UNIVERSITAT POLITÈCNICA DE CATALUNYA

Grup de Recerca en Micro i Nanotecnologies

Simulation and Analysis of Photonic Structures in Macroporous Silicon

Master Thesis

Grupo Micro y Nano Tecnologías

UPC

July 2009

Author: Didac Vega Bru

Advisor: Dr. Angel Rodríguez Martínez

Abstract

This Master Thesis is dedicated to photonic crystal, in particular to macroporous silicon. We introduce the concepts of crystallography, and symmetry, and important associated tools to its study, the *Reciprocal lattice*, and the *Brillouin Zone*. We also give a brief explanation of anodic macropore formation in silicon; the method used in-site to produce photonic crystals. It follows a section explaining Fourier Transform spectrometry, and how it was used in the present work. Following it, we introduce our main software tools, which we have learned and used extensively throughout the development of this Thesis. We go into certain detail, to point some important aspects, or simply to provide a quick use guide. We finally present the results of our measures and simulations, the comparisons made and the conclusion that we can extract from our experiments.

Table of Contents

FIGURE INDEX

Table Index

I – INTRODUCTION

Photonic crystals are a recent discovery. It was in 1991 when Yablonovitch theorised and produced the first "photonic" crystal working in the microwave frequencies[4]. Photonic crystals, or *photonic band-gap materials*, are structured media, with periodic variations of dielectric constant that show characteristic propagation modes. With the appropriate geometry, it is possible to separate the modes in a way that they open frequency ranges where no propagation is possible, therefore creating a "forbidden band" in analogy with solid-state electronics.

Since then, heavy research has been carried out on this subject. A variety of materials, composites and structures have been proposed, but few of them are still commercially viable. The discovery of macroporous silicon etching by anodic oxidation by Lehman[26] and its application to photonic crystals[5] has meant a great advance in photonic crystal development, and has had a large impact in the industrial prospects and applications.

With anodic etching in HF of silicon, we can now produce quickly and economically photonic crystals of very high aspect ratio. Another widespread technique is RIE, but the resulting pores are rougher, their maximum aspect ratio is smaller, and they cannot by modulated in depth. In the MNT group at the UPC, we use the HF etching technique to produce high quality macroporous silicon photonic crystal samples.

The main line of research at the group is the design and fabrication of macroporous silicon based structures and the characterization of their optical response and thermal effects, especially focused to applications such as integrated optical devices or biosensensors. Nevertheless, a new impulse is being given to the research of new applications for this material, for instance, fuel cells or catalysers, and the recently created centre for nanotechnologies will further boost this initiative.

There are quite a number of commercial applications, like optical filters, switches or optical waveguides, and it is expected that demand for them will increase with technology maturation. Laboratory research has produced optical devices embeddable in standard process ICs, a major milestone will be set when this is achieved at the integrator scale, directly from the fab.

The proposal motivated me. Microelectronics has always fascinated me, and specially lasers and other optical communications devices. Optical computing has been buzzing for several years, but development seems to be pretty slow, or stalled. The practical applications of photonic crystals are tremendous, not necessarily because of their optical properties, but because of being structured materials we can engineer their desired characteristics, for instance the thermal response, to a particular application or problem.

II – Objectives

The work presented in this Master Thesis is dedicated to the study of photonic crystals, in particular those made of macroporous silicon. We are highly interested in the optical properties of these new materials. Fully characterizing the optical features of macroporous silicon will lead to improved optical device design.

Our main interest area is nano-photonics, therefore, we must first learn what photonic crystals are, how they are constructed, their properties, and the applications that are envisaged for them. For doing this we must understand the physical properties in which they are based, which are related to solid-state theory and crystallography.

To gain an in depth understanding of macroporous silicon used as photonic crystals, we must know the steps that are involved in fabrication, and how they affect the outcome. Thus, we study the macroporous silicon etching process used by the group, which is simple, fast and economic.

The objectives of this work are the optical characterization in the Near Infrared range with an FT-IR spectrometer equipment recently arrived, and theoretical simulation of the measured samples. With the gathered information, we proceed to verify that theory and real measures agree, and further check that the fabrication process is reliable and repeatable.

Photonic crystal characterization requires that we learn how to use the FT-IR, and that we extend our knowledge on Fourier Transform Spectrometry. We aim to acquire and enhance our skills with this measurement equipment to produce the best spectral measures possible. The FT-IR spectrometer allows us to take very high-resolution spectrum measures, which is of maximum importance to capture the fine details of photonic crystal optical response. This equipment also includes a reflection unit, which allows us to perform high accuracy angle-resolved reflection measures. Furthermore, the reflection unit is completed with a polarization filter that allows us to separate the incident light polarization. The outcome of the performed reflection measures will be the dispersion map of our photonic crystals with the maximum detail and information possible.

Simulations will be executed in parallel to FT-IR measures. To perform them, we will learn how to use the simulation software tools at our laboratory: TMM, MPB and Meep. The goal of using this software is getting familiar with it and being able to solve the typical usage cases of photonic crystals. With these tools, we will obtain the theoretical reflection, band-gaps and electromagnetic wave propagation for our experiments. We will customize these tools as necessary to get the best results.

Our last objective will be comparing the theoretical simulated results with the actual spectral measures taken with the FT-IR, and seek for agreement in both results, proving the validity of our simulations by checking of the used parameters with the real dimensions and characteristics of the samples.

III – PHOTONIC CRYSTALS

Photonic Crystals are a relatively recent development. First work on them was done by Yablonovitch in the late 1980. The main characteristic of photonic crystals is the existence of *light bandgaps*, the equivalent to the potential bandgap on semiconductor devices, which forbid the propagation of light (EM wave) through the device.

Photonics Crystals are *structured materials*. The overall material macroscopic quantities ε and μ, can be tuned to that of interest. It is even possible to create a crystal that exhibits negative refraction index or both parameters negative (these are known as left handed materials); these crystals are referred as *metamaterials*, as they macroscopically behave in ways that are not natural.

The photonic crystal research has opened a large field for applications and devices. For example, Pendry proposed the use of metamaterials to create *perfect lenses*. We can also cite the recent achievement of using them as effective cloaks in the IR and microwave region. The interest areas of photonic crystals reach from the microelectronic industry, to communications, computing or the military.

As commented, photonic crystals are in general an ordered arrangement of dielectrics varying permittivity. Depending on the organization of the material, the dielectric function may vary along 1 (layers), 2 (plane) or 3 axes. In this case, we say that the photonic crystal is 1D, 2D or 3D, respectively.

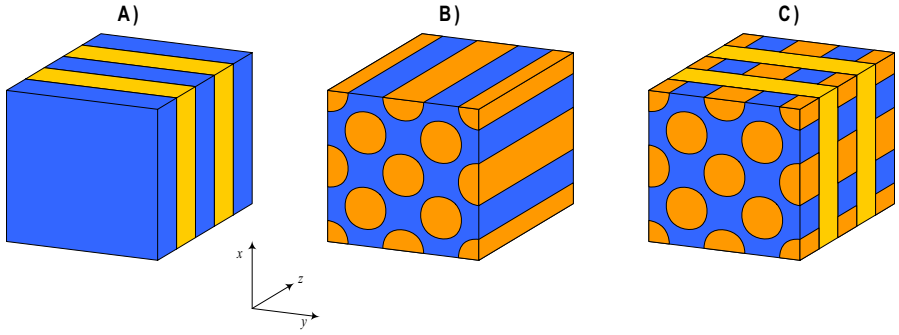


Fig. 1 – PC designation depending on the dielectric function. A) 1D photonic crystal, defines layers of different refraction index. B) 2D photonic crystal, defines a dielectric function on a plane, while maintaining constant along *z*-axis. C) 3D photonic crystal, the refraction index varies in the three space directions.

Photonic Crystals borrow the *crystal* term from crystalline solids where atoms are ordered in regular patterns. Until recently, it was not possible to fabricate crystals usable at optical frequencies, as the feature size of the structure must be of the order of the working wavelength. Precision and regularity are paramount to obtain useful PCs, which exhibit the desired band gaps. Imperfections during fabrication quickly create modes and destroy the PBG. PCs are made in a variety of materials but for our purpose and in this project, the used material is Silicon.

3.1 – Crystallography

Crystallography is a very wide science of great use in physics and chemistry. In a broad sense, its main goal is determining the arrangement of features in a material, and effects to its observable properties, for instance, the atoms in a metal or the organization of LCD molecules.

Crystallography was developed from mineralogy with the study of crystals formed by nature and was restricted to the geometry of the object. Later with the discovery of X-rays, it was extended to the arrangement of the crystals atoms. Further theory and technical advances have allowed crystallography to be used from solid state to biology, among others.

Crystallography provides tools, techniques and methods that are essential in the study of the behaviour of photonic crystals. Whit them, it is possible to determine how electromagnetic waves propagate inside the crystal, and define rules to design and shape crystals to get the sought response.

3.1.1 – Symmetries, the Bravais Lattice and Point Group

Most crystals present a regular structure and crystallography is founded around the characteristics that this regularity provides. The study of periodic or regular structures is heavily based in symmetry transformations and group theory.

In the present study of photonic crystals, we deal with periodic structures. To simplify the problem, all concepts and theory involved, are related to infinite extent crystals. It is obvious that no physically realizable object can have infinite extent, but considering the working wavelength and the feature size of the photonic crystal, we can simplify to such case. This implies that we have a constrained range of useful frequencies, were the device presents the desired response.

We focus now on periodic 3D structures, which are the base for all crystals. If necessary, dimensionality can be reduced, but the main definitions and conclusions remain.

Periodicity is a stronger constraint than regularity. The former demands that feature repetition occurs at fixed distances and angles, while the latter only asks for a common rule, feature or relationship between features; thus, for instance, a periodic layered material presents layers of alternating refraction index and equal width, while a regular layered material could have layers of alternating index but doubling width.

These constraints allow the appearance of symmetries that can be exploited in the analysis of the crystal's behaviour. In space, we identify the following basic geometric symmetry operations: *identity*, *translation*, *rotation*, *reflection*, *glide reflection* and *screw rotation*. The identity operation leaves everything without change and is effectively a "non-operation". The next three are intuitive and well-known transforms. Finally, the glide reflection can be understood as a translation and reflection combined in one operation, and similarly, the screw rotation as a translation and rotation.

We can now explain how these transforms apply to a crystal. In a periodic crystal, we observe that there must be a fixed **discrete translational symmetry** along each space dimension to fulfil the periodicity requirement. Given an arbitrary starting point,

this operation fills the entire space without gaps or overlaps and relates pair of points in the crystal space $\mathbf{g}, \mathbf{r} \in G$ by

$$\mathbf{g} = \mathbf{r} + \sum k_i \mathbf{b}_i \,, \tag{1}$$

where $\{\mathbf{b}\}_i$ is a set of basis vectors and k_i are integer numbers. These points define the *direct lattice* of the crystal. The possible translations are constrained by other symmetries in the crystal, thus there exist a limited number of possible configurations of different basis vectors. These attainable geometries are known as *Bravais Lattice*s, of which there exist 14 different.

The translation symmetry generates copies of a given pattern in all space. As the direct lattice generates equally spaced planes, we define the enclosed volume as the *cell*, which are placed at each lattice point. It is not very important how is the cell placed in respect to the lattice, but for simplicity, the lattice origin is commonly placed either at a vertex or at the centre of the cell. The cell shape may be intricate, provided that holds the translational symmetry; but for simplicity, it is preferred to follow the polygon defined by the lattice, or a basic geometric object.

The cell contains the common geometric features of the crystal, which is repeated by the translation symmetry. Those features may present additional symmetries and form a *Symmetry Point Group*. If the cell does not have any translation symmetry, we can define it as a *primitive cell*, that is, the minimum representation of the crystal. Otherwise, it could be possible to split the cell, and redefine the lattice to include the cell translations. The optimal situation is reducing the cell to its minimum.

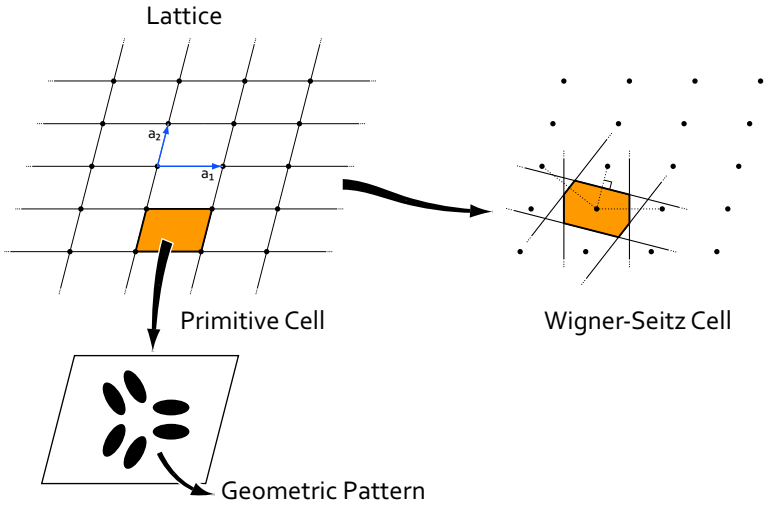


Fig. 2 – Some crystallography basic elements. In the image is depicted a periodic 2D lattice, from which we can define the Wigner-Seitz cell and the primitive cell. In the primitive cell, an example of point group symmetry is shown, with a horizontal reflection plane and three-fold rotation.

In this situation, a primitive cell forms a point group. It is characterized by having only identity, rotation and reflection symmetries with a common axe or centre of symmetry. In the cell this point corresponds to the geometrical centre. Given a central point, the symmetries are determined with respect the other cells. These operations must be compatible with the lattice, thus only 32 combinations are possible. There exist

several notation systems for point group symmetry, the common one used in spectrography being *Schönflies*.

In total, there are 230 feasible arrangements combining Point Group Symmetries and Bravais Lattices, which have been extensively studied and documented elsewhere[3,12,15].

In addition to the primitive cell, in a Bravais lattice we can also define the so-called *Wigner-Seitz cell*. This cell is defined as the volume of points nearest to a lattice point than to any other. It can be easily constructed by building the perpendicular bisectors between the central point and its neighbours, and then choosing the minimum space enclosed by those. Although it does not have much use in the direct lattice, in the reciprocal lattice is of great importance.

3.1.2 – The Reciprocal Lattice and the Brillouin Zone

The periodic property of the lattice allows to consider it as a set of points in space with fixed spacing. This easily lends to analysis by Fourier Series. Taking all cells in the lattice as single points, we find the *spatial spectrum* by applying a Fourier Transform. The obtained result is a new grid of points termed the *Reciprocal Lattice*.

This new lattice has also a set of basis vectors that must satisfy the relation,

$$e^{j2\pi \mathbf{K}\cdot\mathbf{R}} = 1, \tag{2}$$

where **R** are all the direct lattice point positions and **K** the positions of the points in the reciprocal lattice. From this expression, for a 3D space the basis vectors can be determined by

$$\begin{aligned}
\mathbf{b}_1 &= \frac{\mathbf{a}_2 \times \mathbf{a}_3}{\mathbf{a}_1 \cdot (\mathbf{a}_2 \times \mathbf{a}_3)} \\
\mathbf{b}_2 &= \frac{\mathbf{a}_3 \times \mathbf{a}_1}{\mathbf{a}_1 \cdot (\mathbf{a}_2 \times \mathbf{a}_3)} \\
\mathbf{b}_3 &= \frac{\mathbf{a}_1 \times \mathbf{a}_2}{\mathbf{a}_1 \cdot (\mathbf{a}_2 \times \mathbf{a}_3)},
\end{aligned} \tag{3}$$

that can be expressed in matrix notation as

$$[\mathbf{b}_1 \quad \mathbf{b}_2 \quad \mathbf{b}_3]^T = [\mathbf{a}_1 \quad \mathbf{a}_2 \quad \mathbf{a}_3]^{-1}, \tag{4}$$

where $\{\mathbf{a}\}_i$ represents the basis vectors of the direct lattice and $\{\mathbf{b}\}_i$ the basis vectors of the reciprocal lattice, and all vectors are column vectors. Matrix notation has also the additional benefits of compactness and easy generalization to other dimensions.

The reciprocal lattice is a purely mathematical construct and has no physical meaning on the crystal. In the reciprocal space, geometry is unaffected and so does all point group symmetries of the cell.

In the same way as we can build the Wigner-Seitz cell in the direct lattice, in the reciprocal lattice we build the *Brillouin Zone*. This volume encloses the set of points that fully describe the spatial spectrum of the lattice. This is due to the sampling theorem of the Fourier Transform. This theorem creates spectrum *foldings* at the

midpoints between the origin and the rest of the reciprocal space; which implies that these folding lay at the boundary of the Brillouin Zone. This folding gives us the following result[1],

$$\mathbf{v} \in G \Rightarrow -\mathbf{v} \in G\,, \tag{5}$$

where G is the group of points that belong to the Brillouin Zone. By this relation, we can exclude half of the Brillouin Zone in our computations[1].

Taking into account this periodicity of the spatial spectrum, the electromagnetic fields inside the crystal can be found to be standing waves. Thanks to this, it is possible to fully determine the fields' distribution by calculating them in the minimal set of points that the Brillouin Zone contains. We simply need to express the propagation vector of the fields in the reciprocal lattice basis, and constrain them to be inside the Brillouin Zone.

Even though the Brillouin Zone represents the minimum set of points for the translational symmetry, we must recall that the point group may present additional symmetries. As stated above, the reciprocal lattice is a simple mathematical artefact, thus the point group symmetries are not altered and allow for further reduction of the minimal point set. The result of all simplifications is the *Irreducible Brillouin Zone*, which holds the minimum points to uniquely determine the fields inside the crystal.

We can identify high symmetry points in the crystal. These points are invariant under several transforms and present degeneracy and the highest energy of the group. The degeneracy issue is due to the fact that several field modes collapse to a single value, as transforms yield the same or equivalent point. Therefore, one technique to open a (photonic) band gap is lifting the degeneracy by breaking symmetry.

Considering the previous points, it suffices for the study of the crystal behaviour to calculate the fields along the edges of the irreducible Brillouin zone. The most common geometries give special names to these high-symmetry points. In the following section we present some of the studied cases.

3.1.3 – 2D Crystals

In this project we concentrate in planar structures, primarily straight cylinders etched on bulk. In some cases, full 3D designs have been studied, but only limited to their reflection that simplify to 2D cases.

The typical two-dimensional structures used, have been square and triangular lattices of cylindrical rods. In fact the etching process is not perfect and the actual samples measured present a certain degree of asymmetry that possibly may be need to be considered.

In Fig. 3 are depicted these lattices with their symmetries and Brillouin Zones. The typical basis vectors along with the reciprocal are used. When determining the Piont Group Symmetries, is important to note that all rods maintain orientation, thus for the near-square case, all cylinders are faced identically. We also appreciate that as we work

1 This relationship does not hold for cells without symmetry or inhomogeneous or anisotropic materials.

from circular to square cross-sections, in the square lattice the point group symmetry is kept, but for the triangular lattice is not.

Similarly, different pore shape or lattice need to be specifically studied to determine the present symmetries and thus the Irreducible Brillouin Zone.

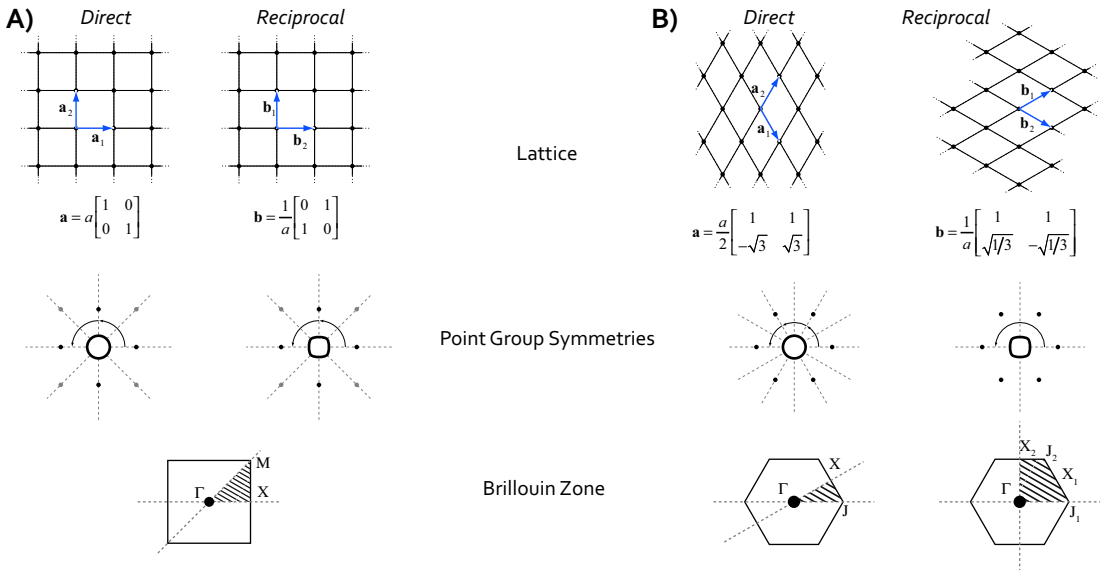


Fig. 3 – Square and Triangular Lattices. A) shows the square lattice while B) illustrates the triangular one. In all cases both the direct lattice and the resulting reciprocal lattice are shown for the given direct lattice basis. Point Group Symmetries are found for each lattice in two cases, for a circular cross-section cylinder, and for a "square circle" cross-section cylinder. Rotations are indicated by the curved arrows and reflection planes by dashed lines. The Brillouin Zone is found directly form the reciprocal lattice, while the Irreducible Brillouin Zone, shown shaded, is found applying the symmetries.

In the present work, despite the samples used present "square circle" pore cross-section, we always use the irreducible Brillouin zone of maximum symmetry.

Table I – Basic square and triangular lattices parameters. In this table the typical basis are given for each lattice configuration; from these, the reciprocal lattice and irreducible Brillouin zone are derived.

	Square	Triangular
Direct Lattice	$a\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$	$\frac{a}{2}\begin{bmatrix}1 & 1\\ -\sqrt{3} & \sqrt{3}\end{bmatrix}$
Reciprocal Lattice	$\frac{1}{a}\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$	$\frac{1}{a}\begin{bmatrix}1 & 1\\ -\sqrt{1/3} & \sqrt{1/3}\end{bmatrix}$
Irreducible Brillouin Zone	$\mathbf{\Gamma} = (0,0)$ $\mathbf{X} = \left(\frac{1}{2},0\right)$ $\mathbf{M} = \left(\frac{1}{2},\frac{1}{2}\right)$	$\mathbf{\Gamma} = (0,0)$ $\mathbf{J} \equiv \mathbf{J}_1 = \left(\frac{1}{3},\frac{1}{3}\right)$ $\mathbf{X} \equiv \mathbf{X}_1 = \left(\frac{1}{2},0\right)$ $\mathbf{J}_2 = \left(\frac{2}{3},-\frac{1}{3}\right),\ \mathbf{X}_2 = \left(\frac{1}{2},-\frac{1}{2}\right)$

3.2 – Macroporous Silicon

Porous silicon has been known for some years now. Porous silicon is basically a Si substrate were some material has been removed forming *pores* (cavities) in the bulk. According to the pore diameter, the resulting material is further classified as *microporoous Si*, *mesoporous Si* and *macroporous Si* (MP-Si), for less than 2 nm, less than 50 nm and greater than 50 nm, respectively.

Microporous and mesoporous Si growth is mostly an uncontrolled process. We are able to control mean porosity and in a lesser degree depth, but pore placement and diameter is random. Though such material does exhibit interesting optical behaviour, such as electroluminescence, our interest is centred in obtaining controlled pore growth to fit the optical application.

MP-Si is one of the best alternatives available for the fabrication of photonic crystals in the IR and visible spectrum. The features obtained are in the order of the wavelength and we can control precisely enough the growth parameters and progress to obtain practical and cheap devices.

One of the main advantages of MP-Si is its high availability and the extended silicon semiconductor industry and techniques already developed. Silicon wafers of different doping are readily accessible with cheap fabrication processes and the possibility to directly integrate the optical path with the CMOS digital part.

MP-Si is suitable to build 2D and 3D photonic crystals. 2D PCs are simply designed by patterning the Si surface where the pores should grow. 3D PCs can be built by modulating the pore profile, thus varying the radius of the pore along the pore axis. Other possibilities exist, such as filling the pores with other materials using the bulk Si as a mould or creating a membrane.

The principal patterns used in macroporous silicon are the rectangular and triangular arrays. The typical pore geometry varies between circular and rectangular shapes, the common case being something in between as the extreme cases are difficult to create depending on the etching process.

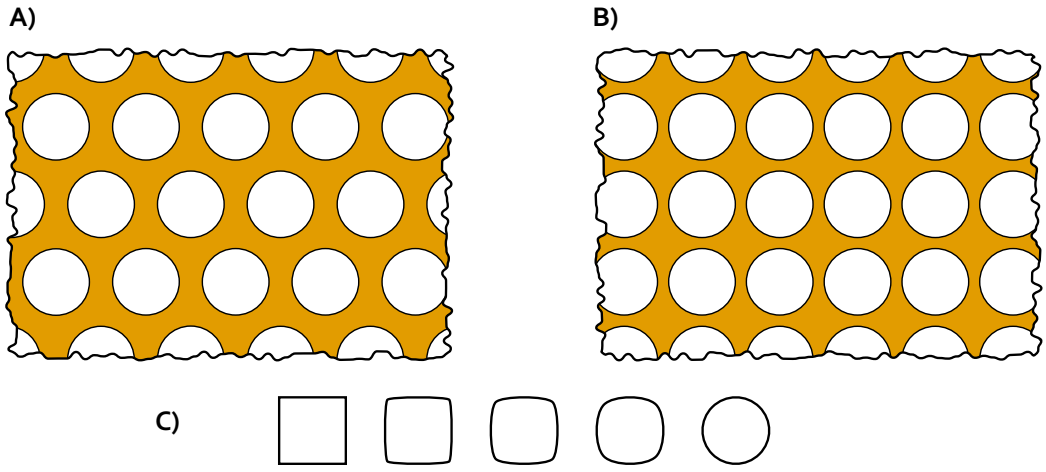


Fig. 4 – Typical MP-Si lattices: A) Triangular pattern. B) Rectangular array. C) Different pore shapes attainable during MP-Si fabrication.

After etching, the photonic crystal obtained can be post processed altering the pore shape and profile. For example, oxidation and oxide removal steps can be used to widen the pores and give them a rounder shape.

First MP-Si development for bulk processing was conducted by Lehman et al. They developed the wet etching technique and found empirical formulae to control with high degree of precision the pore shape and growth.

Main uses of MP-Si lay in the optical area such as optical filters, integrated waveguides, couplers or cavity resonators. Additionally, they have been proposed for the microfluidics and biological field.

This project focuses on the study of the optical behaviour of MP-Si at mid-IR range.

3.2.1 – Fabrication of MP-Si

There are many techniques at our disposal to fabricate macroporous silicon. Some exotic processes allow for structures and geometries that cannot be achieved by more conventional methods. Nevertheless, we are specifically interested in those methods that allow bulk processing and integration with other processes such as CMOS or MEMS devices.

There are two outstanding techniques for our purpose, one is the RIE dry etching and the other is the PECE wet etching process. RIE etching is suitable for moderate aspect ratios and has the advantage of being capable of creating fine and accurate shapes.

In our case we use the *Light Assisted Electrochemical Etching* (PECE for Photon Electrochemical Etching). This is an electrolytic process that uses anodic oxidation of silicon in hydrofluoric acid to create the pores. Being a wet technique, it is less suitable for direct integration but permits the creation of very high aspect ratios for pores (we can etch almost through all the silicon bulk depth) and is a relatively quick method.

The etching model was given by Lehmann, where the controlling parameters are the wafer doping, the acid concentration, the bias voltage, the light intensity and the working temperature. The Si forms a *Shottky contact* with the acid solution where the electrolyte acts as the metal interface. Current flow is important, for the *cathodic* regime, where the Si cathode is negative polarity; it acts as a diode in forward bias creating H_2 molecules at the interface. On the other hand, for *anodic* regime, the contact is in reverse bias, but the electrical behaviour can be altered by illumination.

The etching process is explained as follows; in anodic regime, free holes at the interface promote the divalent reaction,

$$\mathrm{Si}+4\mathrm{HF}_2^- + \mathrm{h}^+ \rightarrow \mathrm{SiF}_6^{2-} + 2\mathrm{HF} + \mathrm{H}_2 + \mathrm{e}^-, \tag{6}$$

which dissolves one atom of Si generating porous silicon. Reaction is limited by *hole transport* as all holes at the interface are consumed. Bias voltage has also a very important role; as voltage is increased, it is found that anodic current forms a peak and then continues growing again; an excessive voltage will provoke breakdown and oscillations in the current. The peak defines a critical current density j_{PS} below which

the previous reaction of **Eq (6)** is the main activity. Above this current, a tetravalent reaction is promoted and causes *electropolishing*, a uniform etching of the Si surface. In this case the reaction is limited by *mass transport* as there is an excess of holes at the contact surface that cannot be consumed by the reaction as quickly as generated, they diffuse through all the surface evenly.

Critical current density in steady state is found to be *Ahrrenius* type, depends only on acid concentration and solution temperature by the empirical relation:

$$j_{\mathrm{PS}} = Cc^{3/2} \exp\left(\frac{-E_{\mathrm{a}}}{kT}\right) \quad [\mathrm{mA} \cdot \mathrm{cm}^{-2}], \tag{7}$$

where C is determined to be 3,300 mA·cm^{-2}·%wt.$^{-3/2}$, c is the HF concentration in %wt, E_{a} = 343 meV is the activation energy, k is the Boltzman's constant and T is the absolute temperature.

As the current depends on hole quantity at the interface, we either need a p-type substrate or to generate the holes. For the p-Si wafer we can't control the hole density during the etching. In contrast, for an n-type substrate, we can use an IR source to photon generate holes. Illumination can be done from the both top (patterned) or bottom faces; in the first case HF doesn't pose a problem since it is not photoreactive, but requires a transparent reaction cell and a focusing system. Front illumination generates holes in the interface; this opens the possibility of using an image pattern to further control the pore features. In contrast, back illumination generates holes on the backside; this requires that the diffusion length of holes to be of the order of the wafer thickness. The cell construction in this case is simpler as the backside of the wafer can be left exposed to air. Unfortunately, the IR intensity to current relation must be determined empirically by measuring, at a fixed bias, the output current; once known, this relation stays almost constant during the etch process.

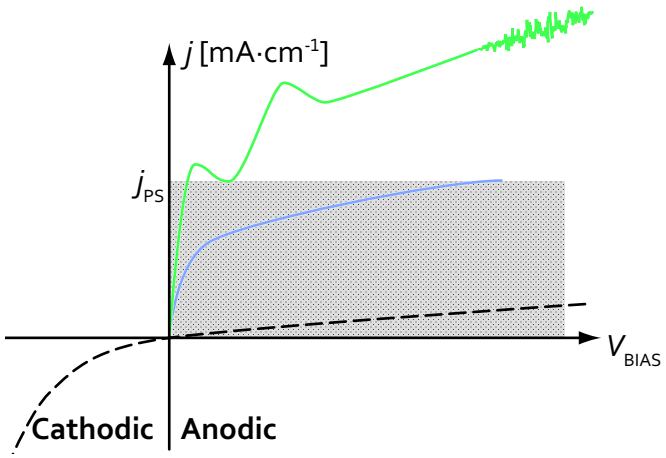


Fig. 5 – Steady state current characteristic of the etch process, current density is proportional to contact surface. Dashed line) dark current. Blue line) low illumination condition. Green trace) Strong illumination condition, showing also instability at high bias voltage. Critical current j_{PS} is set at the first peak of the current waveform. The shadowed area shows the region of PS formation.

Besides the light controlled current, in the absence of illumination, we can observe a small current in anodic regime termed *dark current*. This current corresponds to the inverse biasing of the Shottky contact and is caused by thermal hole generation. Thermally generated holes diffuse along the entire surface not only on the pore tips, causing a shape deviation from the expected. To minimise this effect this requires using the lowest admissible temperature. We must note that dark current depends on the contact area, thus for increased area, the current rises. This is observable before and after the etching, as the area is much greater when the pores are formed, we appreciate an important increase in dark current.

Although at the tips the reaction is in steady state, pore formation is a time dependent process. As pores grow, material is removed from the bulk, altering the etching conditions. Equally important is the HF concentration variation; the resulting by-products of the reaction in **Eq (6)** remain in the solution and if pores are very long, it may happen that the HF_2^- ions get depleted inside the pore, becoming a mass transport limited situation. As a result, we have that pore width and growth speed varies along the pore axis. To counter these effects we must modify illumination or bias voltage in a suitable way; unfortunately, this has to be determined empirically.

Macropores are started by a *nucleation phase*, the top surface experiences some electropolishing, in the course some locations are more etched than others. This generates a *Space Charge Region* (SCR) than follows the surface geometry, therefore those deeper locations extend the SCR towards the back face promoting hole attraction to their tips. It is at these points that pores will develop after the nucleation phase. As we are interested in placing the pores at specified locations, we need to *mark* where we want them; this is why the *pre-patterning* of the wafer surface is necessary.

Once we are in the *pore grow regime*, it is found that the reaction occurs at the tip in steady state. It was found that to obtain well formed pores, exists a relationship involving the pore geometry, PC lattice, doping density and the bias voltage. The SCR loosely follows the surface, therefore if pores are set too apart the SCR will slip between them and attract holes to the pore walls. If this happens, there appear uncontrolled defects such as *branching* or *pore death* as shown in **Fig. 6**. It was found that the wall thickness should be near the SCR width, which is given by:

$$x_{\mathrm{SCR}} = \sqrt{2\frac{\varepsilon_0 \varepsilon_{\mathrm{Si}} V_{\mathrm{eff}}}{qN_{\mathrm{D}}}}\,, \tag{8}$$

$$V_{\mathrm{eff}} = V_{\mathrm{bi}} - V - kT\,, \tag{9}$$

where $\varepsilon_{\mathrm{Si}}$ is silicon relative permittivity, N_{D} is the doping density, q is the charge, V_{eff} is the effective contact potential, where V_{bi} is the built-in potential, about 0.5 V. In **Fig. 6** we can appreciate the SCR and the hole current flow when working in steady state below j_{PS}. Diffusion holes from back illumination will preferentially reach the pore tip, where the dissolution reaction takes place. Some holes may diffuse inside the SCR but the majority is thermal generated. These holes cause the pore to have a conical shape and generate a thin layer of microporous Si on the pore wall. The fluoride ions must travel the pore to reach the tip, as the pore grows the flow rate decreases as commented before.

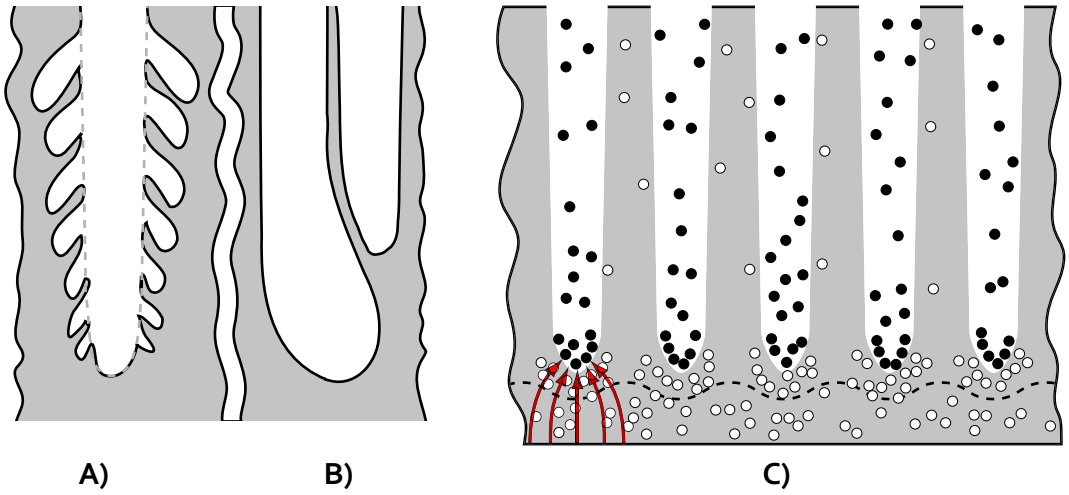


Fig. 6 – Macropore etching. A) Branching defect. B) Pore death and deformation defect. C) Schematic representation of the etching process in steady state. Black dots represent electrons, while white dots are holes. The SCR is shown as the dashed line and the preferred current for a single pore in red.

It must be mentioned that, in general, p-Si is not suitable for macropore formation. As holes are already present throughout the wafer, even when pre-patterning the surface, pores develop defects, specially branching. Nevertheless, it is well suited for micro- and meso-pore formation.

Taking into account the SCR formation, we now have to impose a constraint on the lattice geometry. In steady state regime, Lehmann's model states that the porosity of the PC is directly proportional to the current density:

$$p = \frac{j}{j_{\mathrm{PS}}} = \frac{A_{\mathrm{pores}}}{A_{\mathrm{sample}}} = \frac{A_{\mathrm{pore}}}{A_{\mathrm{unit\ cell}}}. \tag{10}$$

Porosity is defined as the ratio of the total cross section of the pores over the total sample area. For a regular lattice, the rightmost term of **Eq (10)** holds true: the unit cell is repeated over the whole surface, therefore, total features areas are a multiple of the unit cell features areas. This relationship gives us one mechanism to control the pore width during the etching process, and creating 3D PCs. As current density depends on the bias voltage by SCR and the illumination, those are our effective parameters to achieve pore modulation.

To obtain a controllable pore growth we use low doped n-type silicon with resistivity between 4 and 5 $\Omega{\cdot}\mathrm{cm}^{-2}$. Crystal orientation is ⟨100⟩ to obtain vertical, straight pores, as HF etch is anisotropic, it prefers those directions, thus for other directions, for instance ⟨111⟩, we would obtain pores with three branches at 54.7º from the surface. The HF electrolyte is a low concentration (2.5 %wt.) solution of HF, water and ethanol.

To generate an ordered array, the process starts with the *pre-patterning* of the wafer surface. This is done by conventional lithography and a shallow etch. This first step is needed to mark the locations where pores will grow. The tip of the pyramid creates a potential curvature, attracting holes and promoting the etching locally. The process is

shown in **Fig. 7**: we start with a low resistance n-type 4" wafer to which we perform a n^+ ion implant on the backside; the next step is performing an oxidation and annealing to obtain a 100nm oxide layer. The top surface is patterned and etched by RIE. The final preparation step is anisotropically etching the wafer in KOH or TMAH to produce the dips in the Si surface.

The wafer once prepared is sliced in 1 in square dices suitable for the oxidation chamber. Just before performing PECE, we must remove the oxide layer from the wafer, done by rinsing the die in 5 % HF for about a minute.

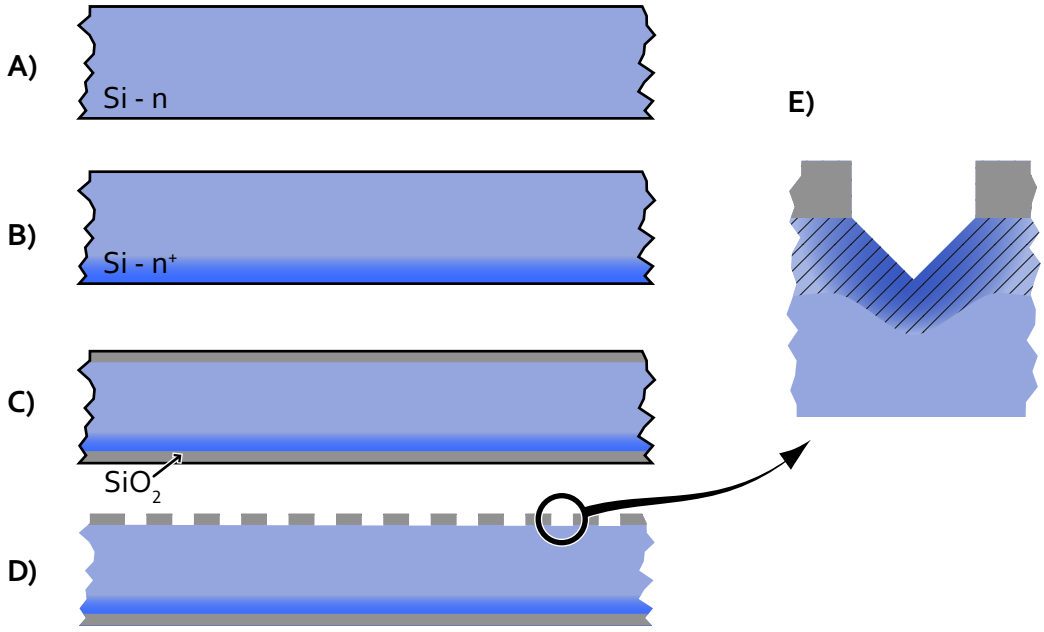


Fig. 7 – Silicon wafer preprocessing. A) Original n-type wafer. B) backside ion implant. C) oxidation. D) lithography and RIE etching of the pore array E) Detail of the pit produced after anisotropic etching, showing the potential region under bias in contact the electrolyte (and the oxide layer removed).

The etching system is composed by an electrolytic cell, a bypass cell, two HF reservoirs, a pump, a cooling system, and measure and control equipment. All components in contact with the hydrofluoric acid are made or covered of Teflon to avoid corrosion. The cooler circuit set the acid solution at a temperature of 15°C using a heat exchanger in the main reservoir; this process takes between 30 minutes to 1 hour depending on ambient temperature. The reaction cell is formed by a spring-loaded plate with a Pt cathode and a fixed plate on a detachable support. The Si sample is placed between the plates with the backside (unpolished face) on the fixed plate. This plate has a circular opening of ¾" exposing the bottom face of the sample to the IR light. The illumination is provided by an array of IR LED controlled by a current source up to 600 mA, though they are limited to less than 200 mA to avoid burning them.

The whole etching unit is controlled by computer with a homemade LabView® program. The etch characteristics are fed to the program with a text configuration file and a waveform text file. Even though is possible to specify other working temperatures, currently, the best characterized etching behaviour at our disposal is for HF 2.5 %wt. at

15°C. The program also produces an output file with the measures of the actual process, allowing later inspection.

Several samples were etched with run times ranging from 4.5 to 6 hours, with lattice constant of 2 μm and achieving lengths from 200 μm to 290 μm. Generated samples were modulated in the pore axis, this needed adjusting the current and bias waveforms to get the desired profile. As the actual pore development is approximate, to ensure that the result is near the required, some samples were created just to test the profile was produced as expected. Those samples had periodic current pulses to generate notches in the profile at equal spacing. The current waveforms were computed using an empirical model, and it can be appreciated that to obtain straight pores with the notches, the waveforms were compressed in time at the beginning gradually expanding until the finish of the process. This manifests the mentioned effect of reaction speed slowing as pores grow deeper.

After the PECE, some samples were oxidized and later etched in KOH and TMAH to produce porous membranes. The test samples were split along a principal direction and inspected using an optical microscope to determine the growth development.

IV – Fourier Transform Infrared Spectroscopy

One of the main measurement methods available in the materials fields is *Spectroscopy*. This technique consists on characterizing the transmission or reflection features of an incident light onto the sample. Depending on the spectral range of the light, the observed traits correspond to different physical phenomena that range from atom excited levels at the UV to molecule oscillatory excitation at IR.

This properties of spectroscopy allow to assign distinctive attributes to different compounds, so that is possible to latter determine which are in an unknown sample. This is the main use in the chemistry area, where this system was developed.

In a broader sense, spectroscopy simply relates the spectra of the incident light to that of the received light as shown in **Fig. 8**. The result of such relation provides us with the general reflectance or transmittance as function of wavelength and incident angle. Measurements can be refined to include other parameters as well, such as polarization dependence or non-specular reflection.

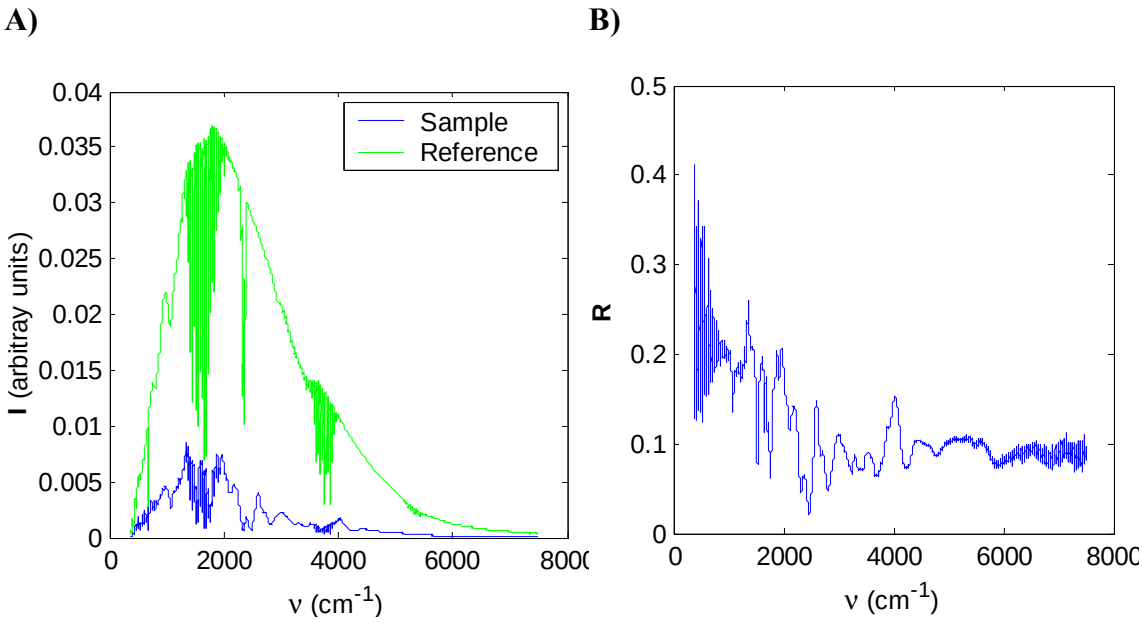


Fig. 8 – A) Reflection Spectrum as obtained by direct measurement. B) Reflectance of the sample given as the quotient S(*ν*)/R(*ν*).

In this aspect, the incident radiation does not need to be in the optical range. With appropriate equipment, such measurements can be taken down to the microwave region. Nevertheless, working at lower frequencies requires bigger samples and equipment, so there is a practical limit to the lowest measurable frequencies. While from and above X-ray, the limit is imposed by the optical system.

4.1 – Operation

Typical optical spectroscopy consist in filtering the broadband incident light with a *diffraction grid*, and then select the wavelength of interest by tilting some element—like

a mirror. This effectively works as a wavelength filter, where the diffraction grid scatters the incident light wavelength in different directions. It is analogous to a Spectrum Analyzer used in RF.

Nevertheless, this has several disadvantages. We may mention the power reduction of the analyzed slice of spectra. As we narrow the filter bandwidth, power density remains constant, thus total power arriving to the sensor reduces greatly. This imposes a technical limit on the minimum bandwidth by the maximum sensitivity of the detector.

On the other hand, *Fourier Transform* spectroscopy uses numerical techniques to achieve the same goal with better results. FT-IR is based on the duality between *spatial interference* and *Power Spectrum Density* of the Fourier Transform.

Instead of using a *monochromator* to resolve the different frequencies, the full broadband light spectrum is used at a time. Basically, the light is split in two different optical paths and then is recombined with an *interferometer*. Making one of the paths fixed, and the other of variable length, a certain delay is introduced. When both beams are recombined coherently, the relative phase between them creates the interference pattern that arrives at the detector. The most common interferometer construction is the *Michelson Interferometer* shown in **Fig. 9**.

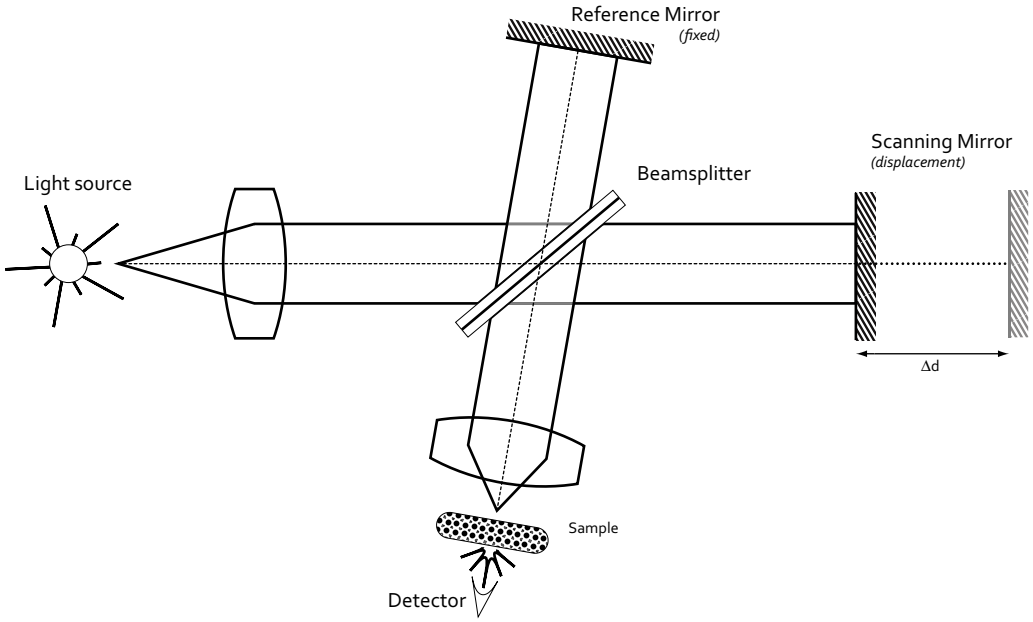


Fig. 9 – Michelson Interferometer. Basic set up. The reference mirror provides a fixed delay beam, while the scanning mirror delays the wave by a factor of *2d* respect the reference.

One of the most delicate and critical parts of the instrument is the *beamsplitter*. This device is a semitransparent mirror that divides the incoming light. The beamsplitter must be able to act as ideally as possible for a broad frequency range. However small phase and amplitude mismatch occur at the different wavelengths. This has some implications in the received signal and its post processing. For our working range in mid-IR, we use the standard KBr beamsplitter suitable for a spectral range from 7,500 to 370 cm^{-1}.

The interfered light is directed to the sample and then sent to the detector sampling the intensity of the signal. The linearity property of the Fourier Transform implies that the sample placement before or after the interference process is not important. To control the sampling points, the spectrometer uses a reference He-Ne laser at a wavelength $\lambda = 615.8\ \mathrm{nm}$ that undergoes the same interference process.

The scan procedure consists on displacing the free mirror a distance. The amount of displacement controls the maximum resolvable frequency by the relation $\Delta x = d \leftrightarrow \Delta f = 1/d$.

The interference process is described by

$$s'(t,\tau) = \frac{1}{2}s(t) + \frac{1}{2}s(t-\tau), \tag{11}$$

as the detector only takes time averages of received power, the output signal is thus

$$o(\tau) = \overline{s'(t,\tau)}, \tag{12}$$

which has a form of an autocorrelation of a stochastic process. Applying the Fourier Transform to **Eq (12)** we obtain the PSD of the signal which is the spectrum we are interested. For instance, the single frequency provided by the laser reference produces the following interference pattern:

$$s(t) = I^2\cos^2(\omega t) = \left[I\,\mathrm{Re}\{\mathrm{e}^{j\omega t}\}\right]^2, \tag{13}$$

$$\begin{aligned} s'(t) &= \frac{1}{2}\left[I\,\mathrm{Re}\{e^{j\omega t} + e^{j\omega(t-\tau)}\}\right]^2 = \frac{1}{2}\left[I\,\mathrm{Re}\{e^{j\omega t}(1+e^{-j\omega\tau})\}\right]^2 = \\ &= \frac{1}{2}\left[I\,\mathrm{Re}\{e^{-j\omega(t-\tau/2)}\}\,2\cos(\omega\tau/2)\right]^2 = 2\cos^2(\omega\tau/2)\,s(t-\tau/2), \end{aligned} \tag{14}$$

where $s(t)$ is the power of the laser beam and $s'(t)$ is the power of the *coherent sum* of the split beam; therefore, the output signal is

$$o(\tau) \propto 2\cos^2(\omega\tau/2) = \cos(\omega\tau) + 1, \tag{15}$$

and the associated spectrum after the Fourier Transform,

$$I(f) = \mathcal{F}\{o(\tau)\} \propto \delta(f) + \delta(f-\omega) + \delta(f+\omega). \tag{16}$$

From the above we see that the spectrum corresponds to that of the light frequency. This makes it possible to sample the signal at twice the reference frequency (zero crossings) giving a total spectral range from 0 to $\lambda_{\mathrm{ref}} = \lambda_{\mathrm{He\text{-}Ne}}$.

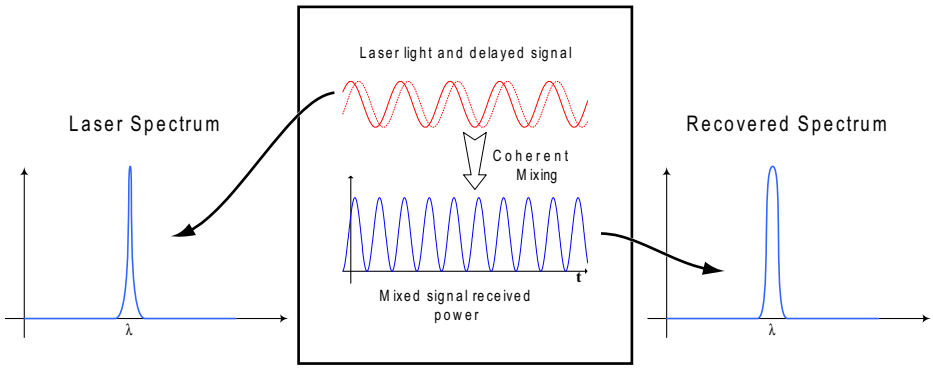


Fig. 10 – FT-IR operation for He-Ne laser reference. The original source has a very narrow spectral line; the interferometer operation is shown in the middle, where both the reference beam and a delayed copy are added coherently. The power signal of result given by the detector is shown below, consequently, the calculated spectrum is similar to the original. Broadening may occur by noise added by the system.

The numerical processing of the interference pattern has certain aspects that must be known ant taken into account. As commented, the beamsplitter is not perfect and adds certain imbalance in phase and amplitude of the spectrum. The sampling process also sums some phase noise in. These phase variations translate to the interferogram loosing its symmetry; this is important when performing the Fourier Transform as the received signal has lost the phase information. To mitigate this effect we can apply the *Mertz* algorithm that estimates the phase from the power signal. A second enhancement is performing a *double-sided forward-backward* scan, such that the sampled signal contains both sides of the interference patters, at the cost of double scan time.

The biggest advantages of FT-IR are known as the CONNES, JACQUINOT and FELLGETT advantages. The first one refers to the increased accuracy of the sampling points thanks to the use of a highly precise He-Ne reference. The second one refers to the increased power arriving to the detector because this type of interferometer uses circular apertures instead of slits. Finally, the last one states that the entire spectrum is captured simultaneously. These advantages imply an improved Signal-to-Noise Ratio. Additionally to those listed, this spectrometer has also an increased scanning speed, as the moving mirror has to travel relatively short distances.

4.2 – Measuring with the FT-IR

The equipment used is a *Brucker Optics Vertex 70*, with two light sources, a *globar* source for the MIR (Medium IR) and a tungsten lamp for NIR (Near IR) and visible. It also provides two different beamsplitters, KBr and CaF2, to best suit the measure range. There are also two detectors available, DLatGS and MCT, but the last one requires to be cooled with liquid nitrogen to operate, and it is not used for our purposes. Additionally, it was acquired the reflection accessory and its polarization filter.

This equipment allows performing different kind of measures, at different light ranges and experiment configurations. In this work, we are primarily studying the reflection properties of macroporous silicon. The *Vertex 70* also integrates an external furnace to heat samples and measure their emission spectra.

The FT-IR connects by a standard Ethernet link to a PC, which hosts the dedicated measuring software *OPUS*, from the same manufacturer. The equipment is designed to

be operated from this software; all configuration and control of the FTIR are specified inside the application. Using this software, we configure the optical path and the measure type and parameters. With the *OPUS* software we can perform simple measures and later post-process the acquired spectrums, apply different algorithms and extract important information, such as substance recognition or layer thickness. The application provides also an external API interface and an internal scripting engine, which allow to create tailored measuring scripts, and to automate them.

Fig. 11 – OPUS interface. A) shows the "Advanced Measure**" Dialog, in particular the optical path configuration tab. B) shows the angle sweep script interface: (a) is the output directory, (b) the sample, (c) chooses the experiment setup, (d) shows the sweep controls, (e) are other manual controls, and (f) is an output panel showing progress messages and errors.**

Even though we can work in both the MIR and the NIR range, the current samples used have responses around 10 μm, thus we limit ourselves to work in the MIR range from 7500 cm^{-1} to 375 cm^{-1}.

Table II – Spectral ranges. For both MIR and NIR, we give the spectral range of each band in both wavenumber and wavelength.

	MIR	NIR
wavenumber k (cm^{-1})	7000 – 375	15000 – 4000
wavelength λ (μm)	1.43 – 26.67	0.67 – 2.5

The typical optical parameters used are an aperture of 2 mm, KBr splitter, DLatGS detector, 10 kHz scanner speed and no filtering. With those settings we achieve good sensitivity and acceptable spectrum acquire speed. To reduce noise, we average from 50 to 100 measures, depending on the sample.

The Fourier Transform parameters usually applied are full spectral range from 7500 to 375 cm^{-1}, .Mertz phase algorithm, 3-term Blackmann-Harris filter and zero-fill factor of 2 or 4.

4.2.1 – Reflection measures

Our main measures consist of the reflection characterization of macroporous Silicon at different angles, discriminating TE and TM polarizations. To perform such measures we use the *reflection accessory* coupled with a polarizer. Both add-ons are fully motorized and provide an *Automated Tool Interface* allowing them to be remote controlled.

As we need to execute an angle sweep, an automation script was developed to help carry out this task. This script is run inside the OPUS interpreter and provides a number of options to set up the measure. We first need to specify the optical and transform parameters in the "*Advanced Measure*" dialog; the typical values used are those commented before. Alternatively, we can directly choose a saved configuration file. We next give the sample name and output directory.

Next item to specify is the angle sweep range and step. Practical values are from 13° to 87° sweep and 2° step. The script allows choosing which polarization to take, S, P, or both. There are additional controls to operate manually other parts of the equipment, like the mirror position.

The script generates two files per iteration, one file with extension "*.0*" saving the original spectrum in OPUS format, and the other with extension "*.dpt*" saving only the sample spectrum in text format. Files generated will have the name prototype:

sample-name_[*S*,*P*]Pol_*nn*deg.<extension>

where *S,P* indicates the polarization and *nn* expresses the incidence angle at which the measure was taken. The text format file is later used in MATLAB to perform data analysis.

4.2.2 – Emission measures

For emission measures we need to use the external oven. In contrast to the reflection tool, this oven has a manually controlled temperature regulator, thus for temperature sweeps, we cannot create and automation script. To use the external furnace, we need to set up the optical parameters accordingly in the "*Measure*" dialog.

The oven consists of an aluminium electric heater were we place the sample, inside a water-cooled enclosure. This enclosure has a window were the thermal radiation exits to inside the FT-IR.

4.3 – Calibrating the FT-IR

Prior to use the equipment must be calibrated. This is initially done after installation by the technician, with no further intervention. Similarly, switching the beamsplitter does not require recalibration of the FT-IR.Using the add-on accessories is also straightforward as they come pre-calibrated from the manufacturer.

A special case is the polarizer. This device is mounted on the reflection unit, and lacks an alignment mechanism. The installation is guided by some marks in the accessory to correctly align the filter, but nevertheless, is difficult to achieve high

accuracy just by visual inspection. This means that the filter was installed in an approximate position and that the angular error was to be corrected afterwards by rotating the filter by an adjusting factor.

We need to install the filter to perform polarized measurements, as the instrument source light is ideally not polarized. When measuring reflectance with the *a513* accessory it must be taken into account that light polarization on the sample is exchanged with respect the polarizer set-up, as shown in **Fig. 12**.

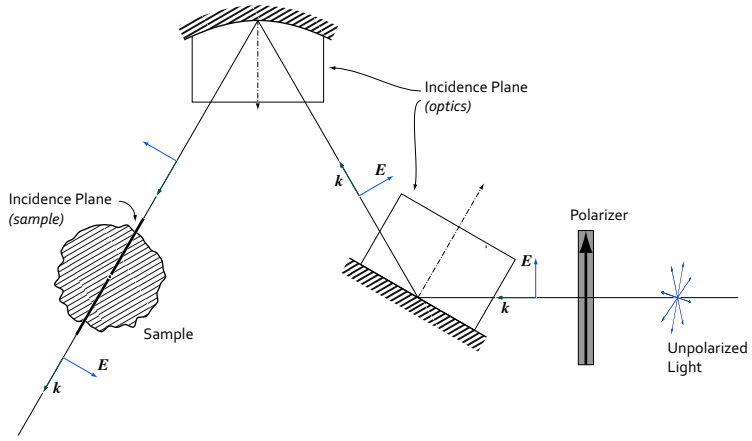


Fig. 12 – Polarization reversal on the sample respect the system polarization. The incidence plane of the sample is perpendicular to the incidence plane on the mirror ensemble.

this occurs as the plane of incidence is shifted 90° in the sample with respect the measuring equipment mirrors. We must also take into account that, as we change the incidence angle, the polarizer must also rotate to maintain the polarization orientation. To accomplish this we first need to calibrate the instrument. The reflection accessory is calibrated from the manufacturer, but the filter was manually installed after.

The calibration procedure consisted on measuring the reflection on a reference mirror for a fixed incidence angle near the **Brewster angle**. The polarizer filter was swept for a range of 8° with steps of 0.25° for both P and S (90° apart) polarizations. The collected data was then filtered and a second order surface (paraboloid) was fitted. The minimum of the data then corresponds to the correction factor in degrees needed to apply to the polarizer angle.

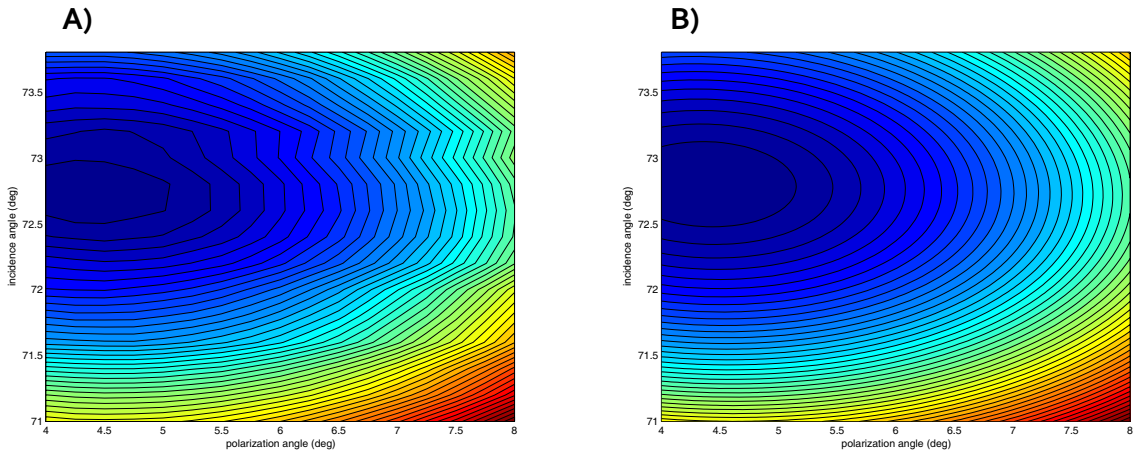


Fig. 13 – Brewster angle calibration of the polarizer. A) direct measure, B) after fitting.

The analysis showed that the angle deviation was about 4.4° for an incidence angle of 72.8°, but as the polarizer stepper motor moves only in 0.25° increments, the applied correction was 4.5°.

When performing measures, we need to calibrate them against a reference. We need to do this to eliminate the deterministic error introduced by the optical path. For the reflection measures, this is accomplished by using a reference mirror instead of the sample. This mirror acts as a near perfect reflector, thus the received signal contains only the information regarding the optical path. The procedure consists in measuring both the sample and the reference mirror at same angles and polarizations, then the calibrated spectrum is obtained by dividing both:

$$S(\nu) = \frac{S_{\text{sample}}(\nu)}{S_{\text{ref}}(\nu)}, \tag{17}$$

For the emission case, things change slightly. To calibrate the optical path, including the oven, we need a reference black body. We simply need to do this once[1], as the optical path remains unchanged. The black body is constructed using a piece of silicon substrate and coating it with soot. We must also pay attention to correctly align the sample in the heater with the oven window.

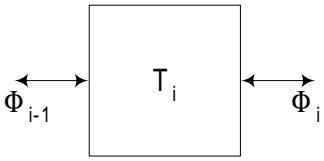
1 If we could discriminate between polarizations, we would need to take two measures, one per polarization.

V – Software Tools

5.1 – TMM

One important aspect that arises when designing photonic crystals is the need to simulate their behaviour. The study of optical materials requires knowledge of the expected behaviour of those, and the ability to predict how changes will affect the results. Even though obtaining analytical solutions or models is the best achievement, often this is not possible or desirable due to the complexity of the problem or the presence of expressions that prevent it. Nevertheless, it is desirable to obtain at least approximate models to qualitatively explain the results. In this aspect, the simulated model is compared against the collected data, and if the model is good, both should agree qualitatively.

One important simulation tool available for the study of wave propagation is the *Transfer Matrix Method* (TMM). This is a general formulation that relates the waves at one side of the interface with the waves at the other side using the so called *transfer matrix* (T-matrix). In fact, the TMM can describe the behaviour of any linear system, for example a microwave transmission line.



$$\text{TMM:} \quad [\mathbf{\Phi}_i] = \mathbf{T}_{i-1\to i}[\mathbf{\Phi}_{i-1}], \tag{18}$$

cascaded system:

$$[\mathbf{\Phi}_N] = \mathbf{T}_N\mathbf{T}_{N-1}\cdots\mathbf{T}_2\mathbf{T}_1[\mathbf{\Phi}_0]. \tag{19}$$

Usually, the T-matrix is derived using analytical expressions, as is the case of electromagnetic wave propagation. At the frequencies of our interest, we can use the *Geometrical Optics* approach that considers the incident wave a plane wave and the point on the surface at where we evaluate the equations to be locally flat. This allows us to assume that both the reflected and refracted waves are also plane waves, independent of the fields at adjacent points. The resulting matrix then relates the fields at one point at either side of the interface.

When studying PCs, the TMM is useful to determine the total reflection and transmission coefficients, but it does not show us the actual fields inside the crystal. Simple TMM models are very useful to study 1D PC, as the cascading property allows relating easily the different layers to one T-matrix each directly. For 2D and 3D PCs, the TMM can be extended using *Planewave Expansion* and *slicing*, but this is not our case.

We are interested in using the 1D TMM to simulate the reflectance of a thin layer of Si as a function of frequency to see the expected features of the measured spectra with the FT-IR of such layer. The transfer matrix used for our simulation is formulated assuming that the layers are dielectric parallel slabs, free of charges and no sources; we now impose at each side of the interface the presence of both advancing and backwards plane waves with fixed directions:

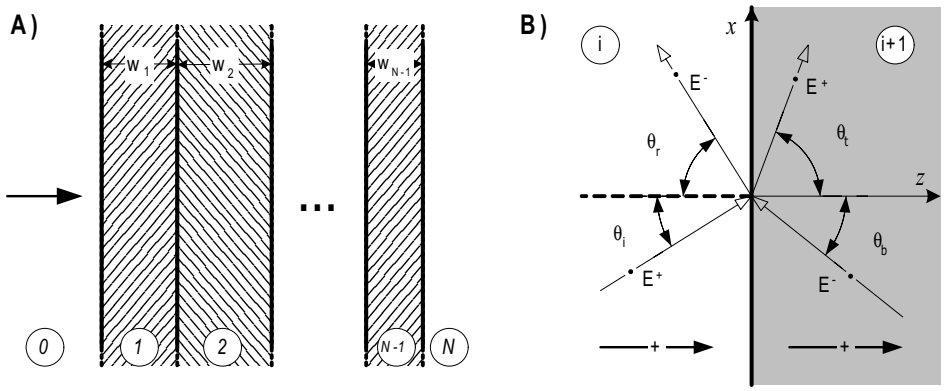


Fig. 14 – A) stack of layers defining the crystal .B) Geometry definition for deriving the TE polarized TMM expressions. Advancing waves are denoted with a '+' sign.

For TE polarization, we have the following fields:

$$\mathbf{E}_i = \hat{\mathbf{y}} E_i^+ \, \mathrm{e}^{-j\beta_i (x\sin\theta_i + \cos\theta_i)} + \hat{\mathbf{y}} E_i^- \, \mathrm{e}^{-j\beta_i (x\sin\theta_r - \cos\theta_r)}, \tag{20.a}$$

$$\mathbf{E}_{i+1} = \hat{\mathbf{y}} E_{i+1}^+ \, \mathrm{e}^{-j\beta_{i+1} (x\sin\theta_t + \cos\theta_t)} + \hat{\mathbf{y}} E_{i+1}^- \, \mathrm{e}^{-j\beta_{i+1} (x\sin\theta_b - \cos\theta_b)}, \tag{20.b}$$

$$\mathbf{H}_i = (-\cos\theta_i \hat{\mathbf{x}} + \sin\theta_i \hat{\mathbf{z}}) \frac{E_i^+}{\eta_i} \mathrm{e}^{-j\beta_i (x\sin\theta_i + \cos\theta_i)} + (\cos\theta_r \hat{\mathbf{x}} + \sin\theta_r \hat{\mathbf{z}}) \frac{E_i^-}{\eta_i} \mathrm{e}^{-j\beta_i (x\sin\theta_r - \cos\theta_r)}, \tag{20.c}$$

$$\mathbf{H}_{i+1} = (-\cos\theta_t \hat{\mathbf{x}} + \sin\theta_t \hat{\mathbf{z}}) \frac{E_{i+1}^+}{\eta_{i+1}} \mathrm{e}^{-j\beta_{i+1} (x\sin\theta_t + \cos\theta_t)} + (\cos\theta_b \hat{\mathbf{x}} + \sin\theta_b \hat{\mathbf{z}}) \frac{E_{i+1}^-}{\eta_{i+1}} \mathrm{e}^{-j\beta_{i+1} (x\sin\theta_b - \cos\theta_b)}, \tag{20.d}$$

if we now impose the continuity conditions at the interface, we have:

$$\begin{aligned} &|\mathbf{E}_i|_{z=0}^{\tan} = |\mathbf{E}_{i+1}|_{z=0}^{\tan} \rightarrow \\ &E_i^+ \, \mathrm{e}^{-j\beta_i x\sin\theta_i} + E_i^- \, \mathrm{e}^{-j\beta_i x\sin\theta_r} = E_{i+1}^+ \, \mathrm{e}^{-j\beta_{i+1} x\sin\theta_t} + E_{i+1}^- \, \mathrm{e}^{-j\beta_{i+1} x\sin\theta_b}, \end{aligned} \tag{21.a}$$

$$\begin{aligned} &|\mathbf{H}_i|_{z=0}^{\tan} = |\mathbf{H}_{i+1}|_{z=0}^{\tan} \rightarrow \\ &\frac{1}{\eta_i}\left[-\cos\theta_i \, E_i^+ \, \mathrm{e}^{-j\beta_i x\sin\theta_i} + \cos\theta_r \, E_i^- \, \mathrm{e}^{-j\beta_i x\sin\theta_r}\right] = \\ &\frac{1}{\eta_2}\left[-\cos\theta_t \, E_{i+1}^+ \, \mathrm{e}^{-j\beta_{i+1} x\sin\theta_t} + \cos\theta_b \, E_{i+1}^- \, \mathrm{e}^{-j\beta_{i+1} x\sin\theta_b}\right], \end{aligned} \tag{21.b}$$

solving the complex part we find the angle relations given by *Snell's law*: $\theta_i = \theta_r$, $\theta_t = \theta_b$ and $\beta_{i+1}\sin\theta_t = \beta_i \sin\theta_i$; so we can express them in matrix form:

$$\begin{bmatrix} 1 & 1 \\ \dfrac{\cos\theta_i}{\eta_i} & -\dfrac{\cos\theta_i}{\eta_i} \end{bmatrix} \begin{bmatrix} E_i^+ \\ E_i^- \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ \dfrac{\cos\theta_t}{\eta_{i+1}} & -\dfrac{\cos\theta_t}{\eta_{i+1}} \end{bmatrix} \begin{bmatrix} E_{i+1}^+ \\ E_{i+1}^- \end{bmatrix}, \tag{22.a}$$

for non-magnetic material: $\eta = \sqrt{\mu_0}/n$, thus:

$$\begin{bmatrix} 1 & 1 \\ n_i \cos\theta_i & -n_i \cos\theta_i \end{bmatrix} \begin{bmatrix} E_i^+ \\ E_i^- \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ n_{i+1} \cos\theta_t & -n_{i+1} \cos\theta_t \end{bmatrix} \begin{bmatrix} E_{i+1}^+ \\ E_{i+1}^- \end{bmatrix}, \tag{22.b}$$

$$\mathbf{D}_i \begin{bmatrix} E_i^+ \\ E_i^- \end{bmatrix} = \mathbf{D}_{i+1} \begin{bmatrix} E_{i+1}^+ \\ E_{i+1}^- \end{bmatrix} \rightarrow [\mathbf{E}_{i+1}] = \mathbf{D}_{i+1}^{-1} \mathbf{D}_i [\mathbf{E}_{i+1}], \tag{23}$$

for TM polarization, we perform a similar procedure and get:

$$\mathbf{D}_i = \begin{bmatrix} \cos\theta_1 & -\cos\theta_i \\ n_i & n_i \end{bmatrix}. \tag{24}$$

We now find the propagation matrix, that is, the transfer matrix that accounts the layer thickness; this is trivial, for layer *i*:

$$[\mathbf{E}_i] = \mathbf{P}_i [\mathbf{E}_i']; \quad \mathbf{P}_i = \begin{bmatrix} e^{j\beta_i \Phi_i} & 0 \\ 0 & e^{-j\beta_i \Phi_i} \end{bmatrix}, \tag{25}$$

where **E** represents the fields at interface (i-1,i), **E**' at interface (i,i+1), $\Phi_i = w_i \cos\theta_i$ is the electric thickness of the layer, and *w* the actual layer thickness.

The complete TMM is finally:

$$[\mathbf{E}_0] = \mathbf{D}_0^{-1} \prod_{j=1}^{N-1} \mathbf{D}_j \mathbf{P}_j \mathbf{D}_j^{-1} \mathbf{D}_N [\mathbf{E}_N] \tag{26}$$

This system was implemented as MATLAB scripts.

5.1.1 – Non-coherent TMM

The TMM method developed previously accounts for both the field's amplitude and phase at the interface, and propagates them both as well. This has the consequence of adding fields in a coherent way. This effect is specially pronounced in thick flat surfaces, where TMM simulations yield high frequency Fabry-Perot oscillations. In essence, the T-matrix simulates perfect mirrors and propagation media.

In reality, perfectly flat surfaces are difficult to obtain, in particular for thick layers. Surfaces generally have a certain degree of roughness that scatters the incident light in all directions. This roughness determines the behaviour of the surface as either *specular reflector*(mirror) or *diffuse reflector* (matte surface). This roughness is of random nature and thus cannot be simulated directly by TMM. This raises the need to use indirect methods to account for its effect.

Roughness affects phase, amplitude and direction; but if for small value, Gaussian distributed noise, it can be simplified to affecting only phase and amplitude. In a TMM simulation, the interest is obtaining the *average* fields; hence, we can add a correcting factor for both in the T-matrix formulation.

As the layer thickens, the phase and amplitude mismatch becomes greater. At a certain point, all light reflected back by the back surface is no longer correlated to the

incident light. In such situation, all light adds up, regardless of phase, thus we can simplify the TMM to the addition of *light intensity*.

In our study, we limit to such simplified case. With such simplification, the T-matrix for propagation and interface change.

5.1.2 – Grazing angle incidence

An important aspect to take into account is that TMM simulates surfaces of infinite extent. The T-matrix formulation computes the reflection and transmission *of a single ray* at some point and assumes the surface is perfectly flat.

Actually, samples are of finite size and incident light generally are Gaussian beams. In this situation, we are illuminating a definite area of the sample at different incident angles determined by the *spot size* and the *beam divergence*. As the incident angle is increased, so does the spot illuminated in the sample, until it finally falls out of the sample boundaries. The portion of the beam that does not light the sample, may be lost or affect in some other way, depending on the experiment setup and the surrounding of the sample.

Even if we consider the sample of infinite size, depending on the divergence, from a certain angle on, part of the incident beam lies below the sample surface. This portion of the beam will be definitely lost.

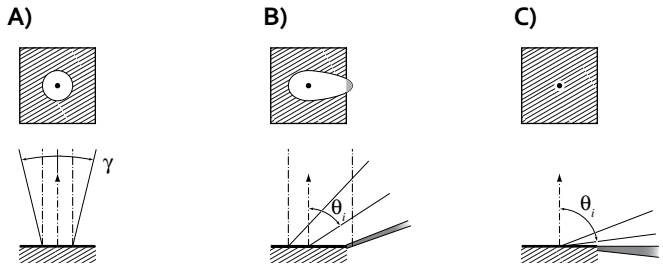


Fig. 15 – Effect of finite sample size and focused beam. A) normal incidence; the sample is illuminated on a spot. The beam divergence is given by γ. B) for a certain angle, part of the light spot falls out of the sample. C) Ideal focused beam; for a certain incidence angle, part of the beam is bellow the sample.

We can easily account for the effect of an ideal perfectly focused beam when part of it is not reflected. For it, it is enough to know the beam divergence, as spot size is zero by definition, and thus we can ignore the finite size of the sample. We need to find the ratio between the received power, reflected by the sample, and the incident power. This will yield the *effective* reflection coefficient:

$$r_{\text{eff}} = \frac{P_{\text{reflected}}}{P_{\text{incident}}} = \frac{\int_{S'} r(\theta')|\mathbf{E}|^2 \, d\mathbf{S}}{\int_{S} |\mathbf{E}|^2 \, d\mathbf{S}}, \tag{27}$$

for a perfectly focused beam, we can consider that light comes from a perfect spherical surface, therefore, the reflected wave is also a spherical wave. If we take both the emitter and receiver to be at the same distance, we can simplify the problem to integrals over the unity sphere. We further consider that the incident field is of constant amplitude over the whole emitter. Applying these simplifications, we obtain,

$$r_{\text{eff}} = \frac{|\mathbf{E}|^2 \int_{\mathbf{S}'} r(\theta')\, d\mathbf{S}}{|\mathbf{E}|^2 \int_{\mathbf{S}} d\mathbf{S}} = \frac{\int_{\mathbf{S}'} r(\theta')\, d\mathbf{S}}{2\pi[1-\cos(\gamma)]}, \tag{28}$$

To perform the computations, we must find the reflected beam integral boundaries, which change with incidence angle. It is easy to determine the boundaries, to achieve this purpose, one builds a map of the actual incidence angle over the spherical surface. Once the map is built, it is only needed to find and discard those points where the angle is greater that 90°. Those points can be thought of as a *domain mask*, where the integrand evaluates to zero,

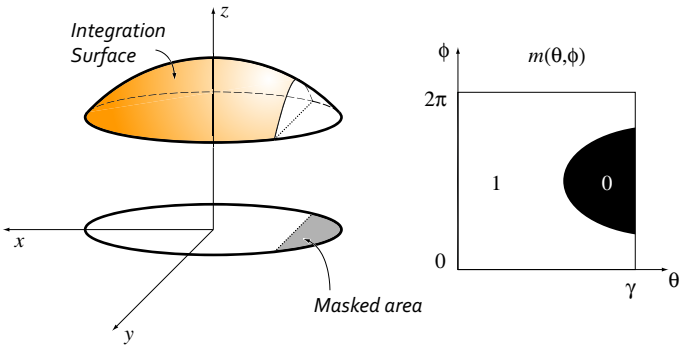


Fig. 16 – Observable solid angle at grazing incidence.

$$\int_{\mathbf{S}'} r(\theta')\, d\mathbf{S} = \int_{\mathbf{S}} m(\theta,\phi)\, r(\theta')\, d\mathbf{S} = \int_0^{2\pi}\int_0^{\gamma} m(\theta,\phi)\, r(\theta')\sin(\theta)\, d\theta d\phi\,, \tag{29}$$

$$\theta' = \arccos(\sin(\theta_i)\sin(\theta)\cos(\phi) + \cos(\theta_i)\cos(\theta))\,, \tag{30}$$

$$m(\theta,\phi) = \begin{cases} 0 & \theta' > \pi/2 \\ 1 & \text{otherwise} \end{cases}. \tag{31}$$

In all these equations, r is the reflection coefficient for angle θ, at the frequency of interest, found by the direct application of TMM.

5.2 – MPB

The *MIT Photonic Bands* (**MPB**) is a free eigenmode solver for computing the resonant electromagnetic modes of infinite crystal structures. This software package uses harmonic decomposition to find the field modes that satisfy the *master equation*,

$$\nabla \times \left[\frac{1}{\varepsilon(\mathbf{r})}\nabla \times \mathbf{H}(\mathbf{r})\right] = \left(\frac{\omega}{c}\right)^2 \mathbf{H}(\mathbf{r}), \tag{32}$$

The MPB software is an open source, text mode program that runs on UNIX type platforms. This software is also capable of running in multiple processors and machines using the version compiled with MPI support.

To use the **MPB** application, is necessary to specify the primitive cell that generates the crystal lattice in space. This may either form a 1-dimension to a 3-dimension crystal by giving a set of lattice basis vectors and the cell size in each

direction. The system is solved applying discrete translational symmetry by size along each lattice direction, as shown in Fig. 17. A detailed description will be given in the following subsections.

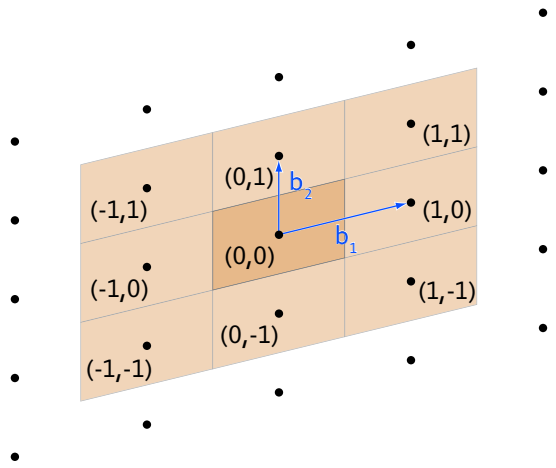


Fig. 17 – The lattice basis describes the primitive Cell of a 2D periodic structure.

We also need to give the *k-points* of interest in the *reciprocal space*. The k-points are the normalized propagation vector of the incident electromagnetic wave. For this reason to improve accuracy is advisable to constrain those to the *first Brillouin Zone*. By the *Bloch Theorem*, to completely determine the fields inside an infinite regular crystal, it suffices to find the fields for a given k-point inside the first Brillouin Zone. Furthermore, additional symmetry operations of this zone allow a further reduction of the unique k-points to the *irreducible Brillouin Zone*.

The program does not calculate the irreducible Brillouin Zone, thus when designing a photonic crystal, is necessary to manually find the points of interest in this zone and feed them to the application. Nevertheless, the program offers the possibility to reduce an arbitrary k-point to the first Brillouin Zone.

This software package allows to selectively output information as the specified bands at each k-point of interest, the fields distribution in the primitive cell, the dielectric function and other used-defined parameters as needed. Data may be obtained either as text in standard output, or as a separate dataset file in HDF5 format.

5.2.1 – User Interface and Control Language

A particularly interesting feature of the MPB simulation software is the use of a simple scripting programming language for control and problem description. The program takes text commands to define the setting for the simulation, instead of taking a fixed parameter file describing the geometry, sources and environment, or using a graphical user interface.

As commented, MPB provides an interactive interface to the user. It presents an input accepting *Scheme* instructions, which are processed by the *GNU Guile* Scheme interpreter. *Scheme* is a high level programming language, whose syntax is very similar

to Lisp and freely available. Alternatively, MPB can run in *batch* (non-interactive) mode, by supplying an input control file at command invocation. This file is in fact a list of Scheme commands, as one would put in the interactive input from MPB.

It must be pointed out that the Guile interpreter is case sensitive; this is in conflict to the official Scheme syntax, which is case insensitive. This implies that some code written for other interpreters may not work as expected.

Using an easy and standard scripting language gives a very powerful interface to the user. The user has at his disposal the capabilities and features of a full programming language. This opens the possibility to interactively set up the simulation geometry and parameters, and do quick prototyping before performing the final, and more detailed, simulation. The existence of flow control structures allows the definition of complex, non-typical, regular or irregular patterns with ease in a few lines, or the creation of complex computation algorithms. The MPB command interpreter also allows defining generic functions to generate common geometry elements or other parameters in external shared files, which can later be used as *loadable modules* to extend the original programming language environment.

Aside form the generic capabilities offered by the scripting interpreter, MPB relies on a set of specific libraries created for electromagnetic simulation programs and mathematic analysis. We remark the *libctl* library, which provides a rich set of geometric manipulation routines for both C and Scheme programming interface. It also defines a convenient set of vector and matrix handling functions. This library was created by the same developer of MPB, initially as part of the main project. It has since evolved to provide a skeleton for user created applications based on it.

Additionally to the types and functions defined in the *libctl* library, MPB defines a specific set of variables and functions exclusively for the computations performed by it. The available and most used definitions are described bellow:

The MPB programming environment presents to the user the basic Scheme types—such as lists, real and complex numbers, strings or vectors—and a basic class based object system for the more elaborate types—such as the geometric objects. Objects are created using the following syntax:

```
(make class (property value)...(property value))
```

class specifies the object prototype class and must be given. The most common used classes are *lattice*, *dielectric*, *material-function*, *block*, *sphere*, *cylinder*, *cone* and *ellipsoid*. When creating an object it will be necessary to specify some of its properties. A class can define a default value for a property if the user does not specify one at object creation, or may leave the field blank forcing user input, and throwing an error if the user forgets to set it.

Besides the objects used to describe the computation cell and geometry, MPB uses a set of predefined variables to set-up the parameters for the simulation. The most important are explained in the following list.

- **resolution**, *number or vector*: it defines the computational cell grid spacing. If it is a number, each direction is affected in the same amount. Conversely, if a 3D

vector, defines the grid spacing in each space direction. It has a default value of 10 divisions per unit.

- **num-bands**, *number*: the number of states we want to find. For each k-point it solves the master equation determining *num-bands* eigenvectors. By default, it computes 1 band.

- **k-points**, *list of vector*: a list of 3D vectors. MPB computes the egeinvalues for each k-point in the list in order. The vectors are given in the *reciprocal space* of the lattice. The list is initially empty.

- **geometry-lattice**, *lattice*: object of class lattice. It describes the computational cell. If not specified, defaults to Cartesian basis.

- **geometry**, *list of geometric-object*: a list of objects of class geometric-object. Each object determines the epsilon tensor in its volume. When overlapping geometry, the later object takes precedence. The default is an empty list.

- **default-material**, *material-type*: object of class material-type. It defines the material that will be used wherever there is no geometric object. Any points not covered by an object will default to *air* if not defined.

MPB defines some useful constants, particularly used to describe the geometry and materials:

- **air**, *dielectric*: isotropic material of dielectric constant of 1. It is an alias to *vacuum*.

- **nothing**, *material-type*: a material that produces a hole in the underlying geometry to the background *default-material*.

- **infinity**, *number*: a very large number used to flag and “infinite” value. Any value larger than this is considered infinite.

- **no-size**, *number*: a flag used to discard a dimension.

The most important classes defined in MPB are those cited above. In the following list, we will describe them shortly:

To obtain useful simulations we must define at least a non-empty geometry list and assign some user defined materials to the objects. If not, by default we would simulate vacuum. The next thing to adapt to our needs is the lattice. The initial lattice is cubic, which is not always the desired cell. To simplify and speed-up calculations, we can reduce the dimensionality of the problem from 3D by declaring one or more dimensions **no-size**. Of course, it is an error to declare this way all three dimensions. Finally, it is mandatory to define at least one k-point to populate the *k-points* list. If the list is left empty, an attempt to run the simulation will throw an error.

5.2.2 – Describing geometry

The photonic crystal geometry is described by the *lattice-geometry* variable and the *geometry* list. The *lattice* object has the properties shown in Table III that describe the computational cell configuration. In Fig. 18 is illustrated the resulting vector space.

Table III – lattice object properties

lattice class
▪ **basis*N***, *vector*: where N is 1,2 or 3. 3D vectors forming the basis of the cell vector space. Each one refers to the corresponding space direction. They do not need to be orthogonal or normalized, but they must be linearly independent. If not specified, they take the Cartesian $\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}}$ directions. ▪ **basis-size**, *vector*: the length of each basis vector. All space units are normalized with respect to them. By default, they are unity. ▪ **size**, *vector*: defines the size of the primitive cell along each direction.

When specifying the lattice vectors we must avoid degenerate cases. For instance, it is not valid to use a null vector, or a linearly dependent vector. Furthermore, basis vectors can be given un-normalized, the software will internally normalize them before starting the computations.

In respect to basis normalization and cell size, we have two controlling properties: *basis-size* and *size*. As basis vectors are internally normalized to unit-length, the actual length of each vector is given by the *basis-size* property. It is a 3D vector, where each component corresponds consecutively to each *basisN*. All spatial dimensions are later normalized to *basis-size*, in particular, scalar lengths are scaled by the first dimension.

As *basis-size* controls the lengths of the space vector basis, *size* controls the extents of the primitive cell. It is a 3D vector which each element define the lattice spacing along each direction in the basis. It has no effect other than determining the periodicity spacing.

For both parameters, any value not set will take the default value 1. Similarly, if one element takes the special value **no-size**, then the problem dimension will be reduced, assuming that the dielectric function is constant along those directions. Thus this way, it is possible to simulate 1D or 2D structures apart from the full 3D crystals.

Finally, the origin of the vector space is centred on the primitive cell.

$$\text{basisN}: \mathbf{b}_N = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\text{basis-size}: \mathbf{s} = \begin{bmatrix} s_1 & s_2 & s_3 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} s_1\bar{\mathbf{b}}_1 & s_2\bar{\mathbf{b}}_2 & s_3\bar{\mathbf{b}}_3 \end{bmatrix}$$

$$\text{size}: \mathbf{l} = \begin{bmatrix} l_1 & l_2 & l_3 \end{bmatrix}$$

$$s_3 = \texttt{no-size}$$

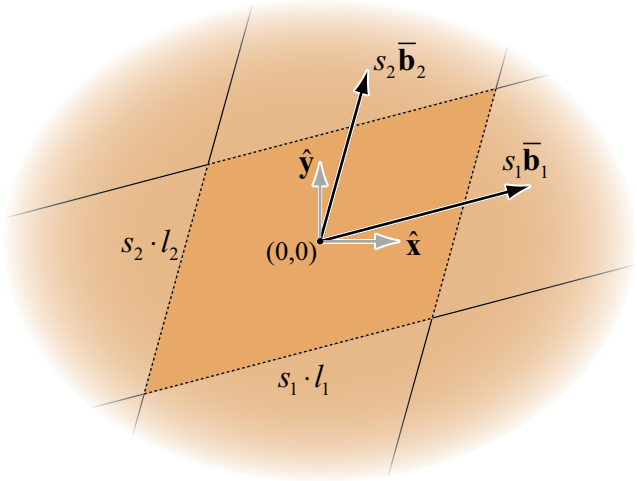


Fig. 18 – Description of the computational cell by the lattice **object. In the figure is shown a 2D lattice by specifying the third element of** basis-size **to be** no-size**. The primitive cell is shown in darker shading showing actual size.**

The geometry list is composed of *geometric-object* class objects. This is the base class for all other geometric objects used, thus the common properties for all geometry objects are inherited from *geometric-object* and must be specified, as they have no default value. They are described in Table IV.

Table IV – geometric-object object properties

geometric-object class
▪ **center**, vector: the object's centre position. ▪ **material**, material-type: the material of the object

The useful geometric objects are the ones cited above: *sphere*, *cylinder*, *cone*, *block* and *ellipsoid* and are described in the following tables. The program only defines these classes and *compound-geometric-object*, that accepts a list of geometric objects and groups them. Unfortunately, the latter is not usable, as any attempt will cause an error and the program will abort.

Aside from that, to describe complex object structures, the user has very limited options. The objects are defined in the *libctl* library, which also includes all code related to object searching, and basis fixing. Any attempt to define a new geometric object class needs to modify the *libctl* source code. Furthermore, the MPB source will also need to be modified to note the existence of the new class.

Nevertheless, this task will require compiling and make the local version incompatible with the standard version. For these reasons and others, modifying the source is discouraged. However, there is a workaround that allows to define a new class without the need to tamper with *libctl* and will be commented latter.

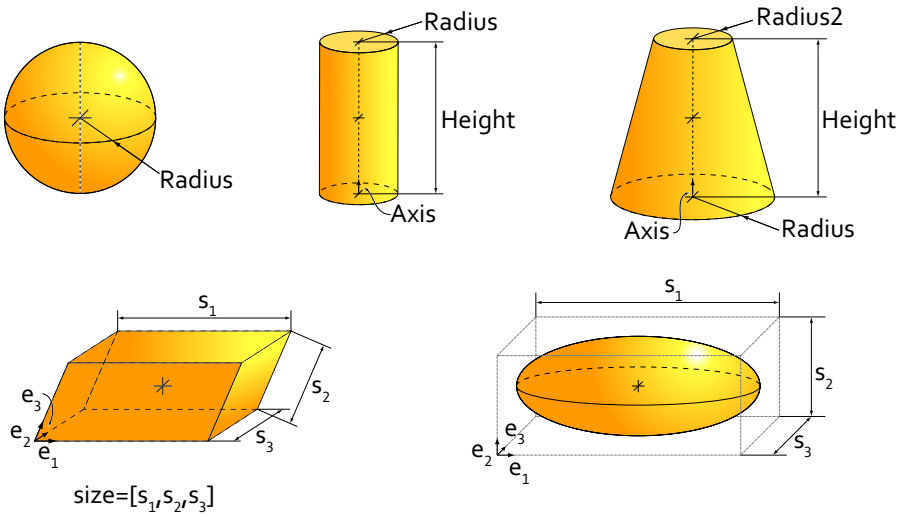


Fig. 19 – The different geometric objects defined in MPB. A cross marks the centre of each object. The parameters for each construction are shown.

Table V – shpere object properties

sphere class ← *geometric-object*
▪ **radius**, *number*: the radius of the sphere. It must be set.

Table VI – cylinder object properties

cylinder class ← *geometric-object*
▪ **radius**, *number*: radius of the cylinder cross section. It must be specified. ▪ **height**, *number*: the height of the cylinder. Has no default. ▪ **axis**, *vector*: a vector giving the direction of the cylinder axis. Its length is unimportant. By default, it is Z-axis.

The axis property is specified in the lattice basis. This has implications that are discussed later on, for the block class.

Table VII – cone object properties

cone class ← *cylinder*
▪ **radius2**, *number*: the radius of the tip of the cone. It defaults to 0.

It inherits from cylinder. In this case, the **radius** property gives the radius of the base of the cone. The tip is in the positive direction of axis.

Table VIII – block object properties

block class ← *geometric-object*
▪ **eN**, *vector*: where N is 1,2 or 3. Three 3D vectors describing the axes of the parallelepiped. Their lengths are ignored and they must be linearly independent. They default to the basis vectors. ▪ **size**, *vector*: this vector sets the lengths of each of the block axes. Each value is a scalar related to the corresponding **eN** property. Their default value is unity and they must not take a value of zero.

As indicated, the block axes are defined to respect the lattice directions. This implies that if the lattice is not orthogonal or has a different orientation, the block **will transform** according to the lattice. This means that a *square* will be square in the lattice basis, but in the Cartesian basis, it will be distorted.

This behaviour is in contrast to the radius property of the *sphere*, *cylinder* and *cone* classes. In those cases, (great) circles will be so in the Cartesian basis, while they will be warped in the lattice basis.

Table IX – ellipsoid object properties

ellipsoid class ← *block*
no new properties defined

This class simply defines an ellipsoid enclosed by a block. Thus, there are no new properties for it.

As with the block, its axes are defined according to the lattice basis and the same behaviour will be observed. This can be illustrated by comparing the results of defining a sphere both using the sphere class and the ellipsoid class in a non-orthogonal basis, shown in Fig. 20.

5.2.3 – Libctl class system. Defining a new geometric class object

As already commented, the *libctl* library provisions the necessary methods to declare new classes in Scheme, but the class code must be directly programmed into the source code of the library, and then recompiled to a local specialized module replacing the original one.

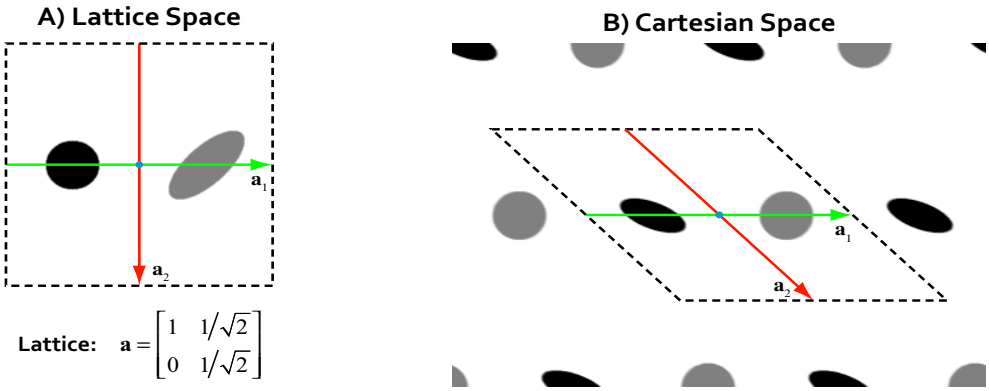


Fig. 20 – Epsilon function output by MPB. There are two spheres defined, the grey one with the sphere **object, and the black one with the** ellipsoid **object, in a non-square lattice. In A) is shown the output produced by MPB, where the unit cell is uniformly sampled in the coordinate system defined by the lattice basis. In B) is shown the actual Cartesian dielectric function. In the image it can be appreciated how both geometric constructs behave differently subject to the lattice basis.**

Nevertheless, the Scheme functions to define new classes can be used to declare a class without bindings to the C interface of *libctl*. The functions of interest are the following:

```
(define-class name parent [ properties... ])
(define-property name default-value type [ constraints... ])
(define-derived-property name type derive-func)
(define-post-processed-property name default-value type
        process-func [ constraints... ])
(object-property-value object property-name)
```

libctl Scheme objects are completely passive classes. They do not provide methods to perform actions, only accessor methods to the class properties. libctl does not implement a full class system, as its initial development was tightly bound to that of MPB, it was tailored to its specific needs. For instance, all properties in a class are accessible and are defined in a global namespace.

To declare a new class, we must give a unique name for it in the current Scheme environment. If done otherwise, the old definition will be overwritten and most probably, the program will not operate correctly. Classes have a simple single parent inheritance system. All properties from the parent will be inherited, and overridden if necessary. A class may have no base parent class by setting the *parent* parameter the special value **no-parent**.

A class may have zero or more properties defined. Properties are declared by any of the three *define-*-property* methods. The name of the property is mandatory and may already be defined. The next mandatory field is the *type* of the property. It is a symbol of an existing type declarator, any of the following list:

'number	**'cnumber**	**'integer**	**'vector3**
'matrix3x3	**'cvector3**	**'cmatrix3x3**	**'boolean**
'string	**'function**	**'class**	**'SCM**

‘(make-list-type el-type)

As stated, the property identifier may be reused across several classes. However, there is the restriction that they must share the same type. The user must also be careful when defining other variable in Scheme. As pointed out, the type name is defined in the calling environment, thus any redefinition of the identifier will most surely create errors with the classes using them.

A property may have a default value if the user specifies one. If this field is given the special keyword **no-default**, that forces the user to set this property at object instantiation time.

Properties may have a set of validating functions in the *constraints* sequence. Those are single parameter functions that receive the user input and result in a boolean telling if the input is acceptable.

A derived property is one that is not set by the user at instancing time, but automatically calculated after setting the other non-derived properties. For this reason, there is no need for a *default-value*, and it is an error if the user sets it. A derived property has a function that takes the current object as its sole parameter and outputs the calculated value of type *type*. The calling order for several derived properties is unspecified, and could cause dependency problems.

A post-processed property is similar to a derived one. It has a function that takes the user input as a parameter, and outputs a value based on in. In contrast to the derived case, instead of waiting after all properties are defined before evaluating it is executed instantly, and it does not have direct knowledge of the object instance it is.

We exploit this to define a new class. For instance, we may want to declare a class to create a prism of arbitrary cross section. An example is given to define a rounded square cylinder in the Z-axis.

The key is using a *derived-property* overloading the default *material* property from *geometric-object*. As required by *libctl*, the trick is maintaining the type the same as the original definition to avoid problems, as the *derived-func* is private for the class definition.

In this way, it is possible to use the *block* geometric class code use the material function returned at instantiation time. This is equivalent to define a block object normally and setting its material type to a material function explicitly. The advantage of such approach is the encapsulation to the end user of the use of the material function, by providing a more recognisable name and cleanliness.

We cannot descend directly from *geometric-object*, as this class has no code implemented, and we would end in a non-functional class. Similarly, using other classes, for example *cylinder*, has also some considerations to care about. Those classes define an implicit boundary for the *material-function* we return, thus it only evaluates inside the volume originally defined by the base class. This prompts our choice for the *block* class as the parent class.

This has its drawbacks, though. The code is defined in Scheme, and that means that the C portion of the block class code will be making repeated calls to the Scheme

environment incurring in a great time penalty. Also for the example given, the returned material may be anything but the special *nothing*. If *nothing* is returned, an error will abort computation. In fact, these are a limitation of the C implementation of MPB and the *libctl* library, and not of the solution presented.

```
(define-class squircle-cyl block
  (define-property roundness no-default 'number)
  (define-derived-property material 'material-type
    (lambda (me) (make material-function (material-func
      (lambda (position)
        (let*
          ((center    (object-property-value me 'center))
           (size      (object-property-value me 'size))
           (roundness (object-property-value me 'roundness))
           (mat       (object-property-value me 'material))
           (x         (- (vector3-x position) (vector3-x
center)))
           (y         (- (vector3-y position) (vector3-y
center)))
           (a (/ x (vector3-x size) 0.5))
           (b (/ y (vector3-y size) 0.5))
           (n (+ 1 roundness)) )
          (if (< (+ (expt (abs a) n) (expt (abs b) n) ) 1)
            mat
            default-material; cannot use nothing ...
          )
        )
      )
    )))
  )
)
```

Listing 1 – Definition of a new geometric class in Shceme, with no C bindings.

5.2.4 – Materials

Materials in MPB are defined by classes descendant from *material-type*. The base class has no default properties, and as in the *geometric-object* class case, it is not useful for direct use.

MPB defines three different material classes: *dielectric*, *dielectric-anisotropic* and *material-function*. They are described in the following tables.

Table X – Delectric object properties

dielectric class ← *material-type*
▪ **epsilon**, *number*: the dielectric constant of the medium. Must be specified a positive number.

The *dielectric* class defines an homogeneous isotropic media, characterized by its *relative epsilon* constant, ε_r. Alternatively to *epsilon*, we can give the refraction index *n* by using *index*, which is an alias to `(epsilon (* n n))`.

Table XI – Dielectric-anisotropic object properties

dielectric-anisotropic class ← *material-type*
▪ **epsilon-diag**, *vector*: the tuple *a,b,c*. They correspond to the principal axis constants. They must be given. ▪ **epsilon-offdiag**, *complex vector*: the tuple *u,v,w*. If given complex values, we can define *gyrotropic* materials. It has a default value of zero. ▪ **epsilon-offdiag-imag**, *vector*: They represent the imaginary part of the *u,v,w* elements. It is deprecated in favour of a complex *epsilon-offdiag*. It is null by default.

This class defines an homogeneous, possibly anisotropic, material defined by the dielectric tensor:

$$\varepsilon = \begin{pmatrix} a & u & v \\ u^* & b & w \\ v^* & w^* & c \end{pmatrix}$$

This class allows to define materials with complex epsilon, but not dispersive materials, as the main diagonal elements must be real numbers.

Table XII – Material-function object properties

material-function class ← *material-type*
▪ **material-func**, *function*: a function taking a 3D vector for position as input and returning a *material-type* class object. It has no default and must be set.

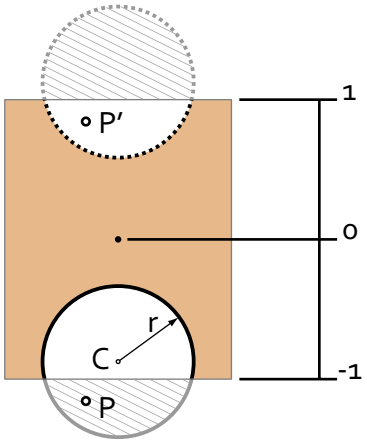
This class allows to define a very general media. Any geometric object having this material class, will be probed at all grid points inside the volume enclosed by it, and this function evaluated. The parameter the function expects, is a single 3D vector containing the absolute position of the object's point inside the computational cell.

A very interesting feature of this class is the ability to return a generic *material-type* object, which will be recursively evaluated. This means that several *material-function* instances can be nested to form very complex dielectric distributions.

One serious drawback of using this type of material, is it slowness. As each point belonging to a geometric object is tested, the call stack must go all the way from the

libctl C code to the Scheme code, making it very time expensive. Moreover, some points may be tested several times due to the algorithm implementation.

There are other delicate points to take into account when using this class. One very serious is the fact that, if an object crosses the cell boundary and the crystal is periodic—it is by default,—the object will be wrapped around. The problem arises because the point passed to the function is in **absolute coordinates inside the cell**. That is illustrated in Fig. 21. As there is no direct way of determining that the object has wrapped, and where it did, the function most probably will return incorrect values.



$$f(x)=\begin{cases}1 & \|x\|<r\\ 0 & \text{otherwise}\end{cases}$$

$$\overline{P}=P-C;\overline{P'}=P'-C$$
$$f(\overline{P})=1$$
$$f(\overline{P'})=0$$

Fig. 21 – Point P and P' lay on the same object, but as MPB returns P', the function returns an incorrect value. There is no chance to correct the situation, as the function has no knowledge of the wrapping.

The last point to consider is the use of the **nothing** material as a return value. Even though it is a valid material, its special behaviour is not well tolerated. An examination of the C code of *libctl* dealing with determining the material, explicitly throws an error if it is used in a material function. This contravenes the definition of this material and of the class.

As a final note on the material types supported by MPB, currently there is no support for dispersive—metallic—and non-linear materials. The *material-function* class is not practical for non linear materials, as the dielectric function is evaluated just once at the beginning of the computation. Similarly dispersive materials with high losses, require very high discretization, yielding long simulation times. Even more, accuracy and convergence are negatively affected.

5.2.5 – Simulation

Once the simulation environment is et up, we are ready to perform the simulation. Typically, band calculation is started by the **run functions**. These functions load the input variables and prepare the C environment. With the run functions we can separate the simulation of different parities in even and odd modes for the Y and Z direction, or perform a general simulation with no separation. The run functions take several forms:

```
(run [ band-function ])
(runte [ band-function ])
(runtm [ band-function ])
…
(run-parity parity reset-fields [ band-function ])
```

The last type is the more general case, and it will not normally be used, in favour of the more convenient functions above. When simulating different parities in the same job, we need one **run** function per parity. Each time they are called, the simulator is reset and calculations started form zero. This can be avoided setting the *reset-fields* boolean parameter to false, but in the occasions we used it, the program generated errors.

In the 2D cases we are simulating, were the dielectric function is constant along the Z direction, separating parities in the Z plane is trivial and correspond the TE and TM polarizations. Nevertheless, this is not the case when simulating separate Y parities. There are some hard restrictions as the manual states:

```
You should use these functions only for structures that are
symmetric through the y=0 mirror plane, where the second basis
vector is in the y direction (0,1,0) and is orthogonal to the
other two basis vectors, and when the k vectors are in the xz
plane.
```

This effectively limits the "Y symmetry" to the Cartesian Y axis. For our purposes, this is a severe limitation, as we are interested in determining the symmetries at different, oblique, planes that do not necessarily lay on the XZ plane.

We dealt with this problem using a non-minimal cell. In this cell, we rotated the geometry such as to have the direction of interest oriented along the Cartesian X axis. As we just rotated geometry as a whole, this is equivalent as just defining a new, rotated, cell; thus all point group symmetries are preserved. Nevertheless, the reciprocal lattice changes, therefore we need to recalculate the Irreducible Brillouin Zone, and the k-points of interest.

In particular, we have applied this workaround to simulate separately TE and TM polarizations on the ΓM direction of a square lattice. The results are shown in Fig. 22.

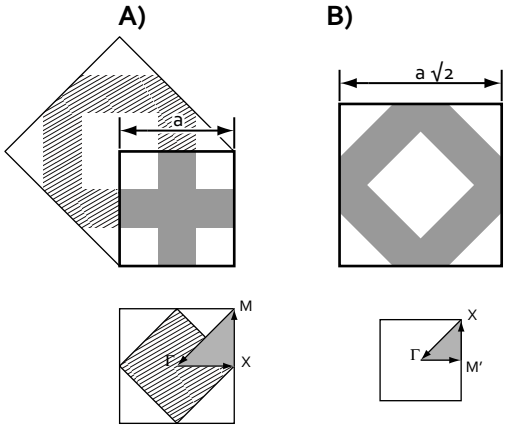


Fig. 22 – Cell definition for oblique plane parity separation. In A) is shown the standard cell definition for a square lattice. The new, rotated, cell is shown hatched. This new cell is not minimal as it is larger. In B) is shown the new geometry definition based on the rotated cell; we rotate the geometry to get a default-oriented square cell. Bellow are shown the Brillouin zone for these cells. In A) the BZ for the rotated cell is shown hatched, showing the correspondence with the standard BZ. For the B) case, the BZ is the rotated version of the former. We can see how the direction of interest ΓM' now lays on the XZ plane.

As can be seen from Fig. 22, the new Brilluoin zone differs from the original one. This has the effect that the folding points of the reciprocal space change accordingly.

For the specific case depicted, of the square lattice in the ΓM direction, we highlight several points.

We first notice that the cell basis size has changed by a $\sqrt{2}$ factor. This is not strictly necessary, as all dimensions are normalized, but it helps when comparing the results obtained with those of the standard cell definition. We do not need to deal with correction factors when performing numerical analysis.

The folding issue is the next important effect to consider. As we can appreciate, the **M'** point lays midway of **ΓM**. This has the consequence that the bands along this direction will fold at **M'**. The goal is to separate the bands by Y symmetry, hence we need to unfold them. This is not a trivial operation, as we have not direct information of which points correspond to a folded band.

The technique used to obtain the unfolded bands, extends the **ΓM'** to **ΓM'm**, where **m** is the equivalent to point **M** in the standard cell and then perform a point-by-point comparison of a simulation run done with the standard cell and no symmetry discrimination. We pick the points that are coincident on both datasets to get the bands at **ΓM** direction with their corresponding oblique symmetry.

This can be done either in the same simulation run by specifying a k-point list accordingly, or during numerical analysis by taking the **ΓM'** data segment and replicating it with a mirror operation at the **M'** point. Doing this we are effectively doubling the bands and we are now able to perform this comparison.

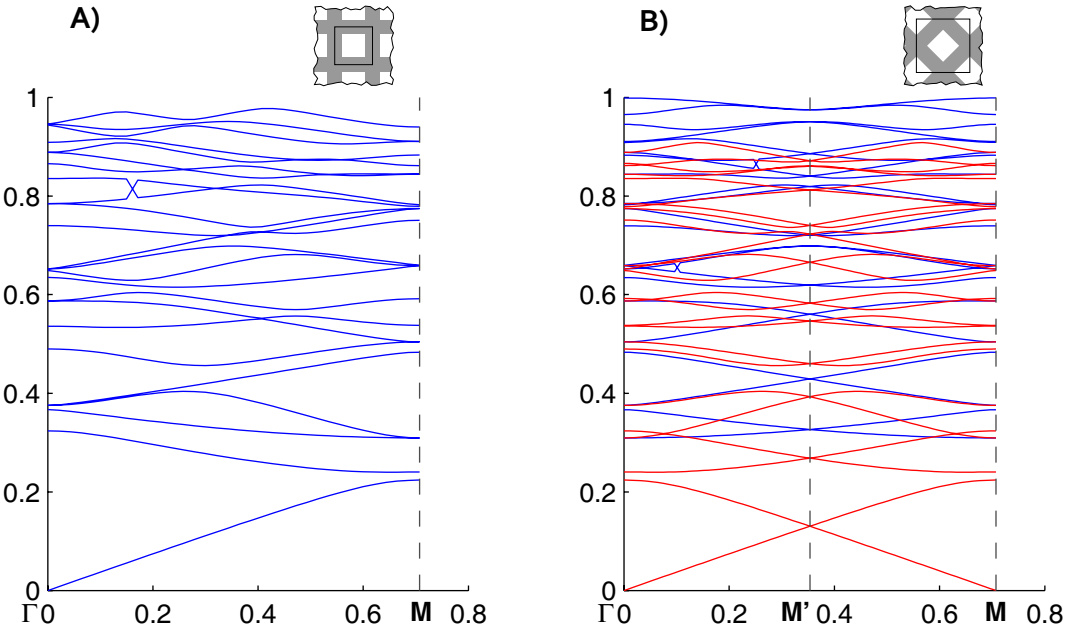


Fig. 23 – Band folding effect for non-minimal cell. In A) is shown the ΓM direction for the standard square cell. Compare this to B) where we have used the rotated cell. We can appreciate how at M' there is a folding line but the bands are conserved. We have separated the Y symmetries at the cost of adding a folding. Separating the folded bands is easier as it can be done in post-processing, while MPB offers no possibility to calculate the Y symmetries for oblique planes.

5.2.6 – Simulation Output

MPB always generates a log of the current simulation progress on the standard output. In this log, MPB outputs the initial geometry, initialization time and lattice configuration. It also writes the current k-point simulated, along with the iterations required to converge and the time it took. Once finished the current k-point it writes the eigenvalues found in *comma separated* format:

```
xfreqs:, k index, k1, k2, k3, kmag/2pi, x band 1, … , x band N
```

where, excluding the first entry, all values are numeric. Depending on the simulation, **x** may be empty or the symmetry used—e.g. *te*, *tm*, *yodd* or *teyeven*. At the simulation end, it writes a summary of the bands, their span, and their maxima and minima:

```
Band N range: min at #(k-point_min) to max at #(k-point_max)
```

Band data is ordered from minimum to maximum, thus, lower band number corresponds to lower eigenvalue. This can present in some situations a problem as bands may cross each other, and separating them is no trivial task.

MPB may also find band-gaps between bands, provided they do not cross at any of the simulated k-points. The gap width is calculated relative to the central frequency of the gap. Owing to the commented ordering issue, it may happen that some band-gaps are erroneously calculated, as bands may not have the same polarization. Moreover, Small *width* values are suspect of numerical errors. For these reasons, it is always advisable to visually inspect the band diagram and discard false positives.

```
Gap from band N (value) to band M (value), relative_width%
```

The MPB programming interface provides the user with additional tools to generate supplementary information. The most useful functions are the **band-functions**. These are a list of functions provided to the *run* function. The *run* function then calls in succession each of the functions in this list after computing a *k-point*. **Band-functions** may receive one or no parameters; if the function does not receive a parameter, then it is only called once after the *k-point* is calculated. Conversely, if it receives a parameter, then it is called once per band found after the *k-point*, where the parameter is the index of the band.

There are several run-functions defined in MPB, though the user may implement his own. The defined functions basically output the fields corresponding to each mode inside the primitive cell. They generate HDF v5 files with the following naming scheme:

```
field_type.kN.bM.parity.h5
```

where *field_type* is a letter indicating the kind of filed the file contains, e.g. **e** for electric field or **h** for magnetic. **N** and **M** are numbers indicating the current index of the actual k-point and band list respectively, and *parity* shows the actual simulated parity.

5.2.7 – Other capabilities

Aside from the discussed functions and methods, MPB provides further utility functions not described here. Many of these functions are related to the *libctl* library.

For a complete reference, check the on-line MPB documentation.

5.3 – Meep

The *MIT Electromagnetic Equation Propagation* (**Meep**) is a free *Finite-Difference Time-Domain* (**FDTD**) solver for computing the propagation of electromagnetic waves. It was developed by the same group at MIT, and share many characteristics in common.

As MPB, Meep is a text mode program that runs on UNIX environments and has the ability to be multi-processor compatible using the MPI interface.

Owing to the strong resemblances of both programs, we will simply give outlines and point out the differences where necessary. This is a quick summary and further information can be found on the Meep manual in the Internet.

5.3.1 – Time-domain Simulation

In contrast to MPB, this is a simulator in time domain. The goal of such simulations is not finding the steady state solutions of an electromagnetic wave, but the actual wave propagation and its transient response. The simulator takes Maxwell's equations and evolves them over time within some finite computational region, essentially performing a kind of numerical experiment.

Depending on the experiment set-up, we can obtain different information, even steady state modes of a particular geometry. In our case, we are especially interested in obtaining the reflection, and perhaps the transmission, spectrum of macroporous silicon photonic crystals.

Other interesting information that we can extract from such simulations include resonant modes and frequencies, field patterns (Green's functions) in response to an arbitrary source, or the local density of states.

The main drawback of time domain electromagnetic simulations is the time invested and the higher resolutions needed.

5.3.2 – User Interface

As MPB, Meep offers the *scheme* interface using the *Guile* interpreter. As an alternative, it presents a C++ interface, allowing to create a simple C program. The C++ API has the advantage of being flexible and very fast, however, using the C++ API, requires compiling the program and is less user-friendly than the interactive shell provided by *Guile*.

In this project, we use exclusively the *scheme* interface, and for the rest of this section we assume that this is the execution environment.

Aside from common definitions with MPB, listed before, Meep adds some new variables and constants. The most relevant for our needs are detailed below:

- **sources**, *list of source class*: Specifies the current sources to be present in the simulation; defaults to none.

- **symmetries**, *list of symmetry class*: Specifies the spatial (mirror/rotation) symmetries to exploit in the simulation. The symmetries must be obeyed by both the structure and by the sources. By default is empty.

- **pml-layers**, *list of pml class*: Specifies the absorbing PML boundary layers to use; defaults to none.

- **dimensions**, *integer*: When working in Cartesian coordinates, it has the same meaning that in MPB. It sets the *maximum* dimension of the problem, as any **no-size** value of the lattice will reduce dimensionality. If dimensions is 1, then the cell must be along the Z direction and only E_x and H_y field components are permitted. If dimensions is 2, then the cell must be in the XY plane. It can be the special value **CYLINDRICAL**, which uses cylindrical coordinates and X and Z dimensions are interpreted as R and Z, respectively. By default is 3.

- **m**, *number*: For cylindrical simulations, specifies that the angular φ dependence of the fields is of the form $e^{jm\varphi}$; the default is m=0.

- **resolution**, *number*: Specifies the computational grid resolution, in pixels per distance unit. Its minimum value is 10 regardless of user input.

- **k-point**, *false or vector*: In this case we specify either false to deactivate periodic boundary conditions, or a *single k-point* giving the propagation vector of a plane wave in Cartesian coordinates, in units of 2π/distance, By default it is *false*.

- **ensure-periodicity**, *boolean*: If true, and if the boundary conditions are periodic (k-point is not false), then the geometric objects are automatically repeated periodically according to the lattice vectors. By default is true.

- **eps-averaging?**, *boolean*: sets subpixel averaging when initializing the dielectric function. Using this will generate a more accurate, but slower, computation of the average ε, with diminishing returns for the actual FDTD error. It is true by default.

- **extra-materials**, *list of material-type class*: if there is no *standard* geometry objects with dispersive, nonlinear or similar material, Meep will not detect such materials when returned from a **material-function**. In this case, those materials need to be added to this list, which is empty by default.

and the constants:

- **perfect-electric-conductor**, **metal,** *material-type class*: A predefined material type corresponding to a perfect electric conductor, at the boundary of which the parallel electric field is zero

- **perfect-magnetic-conductor**, *material-type class*: A predefined material type corresponding to a perfect magnetic conductor, at the boundary of which the parallel magnetic field is zero.

- **{X**, **Y**, **Z**, **R**, **P}**, *direction constants*: correspond to the directions x, y, z, r, φ, respectively.

- **{High, Low}**, *boundary-side constants*: the boundary direction, + or – the direction.

- **{Ex, Ey, Ez, Er, Ep},**
  {Hx, Hy, Hz, Hr, Hp},
  {Bx, By, Bz, Br, Bp},
  {Dx, Dy, Dz, Dr, Dp},
  {Dielectric, Permeability}, *component constants*: for: Ex, Ey, Ez, Er, Eφ, Hx, Hy, Hz, Hr, Hφ, Bx, By, Bz, Br, Bφ, Dx, Dy, Dz, Dr, Dφ, ε, μ, respectively.

Meep produces two types of outputs, it always writes status and progress information on the *standard output*, and it can generate data files from user commands in the control program. In general, files are output to the *working directory*, which is the directory where Meep was invoked. We can alter this behaviour and set up an alternate *output directory*, where all data files will be stored, with the statement:

```
(use-output-directory [ dir ] )
```

the **dir** parameter is an optional string giving the path for the output, if left blank it will take the default value "*control_program_file_name*-out". Once set-up, all accesses to files using Meep functions will search files in this path.

Data files generated are normally all in HDFv5 format, thus most functions will append the "*.h5*" extensions when accessing files.

5.3.3 – Describing Geometry and Materials

Given that Meep uses the same *libctl* library, geometry is defined in Meep very much like in MPB. As commented, we will mention the specifics of Meep.

We must first point out that in contrast to MPB, Meep can only simulate cells with Cartesian basis. Also, by default the geometry defined is isolated in space, thus we must explicitly say if we ant a periodic repetition of the computation cell.

As MPB, we can simulate in 1D, 2D, and 3D Cartesian dimensions, but Meep offers the possibility of using cylindrical coordinates, for example for coaxial waveguides.

To simplify the problem and speed up computations, we can explicitly define symmetries of the geometry—even/odd mirror symmetries and 90°/180° rotations. This also affects the fields, so care must be taken to ensure that they are also symmetric.

Meep units are also normalized. Distances are normalized as in MPB, by the **size** attribute of the lattice object. Besides, time is also normalized given the relation of frequency and wavelength in vacuum, such that

$$\omega = \omega' \frac{2\pi c}{a}, \quad \tau = \frac{1}{f} = \frac{1}{2\pi\omega} = \frac{c/a}{\omega'}\,[\mathrm{s}^{-1}], \quad \tau' = \frac{1}{f'} = \frac{1}{f\frac{a}{c}} = \frac{1}{\omega'}\,[1] \tag{33}$$

where all primed quantities are the normalized units used in Meep, and a is the normalizing distance factor.

Another remarkable difference in Meep in comparison to MPB is the type of materials that are available. MPB could not simulate non-linear materials, or metals, restricted to the problem proposal. In Meep, however, we can simulate the transient and time dependent characteristics of materials, thus we can define arbitrary anisotropic electric permittivity ε and magnetic permeability μ, along with dispersive ε(ω) and μ(ω), including loss or gain, and nonlinear (Kerr and Pockels effects) dielectric and magnetic materials, and electric or magnetic conductivities σ

5.3.4 – Boundary Conditions

An important distinction in Meep from MPB, is that given that the simulation cell is isolated in space, we must give the boundary conditions of the cell walls. We have several choices. By default, if not otherwise specified, the walls are perfect metallic conductors. This in general will cause unwanted reflections that will bounce indefinitely inside the computation cell. To eliminate this effect and make it like if the cell was effectively surrounded by void, we can add *perfect absorbing layers* to the walls. This type of layer is known as *Perfect Matching Layer* (**PML**) and represents a fictitious material that absorbs all incident radiation and has null reflection.

Achieving the PML condition is difficult for all incidence angles and frequencies and little reflection may occur, thus we must specify a thickness and damping factor for these layers. A thicker layer will reduce the numerical reflections but will increases computation time considerable. Conversely, higher damping factor, will be more absorptive, but reflections will increase.

Any PML is added to the inside of the computational cell, overlapping any geometry without loss of functionality. PML layers are objects that are placed in the list **pml-layers**, defined at the top-level scope; their properties are described in the following table:

Table XIII – pml object properties

pml class
▪ **thickness**, *number*: The spatial thickness of the PML layer. The thinner it is, the more numerical reflections become a problem. No default value. ▪ **direction**, *direction constant*: Specify the direction of the boundaries to put the PML layers next to. Default is the special value ALL, which puts PML layers on the boundaries in all directions. ▪ **side**, *boundary-side constant*: Specify which side, Low or High of the boundary or boundaries to put PML on. Default is the special value ALL, which puts PML layers on both sides. ▪ **strength**, *number*: A strength to multiply the PML absorption coefficient by. Higher values will make smaller the theoretical asymptotic reflection coefficient of the PML, but will also increase numerical reflections. Alternatively, you can change R-

asymptotic, below. Default is 1.

- **R-asymptotic**, *number*: The asymptotic reflection in the limit of infinite resolution or infinite PML thickness, for reflections from air. The default value is 10−15.

- **pml-profile**, *function*: By default, Meep turns on the PML conductivity quadratically within the PML layer to mitigate reflections due to the discretization. This property allows to use a user defined function for the profile.

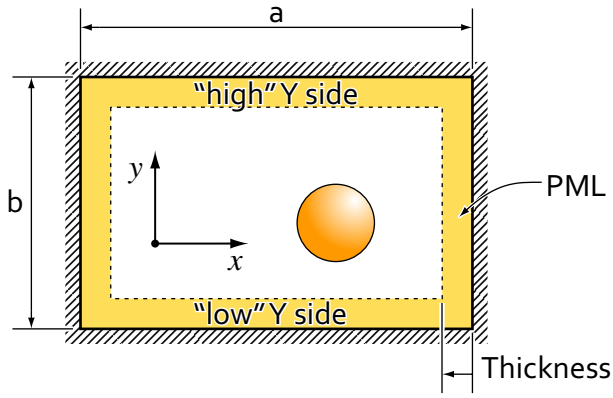


Fig. 24 – PML definition in a cell. PML "grows" inwards the cell, subtracting thickness **from it. In the figure,** a **and** b **represent the 2D cell dimension; a PML is layered on all surfaces, occupying a total of 2**×thickness**.**

Of all the PML properties, we really use only the **thickness** and **direction**, and leave the others to their default values. This is both for convenience and because they already suit our needs.

Until here, these boundary conditions fully enclose the cell, isolating it from space, as we cannot simulate it. On the other hand, for an infinitely extended crystal, we have already seen that symmetry operations extend a finite region to all the space. This suggests that we can apply boundary conditions that follow this symmetry, allowing the calculation of the fields in a finite region that extend to the whole crystal, in the same way as MPB.

To define periodic boundary conditions we need to give a valid *k-point* for the propagation vector. This vector represent a plane wave, thus fields are repeated along this vector. When using this type of boundary, we must be careful that the geometry is also periodic; otherwise, we could obtain erroneous results. It is also important to pay attention to any symmetry simplifications, it is easy to make a mistake and cancel the fields.

This type of boundary is especially important for crating plane wave sources.

5.3.5 – Sources

A major difference with MPB is that in Meep we need to define excitation sources. The MPB solver was based in *plane wave decomposition*, therefore the excitation was

already known, plane waves from an infinite distant source, and it was only needed to give the propagation vector of the wave.

In Meep the situation changes drastically. We cannot have an infinite distant source because it would take infinite time for it to propagate and arrive to our geometry. We could however apply the **equivalence** property of electromagnetic waves in a homogeneous media, and define the equivalent sources or fields in the cell boundaries. Nevertheless this is limited by Meep, as we can only define current sources, with hard geometric and polarization constraints[1].

As geometry, sources are classes based on the *libctl* interface. Meep has three source classes that actually define an excitation: *continuous-src*, *gaussian-src*, and *custom-src*, and class *source* that defines the geometric extent of the source. In the following tables are described their properties.

Table XIV – source object properties

source class
▪ **src**, *src-time class*: specifies the time dependence of the source. There is no default, and must be one of the classes defined below. ▪ **component**, *component-constant*: gives the direction and type of the current component. Valid values are **Ex**, **Ey**, **Ez**, **Er**, **Ep**, **Hx**, **Hy**, **Hz**, **Hr**, **Hp**. Must be set. ▪ **center**, *vector*: The location of the centre of the current source in the computational cell; no default. ▪ **size**, *vector*: The size of the current distribution along each direction of the computational cell. The default is (0,0,0): a point source. ▪ **amplitude**, *complex number*: The overall, complex, amplitude of the current source. The scale factor is 1.0 by default. ▪ **amp-func**, *function*: A Scheme function of a single argument, that takes a 3D vector giving a position and returns a (complex) current amplitude for that point. This is an optional property and if not given returns 1.

This class describes the spatial distribution of the currents for the source. The time dependant part of the source is delegated to a second object of class *src-time*.

When specifying the type of source, we give the current type and orientation, which is limited to the presented list, to either electric or magnetic currents along the Cartesian axis. Any source can only have one current definition, thus for polarized

1 The developers claim that in a future version of Meep, it will be possible also to define field distributions over a region.

sources, we need to duplicate the source definition changing the current component accordingly.

We can create volume, plane, or point sources, by setting one or more values of the **size** triad to zero. Nevertheless, we can only have sources oriented as the default axes.

Finally, the source can have a spatially dependant function to define the distribution of the currents in its volume. This function takes a single argument holding the relative position of the point in the source volume being evaluated and returns a single, possible complex, scalar value.

By using a volume source and a amplitude function we are able to define sources of any shape, by simply masking out—null amplitude—the points outside our irregular shaped source. Nevertheless, as the case with custom defined material function, this is a very slow solution when using the *scheme* interface, to the point of being worthless for large problems. It speeds up considerable using the C++ interface, but at a greater complexity cost.

Table XV – continuous-src object properties

continuous-src class ← *src-time*
▪ **frequency**, *number*: the frequency ω, in units of 2πc/distance; thus 1/ω in these units is the vacuum wavelength or the temporal period. Must be specified. ▪ **start-time**, *number*: the starting time for the source; by default is 0. ▪ **end-time**, *number*: the end time for the source; default is infinity, thus it never turns off. ▪ **width**, *number*: the temporal width of the smoothing. By default, there is no smoothing: value 0. ▪ **cutoff**, *number*: How many widths the current decays before the source is turned off and set it zero; the default is 3.

This class defines a harmonic, constant frequency source. The **frequency** property can be set with the alternative expressions `(wavelength x)` or `(period x)`, which are synonyms of `(frequency (/ 1 x))`. The source has the expression:

$$s(t)\mathrm{e}^{-j\omega t}, \tag{34}$$

where $s(t)$ deals with the starting, stopping and smoothing of the source.

By default the source starts and finishes "instantly", this means that a large amount of high frequency components appear in the transient. To eliminate these unwanted components we can act on the **width** and **cutoff** parameters. The first one sets the temporal width of the smoothing; the rate of change for the currents, while a larger value of **cutoff** will reduce the difference between zero and the current value. This way,

the amount of high-frequency components that are introduced by the starting/stopping of the source are diminished, but will lead to longer simulation times. For the **width** parameter, you can instead specify `(fwidth x)`, which is a synonym for `(width (/ 1 x))`.

Table XVI – gaussian-src object properties

gaussian-src class ← *src-time*
▪ **frequency**, *number*: The centre frequency ω, in units of 2πc. No default value. ▪ **width**, *number*: The width *w* used in the Gaussian. It has no default value. ▪ **start-time**, *number*: The starting time for the source; default is to turn on at t = 0. ▪ **cutoff**, *number*: How many widths the current decays before we cut it off and set it to zero—this applies for both turn-on and turn-off of the pulse. The default is 5.

This class defines a Gaussian shaped pulse source. The frequency is varied following a Gaussian curve:

$$s(t)\,\mathrm{e}^{-j\omega t-\left(\frac{t-t_0}{2w}\right)^2}, \tag{35}$$

where $s(t)$ deals with the starting, stopping and smoothing of the source, w is the **width** parameter, and t_0 depends on the **width**, **start-time** and **cutoff** parameters.

For the **width** parameter we can instead specify `(fwidth x)`, which is a synonym for `(width (/ 1 x))`, that is, the frequency width is proportional to the inverse of the temporal width.

Table XVII – custom-src object properties

custom-src class ← *src-time*
▪ **src-func**, *function*: The function f(t) specifying the time-dependence of the source. It should take one argument and return a complex number. ▪ **start-time**, *number*: The starting time for the source; default is (- infinity) but the simulation starts at t = 0 with zero fields as the initial condition, so there is always an implicit sharp turn-on at t = 0. ▪ **end-time**, *number*: The end time for the source; default is infinity, so it never turns off.

This class allows to define a source with arbitrary time dependence.

5.3.6 – Field Fluxes

Obtaining the fluxes of the wave is essential to calculate the reflection and transmission coefficients. To obtain the spectrum of the fields in a region we must set up the volume to consider. Meep uses the Fourier transform of the time-variant fields to find the flux in that zone.

In Meep we can define several regions where to compute the fluxes that traverse it. To add a flux, we use the invocation:

(**add-flux** fcen df nfreq [*flux-region_class* …])

this statement returns a *flux* class object, which can be saved to a variable for later use. The parameter **fcen** is the central frequency of the calculated spectrum, **df** is the frequency span, and **nfreq** the number of points in the computed spectra. The created object returns the spectra at the indicated regions from *fcen–df*/2 to *fcen+df*/2.

A *flux-region* object defines the volume where we want to compute the fields' fluxes; it has the properties:

Table XVIII – flux-region object properties

flux-region class
▪ **center**, *vector*: The centre of the flux region; it has no default. ▪ **size**, *vector*: The size of the flux region along each of the coordinate axes; default is a single point: (0,0,0). ▪ **direction**, *direction constant*: The direction in which to compute the flux (e.g. X, Y, etcetera). The default is AUTOMATIC, in which the direction is determined by taking the normal direction if the flux region is a plane (or a line, in 2d). If the normal direction is ambiguous (e.g. for a point or volume), then you must specify the direction explicitly, not doing so will lead to an error. ▪ **weight** *complex number*: A weight factor to multiply the flux by when it is computed; default is 1.

Note that the flux is always computed in the positive coordinate direction, although this can effectively be flipped by using a weight of –1; this is useful, for example, if you want to compute the outward flux through a box.

Meep provides a series of functions to manipulate *flux* objects. With them, we can print the values, save and load them from a file, and add or subtract different fluxes; they are presented in the list:

```
(display-fluxes flux …)
(save-flux filename flux)
(load-flux filename flux)
(load-minus-flux filename flux)
```

the first statement prints in *comma separated value* format the values of the computed flux at each frequency point; we can specify more than one flux, whose values will be appended as extra columns. The *load* and *save* operations, take a single flux and they perform the said operation from a file with name *filename*. The actual file will be placed in the output directory and will have the extension "*.h5*" appended. This is important to remember when using the standard file-accessing functions of *Scheme* as they are not aware of this behaviour.

The load functions will replace the data of the flux object *flux*. Noting that during computation values are accumulated, this provides an effective way of adding, or subtracting, a previously saved flux with the current one. To subtract spectra, we use the **load-minus-flux**, which reverses the sign of the loaded flux.

The loading of a flux file will force initialization of the Meep input variables. It is a requirement that both the loaded flux and the current cell definition are compatible. That means that the saved flux had to be generated in a cell with the same definition—size and resolution—and to a flux with the same definition—spectrum characteristics. Even more, in multi-processor environments, the loaded flux has to be generated with the same number of processes. If those restrictions are not observed, there will be an error and computation aborted.

Taking into account that creating a flux region initializes the input variables if needed, the order of operations becomes of great importance. Fluxes should be defined only after geometry and sources have been set, basically, the last definition action to perform.

5.3.7 – Simulation

Once the problem is fully set-up, we invoke the **run functions** to start the simulation. Meep has a specific set of run functions; of which we will present the most relevant to us:

```
(run-until cond?/time [ step-functions … ])
(run-sources [ step-functions … ] )
(run-sources+ cond?/time [ step-functions … ] )
```

The first form runs the simulation until a certain time or condition is reached. The condition test is a function that receives no arguments and returns a boolean, which will stop the simulation returning *true*.

The second statement runs until all the sources are turned off, regardless of the present fields, which may not have dissipated. The last one is a variation of the former. In this case, there is an additional first argument identical to that of the **run-until** statement. In the case of being numeric, it is an additional time to run for after the sources are off, or it is a function. On the other hand, for the test function, the simulation runs until the sources are off and (cond?) returns true. All run-functions call given step functions at each time step.

The condition function can be user defined, but Meep already provides a useful first argument to **`run-sources+`** or **`run-until`**:

```
(stop-when-fields-decayed dT c pt decay-by )
```

which will return true when the fields' absolute value squared has decayed by at least **decay-by** from its maximum previous value. This function will check the fields' **c** component at point **pt**, every **dT** time units. All parameter are positive real numbers, except **pt**, a 3D vector, and **c**, one of the component constants.

5.3.8 – Simulation Output

The output generated by Meep is ruled by the control program passed to it. By default, it generates a log of the computation progress, it first generates a summary of the problem set-up, and afterwards, every four seconds, prints the current time step. At the end, it prints out the total time spent.

Like MPB, we have a range of **step-functions** in the *scheme* interface that output diverse information at our need. As there is now a time component in the simulation, **step-functions** are called at each simulation step. We can alter the temporal behaviour using the **modifier-functions** that Meep provides, or creating customized **step-functions**. The **modifier-functions** we used are described here:

```
(at-every dT [ step-function … ] )
```

this calls its **step-functions** at fixed time intervals, determined by the **dT** parameter.

```
(at-time T [ step-functions … ] )
(at-beginning [ step-functions … ] )
(at-end [ step-functions … ] )
```

those evaluate the **step-functions** only once, at time **T**, the beginning, or the end of the run, respectively.

The output functions are similar to those of MPB, we can basically save files of the dielectric constant, and the fields. Generally, at each time step Meep will either overwrite any generated files, or generate multiple versions for each time. We can control this behaviour with the following modifier-function:

```
(to-appended filename [ step-functions … ] )
```

this creates a file named "*filename.h5*"[1] where the output of the **step-functions** is appended in an added temporal dimension.

1 Note that the function takes the file name without the extension, as Meep adds it automatically.

VI – EXPERIMENTAL

Our first steps into the Photonic Crystals area and the Macroporous Silicon world consisted in understanding the macroporous silicon formation process, the primary technique used in-site, and the parameters involved. To aid in this understanding, we developed some simple macropore structures, using the provided data files and programs also developed in-site.

As other members of the team would already focus on the photonic crystal development, we promptly focused on the characterization and simulation of photonic structures, the ultimate goal of which is the understanding of photonic-crystal macroporous-silicon behaviour and the creation of design procedures for devices based on it.

The characterization and simulation consisted on two simultaneous work lines. On one side, we have the optical characterization at Medium Infrared of the samples produced at our laboratory with the FT-IR equipment at our disposal. The other task line involved computer simulation using the MPB and Meep programs, as well as the TMM method.

6.1 – FT-IR Measures

Measuring the reflection spectrum on the FT-IR is straightforward and has already been discussed. We have done several measures of different samples; which in general consisted on taking angle sweeps from 13° to 87° in 2° increments for acceptable angle resolution. For the measures, we separately took S and P polarizations in the MIR range.

As commented, to make a calibrated measure, we need a reference, for this reason, when taking the sample spectrum, we also take the spectrum of a gold mirror. The FT-IR equipment provided a reference mirror, which was used for the first measures. It was later commented that a gold reference could provide a better response in the MIR range; for this, a piece of silicon bulk coated with deposited gold was used. The comparison showed that they both performed similarly, as shown in Fig. 25. In any case, as the gold mirror was considerably larger, we settled to use it as the reference.

The optical characteristics of the measures were the standard ones described before: 1 cm^{-1} resolution, 2 mm aperture, two-sided–backward-forward scan, full MIR range, 2 zero-filling factor, 3-term Blackmann-Harris apodization, and Mertz phase correction. To reduce noise, as any measure has a certain degree of uncertainty, we performed an averaging of 100 scans per angle for the sample and somewhat less, between 50 and 70, for the reference gold mirror.

The angle-resolved measured spectra, shows certain features, peaks and valleys, that depend on incidence angle and displace accordingly, as shown in Fig. 26, where some spectrums for different incidence angles have been plotted in a stacked fashion.

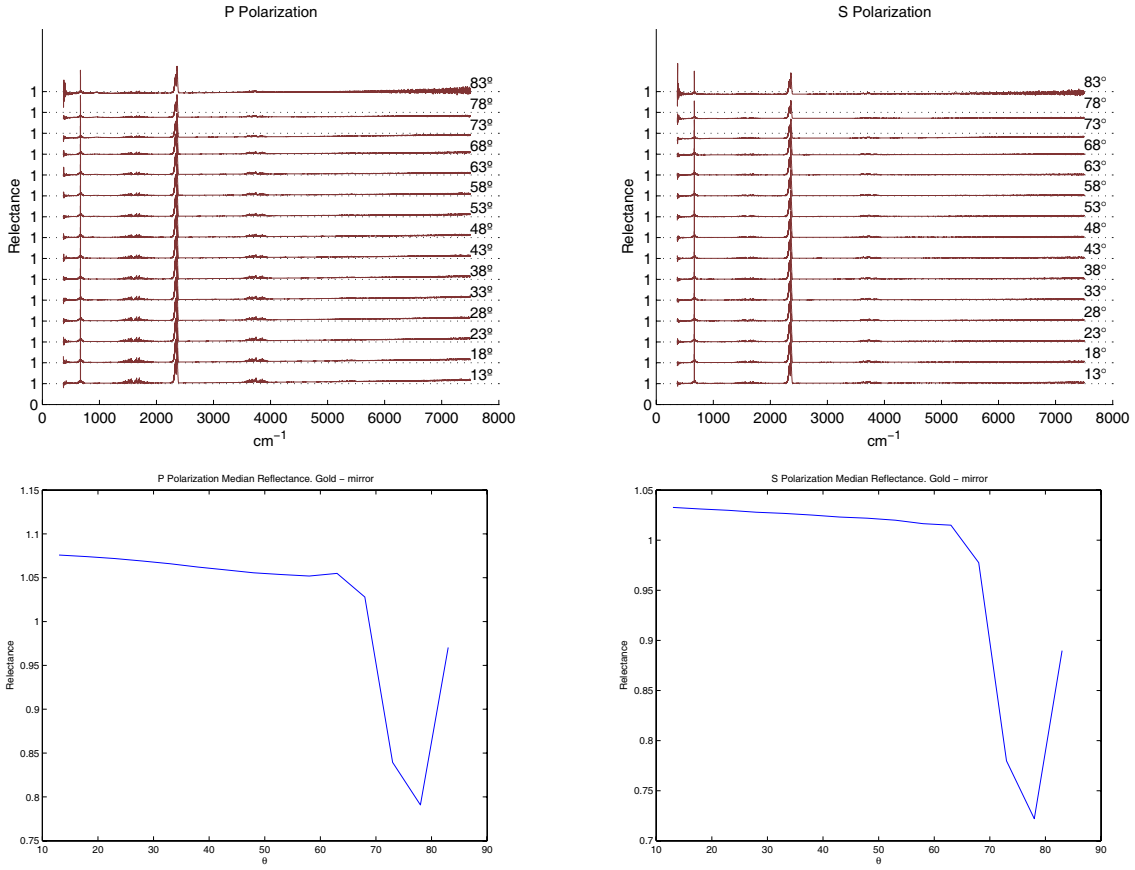


Fig. 25 – Gold mirror reflectance against the reference FT-IR mirror. The graphs show both polarizations. The median reflectance shows that the gold mirror performs slightly better, though it shows a drop in the 70°–80° range.

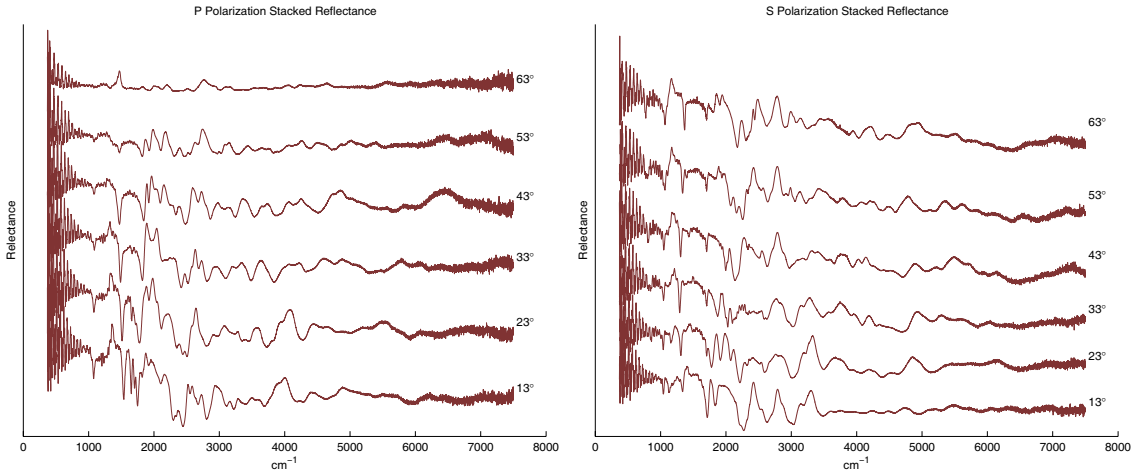


Fig. 26 – Stacked spectra for the sample 2209g. Spectrum is taken in the ΓX direction.

Using the stacked representation does not yield enough information, but we can appreciate that for low frequencies—$\nu < 800$ cm^{-1}, $\lambda > 12.5$ μm— there are oscillations caused by Fabry-Perot resonances, captured thanks to the high resolution of the instrument. We also notice that above $\nu = 5000$ cm^{-1}, noise increases considerably, and making such range of little usefulness, thus from this frequency onwards, we will generally ignore the measured data.

The key point of photonic crystals is that the appearance of a photonic band gap alters the reflection and transmission spectra, and thus its effect must be observable. In effect compared to a simple flat, polished silicon surface, the response of the sample is quite different, see Fig. 27. The task now is relating the response of the measured sample with the theoretical band diagram obtained by simulation with MPB.

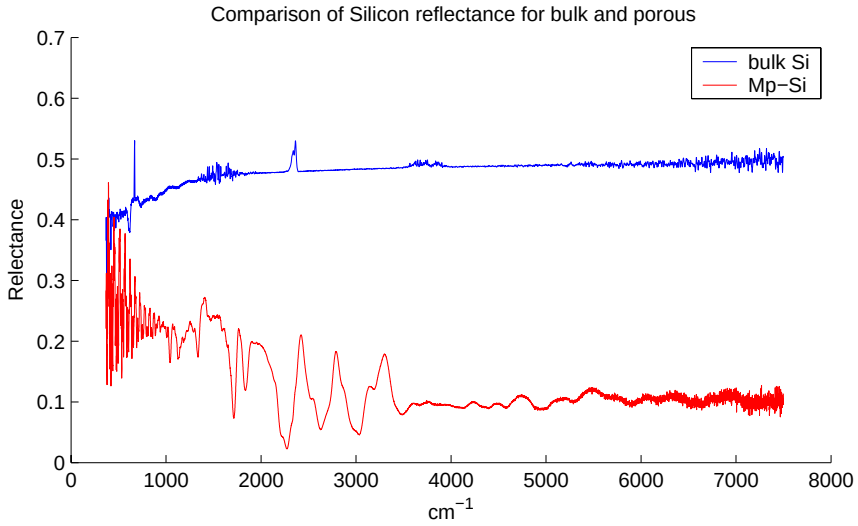


Fig. 27 – Comparison of bulk Si and macroporous Si reflectance (ΓX direction).

We measured diverse macroporous Si photonic crystals; in the following table we show the measured samples and their characteristics.

Table XIX – Measured samples' characteristics.

Sample	Substrate Type	Lattice	Pitch (μm)	Directions	Other
2209g	bulk	square	4	ΓX, ΓM	337 μm
2180l	bulk	triangular	4	ΓJ	259 μm
2207f	Membrane	square	4	ΓX, ΓM	100 μm
2178h	Membrane	square	4	ΓX, ΓM	52.5 μm
2178i	Membrane	square	4	ΓX, ΓM	26.8 μm
2210d *2210h* *2210g* *2210i*	Membrane	square	4	ΓM	Modulated Nickel
2210a					NIR range

6.1.1 – Dispersion Map

From the measured spectra, we build the *Dispersion Map* of the photonic crystal, this is a more informative graph, and allows us to overlap the calculated bands and perform matching. The produced graph is a filled contour map, relating normalized frequency, $\bar{f}$, and parallel propagation vector, $\bar{k}_{\|}$, variables. As the spectrum data is given in frequency, f, by incident angle, θ_i, we need to translate units. The relationship is the following:

$$\mathbf{k} = \mathbf{k}_{\perp} + \mathbf{k}_{\|},$$

$$\|\mathbf{k}\| = \frac{\omega}{c} n \quad \|\mathbf{k}_{\|}\| = \|\mathbf{k}\| \sin\theta_i, \tag{36}$$

as we are incident from air to silicon, and the boundary conditions for electromagnetic fields force the continuity of the tangent components, we obtain,

$$k_{\|}^{\text{air}} = k_{\|}^{\text{Si}} = \frac{\omega}{c}\sin\theta_i = \frac{2\pi f}{c}\sin\theta_i = \frac{2\pi}{a}\bar{f}\sin\theta_i \rightarrow \underline{\bar{k}_{\|} = \bar{f}\sin\theta_i}, \tag{37}$$

where the normalization constant a is the lattice pitch. Thus, it is straightforward to convert from the data space to the *k-vector* space, the frequency variable maps to a line with an sloped dependant on the incidence angle. From this equation we can see that there exists a limiting case of the radiation that we can couple to the photonic crystal when coming from air: $\theta_{i\max} = 90° \rightarrow \bar{k}_{\|} = \bar{f}$. This limit defines what is termed the *light cone*; only the incident radiation inside the light cone can excite a crystal mode. If we wanted to extend the range of the excitable modes, we need a different incoming media or an interface material.

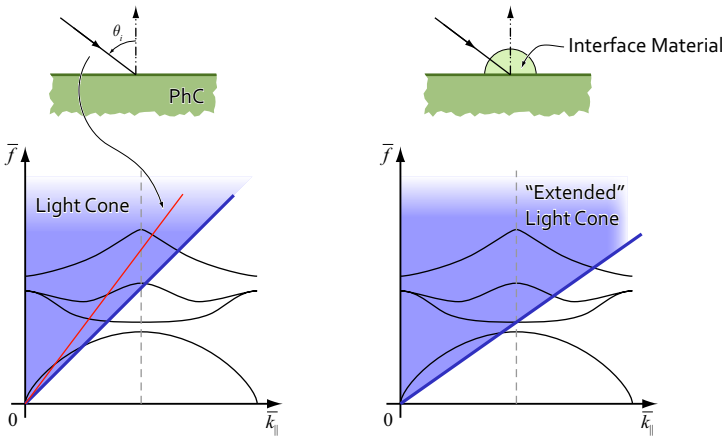


Fig. 28 – Light cone at the interface. A wave coming from a less dense media can only excite those modes inside the light cone. On the other hand, if the wave comes from the denser medium, it will suffer total internal reflection as it cannot couple to the exit media. The "coverage" can be extended using an interface material with different refraction index or geometry. Also shown, is how the spectrum at a certain angle maps to the *k-f* space.

When creating the dispersion maps, it was found that the 1 cm^{-1} resolution was too high, and excess of detail made difficult the band extraction; for this reason, the spectral data was post-processed and down-sampled to a resolution of 8 cm^{-1} or less. This preserves the most notable features, peaks and valleys, due to the photonic band-gaps,

while eliminating the high frequency, small valued detail. This post-processing was performed within OPUS with a specially crafted script.

An effect of this coordinate conversion is that lower frequencies have very high point density, while data spreads thinner as we go up in frequency. This prompts for a resample of the data in the new domain to a square grid, but low data density in the higher frequency ranges result in a bad approximation. In addition, we are not interested in the relative reflectance, we just need to focus on the reflection features, for this reason, we normalise all angle runs, to enhance the feature characteristics.

Data interpolation requires special attention as data points are scattered in a non-uniform way. For speed reasons we perform a two-pass linear interpolation, choosing the destination k and f accordingly to get the square grid. In fact, the f variable is already well defined; however, we reduce the number of points to use more manageable datasets. To further enhance the singular features we perform the derivative of the data along the frequency variable; this step is performed previous to interpolation for convenience reasons.

6.1.2 – Straight Samples

We first measured simple macroporous silicon photonic crystals consisting on near circular cylindrical straight pores. For all samples the lattice constant was 4 μm, and except for sample 2180l, lattice was square.

From literature and our simulations, we know that a square array of straight pores does no present a full band gap. In effect, this can be appreciated in the previous graphics, where reflection is always less than for the bulk silicon but shows peaks and valleys that indicate the presence of crystal electromagnetic modes.

The first sample measured was the *2209g*. This one had 337 μm deep pores of conical section: 2.3 μm at the top, to 1.3 μm at the bottom, etched on bulk silicon. The resulting dispersion map is shown in Fig. 29 and Fig. 30.

It is now clear from the figures, that the photonic crystals present bands, and a photonic band gap can thus be created. In the next subsection, our next task will be verifying that the theoretical model obtained by simulation and the measure coincide.

From these graphs, we can make some remarks. The measured presented considerable noise above $\nu = 5000$ cm^{-1}, thus the data range was clipped as commented before. Additionally, we observed that due to finite layer thickness and pore depth, some high-frequency oscillations where present all over the spectrum. As these ringing is annoying and burdens the analysis, we applied some heavy filtering; this eliminates the unwanted portion at the cost of smearing the interesting features. A good compromise was filtering to an equivalent resolution of 8 or 16 cm^{-1}.

As said, the square lattice of cylindrical pores of sample *2209g* does not present a useful band gap, but there are several non-ideal effects. First, pores are cylindrical, and are not perfectly circular cross-section. We must also consider that the actual pore dimensions are not constant but have a certain degree of variation and if uncertainty is high, it can have a negative impact. The sample was also rough on the backside, but this

does not prevent from having light backscattered. Finally, there are the errors caused by sample misalignment, or residual cross-polarization.

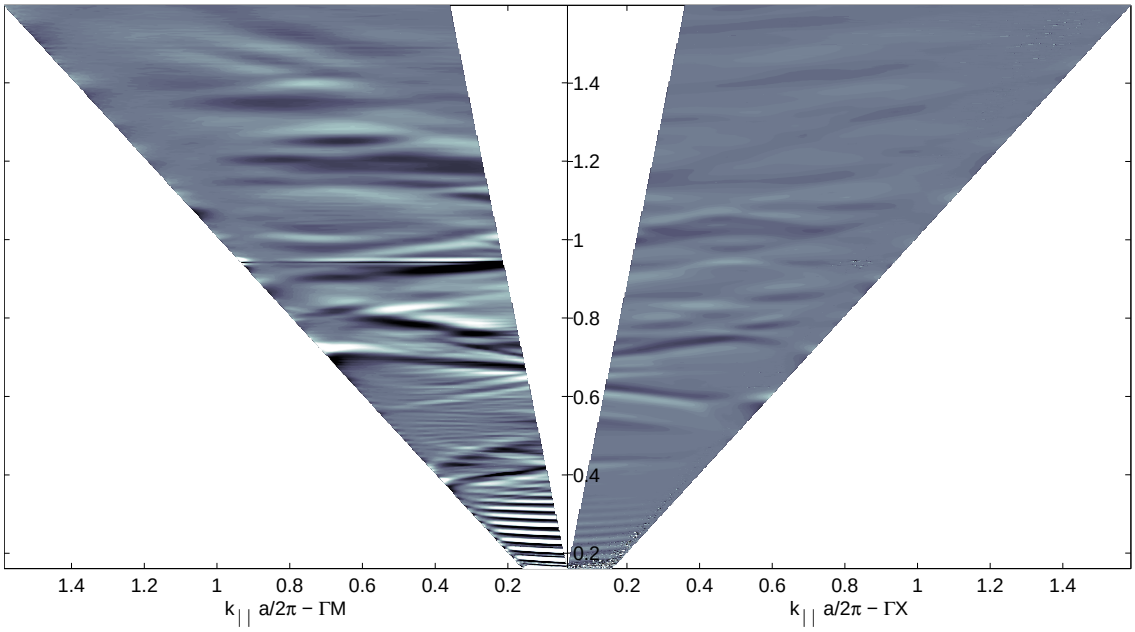


Fig. 29 – P polarization dispersion map of sample 2209g. All units are normalized.

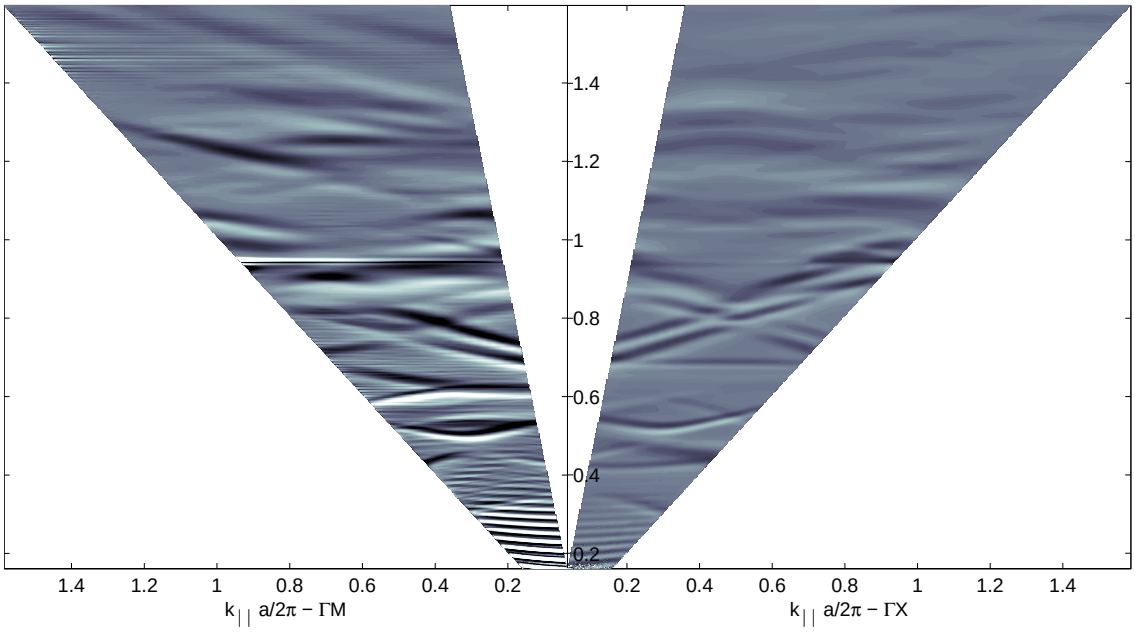


Fig. 30 – S polarization dispersion map of sample 2209g. All units are normalized.

Sample *2180l*, though using a triangular lattice, produced similar results, the bands just different, as shown in Fig. 31. For this case, the results were much cleaner, the bands showed stronger and the peaks were wider, despite the triangular lattice introduces more symmetries, thus the degeneracy of the bands increase in turn.

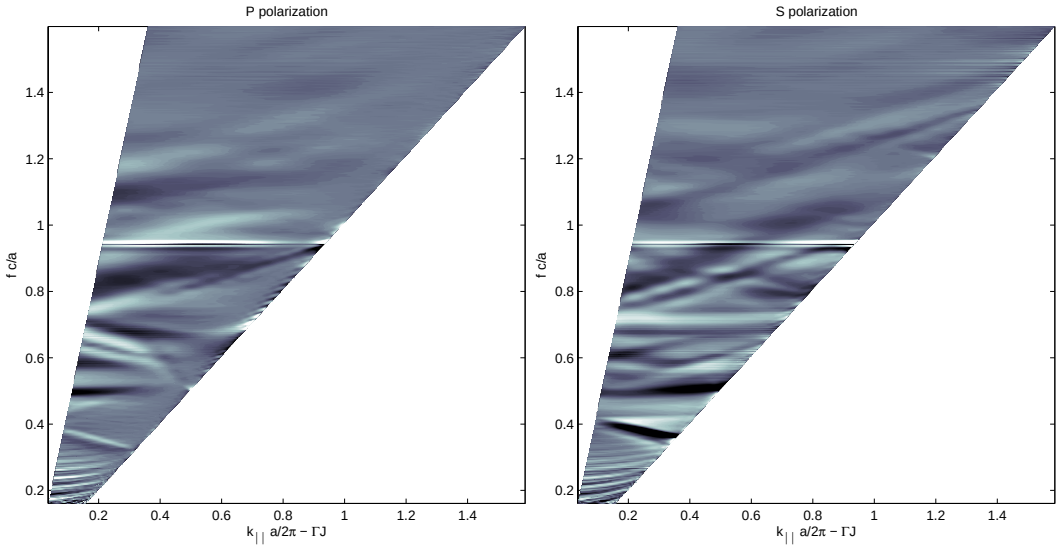


Fig. 31 – Dispersion map for sample 2180l.

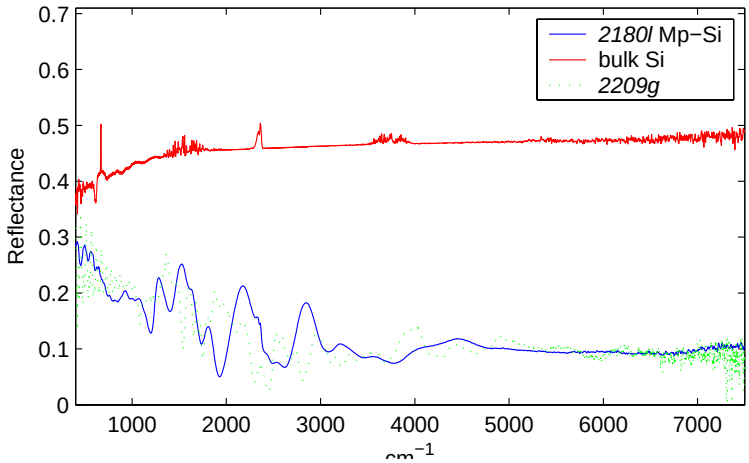


Fig. 32 – Comparison of reflectance of the sample 2180l to bulk silicon and sample 2209g. It shows more wide and defined features than the square case.

To obtain better measures, the next samples used were membranes of differing thickness. The next results are based on sample *2178i*. In Fig. 33 we see the actual SEM pictures of the sample. From them we find that the membrane layer is 33 μm approximately, of which pore extend up to 26.8 μm. The pores are relatively short, so pore diameter is conserved through the full length of the hole, with minimal variation, except for the rounded end. Pore diameter, on the other hand, does show higher variation, in the figure we have measured 2.12 to 2.36 μm, if we assume them to be the extreme cases of a uniform *cdf*, the *mean* pore diameter is 2.24 μm. The bird-eye view of the sample also reveals that the pore cross-section is neither a perfect circle or square, but a shape somewhat near a square, though we cannot really determine the form factor. We can calculate the fill-factor for the photonic crystal to be between

$$p = \frac{A_{\text{pore}}}{A_{\text{cell}}}, \quad p_{\text{CR}} < 1 - p < p_{\text{SQ}}, \tag{38}$$

where we need the reciprocal because we have air pores, and p_{CR} and p_{SQ} are the porosities for the circular and square **cylinders** respectively, given by:

$$p_{\mathrm{SQ}} = \left(\frac{\overline{d}}{a}\right)^2$$

$$p_{\mathrm{CR}} = \frac{\pi \overline{r}^2}{a^2} = \pi\left(\frac{\overline{d}}{2a}\right)^2 = \frac{\pi}{4}\, p_{\mathrm{SQ}}\,.$$

which for our case gives the fill-factor to be: 68.6 % < p < 75.4 %.

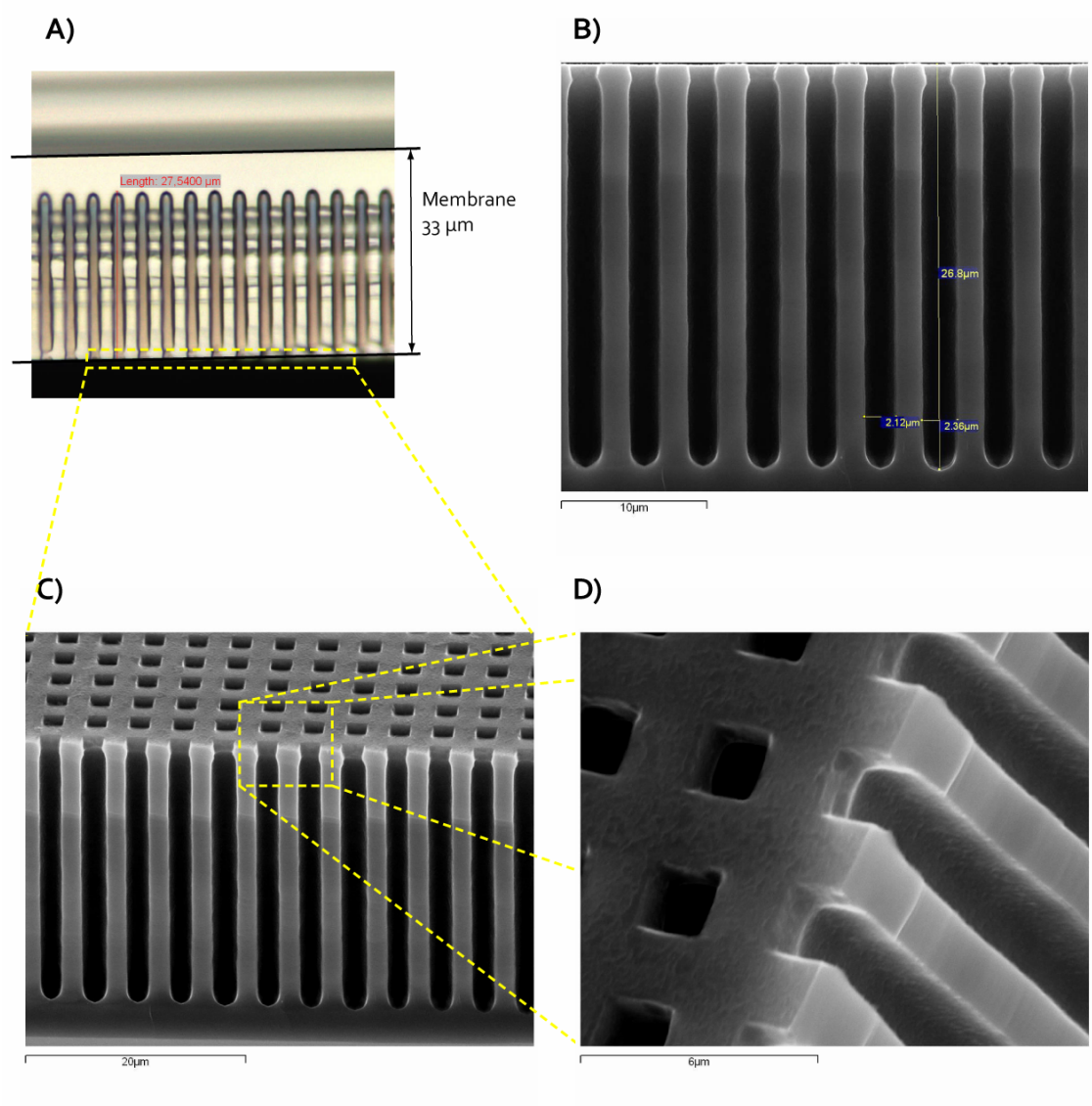


Fig. 33 – Images of sample 2178i. A) shows a 100x magnification optical microscope image, from where we can appreciate the membrane width. We can take a first approximation of pore depth and layer thickness to be 27 µm and 33 µm respectively. B) shows a SEM close-up and a more precise measure of the pore dimensions. C) and D) are SEM bird's-eye views of the lattice and pores, showing that the pore has an almost square shape and a slight kink at the beginning.

The FT-IR measures of sample *2178i* show clear and defined bands, in contrast with the square lattice of sample *2209g*. We can appreciate also the marked oscillations in the low-frequency range; as we are dealing with a thin membrane, it is expected that this oscillations are caused by Fabry-Perot interferences, in fact we will use them later to find the approximate effective index of the photonic crystal at those wavelengths.

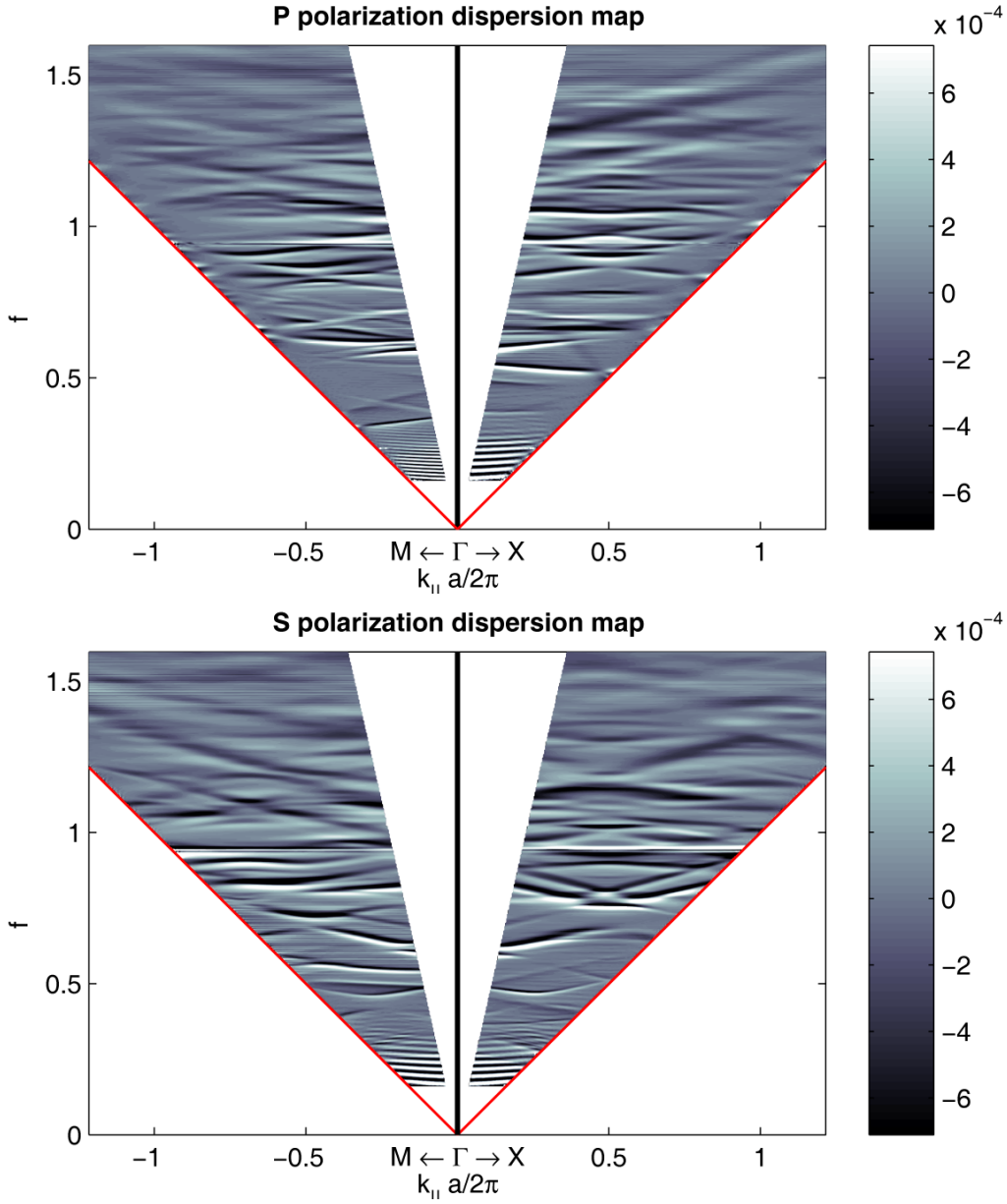


Fig. 34 – Dispersion maps for sample 2178i, P and S polarizations. Photonic Bands can be easily distinguished, though there is no photonic band-gap. At the low frequencies, $f <$ 0.4, we can notice F-P oscillations.

We will use the maps of Fig. 34 to match the simulated bands and perform other data analysis, which is deferred to simulation subsection.

6.1.3 – Nickel Samples

Late in the course of this work, some nickel-filled macroporous-silicon photonic crystals were produced. As we have yet to develop and understand how metallic photonic crystals behave, we aimed solely for quick characterization of the samples during the process of fabrication. The aim was to observe how the optical characteristics of a normal, modulated macroporous silicon crystal, changed when filled with nickel, and afterwards, stripping the silicon and leaving only nickel pillars. We seek confirmation that nickel structures can work as photonic crystals, and in a future apply the results to other metals.

For theses reasons, the FT-IR measures were less demanding, as accuracy was not our concern, but only the general spectral behaviour. Instead of doing the same sweep as for the previous, for these samples, we used a 10° step, and we did only a 50 scans average for all measures. In fact, during the measuring, we ascertain that humidity in the equipment was too high; this caused excessive absorption and noise in the water bands that appear clearly in the generated spectra. Nevertheless, the measures are still valid for our purposes.

We started with sample *2210d*. This sample is a square lattice, modulated porous membrane. This sample has nickel grown by electrolysis from the top face to the back face of the membrane. The first measures on the top face showed that the structure had a somewhat small band gap in the 600 to 800 cm^{-1} frequency range. This band-gap is due to the modulated pore, and corresponds to the equivalent sample without metal.

This band gap showed high-frequency oscillations and very low reflection, this is supposed to be caused by the nickel filling, as the equivalent sample with similar characteristics but without filling, showed near full reflection. Several factors may contribute to this poor performance, mainly related to the nickel electroplating. For instance, the surface was not perfect, as in many areas the nickel had poured out of the pores, creating a bumpy surface. Also not all pores filled, leaving some areas without metal.

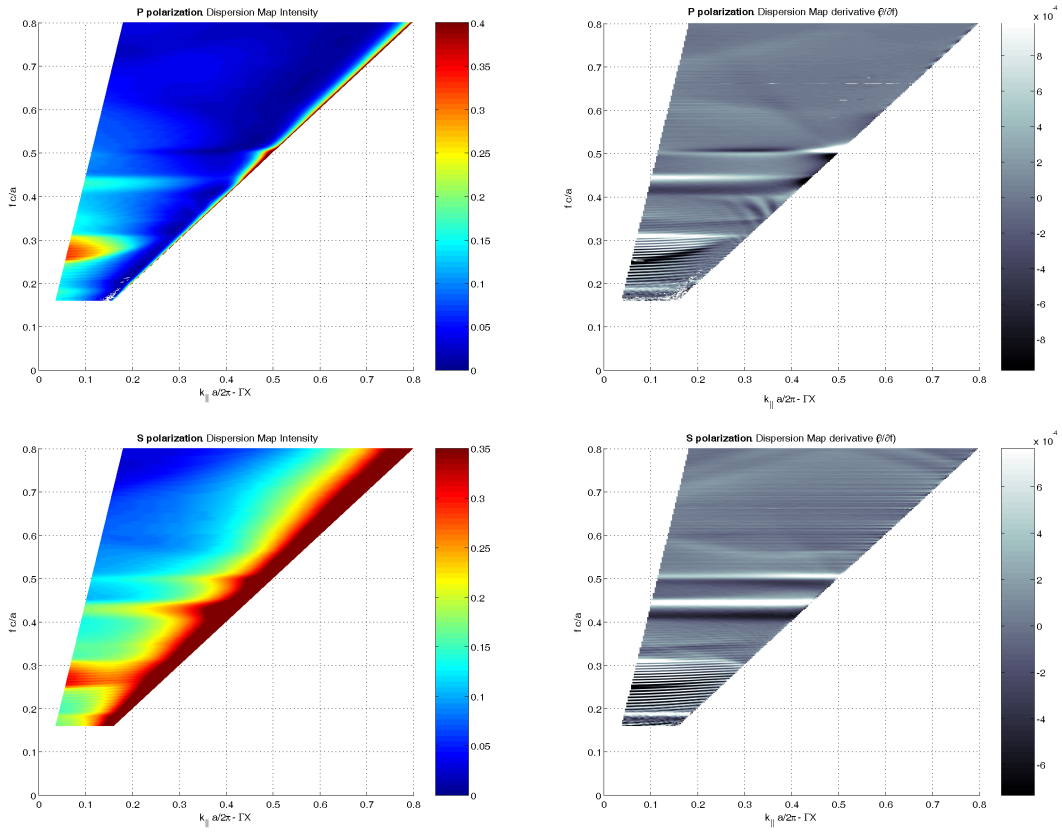


Fig. 35 – Dispersion maps for the first iteration of sample 2210d. We show both the direct intensity and its derivative to remark the appearance of photonic band-gaps, albeit of poor performance. The thick, embossed, bands on the derivative map equate to the band-gaps, as shown by the corresponding intensity map.

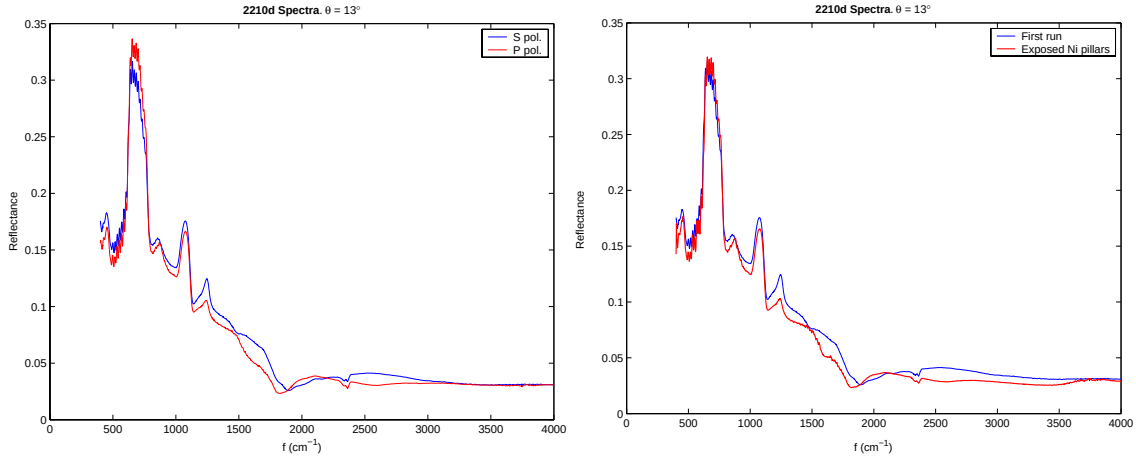


Fig. 36 – Spectra of sample 2210d at 13° incidence angle. The band-gap is clear in the 500 to 800 cm^{-1} range. After releasing the nickel pillars, response stays practically equal.

The first band-gap ranges $\overline{f}$ from 0.25 to 0.3, thus we have a 18 % band-gap at $\overline{f} = 0.275$. After this first measure, the sample was etched in KOH for a short time to eliminate a few microns of the surface and thus free the nickel pillars. The optical response was then measured again. Surprisingly, the obtained result was almost identical. This suggests that either not enough silicon was etched away or that nickel had no effect from the beginning.

This encouraged to use other similar structures for comparison. Samples *2210i*, *2210h*, and *2210g*, were analysed in the same fashion: we first measured the spectrum of the silicon structure filled with nickel, and again after stripping some of silicon surface and releasing the nickel. The spectral plots for these samples at 13° incidence is shown in Fig. 37.

The sample *2210i* improved respect to *2210d*, the overall reflection was higher, but it still had bad selectivity. They differ only in the electroplating process, the former being done better. For sample *2210h*, the pore shape was changed to have full 3D interconnected pores, forming an inverse opal structure. This creates a nice band-gap with high selectivity, of great similitude to other photonic crystals with the same structure but without nickel filling. Unfortunately, there is no measure with the nickel pillars released because the sample was destroyed in the process. Sample *2210g* shared characteristics with the previous, as seen by the blue spectrum in Fig. 37. In this occasion the removal of silicon to free the nickel pillars, changed radically the response, to practically near-zero. The sample was examined under the optical microscope, and revealed that, in fact, nickel had barely penetrated the pores; this caused that when removing the silicon, the whole structure collapsed, thus suppressing reflection.

This discovery lead to reckon that sample *2210i* had the same problem, it is highly probable that nickel had not grown inside the pores, hence giving such a similar response to non-filled photonic crystals.

Sample *2210a*, of the same characteristics as the previous, was analysed in the NIR range instead of MIR. In plain sight, the sample had the same lead-grey tone, but the FT-IR measure showed that its response was very poor at the visible range. This again could be caused by the problem described above, but it is presently unclear.

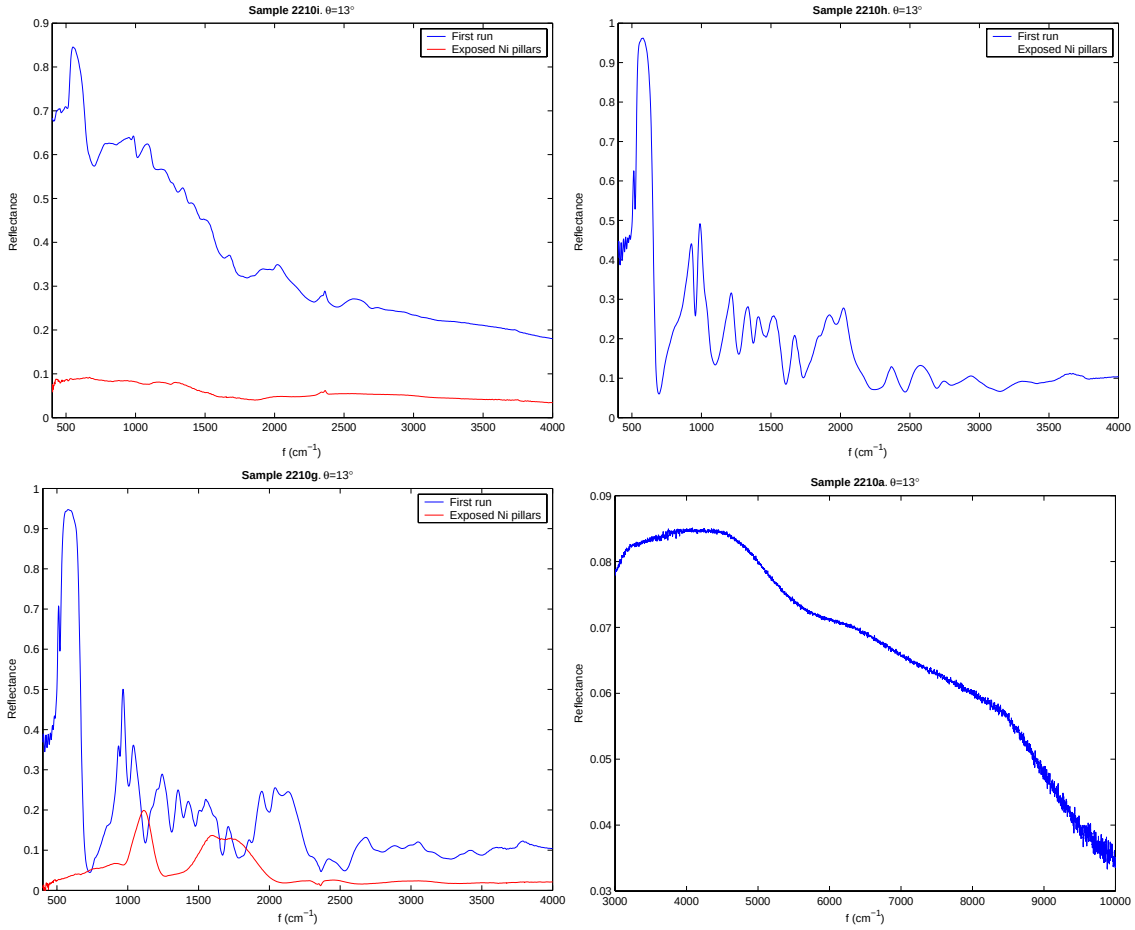


Fig. 37 – Spectrums measured of the different samples at 13° incidence showing the variation before and after freeing the nickel pillars. Sample 2210h was lost in the silicon etch process; sample 2210a is measured in NIR range. The response in the NIR was so poor that no further measures were taken. The change it the pore profile and electroplating process showed a great improvement of the optical characteristics from sample 2210d in the NIR range.

As the *2210d* sample did not perform as expected, and presented some defects, it was cleaned and refurbished. The first nickel deposit was eliminated with nitric acid, the sample itself underwent other cleaning steps, and a new nickel galvanization performed. This time, the electroplating set-up was changed to improve the final result, and extra oxidations were performed on the sample to make it a full 3D, inverse opal structure. The new FT-IR measures showed a vast improvement in relation to its first process.

It is remarkable that in this occasion, the sample behaves almost identically after all processing is done. The response is clearly due to the nickel, as the SEM images confirm its existence. In fact nickel has a broader range of high reflection, with half reflection at about 0.4 μm (25000 cm^{-1}), thus the cut observed near 1600 cm^{-1}, is due to the structure.

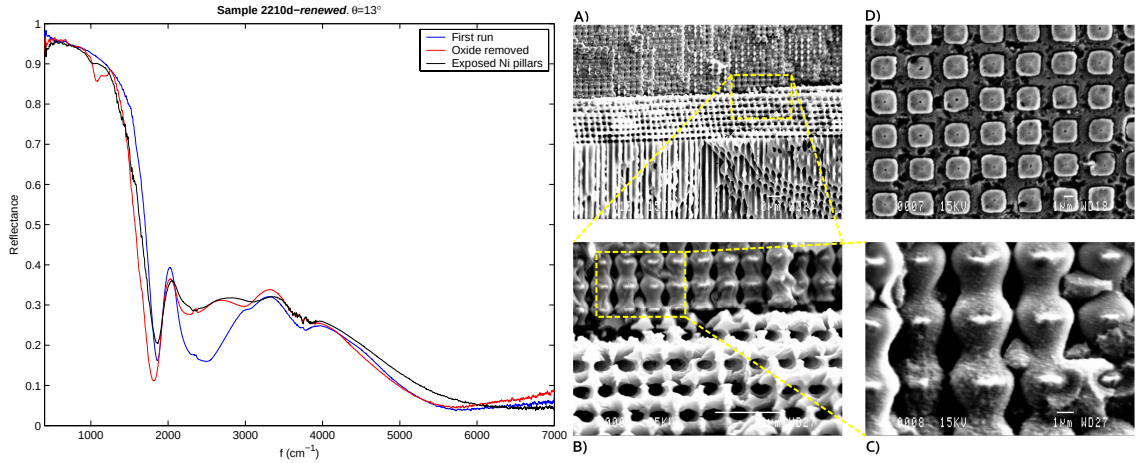


Fig. 38 – Spectrum of the renewed 2210d sample, and SEM images of it. The spectrum shows a great difference to its first iteration, having an improved reflection and band-gap. The SEM images clearly show the existence of nickel pillars. A) is a general lateral view of the membrane, showing the modulated nickel pillars freed of the silicon (whiter). B) and C) are close-ups of the metal structure. D) is a top view showing how the pores where filled, from the pore surface towards its centre.

6.2 – MPB Simulation

The theoretical and simulation part took two phases. We first simulated ideal, infinite photonic crystals with MPB. From these simulations we basically extracted the standing waves modes of the crystal and the propagation bands for a wave propagating on the surface. With these bands we extracted the equivalent low-frequency effective refraction index and we performed matching with the FT-IR measures, to prove that the fabricated crystals adjusted to the theory. Unfortunately, with MPB we are limited with the type of simulations and geometry as already commented in Section 5.2–; for instance, the simulated pores are of infinite extent, we have not considered conicity of the pore profile, or the surface interface.

The second stage consisted in using Meep to perform full transient simulation of the electromagnetic waves propagating into a macroporous-silicon crystal structure and will be discussed later in Section 6.3–.

6.2.1 – Band matching

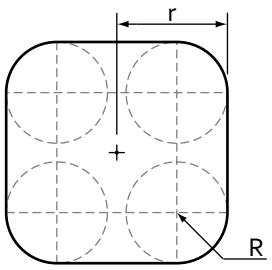
We simulated the measured structures in MPB to obtain the bands. The aim of it, was determining which incident waves can couple and to check that the simulation was accurate enough, at least in the range of interest.

To perform this matching, we took the derivative of the intensity in the dispersion map, basically because it offers greater contrast as explained in 6.1.1–, and overlaid the generated band diagram by MPB.

We did the initial simulations with perfectly round pores, thus reducing the variables to only one, the ratio of the pore radius to the lattice pitch, that is, the porosity of the macroporous silicon. The matching process involved sweeping the pore radius variable over a range near the theoretically etched one, and then visually seeking for the best option. This should give us an approximation to the real porosity of the sample.

In general, band separation and relative location is governed by porosity, while pore shape basically affects degeneracy. Thus, once we have the photonic crystal porosity, we redo the simulations for the given fill factor, while using the pore shape as the new variable.

For the macroporous silicon fabricated in our laboratory, in n^+ ⟨100⟩ silicon, the pore precursors are square pyramidal pits. After etching and various oxidation cycles, the final pore shape is something similar to a square with round corners. There is a mathematical formula that allows to describe with great accuracy this shape, and its implementation for MPB is given in Listing 1. Nevertheless, an easier approach was already made at the department, which simply created a square with true round corners. This implementation takes the usual geometrical parameters for position and material, and defines the radius *r* and form factor *f* shown in Fig. 39.



r : radius
f : form factor

$$R = r \cdot f$$

$$A = 4r^2\left[1 - f^2\left(1 - \frac{\pi}{4}\right)\right]$$

Fig. 39 – Geometrical definition for the "square with round corners" pore shape.

for this pore shape, the form factor goes from 0 for a perfect square, to 1 for a perfect circle. The area is very easy to find and allows using it in calculations inside MPB.

Using this method we found that, for sample *2209g*, we had an acceptable match using a coarse initial simulation with round pores, at pore radius $r = 0.35a$. From this point we re-simulated for pore radius from 0.312a to 0.350a, maintaining a constant porosity by the previous found value. This was done for TE polarization, and we had excellent match for the final $r = 0.336a$, which corresponds to a form factor $f \approx 0.8299$.

Unfortunately, when overlying the TM bands over the P polarization dispersion map there was no match whatsoever. For the moment, we tried to find a compromise solution, matching both polarizations at the same time, but the results were not satisfactory.

Observing more closely the TE polarization band overlay, we can appreciate how some of the bands do not couple to the photonic crystal, and how some features in the dispersion map have no respective in the TE bands. It was concluded that not all the bands could couple due to symmetry mismatch with the incident light, and that some features where excited by cross-polarization. However, we later see that the picture is a bit different.

If we just take the TE polarization as valid, being the best match, we find that the equivalent radius is much larger than that of theoretical dimensions of the sample, which stated that for sample *2209g*, the pores where conical from 2.3 μm to 1.7 μm (0.2875a to 0.2125a). This could be caused by oxidation steps widening and making rounder the pore, or that the HF etch process was slightly out of setting.

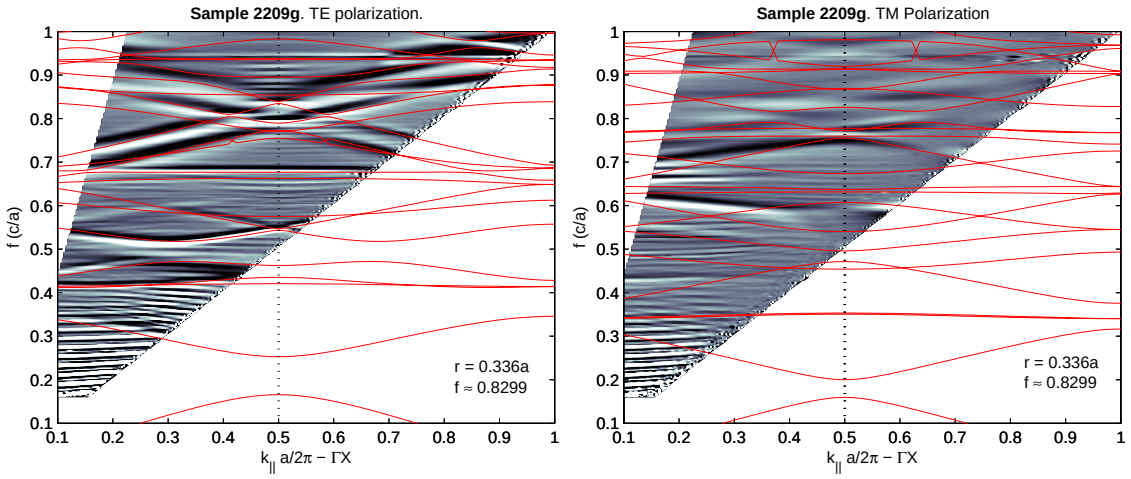


Fig. 40 – Band overlay for sample 2209g derivative of the dispersion map. We can appreciate how for TE (S) polarization, band adjustment is near perfect, while for TM (P) polarization, there is no matching at all. Even trying to adjust the TM polarization independently, gave deficient results.

We now centre on sample 2178i, which gave the best results for the dispersion map. Nevertheless, using the same approach to match the MPB calculated bands, lead to the same problem above mentioned. We had no answer for it, so it was proposed that pores perhaps had unbalanced principal axes.

6.2.2 – Testing asymmetrical pores

As commented previously, the fact that we could achieve excellent match with TE bands and S polarization dispersion map, but only poor coincidence for the TM case, prompted the study of the case that pores where not square or circular, but rectangular or elliptical.

It seemed like we could match TE bands at some pore radius, while for TM this radius was larger. We started simulating elliptical pores such that in X direction, the pore had the diameter for the TE case, while for Y direction it had the one corresponding to TM. This showed a problem with difficult solution: the porosity changed, but in this case, we could not determine which was the “correct” porosity, as obviously TE and TM modes showed different ones.

As such, it was not clear what approach to take, we could use directly the diameter for each case as the direct size in X and Y directions, but that altered substantially the result, giving incorrect bands for both polarizations. Using the TE porosity as base, helped to get a better adjustment to S polarization dispersion map but did not help with the other one. Even more, the fact that we have less symmetry lifted some of the existent degeneracy, thus setting off some of the bands for TE. We tried the same, but now fixing the TM porosity; the results were similar, TM had poor fitting, while TE now was completely off.

In the end, we concluded that ellipticity was not a factor in our samples, and that some other effect was causing the poor fit for TM polarization. This also precluded the possible need of measuring the sample in both principal axes to see if there was any change in the spectral response of the photonic crystal.

6.2.3 – Band symmetry coupling

Earlier, it was seen how not all bands did couple to the photonic crystal, as said elsewhere[11,27], the mirror symmetry of the incoming wave about the incidence plane must match to that of the photonic crystal's band to couple. Our aim, was to separate the bands according to its symmetry along the incidence plane to check if it really matched only to the corresponding symmetry of the incident wave.

We first found the parameters for the simulation, such that bands matched at least for the TE polarization. The final values where a pore radius $r = 0.302a$ (2.41 μm) and a pore form factor $f = 1$; these values deviate slightly from the actual measured dimensions with the SEM pictures as seen in Fig. 33, showing that good agreement between theory and experiment can be achieved. In fact, maintaining porosity, the bands changed very little for different form factors, and we could have easily chosen another value, such as $r = 0.288a$ (2.3 μm), $f = 0.7972$, which would seem a better fit[1].

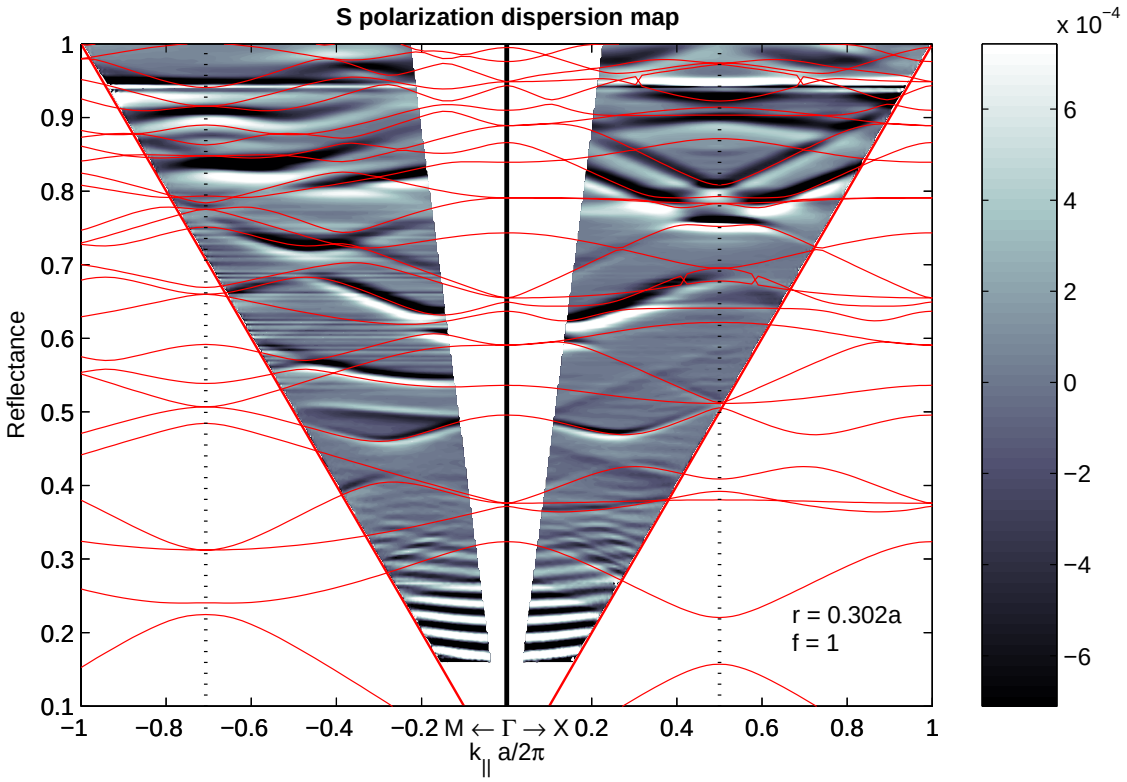


Fig. 41 – Sample 2178i band matching for TE polarization. The chosen values of *r* and *f* are the ones that give, in the author's opinion, the best match and easiest geometry to use. Fitting is excellent, up to high frequencies, were small deviations can be seen.

Once we had the working crystal dimensions, we did the simulation specifically for these, but separating the mirror symmetries about the incidence plane. Here enters into play what was commented in section 5.2.5–. For the typical ΓX direction, the plane of incidence is coincident with the XZ plane, thus we can directly perform the simulation with separated symmetries without any complication. However, for the ΓM direction we have to perform the technique described previously.

1 The matching proces is currently based on a "best looking" approach, so which is the best fit is hihgly subjective inside a reasonable range.

To separate the symmetries for the ΓM direction incidence plane, we had to rotate the geometry as described. This involved altering the in-site made rounded pore generation routine to include a rotation angle; once done, we proceeded to perform the simulation.

After preparing the bands, and separating their Y symmetries, we tested all resulting sets of bands with each dispersion map. We found that, as expected, matchig for S polarization was excellent with TE, Y-odd bands, but surprisingly, the discarded bands for the S dispersion map, matched with the same precision in the P polarization dispersion map. This is shown in the final results in Fig. 42 and Fig. 43.

Apart from the fact the we are basically only matching TE bands for both measured polarization, is that TM bands have little influence. The final result is an exceptional matching of theory and simulated bands of our simplified model, with the actual sample measured spectra.

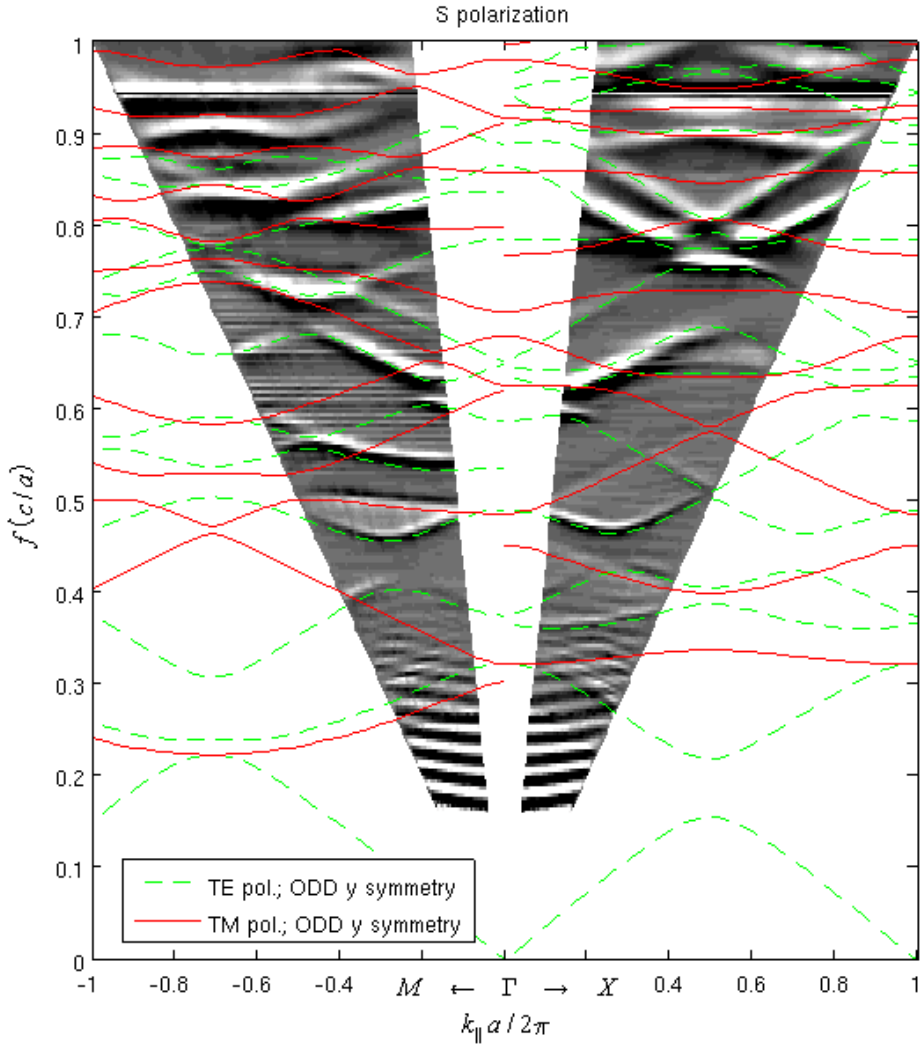


Fig. 42 – Sample 2178i S polarization band matching. We have overlaid the measured dispersion map with the simulated TE and TM bands with the correct Y symmetry. S polarization couples odd **Y-symmetry from what we can see.**

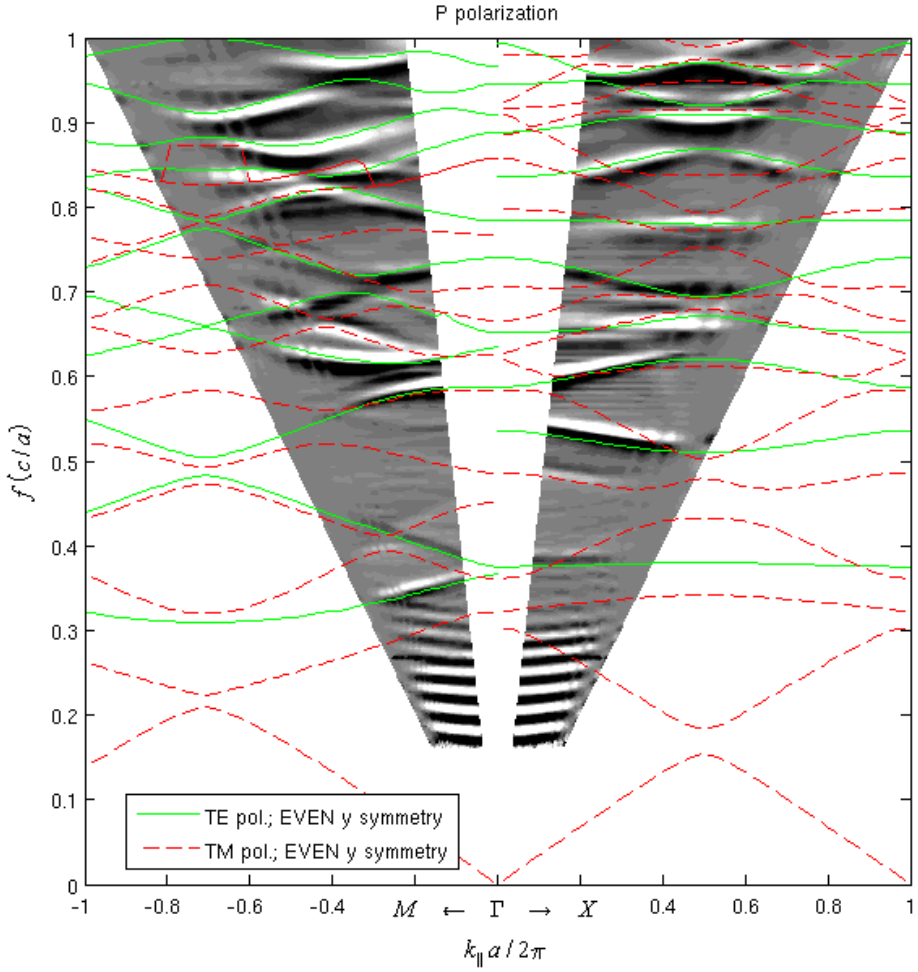


Fig. 43 – Sample 2178i P polarization band matching. Symmetries have been separated and overlaid to the sample dispersion map. We can appreciate how for this case, both TE and TM Y-even bands couple for P polarized incident light.

6.2.4 – Low frequency effective index of refraction

Observing carefully the dispersion maps, we can notice how at the low part of the graph, there are some noticeable oscillations. These oscillations are caused by Fabry-Perot interferences, for the medium does not have a fixed dielectric constant. It turns out that for different short wavelengths, the wave sees an *effective* medium of *lower* dielectric constant than the base material, in this case silicon.

Several studies for inhomogeneous media and colloids, for example, composite materials, give two approximations at DC: Maxwell-Garnett[1] and Bruggeman. These approximations consider a material which is not uniform but is a mix of interconnected particles, with different conductivities, or non at all, where there may be voids inside the material. These approximations differ in the assumptions made to simplify the equations and give slightly different closed formulas for the effective conductivity al very low frequencies. We present both here, the M-G:

1 This formula is also known as Clausius-Mossotti

$$\frac{\varepsilon - \varepsilon_0}{\varepsilon + 2\varepsilon_0} = p_1 \frac{\varepsilon_1 - \varepsilon_0}{\varepsilon_1 + 2\varepsilon_0}, \tag{39}$$

where p_1 is the volume fraction occupied by material 1 in the background material, ε_1 is the dielectric constant for material 1, and ε_0 for the background. Bruggeman's symmetrical approximation considers N materials embedded in a background material:

$$\sum \delta_i \frac{\sigma_i - \sigma_m}{\sigma_i + 2\sigma_m} = 0, \tag{40}$$

where δ_i are the relative volume for each material, σ_i are their conductivities, and σ_m is the conductivity of the background material.

Although each approximation has its shortcomings, the most easy to use for our purposes is the M-G solution, which we will use for our comparisons. The results using M-G, give an effective dielectric constant of:

$$\varepsilon_{\text{eff}} \approx 4.8 \rightarrow n_{\text{eff}} \approx 2.2 .$$

The F-P oscillations occur because the incident light transmits through the crystal to its back face, where it undergoes a new reflection. We are thus interested in the component of the propagation vector that penetrates the crystal and travels perpendicular to the interface. This component determines the propagation speed inside the crystal, and thus defines the effective refraction index.

For this reason, to perform the comparisons, we simulate an on-axis, normal incident wave, going into the crystal. The obtained bands represent the propagating modes as if the photonic crystal was a waveguide; as there is no tangent component for the propagation, there will be no diffraction effects, and thus the wave in the photonic crystal will solely depend on the material properties and structure.

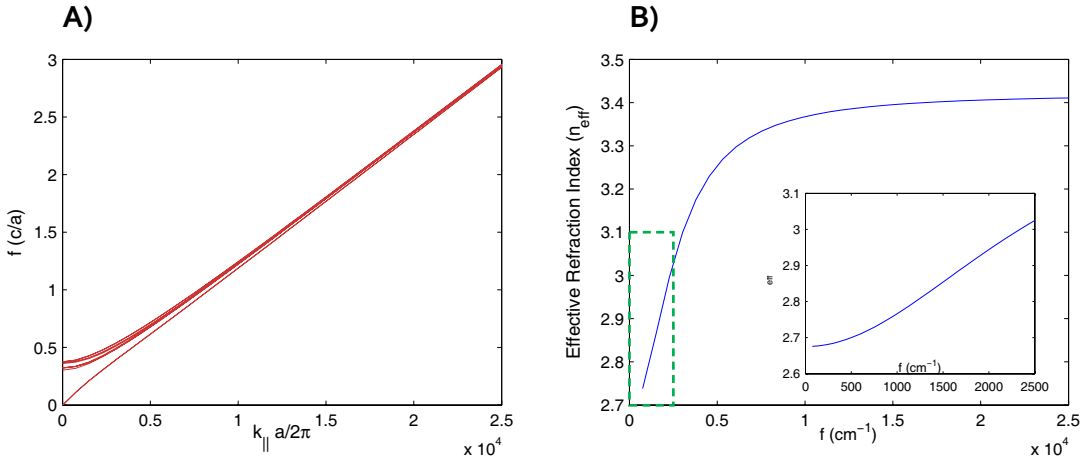


Fig. 44 – Simulation for normal incidence and extraction of effective refraction index. The values used are those for sample 2178i, with r=0.302a and f=1. A) shows the band diagram. B) gives the effective index. The B) inset is a detail for low frequencies.

It is easy to interpret Fig. 44. We pick the fundamental band, which corresponds to the lowest propagating mode, applying the results from equations 36 and 37, we find:

$$\mathbf{k} = \mathbf{k}_{\perp} = \hat{\mathbf{n}} k_{\perp} \cos\theta_i \,,$$

$$\overline{k_{\parallel}} = n_{\text{eff}} \overline{f} \cos\theta_i \xrightarrow{\theta=0^{\circ}} n_{\text{eff}}(k) = \frac{\overline{k_{\parallel}}}{\overline{f}(k)}, \qquad 41$$

where the bands expresses the coupled frequency at a given propagation vector, as we can further express the propagation vector for its frequency in *air*, we can give the effective refraction index respect to air.

From Fig. 44, it is interesting to observe that for high frequencies, the effective index tends asymptotically to the index of the substrate, in our case silicon. This is due that high frequency components tend to concentrate on higher index materials, thus the wave is basically affected only by the silicon bulk.

It is also worth noting, that the MPB simulation seems to give, using linear extrapolation, a value of about 2.67. This value differs substantially of the theoretical M-G effective index. We must be careful when using such approximations for photonic crystal design, as they are very coarse and we can introduce considerable error.

We now relate the simulated index to that of the F-P oscillations in de dispersion map. To extract those oscillations, we take only the range from 400 to 1000 cm^{-1}, where we can clearly see the oscillations. From this data, we take a much-filtered copy, to smooth out the F-P oscillations and use it as a base line. We next de-trend our dataset with this baseline, thus having a zero-mean oscillating signal. To extract the effective index we simply find the period of the data and apply the F-P equation:

$$\Delta k = \frac{1}{2wn\cos(\theta)} \rightarrow n = \sqrt{\sin^2(\theta_i) + \left(\frac{1}{2w\Delta k}\right)^2}, \qquad 42$$

where Δk is the wavenumber period, w is the sample thickness, n is the refraction index, θ is the transmitted angle, and θ_i is the incidence angle.

To perform the calculations we choose 13° for angle of incidence, as it gives us the best signal, the polarization is not important, where the results are shown in Fig. 45.

It is puzzling the low resemblance between measure and simulation, given that the dispersion map and bands had an excellent match. We can see, how, though the measure-extracted index is centred o the simulated band, the tendency is completely divergent. The simulations shows a slow, gentle transition from DC to silicon effective material, however, the experimental data shows an abrupt change in index value, passing from a *smaller* value to silicon dielectric constant in a much smaller span. The computed median value for it was

$$n_{\text{eff}}^{\text{measure}} \approx 2.67 \,.$$

When extracting the effective index with this method, we have some indeterminacies that affect the obtained values, especially the effective layer depth, as it is not clear whether to use the membrane thickness, the pore depth, or a smaller value. Nevertheless, changing the thickness parameter only affects the overall value, as for small angles we can approximate expression 42 by

$$n \approx \sqrt{\theta_i^2 + \left(\frac{1}{2w\Delta k}\right)^2} \approx \frac{1}{2w\Delta k}; \quad \theta_i \ll \frac{1}{2w\Delta k},$$

indeed, for the thicknesses we work with (tenths of μm,) and the frequency band (with a period near 60 cm^{-1},) gives us that the incidence angle is approximately 0.22 rad, while the fraction is about 2.083. Nevertheless, this is no reason alone to explain the different results.

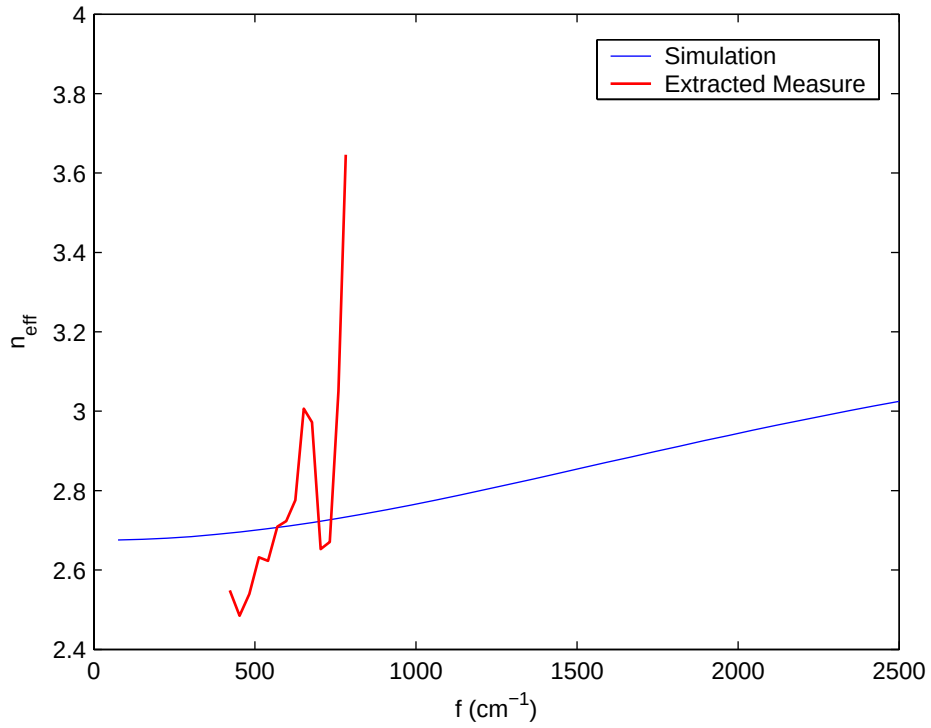


Fig. 45 – Comparison of the effective index as simulated and extracted from FT-IR measures.

Currently, there is no clear solution to obtain a coherent value for refraction index at low frequencies. M-G method delivers a considerable underestimate, thus one must be very cautious when doing quick numbers, but is useful for initial design. It is probable that the measures do not take into account many factors that alter the received spectrum, or that the algorithm is not adequate. Finally, it seems that the simulation value should be a starting point for the next phase of the design process for a photonic crystal,

6.3 – Meep

The next step in the simulation area was using Meep. It represents a radical change from the MPB concept. Meep is based in one of the fundamental simulation architectures used in electronics and wave propagation, the time transient simulation.

The principal aim of the use of Meep in the present work, was learning to simulate a macroporous silicon photonic crystal structure, and extract the same dispersion map we obtained from FT-IR measures. The ability to simulate photonic structures, will allow us in a future to extend this work to more complex situations; it will also allow to slightly speed up the design-fabricate-measure-review cycle.

As all transient simulations for electromagnetic propagation, the time necessary was considerably higher than with MPB. While with the later we could spend from a typical couple of minutes to a quarter of hour, with Meep, simulations time easily extended over one hour, the fastest, and coarser, lasted in excess of half an hour. From this, it is easy to see how we are limited in the number of simulations and test we may do to check that a proposed simulation is valid or gives interesting results.

6.3.1 – Plane Wave Propagation and Spectra

One of the main excitations used in electromagnetics are Plane Waves. We can decompose fields by Fourier Analysis in plane waves, or we can approximate distant sources as a plane wave, for a sufficiently small receiver. We use plane wave for the reflection spectra simulation, as the typical analysis for a wave impinging on an interface material; for instance, the Fresnel reflection and transmission coefficients are found using plane wave on infinite, flat surfaces.

Unfortunately, Meep does not provide yet field sources, only current sources; and those sources are limited in the shape, components and functions achievable. We could use very generic functions with volumetric sources defined over the entire computation cell, but this is highly inefficient and slows exponentially simulation speed.

For this reason, to do acceptable quick simulations, we are forced to search for alternative configurations of current sources, such that the effective wave generated is the desired plane wave. The situation is further complicated by the fact that we to b able to produce *oblique* plane waves, as we want to simulate the behaviour of the photonic crystal for light incoming at different incidence angles.

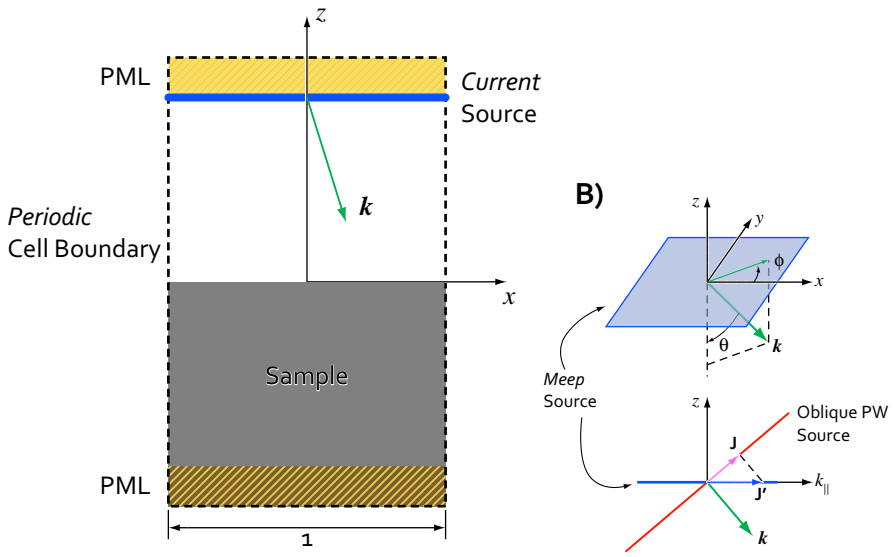


Fig. 46 – Geometry and source parameter definitions. The cell consists of a periodic boundary condition with PML layers on both top and bottom Z planes, the sample is placed below Z=0, while the source is at the top. In B) we show the actual reference system for defining the wave propagation, and the projection of the original oblique source to our XY source definition.

We started by defining the basic cell geometry, shown in Fig. 46. As we wanted a steerable plane wave, we needed to define a source that calculated the currents according to the desired parameters. We choose the source to be on top of the cell, anb

be a plane on XY. To define a plane wave we first need to set up periodic boundary conditions, this is necessary to extend the wave to all space, along the propagation vector, thus we need to define the **k-point** variable[1] to the *effective* k vector of our wave:

$$\bar{\mathbf{k}} = \begin{bmatrix} \sin\theta \cdot \cos\phi \\ \sin\theta \cdot \sin\phi \\ -\cos\theta \end{bmatrix} \rightarrow k\text{ - }point = \bar{f} \cdot \bar{\mathbf{k}}\,,$$

where the angles follow the convention shown in Fig. 46. As the original current source is oblique, we need to find the equivalent currents for our XY plane source. These currents are the projection of the original to the new plane:

$$\mathbf{E}_0 = e^{j2\pi\bar{f}\,\bar{\mathbf{k}}\cdot\mathbf{p}}\,.$$

where **p** is the position vector referenced to the source origin. The fasor amplitude function defines the plane wave, so now we must choose the time profile. For testing purposes, we choose a pulsed continuous source, whose results are shown in Fig. 47. When we perform the reflection spectra calculations we will use a Gaussian source, as it gives a nicer spectral response.

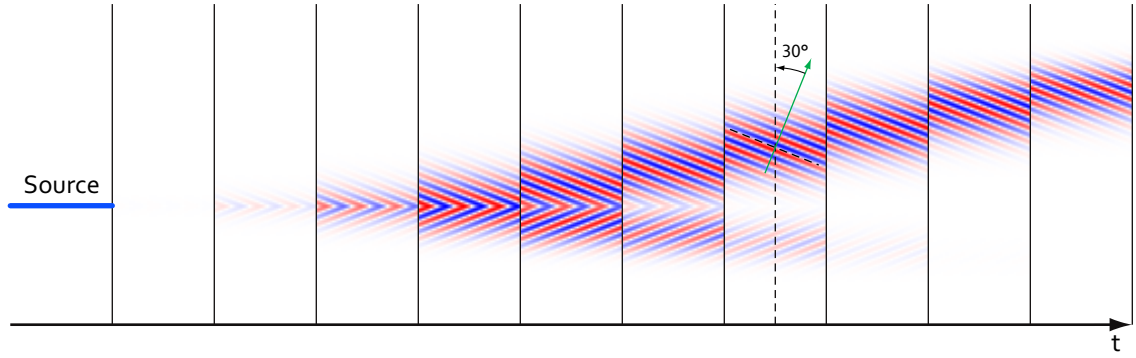


Fig. 47 – Time evolution of a 30° plane wave pulse. Several time-steps have been captured from a test Meep simulation to check that the source behaved correctly.

In our present work, we have used always P polarization, by setting the source current to be E_x component; we could have S polarization if we used magnetic currents.

6.3.2 – Reflection Spectra

With our source defined, we are now ready to perform simulations of our structures to extract the reflection flux spectrum. As commented in section 5.3.6–, to compute the reflection flux we need to perform two simulation runs, one with an empty space to take the reference flux, and another with the studied geometry to take the combined flux of source and reflection. Meep then automatically subtracts the reference to obtain only the reflection flux. To get the reflection spectra, we simple need to divide the reflection by the reference flux.

1 See pag. 48

When doing these simulations we use a P polarized,[1] Gaussian pulsed plane wave. We use a Gaussian profile because it gives the best spectrum shape. If we used a continuous source, we would have spectral leakage, as there is no defined start and end time; to avoid this problem, we need to window the data, but this would add uncontrolled harmonics. Thus, the best solution is the Gaussian, as approaches smoothly to zero, ensuring that the flux will also have a Gaussian shape.

Using a pulsed source also has the benefit that we get the response for several frequencies and angles in a single simulation. We will exploit this to extract the reflection spectra. Indeed, when we set up the source, we give it a central frequency and a **fixed** k-point; as at the output we have a frequency range, all frequencies *share the same k-point*, thus each frequency corresponds to a *different incidence angle*. This is in opposition to the FT-IR measures, where the incidence angle is constant for the spectrum, while k-points change accordingly with frequency.

The situation is depicted in Fig. 48. We can see from the figure that for k greater that 0 (normal incidence), part of the simulated flux falls out of the light cone, this means that part of the simulated flux will not couple to the crystal, and thus there will be no reflection. Certainly, this effect can be seen in the sample given in Fig. 48; at 30° incidence for a central frequency $f_c = 1.06$, the light cone cut is at $f = k = 0.53$.

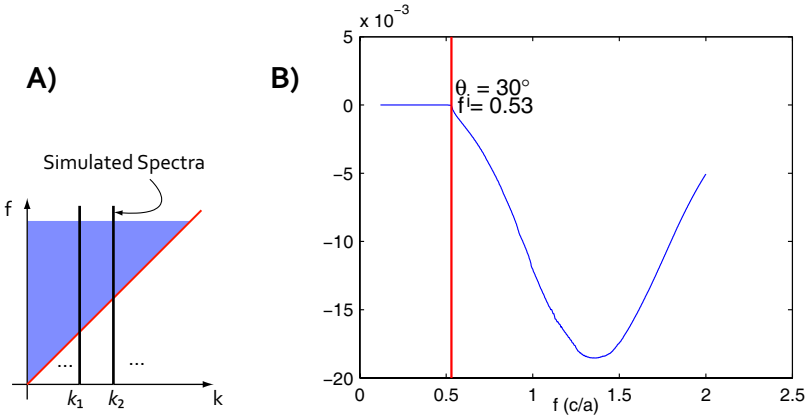


Fig. 48 – Meep simulates spectrum for fixed *k* vector. A) is a schematic diagram of where are placed the simulated values on the dispersion map. This involves cutting the portion of the spectrum outside the light cone that does not couple to the photonic crystal, as shown in B).

As we want to get the dispersion map, we need to perform a sweep for the incident angle. We have developed a Meep control script that takes the incidence angle and the desired band of interest, and runs a single simulation; we then developed a shell script that takes the list of angles of interest and automates the simulations for each angle, and for the reference simulation.

We first did a rapid simulation with low resolution, *res*=10, to see if the results were correct; the obtained spectra showed that for the high frequencies, the flux obtained was incorrect due to aliasing. This happenes around $f = 1$ as shown in Fig. 49, this effect is in fact documented in the Meep's User Manual.

1 see above

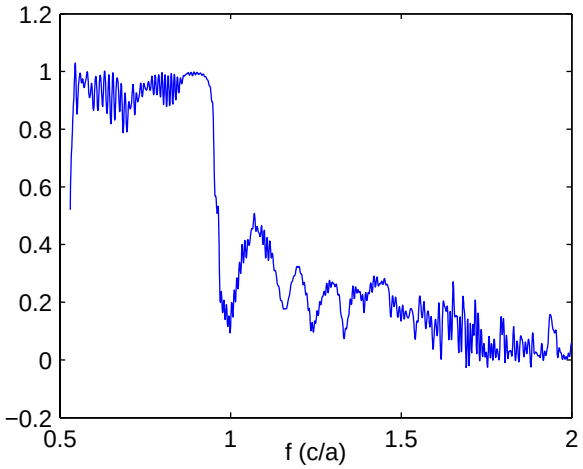


Fig. 49 – The low resolution causes aliasing and artefacts for frequencies above 2×res. This is due to the sampling theorem of the Fourier transform used to calculated fluxes.

6.3.3 – Sample Simulation

We tried simulating sample *2178i* with Meep. For this, we defined a generic cell based on the previous one showed. This cell has a single cylindrical pore in the center of a square lattice. The pore is finished by a hemisphere and is supported by a silicon layer of configurable thickness. The silicon substrate layer is suspended in air by a *padding* amount, such that we can have either a pore in silicon bulk, a pore in a silicon membrane, or a porous membrane, simply adjusting the corresponding parameters. All dimensions are normalized to the cell side length a, that is 4 µm by default.

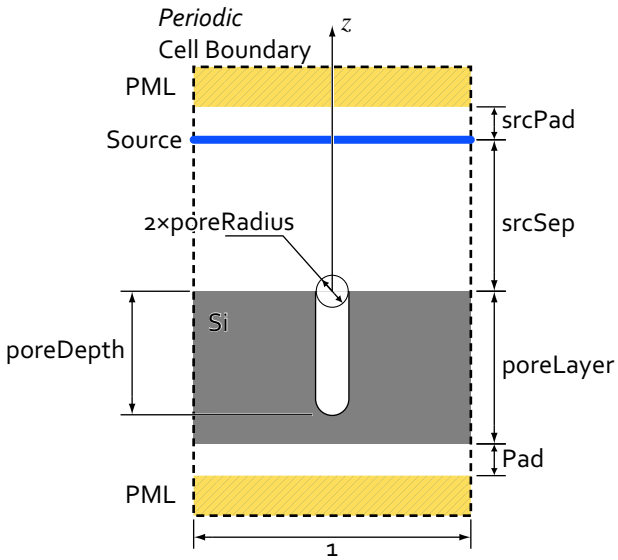


Fig. 50 – Cell description and parameters for simulating a square lattice of cylindrical pores. At the tip of the pore is a hemisphere of the same radius as the pore.

The first attempt tried to mimic the complete structure and surroundings: we defined a 33 µm thick membrane, with 26.5 µm deep pores at a resolution of 30 pixels per unit length.

The obtained results were discouraging; there was a huge amount of oscillations masking any interesting features. This heavy oscillations are caused by the perfectly

polished back surface of the membrane, which reflects back almost all light and causing Fabry-Perot interferences.

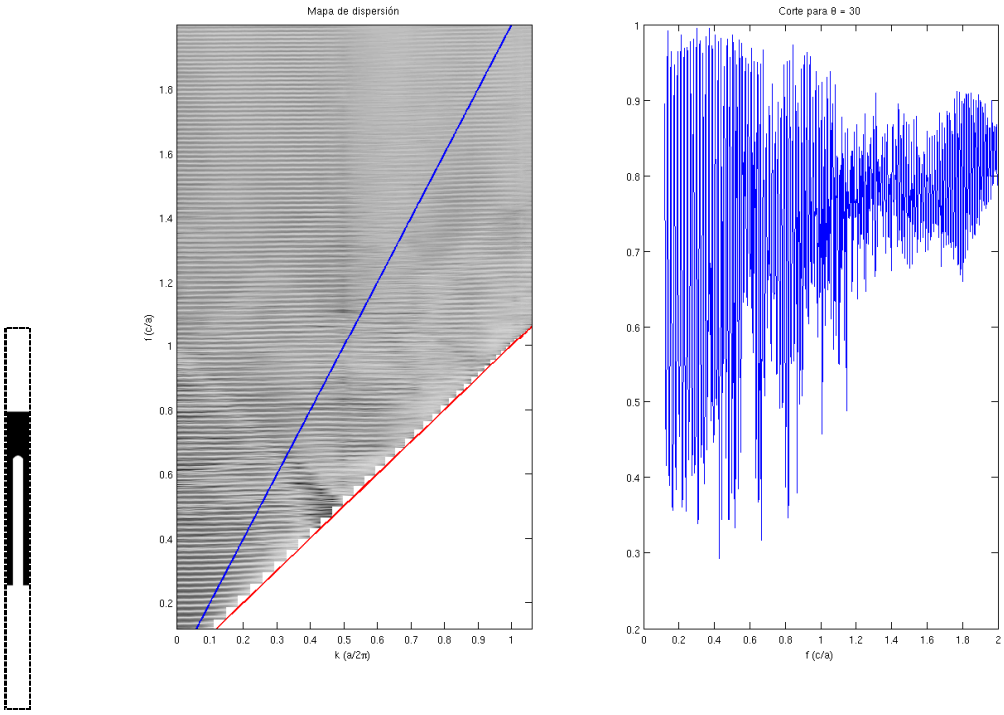


Fig. 51 –Simulation results for the membrane shown. F-P oscillations cover the features.

It was again simulated with a higher resolution of 40 pixels per unit length. Despite the change it was obvious after a few iterations of the angle sweep loop, that there was no improvement. Thus the simulation was aborted early.

In this situation, it was thought that perhaps we could get a better result if we approximated the membrane by an infinite silicon semi space, as if it was porous bulk silicon. This approximation yielded better-looking results, but still the effects are very thin and subtle.

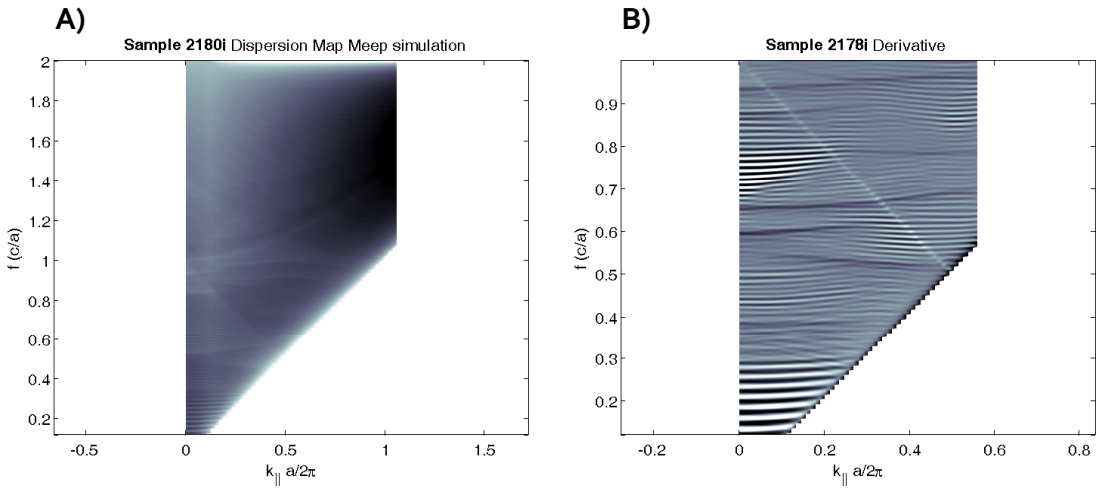


Fig. 52 – Simulation results for the same pore, but on bulk substrate. Oscillations have reduced considerably, but there still are. We can now appreciate more detail that for the previous case.

This simulation can be considered acceptable if we take into account that the back face of the membrane is fairly rough, thus any light arriving to it will have less backscattering influence, thus reducing the Fabry-Perot oscillations observed before.

Nevertheless, the spectrum bands are still thin and difficult to observe. In the derivative portion, we have made a close-up, to view better the bands, and we can see how oscillations are still very annoying.

In fact, in the high-resolution measure made of sample *2178i*, these oscillations were visible, though of a much smaller magnitude. This was one of the reasons to filter out the original measured spectra. Nevertheless, even when we tried filtering the Meep results for the membrane, oscillations were so strong and band coupling so feeble, that there was no effective means to smooth them out without loosing the spectrum features.

6.3.4 – Adding and Adaptation Layer

The observed oscillations difficult greatly the analysis of the generated data, and renders the simulation useless. To alleviate this situation, it was proposed to use an adaptation layer between the membrane back face and air. The aim was to provide a smooth transition in refraction index, and use the layer as an *antireflection* coating.

We created a Meep procedure to create a 4 µm thick stack of thin layers where refraction index varied linearly, which gave a smooth grading for dielectric constant, with continuous first derivative.

The simulations were compared against a TMM analysis of the stack, which gave acceptable match up to our working frequencies.

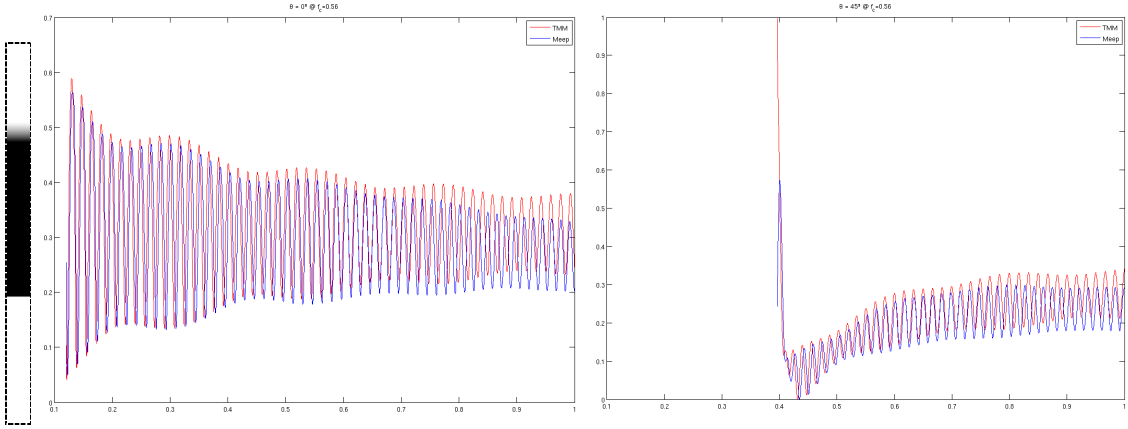


Fig. 53 – TMM and Meep comparison for the reflection of a membrane with the adaptation layer shown in the left part of the figure. The refractive index is varied linearly over 4 µm in a stack of 128 thin layers.

Still, although oscillations were reduced, the results were still unacceptable for our purposes, and the approximation of infinite silicon substrate worked suitable enough. Thus, we did a test to check if improvements would be noticeable on the sample geometry, but the obtained results did not differ substantially from the first membrane simulation.

VII – Conclusions

The objectives initially established for this work have been satisfied: Understanding of the mechanisms involved in the Mid and Near infrared by Fourier Transform Spectrometry and the learning of the use of the recently acquired FTIR spectrometer of the group have led us to perform high precision and low noise measurements. In order to extent the capabilities of the software originally included in the setup, new code has been developed in this work. In another hand, simulation tools have been applied to the calculation of the response of real samples of Photonic Crystals. Two software packages have been used for the simulation of our samples: MPB and MEEP.

Finally, measurements and simulations have been compared. The agreement was excellent in many of the measurements but important differences arose in other. After the study of the band structure of these samples, we have explained the reason of the differences and new simulations were done including a correction for the modes that are actually excited in the crystal by the external radiation. The new match between simulations and experimental measurements was excellent in the entire measured spectrum. A more detailed description of the work and the obtained results is given next.

We have first learned to use fully the Vertex70 FT-IR spectrometer that was recently acquired by the group. We now have knowledge of the detailed operation of the FT-IR and the concepts behind spectrometry. We have gained skills to extract useful information from the measured spectra and operate the equipment in different conditions. For improved accuracy and repeatability, we have learned how to control the environment humidity that has pronounced absorption regions in the NIR range, and to achieve precise sample placement.

The FT-IR measures have produced a number of reflection spectra sweeps at different incident angles with various incidence directions and both P and S polarizations separated, which have been post processed to obtain the dispersion maps of our macroporous silicon samples. In these maps we are able to observe the photonic crystals' characteristics. We can see from the figures in Section 6.1.1– how the incident radiation couples to the crystal modes creating peaks and valleys that follow the photonic crystal's band structure. The initial samples were difficult to measure and produced blurred diagrams. Nevertheless, the later samples produced the highest quality dispersion maps, clearly showing the desired effects. We later confirm that these features show an extremely good agreement with theoretical simulations.

The used resolution was high enough to record the Fabry-Perot oscillations from the back surface and the porous layer. This was exploited to calculate the effective refraction index of the macroporous silicon photonic crystals at the low frequency range from 500 to 1000 cm-1. The computed effective index then was matched against simulation and theoretical approximations. On one hand, the analytical approximations lead to gross underestimation, and did not yield usable results. On the other hand, simulation and measure showed better agreement, they had the same mean value and a similar span, but the index rate of change changed dramatically for both cases. This suggests either that the simulation set-up is not correct, or that the algorithm used to extract the effective index from the spectrum measures is not adequate. Nevertheless, to

get an approximation at the low frequencies, the simulation gives an acceptable approximation

In summary, we have successfully characterized our 2D and 3D samples in the NIR region with the FT-IR, we have shown that the macroporous silicon samples behave like photonic crystals and the existence of a photonic band-gap.

The next stage run in parallel with the FT-IR measures has been simulation. We have used the Transfer Matrix Method to verify some relevant aspects of the obtained results, for those cases where we could express the set-up as a stack of alternating material layers. The TMM can provide exact solutions, fast and accurately, that we have used to perform quick verifications.

We have used MPB to simulate the standing field's patterns in the silicon photonic crystals and to extract the photonic crystal bands in edges of the Irreducible Brillouin Zone.

The obtained bands have been overlaid over the corresponding dispersion map picking the best matching results. As already said, we have achieved excellent agreement between simulated bands and the measured spectra features. The search for the optimal fit of the bands consisted of generating a simple simulation of cylindrical pores for a small range of pore radius. From this band set we pick the one with in our opinion presents the best option. This determines the effective porosity of the sample, and now we can perform the second step simulation, keeping fixed porosity while varying the pore shape factor. In general, band results vary slightly, and the final band set chosen depends largely on subjective criteria, but the chosen geometric parameters generally correspond to the actual SEM dimensioning of the samples.

When doing this analysis we faced the puzzling behaviour of the measured P polarization and simulated TM polarization. Apparently, there was no correlation between them. It was explored the possibility that pores were asymmetrical, of elliptical cross-section. This was not the case, and the issue was solved when studying the coupling for different Y-symmetries.

Effectively, in the maps we clearly appreciate that not all bands are excited by the incident radiation for a given polarization. This is related to the incidence plane symmetry (Y-symmetry in MPB). Indeed, after separating the bands according to it Y-symmetry, we could observe that the complete set of each TE matched each one of the of the measured dispersion maps: TE, Y-odd bands matched S polarization, while TE, Y-even bands did it for P polarization. Even more, the same was done for TM bands, and the same trends could be observed; the difference was the fainter response of the crystal to them.

In order to separate the Y-symmetries in MPB when simulating directions other that the standard Cartesian, we had to develop a new technique. This method extracted the mirror symmetries along the incidence plane trading-off extra band folding. Though we have needed to perform some post-processing, this approach has given very accurate results, and we are confident that we can further improve this technique to make it more robust and easy to use.

We have next used Meep to simulate the propagation of incident light on our samples. We accomplished the goal of simulating a plane wave at any incidence angle, and we have studied the generated reflection response of a porous structure similar to our samples, trying to obtain the same dispersion map as from FT-IR measures.

We have achieved this feat with acceptable results. We were able to produce the dispersion map, and we could hint some photonic crystal bands. Nevertheless, the obtained results need to be improved. We studied different configurations to simulate a 33 μm thick silicon membrane with pores, polished back-face, adding an adaptation layer to the back-face, and infinite Si semi space. The first two lead to excessive oscillation at the output, thus the chosen configuration was the last. It is a valid approximation for our membrane, as the back-face of the sample is rough and scatters light, diminishing interference effects.

Meep can also be used to simulate the optical response of photonic crystals with metallic materials, in contrast to MPB, that only accepts dielectric materials. We will need to review and adapt the software to this purpose.

We have made some preliminary measures of nickel filled macroporous silicon with the FT-IR, and we have obtained highly interesting results. The measured optical response shows the desired optical filtering traits, and they have exceptional electrical and thermal properties.

VIII – Future Work

In the present work, we have studied, fundamentally, square structures of straight pores of differing length and layers: bulk, membrane, and porous membrane. During this process, some samples with special characteristics were measured, which are related to the novel macroporous silicon and nickel combination, with the intention to create optical filters.

These MpSi-Ni samples were measured with our FTIR to characterize them and see if it was possible to attain the desired response. This is a new development, and improvements are being made constantly. Sample availability has been scarce, but this has changed, and with a new batch of samples, we are ready to perform theoretical studies and simulations. The process is giving good results, and new metal fillings will be experimented to get better results and knowledge.

As previously said, we have now at our disposal new nickel filled samples, and the next steps are going to be focused on the theoretical study of them. Currently, we cannot use MPB to quickly obtain the most interesting data, that is, the photonic band gaps of the Nickel photonic crystals, as there is no support for metallic materials. This reverts to the usage of Meep to perform a more tedious analysis and extract the sought information. Nevertheless, bibliography research has revealed some experimental algorithms that allow using such materials for the Plane Wave Expansion method.

Implementing this software will boost our simulation capabilities. Aside from the study of metal-filled pores, there is also an area that has not been sufficiently explored yet in the present work. We refer to the modulated, 3D macroporous silicon photonic crystals. It is known that to obtain a full band-gap, we need the so-called “inverse opal” structure, which is an array of interconnected void spheres in s substrate. Even more, we will explore other interesting 3D structures, such as chirped pores that allow for an extended band gap. It is expected that both tasks will be taken in parallel, as most nickel samples produced are modulated.

On another hand, related to the emission properties of photonic crystals, they promise the ability to radiate in specific spectral ranges, as the photonic band gaps forbid light propagation enhancing other spectrum areas. This is a way to obtain highly efficient thermal radiators, with an output engineered to our needs.

In other respects, remark what has already been said, most of our samples are simple square lattice. There is a need to investigate alternative choices for the lattice. The study of symmetries and degeneracy can lead to improved photonic band gaps. The aim is the use of simple pattern arrays, like honeycomb or triangular, and different pore shapes, to widen our knowledge of photonic crystals and their behaviour. In relation to this, innovative fabrication and etching methods will be pursued.

In addition, it is of outmost interest the study of defect inclusion in the fabricated crystals. As a significant number of important papers demonstrate, photonic crystals are useful to produce light-guides, couplers, and cavity resonators[1,5,7,10]. We can create high quality optical elements, and study its behaviour, the interconnection and overall performance.

Some effort has already been made in the biological and sensing areas with macroporous silicon. The use of the optical properties of such devices will increase the type and enhance usability of every-day and industrial devices, like ultra fast and reliable gas sensors.

Moreover, micro-fluidics and biological sensing are a very active topic in the Department. Macroporous Silicon Photonic Crystal will play an important role, owing the cheap fabrication and readiness of materials. Some creative ideas have appeared in the Department to create devices with immediate application in these areas.

IX – REFERENCES

[1] A. Birner, R. B. Wehrspohn, U. M. Gösele, K. Busch, "*Silicon-Based Photonic Crystals*." Adv. Mater., Vol. 13, No. 6, March 2001.

[2] C. C. Katsidis, D. I. Siapkas, "*General transfer-matrix method for optical multilayer systems with coherent, partially coherent, and incoherent interference*." Appl. Opt., Vol. 41, No. 19, 2002.

[3] C. J. Bradley, A. P. Cracknell, *The Mathematical Theory of Symmetry in Solids*. Calrendon press, Oxforfd 1972.

[4] E. Yablonovitch, "*Photonic band-gap structures*." J. Opt. Soc. Am. B, Vol. 10, No. 2, 1993.

[5] H. Föll, M. Christophersen, J. Carstensen, G. Hasse, "*Formation and application of porous silicon*." Mater. Sci. & Eng. R, Vol. 39, No. 4, 2002.

[6] J. B. Pendry "*Photonic band structures*." J. Modern Opt., Vol. 41, No. 2, 1994.

[7] J. D. Joannopoulos, S. G. Johnson, J. N. Winn, R. D. Meade, *Photonic Crystals. Molding the flow of light*. 2nd ed., Princeton University Press, 2008.

[8] J. Gronholz, W. Herres, "*Understanding FT-IR Data Processing. Part 2: Details of the spectrum calculation*." Instruments & Computers, Vol. 3, No. 10, 1985.

[9] J-M. Lourtioz, H. Benisty, V. Berger, J-M. Gerard, D. Maystre, A. Tchelnokov, *Photonic Crystals. Towards Nanoscale Photonic Devices*. 2nd ed., Springer, 2008.

[10] K. Busch, S. Lölkes, R. B. Wehrspohn, H. Föll, *Photonic Crystals*. Wiley-VCH, 2004.

[11] K. Sakoda, "*Symmetry, degeneracy, and uncoupled modes in two-dimensional photonic lattices*." Phys. Rev. B, Vol. 52, No. 11, 1995.

[12] K. Sakoda, *Optical Properties of Photonic Crystals*. Springer, 2001.

[13] M. Garín, T. Trifonov, A. Rodríguez, L. F. Marsal, R. Alcubilla, "*Optical properties of 3D macroporous silicon structures*." Mater. Sci. & Eng. B, Vol. 149, No. 3, 2008

[14] M. Garín, T. Trifonov, A. Rodríguez, R. Alcubilla, "*Infrared thermal emission in macroporous silicon three-dimensional photonic crystals*." Appl. Phys. Lett., Vol. 91, No. 18, 2007

[15] M. S. Dresselhaus, G. Dresselhaus, A. Jorio, *Group Theory Application to the Physics of Condensed Matter*. Springer-Verlag, Berlin, 2008.

[16] MIT, *Meep*. *User documentation*. [Online]. Available: http://ab-initio.mit.edu/wiki/index.php/Meep

[17] MIT, *MPB*. *User documentation*. [Online]. Available: http://ab-initio.mit.edu/wiki/index.php/MIT_Photonic_Bands

[18] P. Yeh, A. Yariv, C-S. Hong, "*Electromagnetic propagation in periodic stratified media. I. General theory*." J. Opt. Soc. Am., Vol. 67, No. 4, April 1977.

[19] P. Yeh, *Optical Waves in Layered Media*. Wiley, New York, 1988.

[20] S. Khorasani, B. Rashidian, "*Modified transfer matrix method for conducting interfaces*." J. Opt. A, Vol. 4, No. , 2002.

[21] S. L. Altmann, *Band Theory of Solids: an introduction from the point of view of symmetry*. Oxford University Press, 1991.

[22] S. Matthias, F. Müller, J. Schilling, U. Gösele, "*Pushing the limits of macroporous silicon etching*." Appl. Phys. A, Vol. 80, No. 7, 2005.

[23] T. Trifonov, M. Garín, A. Rodríguez, P.Ortega, L.F. Marsal, R. Alcubilla, "*Post etching shaping of macroporous silicon*." Proc. SPIE, Vol. 6593, 65931Q, 2007.

[24] T. Trifonov, *Photonic bandgap analysis and fabrication of macroporous silicon by electrochemical etching*. PhD Thesis, URV, Departament d'Enginyeria Electrònica, Elèctrica i Automàtica, 2004.

[25] V. Lehmann, *Electrochemistry of Silicon: Instrumentation, Science, Materials and Applications*. Wiley-VCH Verlag GmbH, 2002.

[26] V. Lehmann, H. Föll, "*Formation mechanism and properties of electrochemically etched trenches in n-type silicon*." J. Electrochem. Soc., Vol. 137, No. 2, 1990

[27] V. N. Astratov, R. M. Stevenson , I.S. Culshaw, D. M. Whittaker, M.S. Skolnick, T.F. Krauss, R.M. De La Rue, "*Reflectivity Studies of Photonic Band Structure Effects in Two-Dimensional Air/Semiconductor Lattices*." Phys. Stat. Sol. (a), Vol. 178, No. 1, 2000.

[28] W. Herres, J. Gronholz, "*Understanding FT-IR Data Processing. Part 1: Data acquisition and Fourier transformation*." Comp. Appl. Lab., Vol. 2, No. 216, 1984.

[29] Z. Král, J. Ferré-Borrull, T. Trifonov, L. F. Marsal, A. Rodriguez, J. Pallarés, R. Alcubilla, "*Mid-IR characterization of photonic bands in 2D photonic crystals on silicon*." Thin. Solid films, Vol. 512, No. 22, 2008.