

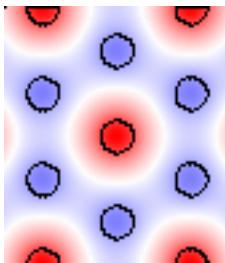


**Northumbria
University
NEWCASTLE**

KD6041

Photonic crystal simulations

Section 4: Task 3 – MPB: 2D Photonic crystal band calculations (Week 7)



Laboratory notes

Dr. Ying-Lung Daniel Ho and Dr. Mike Taverne

Nanophotonic Engineering Laboratory

<https://nanophotonicenglab.github.io/>

Department of Mathematics, Physics and Electrical Engineering
Northumbria University, Newcastle

Version: November 9, 2022

Contents

1 Scaling properties of the Maxwell equations	1
2 Direct lattice and reciprocal lattice	2
2.1 Other crystallography concepts	3
2.1.1 Constructing the Wigner-Seitz cell	3
2.1.2 Why the Brillouin zone matters	4
2.2 General implementation in MPB	5
3 Guided example: The square lattice, a 2D photonic crystal	6
3.1 Calculate the reciprocal lattice vectors	6
3.1.1 Solution	7
3.2 Compute the photonic band structure	9
3.2.1 Creating the plot	10
3.2.2 MPB code explanations	14
3.2.3 MPB output analysis	14
4 Homework	16
4.1 Calculate the reciprocal lattice vectors of a 2D photonic crystal (triangular/hexagonal lattice)	16
4.2 Compute the photonic band structure of a 2D photonic crystal (triangular/hexagonal lattice)	17
5 Appendix	19
5.1 Vector operations	19
Bibliography	20

1 Scaling properties of the Maxwell equations

An important concept when working with photonic crystals is **scalability**. The master equation used to calculate the bandstructure is:

$$\vec{\nabla} \times \left(\frac{1}{\varepsilon(\vec{r})} \vec{\nabla} \times \vec{H}(\vec{r}) \right) = \left(\frac{\omega}{c_0} \right)^2 \vec{H}(\vec{r}) \quad (1.1)$$

For a **permittivity scaling** $\varepsilon(\vec{r}) \rightarrow \varepsilon'(\vec{r}) = \varepsilon(\vec{r})/s^2$:

$$\vec{\nabla} \times \left(\frac{1}{\varepsilon(\vec{r})} \vec{\nabla} \times \vec{H}(\vec{r}) \right) = \left(\frac{\omega}{c_0} \right)^2 \vec{H}(\vec{r})$$

↓

$$\vec{\nabla} \times \left(\frac{1}{\varepsilon(\vec{r})/s^2} \vec{\nabla} \times \vec{H}(\vec{r}) \right) = \left(\frac{s\omega}{c_0} \right)^2 \vec{H}(\vec{r})$$

↓

$$\omega \rightarrow \omega' = s \cdot \omega \quad (\text{equivalently, wavelength } \lambda = 2\pi c_0/\omega \rightarrow \lambda' = \lambda/s)$$

For a **spatial scaling** $\varepsilon(\vec{r}) \rightarrow \varepsilon'(\vec{r}) = \varepsilon(\vec{r}/s)$ (geometry scaled by s), in which case the lattice period is scaled as follows $a \rightarrow a' = s \cdot a$:

$$\vec{\nabla}_r = \frac{\partial \vec{R}}{\partial \vec{r}} \vec{\nabla}_R, \vec{R} = s\vec{r} \Rightarrow \nabla_r = s\nabla_R$$

$$\vec{\nabla}_r \times \left(\frac{1}{\varepsilon(\vec{r})} \vec{\nabla}_r \times \vec{H}(\vec{r}) \right) = s^2 \vec{\nabla}_R \times \left(\frac{1}{\varepsilon(\vec{R}/s)} \vec{\nabla}_R \times \vec{H}(\vec{R}/s) \right) = \left(\frac{\omega}{c_0} \right)^2 \vec{H}(\vec{R}/s)$$

↓

$$\vec{\nabla}_R \times \left(\frac{1}{\varepsilon(\vec{R}/s)} \vec{\nabla}_R \times \vec{H}(\vec{R}/s) \right) = \left(\frac{\omega/s}{c_0} \right)^2 \vec{H}(\vec{R}/s)$$

↓

$$\vec{\nabla}_R \times \left(\frac{1}{\varepsilon'(\vec{R})} \vec{\nabla}_R \times \vec{H}'(\vec{R}) \right) = \left(\frac{\omega/s}{c_0} \right)^2 \vec{H}'(\vec{R})$$

↓

$$\omega \rightarrow \omega' = \omega/s \quad (\text{equivalently, wavelength } \lambda = 2\pi c_0/\omega \rightarrow \lambda' = s \cdot \lambda)$$

Note that in this case, the mode profile also changes: $\vec{H}(\vec{r}) \rightarrow \vec{H}'(\vec{r}) = \vec{H}(\vec{r}/s)$, i.e. it is rescaled the same way as the permittivity.



See chapter 2, pages 20-21 in reference [1] for a more detailed discussion.

2 Direct lattice and reciprocal lattice

Crystals are defined using unit-cells, which are then replicated using lattice vectors $\vec{a}_1, \vec{a}_2, \vec{a}_3$ as illustrated in figure 2.1.

This lattice in “real space” is called **direct lattice**.

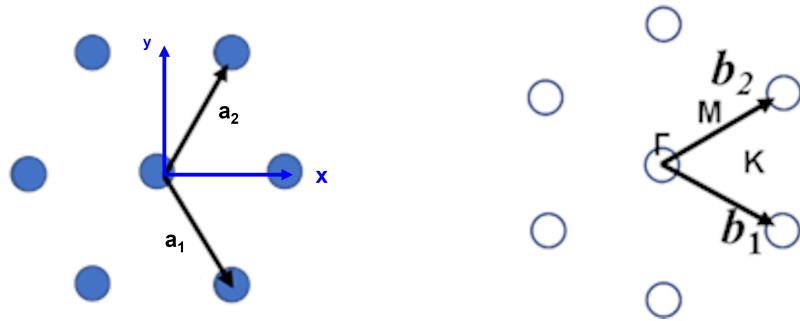


Figure 2.1: The real space and reciprocal space lattice of the triangular (or hexagonal) lattice.

Based on this direct lattice, we can define another lattice in “spatial frequency space” called the **reciprocal lattice** using the following equations to define its basis vectors $\vec{b}_1, \vec{b}_2, \vec{b}_3$:

$$\begin{cases} \vec{b}_1 = \frac{2\pi\vec{a}_2 \times \vec{a}_3}{\vec{a}_1 \cdot (\vec{a}_2 \times \vec{a}_3)} \\ \vec{b}_2 = \frac{2\pi\vec{a}_3 \times \vec{a}_1}{\vec{a}_1 \cdot (\vec{a}_2 \times \vec{a}_3)} \\ \vec{b}_3 = \frac{2\pi\vec{a}_1 \times \vec{a}_2}{\vec{a}_1 \cdot (\vec{a}_2 \times \vec{a}_3)} \end{cases} \quad (2.1)$$



See section §5.1 if you need a reminder on vector operations.

2.1 Other crystallography concepts

- **Unit-cell:** A unit cell is a repeating unit formed by the vectors spanning the points of a lattice.
- **Primitive unit-cell:** A unit-cell of minimum area or volume.
- **Wigner-Seitz cells:** The Wigner-Seitz cell is a primitive cell which has been constructed by applying Voronoi decomposition to a crystal lattice. See https://en.wikipedia.org/wiki/Wigner%20Seitz_cell for more information.
- **Brillouin zone:** The Wigner–Seitz cell in the reciprocal space.
- **Irreducible Brillouin zone:** The Brillouin zone reduced by all of the symmetries in the point group of the lattice (point group of the crystal).
- **Critical points (or high symmetry points):** Corners and centres of the irreducible Brillouin Zone with standard symbols/labels. See https://en.wikipedia.org/wiki/Brillouin_zone#Critical_points.

2.1.1 Constructing the Wigner–Seitz cell

Figure 2.2 illustrates the construction of a Wigner–Seitz primitive cell.

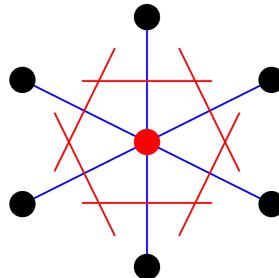


Figure 2.2: Construction of a Wigner–Seitz primitive cell.

1. The cell may be chosen by first picking a lattice point.
2. After a point is chosen, lines are drawn to all nearby lattice points.
3. At the midpoint of each line, another line is drawn normal to each of the first set of lines.
4. The smallest area enclosed in this way is called the Wigner–Seitz primitive cell.

2.1.2 Why the Brillouin zone matters

When using the Plane-Wave Expansion method, the solutions ω and \vec{H} for any wavevector $\vec{k}' = \vec{k} + \vec{G}$ where $\vec{G} = l \cdot \vec{b}_1 + m \cdot \vec{b}_2 + n \cdot \vec{b}_3$ and l, m, n are integers will be the same as for the wavevector \vec{k} . See figure 2.3.

For this reason, it is sufficient to only consider wavevectors \vec{k} inside the Brillouin zone.

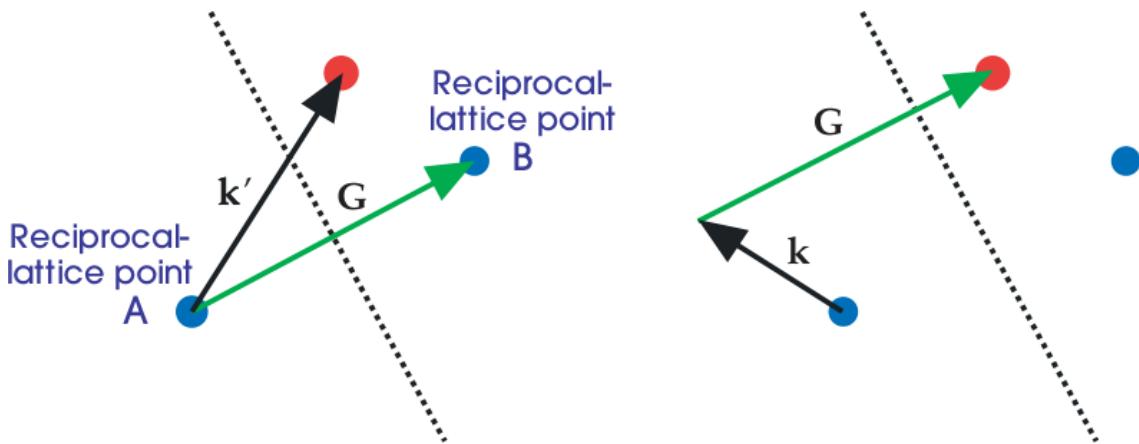


Figure 2.3: Characterization of the Brillouin zone. The dotted line is the perpendicular bisector of the line joining two reciprocal lattice points (blue). If we choose the left point as the origin, any lattice vector (such as \vec{k}') that reaches to an arbitrary point on the other side (red) can be expressed as the sum of a same-side vector (such as \vec{k}) plus a reciprocal lattice vector \vec{G} .



! See appendix B in reference [1] for more details.

2.2 General implementation in MPB

- How to define the *direct lattice* in MPB:

```
(set! geometry-lattice
  (make lattice
    (basis1 a1x a1y a1z) ; direction of a1
    (basis2 a2x a2y a2z) ; direction of a2
    (basis3 a3x a3y a3z) ; direction of a3
    (basis-size a1 a2 a3) ; length of the lattice vectors a1,a2,a3
    (size 1 1 1) ; number of unit-cells along each lattice vector
  )
)
```

! *basis1*, *basis2* and *basis3* are the three lattice directions of the crystal, specified in the cartesian basis. The lengths of these vectors are ignored--only their directions matter. The lengths are determined by the *basis-size* property. These vectors are then used as a basis for all other 3-vectors in the ctl file (except *k-points*, see below). They default to the x, y, and z directions, respectively.

The *size* property can be used to reduce the dimensionality of the computation, as we have seen before:

- 1D: (size 1 no-size no-size)
- 2D: (size 1 1 no-size)
- 3D: (size 1 1 1)

- How to define the wavevectors \vec{k} (*k-points*) to solve for in MPB:

```
(set! k-points (list
  (vector3 k1 k2 k3); k = k1*b1 + k2*b2 + k3*b3
  ...
  (vector3 k1 k2 k3); k = k1*b1 + k2*b2 + k3*b3
))
```

! The k-points are defined in the reciprocal lattice, i.e. the real wavevector in the example above is $\vec{k} = k_1 \vec{b}_1 + k_2 \vec{b}_2 + k_3 \vec{b}_3$.

See https://mpb.readthedocs.io/en/latest/Scheme_User_Interface/#lattice for the official documentation.

3 Guided example: The square lattice, a 2D photonic crystal

3.1 Calculate the reciprocal lattice vectors

Figure 3.1 shows a simple *square lattice*. The *direct lattice* vectors are defined as follows:

$$\begin{cases} \vec{a}_1 = a_{12}\vec{x} \\ \vec{a}_2 = a_{12}\vec{y} \\ \vec{a}_3 = a_3\vec{z} \end{cases} \quad (3.1)$$

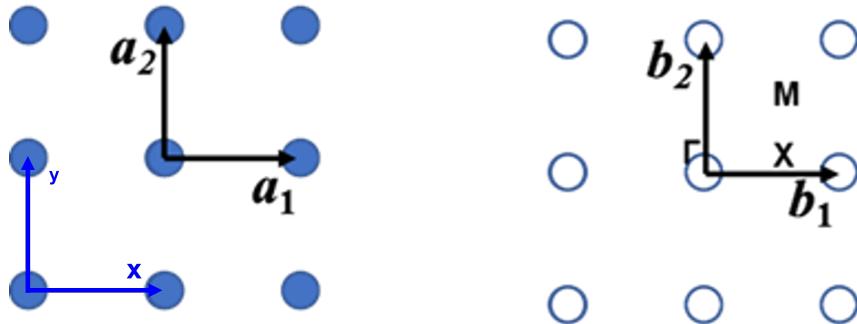


Figure 3.1: The real space and reciprocal space lattice of the square lattice.

1. Draw the following on top of figure 3.1:
 - a) On the real-space lattice:
 - i. The *Wigner-Seitz cell*.
 - ii. One *alternative unit cell*
 - b) On the reciprocal lattice:
 - i. The *Brillouin zone*
 - ii. An *irreducible Brillouin zone* with the critical points Γ , X , M :
 - A. Γ : Center of the Brillouin zone
 - B. X : Center of an edge
 - C. M : Corner

2. Calculate the coordinates in the provided cartesian basis ($\vec{x}, \vec{y}, \vec{z} = \vec{x} \times \vec{y}$) (in blue in figure 4.1) of the following:

- a) The real (direct) lattice basis vectors: $\vec{a}_1, \vec{a}_2, \vec{a}_3$
- b) The reciprocal lattice basis vectors: $\vec{b}_1, \vec{b}_2, \vec{b}_3$
- c) The coordinates of the critical points: Γ, X, M

3.1.1 Solution

Question 1

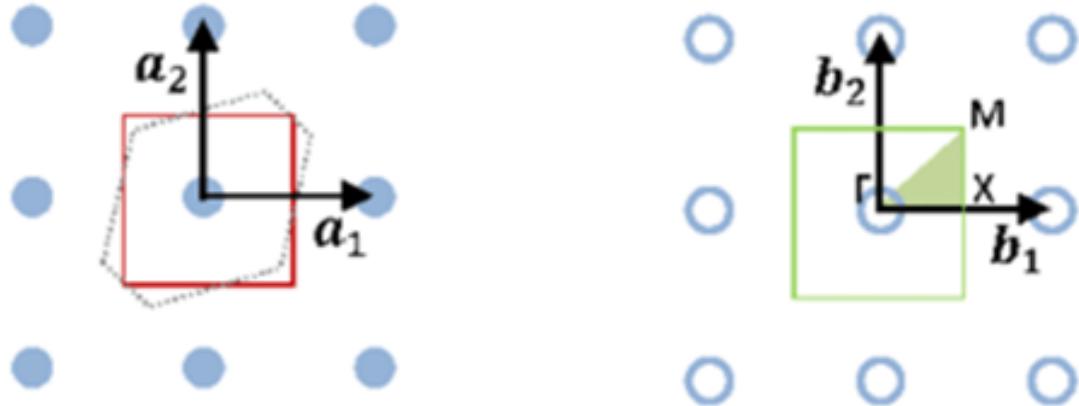


Figure 3.2: The real space and reciprocal space lattice of the square lattice. The Wigner-Seitz cell (red) and one alternative unit cell (grey dotted) are indicated. The full Brillouin zones are outlined (green) and the irreducible Brillouin zones shown (green shaded area).

Question 2

See equation (2.1) for the general formula for calculating the reciprocal lattice vectors. Since all three formulas for $\vec{b}_1, \vec{b}_2, \vec{b}_3$ have the same denominator $\vec{a}_1 \cdot (\vec{a}_2 \times \vec{a}_3)$, we will start by calculating that.

$$\begin{aligned}
 \vec{a}_1 \cdot (\vec{a}_2 \times \vec{a}_3) &= a_{12}\vec{x} \cdot (a_{12}\vec{y} \times a_3\vec{z}) \\
 &= a_{12}a_{12}a_3(\vec{x} \cdot (\vec{y} \times \vec{z})) \\
 &= a_{12}a_{12}a_3(\vec{x} \cdot \vec{x}) \\
 &= a_{12}a_{12}a_3
 \end{aligned}$$

The new expressions for the reciprocal lattice vectors are now:

- $\vec{b}_1 = \frac{2\pi\vec{a}_2 \times \vec{a}_3}{a_{12}a_{12}a_3}$
- $\vec{b}_2 = \frac{2\pi\vec{a}_3 \times \vec{a}_1}{a_{12}a_{12}a_3}$
- $\vec{b}_3 = \frac{2\pi\vec{a}_1 \times \vec{a}_2}{a_{12}a_{12}a_3}$

This leads to:

$$\begin{aligned}\vec{b}_1 &= 2\pi \frac{\vec{a}_2 \times \vec{a}_3}{a_{12}a_{12}a_3} \\ &= 2\pi \frac{a_{12}a_3}{a_{12}a_{12}a_3} \cdot (\vec{y} \times \vec{z}) \\ &= \frac{2\pi}{a_{12}} \vec{x} \\ \vec{b}_2 &= 2\pi \frac{\vec{a}_3 \times \vec{a}_1}{a_{12}a_{12}a_3} \\ &= 2\pi \frac{a_3a_{12}}{a_{12}a_{12}a_3} \cdot (\vec{z} \times \vec{x}) \\ &= \frac{2\pi}{a_{12}} \vec{y} \\ \vec{b}_3 &= 2\pi \frac{\vec{a}_1 \times \vec{a}_2}{a_{12}a_{12}a_3} \\ &= 2\pi \frac{a_{12}a_{12}}{a_{12}a_{12}a_3} \cdot (\vec{x} \times \vec{y}) \\ &= \frac{2\pi}{a_3} \vec{z}\end{aligned}$$

The critical points of the Brillouin zone are:

- Γ : Center of the Brillouin zone
- X : Center of an edge
- M : Corner

So $\Gamma = \vec{0}$, while for X and M , there are 4 possible points. Here we will choose the M and X points as defined in figure 3.2, i.e. X is the midpoint of \vec{b}_1 and M is the centre of the square formed by \vec{b}_1 and \vec{b}_2 . This allows us to write Γ , X and M in terms of the reciprocal lattice vectors as follows:

$$\begin{cases} \Gamma = \vec{0} \\ X = \frac{1}{2}\vec{b}_1 \\ M = \frac{1}{2}(\vec{b}_1 + \vec{b}_2) \\ X = \frac{1}{2}\vec{b}_1 \\ = \frac{2\pi}{a_{12}} \frac{\vec{x}}{\frac{a_{12}}{2}} \\ = \frac{2\pi}{a_{12}} \begin{pmatrix} 1/2 \\ 0 \\ 0 \end{pmatrix} \\ M = \frac{1}{2}(\vec{b}_1 + \vec{b}_2) \\ = \frac{1}{2} \left(\frac{2\pi}{a_{12}} \vec{x} + \frac{2\pi}{a_{12}} \vec{y} \right) \\ = \frac{2\pi}{a_{12}} \begin{pmatrix} 1/2 \\ 1/2 \\ 0 \end{pmatrix} \end{cases}$$

Summary, in column vector form for all:

- $\vec{a}_1 = a_{12} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \vec{a}_2 = a_{12} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \vec{a}_3 = a_3 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$
- $\vec{b}_1 = \frac{2\pi}{a_{12}} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \vec{b}_2 = \frac{2\pi}{a_{12}} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \vec{b}_3 = \frac{2\pi}{a_3} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$
- $\Gamma = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, X = \frac{2\pi}{a_{12}} \begin{pmatrix} 1/2 \\ 0 \\ 0 \end{pmatrix}, M = \frac{2\pi}{a_{12}} \begin{pmatrix} 1/2 \\ 1/2 \\ 0 \end{pmatrix}$

3.2 Compute the photonic band structure

We are now going to compute the photonic band structure of a two-dimensional square lattice of dielectric rods in air, based on the geometry defined in figure 3.1 and figure 3.3. The parameters will be as follows:

- Relative permittivity of the rods: $\varepsilon_r = 8.9 \implies n_{rods} = \sqrt{\varepsilon_r} \simeq 2.98$
- The background (or backfill) material is air: $\varepsilon_r = 1 \iff n_{backfill} = 1$
- Radius of the rods: $r = 0.2 \cdot a$ (where $a = a_{12}$)

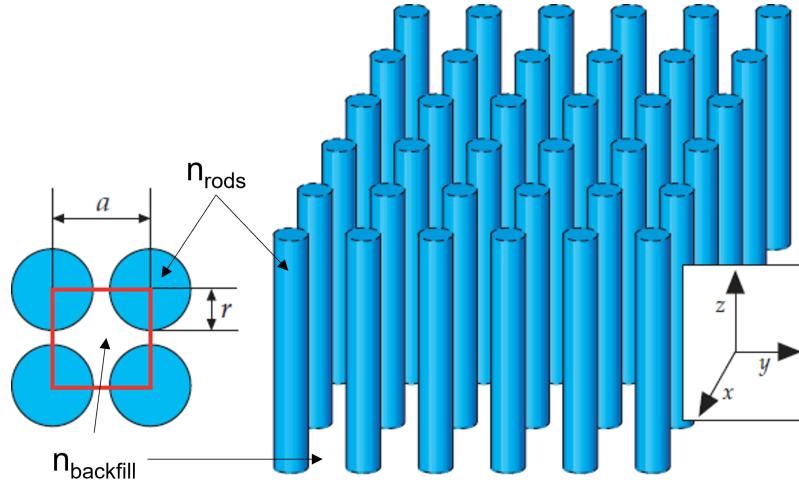


Figure 3.3: The square lattice geometry.

3.2.1 Creating the plot

The necessary scripts can be found in `KD6041-resources\MPB\平方_lattice` (make sure to update your resource directory like last time) and are shown in listings 1, 2 and 3. The resulting plot is shown in figure 3.4 and the generated geometry images in figure 3.5.

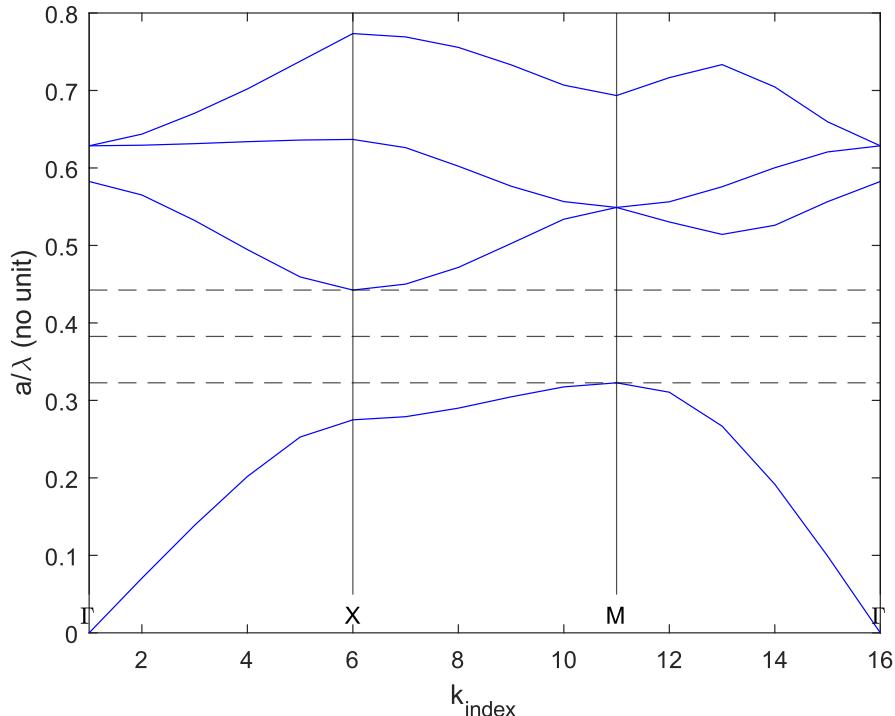


Figure 3.4: The photonic bandstructure of the square lattice of dielectric rods in air for TM modes (magnetic fields in the XY plane, electric field along Z). The dashed horizontal black lines indicate the bottom, middle and top of the fundamental bandgap. The solid vertical black lines indicate the critical k-points Γ , X , M .

Reminder on how to run such a set of scripts:

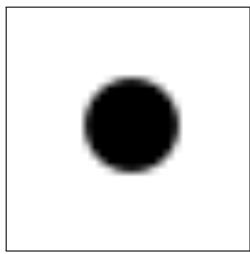
1. Open Cygwin.
2. Change into the directory with the scripts:

```
cd ~/KD6041-resources/MPB/square_lattice
```

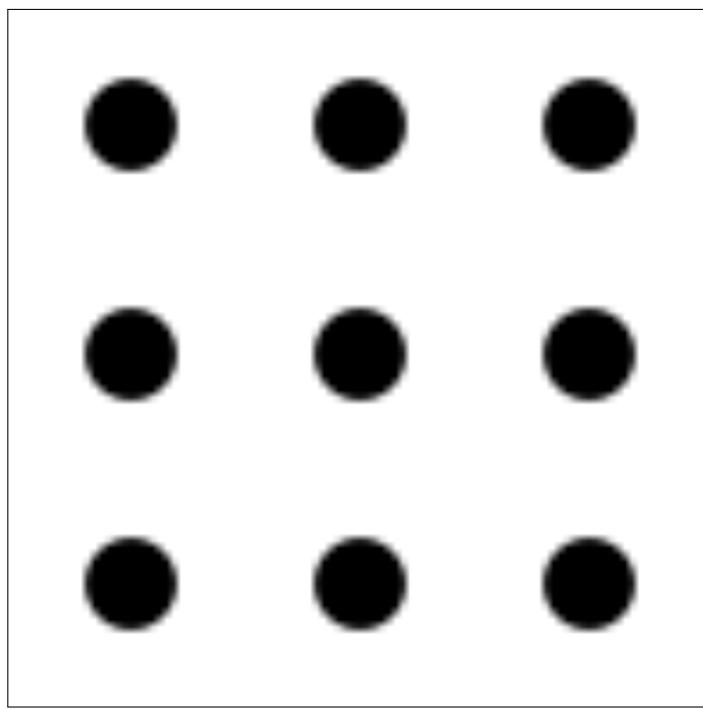
3. Run the Bash script in the Cygwin terminal:

```
bash square_lattice.sh
```

4. Open the Matlab script `square_lattice.m` in Matlab and run it from the directory it is in.



(a) The primitive unit-cell.



(b) A crystal made of 3x3 primitive unit-cells.

Figure 3.5: The generated output images.

```

;;;;; Define new parameters.
(define-param a12 1)
(define-param a3 1)

;;;;; Set parameters.
(set-param! num-bands 4)
(set-param! resolution 32)

;;;;; Define the geometry lattice.
(set! geometry-lattice
  (make lattice
    (basis1 1 0 0) ; a1
    (basis2 0 1 0) ; a2
    (basis3 0 0 1) ; a3
    (basis-size a12 a12 a3) ; length of the lattice vectors
    (size 1 1 no-size) ; make the simulation 2D
  )
)

;;;;; Define the geometry.
; The background material.
(set! default-material (make dielectric (epsilon 1)))
; The geometry
(set! geometry
  (list (make cylinder
    (center 0 0 0)
    (radius 0.2)
    (height infinity)
    (material (make dielectric (epsilon 8.9)))
  )))
)

;;;;; Define the desired k-points.
(set! k-points (list (vector3 0 0 0) ; Gamma -> k_index = 1
                      (vector3 0.5 0 0) ; X -> k_index = 1 + 4 + 1= 6
                      (vector3 0.5 0.5 0) ; M -> k_index = 6 + 4 + 1=11
                      (vector3 0 0 0))) ; Gamma -> k_index = 11+ 4 + 1=16
; Add 4 points between consecutive initial k-points.
(set! k-points (interpolate 4 k-points))

;;;;; Run the simulation.
(run-tm); run in TM mode

```

Listing 1: MPB CTL script: *KD6041-resources\MPB\square_lattice\square_lattice.ctl*

```

#!/bin/bash
mpb square_lattice.ctl | tee square_lattice.out
grep freq square_lattice.out > square_lattice.dat
# Create an image of the primitive unit-cell:
mpb-data -r -n 320 -m 1 square_lattice-epsilon.h5
h5topng -o square_lattice-1x1.png -X10 -Y10 square_lattice-epsilon.h5:data-new
# Create an image of 3x3 unit-cells:
mpb-data -r -n 320 -m 3 square_lattice-epsilon.h5
h5topng -o square_lattice-3x3.png -X10 -Y10 square_lattice-epsilon.h5:data-new

```

Listing 2: Bash script: *KD6041-resources\MPB\平方晶格\平方晶格.sh*

```

% Defining some default options to label the vertical lines:
set(groot,'defaultConstantLineLabelOrientation', 'horizontal');
set(groot,'defaultConstantLineLabelVerticalAlignment', 'bottom');
set(groot,'defaultConstantLineLabelHorizontalAlignment', 'center');

% load data
s = MPB_load_data('square_lattice.dat');

% get edges of the fundamental bandgap (between band 1 and 2)
gap_edges_fn = [max(s.fn(:,1)), min(s.fn(:,2))];

% plot against normalized frequency
figure; % create new figure
plot(s.kindex, s.fn, 'b-');
xlabel('k_{index}');
ylabel('a/\lambda (no unit)');
xlim([s.kindex(1), s.kindex(end)]);
% highlight edges and middle of the bandgap
yline(gap_edges_fn(1), 'k--'); % gap edge
yline(mean(gap_edges_fn), 'k--'); % gap middle
yline(gap_edges_fn(2), 'k--'); % gap edge
% indicate high-symmetry points
xline(1, 'k-', '\Gamma');
xline(6, 'k-', 'X');
xline(11, 'k-', 'M');
xline(16, 'k-', '\Gamma');

```

Listing 3: Matlab script: *KD6041-resources\MPB\平方晶格\平方晶格.m*

3.2.2 MPB code explanations

The main differences compared to the 1D photonic crystal code from last time are:

1. Making the simulation 2D by using `(size 1 1 no-size)`.
2. We used `(set! default-material (make dielectric (epsilon 1)))` to define a backfill material, which can be useful if you want to study the inverse structure with circular air holes in a dielectric.
3. The geometry now only consists of a single cylinder instead of two blocks.
4. We used the critical points Γ , X , M , Γ as main k-points based on the previously calculated coordinates. Our k-points will follow the outline of the triangle visible on the right in figure 3.2.
5. To create the plot, we plot against the k-index this time instead of against $k_x/(2\pi/a)$: `plot(s.kindex, s.fn, 'b-');`
6. We also added vertical lines to indicate the critical points critical k-points Γ , X , M :

```
xline(1, 'k-', '\Gamma');
xline(6, 'k-', 'X');
xline(11, 'k-', 'M');
xline(16, 'k-', '\Gamma');
```

Why 1, 6, 11 and 16? Because we added 4 interpolated k-points between the 4 initially specified k-points Γ , X , M , Γ , we will have $1+4+1+4+1+4+1=16$ k-points in total. However, among those 16 k-points the 4 initial ones correspond to:

- $k_{index} = 1 \rightarrow \Gamma$
- $k_{index} = 1 + 4 + 1 = 6 \rightarrow X$
- $k_{index} = 1 + 4 + 1 + 4 + 1 = 11 \rightarrow M$
- $k_{index} = 1 + 4 + 1 + 4 + 1 + 4 + 1 = 16 \rightarrow \Gamma$

3.2.3 MPB output analysis

It can be useful to look at the output of the MPB script we just ran, which should be visible in the Cygwin terminal after running the Bash script, but also in the generated output file `square_lattice.out`. You can use it to check the reciprocal lattice vectors and k-point coordinates calculated in question 2. See figure 3.6.

```

$ cd ~/KD6041-resources/MPB/square_lattice
~/KD6041-resources/MPB/square_lattice
$ bash square_lattice.sh
init-params: initializing eigensolver data
Computing 4 bands with 1.000000e-07 tolerance.
Working in 2 dimensions.
Grid size is 32 x 32 x 1.
Solving for 4 bands at a time.
Creating Maxwell data...
Mesh size is 3.
Lattice vectors:
  (1, 0, 0)
  (0, 1, 0)
  (0, 0, 1)
Cell volume = 1
Reciprocal lattice vectors (/ 2 pi):
  (1, -0, 0)
  (-0, 1, -0)
  (0, -0, 1)

Geometric objects:
  cylinder, center = (0,0,0)
    radius 0.2, height 1e+20, axis (0, 0, 1)
    epsilon = 8.9, mu = 1
Geometric object tree has depth 1 and 1 object nodes (vs. 1 actual objects)
Initializing epsilon function...
Allocating fields...
16 k-points:
  (0.0,0)    $\Gamma$ 
  (0.1,0,0)
  (0.2,0,0)
  (0.3,0,0)
  (0.4,0,0)
  (0.5,0,0)    $X$ 
  (0.5,0.1,0)
  (0.5,0.2,0)
  (0.5,0.3,0)
  (0.5,0.4,0)
  (0.5,0.5,0)    $M$ 
  (0.4,0.4,0)
  (0.3,0.3,0)
  (0.2,0.2,0)
  (0.1,0.1,0)
  (0,0,0)    $\Gamma$ 

Solving for band polarization: tm.
Initializing fields to random numbers...
elapsed time for initialization: 0 seconds.
epsilon: 1-8.9, mean 1.99274, harm. mean 1.13848, 14.5508% > 1, 12.5663% "fill"
Outputting square_lattice-epsilon...
solve_kpoint (0,0,0):
tmfreqs:, k index, k1, k2, k3, kmag/2pi, tm band 1, tm band 2, tm band 3, tm band 4
Solving for bands 2 to 4...
Finished solving for bands 2 to 4 after 8 iterations.
tmfreqs:, 1, 0, 0, 0, 0, 0, 0.582551, 0.628589, 0.628591
elapsed time for k point: 0 seconds.
solve_kpoint (0.1,0,0):
Solving for bands 1 to 4...
Finished solving for bands 1 to 4 after 5 iterations.
tmfreqs:, 2, 0.1, 0, 0, 0.1, 0.070498, 0.565122, 0.629353, 0.64367
elapsed time for k point: 0 seconds.
solve_kpoint (0.2,0,0):

```

The *direct lattice vector* coordinates **in the cartesian basis**.

The *reciprocal lattice vector* coordinates **in the cartesian basis**.

The *k-point* vector coordinates **in the reciprocal lattice basis**.

Figure 3.6: You can use MPB to calculate or validate coordinates of reciprocal lattice vectors and k-points.

4 Homework

4.1 Calculate the reciprocal lattice vectors of a 2D photonic crystal (triangular/hexagonal lattice)

Figure 4.1 shows a simple *triangular* (also called *hexagonal*) lattice. The *direct lattice* vectors are defined as follows:

$$\begin{cases} \vec{a}_1 = a_{12} \left(\frac{1}{2}\vec{x} - \frac{\sqrt{3}}{2}\vec{y} \right) \\ \vec{a}_2 = a_{12} \left(\frac{1}{2}\vec{x} + \frac{\sqrt{3}}{2}\vec{y} \right) \\ \vec{a}_3 = a_3\vec{z} \end{cases} \quad (4.1)$$

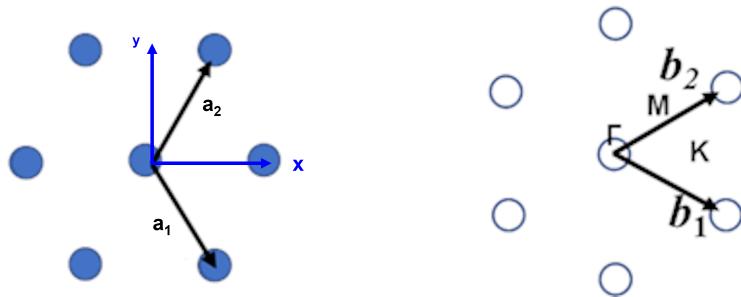


Figure 4.1: The real space and reciprocal space lattice of the triangular (or hexagonal) lattice.

1. Draw the following on top of figure 4.1:
 - a) On the real-space lattice:
 - i. The *Wigner-Seitz cell*.
 - ii. One *alternative unit cell*
 - b) On the reciprocal lattice:
 - i. The *Brillouin zone*
 - ii. An *irreducible Brillouin zone* with the critical points Γ , M , K :
 - A. Γ : Center of the Brillouin zone
 - B. M : Center of an edge
 - C. K : Corner joining two hexagonal faces

2. Calculate the coordinates in the provided cartesian basis ($\vec{x}, \vec{y}, \vec{z} = \vec{x} \times \vec{y}$) (in blue in figure 4.1) of the following:
 - a) The real (direct) lattice basis vectors: $\vec{a}_1, \vec{a}_2, \vec{a}_3$
 - b) The reciprocal lattice basis vectors: $\vec{b}_1, \vec{b}_2, \vec{b}_3$
 - c) The coordinates of the critical points: Γ, M, K

4.2 Compute the photonic band structure of a 2D photonic crystal (triangular/hexagonal lattice)

We want to compute the photonic band structure of a two-dimensional hexagonal lattice of air rods in a dielectric slab, as illustrated in figure 4.2. The lattice is the same as in section 4.1. The parameters will be as follows:

- $n_{backfill} = 3.2$
- $n_{holes} = 1$
- Distance between holes: $a = 0.4\mu m$, with $a = a_{12} = \|\vec{a}_1\| = \|\vec{a}_2\|$
- Hole radius: $r = 0.14\mu m$, i.e. $r/a = 0.35$

Tasks:

1. Define the corresponding geometry in MPB.
 2. Set the *k-points list* in MPB to Γ, M, K, Γ based on the coordinates you found in section 4.1, then add 4 extra interpolated points between them.
 3. Plot the first eight TE-bands (electric field \vec{E} in the XY plane, i.e. magnetic field \vec{H} aligned with Z) against the previously defined *k-points* in terms of:
 - a) normalized frequency a/λ
 - b) wavelength in μm .
- !** To compute TE bands instead of TM bands, you can just replace “*run-tm*” with “*run-te*”.
4. What is the range of the fundamental TE bandgap in μm ? (use *TE mode, resolution=32*)

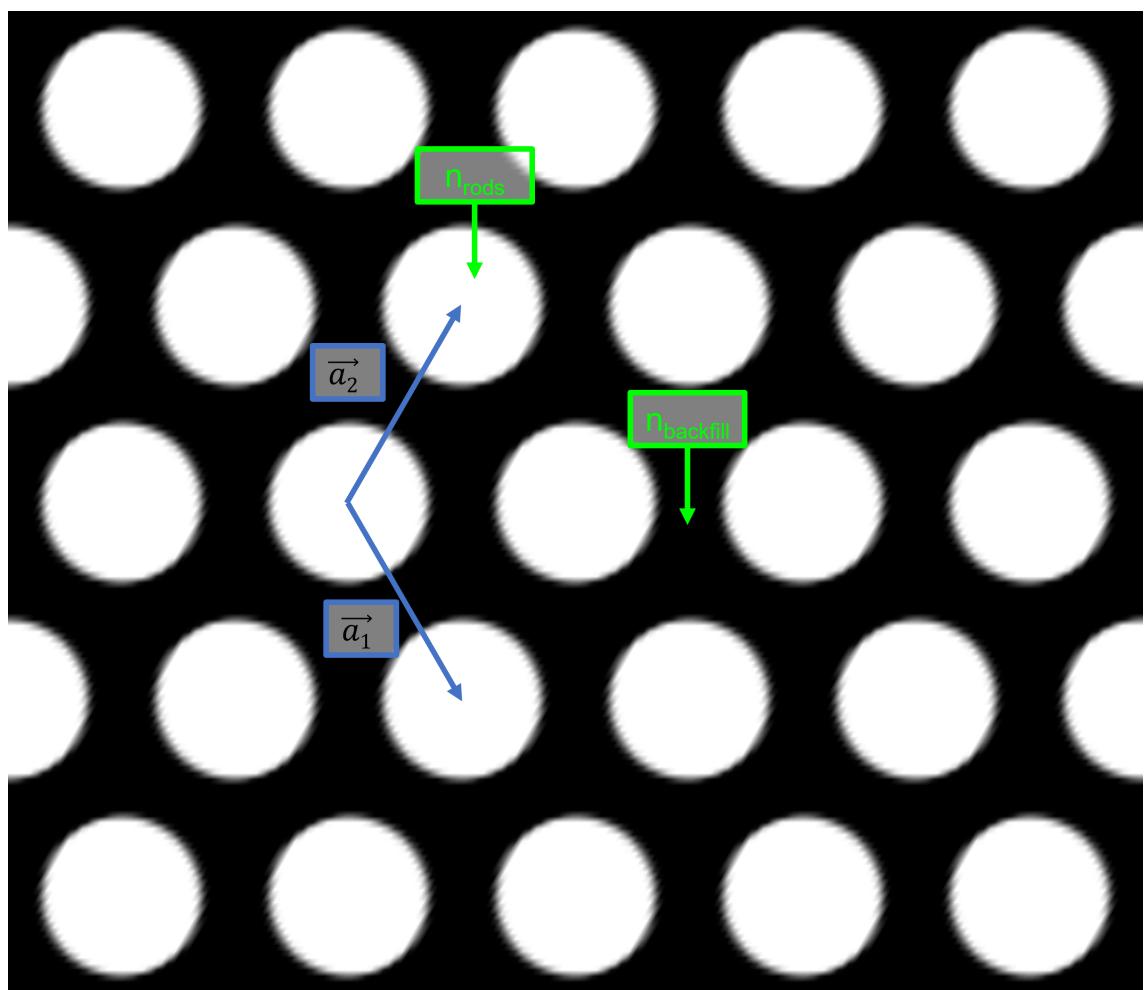


Figure 4.2: An inverse hexagonal photonic crystal.

5 Appendix

5.1 Vector operations

Consider two arbitrary 3D vectors \vec{u} and \vec{v} with the following coordinates in a cartesian basis:

$$\vec{u} = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix}, \vec{v} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} \quad (5.1)$$

The **dot product** is then defined as:

$$\vec{u} \cdot \vec{v} = u_1v_1 + u_2v_2 + u_3v_3 \quad (5.2)$$

And the **cross product** as:

$$\vec{u} \times \vec{v} = \begin{pmatrix} u_2v_3 - u_3v_2 \\ u_3v_1 - u_1v_3 \\ u_1v_2 - u_2v_1 \end{pmatrix} \quad (5.3)$$

$$\vec{u} \times \vec{v} = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \times \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} \cancel{u_2}v_3 - \cancel{u_3}v_2 \\ u_3v_1 - \cancel{u_1}\cancel{v_3} \\ u_1v_2 - \cancel{u_2}v_1 \end{pmatrix}$$

$$\vec{u} \times \vec{v} = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \times \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} \cancel{u_3}v_1 - \cancel{u_1}\cancel{v_3} \\ u_1v_2 - \cancel{u_2}v_1 \\ u_2v_3 - \cancel{u_3}v_2 \end{pmatrix}$$

$$\vec{u} \times \vec{v} = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \times \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} u_1v_2 - u_2v_1 \\ u_2v_3 - u_3v_2 \\ u_3v_1 - u_1v_3 \end{pmatrix}$$

Figure 5.1: One way of remembering the cross-product formula: To calculate each coordinate, draw a “Gamma” symbol between the two “next” rows of coordinates. Keep in mind that it is cyclic, so after 3 you go back to 1. Then you just calculate the difference between the product of the first two coordinates and the last two coordinates.

Bibliography

- [1] J. D. Joannopoulos, S. G. Johnson, J. N. Winn, and R. D. Meade, *Photonic Crystals: Molding the Flow of Light*, 2nd ed. Princeton University Press, 2008. [Online]. Available: <http://ab-initio.mit.edu/book/>