

Plane Wave Expansion Method For Photonic Band Gap

Calculation Using MATLAB®

Manual Version 1.00

Shangping Guo

Department of Electrical & Computer Engineering
 Old Dominion University, Norfolk VA23529, USA

Preface

This software package is developed by Shangping Guo independently during his stay in Department of Electrical & Computer Engineering at Old Dominion University. This is a part of his PhD dissertation work. You can use the software freely, but you have to keep those author informations.

This manual is based on my implementation of PWM(Plane wave method) using matlab. I am not going to list all source files here, they are pretty short and can be easily modified for your own purpose if you know a little bit knowledge about matlab programming. I show how I deal with the 1D, 2D and 3D cases, so you can deal with other problems in any dimensions. You can download this pdf file and related source file from my personal homepage: <http://www.lions.odu.edu/~sguox002>

1. Theory of PWM

Considering electromagnetic wave in materials with:

- Source-free space: $\mathbf{r}(r) = 0, \mathbf{J}(r) = 0$,
- Lossless medium: $\mathbf{e}(r)$ is real in our interested region
- Linear and time-invariant: we can use plane wave theory or Fourier theory
- Magnetic uniform: $\mathbf{m}(r) = \text{const.}$

The full macroscopic Maxwell's equations can be described below:

$$\left\{ \begin{array}{l} \nabla \cdot \vec{D}(r, t) = \mathbf{r}(r, t) \\ \nabla \cdot \vec{B}(r, t) = 0 \\ \nabla \times \vec{H}(r, t) = \frac{\partial \vec{D}(r, t)}{\partial t} + \vec{J}(r, t) \\ \nabla \times \vec{E}(r, t) = -\frac{\partial \vec{B}(r, t)}{\partial t} \end{array} \right. \quad (1)$$

In general, E and H are complicated functions of time and space, using the trick of using complex valued field for mathematical convenience, remembering to take the real part to obtain the physical fields.

Assuming we impose a field oscillating sinusoidally with time, i.e., we can write the mode as:

$$\begin{aligned} \vec{E}(r, t) &= \vec{E}(r) e^{j\omega t} \\ \vec{H}(r, t) &= \vec{H}(r) e^{j\omega t} \end{aligned}$$

Using $D(r) = \mathbf{e}_0 \mathbf{e}(r) E(r)$, $B(r) = \mathbf{m}_0 H(r)$ and $\frac{\partial}{\partial t} \rightarrow j\omega$ (2)

$$\begin{cases} \nabla \cdot \mathbf{e}(r) \vec{E}(r) = 0 \\ \nabla \cdot \vec{H}(r) = 0 \\ \nabla \times \vec{H}(r) = j\omega \mathbf{e}_0 \mathbf{e}(r) \vec{E}(r) \\ \nabla \times \vec{E}(r) = -j\omega \mathbf{m}_0 \vec{H}(r) \end{cases} \quad (3)$$

- $\nabla \cdot \vec{H} = \nabla \cdot \vec{D} = 0$ means 1) there is no charge or current source. 2). H and D are built up of transverse plane waves, i.e., each of the plane wave is perpendicular to its k-vector, i.e. $\vec{k} \cdot \vec{H} = 0$ and $\vec{k} \cdot \vec{D} = 0$, but E is not necessarily transverse. We need to force this transversality to satisfy these two equations.
- D and H field continues everywhere. E is not necessarily continuous.
- Electromagnetic field is essentially vectorial and the vector nature is very important for us to obtain the correct results.

For H, we get the next eigen value equation:

$$\nabla \times \frac{1}{\mathbf{e}(r)} \nabla \times \vec{H}(r) = \frac{\omega^2}{c^2} \vec{H}(r) \quad (4)$$

For E, we get the next eigen value equation:

$$\nabla \times \nabla \times \vec{E}(r) = \frac{\omega^2}{c^2} \mathbf{e}(r) \vec{E}(r) \quad (5)$$

For D, we get the following eigen value equation:

$$\nabla \times \nabla \times \frac{1}{\mathbf{e}(r)} \vec{D}(r) = \frac{\omega^2}{c^2} \vec{D}(r) \quad (6)$$

For $\mathbf{e}(r)$ is a highly discontinuous function, and E will be also discontinuous, so E is not proper for calculation. H and D are convenient for calculations. Once one is solved, the other one can be obtained by the following relation:

$$\vec{E}(r) = \frac{1}{j\omega \mathbf{e}_0 \mathbf{e}(r)} \nabla \times \vec{H}(r) \quad (7)$$

$$\vec{H}(r) = \frac{1}{-j\omega \mathbf{m}_0} \nabla \times \vec{E}(r) \quad (8)$$

It can be seen, for E is discontinuous, it is much more convenient to use H to get E, instead of using E to get H. So we use the H eigen equation (4) and (7) to solve this problem.

According to J D Joannopoulos, (4) has some important features, and more important, it holds for general condition, requiring only real dielectric function:

1. The operator $\Xi = \nabla \times \frac{1}{\mathbf{e}} \nabla \times$ is a linear operator, if H_1 is a mode and H_2 is also a mode, then $H_1 + H_2$ is also a mode.
2. The operator is an Hermitian operator, i.e., $(f, \Xi g) = (\Xi f, g)$

3. The operator must have real eigen values only if the dielectric function is positive everywhere. This gives the theory support that our algorithm will get real eigen values.
4. Any two different modes will be orthogonal, i.e., $(H_i, H_j) = \mathbf{d}_{ij}$. Degenerate modes are not necessary orthogonal. This is also a way to determine if a mode is the same as other mode. Degenerate modes are generally caused by system symmetry.
5. Mode energy: $E_D = \frac{1}{8\mathbf{p}} \int \frac{1}{\mathbf{e}(r)} |D(r)|^2 dr$ and $E_H = \frac{1}{8\mathbf{p}} \int |H(r)|^2 dr$
6. Symmetry property in crystal. Symmetry operator commutes with the magnetic operator and means that mode and mode operated by symmetry operation will be the same.
7. Scaling properties, those properties can be extended to any frequency region. Note here we assume the dielectric constants of these materials in these regions are constants. This is valid for most optical materials in the visible and infrared region.

2. Periodic dielectric function

So far, nothing is related with the periodic property. In periodic structure, according to Bloch theorem, a plane wave in a periodic structure will be modulated by the periodicity. So, the complex-valued magnetic field \mathbf{H} can be expressed as:

$$\begin{aligned}\vec{H}(r) &= e^{i\vec{k}\cdot\vec{r}} h(\vec{r}) \hat{e}_k \\ h(\vec{r}) &= h(\vec{r} + \vec{R}_l)\end{aligned}\quad (9)$$

where R_l is an arbitrary lattice vector, \hat{e}_k is an unit vectors perpendicular to the vector \vec{k} and parallel to the \mathbf{H} vector. We use the description of semiconductor crystal in solid-state physics here.

Any periodic function can be expanded using Fourier series, so is for \mathbf{e} and h :

$$\mathbf{e}(r) = \sum_{G_i} \mathbf{e}(\vec{G}_i) e^{i\vec{G}_i\cdot\vec{r}} \quad (10)$$

$$\frac{1}{\mathbf{e}(r)} = \sum_{\vec{G}_i} \mathbf{e}^{-1}(\vec{G}_i) e^{i\vec{G}_i\cdot\vec{r}} \quad (11)$$

and

$$h(r) = \sum_{G_i} h(G_i) e^{i\vec{G}_i\cdot\vec{r}} \quad (12)$$

$$\text{So,} \quad \vec{H}(r) = \hat{e}_k e^{i\vec{k}\cdot\vec{r}} \sum_{G_i} h(G_i) e^{i\vec{G}_i\cdot\vec{r}} = \sum_{G_i, \mathbf{I}} h(G_i, \mathbf{I}) e^{i(\vec{k}+\vec{G}_i)\cdot\vec{r}} \hat{e}_{1, \vec{k}+\vec{G}_i} \quad (13)$$

In (13), the transverse property is used to decompose the wave into sum of a set of plane waves, which is the heart of this method. $e_{1, \vec{k}+\vec{G}}$ is the unit vector perpendicular to $\vec{k} + \vec{G}$ and $h(\vec{G}_i, \mathbf{I})$ is the component amplitude along the unit vector. $\vec{k} + \vec{G}$, $\hat{e}_{1, \vec{k}+\vec{G}}$, $\hat{e}_{2, \vec{k}+\vec{G}}$ forms a Cartesian triad. G_i is the frequency component, λ represents the two axes in the

perpendicular plane and \hat{e}_l is the unit vector along the axis. All the G_i forms the reciprocal lattice. All the R_l forms the crystal lattice:

$$\vec{R}_l = l_1 \vec{a}_1 + l_2 \vec{a}_2 + l_3 \vec{a}_3 \quad (14)$$

l_1, l_2, l_3 are any integers, $\vec{a}_1, \vec{a}_2, \vec{a}_3$ are called the primitive lattice vectors to describe the lattice.

$$\vec{G}_h = h_1 \vec{b}_1 + h_2 \vec{b}_2 + h_3 \vec{b}_3 \quad (15)$$

where $\vec{b}_1, \vec{b}_2, \vec{b}_3$ are called the primitive reciprocal lattice vectors and h_1, h_2, h_3 are any integers.

The two sets of base vectors are orthogonal to each other, and they are related by:

$$\vec{b}_i \cdot \vec{a}_j = 2\pi \mathbf{d}_{ij} \quad (16)$$

$$\vec{b}_1 = \frac{2\pi}{V} \vec{a}_2 \times \vec{a}_3, \vec{b}_2 = \frac{2\pi}{V} \vec{a}_3 \times \vec{a}_1, \vec{b}_3 = \frac{2\pi}{V} \vec{a}_1 \times \vec{a}_2 \quad (17)$$

Where V is the volume, $V = \vec{a}_1 \cdot \vec{a}_2 \times \vec{a}_3$

The relation for 2D and 1D can be simplified using the above expressions.

For 2D problems, $\vec{a}_3 = \hat{e}_z$ and can be omitted from the above equations.

$\vec{a}_1 = (x_1, y_1), \vec{a}_2 = (x_2, y_2)$, the reciprocal base vector can be expressed as:

$$\vec{b}_1 = \frac{2\pi}{A} (y_2, -x_2), \vec{b}_2 = \frac{2\pi}{A} (y_1, -x_1), A \text{ is the area: } A = |\vec{a}_1 \times \vec{a}_2|.$$

For 1D problems, $\vec{a}_2 = \hat{e}_y, \vec{a}_3 = \hat{e}_z, \vec{a}_1 = a\hat{e}_x$, and $\vec{b}_1 = \frac{2\pi}{a} \hat{e}_x$.

Substitute (13) and (11) into eigen equation (4):

$$\nabla \times \sum_{\vec{G}_i} \mathbf{e}^{-i\vec{G}_i \cdot \vec{r}} \nabla \times \sum_{\vec{G}_i, \mathbf{I}} h(\vec{G}_i, \mathbf{I}) e^{i(\vec{k} + \vec{G}_i) \cdot \vec{r}} \hat{e}_{1, \vec{k} + \vec{G}_i} = \frac{w^2}{c^2} \sum_{\vec{G}_i, \mathbf{I}} h(\vec{G}_i, \mathbf{I}) e^{i(\vec{k} + \vec{G}_i) \cdot \vec{r}} \hat{e}_{1, \vec{k} + \vec{G}_i} \quad (18)$$

Move the $\nabla \times$ operator into the sum:

$$\nabla \times \sum_{\vec{G}_i} \mathbf{e}^{-i\vec{G}_i \cdot \vec{r}} \sum_{\vec{G}_i, \mathbf{I}} h(\vec{G}_i, \mathbf{I}) \nabla \times e^{i(\vec{k} + \vec{G}_i) \cdot \vec{r}} \hat{e}_{1, \vec{k} + \vec{G}_i} = \frac{w^2}{c^2} \sum_{\vec{G}_i, \mathbf{I}} h(\vec{G}_i, \mathbf{I}) e^{i(\vec{k} + \vec{G}_i) \cdot \vec{r}} \hat{e}_{1, \vec{k} + \vec{G}_i} \quad (19)$$

$$\text{Since } \nabla \times u \vec{A} = \nabla u \times \vec{A} + u \nabla \times \vec{A} \quad (20)$$

And the second item for r-independent vector is zero:

$$\nabla \times \sum_{\vec{G}_i} \mathbf{e}^{-i\vec{G}_i \cdot \vec{r}} \sum_{\vec{G}_i, \mathbf{I}} h(\vec{G}_i, \mathbf{I}) \nabla e^{i(\vec{k} + \vec{G}_i) \cdot \vec{r}} \times \hat{e}_{1, \vec{k} + \vec{G}_i} = \frac{w^2}{c^2} \sum_{\vec{G}_i, \mathbf{I}} h(\vec{G}_i, \mathbf{I}) e^{i(\vec{k} + \vec{G}_i) \cdot \vec{r}} \hat{e}_{1, \vec{k} + \vec{G}_i} \quad (21)$$

$$\text{Since } \nabla e^{i\vec{k} \cdot \vec{r}} = i\vec{k} e^{i\vec{k} \cdot \vec{r}} \quad (22)$$

$$\nabla \times \sum_{\vec{G}_i} \mathbf{e}^{-i\vec{G}_i \cdot \vec{r}} \sum_{\vec{G}_i, \mathbf{I}} h(\vec{G}_i, \mathbf{I}) e^{i(\vec{k} + \vec{G}_i) \cdot \vec{r}} i(\vec{k} + \vec{G}_i) \times \hat{e}_{1, \vec{k} + \vec{G}_i} = \frac{w^2}{c^2} \sum_{\vec{G}_i, \mathbf{I}} h(\vec{G}_i, \mathbf{I}) e^{i(\vec{k} + \vec{G}_i) \cdot \vec{r}} \hat{e}_{1, \vec{k} + \vec{G}_i} \quad (23)$$

$$\nabla \times \sum_{\vec{G}_i, \mathbf{I}} \sum_{\vec{G}'_i} \mathbf{e}^{-i\vec{G}'_i \cdot \vec{r}} h(\vec{G}_i, \mathbf{I}) e^{i(\vec{k} + \vec{G}_i) \cdot \vec{r}} i(\vec{k} + \vec{G}_i) \times \hat{e}_{1, \vec{k} + \vec{G}_i} = \frac{w^2}{c^2} \sum_{\vec{G}_i, \mathbf{I}} h(\vec{G}_i, \mathbf{I}) e^{i(\vec{k} + \vec{G}_i) \cdot \vec{r}} \hat{e}_{1, \vec{k} + \vec{G}_i} \quad (24)$$

Move the $\nabla \times$ operator into the sum again:

$$\sum_{\vec{G}_i, \mathbf{I}} \sum_{\vec{G}'_i} h(\vec{G}_i, \mathbf{I}) \mathbf{e}^{-1}(\vec{G}'_i) \nabla \times \left[e^{i(\vec{k} + \vec{G}_i + \vec{G}'_i) \cdot \vec{r}} i(\vec{k} + \vec{G}_i) \times \hat{e}_{1, \vec{k} + \vec{G}_i} \right] = \frac{w^2}{c^2} \sum_{\vec{G}_i, \mathbf{I}} h(\vec{G}_i, \mathbf{I}) e^{i(\vec{k} + \vec{G}_i) \cdot \vec{r}} \hat{e}_{1, \vec{k} + \vec{G}_i} \quad (25)$$

Write $\vec{G}_i + \vec{G}'_i \rightarrow \vec{G}'_i$, then

$$\sum_{\vec{G}_i, \mathbf{I}} \sum_{\vec{G}'_i - \vec{G}_i} h(\vec{G}_i, \mathbf{I}) \mathbf{e}^{-1}(\vec{G}'_i - \vec{G}_i) \nabla \times \left[e^{i(\vec{k} + \vec{G}_i) \cdot \vec{r}} i(\vec{k} + \vec{G}_i) \times \hat{e}_{1, \vec{k} + \vec{G}_i} \right] = \frac{w^2}{c^2} \sum_{\vec{G}_i, \mathbf{I}} h(\vec{G}_i, \mathbf{I}) e^{i(\vec{k} + \vec{G}_i) \cdot \vec{r}} \hat{e}_{1, \vec{k} + \vec{G}_i} \quad (26)$$

$$\sum_{\vec{G}_i, \mathbf{I}} \sum_{\vec{G}'_i - \vec{G}_i} h(\vec{G}_i, \mathbf{I}) \mathbf{e}^{-1}(\vec{G}'_i - \vec{G}_i) e^{i(\vec{k} + \vec{G}_i) \cdot \vec{r}} i(\vec{k} + \vec{G}_i) \times \left[i(\vec{k} + \vec{G}_i) \times \hat{e}_{1, \vec{k} + \vec{G}_i} \right] = \frac{w^2}{c^2} \sum_{\vec{G}_i, \mathbf{I}} h(\vec{G}_i, \mathbf{I}) e^{i(\vec{k} + \vec{G}_i) \cdot \vec{r}} \hat{e}_{1, \vec{k} + \vec{G}_i} \quad (27)$$

Use the transverse property, the above equation can be written as:

$$\sum_{\vec{G}_i, \mathbf{I}} \sum_{\vec{G}'_i - \vec{G}_i} \sum_{\mathbf{I}'} h(\vec{G}_i, \mathbf{I}') \mathbf{e}^{-1}(\vec{G}'_i - \vec{G}_i) e^{i(\vec{k} + \vec{G}_i) \cdot \vec{r}} \left[(\vec{k} + \vec{G}_i) \times \hat{e}_{1, \vec{k} + \vec{G}_i} \right] \times (\vec{k} + \vec{G}'_i) = \frac{w^2}{c^2} \sum_{\vec{G}_i, \mathbf{I}} h(\vec{G}_i, \mathbf{I}) e^{i(\vec{k} + \vec{G}_i) \cdot \vec{r}} \hat{e}_{1, \vec{k} + \vec{G}_i} \quad (28)$$

$$\left[(\vec{k} + \vec{G}_i) \times \hat{e}_{1, \vec{k} + \vec{G}_i} \right] \times (\vec{k} + \vec{G}'_i) = \left\{ \left[(\vec{k} + \vec{G}_i) \times \hat{e}_{1, \vec{k} + \vec{G}_i} \right] \cdot \left[(\vec{k} + \vec{G}'_i) \times \hat{e}_{1, \vec{k} + \vec{G}_i} \right] \right\} \hat{e}_{1, \vec{k} + \vec{G}_i} \quad (29)$$

After equating the items of the right hand side and left hand side, the final equation is listed below:

$$\sum_{\vec{G}', \mathbf{I}'} \left[(\vec{k} + \vec{G}') \times \hat{e}_1 \right] \cdot \left[(\vec{k} + \vec{G}') \times \hat{e}'_1 \right] \mathbf{e}^{-1}(\vec{G}' - \vec{G}) h(\vec{G}', \mathbf{I}') = \frac{w^2}{c^2} h(\vec{G}, \mathbf{I}) \quad (30)$$

where $\mathbf{I}' = 1, 2$, and \vec{G} is any reciprocal lattice vector.

The equivalent form of (30) is:

$$\sum_{\vec{G}'} \left[\vec{k} + \vec{G}' \right] \left[\vec{k} + \vec{G}' \right] \mathbf{e}^{-1}(\vec{G} - \vec{G}') \begin{bmatrix} \hat{e}_2 \hat{e}'_2 & -\hat{e}_2 \hat{e}'_1 \\ -\hat{e}_1 \hat{e}'_2 & \hat{e}_1 \hat{e}'_1 \end{bmatrix} \begin{bmatrix} h'_1 \\ h'_2 \end{bmatrix} = \frac{w^2}{c^2} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \quad (31)$$

This is a standard eigen value problem, which can be implemented by numerical method. However, we cannot use infinite number of \vec{G} vectors and we can only use finite number of them. This is similar to the truncation of high frequency components in signal processing, if the number of frequencies is large enough, we can approach the true signal very well. Here, we also use finite number of \vec{G} vectors, corresponding to those lower spatial frequencies.

For N \vec{G} vectors (or N grid points in the reciprocal space are included), this is an $2N$ equations with $2N$ unknown. In matrix form, the coefficient matrix dimension will be $2N$ by $2N$.

The matrix form is too large to fit here, just pick the i th equation. Assuming we have chosen N \vec{G} vectors and named them as $\vec{G}_1, \vec{G}_2 \dots \vec{G}_i \dots \vec{G}_N$:

$$\begin{aligned}
 & [(k + G_i) \times e_1(G_i)] \bullet [(k + G_1) \times e_1(G_1)] e^{-1}(G_i - G_1) h_1(G_1) + \\
 & [(k + G_i) \times e_1(G_i)] \bullet [(k + G_2) \times e_1(G_2)] e^{-1}(G_i - G_2) h_1(G_2) + \\
 & \dots + \\
 & [(k + G_i) \times e_1(G_i)] \bullet [(k + G_N) \times e_1(G_N)] e^{-1}(G_i - G_N) h_1(G_N) + \\
 & [(k + G_i) \times e_1(G_i)] \bullet [(k + G_1) \times e_2(G_1)] e^{-1}(G_i - G_1) h_2(G_1) + \\
 & [(k + G_i) \times e_1(G_i)] \bullet [(k + G_2) \times e_2(G_2)] e^{-1}(G_i - G_2) h_2(G_2) + \\
 & \dots +
 \end{aligned} \tag{32}$$

$$\begin{aligned}
 & [(k + G_i) \times e_1(G_i)] \bullet [(k + G_N) \times e_2(G_N)] e^{-1}(G_i - G_N) h_2(G_N) = \frac{w^2}{c^2} h_1(G_i) \\
 & [(k + G_i) \times e_2(G_i)] \bullet [(k + G_1) \times e_1(G_1)] e^{-1}(G_i - G_1) h_1(G_1) + \\
 & [(k + G_i) \times e_2(G_i)] \bullet [(k + G_2) \times e_1(G_2)] e^{-1}(G_i - G_2) h_1(G_2) + \\
 & \dots + \\
 & [(k + G_i) \times e_2(G_i)] \bullet [(k + G_N) \times e_1(G_N)] e^{-1}(G_i - G_N) h_1(G_N) + \\
 & [(k + G_i) \times e_2(G_i)] \bullet [(k + G_1) \times e_2(G_1)] e^{-1}(G_i - G_1) h_2(G_1) + \\
 & [(k + G_i) \times e_2(G_i)] \bullet [(k + G_2) \times e_2(G_2)] e^{-1}(G_i - G_2) h_2(G_2) + \\
 & \dots +
 \end{aligned} \tag{33}$$

$$[(k + G_i) \times e_2(G_i)] \bullet [(k + G_N) \times e_2(G_N)] e^{-1}(G_i - G_N) h_2(G_N) = \frac{w^2}{c^2} h_2(G_i)$$

In matrix form $[\mathbf{M}][\mathbf{H}] = \mathbf{L}[\mathbf{H}]$, it would be:

$$\begin{bmatrix} m_{11} & m_{12} & \cdots & \cdots & m_{1,2N} \\ m_{21} & m_{22} & \cdots & \cdots & m_{2,2N} \\ \vdots & \ddots & & & \vdots \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ m_{2N,1} & m_{2N,2} & \cdots & \cdots & m_{2N,2N} \end{bmatrix} \begin{bmatrix} h_1(G_1) \\ \vdots \\ h_N(G_N) \\ h_2(G_1) \\ \vdots \\ h_2(G_N) \end{bmatrix} = \frac{w^2}{c^2} \begin{bmatrix} h_1(G_1) \\ \vdots \\ h_N(G_N) \\ h_2(G_1) \\ \vdots \\ h_2(G_N) \end{bmatrix} \tag{34}$$

So far, the problem is solved from general meaning, but there is still a way from implementing it on computer, also, there can be great simplification for some special case.

Here is the general step according to above derivation:

1. given a k point to solve the associating frequency w .
2. Find the reciprocal lattice and choose the reciprocal vector G set.
3. Find each of the Fourier transform coefficient $e^{-1}(G - G')$
4. Find the $\vec{k} + \vec{G}$ and the two unit vector sets.
5. Form the matrix M .
6. Calculate the eigen value.

3. Forming the band structure

k - ω relation for periodic structure is called the band structure. In periodic structure, according to Bloch's theorem, the relation is also periodic, so it should be enough to only calculate only those k -points in the so-called first Brillouin zone. All other k 's outside the 1BZ are the same mode as a k in 1BZ.

Using the symmetry properties, we only need to calculate k - ω relation in a zone with no symmetry property called irreducible Brillouin zone IBZ.

In the IBZ zone, we give a k point, and will get a series of eigen values, i.e., a set of ω points. Continuously changing the k points, we'll get continuously changed ω points for the same band. So usually, we only calculate those points on the IBZ edge, the boundary value, since those other frequencies for those k -points inside the IBZ will fall inside the band region.

4. Numerical considerations

Fourier transform and eigen solver are the two fundamental problems to solve. To understand better about them helps a lot in doing numerical analysis.

Fourier transform of the dielectric function:

The dielectric function can be 1D, 2D or 3D. Some points are very important:

1. When Fourier transform is performed, selection of the integration region is important, i.e., how to select unit cell will affect Fourier transform. Generally, if it has some symmetry properties, like inversion symmetry, the Fourier transform will be pure real.
2. If inversion symmetry cannot be satisfied, Fourier transform would be complex.
3. The rearranged FT matrix $\epsilon^{-1}(G - G')$ should be a symmetric matrix. If FT is pure real, then it is a real symmetric matrix, else it is a complex symmetric matrix.
4. The final form $[M][H]=\Lambda[H]$, whether $[M]$ is symmetric or not, the eigen value should be real, but the eigen vector will not necessarily to be real.
5. The final field, from experiment, only take the real part is proven to be not enough. There is a random phase for the real and imaginary part. I found a way to fix this phase.

Implementation details

See the paper: *A simple plane wave implementation method for photonic crystal calculations*, Shangping Guo, Sarcharia Albin. In this paper, the stability and mathematics behind the implementation is discussed.

5. Examples and source code samples

One-dimensional problems

Typical problem: Multilayer system, including the FBG case.

Normal incidence:

K and G have only two directions +x and -x, and \hat{e}_1, \hat{e}_2 can be constantly set to: $\hat{e}_1 = \hat{y}$, $\hat{e}_2 = \hat{z}$,

and the eigen equation can be simplified to a N by N equation group:

$$\sum_{G'} |k + G| |k + G'| e^{-i(\vec{G} - \vec{G}') \cdot \vec{r}} h(\vec{G}') = \frac{\omega^2}{c^2} h(\vec{G}) \quad (35)$$

In this case, D and H are in the y-z plane and TE and TM are the same.

Here is the source file in *d:\pbg\1d\photo1d3.m*

```

%%% this program is designed for solving the one-dimensional photonic structure
%%% for 1D multilayer structure using data in book, it get correct results.
%%% Photonic crystals, J D Joannopolous 1995, p42
clear
warning off
epsa=13;
epsb=1;
a=1.0;
R=0.2*a;
al=a;
bl=2*pi/a;
n=input('please input n: ');
n=24;
NumberofPW=(2*n+1);

count=1;
G=(-n:n).*bl;
r=(-n:n);

N=2*NumberofPW+1;
m=floor(N*R/a)+1;
%%%this is very important to expand the epsilon function like this
%%%even expansion, so that the imag part is zero
eps1=[epsb*ones((N-m)/2,1);epsa*ones(m,1);epsb*ones((N-m)/2,1)];
eps20=(fftshift(fft(eps1)./N));

%%Now we get the FFT matrix for the dielectric function.
%%We need to get the E(G-G') matrix
%%this matrix include the frequency all we need

for x=1:NumberofPW,
    for y=x:NumberofPW,
        b=r(x)-r(y)+(2*n+1)+1;
        eps2(x,y)=eps20(b);
        eps2(y,x)=eps2(x,y);
    end
end

k0=-pi/a:2*pi/a/30:pi/a;
eps2=inv(eps2);
counter=1;

for ii=1:length(k0),
    k=k0(ii);
    M=abs((k+G')*(k+G)).*(eps2);
    E=sort(abs(eig(M)));
    freq(:,counter)=sqrt(abs(E(1:10))).*a./2./pi;
    display(sprintf('calculation of k=%f is finished',k));
    counter=counter+1;
end

tmpx=1:length(k0);
plot(tmpx,freq,'linewidth',2)
title('Band structure of a 1D square photonic band structure')
xlabel('wave vector')
ylabel('wa/2\pic')

```



```
grid on
axis([1 31 0 1])
```

The calculation is pretty fast and is done in just one minute. The final results are in the following graph, note the frequency is normalized according to $\frac{wa}{2pc}$ or equally $\frac{a}{l}$:

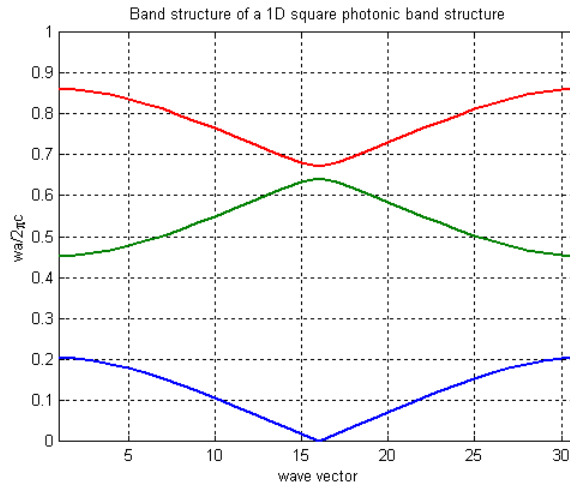


Figure 1: Band structure for a 1D multilayer system

The comments in the program should help the understanding, since the program is pretty short. Here is some more brief explanation of the program:

epsa is the dielectric constants for the atom;

epsb is the dielectric constants for the background material;

a is the actual period of the PBG material;

R is the width of the atom in a period;

N is the number of grids in the reciprocal space along the +x direction (the same number is used for -x direction);

G is the grid vectors in the reciprocal space;

eps1 is the matrix containing the dielectric function in one period in the real space;

eps20 is the Fourier transformed matrix of *eps1*;

eps2 is the matrix $e^{-1}(G - G')$.

Note in the step to obtain $e^{-1}(G - G')$ from the FT matrix, I used a small trick. The FFT matrix contains all the frequency components for $e^{-1}(G - G')$ and I just picked them all out.

The [M] matrix formation is a little bit trick in MATLAB. The .* operation is very useful, and the details are not discussed here. But the expression correctly formed the matrix, the final results proves this.

Non-normal incidence:

k has y-z component, the problem is like a 2D problem, the TE and TM is different. Conventionally, it is called *s* and *p* polarization.

This can be used to analyze the FBG and TFBG, note that FBG is a kind of photonic crystal in x direction and light is bound by TIR in the transverse direction. We can get some hints from FBG. TFBG corresponds to the non-normal incidence PBG case.

Some other important 1D cases are very important, like the air-bridge case. This can be done by a two dimensional program.

I am not going to list the source file here, you can download these files from my website or just modify the 2D TM/TE program for off-plane applications.

Band structure for off-plane propagation for TE: *d:\pbg\1d\offplane1dte.m*

Band structure for off-plane propagation for TM: *d:\pbg\1d\offplane1dtm.m*

Chigrin's 1D omni-reflection structure simulation: *d:\pbg\1d\gapkyte.m*,
d:\pbg\1d\gapkytm.m, see *IEEE Journal of lightwave technology*, Nov. 1999.

Two-dimensional problems

In-plane propagation:

K and G are in the same x-y plane, so K+G is also in x-y plane, and one of the unit vectors can be constantly set to: $\hat{e}_{2,k+G} = \hat{z}$, the other unit vector $\hat{e}_{1,k+G}$ is also in the x-y plane. This also leads some important simplifications:

$\hat{e}_1 = \langle \cos \mathbf{q}, \sin \mathbf{q}, 0 \rangle, \hat{e}_2 = \langle 0, 0, 1 \rangle$, then:

$$\begin{bmatrix} \hat{e}_2 \hat{e}'_2 & -\hat{e}_2 \hat{e}'_1 \\ -\hat{e}_1 \hat{e}'_2 & \hat{e}_1 \hat{e}'_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & \cos(\mathbf{q} - \mathbf{q}') \end{bmatrix} \quad (36)$$

i.e., the 2N by 2N equation group is decoupled into two equation groups: the TE and TM. One group only contains *H* components along \hat{e}_1 , no components along $\hat{e}_2 = \hat{z}$, and so is called TM wave. The other group only contains *H* component along $\hat{e}_2 = \hat{z}$, and E has no component along z, so is called TE wave.

$$\text{TM:} \quad \sum_{\vec{G}'} |\vec{k} + \vec{G}'| |\vec{k} + \vec{G}'| \mathbf{e}^{-1} (\vec{G} - \vec{G}') h_1(\vec{G}') = \frac{w^2}{c^2} h_1(\vec{G}) \quad (37)$$

$$\text{TE:} \quad \sum_{\vec{G}'} |\vec{k} + \vec{G}'| |\vec{k} + \vec{G}'| \mathbf{e}^{-1} (\vec{G} - \vec{G}') (\hat{e}_1 \cdot \hat{e}'_1) h_2(\vec{G}') = \frac{w^2}{c^2} h_2(\vec{G}) \quad (38)$$

The equation for TE can be rewritten as:

$$\sum_{\vec{G}'} (\vec{k} + \vec{G}') \cdot (\vec{k} + \vec{G}') \mathbf{e}^{-1} (\vec{G} - \vec{G}') h_2(\vec{G}') = \frac{w^2}{c^2} h_2(\vec{G}) \quad (39)$$

which makes the two unit vectors set unnecessary.

Example 1: Square lattice: *d:\pbg\2d\square\squideal.m*

```
%%          2D PBG: Square lattice
%%          Shangping Guo
%%          squideal.m
clear all
warning off;
epsa=8.9; %material for the atom, alumina
epsb=1;   %material of the background
a=1.0;    %lattice constant
R=0.20*a; %radius of the 'atom' or the radius of the cylinder
i=sqrt(-1);
```

```

f=pi*R^2/a^2; %volume fraction of atom(s) in the cell
NumberofCell=1; %supercell size=1X1
a1=a;
a2=a*i;
b1=2*pi/a/NumberofCell;
b2=2*pi/a/NumberofCell*i;
n=input('please input n:');
display('Fourier transforming.....');
tic;
NumberofPW=(2*n+1)^2;
mind=(-2*n:2*n)'+2*n+1;
mind=mind(:,ones(1,size(mind)))-2*n-1;
GG=mind'*b1+mind*b2;
%clear mind;
eps2l=2*f*(epsa-epsb)*besselj(1,abs(GG).*R)./(abs(GG).*R);
eps2l(2*n+1,2*n+1)=epsb+f*(epsa-epsb);
%zz=[0,0]*[a1 a2].';%use 1X1 supercell to verify the algorithm
%5X5 super cell
zz=[
-2 -2;-2 -1;-2 0;-2 1;-2 2;
-1 -2;-1 -1;-1 0;-1 1;-1 2;
 0 -2; 0 -1;      0 1; 0,2;
 1 -2; 1 -1; 1 0; 1 1; 1 2;
 2,-2; 2,-1; 2 0; 2,1; 2,2]*[a1 a2].';
zz=[0,0]*[a1 a2].';%use 1X1 supercell to verify the algorithm
%zz=[0 0; 0 1;1 0;1 1]*[a1 a2].';%2X2 supercell
%zz=[-1 -1;-1 0;-1 1;0 -1;0 0;0 1;1 -1;1 0;1 1]*[a1 a2].';
eps20=zeros(length(eps2l));
for x=1:length(zz),
    eps20=eps20+exp(i*(real(GG).*real(zz(x))+imag(GG).*imag(zz(x))))).*eps2l;
end
ff=pi*R^2*length(zz)/(NumberofCell^2*a^2);
eps20=eps20./NumberofCell^2;
eps20(2*n+1,2*n+1)=epsb+ff*(epsa-epsb);
count=1;
for y=-n:n,
    for x=-n:n;
        G(count)=x*b1+y*b2;
        r(count,:)= [x,y];
        count=count+1;
    end
end
display('Building eps(G,G) matrix from the FFT matrix');
for x=1:NumberofPW,
    for y=x:NumberofPW,
        b=r(x,:)-r(y,:)+2*n+1;
        eps2(x,y)=eps20(b(2),b(1));
        eps2(y,x)=eps2(x,y);
    end
end
k1=2*pi/a*0.5.*(0:0.1:1);;
k2=2*pi/a*(0.5+(0.1:0.1:1).*0.5*i);
k3=2*pi/a*(0.5+0.5*i).*(0.9:-0.1:0);
k0=[k1 k2 k3]./NumberofCell;

display('Now calculate the eigen values..');
eps2=inv(eps2);
if max(max(real(eps2))) > 10^7*max(max(imag(eps2)))
display('Your lattice is inversion symmetric');
eps2=real(eps2);
else
display('Imaginary part of FFT is not zero');
stop;
%here we only demonstrate the inversion symmetric case
%however, nonsymmetric case is also supported
end
options disp=0;
    
```

```

counter=1;
for ii=1:length(k0),
    k=k0(ii);
    %k=k0;
    M=(real(k+G.)*real(k+G)+imag(k+G.)*imag(k+G)).*eps2; %TE wave
    %M=abs(k+G.)*abs(k+G).*eps2; %TM wave
    E=sort(abs(eig(M)));
    %E=abs(eigs(M,20,'sm',options));%used to calculate only several smallest eigenvalues
    freq(:,counter)=sqrt(abs(E)).*a./2./pi;
    display(sprintf('calculation of k=%f+%fi is finished',real(k),imag(k)));
    counter=counter+1;
end
toc;
[max(freq(1,:)),min(freq(2,:))]
tmpx=1:length(k0);
plot(tmpx,freq,'o-','linewidth',2)
title('TM:Band structure of a 2D square PBG with a point defect (5X5)')
xlabel('wave vector')
ylabel('wa/2\pic')
grid on
axis([1 31 0 1])
%tmp=ifft2(ifftshift(eps20));
%surf(fftshift(real(tmp)),shading interp,view(2))
%to get the lattice, use tmp=ifft2(ifftshift(eps20));
    
```

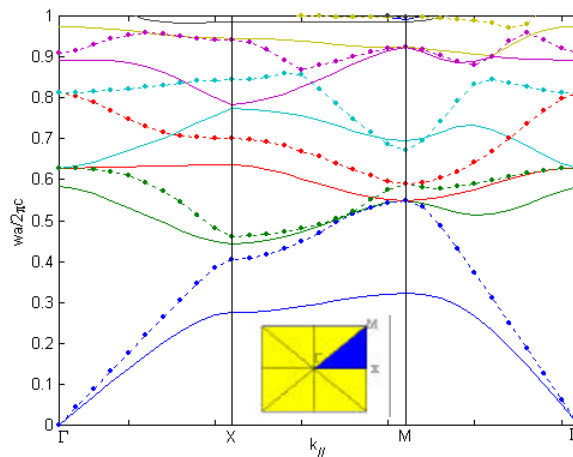


Figure 2: TE/TM band structure for a 2D square lattice with alumina rods in air, the solid line represents TM, the dotted line represents TE mode.

Note to the program:

Similar to the 1D program, there is some modifications for this 2D implementations.

It supports supercell calculations and finds a successful way to treat them the same way.

In the program, I gave the position of those individual atoms in a supercell in terms of the primitive vectors. If atom positions cannot be expressed as sum of the two primitive vectors, minor modifications need be used and is not shown here.

Here, for convenience, the vector in 2D is represented by a complex number: the real part represents the x-component and the imaginary part represents the y-component.

How to get the Fourier transform is the key of this program. Here I used the analytical form for the circular cylinder and used the shift property to get the Fourier transform for more complicated system consisting of finite number of individual ‘atoms’.

Density of states (DOS)

The band structure shows all possible frequency for various k-vectors. Density of states shows how many k-vectors will correspond to a frequency. This is calculated the number of k-vectors in the 1BZ for a certain frequency. Note this is easy for a square lattice, but for other geometry, how to divide the 1BZ and reserve the system symmetry is a problem. Here I only showed the graph.

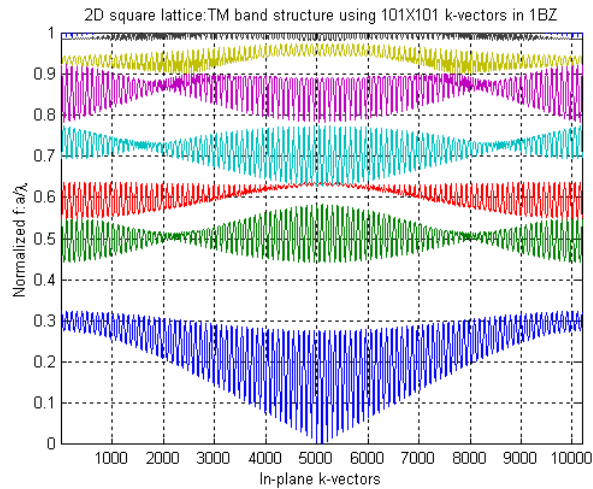


Figure 3: Band structure using 10100 k-vectors in 1BZ for a 2D square lattice with alumina rods in air

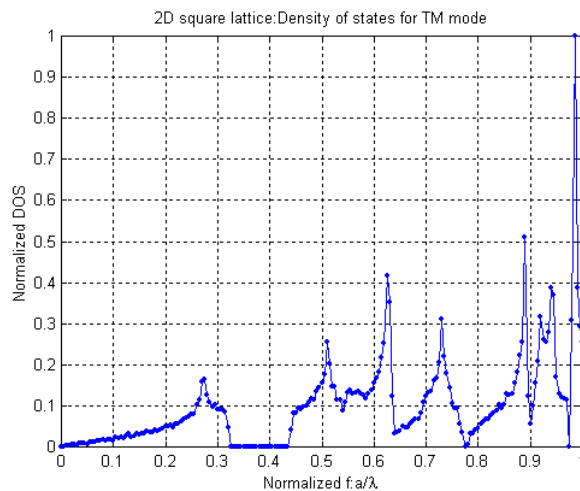


Figure 4: Density of states for a 2D square lattice with alumina rods in air

Next we show the same program can be modified a little bit to solve the triangular lattice.

Example2: Triangular lattice with air holes in Si or GaAs material

Implementation of 2D PBG is of great meaning. Here is a typical example of triangular lattice: [d:\ksp\pbg\triangle\photo2dtm.m](#). dielectric constants are 1 and 13, $R=0.48a$.

```
% this program is designed for solving the two dimensional photonic structure
% we are using the parameters from one of the paper
clear
warning off
```

```

epsa=1;
epsb=13;
a=1.0;
%f=0.8358;
%R=sqrt(sqrt(3).*f/(2*pi)*a^2);
R=0.48*a;
i=sqrt(-1);
f=2*pi/sqrt(3)*R^2/a^2;
b1=2*pi/a*(1-sqrt(3)/3*i);
b2=2*pi/a*2*sqrt(3)/3*i;
n=input('please input n: ');
NumberofPW=(2*n+1)^2;

count=1;
for x=-n:n,
    for y=-n:n,
        G(count)=x*b1+y*b2;
        count=count+1;
    end
end

for x=1:NumberofPW,
    for y=x+1:NumberofPW,
        eps2(x,y)=(epsa-epsb)*2*f*besselj(1,abs(G(x)-G(y))*R)./(abs(G(x)-G(y))*R);
        eps2(y,x)=eps2(x,y);
    end
    eps2(x,x)=f*epsa+(1-f)*epsb;
end

k1=(0:0.1:1.0)/sqrt(3).*i*2*pi/a;
k2=((0.1:0.1:1.0)./3+1/sqrt(3)*i).*2.*pi./a;
k3=(0.9:-0.1:0).*(1.0/3.0+1/sqrt(3)*i).*2.*pi./a;%-(1/3+1/sqrt(3)*i)*2*pi/a;
k0=[k1 k2 k3];
%k0=k1;
%k0=(1/3+1/sqrt(3)*i).*2*pi/a;
counter=1;
eps2=inv(eps2);
for ii=1:length(k0),
    k=k0(ii);
    M=abs(k+G.').*abs(k+G).*(eps2); %%%for TM wave
    %%M=(real(k+G.').*real(k+G)+imag(k+G.').*imag(k+G)).*(eps2); %%%for TE wave
    E=sort(abs(eig(M)));
    freq(:,counter)=sqrt(abs(E(1:15))).*a./2./pi;
    display(sprintf('calculation of k=%f+%fi is finished',real(k),imag(k)));
    counter=counter+1;
end
tmpx=1:length(k0);
%plot(n,freq,'o'),hold on
plot(tmpx,freq,':','linewidth',2)
title('TM:Band structure of a 2D triangular photonic band structure')
xlabel('wave vector')
ylabel('wa/2\pic')
grid on
axis([0,length(k0)+1,0,1.4])
    
```

The calculation is also very fast, only about 100 plane waves are enough to produce exact results, There is a band gap for TM from 0.4301-0.5213 between the 2nd and 3rd band, and a band gap for TE from 0.3600-0.5212 between the 1st and 2nd band. There exists a complete band gap from 0.4301-0.5212.

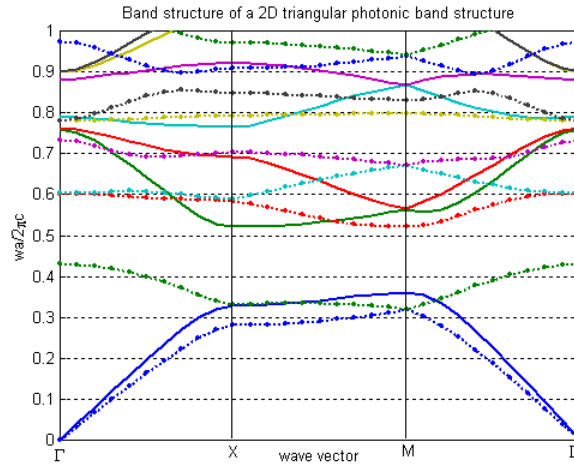


Figure 5: TE/TM band structure of a 2D triangular lattice with air holes in GaAs

Something to discuss:

1. How to make sure we get the correct lattice. For non-square lattice, like the triangular lattice, inverse FFT cannot give exact shape, and it needs to reconfigure. For square lattice, inverse FFT can give exact shape.
2. We get the $\mathbf{e}(G - G')$ directly by using analytical integration. Note this is only applicable to some simple and symmetric shape. If we cannot get the analytical expression for the integration, FFT is still needed.
3. FFT of a unit cell. The selection of a unit cell can be different, however, it is important to keep some symmetry, else the FFT matrix will be complex, then the yielded eigen value cannot be guaranteed to be real. (in that case, we need to use the complex matrix, else the result will be incorrect)

Theory in Fourier transform of periodic structure

For simple shaped atoms (unit cell), such as the cylinder, sphere, square, Fourier transform has an analytical form. For ideal infinite periodic structure, there is a relation between a single atom and the lattice. We use this theory, and greatly reduce the calculation relating to FFT.

Considering a unit cell with an area of A , an circular atom which is inversion symmetric is located in the center which ensures the Fourier transform is real. If the unit cell is represented by function $f(r)$, the Fourier transform of it is defined:

$$F(k) = \frac{1}{A} \iint_A f(r) e^{-i\vec{k} \cdot \vec{r}} ds \quad (40)$$

$$f(r) = \begin{cases} \mathbf{e}_a \cdots r < R \\ \mathbf{e}_b \cdots r > R \end{cases} = \mathbf{e}_b + \begin{cases} \mathbf{e}_a - \mathbf{e}_b, r < R \\ 0, r > R \end{cases} \quad (41)$$

So the integration can be solved like below:

$$\begin{aligned}
 F(k) &= \mathbf{e}_b \mathbf{d}(k) + \frac{1}{A} \int_0^R r dr \int_0^{2p} (\mathbf{e}_a - \mathbf{e}_b) e^{-ikr \cos f} d\mathbf{f} \\
 &= \mathbf{e}_b \mathbf{d}(k) + (\mathbf{e}_a - \mathbf{e}_b) \frac{1}{A} \int_0^R r dr \int_0^{2p} e^{-ikr \cos f} d\mathbf{f} \\
 &= \mathbf{e}_b \mathbf{d}(k) + (\mathbf{e}_a - \mathbf{e}_b) \frac{1}{A} \int_0^R 2pr J_0(kr) dr \\
 &= \mathbf{e}_b \mathbf{d}(k) + (\mathbf{e}_a - \mathbf{e}_b) \frac{2pR^2}{A} \frac{J_1(kR)}{kR}
 \end{aligned} \tag{42}$$

This assumes periodicity, treating it like one period in 1D problem. For infinite periodic lattice, only those reciprocal lattice point has frequency component, just like 1D periodic signal has discrete frequency components. So the Fourier transform for the whole lattice would be:

$$F(G) = \mathbf{e}_b \mathbf{d}(G) + (\mathbf{e}_a - \mathbf{e}_b) \frac{2pR^2}{A} \frac{J_1(GR)}{GR} = \mathbf{e}_b \mathbf{d}(G) + 2(\mathbf{e}_a - \mathbf{e}_b) f \frac{J_1(GR)}{GR} \tag{43}$$

f is a fraction parameter,
$$f = \frac{Area_{atom}}{Area_{cell}} \tag{44}$$

This principle is proved very useful in treating periodic structure. Same procedure can be used for square atom, sphere atom etc. Note there is a point which is very important here, the unit cell shape is not included in the above derivation, that means it has nothing to do with the Fourier transform, which guarantee the same FT results used in different lattice.

Another important property of Fourier transform is very important to supercell structure, that is the shifting property:

$$f(r + r_0) \leftrightarrow e^{i\mathbf{k} \cdot \mathbf{r}_0} F(k) \tag{45}$$

Other points which need to pay attention to when using Fourier transform are:

The choosing of a unit cell is very important. Inversion symmetry will get real Fourier transform, else complex results will get. If only real part is used, errors will occur.

Complex Fourier transform in this plane wave method does occur when a super cell cannot satisfy inversion symmetry. In this case, whether there exist real eigen values is still a question. (maybe it is worthwhile to look into it.)

From the Fourier transform matrix to get the dielectric function may cause some confusions:

Generally, we need use complex FT matrix to IFFT to get the original function, else there will be mistakes. If the FFT matrix is real, then only use the real part.

IFFT matrix will be generally complex, if so, absolute value can be used to get the original function. If it is pure real, only real part is used.

Generally, IFFTSHIFT should be used to get original function, this also show half a atom or a quarter atom at symmetric positions can be combined.

Numerical FFT: FFT of a unit cell will get all the required Fourier transform coefficients. For non-square lattice, this is not the case. The way how to deal with

non-square unit cell by FFT is not solved yet and analytical expression is the only way to get the numerical results.

Mode field distribution calculation:

Example 1: Square lattice with Alumina rods in air

Rod radius $R=0.2a$, dielectric for Alumina rod is 8.9, for air is 1.0.

After we solved the problem, we get the eigenvalue and associated eigen vector, the eigen value is real and eigen vector is not necessary to be real. Here is some features in calculation:

- 1) For the square lattice, the field distribution has still some problems to solve. The real and imaginary parts are both solution of field pattern, but there is a random phase along with it, the way to fix the phase is shown in the next section.
- 2) All the eigen vectors are normalized, i.e., $(h_i, h_i) = 1$ and different eigen vectors for a same k are orthogonal to each other, $(h_i, h_j) = \mathbf{d}_{ij}$ the information on amplitude for each mode is not present.
- 3) To get the magnetic field we use (13), and the results are complex-valued. Which part should be used then is determined by the requirement that: $(h_1(r), h_2(r)) = 0$. This requirement forces us to take real part for one mode, and take the imaginary part for the neighboring mode. Take one example, the integration in a unit cell is:

$$\langle \text{Re } H_1(r), \text{Re } H_2(r) \rangle = 2.069; \langle \text{Re } H_1(r), \text{Im } H_2(r) \rangle = -1.3e - 18$$

$$\langle \text{Im } H_1(r), \text{Re } H_2(r) \rangle = -8e - 13; \langle \text{Im } H_1(r), \text{Im } H_2(r) \rangle = -2.769$$
- 4) There still remains one problem: for a given mode, which one should be taken, the real or the imaginary part? This should be determined by the fact that the integration $\langle H, H \rangle = 1$ in a unit cell.
- 5) The real and imaginary part of the field are actually the same field at different time, i.e., $\text{Re}(r) = A(r)\cos \mathbf{d}$, $\text{Im}(r) = A(r)\sin \mathbf{d}$. Trying to solve the $A(r)$ from the real and imaginary parts will not be successful, for the phase=0 will give a problem. There might be some way to fix this problem.

This point is very important so that we can get the right answer for H field. (01/07/2001)
 What about the D and E field since there are no similar normalized & orthogonal

relations? We can use this fact that: $E_D = \int \frac{1}{\mathbf{e}(r)} |D(r)|^2 dr = 1$ to determine which part we

need to use. To make it easy, in a unit cell, the integration $\int |D(r)|^2 dr$ will be a little bit greater than 1 and we can use it as a standard to determine which one should be used.

But this method may be hard to apply to other non-square lattice for the integration is hard to evaluate.

Finally, the D field distribution using (7) would be complex. The calculation method can be derived as below:

For TM wave, the H field is in-plane, the D field is along z direction, so it is convenient to plot D

$$H_k^w(r) = \sum_G h_1(G) \hat{e}_1 e^{i(k+G) \cdot r} \quad (46)$$

$$\begin{aligned} D_k^w(r) &= \mathbf{e}(r) E(r) = \mathbf{e}(r) \frac{-ic}{w \mathbf{e}(r)} \nabla \times H_k^w(r) \\ &= \frac{-ic}{w} \sum_G h_1(G) i(k+G) \times \hat{e}_1 e^{i(k+G) \cdot r} \\ &= \frac{1}{|k|} \sum_G |k+G| h_1(G) e^{i(k+G) \cdot r} \hat{z} \end{aligned} \quad (47)$$

Note we only need the real part of the D field.

For TE wave, the D, E field are in-plane, H field is in z-direction, so H field is convenient for plotting:

$$H_k^w(r) = \sum_G h_1(G) \hat{z} e^{i(k+G) \cdot r} \quad (48)$$

$$\begin{aligned} D_k^w(r) &= \mathbf{e}(r) E(r) = \mathbf{e}(r) \frac{-ic}{w \mathbf{e}(r)} \nabla \times H_k^w(r) \\ &= \frac{-ic}{w} \sum_G i(k+G) \times \hat{z} e^{i(k+G) \cdot r} \\ &= \frac{1}{|k|} \sum_G |k+G| e^{i(k+G) \cdot r} \hat{e}_2 \end{aligned} \quad (49)$$

Note, these fields are all complex, we only need the real part, based on the definition of these complex waves, the imaginary part is the field after half a period and cannot be zero.

Off-plane propagation

For off-plane propagation, \vec{k} has 3 components in x, y and z. TE and TM are not decoupled any more, we need to use the most general form of (39). The most typical example of this is the photonic crystal fiber. The source file is:

D:\ksp\pbg\pcf\findmode.m

```
% this program is designed for solving the two dimensional photonic structure
% we are using the parameters from one of the paper
clear
warning off
epsa=1;
%epsb=13;
epsb=1.46^2; %%%Silica glass air system
a=1.0;
f=0.7;
R=sqrt(sqrt(3).*f/(2*pi)*a^2);
%R=0.48;
i=sqrt(-1);
%f=2*pi/sqrt(3)*R^2/a^2;
%b1=2*pi/a*(1-sqrt(3)/3*i);
%b2=2*pi/a*2*sqrt(3)/3*i;
b1=2*pi/a*[1 -1/sqrt(3) 0];
b2=2*pi/a*[0 2/sqrt(3) 0];
%n=input('please input n: ');
n=5;
NumberofPW=(2*n+1)^2;
```

```

count=1;
for x=-n:n,
    for y=-n:n,
        G(count,:)=x*b1+y*b2;
        count=count+1;
    end
end

for x=1:NumberofPW,
    for y=x+1:NumberofPW,
        eps2(x,y)=( epsa-epsb)*2*f*besselj(1,norm(G(x,:)-G(y,:))*R)/(norm(G(x,:)-G(y,:))*R);
        eps2(y,x)=eps2(x,y);
    end
    eps2(x,x)=f*epsa+(1-f)*epsb;
end

%k1=(0:0.1:1.0)/sqrt(3).*i*2*pi/a;
%k2=((0.1:0.1:1.0)./3+1/sqrt(3)*i).*2.*pi./a;
%k3=(0.9:-0.1:0).*(1.0/3.0+1/sqrt(3)*i).*2.*pi./a;%-(1/3+1/sqrt(3)*i)*2*pi/a;
%k0=[k1 k2 k3];
%k0=k1;
%k0=(1/3+1/sqrt(3)*i).*2*pi/a;
k0=zeros(length(0:0.1:1)+length(0.1:0.1:1)+length(0.9:-0.1:0),3);
mm0=length(0:0.1:1);
k0(1:mm0,2)=2*pi/a/sqrt(3).*(0:0.1:1)';
mm0=length(0:0.1:1);
mm1=length(0.1:0.1:1);
k0(mm0+1:mm0+mm1,1)=2*pi/a/3.*(0.1:0.1:1)';
k0(mm0+1:mm0+mm1,2)=2*pi/a/sqrt(3);
mm0=mm0+mm1+1;
mm1=length(0.9:-0.1:0);
k0(mm0:mm0+mm1-1,1)=2*pi/a/3.*(0.9:-0.1:0)';
k0(mm0:mm0+mm1-1,2)=2*pi/a/sqrt(3).*(0.9:-0.1:0)';
%k0(:,3)=9.0;

%band=zeros(3,100);
bandcount=1;
eps2=inv(eps2);
for xx=7:0.5:15,
    k0(:,3)=xx;
    counter=1;
    for ii=1:size(k0,1),
        k=k0(ii,:);
        K(:,1)=k(1)+G(:,1);
        K(:,2)=k(2)+G(:,2);
        K(:,3)=0;
        %%find the unit cell perpendicular to it in xy plane
        %%be sure to deal with the case of modulus(k)=0
        %%NaN in this case
        %%to save computing time, use absk=modulus(K)
        absk=modulus(K);
        e1=[K(:,2)./absk,-K(:,1)./absk,zeros(length(K),1)];
        e1(isnan(e1))=1/sqrt(2); %%when Kz is not zero,kx,ky is zero choose arbitrary e1
        K(:,3)=k(3)+G(:,3);
        %%find the other perpendicular unit cell
        %%be sure to deal with the case of modulus(k)=0
        %%NaN in this case
        absk=modulus(K);
        e2=cross(e1, K);
        e2=[e2(:,1)./absk,e2(:,2)./absk,e2(:,3)./absk];
        e2(isnan(e2))=0;
        %%form the equation matrix, it should be 2N by 2N
        %%in this case we will have no TE and TM decoupling

M1=([absk.*e2(:,1),absk.*e2(:,2),absk.*e2(:,3)],[absk.*e2(:,1),absk.*e2(:,2),absk.*e2(:,3)
]')*.eps2;

M2=([absk.*e1(:,1),absk.*e1(:,2),absk.*e1(:,3)],[absk.*e2(:,1),absk.*e2(:,2),absk.*e2(:,3)
]')*.eps2;
    
```

```

M3=[absk.*e2(:,1),absk.*e2(:,2),absk.*e2(:,3)]*[absk.*e1(:,1),absk.*e1(:,2),absk.*e1(:,3)
]'].*eps2;

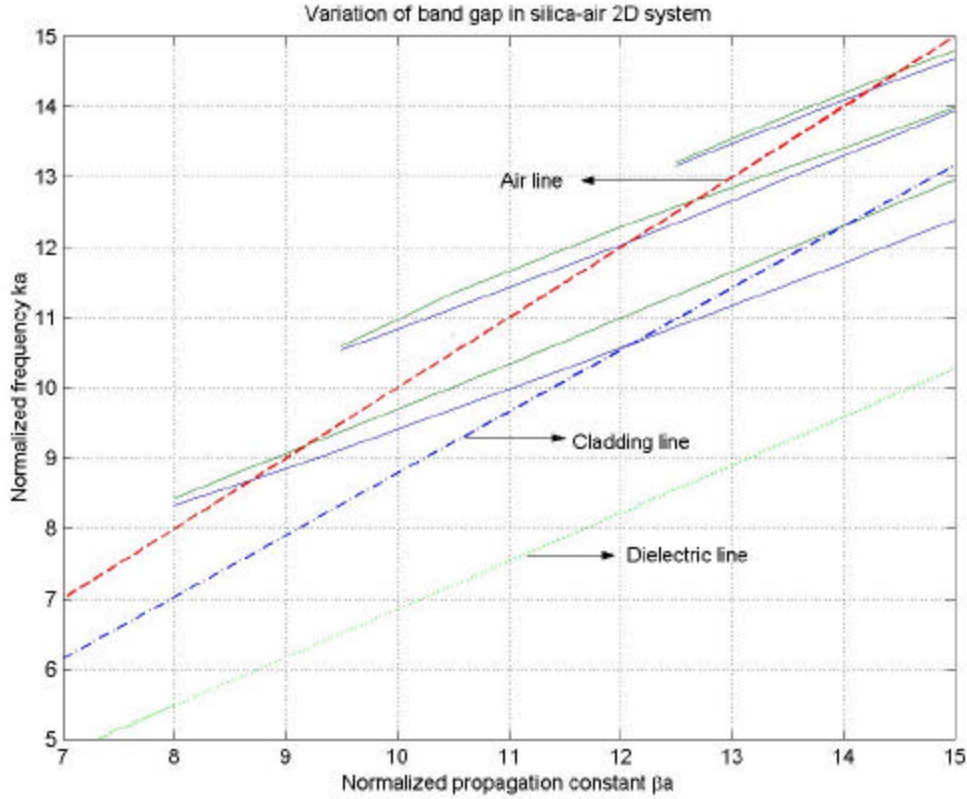
M4=[absk.*e1(:,1),absk.*e1(:,2),absk.*e1(:,3)]*[absk.*e1(:,1),absk.*e1(:,2),absk.*e1(:,3)
]'].*eps2;
M=[M1,M3;M2,M4];
E=sort(abs(eig(M)));
freq(:,counter)=sqrt(abs(E(1:20))).*a./2./pi;
% display(sprintf('calculation of k=[%f,%f,%f] is finished',k(1),k(2),k(3)));
counter=counter+1;
end
%%looking for the band gap and save them into band
display(sprintf('Searching bandgap for beta=%f',xx));
for ii=1:19,
    if min(freq(ii+1,:)) > max(freq(ii,:))+0.005 %%find a band gap, save these values
        band(1,bandcount)=xx;
        band(2,bandcount)=max(freq(ii,:));
        band(3,bandcount)=min(freq(ii+1,:));
        band(4,bandcount)=ii;
        display(sprintf('find a band (%f %f
%f):',band(1,bandcount),band(2,bandcount),band(3,bandcount)));
        bandcount=bandcount+1;
    end
end
end
%%now plotting the variation of the band gap, divide the array data according to the
band number
abl=band;
count1=1;
while length(band)>0,
    ab(:,1,count1)=band(:,1);
    band(:,1)=[];
    count2=1;
    ii=1;
    while ii<=size(band,2),
        %display(sprintf('ii=%f band(4,ii)=%f',ii,band(4,ii)));
        if band(4,ii)==ab(4,1,count1),
            count2=count2+1;
            ab(:,count2,count1)=band(:,ii);
            band(:,ii)=[];
        else
            ii=ii+1;
        end
    end
    count1=count1+1;
end
end

plot(ab(1,:,1),ab(2,:,1)*2*pi,'+-',ab(1,:,1),ab(3,:,1)*2*pi,'o-')
hold on
plot(ab(1,1:6,2),ab(2,1:6,2)*2*pi,'+-',ab(1,1:6,2),ab(3,1:6,2)*2*pi,'o-')
plot(ab(1,1:6,3),ab(2,1:6,3)*2*pi,'+-',ab(1,1:6,3),ab(3,1:6,3)*2*pi,'o-')
plot(ab(1,1:5,4),ab(2,1:5,4)*2*pi,'+-',ab(1,1:5,4),ab(3,1:5,4)*2*pi,'o-')
plot(ab(1,1:4,5),ab(2,1:4,5)*2*pi,'+-',ab(1,1:4,5),ab(3,1:4,5)*2*pi,'o-')

%plot(ab(1,1:3,3),ab(2,1:3,3)*2*pi,'+-',ab(1,1:3,3),ab(3,1:3,3)*2*pi,'o-')
xx=7:15;
plot(xx,xx,'r--','linewidth',1);
nav=sqrt(epsb)+f*(sqrt(epsa)-sqrt(epsb));
plot(k0(3,:),k0(3,+)/nav,'r--','linewidth',1);
plot(xx,xx/sqrt(epsb),'r:','linewidth',1);
hold off

title('band gap variation of a 2D triangular SiO2-Air PBG')
xlabel('wave vector k_za')
ylabel('ka or wa/c')
grid on
%axis([0,length(k0)+1,0,1.4])
  
```

This file solve the triangular silica-air photonic crystal fiber with air filling factor 70%.
 The band gap variation is shown below graph:



The glass-tube photonic crystal fiber

Using laboratory facilities, glass-tube structure is shown to be able to manufacture. Here, we offer the numerical analysis.

Three dimensional problems

3D diamond lattice:

The diamond lattice is a complex FCC lattice with two spherical atoms in the unit cell. Assuming the length of the simple cubic side is a , the primitive lattice vector basis are: $\vec{a}_1 = [0,1,1]a/2$, $\vec{a}_2 = [1,0,1]a/2$, $\vec{a}_3 = [1,1,0]a/2$. The locations of the two atoms in the primitive cell are: $\vec{r}_0 = [-1,-1,-1]a/4$ and $\vec{r}_1 = [1,1,1]a/4$. The basis vector in the reciprocal lattice is calculated according to:

$$\vec{b}_1 = 2\pi \frac{\vec{a}_2 \times \vec{a}_3}{\vec{a}_1 \cdot \vec{a}_2 \times \vec{a}_3}, \vec{b}_2 = 2\pi \frac{\vec{a}_3 \times \vec{a}_1}{\vec{a}_1 \cdot \vec{a}_2 \times \vec{a}_3}, \vec{b}_3 = 2\pi \frac{\vec{a}_1 \times \vec{a}_2}{\vec{a}_1 \cdot \vec{a}_2 \times \vec{a}_3}$$

Assuming the radius of the sphere is R , the Fourier coefficient at the reciprocal lattice grid is expressed as:

$$e(\vec{G}) = 3f(\mathbf{e}_a - \mathbf{e}_b) \left(\frac{\sin GR - GR \cos GR}{(GR)^3} \right) \cos(\vec{G} \cdot \vec{r}_0)$$

using the shift property and Fourier transform for a sphere, where $f = 2 \frac{4\pi R^3/3}{V}$ and

$$V = \vec{a}_1 \cdot \vec{a}_2 \times \vec{a}_3.$$

We show the band gap for $R = \sqrt{3}/8a$ using our simple program with $7^3 = 343$ plane waves. The graph shows excellent agreement with the result in PRL 65-25, 3152 even that we used a very small amount of plane waves.

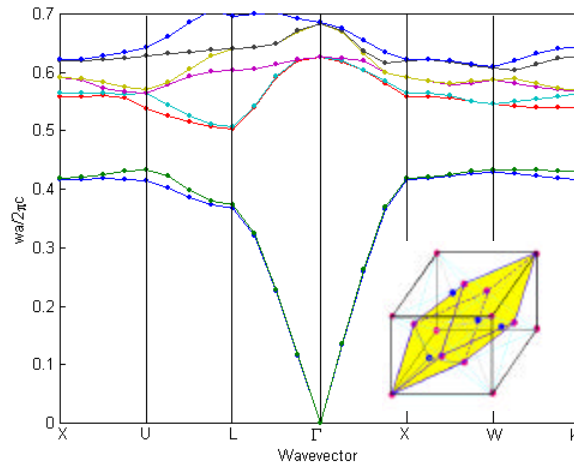


Figure 6: Band structure for a 3D diamond lattice

Note for the 2D honeycomb lattice, each primitive unit cell contains two atoms too. The same way applies when obtaining the Fourier coefficients.

See diamond.m.

```
% 3D Diamond Lattice
% Shangping Guo
% Reference data: K M Ho et al, PRL,65-25
% Also see S G Johnson's webpage on his software
% The k0 along the 1BZ is obtained from his website.
% This program yields accurate results,
```

```
clear
warning off
%epsa=11.56;
epsa=3.6^2;
epsb=1; %
a=1.0;
tic;
R=sqrt(3)/8*a;%a is the length of the cubic unit cell
i=sqrt(-1);
%define the lattice vectors
a1=[0 1 1]*a/2;
a2=[1 0 1]*a/2;
a3=[1 1 0]*a/2;
%calculate the reciprocal lattice vectors
volcell=dot(a1,cross(a2,a3));
b1=2*pi*cross(a2,a3)/volcell;
b2=2*pi*cross(a3,a1)/volcell;
b3=2*pi*cross(a1,a2)/volcell;
%n=input('please input n: ');
f=2*(4*pi*R^3/3)/volcell;
n=input('Input n: ');

%n=3;
NumberofPW=(2*n+1)^3;
display('Forming G vectors...');
```

```

count=1;
for x=-n:n,
    for y=-n:n,
        for z=-n:n,
            G(count,:)=x*b1+y*b2+z*b3;
            count=count+1;
        end
    end
end
display('Calculating eps(G-G)...');
r0=[0.125 0.125 0.125];%*[a1;a2;a3]/modulus(a1);
for x=1:NumberofPW,
    for y=x+1:NumberofPW,
        tmpG=norm(G(x,:)-G(y,:));
        eps2(x,y)=(epsa-epsb)*3*f*(sin(tmpG*R)-
tmpG*R*cos(tmpG*R))/(tmpG*R)^3*cos(dot(G(x,:)-G(y,:),r0));
        %This is problematic, for we add two spheres together,the f already contains it
        eps2(y,x)=eps2(x,y);
    end
    eps2(x,x)=f*epsa+(1-f)*epsb;
end
eps2=inv(eps2);
%forming the list of k-vectors on the edge of the 1BZ
display('Forming k-vector array...');
k1=interpolate([0 0.5 0.5]*[b1;b2;b3],[0 0.625 0.375]*[b1;b2;b3],4,0);
k2=interpolate([0 0.625 0.375]*[b1;b2;b3],[0 0.5 0]*[b1;b2;b3],4,0);
k3=interpolate([0 0.5 0]*[b1;b2;b3],[0 0 0]*[b1;b2;b3],4,0);
k4=interpolate([0 0 0]*[b1;b2;b3],[0 0.5 0.5]*[b1;b2;b3],4,0);
k5=interpolate([0 0.5 0.5]*[b1;b2;b3],[0.25 0.5 0.75]*[b1;b2;b3],4,0);
k6=interpolate([0.25 0.5 0.75]*[b1;b2;b3],[0.375 0.375 0.75]*[b1;b2;b3],4,1);
k0=[k1;k2;k3;k4;k5;k6];
display('Calculating eigen frequency...');
counter=1;
for ii=1:size(k0,1),
    tic;
    k=k0(ii,:);
    K(:,1)=k(1)+G(:,1);
    K(:,2)=k(2)+G(:,2);
    K(:,3)=0;
    %%find the unit cell perpendicular to it in xy plane
    %%be sure to deal with the case of modulus(k)=0
    %%NaN in this case
    e1=[K(:,2)./modulus(K),-K(:,1)./modulus(K),zeros(length(K),1)];
    e1(isnan(e1))=1/sqrt(2); %%when Kz is not zero,kx,ky is zero choose arbitrary e1
    K(:,3)=k(3)+G(:,3);
    %%find the other perpendicular unit cell
    %%be sure to deal with the case of modulus(k)=0
    %%NaN in this case
    e2=cross(e1, K);
    e2=[e2(:,1)./modulus(e2),e2(:,2)./modulus(e2),e2(:,3)./modulus(e2)];
    e2(isnan(e2))=0;
    %%form the equation matrix, it should be 2N by 2N
    %%in this case we will have no TE and TM decoupling
    mK=modulus(K);
    M1=( [mK.*e2(:,1),mK.*e2(:,2),mK.*e2(:,3)]* [mK.*e2(:,1),mK.*e2(:,2),mK.*e2(:,3)]' ).*eps2;
    M2=( [mK.*e1(:,1),mK.*e1(:,2),mK.*e1(:,3)]* [mK.*e2(:,1),mK.*e2(:,2),mK.*e2(:,3)]' ).*eps2;
    M3=( [mK.*e2(:,1),mK.*e2(:,2),mK.*e2(:,3)]* [mK.*e1(:,1),mK.*e1(:,2),mK.*e1(:,3)]' ).*eps2;
    M4=( [mK.*e1(:,1),mK.*e1(:,2),mK.*e1(:,3)]* [mK.*e1(:,1),mK.*e1(:,2),mK.*e1(:,3)]' ).*eps2;
    M=[M1,M3;M2,M4];

    E=sort(abs(eig(M)));
    freq(:,counter)=sqrt(abs(E(1:10))).*a./2./pi;
    % display(sprintf('calculation of k=[%f,%f,%f] is finished',k(1),k(2),k(3)));
    display(sprintf('%d/%d is completed:%f s used',counter,length(k0),toc));
    counter=counter+1;
    save diafreq3 freq;%long time may needed
end
tmpx=1:length(k0);

```

```
plot(tmpx,freq,'o-','linewidth',2)
title('Full band structure of a 3D diamond photonic crystal')
xlabel('wave vector k')
ylabel('wa/2\pic')
grid on
%axis([0,length(k0)+1,0,1.4])
toc
```

Supercell method to calculate defect

Supercell method is very important to model some more complicated system.

Point defect in a 2D square lattice

A rod is removed from the center of a 5X5 supercell lattice. See sqdef5.m

```
%%          2D PBG:Square lattice
%%          Shangping Guo
%%          sqband.m
clear all
warning off
tic;
epsa=8.9;
epsb=1;
a=1.0;
R=0.2*a;
i=sqrt(-1);
f=pi*R^2/a^2;
NumberofCell=5;
a1=a;
a2=a*i;
b1=2*pi/a/NumberofCell;
b2=2*pi/a/NumberofCell*i;
n=input('please input n:');
display('Fourier transforming.....');
NumberofPW=(2*n+1)^2;
mind=(-NumberofPW:NumberofPW)+NumberofPW+1;
mind=mind(:,ones(1,size(mind)))-NumberofPW-1;
GG=mind'*b1+mind*b2;
%clear mind;
eps21=2*f*(epsa-epsb)*besselj(1,abs(GG).*R)./(abs(GG).*R);
eps21(NumberofPW+1,NumberofPW+1)=epsb+f*(epsa-epsb);
%zz=[0,0]*[a1 a2].';%use 1x1 supercell to verify the algorithm
%5X5 super cell
zz=[
-2 -2;-2 -1;-2 0;-2 1;-2 2;
-1 -2;-1 -1;-1 0;-1 1;-1 2;
0 -2; 0 -1; 0 1; 0,2;
1 -2; 1 -1; 1 0; 1 1; 1 2;
2,-2; 2,-1; 2 0; 2,1; 2,2]*[a1 a2].';
eps20=zeros(length(eps21));
for x=1:length(zz),
    eps20=eps20+exp(i*(real(GG).*real(zz(x))+imag(GG).*imag(zz(x))))).*eps21;
end
ff=pi*R^2*length(zz)/(NumberofCell^2*a^2);
eps20=eps20./NumberofCell^2;
eps20(NumberofPW+1,NumberofPW+1)=epsb+ff*(epsa-epsb);

%clear GG;
count=1;
r=ones(2*n*NumberofCell+1,2);
for y=-n:n,
    for x=-n:n;
        G(count)=x*b1+y*b2;
        r(count,:)=[x,y];
        count=count+1;
    end
end
end

display('Building eps(G,G) matrix from the FFT matrix');
```



```

for x=1:NumberofPW,
    for y=x:NumberofPW,
        b=r(x,:)-r(y,)+(2*n+1)^2+1;
        eps2(x,y)=eps20(b(1),b(2));
        eps2(y,x)=eps2(x,y);
    end
end

k1=2*pi/a*0.5*(0:0.1:1);;
k2=2*pi/a*(0.5+(0.1:0.1:1).*0.5*i);
k3=2*pi/a*(0.5+0.5*i).*(0.9:-0.1:0);
k0=[k1 k2 k3]./NumberofCell;
display('Now calculate the eigen values..');
eps2=inv(eps2);
if max(max(real(eps2))) > 10^7*max(max(imag(eps2)))
display('Your lattice is inversion symmetric');
eps2=real(eps2);
else
display('Imaginary part of FFT is not zero');
stop;
%here we only demonstrate the inversion symmetric case
%however, nonsymmetric case is also supported
end

counter=1;
for ii=1:length(k0),
    k=k0(ii);
    %k=k0;
    %M=(real(k+G.)*real(k+G)+imag(k+G.)*imag(k+G)).*eps2; %TE wave
    M=abs(k+G.)*abs(k+G).*eps2; %TM wave
    E=sort(abs(eig(M)));
    freq(:,counter)=sqrt(abs(E)).*a./2./pi;
    display(sprintf('calculation of k=%f+%fi is finished',real(k),imag(k)));
    counter=counter+1;
end

tmpx=1:length(k0);
plot(tmpx,freq,'linewidth',2)
title('TM:Band structure of a 2D square PBG with a point defect (5X5)')
xlabel('wave vector')
ylabel('wa/2\pic')
grid on
axis([1 31 0.3 0.5])
tmp=ifft2(ifftshift(eps20));
%surf(fftshift(real(tmp)),shading interp,view(2)
%to get the lattice, use tmp=ifft2(ifftshift(eps20));

```

Supercell method is very similar to the method used in dealing with the ideal PBG.

The following program lists how to obtain the mode field, see sqptdef5fld.m

```

%%      Square lattice with a linear waveguide along x
%%      Shangping Guo
%%      Finally revised Jan 01 2001
%%      sqwavegd.m

clear all
warning off
tic;
epsa=11.56;%8.9;
epsb=1;
a=1.0;
R=0.2*a;
i=sqrt(-1);
f=pi*R^2/a^2;
NumberofCell=5;
a1=a;
a2=a*i;
b1=2*pi/a/NumberofCell;
b2=2*pi/a/NumberofCell*i;
n=9;

```

```

NumberofPW=(2*n+1)^2;
display('Forming regular Fourier matrix...');
mind=(-NumberofPW:NumberofPW)+NumberofPW+1;
mind=mind(:,ones(1,size(mind)))-NumberofPW-1;
GG=mind'*b1+mind*b2;

eps21=2*f*(epsa-epsb)*besselj(1,abs(GG).*R)/(abs(GG).*R);
eps21(NumberofPW+1,NumberofPW+1)=epsb+f*(epsa-epsb);
%5X5
zz=[
-2 -2;-2 -1;-2 0;-2 1;-2 2;
-1 -2;-1 -1;-1 0;-1 1;-1 2;
0 -2; 0 -1; 0 1; 0,2;
1 -2; 1 -1; 1 0; 1 1; 1 2;
2,-2; 2,-1; 2 0; 2,1; 2,2]*[a1 a2].';
eps20=zeros(length(eps21));
for x=1:length(zz),
    eps20=eps20+exp(i*(real(GG).*real(zz(x))+imag(GG).*imag(zz(x)))).*eps21;
end

display('Modifying the defect matrix...');
R00=0.80*a;
f00=pi*R00^2/a^2;
eps00=2*f00*(epsa-epsb)*besselj(1,abs(GG).*R00)/(abs(GG).*R00);
eps00(NumberofPW+1,NumberofPW+1)=epsb+f00*(epsa-epsb);
eps20=eps20+eps00;
ff=(pi*R^2*length(zz)+pi*R00^2)/(NumberofCell^2*a^2);
eps20=eps20./NumberofCell^2;
eps20(NumberofPW+1,NumberofPW+1)=epsb+ff*(epsa-epsb);

clear GG;
count=1;
r=ones(2*n*NumberofCell+1,2);
for y=-n:n,
    for x=-n:n;
        G(count)=x*b1+y*b2;
        r(count,:)=[x,y];
        count=count+1;
    end
end

display('Building eps(G,G) matrix from the FFT matrix');
for x=1:NumberofPW,
    for y=x:NumberofPW,
        b=r(x,:)-r(y,)+(2*n+1)^2+1;
        eps2(x,y)=eps20(b(1),b(2));
        eps2(y,x)=eps2(x,y);
    end
% eps2(x,x)=f*epsa+(1-f)*epsb;
end

k1=2*pi/a*0.5*(0:0.1:1);;
k2=2*pi/a*(0.5+(0.1:0.1:1).*0.5*i);
k3=2*pi/a*(0.5+0.5*i).*(0.9:-0.1:0);
k0=[k1 k2 k3]./NumberofCell;
k0=0;
display('Now calculate the eigen values..');
eps2=inv(eps2);
if max(max(real(eps2))) > 10^7*max(max(imag(eps2)))
display('Your lattice is inversion symmetric');
eps2=real(eps2);
else
display('Imaginary part of FFT is not zero');
end

counter=1;
for ii=1:length(k0),
    k=k0(ii);
    %k=k0;
end
    
```

```

%M=(real(k+G.)*real(k+G)+imag(k+G.)*imag(k+G)).*eps2; %TE wave
M=abs(k+G.)*abs(k+G).*eps2; %TM wave
[V,F]=eig(M);
E=diag(F);
%freq(:,counter)=sqrt(abs(E(1:10))).*a./2./pi;
display(sprintf('calculation of k=%f%i is finished',real(k),imag(k)));
counter=counter+1;
end
[lamda,index]=sort(E);
hg0=V(:,index(31));
hg1=V(:,index(32));
hg2=V(:,index(33));
hg3=V(:,index(34));
hg4=V(:,index(35));
hg5=V(:,index(36));
for x=(-2:0.02:2).*a,
for y=(-2:0.02:2).*a,
    fld0(x/0.02+101,y/0.02+101)=exp(i*(real(k+G)*x+imag(k+G)*y))*(hg0.*abs(k+G));
    fld1(x/0.02+101,y/0.02+101)=exp(i*(real(k+G)*x+imag(k+G)*y))*(hg1.*abs(k+G));
    fld2(x/0.02+101,y/0.02+101)=exp(i*(real(k+G)*x+imag(k+G)*y))*(hg2.*abs(k+G));
    fld3(x/0.02+101,y/0.02+101)=exp(i*(real(k+G)*x+imag(k+G)*y))*(hg3.*abs(k+G));
    fld4(x/0.02+101,y/0.02+101)=exp(i*(real(k+G)*x+imag(k+G)*y))*(hg4.*abs(k+G));
    fld5(x/0.02+101,y/0.02+101)=exp(i*(real(k+G)*x+imag(k+G)*y))*(hg5.*abs(k+G));
end
end
fld0=fixphase(fld0);
fld1=fixphase(fld1);
fld2=fixphase(fld2);
fld3=fixphase(fld3);
fld4=fixphase(fld4);
fld5=fixphase(fld5);
subplot(2,2,1),surf(real(fld0+fld1)),shading interp,view(2),axis image,axis
off,title('R=0.8a,degenerated level 31,32');
subplot(2,2,2),surf(real(fld2+fld3)),shading interp,view(2),axis image,axis
off,title('R=0.8a,degenerated level 33,34');
subplot(2,2,3),surf(real(fld4)),shading interp,view(2),axis image,axis
off,title('R=0.8a,level 35');
subplot(2,2,4),surf(real(fld5)),shading interp,view(2),axis image,axis
off,title('R=0.8a,level 36');

```

Here fixphase.m is a supporting function to fix the random phase together with the real and imaginary part:

```

function z=fixphase(f)
s0=sum(sum(real(f).^2-imag(f).^2));
s1=sum(sum(2.*real(f).*imag(f)));
theta=0.5*atan2(-s1,s0);
display(sprintf('Phase is:%f',theta));
tmpr=real(f)*cos(theta)-imag(f)*sin(theta);
tmpi=real(f)*sin(theta)-imag(f)*cos(theta);
z=tmpr+sqrt(-1)*tmpi;

```

Below is a graph shown the center rod is enlarged to 0.6a.

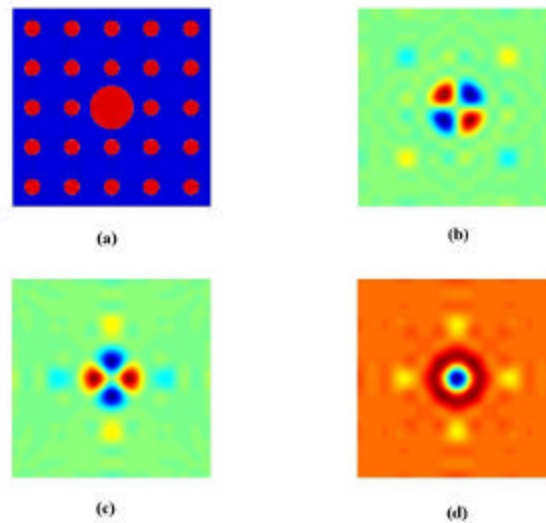


Figure 7: Mode field in a 2D square lattice with the center rod being removed

Linear defect: waveguide

The way to deal with linear waveguide (defect mode frequency and mode field) is the same as point defect. The band structure calculated of the defect mode will also be periodic (not a straight line any more)

We need to use the principle of the band structure, stretch the periodic band structure into a graph like below. Also, we need to project the band structure to the waveguide direction.

The source file for the projection is list as below:

```
% this program is designed for solving the two dimensional photonic structure
% we are using the parameters from one of the paper
% The structure is a square 2d periodic structure
% the higher index rod is in the background of air
clear
warning off
epsa=8.9;
epsb=1;
a=1.0;
R=0.2*a;
i=sqrt(-1);
f=pi*R^2/a^2;
a1=a;
a2=a*i;
b1=2*pi/a;
b2=2*pi/a*i;
%n=input('please input n: ');
n=5;
NumberofPW=(2*n+1)^2;

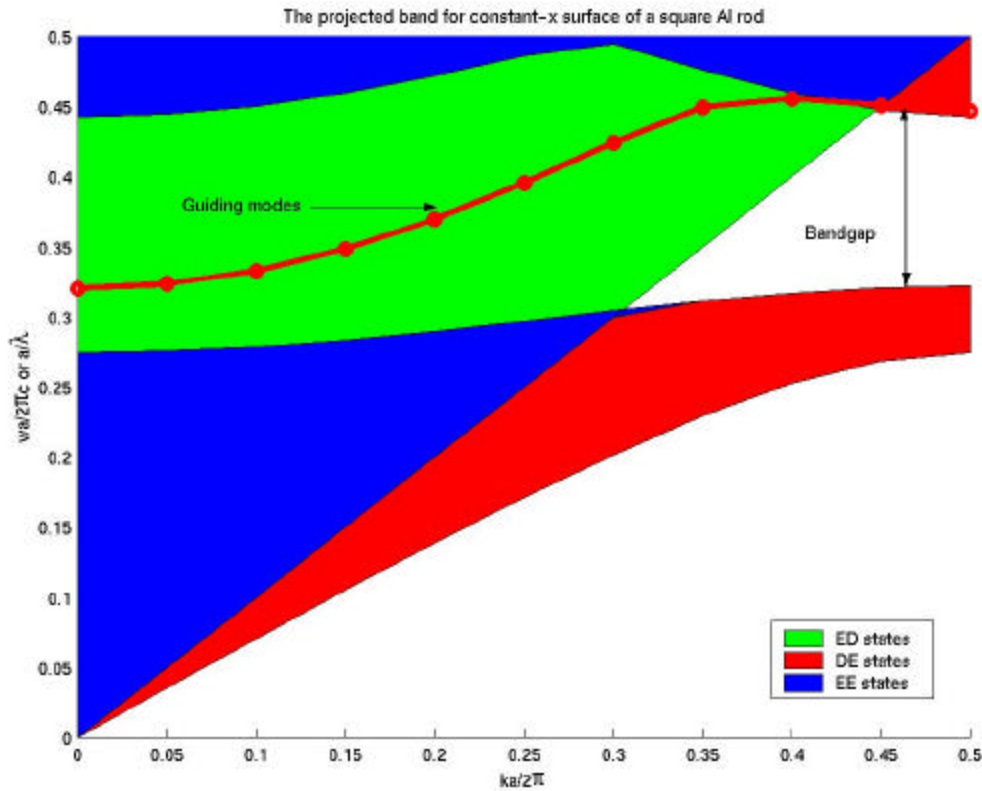
count=1;
for x=-n:n,
    for y=-n:n,
        G(count)=x*b1+y*b2;
        count=count+1;
    end
end

for x=1:NumberofPW,
    for y=x+1:NumberofPW,
        eps2(x,y)=(epsa-epsb)*2*f*besselj(1,abs(G(x)-G(y))*R)./(abs(G(x)-G(y))*R);
```

```

    eps2(y,x)=eps2(x,y);
  end
  eps2(x,x)=f*epsa+(1-f)*epsb;
end
eps2=inv(eps2);
count1=1;
for aa=(0:0.1:1)*pi/a,
  bb=(0:0.1:1)*pi/a;
  k0=aa+bb*i;
  counter=1;
  for ii=1:length(k0),
    k=k0(ii);
    M=abs(k+G.').*abs(k+G).*(eps2);
    E=sort(abs(eig(M)));
    freq(:,counter)=sqrt(abs(E(1:20))).*a./2./pi;
    % display(sprintf('calculation of k=%f+%fi is finished',real(k),imag(k)));
    counter=counter+1;
  end
  min1(count1)=min(freq(1,:));
  max1(count1)=max(freq(1,:));
  min2(count1)=min(freq(2,:));
  max2(count1)=max(freq(2,:));
  count1=count1+1;
end
tmpx=(0:0.1:1)/2;
%plot(tmpx,min1,tmpx,max1,tmpx,min2,'linewidth',2)
hold on
fill([tmpx,rot90(tmpx)],[tmpx,rot90(max2)],'g','erasemode','none')
fill([tmpx,rot90(tmpx)],[min2,rot90(max2)],'r','erasemode','none')
fill([tmpx,rot90(tmpx)],[min1,rot90(max1)],'r','erasemode','none')
fill([tmpx,rot90(tmpx)],[min(max1,tmpx),rot90(max1)],'b','erasemode','none')
fill([tmpx,rot90(tmpx)],[max(min2,tmpx),rot90(max2)],'b','erasemode','none')
axis([0 0.5 0 0.5])
xlabel('ka/2\pi')
ylabel('wa/2\pic or a/\lambda')
title('The projected band for constant-x surface of a square Al rod')
[h1 h2]=legend('ED states','DE states','EE states')
set(h2(2),'facecolor','g')
set(h2(3),'facecolor','r')
set(h2(4),'facecolor','b')

```



As is shown, Plane wave method is able to deal with a lot of PBG related problems. We have verified our program using a lot of published results. Minor modifications needed to tailor them for other applications. For various applications and the divergence see my previous paper.

Convergence and stability Sizetest.m

	#of plane waves	5X5	7X7	9X9	11X11	13X13	15X15	17X17	19X19
6	169	0.403069	0.410824	0.418774	0.429622				
7	225	0.397684	0.407251	0.414597	0.421022				
8	289	0.396169	0.405432	0.410860	0.417016				
9	361	0.394902	0.404108	0.407977	0.413684				
10	441	0.394326	0.399563	0.406205	0.410681	0.415767			
11	529	0.394169	0.398201	0.405243	0.408290	0.412992			
12	625		0.397168	0.403696	0.406652	0.410508			
13	729		0.396306	0.399946	0.405695	0.408477			
14	841		0.395868	0.398680	0.404899	0.406981			
15	961		0.395736	0.397886	0.403183	0.406020	0.408608		
16	1089			0.397093	0.400051	0.405367	0.407242		
17	1225			0.396476	0.398873	0.404526	0.406297	0.408707	0.411785
18	1369			0.396130	0.398225	0.402755	0.405660	0.407456	0.410178
19	1521				0.397558	0.400091	0.405090	0.406540	0.408786
20	1681				0.396945	0.398997	0.404164	0.405899	0.407633
21	1849				0.396476	0.398433	0.402417	0.405404	0.406754
22	2025				0.396191	0.397874	0.400111	0.404826	0.406112
23	2209					0.397318	0.399095	0.403831	0.405633
24	2401					0.396831	0.398579	0.402149	0.405187
25	2601					0.396456	0.398101	0.400123	0.404571
26	2809						0.397612	0.399178	0.403531
27	3025						0.397148	0.398691	0.401935

28	3249					0.396748	0.398272	0.400131
29	3481					0.396438	0.397842	0.399250
30	3721						0.397416	0.398781
31	3969						0.397023	0.398405
32	4225							0.398024

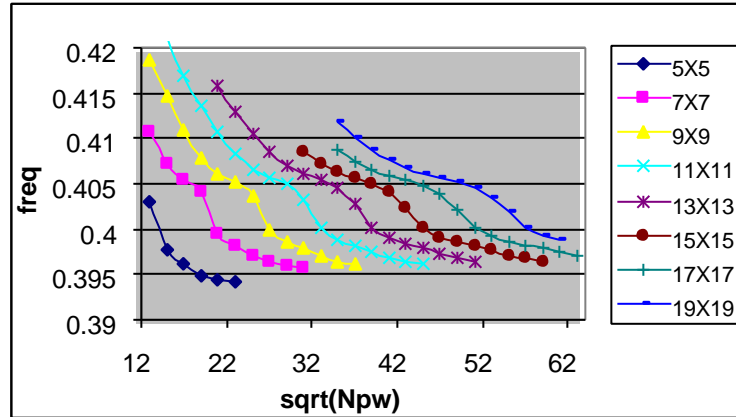


Figure 8: Convergence of defect frequency for TM mode using different supercell size in a square lattice with the center rod being removed

There is an apparent oscillation, this is due to the Gibbs' phenomena.

Plane wave method is proved successful for photonic band gap calculations. The algorithm is simple and fast for simple case, and it is generally stable and it provides accurate and complete band information of a PBG. There are some weaknesses in this method:

1. Huge computations for large supercell, large number of plane waves, high accuracy calculations. Desktop PC and workstation is not sufficient to cover such kind of computation work. Parallel computing is needed.
2. Assuming the dielectric is constant in the region of interest. No dispersion is considered, though some plasma-like dispersion can be incorporated into PWM method.
3. Material loss is not considered in this method.
4. Only deal with infinite large photonic crystal. To deal with finite PBG, it is not good.
5. Cannot deal with dynamics in PBG devices, which limits the usage of this method. FDTD is far better than PWM in this point.
6. Provides no reflection and transmission property of a PBG material. FDTD and TMM (Transfer Matrix Method) can give transmission spectrum.
7. Both FDTD and TMM are not safe in providing band structure information because of mode coupling mechanism. These methods are also suffering by stability and accuracy problem. In this point of view, these 3 methods should be combined together to get the complete information of a PBG.

For TMM simulation tool, see homepage of University of Glasgow. I am going to post my FDTD code soon.

FAQ: Frequently asked questions

*** Please send your comments to shangpingguo@hotmail.com. Thanks a lot.***

Manual prepared in 2000-2001