# High-resolution Sub-Wavelength Imaging Using Photonic Crystals

Student Name: Rounak Saraf
Roll Number: 2018183

BTP report submitted in partial fulfillment of the requirements
for the Degree of B.Tech. in Electronics and Communication Engineering

on 18th December 2020

**BTP Track:** Research Track

**BTP Advisor**
Dr. Sayak Bhattacharya

Indraprastha Institute of Information Technology
New Delhi

## Student's Declaration

I hereby declare that the work presented in the report entitled **"High-resolution Sub-Wavelength Imaging Using Photonic Crystals"** submitted by me for the partial fulfillment of the requirements for the degree of *Bachelor of Technology* in *Electronics and Communication Engineering* at Indraprastha Institute of Information Technology, Delhi, is an authentic record of my work carried out under guidance of **Prof. Sayak Bhattacharya**. Due acknowledgements have been given in the report to all material used. This work has not been submitted anywhere else for the reward of any other degree.

Rounak Saraf New Delhi, 18 December 2020

## Certificate

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Dr. Sayak Bhattacharya New Delhi, 18 December 2020

## Abstract

The project is on the topic of High-resolution sub-wavelength imaging using Photonic Crystals. It used the concept of angular spectrum representation. The sub-wavelength particles emit evanescent waves. The photonic crystals are able to pick up these evanescent waves at their surface. This is closely based on the phenomenon of solid state crystals that give rise to evanescent wave function of electrons at their surface.

The project includes the use of two softwares - MPB and MEEP. MPB is used for computing the definite-frequency eigenstates of Maxwell's equations in periodic dielectric crystals. MEEP implements the finite-difference time-domain (FDTD) method for computational electromagnetics.

## Acknowledgments

I would like to thank my advisor, Dr. Sayak Bhattacharya, who gave me the opportunity to work on this project. He guided me through the course of the project and gave his valuable feedback and advice that helped in the completion of the project. I am thankful to him.

# Contents

# Chapter 1

# Introduction and Objective

Photonic crystals are used to gather the evanescent waves emitted by sub-wavelength particles. This helps in high-resolution sub-wavelength imaging.

Our main objective of the project is :

A. **To understand the theory of Photonic Crystals.**
B. **To understand the working of MPB**
C. **To understand the working of MEEP**
D. **To plot the electromagnetic bands of a square and triangular lattice using MPB**
E. **To simulate certain objects using Meep and understand the transmittance spectra**

## MPB

- **MPB** is a free and open-source software package for computing the band structures, or dispersion relations, and electromagnetic modes of periodic dielectric structure.
- MPB is an acronym for *MIT Photonic Bands*.
- MPB computes definite-frequency eigenstates, or harmonic modes, of Maxwell's equations in periodic dielectric structures for arbitrary wave vectors, using fully-vectorial and three-dimensional methods.

**Simulations done -**

Simulation of square and triangular lattice of dielectric rods in air. Plotting the TE and TM modes along with the band gaps.

## Meep

**Meep** is a free and open-source software package for electromagnetics simulation via the finite-difference time-domain (**FDTD**) method spanning a broad range of applications.

**Simulations done -**

- Simulated a 1 micron thick silicon slab and observing the fields inside the slab as it passed through the slab.
- Simulated a silicon cylindrical rod through a Gaussian Source and observed the transmittance spectra on a computational cell.

# Chapter 2

# Theory of Photonic Crystals:

## 2.1 Photonic Crystals

Optical Analogue in which atoms and molecules are replaced by microscopic media with differing dielectric constants and the periodic potential is replaced by a periodic dielectric function. (or, equivalently a periodic index of refraction).

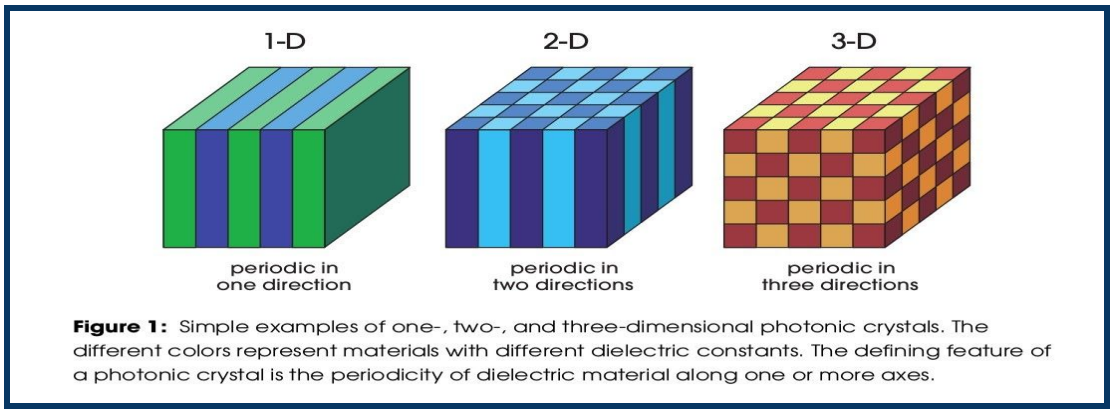



**Figure 1:** Simple examples of one-, two-, and three-dimensional photonic crystals. The different colors represent materials with different dielectric constants. The defining feature of a photonic crystal is the periodicity of dielectric material along one or more axes.

**Photonic Band Gap -**

Prevention of light source in a frequency range from any direction, any source and any polarization. (despite near-normal incidence)

## 2.2 Maxwell Equations

Basic Maxwell Equations -

Maxwell equations as a linear Hermitian eigenvalue problem.
$\nabla .B = 0$
$\nabla .D = Ƥ$
$\nabla \times E + \partial B/\partial t = 0$
$\nabla \times H - \partial D/\partial t = J$

Where, E,H → macroscopic electric and
magnetic field

D → displacement field
B → magnetic induction field
J → current density

- A power series relates D to E (Bloembergen, 1965) as :

  $D_i/\boldsymbol{\varepsilon}_0 = \Sigma_j \boldsymbol{\varepsilon}_{ij} E_j + \Sigma_{jk} \boldsymbol{\chi}_{ijk} E_j E_k + O(E^3)$

- Assuming that dielectric materials-
  - Have small field strengths.
  - Macroscopic and isotropic.
  - No explicit frequency dependence.
  - Transparent . ⇒ real and positive $\boldsymbol{\varepsilon}(r)$

$$D(r) = \boldsymbol{\varepsilon}_0 \boldsymbol{\varepsilon}(r)\, E(r)$$
$$B(r) = \mu_0 \mu(r)\, H(r)$$

- For most dielectric materials of interest : $\mu(r) \approx 1$ : $B(r) = \mu_0 H(r)$
- Refractive index, n, = $(\boldsymbol{\varepsilon}\, \mu)^{0.5}$

The updated Maxwell Equations now become -

- $\nabla \cdot H(r,t) = 0$
- $\nabla \cdot [\, \boldsymbol{\varepsilon}(r)\, E(r,t)\, ] = 0$
- $\nabla \times E(r,t) + \mu_0 ( \partial H(r,t) / \partial t ) = 0$
- $\nabla \times H(r,t) - \boldsymbol{\varepsilon}_0 \boldsymbol{\varepsilon}(r) ( \partial E(r,t) / \partial t ) = 0$

Harmonic Modes :

- We separate time dependence from spatial dependence by expanding the fields into a set of Harmonic Modes / states of the system.
- $H(r,t) = H(r)\, e^{-iwt}$ (spatial pattern or mode profile times a complex exponential)
- $E(r,t) = E(r)\, e^{-iwt}$
- For a given frequency,
  - $\nabla \cdot H(r) = 0$
  - $\nabla \cdot [\, \boldsymbol{\varepsilon}(r)\, E(r)\, ] = 0$

    ⇒ There are no point sources or sinks of displacement and magnetic fields in the medium.

Also, Transverse plane EM waves constitute the field configurations.

- Plane wave, $H(r) = a.e^{ikr}$ , where k is a wave vector.
- Transversality condition : a.k = 0

Explanation - $\nabla \cdot H(r) = 0 \;\Rightarrow\; \nabla(a.e^{ikr}) = 0 \;\Rightarrow\; a.k.r.e^{ikr} = 0 \;\Rightarrow\; a.k = 0$

## 2.3. Master Equation

Considering the last two of the updated Maxwell equations, we have them as -

- $\nabla \times E(r) - iw.\mu_0 H(r) = 0$
- $\nabla \times H(r) + iw.\boldsymbol{\varepsilon}_0 \boldsymbol{\varepsilon}(r) E(r) = 0$

We divide the second equation by $\boldsymbol{\varepsilon}(r)$ and then take the curl :

$\Rightarrow \nabla \times [ ( \boldsymbol{\varepsilon}(r)^{-1} \nabla \times H(r) ) + iw.\boldsymbol{\varepsilon}_0 E(r) ] = 0$
$\Rightarrow \nabla \times ( \boldsymbol{\varepsilon}(r)^{-1} \nabla \times H(r) ) + iw.\boldsymbol{\varepsilon}_0 ( \nabla \times E(r) ) = 0$
$\Rightarrow \nabla \times ( \boldsymbol{\varepsilon}(r)^{-1} \nabla \times H(r) ) + iw.\boldsymbol{\varepsilon}_0 ( iw.\mu_0 H(r) ) = 0$
$\Rightarrow \nabla \times ( \boldsymbol{\varepsilon}(r)^{-1} \nabla \times H(r) ) = w^2 \boldsymbol{\varepsilon}_0 \mu_0 H(r)$
$\Rightarrow \nabla \times ( \boldsymbol{\varepsilon}(r)^{-1} \nabla \times H(r) ) = w^2/c^2 . H(r)$ ,where $c = 1/(\boldsymbol{\varepsilon}_0 \mu_0)^{0.5}$ (vacuum speed light)

We obtain the following equation with manipulating the equations which is also known as the Master equation.

$$\boldsymbol{\nabla \times ( \varepsilon(r)^{-1} \nabla \times H(r) ) = w^2/c^2 . H(r)} \qquad \textbf{MASTER EQUATION}$$

- For a given $\boldsymbol{\varepsilon}(r)$, we solve master equation to know and find the modes H(r) and it's corresponding frequencies subject to transversality requirement. Then use : $\nabla \times E(r) - iw.\mu_0 H(r) = 0$ to get E(r).
  - It is guaranteed E satisfies transversality req. $\nabla . \boldsymbol{\varepsilon} E = 0$ (divergence of a curl is 0)
  - Hence only one such condition is required.

## 2.4. Eigenvalue Problem

- If the result of an operation on a fn. is the fn. itself, multiplied by some constant→ eigenvalue problem (fn. →eigenfunction or eigenvector) (some constant →eigenvalue)
- Master equation : $\nabla \times ( \boldsymbol{\varepsilon}(r)^{-1} \nabla \times H(r) ) = w^2/c^2 . H(r)$ can be written as:
  - $\Theta H(r) = w^2/c^2 . H(r)$ ,where $\Theta H(r) = \nabla \times ( \boldsymbol{\varepsilon}(r)^{-1} \nabla \times H(r) )$
  - $\Theta$ is a linear operator.

Inner Product of Two Vector Fields F(r) and G(r)

- $(F,G) \triangleq \int d^3r \; F^*(r).G(r)$ , where * is a complex conjugate.
- Also, $(F,G) = (G,F)^*$
- (F,F) is always real and non-negative.
- If F is an EM harmonic mode, usually, we can set (F,F) =1 due to freedom of scaling a mode by an overall multiplier. For F'(r) with $(F',F') \neq 1$,
  - $F(r) = F'(r) / (F',F')^{0.5}$ **Normalization**

## 2.5. Hermitian

Defining Hermitian

- An operator Ⓗ is Hermitian if (F, Ⓗ G) = (Ⓗ F, G) for any vector fields F(r) and G(r).
- PROOF : (F, Ⓗ G) = $\int d^3r\ F^* . \nabla \times ( \boldsymbol{\varepsilon}(r)^{-1} \nabla \times G )$
    - Using a vector identity: $\nabla . (F \times G) = ( \nabla \times F ).G - F.( \nabla \times G)$ and integrating it on both sides : $\int \nabla . (F \times G) = \int ( \nabla \times F ).G - \int F.( \nabla \times G)$

      According to divergence theorem : $\int \nabla . A\ dv = \oint A.ds$

      $\Rightarrow \oint (F \times G)\ ds = \int ( \nabla \times F ).G - \int F.( \nabla \times G)$

      $\Rightarrow 0 = \int ( \nabla \times F ).G - \int F.( \nabla \times G)$

      $\Rightarrow \int ( \nabla \times F ).G = \int F.( \nabla \times G)$ (for our usage to prove)

      Now, $\int d^3r\ F^* . \nabla \times ( \boldsymbol{\varepsilon}(r)^{-1} \nabla \times G ) = \int d^3r\ (\nabla \times F)^* . ( \boldsymbol{\varepsilon}(r)^{-1} \nabla \times G )$ (using the above)

      $\Rightarrow \int d^3r\ [\nabla \times ( \boldsymbol{\varepsilon}(r)^{-1} \nabla \times F ) ]^* . G = (Ⓗ F, G)$

      Hence proved : (F, Ⓗ G) = (Ⓗ F, G)

## 2.6. General Properties of Harmonic Modes

- **Eigenvalues of Ⓗ must be REAL numbers.**
  - PROOF : We have the Master equation : Ⓗ H(r) = $w^2/c^2$ . H(r)

    Taking its inner product with H(r) : ( H, Ⓗ H) = ($w^2/c^2$ ) . (H,H)

    Also, ( H, Ⓗ H)* = ($w^2/c^2$ )* . (H,H)

    We know, ( H, Ⓗ H) = ( Ⓗ H, H)  (property of Hermitian)

    We know, ( H, Ⓗ H) = ( Ⓗ H, H)*  (property of inner product)

  - ⇒ ( H, Ⓗ H)* = ($w^2/c^2$ )* . (H,H)
    ⇒ ( Ⓗ H, H) = ($w^2/c^2$ ) . (H,H)
    ⇒ ($w^2/c^2$ )* = ($w^2/c^2$ )
    ⇒ ($w^2$)* = $w^2$
    ⇒ $w^2$ is real.

- **Modes are orthogonal.**
  - Two harmonic modes $H_1(r)$ and $H_2(r)$ with different frequencies $w_1$ and $w_2$ have an inner product of zero due to the hermicity of Ⓗ.
  - ⇒ if $w_1 \neq w_2$, **(**$H_1$,$H_2$) = 0  **orthogonal**
  - ⇒ if $w_1 = w_2$, **(**$H_1$,$H_2$) ≠ 0  **degenerate**
  - For two real one-dimensional functions to be orthogonal, it means :

    $(f,g) = \int f(x)\, g(x)\, dx = 0$

    Orthogonal modes of different frequency have different no. of spatial nodes(where fn(x) = 0). Lower frequency modes implies lower no. of nodes.

- **EM energy and the Variation Principle.**
  - A mode tends to concentrate it's electric field energy in regions of high dielectric constant, while remaining orthogonal to the modes below it in frequency.
  - The smallest eigenvalue $w_0^2/c^2$ and thus the lowest frequency mode corresponds

to the field pattern that minimizes the functional :

- $U_f(H) \triangleq (H, \text{Ⓗ} H) / (H, H)$ RAYLEIGH QUOTIENT / EM 'Energy' Function

- Gradient : rate of change of $U_f$ w.r.t H.
  - $\delta U_f(H) \approx U_f(H + \delta H) - U_f(H)$
  - $\delta U_f = [(\delta H , G)] + (G, \delta H) ] /2$
  - $G(H) = 2/(H,H)\ (\text{Ⓗ}H - [(H,\text{Ⓗ}H) / (H,H)] H)$
- At extremum, $G = 0 \Rightarrow$ H is an eigenvector of Ⓗ.
  - $\Rightarrow \quad (\text{Ⓗ}H - [(H,\text{Ⓗ}H) / (H,H)] H) = 0 \quad \Rightarrow \quad \text{Ⓗ}H = U_f(H)\ H$
- Time-averaged physical energy contribution :
  - $U_E \triangleq \boldsymbol{\varepsilon}_0/4 \int d^3r\ \boldsymbol{\varepsilon}(r)\ | E(r) |^2$
  - $U_H \triangleq \mu_0/4 \int d^3r\ | H(r) |^2$
  - In harmonic modes, the physical energy is periodically exchanged between electric and magnetic fields. $U_E = U_H$
- Rate of energy transport, Poynting Vector, $S \triangleq \frac{1}{2}$ Re [ E* x H ]
  - This is the time average flux of EM energy , in a direction S, per unit time, per unit area, for a time harmonic field.
  - Component of S in a given direction : light intensity
  - Ratio of energy flux to energy density defines the velocity of energy transport.

## 2.7. Scaling Properties of Maxwell Equations

- The Master equation is scale invariant.

  $\nabla \times ( \boldsymbol{\varepsilon}(r)^{-1} \nabla \times H(r) ) = w^2/c^2 . H(r)$

  Now, for $\boldsymbol{\varepsilon}'(r)$ which is a compressed or expanded version of $\boldsymbol{\varepsilon}(r)$, we have $\boldsymbol{\varepsilon}'(r) = \boldsymbol{\varepsilon}(r/s)$, where s is a scale parameter.

  Now, using $r'=sr$ and $\nabla' = \nabla/s$

  $s\nabla' \times ( \boldsymbol{\varepsilon}(r'/s)^{-1} s\nabla' \times H(r'/s) ) = w^2/c^2 . H(r'/s)$ .....(i)

  But, we know $\boldsymbol{\varepsilon}'(r') = \boldsymbol{\varepsilon}(r'/s)$

  Dividing (i) by s and using the above :

  $\nabla' \times ( \boldsymbol{\varepsilon}'(r')^{-1} \nabla' \times H(r'/s) ) = w^2/c^2 . H(r'/s)$

  This is again the master equation but now with mode profile as follows:

$H'(r') = H(r'/s)$

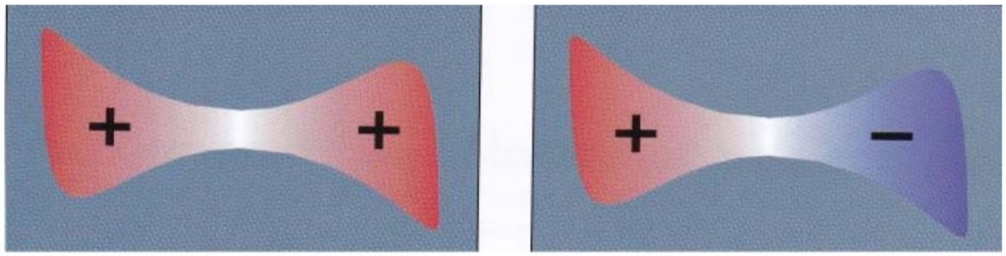
$w' = w/s$

- Rescaling the old mode profile and it's frequency yields a new mode profile. It simply means that the solution of the problem at one length scale determines the solution at all other length scales.

## 2.8. Symmetries

The symmetry of a dielectric structure helps in classifying the EM modes of the system.

### 2.8.1. Inversion Symmetry




even mode : H(r)=H(-r) odd mode : H(r) = -H(-r)

- Shown above is an inversion symmetry in a metallic cavity.
- If a certain pattern has mode H(r) with freq w, then pattern H(-r) must also be a mode with freq w.
- If these two modes are NOT degenerate, they are the same mode.
- Another mode might be a multiple of H(r) : H(-r) = α H(r)

- On inverting the system twice, using another factor of α, we get back H(r).
  - $\alpha^2$ H(r) = H(r) ⇒ α = 1 or -1
- A given non-degenerate mode must be :
  - Either invariant under inversion : H(-r) = H(r) ⇒ **EVEN**
  - Or it becomes its own opposite : - H(-r) = H(r) ⇒ **ODD**
- We can take linear combinations of degenerate modes to form new modes and classify them as even or odd.
- Due to symmetry under inversion, applying an inversion operator $\hat{O}_I$ on Hermitian, we have :
  - $\Theta = \hat{O}_I^{-1} \Theta \hat{O}_I$

- $\hat{O}_I Ⓗ = Ⓗ \hat{O}_I^{-1} \quad \Rightarrow \quad \hat{O}_I Ⓗ - Ⓗ \hat{O}_I^{-1} = 0$

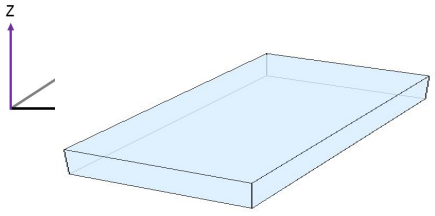
**Commutator**

- A commutator [A, B] of two operators A and B is defined as :
  - $[A, B] \triangleq AB - BA$
- Commutator itself is an operator.
- Under symmetry of inversion : $[\hat{O}_I , Ⓗ] = \hat{O}_I Ⓗ - Ⓗ \hat{O}_I^{-1} = 0$
- In case of non degeneracy : $\hat{O}_I Ⓗ = \alpha H$ , where $\alpha = 1$ or $-1$

## 2.8.2. Continuous Translational Symmetry

- A system with translational symmetry is unchanged by a translation through a displacement d.
  - $\check{T}_d \, \boldsymbol{\varepsilon}(r) = \boldsymbol{\varepsilon}(r-d) = \boldsymbol{\varepsilon}(r)$ or $[\check{T}_d , Ⓗ ] = 0$
- A system with continuous translation symmetry in the z direction is invariant under all of the $\check{T}_d$'s for that direction.
- A mode with functional form : $\mathbf{e^{ikz}}$ is an eigenfunction of any $\check{T}_d$ in z-direction.
  - $\check{T}_{dž} e^{ikz} = e^{ik(z-d)} = (e^{-ikd}) e^{ikz}$ where $(e^{-ikd})$ is the eigenvalue.
- A system with continuous translational symmetry in all three directions is a <u>homogeneous medium.</u>
  - $\boldsymbol{\varepsilon}(r)$ is a constant $\boldsymbol{\varepsilon}$ (=1 for free space). For this, the modes must have the form :
  - $H_k(r) = H_0 e^{ikr}$, where $H_0$ is any constant vector. This represents plane waves polarized in the direction of $H_0$.
  - These plane waves are solutions of the Master Equation with eigenvalue:

    $(w/c)^2 = |k|^2 / \boldsymbol{\varepsilon} \quad \Rightarrow \quad w = c|k| / \boldsymbol{\varepsilon}^{0.5} \quad \Rightarrow$ DISPERSION RELATION

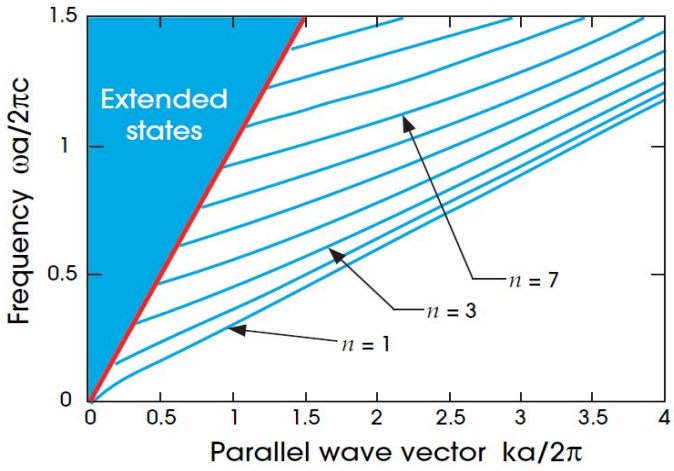



Shown : a plane of glass in which the dielectric function varies in z direction only . $\boldsymbol{\varepsilon}(r) = \boldsymbol{\varepsilon}(z)$.

This system is invariant under all translation operator of x-y plane.

- Modes can be classified according to their in-plane wave vectors : $k = k_x \hat{x} + k_y \hat{y}$

- The x-y dependence : $H_k(r) = e^{ik\rho}\, h(z)$ , where ρ is a vector confined to x-y plane.
- Intuition : three non.collinear points r, r+dx, r+dy --all have the same z value.
    - Due to symmetry, these three points have the same magnetic field amplitude.
    - Only difference could be the phase variation.
    - $k_x$ and $k_y$ universal to the plane.
    - We could distinguish different locations in the plane by their phase relationship.
    - Does NOT follow along z direction : each plane at different distance from the bottom of glass having different amplitude and phase.
- Lining up the modes in order of increasing frequency for given k.
- Let n be the particular mode's place in the line of increasing frequency.
- Each mode uniquely identified by (k,n) name.
- In case of degenerate modes (having same n & k), an additional index may be added.
- n ⟶⟶ band number
- If spectrum is discrete for given k, n is an integer, otherwise a continuous variable.
- As n grows, frequency of mode grows.




Explanation :

- Modes not confined to glass, extend into air and out to infinity -- Extended states
    - Resemble free space plane waves, i.e. superposition of plane waves with $w = c|k| = c\,(k_{II}^2 + k_{\perp}^2)^{0.5}$

- Since $k_\perp$ can take any value, there will be modes with every possible frequency greater than $ck_{II}$ .
- $w = ck_{II}$ ----- **LIGHT LINE**
- Frequencies *above the light* line form a continuous spectrum. This region of band structure is called the **light cone**.
- The modes in the light cone are solutions of the Snell's Law (less than critical angle).

- Frequencies below the light line have lower frequency relative to free space due to larger **ε** of glass.
  - These new solutions must be localized in the vicinity of the glass.
  - Only solutions in air are the ones with imaginary $k_\perp = \pm\ i\ [\ k_{II}^2 - (w^2/c^2)\ ]^{0.5}$ corresponding to the fields that decay exponentially (evanescent) away from the glass. They are called **index guided modes**.
  - They form a set of discrete frequencies, as they are localized in one direction.

### 2.8.3. Snell's Law

**Snell's law : a combination of two conservation laws due to symmetry**

- Conservation of frequency w (from linearity and time-invariance of Maxwell equations).
- Conservation of component $k_{II}$ of k (from continuous translational symmetry along interface).

We have $k_{II} = |k| \sin\theta$ where $|k| = nw/c$ (from dispersion relation)., where n is refractive index

Putting $k_{II}$ equal on both sides of the interface :

$|k| \sin\theta_1 = |k| \sin\theta_2$

$\Rightarrow n_1 w/c \sin\theta_1 = n_2 w/c \sin\theta_2$

$\Rightarrow n_1 \sin\theta_1 = n_2 \sin\theta_2$ ……**Snell's Law**

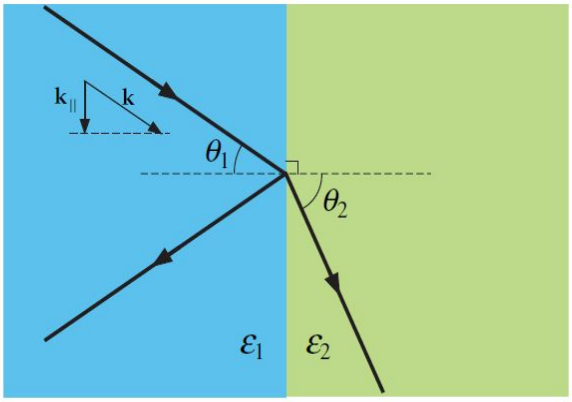
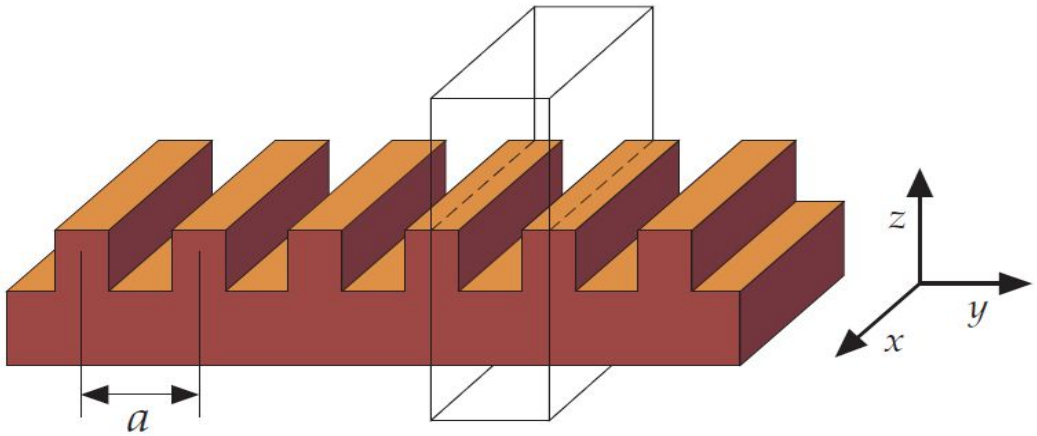



**Figure 4:** For a flat interface between two dielectrics $\varepsilon_1$ and $\varepsilon_2$, light can be described by a ray with an incident angle $\theta_1$ and a refracted angle $\theta_2$ given by Snell's law. When $\varepsilon_2 < \varepsilon_1$, we can have no solution $\theta_2$ for certain $\theta_1$, and the light undergoes total internal reflection. A generalization of this result follows from translational symmetry, which tells us that $k_\parallel$ is conserved.

## 2.8.4. Discrete Translational Symmetry

- Photonic Crystals have **discrete translational symmetry.**




- They are invariant under only distances that are a multiple of some fixed step length.

- In x-direction : continuous translational symmetry.
- In y-direction : discrete translational symmetry.
- Discrete symmetry : $\boldsymbol{\varepsilon}(r) = \boldsymbol{\varepsilon}(r \pm a)$ . On repetition, $\boldsymbol{\varepsilon}(r) = \boldsymbol{\varepsilon}(r + R)$, where $R = la$, l is an integer.
- Due to translational symmetry, Ⓗ must commute with
    - All translational operators in x-direction.
    - All translational operators for lattice vectors $R = la\,\hat{y}$ in y-direction.
- We can identify the modes of Ⓗ as simultaneous eigenfunctions of both translational operators -
    - $\check{T}_{dx\hat{}}\, e^{i\,kx\,x} = e^{i\,kx\,(x-d)} = e^{-i\,kx\,d}\, e^{i\,kx\,x}$
    - $\check{T}_{R}\, e^{i\,ky\,y} = e^{i\,ky\,(y-la)} = e^{-i\,ky\,la}\, e^{i\,ky\,y}$
- All modes with wave vector of the form $k_y + m\,(2\pi/a)$ forms a degenerate set; they all have the same $\check{T}_R$ eigenvalue of $e^{-i\,ky\,la}$ .
- Augmenting $k_y$ by an integral multiple of $b = 2\pi/a$ leaves the state unchanged.
- $b = b\,\hat{y}$ ……..*primitive reciprocal lattice vector*
- Any linear combination of these degenerate eigenfunctions is itself an eigenfunction with the same eigenvalue.
    - $H_{kx,ky}(r) = e^{i\,kx\,x} \Sigma_m\, c_{ky,m}(z)\, e^{i\,(ky+mb)\,y}$
    - $\Rightarrow\ e^{i\,kx\,x} \,.\, e^{i\,ky\,y} \,.\, u_{ky}(y,z)$
    - c's are the expansion coefficients to be determined by explicit solution.
- u(y,z) is aperiodic function in y (by construction)
    - $u(y + la\,,\, z) = u\,(y,z)$
- A discrete periodicity in y-direction leads to a y-dependence in H which is the product of plane wave and a y-periodic function.

**$H(....,y,...) \propto e^{i\,ky\,y} \,.\, u_{ky}(y,...)$ Bloch's Theorem**

- **Bloch's state** with wave vectors $k_y$ differing by integral multiples of $b = 2\pi/a$ are identical.
- Mode frequency must also be periodic in $k_y$ : $w(k_y) = w(k_y + mb)$.

- We need only consider $k_y$ to exist in the range $-\pi/a < k_y <= \pi/a$. This region of important, non-redundant values of $k_y$ is **Brillouin Zone.**

## 2.8.5. Reciprocal Lattice

- Suppose we have f(r), a function periodic on lattice.
  - f(r) =f(r+R) for all R that translate the lattice into itself ( i.e. Connect one lattice point to another). Eg. **ε**(r)
  - We take fourier transforms i.e. build f(r) out of plane waves with various wave vectors.
- $f(r) = \int g(q)\ e^{iqr}\ d^3r$ where g(q) is coefficient on the plane wave with wave vector q.
- $f(r+R) = \int g(q)\ e^{iqr}\ e^{iqR}\ d^3r = \int g(q)\ e^{iqr}\ d^3r = \int g(q)\ e^{iqr}\ d^3r$
- The periodicity of f tells us that it's fourier transform g(q) has a property :

  $g(q) = g(q)\ e^{iqR}$

  This is possible only if $g(q) = 0$ or $e^{iqR} = 1$.

  - The transform g(q) is zero everywhere, except for spikes at the values of q such that $e^{iqR} = 1$ for all R.
  - If we're building a lattice periodic function f out of plane waves, we need use only those plane waves with wave vector q such that : $e^{iqR} = 1$ for all R.
  - In analogy to 1D : if we're building f(x) with period τ out of sinusoids, we need only use the fundamental sinusoids with period τ and its harmonics with period τ/2, τ/4, so on, forming a fourier series.
  - Those vectors q such that $e^{iqR} = 1$ or equivalently $q.R = 2\pi N$ are called **reciprocal lattice vectors.(G)**
    - They form a lattice of their own.
    - Adding $G_1$ and $G_2$ yields another reciprocal lattice vector.

So, f(r) can be built with approximate weighted sum over all the reciprocal lattice vectors :

- $f(r) = \Sigma_G f_G e^{iG.r}$

## 2.8.6. Brillouin Zone

- Modes in Bloch form : $H_k(r)\, u_k(r) = e^{ikr}\, u_k(r+R)$
- A mode with wave vector k and a mode with wave vector k+G are the same mode if G is a reciprocal lattice vector.
- If k is incremented by G, the phase b/w the cells is incremented by G.R(=2πN) and it's not a phase difference. So, incrementing k by G results in same physical mode.
- We can restrict our attention to a finite zone in reciprocal space in which one cannot get from one part of the volume to another by adding G.
- All the values of k outside this zone can be reached from within the zone by adding G and are therefore redundant labels.
- The one closest to k =0 is the **First Brillouin Zone.**

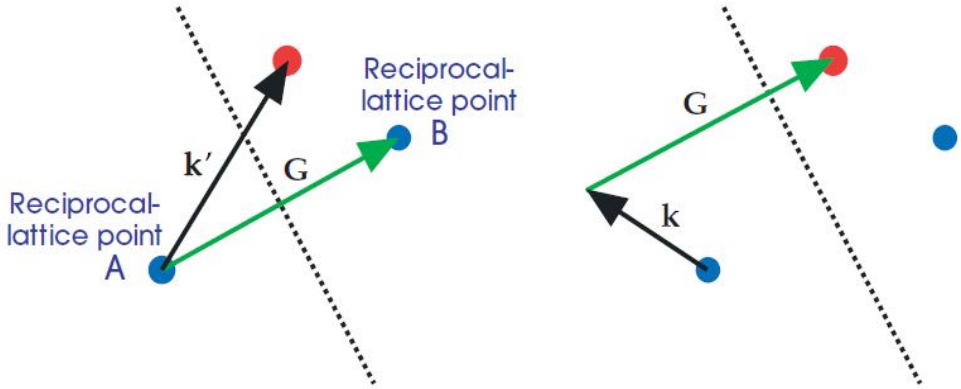



**Figure 1:** Characterization of the Brillouin zone. The dotted line is the perpendicular bisector of the line joining two reciprocal lattice points (blue). If we choose the left point as the origin, any lattice vector (such as $\mathbf{k}'$) that reaches to an arbitrary point on the other side (red) can be expressed as the sum of a same-side vector (such as $\mathbf{k}$) plus a reciprocal lattice vector $\mathbf{G}$.

## 2.8.7. Photonic Band Structure

- Modes of photonic crystals are a family of continuous functions $w_n(k)$ , indexed in order of increasing frequency by band number.
- The information contained in these functions is called **band structure**.
- Band structure is useful to predict optical properties.
- Frequency of each band varies continuously as k varies continuously.
- For each k, an infinite number of modes with discretely spaced frequencies , labelled by band number n.

## 2.8.8. Rotational Symmetry

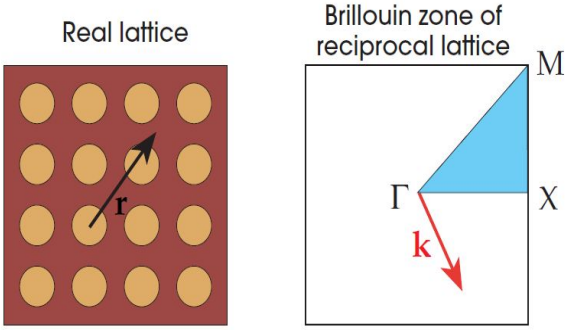
- Suppose the operator (3x3 matrix) □(ň, α) rotates vectors by an angle α about ň axis.
- To rotate a vector field f(r), we take vector f and rotate it with □. : f' = □ f
- We take argument r of the vector field : r' = $□^{-1}$ r
- ⇒ f'(r') = □ f (r') = □ f ($□^{-1}$ r)
- Defining vector rotator $\hat{O}_R$ . f(r) = □ f ($□^{-1}$ r)
- If the rotation by □ leaves the system invariant, then [ Ⓗ , $\hat{O}_R$ ] = 0
- Ⓗ ($\hat{O}_R$ $H_{kn}$ ) = $\hat{O}_R$ (Ⓗ $H_{kn}$) = $(w_n(k)/c)^2$ $\hat{O}_R$ $H_{kn}$
- $\hat{O}_R$ $H_{kn}$ also satisfies the master equation with the same eigenvalue as $H_{kn}$.
- $\check{T}_R$ ($\hat{O}_R$ $H_{kn}$ ) = $\hat{O}_R$ ($\check{T}□^{-1}{}_R$ $H_{kn}$)
  - $\hat{O}_R$( $e^{-i(kR'R)}$ $H_{kn}$) = $e^{-i(kR'R)}$ $\hat{O}_R H_{kn}$
  - Since $\hat{O}_R H_{kn}$ is the Bloch state with wave vector $R_k$ and has the same eigenvalue as $H_{kn}$
    - ⇒ $w_n$ (□ k) = $w_n$ (k)
- Whenever a photonic crystal has a rotation, mirror reflection, or inversion symmetry, the

$w_n(k)$ functions have that symmetry as well.

- This collection of symmetry operations is called the **point group** of the crystal.

## 2.8.9. Irreducible Brillouin Zone

- Since $w_n(k)$ possesses the full symmetry of the point group, we need not consider them at every k point in the Brillouin zone.
- *The smallest region within the Brillouin zone for which $w_n(k)$ are not related by symmetry is called the* ***irreducible Brillouin zone.***
- Eg. square lattice photonic crystal.




The blue shaded region is the irreducible Brillouin zone, constituting 1/8th of the total Brillouin zone. The rest of the Brillouin zone consists of redundant copies of the irreducible zone.

## 2.8.10 Mirror Symmetry

- Considering the case of notched dielectric, the system is invariant under mirror reflections in the yz and xz planes.
- We define a mirror operator $\hat{O}_{Mx}\, f(r) = M_x\, f(M_x r)$
- Two applications of it restore the system back : possible eigenvalues of $\hat{O}_{Mx}$ are +1 or -1
- $[\Ⓗ, \hat{O}_{Mx}] = 0$
- $\hat{O}_{Mx}$ is just a Bloch state with reflected wave vector $M_x k$ : $\hat{O}_{Mx}\, H_k = e^{i\Phi} H_{Mx\,k}$
- However, if $M_x k = k$ : $\hat{O}_{Mx}\, H_k(r) = \pm H_k(r) = M_x\, H_k(M_x r)$
- Modes of every 2-D photonic crystal can be classified into two distinct polarizations :
  - $(E_x , E_y , H_z)$ : Transverse Electric modes (TE)
  - $(H_x , H_y , E_z)$ : Transverse Magnetic modes (TM)

## 2.8.11 Time Reversal Invariance

- Taking complex conjugate of the Master equation :
  - $(\Ⓗ H_{kn})^* = w_n^2(k)/c^2 . H_{kn}^* \quad \Rightarrow \quad \Ⓗ H_{kn}^* = w_n^2(k)/c^2 . H_{kn}^*$ (since eigenvalues are real for lossless materials)
  - $H_{kn}^*$ satisfies the same equation as $H_{kn}$ , with the same eigenvalue.
  - Also, $H_{kn}^*$ is a Bloch state at -k
  - $\Rightarrow w_n(k) = w_n(-k)$
- The frequency bands can have inversion symmetry even if the crystal doesn't.
- Taking complex conjugate of $H_{kn}$ is equivalent to reversing the sign of time t in the Maxwell equations : $H(r,t) = H(r)\, e^{-iwt} \Rightarrow H(r,t)^* = H(r)\, e^{iwt} \Rightarrow \nabla . H(r,t)^* = 0 \Rightarrow \nabla . H(r) = 0$

  $w_n(k) = w_n(-k)$ is a consequence of the time reversal symmetry of the Maxwell equations.

- Exception : magneto-optic materials , $\in^* \neq \in$ even for lossless materials. Here $\in$ is a dielectric tensor (3x3 complex Hermitian matrix)

## Chapter 3

## Chapter 3

# MPB : Results

## MPB

MPB computes definite-frequency eigenstates, or harmonic modes, of Maxwell's equations in periodic dielectric structures for arbitrary wave vectors, using fully-vectorial and three-dimensional methods. MPB is an acronym for MIT Photonic Bands.

### Units in MPB

The unit in MPB is taken to be lattice constant ‘a’. In the calculations of our measurements for a particular distance, we scale the distances by ‘a’. However, we can choose any unit of our choice in MPB. In accordance with the scale-invariance property of the Maxwell equations, we can

apply a solution to all length scales.

## Computing Bands of a Square Lattice of Dielectric Rods in Air

We consider a 2-D photonic crystal that is following discrete translational symmetry along x and y axes and is homogeneous along the z-axis.

A sample of 2-D photonic crystal is a square lattice of dielectric columns.

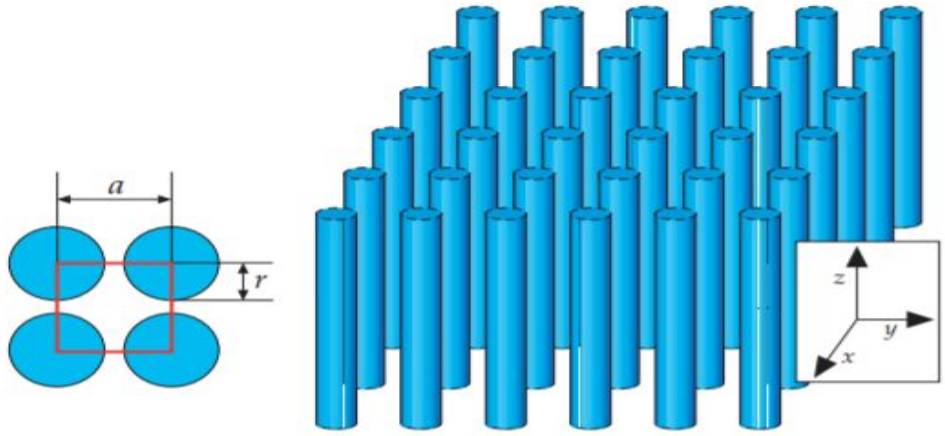



**Figure 1:** A two-dimensional photonic crystal. This material is a square lattice of dielectric columns, with radius $r$ and dielectric constant $\varepsilon$. The material is homogeneous along the $z$ direction (we imagine the cylinders are very tall), and periodic along $x$ and $y$ with lattice constant $a$. The left inset shows the square lattice from above, with the unit cell framed in red.

Consider the light to propagate in the xy plane of a square array of dielectric columns with lattice constant 'a'. The plot is obtained by the simulation of the code as given in **appendix A**. The horizontal axis shows the value of the wave vector $k_{ll}$ which is in plane.

The plot shows the propagation along the edge of the irreducible Brillouin zone from $\tau$ to X to M which is along the triangular edge of the irreducible Brillouin zone.

This is plotting only along the edge of the Brillouin zone as the maxima and minima of a given band occurs at the edges of the Brillouin zone.

## Gap-Midgap Ratio

The frequency of the middle gap is considered to be as $w_m$. The frequency width is noted as $\Delta w$. The gap midgap ratio is then defined as the ratio $\mathbf{\Delta w/w_m}$. This is usually expressed as a percentage. The gap-midgap ratio remains the same even if the system undergoes scaling. So

as to consider the ‘size’ of the gap, we refer to the gap-midgap ratio for it.

This is reason for which the frequency and wave vectors are plotted in dimensionless units - wa/2pi*c and ka/2*pi. The dimensionless frequency is equivalent to a/λ with λ being the vacuum wavelength (λ = 2pi*c/w).

## Simulation for the Square Lattice

Upon running the script in the terminal after activating conda, we get our outputs. It contains information about the band structure that can be extracted to form a plot. Also, the output contains the list of gaps detected in the computed bands. A certain drawback is the presence of false positives of the band gap data. This is primarily due to two reasons, them being -

- Ignoring the band gaps occurring inside the edges. We must compute band gaps inside the Brillouin zone as well.
- The crossing of the band gaps that results in a false gap. This happens because the software assumes that the band gaps are never crossing. This could be rectified when one computes k points very closely to the crossing to identify.

This is a sample output obtained in the terminal, determining the gap-midgap ratio :-

```
Finished solving for bands 1 to 8 after 9 iterations.
tmfreqs:, 27, 0.0555556, 0.0555556, 0, 0.0785674, 0.0507041, 0.527786, 0.553692, 0.575794, 0.816619, 0.859685, 0.947929, 1.05545
elapsed time for k point: 0.004162311553955078
solve_kpoint (0,0,0):
Solving for bands 2 to 8...
    linmin: converged after 4 iterations.
    iteration    1: trace = 4.421072463109207 (0.42765% change)
    linmin: converged after 4 iterations.
    iteration    2: trace = 4.412233703638528 (0.200123% change)
    linmin: converged after 4 iterations.
    iteration    3: trace = 4.40852925538212 (0.0839938% change)
    linmin: converged after 3 iterations.
    iteration    4: trace = 4.404011857683098 (0.102522% change)
    linmin: converged after 4 iterations.
    iteration    5: trace = 4.399647671300552 (0.0991448% change)
    linmin: converged after 4 iterations.
    iteration    6: trace = 4.397893394509974 (0.0398811% change)
    linmin: converged after 4 iterations.
    iteration    7: trace = 4.397405435852761 (0.0110959% change)
    linmin: converged after 4 iterations.
    iteration    8: trace = 4.397260290800525 (0.00330075% change)
    linmin: converged after 2 iterations.
    iteration    9: trace = 4.397198362994489 (0.00140834% change)
    linmin: converged after 2 iterations.
    iteration   10: trace = 4.397178187076239 (0.000458837% change)
    linmin: converged after 2 iterations.
    iteration   11: trace = 4.397175263229611 (6.64937e-05% change)
    linmin: converged after 2 iterations.
    iteration   12: trace = 4.39717466504361 (1.36039e-05% change)
    linmin: converged after 2 iterations.
    iteration   13: trace = 4.39717442329009 (5.49793e-06% change)
Finished solving for bands 2 to 8 after 13 iterations.
tmfreqs:, 28, 0, 0, 0, 0, 0, 0.546329, 0.555399, 0.555401, 0.816082, 0.860677, 0.954062, 1.07925
elapsed time for k point: 0.005299568176269531
Band 1 range: 0.0 at Vector3<0.0, 0.0, 0.0> to 0.2822092729718548 at Vector3<0.5, 0.5, 0.0>
Band 2 range: 0.41675257516371617 at Vector3<0.5, 0.0, 0.0> to 0.5463289546043879 at Vector3<0.0, 0.0, 0.0>
Band 3 range: 0.49727255218376526 at Vector3<0.5, 0.5, 0.0> to 0.5612069639752393 at Vector3<0.5, 0.0, 0.0>
Band 4 range: 0.5554008715829053 at Vector3<0.0, 0.0, 0.0> to 0.717924625812616 at Vector3<0.5, 0.0, 0.0>
Band 5 range: 0.7375086710875903 at Vector3<0.5, 0.0, 0.0> to 0.8700663745711676 at Vector3<0.5, 0.5, 0.0>
Band 6 range: 0.8318399877374086 at Vector3<0.2777777777777778, 0.2777777777777778, 0.0> to 0.8752897239003717 at Vector3<0.5, 0.5, 0.0>
Band 7 range: 0.8700937099345265 at Vector3<0.5, 0.0, 0.0> to 0.9540623456619091 at Vector3<0.0, 0.0, 0.0>
Band 8 range: 0.8786485882250658 at Vector3<0.5, 0.5, 0.0> to 1.079245455532408 at Vector3<0.0, 0.0, 0.0>
Gap from band 1 (0.2822092729718548) to band 2 (0.41675257516371617), 38.498038927516774%
Gap from band 4 (0.717924625812616) to band 5 (0.7375086710875903), 2.6911635616258924%
total elapsed time for run: 0.17610383033752441
done
```

We then experiment with the parameters in order to expect a change in the accuracy, time and precision of the electromagnetic bands and the following results are obtained.

The following table shows the time taken for the simulation with respect to the computational grid.

| Computational Grid | Elapsed Run-Time |
|---|---|
| 12 X 12 | 0.8836 s |
| 40 X 40 | 3.9872 s |
| 60 X 60 | 7.5492 s |
| 100 X 100 | 18.7813 s |

As expected the time taken is directly proportional to the size of the computational grid.

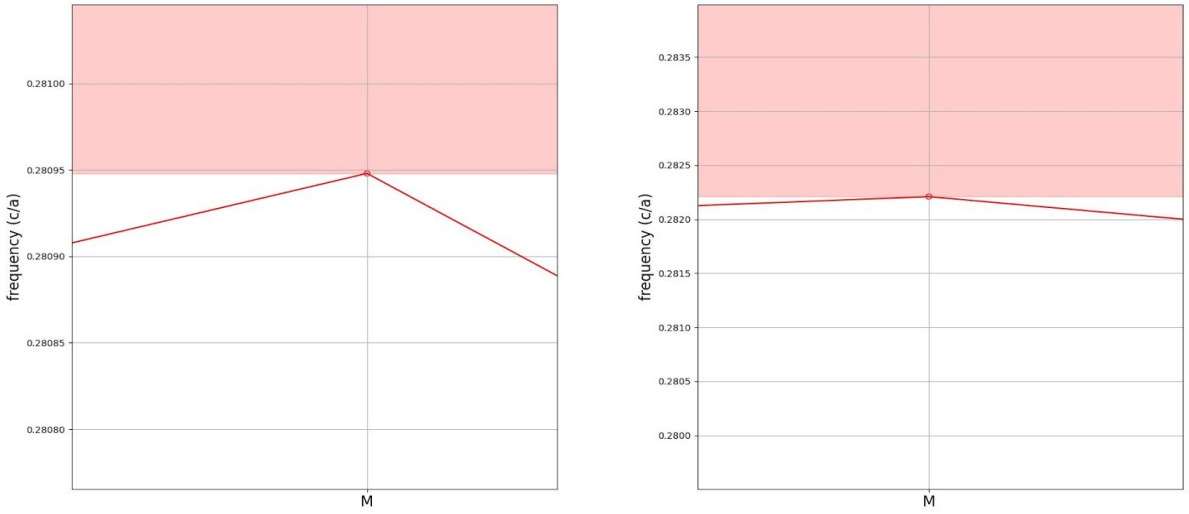
The epsilon here is corresponding to the frequency independent isotropic relative permittivity or the dielectric constant. The band gap here is directly proportional to the value of epsilon.

The below table shows the resolution and elapsed run-time :-

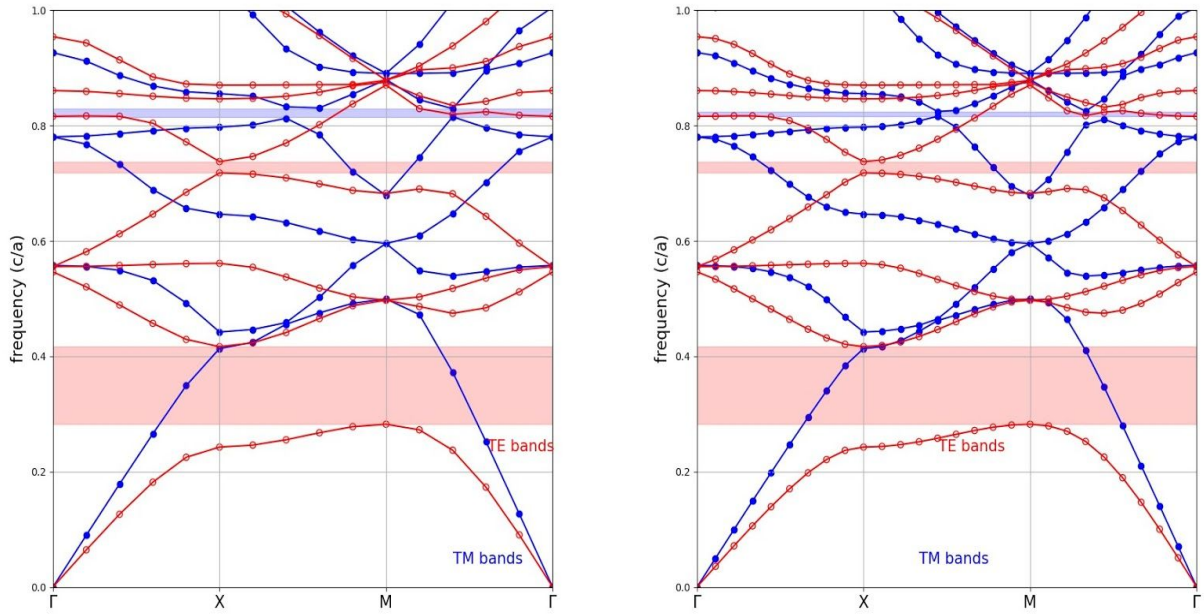
| Resolution | Time Taken |
|---|---|
| 4 pixels/unit | 2.8085 s |
| 8 pixels/unit | 3.4592 s |
| 16 pixels/unit | 4.7327 s |
| 32 pixels/unit | 22.3370 s |
| 64 pixels/unit | 84.4088 s |

We also tried interpolating the resolution to obtain a continuous curve between resolution and time taken in order to obtain a function or relation between them.

The following graph shows the difference between two resolutions taken in order to show their difference in accuracy. The resolutions taken are - 8 pixels/unit and 16 pixels/unit. The graph with 16 pixels/unit is expectedly more accurate.



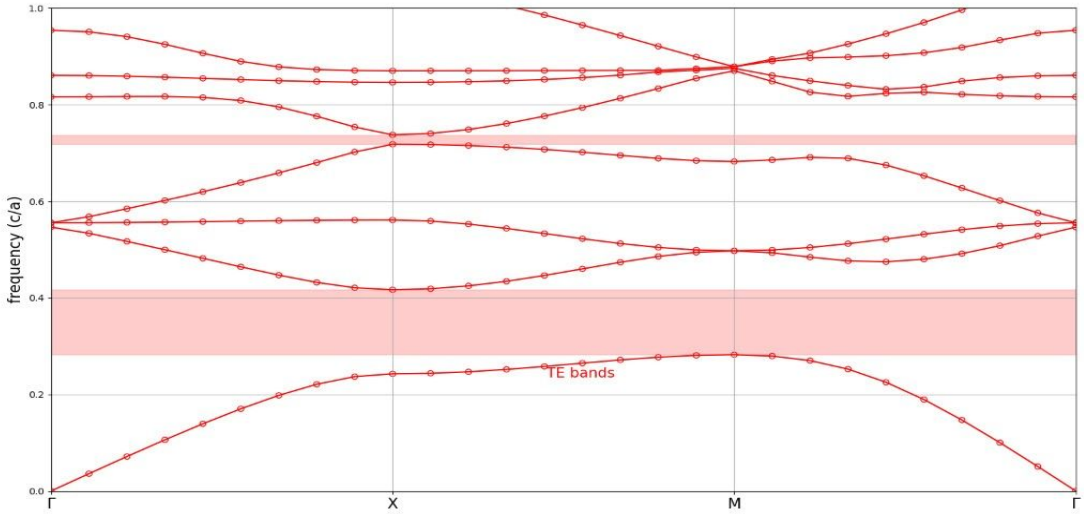

The following graph shows the difference between graphs when there is a difference in the number of points taken for interpolation. The first curve shows interpolation for 4 points and the second curve shows interpolation for 8 points.

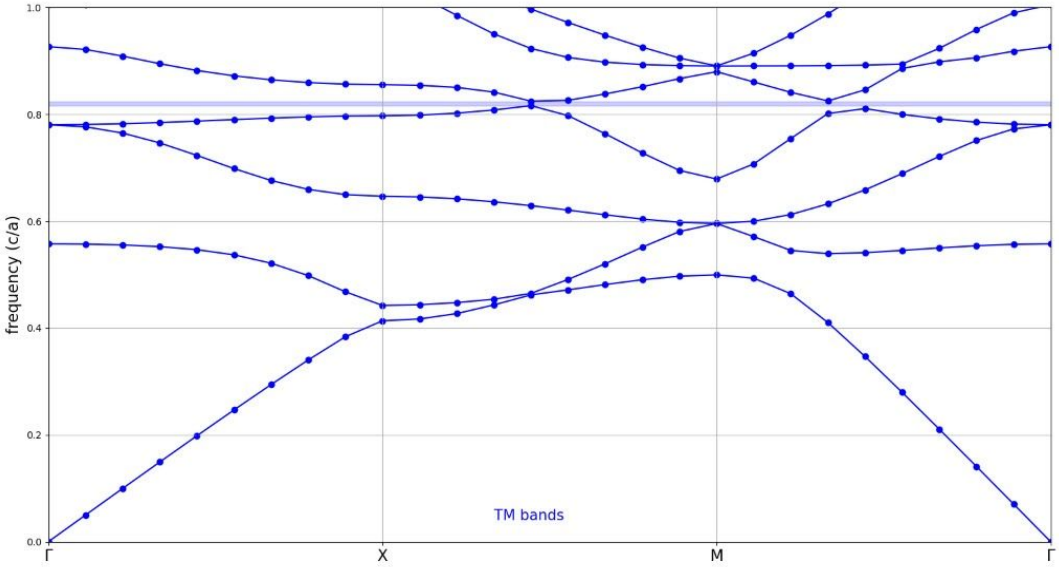


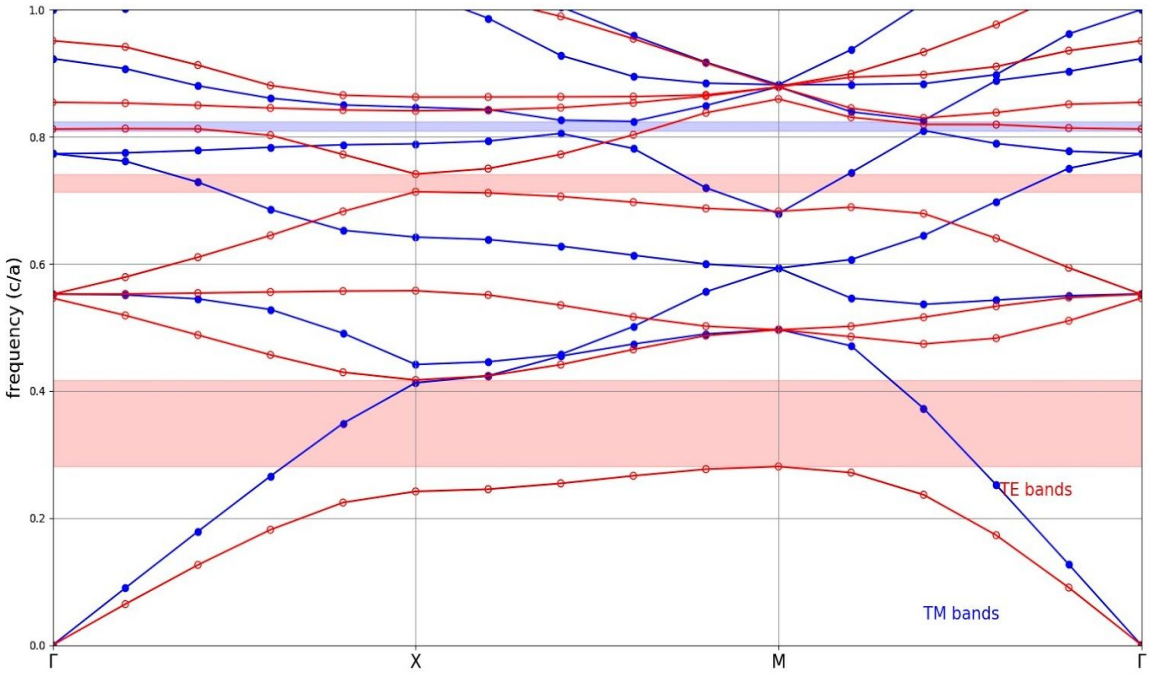

The curve with 8 points for interpolation gives us a more continuous and accurate curve as expected.

## Plotting the TE and T.M. Bands for Square Lattice

The code for potting the bands and the band gaps is given in Appendix A. The blue shaded region and the blue curves are for TM modes whereas the red shaded region and the red curves are for TE modes.


frequency (c/a)
1.0
0.8
0.6
0.4
0.2
0.0
TE bands
Γ
X
M
Γ


frequency (c/a)
1.0
0.8
0.6
0.4
0.2
0.0
TM bands
Γ
X
M
Γ




## Computing Bands of a Triangular Lattice of Dielectric Rods in Air

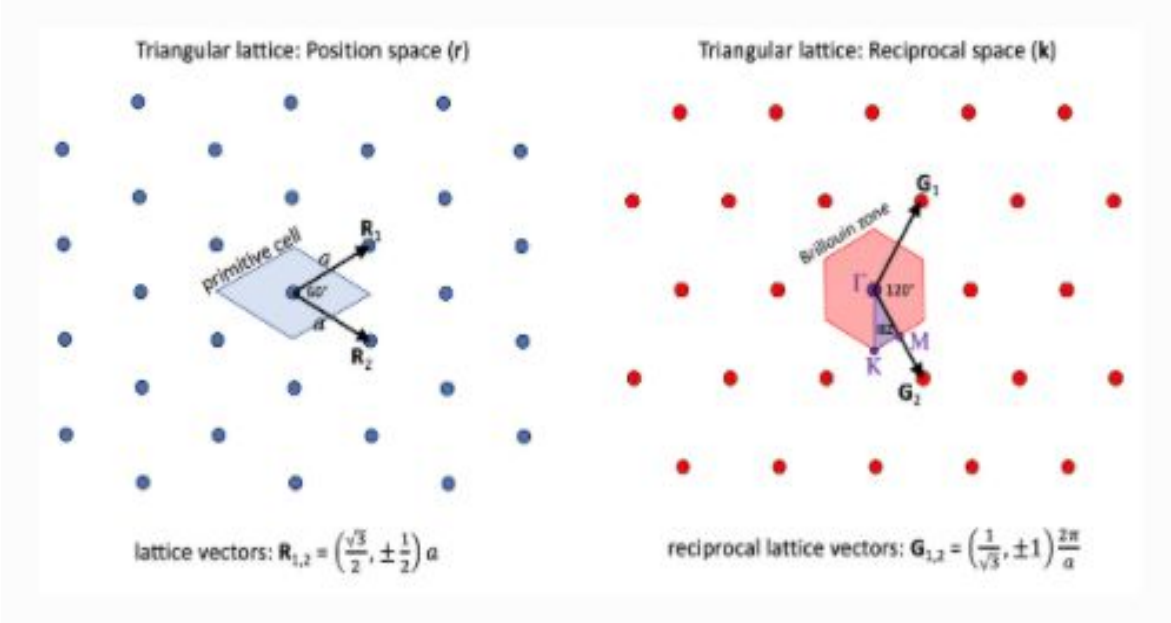
We follow the same procedure for the code as the square lattice part, except the following changes are made -

- The geometry lattice is changed from square to a triangular lattice by setting the basis vectors. The first two basis vectors pointing are set at 30 degrees above and below the x axis. The default value of basis 3 is the z-axis. The basis property only specifies the direction of lattice basis vectors and not their length (length is by default unity).

- The simulation is done at the corners of the irreducible Brillouin zone of the triangular lattice and these changes are made in the code.

The irreducible Brillouin zone of the triangular lattice is different from that of the square lattice.

The below diagram shows the triangular lattice and its irreducible Brillouin zone.




## Simulation for Triangular Lattice

The script for calculating the band gaps, and band information for the TE and TM modes is given in the **Appendix A.**

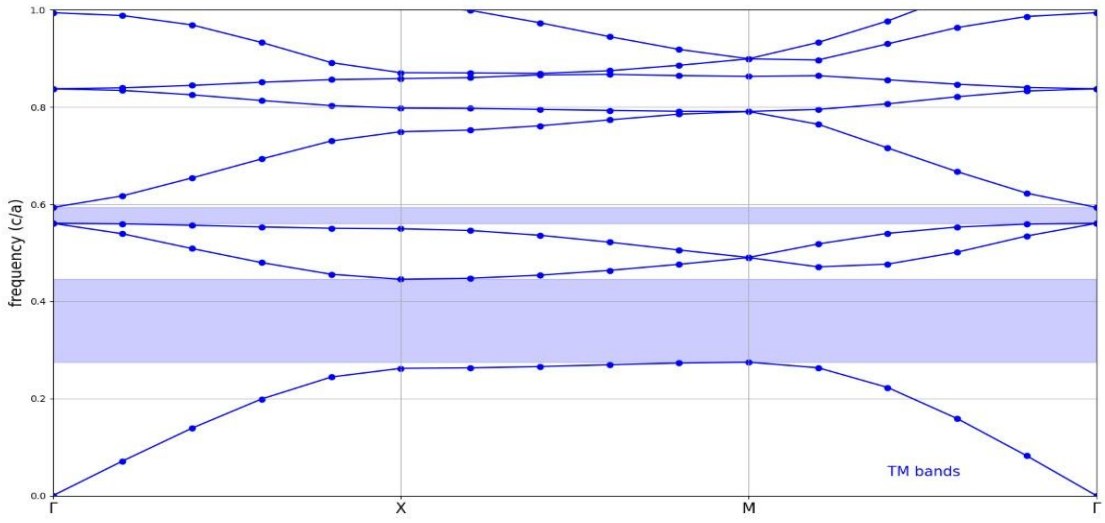
Below is the sample output containing values such as gap-midgap ratio, simulation time is obtained on running the code -

```
Solving for bands 2 to 8...
    linmin: converged after 4 iterations.
    iteration    1: trace = 6.298758274560811 (0.895537% change)
    linmin: converged after 4 iterations.
    iteration    2: trace = 6.245460231327736 (0.849763% change)
    linmin: converged after 3 iterations.
    iteration    3: trace = 6.223341785702516 (0.354781% change)
    linmin: converged after 3 iterations.
    iteration    4: trace = 6.211980615091755 (0.182724% change)
    linmin: converged after 3 iterations.
    iteration    5: trace = 6.208148485367179 (0.0617084% change)
    linmin: converged after 3 iterations.
    iteration    6: trace = 6.206838432143011 (0.0211044% change)
    linmin: converged after 4 iterations.
    iteration    7: trace = 6.206171026935431 (0.0107533% change)
    linmin: converged after 2 iterations.
    iteration    8: trace = 6.205806280560429 (0.00587733% change)
    linmin: converged after 4 iterations.
    iteration    9: trace = 6.205533500746776 (0.00439565% change)
    linmin: converged after 2 iterations.
    iteration   10: trace = 6.20542892228743 (0.00168526% change)
    linmin: converged after 2 iterations.
    iteration   11: trace = 6.205404457703386 (0.000394246% change)
    linmin: converged after 2 iterations.
    iteration   12: trace = 6.20539015819757 (0.000230437% change)
    linmin: converged after 2 iterations.
    iteration   13: trace = 6.205379215254674 (0.000176346% change)
    linmin: converged after 2 iterations.
    iteration   14: trace = 6.205369303786711 (0.000159724% change)
    linmin: converged after 2 iterations.
    iteration   15: trace = 6.205362340329007 (0.000112217% change)
    linmin: converged after 2 iterations.
    iteration   16: trace = 6.20535436123677 (0.000128584% change)
    linmin: converged after 2 iterations.
    iteration   17: trace = 6.205347562043789 (0.00010957% change)
    linmin: converged after 2 iterations.
    iteration   18: trace = 6.205343405078136 (6.69901e-05% change)
    linmin: converged after 2 iterations.
    iteration   19: trace = 6.20534029830957 (5.0066e-05% change)
    linmin: converged after 2 iterations.
    iteration   20: trace = 6.205338899496326 (2.25421e-05% change)
    linmin: converged after 2 iterations.
    iteration   21: trace = 6.205338351100025 (8.83749e-06% change)
Finished solving for bands 2 to 8 after 21 iterations.
tefreqs:, 16, 0, 0, 0, 0, 0, 0.561229, 0.793929, 0.794659, 1.03341, 1.03342, 1.11597, 1.11681
elapsed time for k point: 0.06644082069396973
Band 1 range: 0.0 at Vector3<0.0, 0.0, 0.0> to 0.4912358164155429 at Vector3<-0.3333333333333333, 0.3333333333333333, 0.0>
Band 2 range: 0.47307223980330254 at Vector3<0.0, 0.5, 0.0> to 0.5631059454934055 at Vector3<-0.3333333333333333, 0.3333333333333333, 0.0>
Band 3 range: 0.563150523734041 at Vector3<-0.3333333333333333, 0.3333333333333333, 0.0> to 0.7939289544512708 at Vector3<0.0, 0.0, 0.0>
Band 4 range: 0.7676614802371173 at Vector3<0.0, 0.5, 0.0> to 0.8214230165371811 at Vector3<-0.3333333333333333, 0.3333333333333333, 0.0>
Band 5 range: 0.8651948161670006 at Vector3<-0.3333333333333333, 0.3333333333333333, 0.0> to 1.033413024817744 at Vector3<0.0, 0.0, 0.0>
Band 6 range: 0.8652308007806806 at Vector3<-0.3333333333333333, 0.3333333333333333, 0.0> to 1.0334231251051704 at Vector3<0.0, 0.0, 0.0>
Band 7 range: 1.0213677201238582 at Vector3<0.0, 0.5, 0.0> to 1.1159672190021006 at Vector3<0.0, 0.0, 0.0>
Band 8 range: 1.108662711226582 at Vector3<-0.26666666666666666, 0.26666666666666666, 0.0> to 1.1168110771398998 at Vector3<0.0, 0.0, 0.0>
```

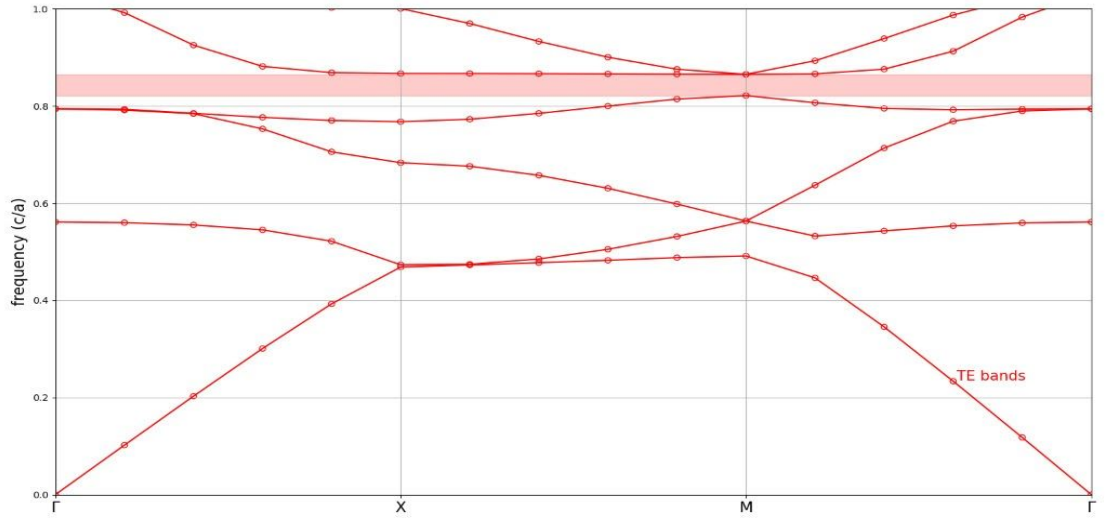
## Plotting the TE and T.M. Bands for Triangular Lattice

The code for plotting the bands and the gaps for the lattices is given in **Appendix A**. The blue shaded region and curves are for TM bands whereas the red shaded region and curves are fot the TE bands.

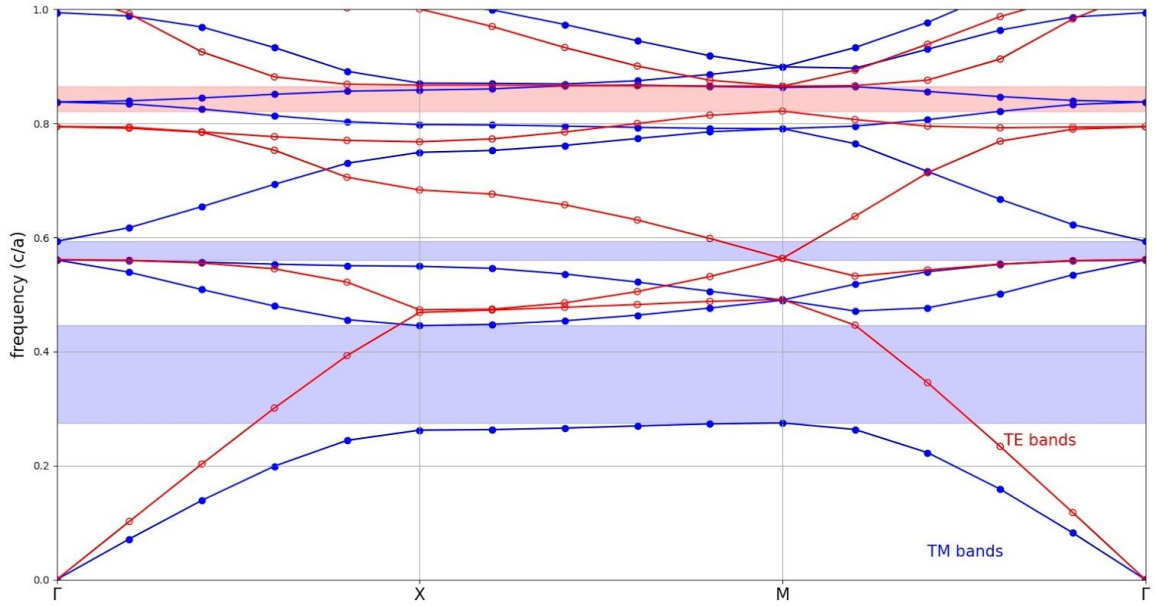
The following plots are obtained -




**This diagram shows the TM bands and curves for the TM modes**




**This diagram shows the TE bands and curves for the TE modes**




**This is the plot comparing both the TE and TM bands and modes**

## Chapter 4

# Meep : Results

## Meep

Meep is an open source software package for electromagnetics simulation through finite-difference time domain (FDTD) method. This technique involves dividing the space into a discrete grid. The fields are then evolved in time using discrete time steps. The grid and the time steps are made finer to achieve a closer approximation for our calculations.

## Units in Meep

The units in Meep are dimensionless. Whatever we compute is expressed in a ratio, so ultimately the units get cancelled. We use scale invariant units to our convenience in electromagnetic problems. In meep units, we take c=1. A (or a/c) is the unit for time. The frequency f is in units of c/a (w in units of 2*pi*c/a). The optical time period is T which is in units of a/c. The frequency f is also specified as a/λ, λ being the wavelength in vacuum.

Especially at infrared frequencies, we tend to specify the distances in microns. We choose 'a' to be 1 micron in length. The Bloch wave vector is specified in the units of 2*pi/a.

## Perfectly Matched Layers

For simulation of the open boundary conditions, we would that all the waves that are incident on the boundaries are absorbed without reflections. We make the use of PML or Perfectly Matched Layers for this. It is a special absorbing material that is positioned adjacent to the boundaries. It is a non-physical material that has zero reflections at the interface. It is required to give some finite thickness to the PML layers so that it can begin its absorption.

## Tasks

**Part 1:** Simulation of a 1 micron thick Silicon slab. Simulating it with a continuous source and observing the field patterns inside the slab. The fields are measured entering from a particular end and coming out of a particular end.

**Part 2:** Simulating a cylindrical rod of 1 micron of height, radius 0.2 micron and a refractive index of 12. The source used is a periodic Gaussian source and we observe the transmission spectra of the electromagnetic field after it traverses through the cylindrical rod.

## Simulating the Silicon Slab with a Continuous Source

The code for simulation and plotting is given in the **Appendix B.**

```
(mp) lenavo@rounak:~/Desktop/BTP/Third_Meeting$ python3 script.py
-----------
Initializing structure...
time for choose_chunkdivision = 0.000453949 s
Working in 2D dimensions.
Computational cell is 16 x 8 x 0 with resolution 10
     block, center = (0,0,0)
          size (1e+20,1,1e+20)
          axes (1,0,0), (0,1,0), (0,0,1)
          dielectric constant epsilon diagonal = (12,12,12)
time for set_epsilon = 0.0308299 s
-----------
run 0 finished at t = 200.0 (4000 timesteps)
XmbTextListToTextProperty result code -2
XmbTextListToTextProperty result code -2
XmbTextListToTextProperty result code -2
XmbTextListToTextProperty result code -2
XmbTextListToTextProperty result code -2
XmbTextListToTextProperty result code -2

Elapsed run time = 13.0655 s
```

**This is the output obtained in the terminal for the simulation**

**This shows the 1 micron thick Silicon slab that is being simulated**

**This shows the field propagation inside the 1 micron thick slab**

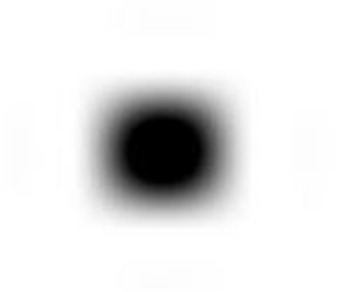
## Simulating a Cylindrical Rod using a Gaussian Source

The code for the simulation is given in **Appendix B**.

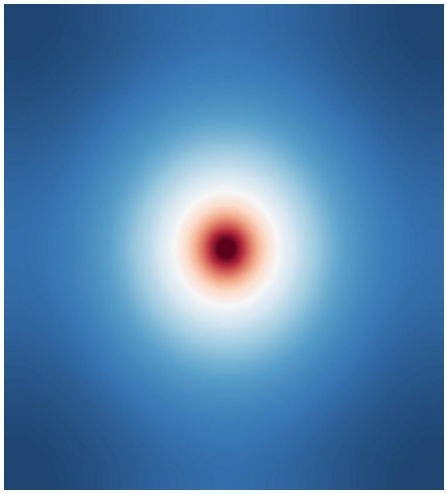
```
(mp) lenavo@rounak:~/Desktop/BTP/Fourth_Meeting$ python3 script.py
-----------
Initializing structure...
time for choose_chunkdivision = 5.10216e-05 s
Working in 2D dimensions.
Computational cell is 4 x 4 x 0 with resolution 10
     cylinder, center = (0,0,0.2)
          radius 0.2, height 1, axis (0, 0, 1)
          dielectric constant epsilon diagonal = (144,144,144)
time for set_epsilon = 0.003057 s
-----------
run 0 finished at t = 200.0 (4000 timesteps)
XmbTextListToTextProperty result code -2
XmbTextListToTextProperty result code -2
XmbTextListToTextProperty result code -2
XmbTextListToTextProperty result code -2
XmbTextListToTextProperty result code -2
XmbTextListToTextProperty result code -2

Elapsed run time = 6.1902 s
```

**This is the terminal output obtained from the simulation of the Cylindrical Rod**

## Plots obtained :-



## The diagram above shows the top view of the structure being simulated.



**The diagram here shows the simulation measured after 200 microseconds using the Gaussian source.**

**Chapter 5**

# Conclusion

The goals achieved were :

- Understanding of the Photonic Crystals, it's theory and formulae
- Understanding the MPB and MEEP softwares and the simulations using it

The results obtained were obtained as expected.

## Chapter 6

# Appendix A

### 1. Code for Square Lattice Band calculation and Simulation

```
import math
import meep as mp
from meep import mpb

# Determining the number of eigenstates computed at each k points.
num_bands=8

# k_points=Bloch wavevectors we want to compute the bands at.
# Setting it to the corners of irreducible Brillouin zone.
k_points=[mp.Vector3(),                 # Gamma
          mp.Vector3(0.5),              # X
          mp.Vector3(0.5,0.5),          # M
          mp.Vector3()]                 # Gamma

# Computing bands at a lot of intermediate points to get a continuous band structure
k_points=mp.interpolate(4,k_points)

# Setting up geometric objects at the center of the lattice
geometry=[mp.Cylinder(0.2,material=mp.Medium(epsilon=12))]

# Setting up the size of the computational cells
geometry_lattice=mp.Lattice(size=mp.Vector3(1,1))

# Setting the resolution
resolution=32

# Creating a ModeSolver object -
ms=mpb.ModeSolver(num_bands=num_bands,
                  k_points=k_points,
                  geometry=geometry,
                  geometry_lattice=geometry_lattice,
                  resolution=resolution)

# Printing and running
print_heading("Square lattice of rods: TE bands")
ms.run_te()
```

```
# This outputs the z field components of the tm mode of the wave.
ms.run_tm(mpb.output_efield_z)

# This outputs the magnetic field z components for the te_modes,
# at the point X, and the energy density power (D power).
ms.run_te(mpb.output_at_kpoint(mp.Vector3(0.5), mpb.output_hfield_z,
mpb.output_dpwr))

# sample points -
tm_freqs = ms.all_freqs
tm_gaps = ms.gap_list
ms.run_te()
te_freqs = ms.all_freqs
te_gaps = ms.gap_list

tm_freqs = ms.all_freqs
tm_gaps = ms.gap_list
ms.run_tm()
te_freqs = ms.all_freqs
te_gaps = ms.gap_list
```

## Code for plotting the obtained data

```
import matplotlib.pyplot as plt


fig, ax = plt.subplots()
x = range(len(tm_freqs))

for xz, tmz, tez in zip(x, tm_freqs, te_freqs):
        ax.scatter([xz]*len(tmz), tmz, color='blue')
        ax.scatter([xz]*len(tez), tez, color='red', facecolors='none')
ax.plot(tm_freqs, color='blue')
ax.plot(te_freqs, color='red')
ax.set_ylim([0, 1])
ax.set_xlim([x[0], x[-1]])

for gap in tm_gaps:
        ax.fill_between(x, gap[1], gap[2], color='blue', alpha=0.2) if gap[0]>1

for gap in te_gaps:
        ax.fill_between(x, gap[1], gap[2], color='red', alpha=0.2) if gap[0]>1

ax.text(12, 0.04, 'TM bands', color='blue', size=15)
```

```
ax.text(13.05, 0.235, 'TE bands', color='red', size=15)

points_in_between = (len(tm_freqs) - 4) / 3
tick_locs = [i*points_in_between+i for i in range(4)]
tick_labs = ['Γ', 'X', 'M', 'Γ']
ax.set_xticks(tick_locs)
ax.set_xticklabels(tick_labs, size=16)
ax.set_ylabel('frequency (c/a)', size=16)
ax.grid(True)

plt.show()
```

## 2. Code for Plotting the Band Diagram and the Band Gaps for TE mode in Square Lattice

```
import math
import meep as mp
from meep import mpb

num_bands = 8
resolution = 32
geometry_lattice = mp.Lattice(size=mp.Vector3(1, 1),
                              basis1=mp.Vector3(math.sqrt(3)/2, 0.5),
                              basis2=mp.Vector3(math.sqrt(3)/2, -0.5))
geometry = [mp.Cylinder(0.2, material=mp.Medium(epsilon=12))]
k_points = [
        mp.Vector3(),           # Gamma
        mp.Vector3(y=0.5),              # M
        mp.Vector3(-1./3, 1./3),        # K
        mp.Vector3(),           # Gamma
]
k_points = mp.interpolate(4, k_points)

ms = mpb.ModeSolver(
        geometry=geometry,
        geometry_lattice=geometry_lattice,
        k_points=k_points,
        resolution=resolution,
        num_bands=num_bands
)
ms.run_tm(mpb.output_at_kpoint(mp.Vector3(-1./3, 1./3), mpb.fix_efield_phase,
```

```
        mpb.output_efield_z))
tm_freqs = ms.all_freqs
tm_gaps = ms.gap_list
ms.run_te()
te_freqs = ms.all_freqs
te_gaps = ms.gap_list
```

## Code for plotting the obtained data from the Simulation

```
import matplotlib.pyplot as plt

fig, ax = plt.subplots()
x = range(len(tm_freqs))
# Plot bands
# Scatter plot for multiple y values, see https://stackoverflow.com/a/34280815/2261298
for xz, tmz, tez in zip(x, tm_freqs, te_freqs):
        #ax.scatter([xz]*len(tmz), tmz, color='blue')
        ax.scatter([xz]*len(tez), tez, color='red', facecolors='none')
#ax.plot(tm_freqs, color='blue')
ax.plot(te_freqs, color='red')
ax.set_ylim([0, 1])
ax.set_xlim([x[0], x[-1]])

# Plot gaps
#for gap in tm_gaps:
#       if gap[0] > 1:
#       ax.fill_between(x, gap[1], gap[2], color='blue', alpha=0.2)

for gap in te_gaps:
        if gap[0] > 1:
        ax.fill_between(x, gap[1], gap[2], color='red', alpha=0.2)



# Plot labels
#ax.text(12, 0.04, 'TM bands', color='blue', size=15)
ax.text(13.05, 0.235, 'TE bands', color='red', size=15)

points_in_between = (len(tm_freqs) - 4) / 3
tick_locs = [i*points_in_between+i for i in range(4)]
tick_labs = ['Γ', 'X', 'M', 'Γ']
```

```
ax.set_xticks(tick_locs)
ax.set_xticklabels(tick_labs, size=16)
ax.set_ylabel('frequency (c/a)', size=16)
ax.grid(True)

plt.show()
```

## Chapter 7

# Appendix B

1. **Code for Simulation and Plotting of the fields propagating through a 1 micron thick Silicon Slab using a Continuous Source**

```
import meep as mp

cell = mp.Vector3(16,8,0)

geometry = [mp.Block(mp.Vector3(mp.inf,1,mp.inf),
                     center=mp.Vector3(),
                     material=mp.Medium(epsilon=12))]

sources = [mp.Source(mp.ContinuousSource(frequency=0.15),
                     component=mp.Ez,
                     center=mp.Vector3(-7,0))]

pml_layers = [mp.PML(1.0)]

resolution = 10

sim = mp.Simulation(cell_size=cell,
                     boundary_layers=pml_layers,
                     geometry=geometry,
                     sources=sources,
                     resolution=resolution)

sim.run(until=200)
```

### Code for Plotting the obtained data using Matplotlib and spline36

```
import numpy as np
import matplotlib.pyplot as plt

eps_data = sim.get_array(center=mp.Vector3(), size=cell, component=mp.Dielectric)
plt.figure()
plt.imshow(eps_data.transpose(), interpolation='spline36', cmap='binary')
plt.axis('off')
plt.show()

ez_data = sim.get_array(center=mp.Vector3(), size=cell, component=mp.Ez)
plt.figure()
plt.imshow(eps_data.transpose(), interpolation='spline36', cmap='binary')
plt.imshow(ez_data.transpose(), interpolation='spline36', cmap='RdBu', alpha=0.9)
plt.axis('off')
plt.show()
```

## 2. Code for Simulation and Plotting the transmission spectra of a Silicon Cylindrical Rod using a Periodic Gaussian Source

```
from __future__ import division

import meep as mp

cell=mp.Vector3(4,4,0)

geometry_cylinder=[mp.Cylinder(center=mp.Vector3(0,0,0.2),height=1,radius=0.2,axis=
mp.Vector3(0,0,1),material=mp.Medium(index=12))]
geometry=geometry_cylinder

k_point=mp.Vector3(0,0,0)
df=0.1
fcen=0.15
w=1
sources = [mp.Source(mp.GaussianSource(fcen,fwidth=df),
                     component=mp.Ez,
                     center=mp.Vector3(0,0,3),
                     size=mp.Vector3(0,w,0))]
```

```
pml_layers=[mp.PML(1.0)]
resolution=10

sim=mp.Simulation(cell_size=cell,geometry=geometry,k_point=k_point,sources=sources
,resolution=resolution)

sim.run(until=200)
```

### Code for Plotting the obtained data from the Simulation using Matplotlib and spline36

```
import numpy as np
import matplotlib.pyplot as plt

eps_data = sim.get_array(center=mp.Vector3(), size=cell, component=mp.Dielectric)
plt.figure()
plt.imshow(eps_data.transpose(), interpolation='spline36', cmap='binary')
plt.axis('off')
plt.show()

ez_data = sim.get_array(center=mp.Vector3(), size=cell, component=mp.Ez)
plt.figure()
plt.imshow(eps_data.transpose(), interpolation='spline36', cmap='binary')
plt.imshow(ez_data.transpose(), interpolation='spline36', cmap='RdBu', alpha=0.9)
plt.axis('off')
plt.show()
```

# References


- https://meep.readthedocs.io/en/latest/Python_Tutorials/Basics/
- https://mpb.readthedocs.io/en/latest/
- *Photonic Crystal : Molding the Flow of Light*, Second edition, John D. Joannopolous et.al.