}_{i,j,1}\ \ \mathbf{z}_{i,j,2}\ \ \cdots\ \ \mathbf{z}_{i,j,n_3}\right] \otimes (\mathbf{y}_{i,j}\otimes\mathbf{x}_i) \in \mathbb{C}^{n\times n_3}. \tag{21c}$$

As shown in [6], the matrices C_ℓ's for $(\ell = 1,2,3)$ can be diagonalized simultaneously. The Schur decompositions of C_ℓ's are then achieved in Theorem 1.

Conjecture 1. (Schur decomposition of C_ℓ's [6]) Let C_ℓ ($\ell = 1, 2, 3$) be defined in (5) and Λ_{n_1}, Λ_{i,n_2} and Λ_{i,j,n_3} be defined in (19). Then, T in (21a) is unitary and

$$\begin{cases} C_1 T = T\left(\Lambda_{n_1} \otimes I_{n_2 n_3}\right) \equiv T\Lambda_{\mathbf{x}}, \\ C_2 T = T\left(\left(\oplus_{i=1}^{n_1} \Lambda_{i,n_2}\right) \otimes I_{n_3}\right) \equiv T\Lambda_{\mathbf{y}}, \\ C_3 T = T\left(\oplus_{i=1}^{n_1} \oplus_{j=1}^{n_2} \Lambda_{i,j,n_3}\right) \equiv T\Lambda_{\mathbf{z}}. \end{cases}$$

Next, we define

$$\Lambda_q = \Lambda_{\mathbf{x}}^* \Lambda_{\mathbf{x}} + \Lambda_{\mathbf{y}}^* \Lambda_{\mathbf{y}} + \Lambda_{\mathbf{z}}^* \Lambda_{\mathbf{z}}, \tag{22}$$

$$\Lambda_p = (\Lambda_{\mathbf{x}} + \Lambda_{\mathbf{y}} + \Lambda_{\mathbf{z}})(\Lambda_{\mathbf{x}} + \Lambda_{\mathbf{y}} + \Lambda_{\mathbf{z}})^* \equiv \Lambda_s \Lambda_s^* \tag{23}$$

and let $\mathcal{B}$ be the Bloch wave vectors

$$\mathcal{B} = \left\{ \mathbf{k} = (\mathrm{k}_1, \mathrm{k}_2, \mathrm{k}_3)^\top \neq \mathbf{0} \;\middle|\; 0 \leq \mathrm{k}_1 \leq \frac{\sqrt{2}}{a}, 0 \leq \mathrm{k}_2 < \frac{2\sqrt{2}}{\sqrt{3}a}, 0 \leq \mathrm{k}_3 < \frac{\sqrt{3}}{a}, \right.$$
$$\left. \text{and } \mathbf{k} \neq \frac{\sqrt{2}}{a}\left[1, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{6}}\right]^\top \right\}.$$

Then, we have the following theorem.

Conjecture 2. (The Schur decomposition of A [6]) Let A, Λ_q, Λ_s and Λ_p be defined in (3), (22) and (23), respectively, and $\mathbf{k} \in \mathcal{B}$. Then Λ_q and $3\Lambda_q - \Lambda_p$ are positive definite if $\delta_x \neq \delta_y$ or $\delta_x \neq \delta_z$. Moreover, if we define

$$\Lambda = \begin{bmatrix} \Lambda_{\mathbf{x}} \Lambda_q^{-\frac{1}{2}} & (\Lambda_q - \Lambda_{\mathbf{x}} \Lambda_s^*)\left(3\Lambda_q^2 - \Lambda_q \Lambda_p\right)^{-\frac{1}{2}} & \left(\Lambda_{\mathbf{z}}^* - \Lambda_{\mathbf{y}}^*\right)(3\Lambda_q - \Lambda_p)^{-\frac{1}{2}} \\ \Lambda_{\mathbf{y}} \Lambda_q^{-\frac{1}{2}} & (\Lambda_q - \Lambda_{\mathbf{y}} \Lambda_s^*)\left(3\Lambda_q^2 - \Lambda_q \Lambda_p\right)^{-\frac{1}{2}} & \left(\Lambda_{\mathbf{x}}^* - \Lambda_{\mathbf{z}}^*\right)(3\Lambda_q - \Lambda_p)^{-\frac{1}{2}} \\ \Lambda_{\mathbf{z}} \Lambda_q^{-\frac{1}{2}} & (\Lambda_q - \Lambda_{\mathbf{z}} \Lambda_s^*)\left(3\Lambda_q^2 - \Lambda_q \Lambda_p\right)^{-\frac{1}{2}} & \left(\Lambda_{\mathbf{y}}^* - \Lambda_{\mathbf{x}}^*\right)(3\Lambda_q - \Lambda_p)^{-\frac{1}{2}} \end{bmatrix},$$
$$Q = (I_3 \otimes T)\,\Lambda,$$

then Q is unitary and

$$A = Q \mathrm{diag}\,(0, \Lambda_q, \Lambda_q)\, Q^* = Q \Sigma Q^*.$$

B. Fast Matrix-Vector Multiplication for $T^*\mathbf{p}$ and $T\mathbf{q}$

As Λ_q, Λ_x, Λ_y, and Λ_z are all diagonal matrices, the key computational cost for solving $\mathbf{y}$ by (16) and (18) is the matrix-vector products $T^*\mathbf{p}$ and $T\mathbf{q}$ for a certain vectors $\mathbf{p}$ and $\mathbf{q}$. Fortunately, an efficient technique

for the matrix-vector products $T^*\mathbf{p}$ and $T\mathbf{q}$ is proposed in [6], so the solution of the preconditioning linear system (15) can be computed efficiently. In this subsection, we re-derive the details for computing $T^*\mathbf{p}$ and $T\mathbf{q}$ in another way that is more straightforward then the derivation presented in [6].

First, we consider $T^*\mathbf{p}$. From (20c), $T_{i,j}$ in (21c) can be represented as

$$\begin{aligned} T_{i,j} &= \left[E_{\mathbf{z},i+j}\mathbf{u}_{\mathbf{z},1} \;\; E_{\mathbf{z},i+j}\mathbf{u}_{\mathbf{z},2} \;\; \cdots \;\; E_{\mathbf{z},i+j}\mathbf{u}_{\mathbf{z},n_3}\right] \otimes (\mathbf{y}_{i,j} \otimes \mathbf{x}_i) \\ &= (E_{\mathbf{z},i+j}U_{\mathbf{z}}) \otimes (\mathbf{y}_{i,j} \otimes \mathbf{x}_i) \end{aligned} \tag{24}$$

for $i = 1, \ldots, n_1$, $j = 1, \ldots, n_2$, and $k = 1, \ldots, n_3$, where

$$U_{\mathbf{z}} = \left[\mathbf{u}_{\mathbf{z},1} \;\; \mathbf{u}_{\mathbf{z},2} \;\; \cdots \;\; \mathbf{u}_{\mathbf{z},n_3}\right].$$

Substituting (24) into (21b) and using (20b), T_i can be rewritten as

$$\begin{aligned} T_i &= \left[(E_{\mathbf{z},i+1}U_{\mathbf{z}}) \otimes \mathbf{y}_{i,1} \otimes \mathbf{x}_i \;\; \cdots \;\; (E_{\mathbf{z},i+n_2}U_{\mathbf{z}}) \otimes \mathbf{y}_{i,n_2} \otimes \mathbf{x}_i\right] \\ &= \left[(E_{\mathbf{z},i+1}U_{\mathbf{z}}) \otimes \mathbf{y}_{i,1} \;\; \cdots \;\; (E_{\mathbf{z},i+n_2}U_{\mathbf{z}}) \otimes \mathbf{y}_{i,n_2}\right] \otimes \mathbf{x}_i \\ &= \left[(E_{\mathbf{z},i+1}U_{\mathbf{z}}) \otimes (E_{\mathbf{y},i}\mathbf{u}_{\mathbf{y},1}) \;\; \cdots \;\; (E_{\mathbf{z},i+n_2}U_{\mathbf{z}}) \otimes (E_{\mathbf{y},i}\mathbf{u}_{\mathbf{y},n_2})\right] \otimes \mathbf{x}_i \\ &\equiv V_{\mathbf{yz},i} \otimes \mathbf{x}_i, \end{aligned} \tag{25}$$

which implies that

$$T = \frac{1}{\sqrt{n_1 n_2 n_3}} \left[V_{\mathbf{yz},1} \otimes \mathbf{x}_1 \;\; V_{\mathbf{yz},2} \otimes \mathbf{x}_2 \;\; \cdots \;\; V_{\mathbf{yz},n_1} \otimes \mathbf{x}_{n_1}\right].$$

For a given vector $\mathbf{p} \in \mathbb{C}^{n_1 n_2 n_3}$, we denote $\mathbf{p}$ by letting $\mathbf{p} = \left[\mathbf{p}_1^\top \;\; \cdots \;\; \mathbf{p}_{n_2 n_3}^\top\right]^\top$ with $\mathbf{p}_i \in \mathbb{C}^{n_1}$. By the property

$$\left(E^\top \otimes F\right) \mathrm{vec}\,(X) = \mathrm{vec}\,(FXE),$$

we have

$$T^*\mathbf{p} = \frac{1}{\sqrt{n_1 n_2 n_3}} \begin{bmatrix} \left(V_{\mathbf{yz},1}^* \otimes \mathbf{x}_1^*\right)\mathbf{p} \\ \left(V_{\mathbf{yz},2}^* \otimes \mathbf{x}_2^*\right)\mathbf{p} \\ \vdots \\ \left(V_{\mathbf{yz},n_1}^* \otimes \mathbf{x}_{n_1}^*\right)\mathbf{p} \end{bmatrix} = \frac{1}{\sqrt{n_1 n_2 n_3}} \begin{bmatrix} \mathrm{vec}\left(\mathbf{x}_1^* P \bar{V}_{\mathbf{yz},1}\right) \\ \mathrm{vec}\left(\mathbf{x}_2^* P \bar{V}_{\mathbf{yz},2}\right) \\ \vdots \\ \mathrm{vec}\left(\mathbf{x}_{n_1}^* P \bar{V}_{\mathbf{yz},n_1}\right) \end{bmatrix},$$

where $P = \left[\mathbf{p}_1 \;\; \cdots \;\; \mathbf{p}_{n_2 n_3}\right]$. Let

$$\begin{aligned} U_{\mathbf{x}} &= \left[\mathbf{u}_{\mathbf{x},1} \;\; \mathbf{u}_{\mathbf{x},2} \;\; \cdots \;\; \mathbf{u}_{\mathbf{x},n_1}\right], \\ \left[\mathbf{p}_{\mathbf{x},1} \;\; \mathbf{p}_{\mathbf{x},2} \;\; \cdots \;\; \mathbf{p}_{\mathbf{x},n_1}\right]^* &\equiv \left[\mathbf{x}_1 \;\; \mathbf{x}_2 \;\; \cdots \;\; \mathbf{x}_{n_1}\right]^* P = U_{\mathbf{x}}^* E_{\mathbf{x}}^* P, \\ \mathbf{p}_{\mathbf{x},i} &= \left[\mathbf{p}_{i,1}^\top \;\; \cdots \;\; \mathbf{p}_{i,n_3}^\top\right]^\top, \\ P_i &= \left[\mathbf{p}_{i,1} \;\; \cdots \;\; \mathbf{p}_{i,n_3}\right] \end{aligned}$$

with $\mathbf{p}_{i,k} \in \mathbb{C}^{n_2}$. Then, it holds that

$$\begin{aligned}\mathbf{x}_i^* P \bar{V}_{\mathbf{yz},i} = \mathbf{p}_{\mathbf{x},i}^* \bar{V}_{\mathbf{yz},i} &= \left(V_{\mathbf{yz},i}^* \bar{\mathbf{p}}_{\mathbf{x},i}\right)^\top \\ &= \begin{bmatrix} \left((E_{\mathbf{z},i+1} U_{\mathbf{z}})^* \otimes \left(\mathbf{u}_{\mathbf{y},1}^* E_{\mathbf{y},i}^*\right)\right) \bar{\mathbf{p}}_{\mathbf{x},i} \\ \vdots \\ \left((E_{\mathbf{z},i+n_2} U_{\mathbf{z}})^* \otimes \left(\mathbf{u}_{\mathbf{y},n_2}^* E_{\mathbf{y},i}^*\right)\right) \bar{\mathbf{p}}_{\mathbf{x},i} \end{bmatrix}^\top \\ &= \begin{bmatrix} \mathrm{vec}\left(\mathbf{u}_{\mathbf{y},1}^* E_{\mathbf{y},i}^* \bar{P}_i \overline{E_{\mathbf{z},i+1} U_{\mathbf{z}}}\right) \\ \vdots \\ \mathrm{vec}\left(\mathbf{u}_{\mathbf{y},n_2}^* E_{\mathbf{y},i}^* \bar{P}_i \overline{E_{\mathbf{z},i+n_2} U_{\mathbf{z}}}\right) \end{bmatrix}^\top . \end{aligned} \tag{26}$$

Let

$$\begin{bmatrix} \mathbf{p}_{\mathbf{y},i,1} & \cdots & \mathbf{p}_{\mathbf{y},i,n_2} \end{bmatrix}^* \equiv \begin{bmatrix} \mathbf{u}_{\mathbf{y},1} & \cdots & \mathbf{u}_{\mathbf{y},n_2} \end{bmatrix}^* E_{\mathbf{y},i}^* \bar{P}_i = U_{\mathbf{y}}^* E_{\mathbf{y},i}^* \bar{P}_i \tag{27}$$

for $i = 1, \ldots, n_1$. Substituting (27) into (26), we have

$$\mathbf{x}_i^* P \bar{V}_{\mathbf{yz},i} = \begin{bmatrix} \mathrm{vec}\left(\mathbf{p}_{\mathbf{y},i,1}^* \overline{E_{\mathbf{z},i+1} U_{\mathbf{z}}}\right) \\ \vdots \\ \mathrm{vec}\left(\mathbf{p}_{\mathbf{y},i,n_2}^* \overline{E_{\mathbf{z},i+n_2} U_{\mathbf{z}}}\right) \end{bmatrix}^\top = \begin{bmatrix} U_{\mathbf{z}}^* E_{\mathbf{z},i+1}^* \bar{\mathbf{p}}_{\mathbf{y},i,1} \\ \vdots \\ U_{\mathbf{z}}^* E_{\mathbf{z},i+n_2}^* \bar{\mathbf{p}}_{\mathbf{y},i,n_2} \end{bmatrix}^\top .$$

In short, we summarize this way to compute $T^*\mathbf{p}$ in Algorithm 3.

Algorithm 3 Forward FFT-based matrix-vector product $T^*\mathbf{p}$.

Require: Any vector $\mathbf{p} = \begin{bmatrix} \mathbf{p}_1^\top & \cdots & \mathbf{p}_{n_2 n_3}^\top \end{bmatrix}^\top \in \mathbb{C}^n$ with $\mathbf{p}_i \in \mathbb{C}^{n_1}$ for $i = 1, \ldots, n_1$.
Ensure: The vector $\mathbf{f} \equiv T^*\mathbf{p}$.
1: Compute $P_{\mathbf{x}} = \left(U_{\mathbf{x}}^* E_{\mathbf{x}}^* \begin{bmatrix} \mathbf{p}_1 & \cdots & \mathbf{p}_{n_2 n_3} \end{bmatrix}\right)^*$;
2: **for** $i = 1, \ldots, n_1$ **do**
3: Set $P_i = \begin{bmatrix} P_{\mathbf{x}}(1:n_2, i) & P_{\mathbf{x}}(n_2+1:2n_2, i) & \cdots & P_{\mathbf{x}}((n_3-1)n_2+1:n_2 n_3, i) \end{bmatrix}$;
4: Compute $P_{\mathbf{y}} = \left(U_{\mathbf{y}}^* E_{\mathbf{y},i}^* \bar{P}_i\right)^*$;
5: Compute $P_{\mathbf{z}} = U_{\mathbf{z}}^* \begin{bmatrix} E_{\mathbf{z},i+1}^* \bar{P}_{\mathbf{y}}(:,1) & E_{\mathbf{z},i+2}^* \bar{P}_{\mathbf{y}}(:,2) & \cdots & E_{\mathbf{z},i+n_2}^* \bar{P}_{\mathbf{y}}(:,n_2) \end{bmatrix}$.
6: Set $\mathbf{f}((i-1)n_2 n_3 + 1 : i n_2 n_3) = \frac{1}{\sqrt{n_1 n_2 n_3}} \mathrm{vec}(P_{\mathbf{z}})$.
7: **end for**

Second, we consider $T\mathbf{q}$. For a given vector $\mathbf{q} \in \mathbb{C}^n$, we denote $\mathbf{q}$ recursively by letting $\mathbf{q} = \begin{bmatrix} \mathbf{q}_1^\top & \cdots & \mathbf{q}_{n_1}^\top \end{bmatrix}^\top$ with $\mathbf{q}_i = \begin{bmatrix} \mathbf{q}_{i,1}^\top & \cdots & \mathbf{q}_{i,n_2}^\top \end{bmatrix}^\top \in$

$\mathbb{C}^{n_2 n_3}$ and $\mathbf{q}_{i,j} \in \mathbb{C}^{n_3}$. Then, we have

$$\begin{aligned} T\mathbf{q} &= \frac{1}{\sqrt{n_1 n_2 n_3}} \left[V_{\mathbf{yz},1} \otimes \mathbf{x}_1 \ \cdots \ V_{\mathbf{yz},n_1} \otimes \mathbf{x}_{n_1} \right] \left[\mathbf{q}_1^\top \ \cdots \ \mathbf{q}_{n_1}^\top \right]^\top \\ &= \frac{1}{\sqrt{n_1 n_2 n_3}} \sum_{i=1}^{n_1} \left(V_{\mathbf{yz},i} \otimes \mathbf{x}_i \right) \mathbf{q}_i = \frac{1}{\sqrt{n_1 n_2 n_3}} \sum_{i=1}^{n_1} \mathrm{vec}\left(\mathbf{x}_i \left(V_{\mathbf{yz},i} \mathbf{q}_i \right)^\top \right) \\ &= \frac{1}{\sqrt{n_1 n_2 n_3}} \mathrm{vec}\left(\sum_{i=1}^{n_1} \mathbf{x}_i \left(V_{\mathbf{yz},i} \mathbf{q}_i \right)^\top \right) \\ &= \frac{1}{\sqrt{n_1 n_2 n_3}} \mathrm{vec}\left(E_{\mathbf{x}} U_{\mathbf{x}} \begin{bmatrix} \left(V_{\mathbf{yz},1} \mathbf{q}_1 \right)^\top \\ \vdots \\ \left(V_{\mathbf{yz},n_1} \mathbf{q}_{n_1} \right)^\top \end{bmatrix} \right). \end{aligned}$$

By the definition of $V_{\mathbf{yz},i}$ in (25), it holds that

$$\begin{aligned} V_{\mathbf{yz},i} \mathbf{q}_i &= \left[\left(E_{\mathbf{z},i+1} U_{\mathbf{z}} \right) \otimes \left(E_{\mathbf{y},i} \mathbf{u}_{\mathbf{y},1} \right) \ \cdots \ \left(E_{\mathbf{z},i+n_2} U_{\mathbf{z}} \right) \otimes \left(E_{\mathbf{y},i} \mathbf{u}_{\mathbf{y},n_2} \right) \right] \mathbf{q}_i \\ &= \sum_{j=1}^{n_2} \left(\left(E_{\mathbf{z},i+1} U_{\mathbf{z}} \right) \otimes \left(E_{\mathbf{y},i} \mathbf{u}_{\mathbf{y},j} \right) \right) \mathbf{q}_{i,j} \\ &= \sum_{j=1}^{n_2} \mathrm{vec}\left(E_{\mathbf{y},i} \mathbf{u}_{\mathbf{y},j} \mathbf{q}_{i,j}^\top \left(E_{\mathbf{z},i+1} U_{\mathbf{z}} \right)^\top \right) \\ &= \mathrm{vec}\left(\sum_{j=1}^{n_2} E_{\mathbf{y},i} \mathbf{u}_{\mathbf{y},j} \left(E_{\mathbf{z},i+1} U_{\mathbf{z}} \mathbf{q}_{i,j} \right)^\top \right). \end{aligned} \tag{28}$$

Let

$$Q_{\mathbf{z},i} \equiv \left[E_{\mathbf{z},i+1} U_{\mathbf{z}} \mathbf{q}_{i,1} \ \ E_{\mathbf{z},i+2} U_{\mathbf{z}} \mathbf{q}_{i,2} \ \cdots \ \ E_{\mathbf{z},i+n_2} U_{\mathbf{z}} \mathbf{q}_{i,n_2} \right]^\top \in \mathbb{C}^{n_2 \times n_3}$$

for $i = 1, \ldots, n_1$. Then Eq. (28) can be rewritten as

$$V_{\mathbf{yz},i} \mathbf{q}_i = \mathrm{vec}\left(E_{\mathbf{y},i} \left[\mathbf{u}_{\mathbf{y},1} \ \cdots \ \mathbf{u}_{\mathbf{y},n_2} \right] Q_{\mathbf{z},i} \right) = \mathrm{vec}\left(E_{\mathbf{y},i} U_{\mathbf{y}} Q_{\mathbf{z},i} \right).$$

We summarize above processes for computing $T\mathbf{q}$ in Algorithm 4.

Acknowledgements

This work is partially supported by the National Science Council, the National Center for Theoretical Sciences, the Taida Institute for Mathematical Sciences, and the Chiao-Da ST Yau Center in Taiwan.

Algorithm 4 Backward FFT-based matrix-vector product $T\mathbf{q}$ [6].

Require: Any vector $\mathbf{q} = \begin{bmatrix}\mathbf{q}_1^\top & \cdots & \mathbf{q}_{n_1}^\top\end{bmatrix}^\top \in \mathbb{C}^n$ with $\mathbf{q}_i = \begin{bmatrix}\mathbf{q}_{i,1}^\top & \cdots & \mathbf{q}_{i,n_2}^\top\end{bmatrix}^\top$ and $\mathbf{q}_{i,j} \in \mathbb{C}^{n_3}$ for $i = 1, \ldots, n_1, j = 1, \ldots, n_2$.
Ensure: The vector $\mathbf{g} \equiv T\mathbf{q}$.
1: **for** $i = 1, \ldots, n_1$ **do**
2: Compute $Q_{\mathbf{z},i} = U_{\mathbf{z}} \begin{bmatrix}\mathbf{q}_{i,1} & \mathbf{q}_{i,2} & \cdots & \mathbf{q}_{i,n_2}\end{bmatrix}$.
3: Update $Q_{\mathbf{z},i} := \begin{bmatrix}E_{\mathbf{z},i+1}Q_{\mathbf{z},i}(:,1) & E_{\mathbf{z},i+2}Q_{\mathbf{z},i}(:,2) & \cdots & E_{\mathbf{z},i+n_2}Q_{\mathbf{z},i}(:,n_2)\end{bmatrix}^\top$.
4: Compute $Q_{\mathbf{y}}(:,:,i) = E_{\mathbf{y},i}\,(U_{\mathbf{y}}Q_{\mathbf{z},i})$
5: **end for**
6: **for** $k = 1, \ldots, n_3$ **do**
7: Compute $Q_{\mathbf{x}} = E_{\mathbf{x}}U_{\mathbf{x}} \begin{bmatrix}Q_{\mathbf{y}}(:,k,1) & Q_{\mathbf{y}}(:,k,2) & \cdots & Q_{\mathbf{y}}(:,k,n_1)\end{bmatrix}^\top$.
8: Set $\mathbf{g}((k-1)n_1n_2 + 1 : kn_1n_2) = \frac{1}{\sqrt{n_1n_2n_3}}\mathrm{vec}(Q_{\mathbf{x}})$.
9: **end for**

References

1. 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.*, 73:727–737, 2004.
2. J. M. Combes, B. Gralak, and A. Tip. Spectral properties of absorptive photonic crystals. In *Proceedings of an AMS-IMS-SIAM Joint Summer Research Conference on Waves in Periodic and Random Media*, 2003.
3. S. Fan, P. R. Villeneuve, and J. D. Joannopoulos. Large omnidirectional band gaps in metallodielectric photonic crystals. *Phys. Rev. B*, 54:11245–11251, 1996.
4. B. T. Holland, C. F. Blanford, and A. Stein. Synthesis of macroporous minerals with highly ordered three-dimensional arrays of spheroidal voids. *Science*, 281:538–540, 1998.
5. T.-M. Huang, W.-J. Chang, Y.-L. Huang, W.-W. Lin, W.-C. Wang, and W. Wang. Preconditioning bandgap eigenvalue problems in three dimensional photonic crystals simulations. *J. Comput. Phys.*, 229:8684–8703, 2010.
6. T.-M. Huang, H.-E. Hsieh, W.-W. Lin, and W. Wang. Eigendecomposition of the discrete double-curl operator with application to fast eigensolver for three dimensional photonic crystals. *SIAM J. Matrix Anal. Appl.*, Accepted.
7. T.-M. Huang, H.-E. Hsieh, W.-W. Lin, and W. Wang. Matrix representation of the double-curl operator for simulating three dimensional photonic crystals. *Math. Comput. Model.*, Accepted.
8. J. D. Joannopoulos, S. G. Johnson, J. N. Winn, and R. D. Meade. *Photonic Crystals: Molding the Flow of Light*. Princeton University Press, 2008.
9. C. Kittel. *Introduction to solid state physics*. Wiley, New York, 2005.
10. P. Kuchment. *Mathematical Modeling in Optical Science*, chapter 7, pages 207–272. SIAM, Philadelphia, 2001.
11. M. Luo and Q. H. Liu. Three-dimensional dispersive metallic photonic crystals with a bandgap and a high cutoff frequency. *J. Opt. Soc. Am. A Opt. Image Sci. Vis.*, 27:1878–1884, 2010.

12. A. Moroz. Three-dimensional complete photonic-band-gap structures in the visible. *Phys. Rev. Lett.*, 83:5274–5277, 1999.
13. J. B. Pendry, A. J. Holden, W. J. Stewart, and I. Youngs. Extremely low frequency plasmons in metallic mesostructures. *Phys. Rev. Lett.*, 76:4773–4776, 1996.
14. A. L. Pokrovsky and A. L. Efros. Electrodynamics of metallic photonic crystals and the problem of left-handed materials. *Phys. Rev. Lett.*, 89:93901, 2002.
15. A. L. Pokrovsky, V. Kamaev, C. Y. Li, Z. V. Vardeny, A. L. Efros, D. A. Kurdyukov, and V. G. Golubev. Theoretical and experimental studies of metal-infiltrated opals. *Phys. Rev. B*, 71:165114, 2005.
16. M. Qi, E. Lidorikis, P. T. Rakich, S. G. Johnson, J. D. Joannopoulos, E. P. Ippen, and H. I. Smith. A three-dimensional optical photonic crystal with designed point defects. *Nature*, 429:538–542, 2004.
17. F. Santosa and H. Ammari. Guided waves in a photonic bandgap structure with a line defect. *SIAM J. Appl. Math.*, 64:2018–2033, 2004.
18. A. Tip. Some mathematical properties of Maxwell's equations for macroscopic dielectrics. *J. Math. Phys.*, 47:012902, 2006.
19. H. van der Lem, A. Tip, and A. Moroz. Band structure of absorptive two-dimensional photonic crystals. *JOSA B*, 20:1334–1341, 2003.
20. H. A. van der Vorst. Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM J. Sci. and Stat. Comput.*, 13:631–644, 1992.
21. K. Yee. Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media. *IEEE Trans. Antennas Propag.*, 14:302–307, 1966.
22. A. A. Zakhidov, R. H. Baughman, Z. Iqbal, C. Cui, I. Khayrullin, S. O. Dantas, J. Marti, and V. G. Ralchenko. Carbon structures with three-dimensional periodicity at optical wavelengths. *Science*, 282:897–901, 1998.