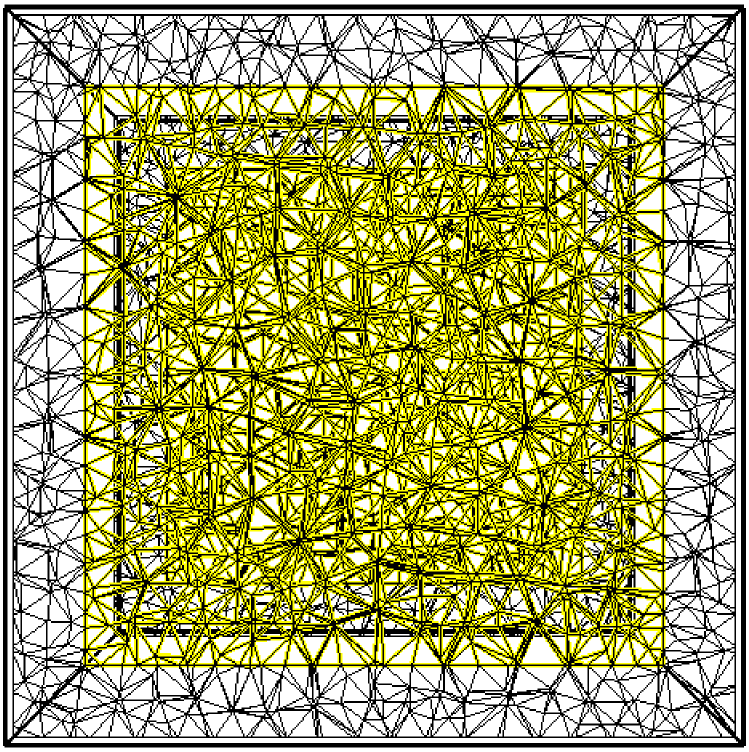
Accurate Modeling of Light Propagation inside Photonic Crystals of Increasing Geometric Complexity

Master Thesis



Jelmer Gietema
s1065211

Mathematics of Computational Science (MACS)
University of Twente
The Netherlands
12-01-2018

Student: J.G. Gietema
Student number: s1065211
Chairman: Prof. Dr. Ir. J.J.W. van der Vegt
Supervisor: Dr. Devashish
Supervisor: S.A. Hack, M.Sc.
Independent Member: Dr. K. Smetana

Cover: Front view of two cross sections in the xy-plane of the unstructured mesh of the Cuboid Crystal unit cell. Figure 14 shows the two cross sections from a different angle.

Contents

1 Introduction 5

2 Accurate Modeling of Light Propagation in 3D Periodic Nanophotonic Structures 8

2.1 Problem Formulation 8

2.2 Function Spaces 10

2.3 Mixed Formulation 11

2.4 Discontinuous Galerkin discretization 12

2.5 Weak Formulation 13

2.6 Interior Penalty Flux 19

2.7 Eigenvalue Problem Formulation 21

2.8 Implicit Divergence Constraint 23

3 Numerical Computations of the Generalized Eigenvalue Problem Implementation in hpGEM 27

3.1 Software Package hpGEM 27

3.2 Software Application DG-Max 28

3.3 Additional Toolkits 30

3.3.1 Structure Modeling 30

3.3.2 Mesh Generation 31

4 Results 32

4.1 Harmonic Solver 32

4.2 Eigenvalue Solver and Internal Meshes 32

4.2.1 Vacuum Crystal 33

4.2.2 Bragg Stack 34

4.3 Eigenvalue Solver and External Meshes 35

4.3.1 Cylindrical Crystal 36

4.3.2 Cuboid Crystal . . . 37
4.4 Convergence plots . . . 38
4.4.1 Vacuum Crystal . . . 38
4.4.2 Bragg Stack . . . 40
4.4.3 Cylindrical Crystal . . . 42
4.4.4 Cuboid Crystal . . . 43
4.4.5 Inverse Woodpile . . . 45

5 Conclusion and Outlook 50

6 Appendix A 51
6.1 Homogeneous Boundary Condition . . . 51
6.2 Periodic Boundary Condition . . . 53

7 Appendix B 54

8 References 58

1 Introduction

Light is crucial in life. Without light, we could not see, plants would not grow, and the earth would be a cold rock. The importance of light is easy to understand, however the understanding of light can be a hard subject. Over the past decades we gained more knowledge about light, but light still carries mysteries. If we understand light better, this can lead to big changes in the future. One can think of optical computers, where photons are used instead of electrons to store and transfer information [1], or using light in the distribution of quantum keys for sending encrypted information with the use of quantum cryptography [2, 3].

To understand the behaviour of light, it is necessary to look at the reflection, absorption, and propagation of light. One way to reflect, manipulate, trap, or even slow down light is using photonic crystals. A photonic crystal is a natural or artificial composite of dielectric materials with periodic structure on a length scale of the wavelength of light. Due to the periodicity of the structure, photonic crystals have a discrete translation symmetry, which means that they are invariant under translations of lengths that are multiples of a fixed step-size [4]. In vector notation, it can be described as

$$\epsilon(r) = \epsilon(r + l \cdot a), \tag{1.1}$$

where ϵ is the permittivity at position r in the photonic crystal, a is the lattice parameter, and l is an integer. Diffraction can occur for nearly all of the incoming light, if the crystal is perfectly ordered. In such a case, all the light of a specific wavelength, or of a range of specific wavelengths, is reflected. This light is not allowed from propagation at the boundary of the photonic crystal and can not penetrate into the material. Such an exclusion of a range of wavelengths in one direction is called a stop gap [4]. If light is forbidden to propagate in all three directions, it is called a band gap. Light within a photonic band gap, incident from any direction, is forbidden to propagate through the photonic crystal [4].

Geometries of materials with different repeated structures are shown in Fig. 1. The left example shows the Bragg Stack, a one-dimensional repeated material which produces a stop gap. The permittivity alternates between two values from layer to layer. The periodical differences in permittivity of layers cause the refractive index to differ per layer. If an incident wave hits the crystal, it is (partially) reflected. Depending on the wavelength and the periodicity of the refractive index in the crystal, the incident and reflected wave interfere constructively or destructively. These interference processes can lead to peculiar optical phenomena. Namely, if the incident wave has a wavelength equal to the lattice spacing of the crystal, the incident and reflected wave have the same wavelengths. However, the reflected wave travels in opposite direction of the incident wave. Interference of both waves causes a standing wave. The low frequency modes of this standing wave concentrate their energy in the regions with high permittivity, whereas the high frequency modes concentrate their energy in the regions with low permittivity. Both standing waves have the same wavelengths, but are concentrated in a medium with different dielectric permittivity and thus refractive index. These differences cause the

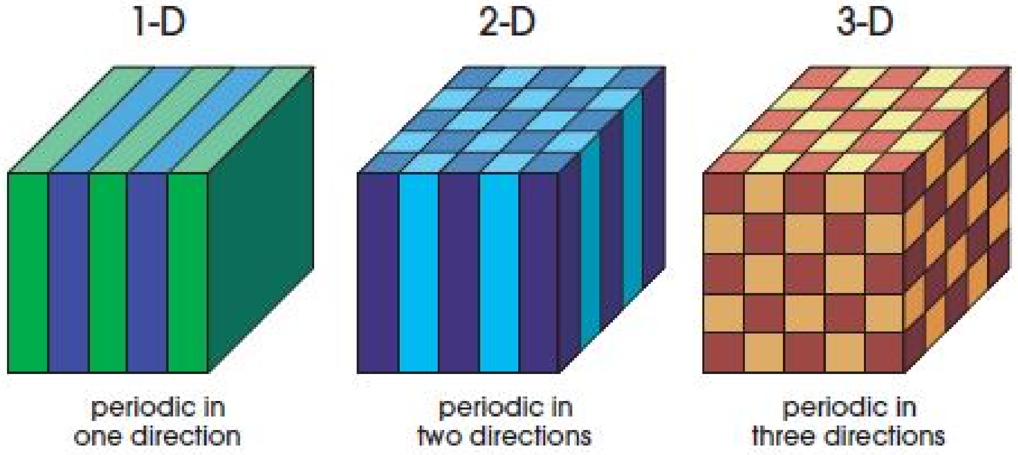


Figure 1: Examples of periodicity in materials, different colours indicate a different dielectric permittivity value. Figure courtesy: [4].

frequency to change. This can be obtained from the dispersion relation, given as

$$\omega = \frac{c}{n} \cdot k = \frac{c}{n} \cdot \frac{2\pi}{\lambda}, \tag{1.2}$$

with ω being the frequency of light, c the speed of light, n the refractive index, and λ the wavelength. This relation shows that differences in refractive index result in differences in frequency. With layers differing in permittivity, and as a result modes differing in frequencies, a stop gap is created. Hence, a region of frequencies is created where waves can not propagate in a certain direction. If this stop gap occurs in every direction it is called a band gap. An increase in difference in permittivity between the layers, will result in an increase in the region of frequencies of the stop or band gap [4]. The band gap is a range of forbidden frequencies, causing light, with those frequencies, to fully reflect independent of angle of incidence.

A photonic nanostructure known to have a large 3D band gap is the 3D Inverse Woodpile crystal. This 3D structure consists of two 2D arrays of identical pores running in the x and y direction. If the pores are taken to be low-indexed materials having a small dielectric permittivity in comparison to the surrounding structure, a highly-indexed material with large dielectric permittivity a band gap is obtained. The radius of the pores can differ, but is mostly taken between $r = 0.19a$ and $r = 0.245a$, with a the lattice parameter, in order to obtain a wide as possible 3D photonic band gap [34]. The unit cell of an Inverse Woodpile crystal is a cubic with lattice parameters a and c according to the ratio $\frac{a}{c} = \sqrt{2}$. In Fig. 2, the Inverse Woodpile is depicted with pores of radius $r = 0.19a$. We are interested in the Inverse Woodpile crystal, since the band gap can have a relative bandwidth up to 25.4% [34].

The reflectivity properties due to the band gap can be used in many different applications. In solar cells photonic crystals could be used as back reflector to reflect incident light. Photonic crystals can be applied in solar cells in such manner that light is perfectly reflected from every angle of incidence for a large range of frequencies. A higher reflection rate in solar cells causes more light to be absorbed, which increases the efficiency of

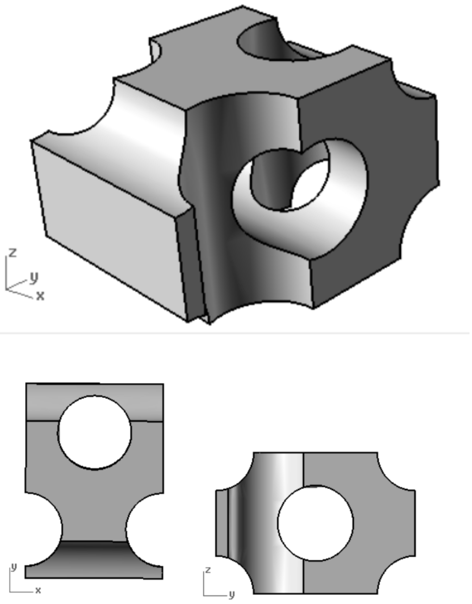


Figure 2: The Inverse Woodpile crystal is an example of a 3D photonic band gap crystal. In top view the whole unit cell is depicted. At the bottom left we have a view in xy plane and at the bottom right in yz plane of the Inverse Woodpile crystal.

the solar cell. In fibre-optics, a fibre could be coated by photonic crystals to keep light trapped inside with reduced or without loss [8]. Photonic crystals can also be found in nature. The butterfly Thecla Opisena uses this unusual hierarchical structure in its wings as camouflage [9].

Besides perfectly periodic photonic crystals, impurities can have many different applications. If the periodicity is intentionally broken, for example due to a single pore having a larger or smaller radius, a point defect cavity is created. Multiple point defects aligned in one direction can create a line defect. Point defects can cause single frequency modes to exist in the band gap. The defect mode cannot penetrate into the rest of the crystal, since it has a frequency in the band gap [4]. These structures with certain defects provide index guiding by total internal reflection in a direction normal to the plane of the crystal, as in a planar waveguide [10]. Light can be trapped by use of point and line defects and in special cases light can even be slowed down [11].

Photonic crystals have interesting properties, but are also complicated nano-structures. In combination with the size constraints on photonic crystals and the impact of impurities, production of the crystals is hard. The prediction of the optical properties of different photonic crystal structures is also useful. In order to minimize trial and error research work, numerical simulations are applied. In this report we investigate the properties of photonic crystals by accurately modelling the propagation of light inside nanophotonic structures.

2 Accurate Modeling of Light Propagation in 3D Periodic Nanophotonic Structures

2.1 Problem Formulation

Light is an electromagnetic wave [13]. Light propagation is described using the Maxwell equations, which express the electromagnetic propagation on a macroscopic level. In SI units, Maxwell's equations are given by

$$\nabla \cdot D(r,t) = \rho(r,t), \tag{2.1a}$$

$$\nabla \times E(r,t) + \frac{\partial B(r,t)}{\partial t} = 0, \tag{2.1b}$$

$$\nabla \cdot B(r,t) = 0, \tag{2.1c}$$

$$\nabla \times H(r,t) - \frac{\partial D(r,t)}{\partial t} = J(r,t), \tag{2.1d}$$

where E is the electric field and H is the magnetic field, D and B describe the displacement and magnetic induction fields, respectively, ρ is the free charge density, and J is the current density [4]. Equation (2.1a) is known as Gauss' law and describes the total electric flux out of a closed surface [14]. Equation (2.1b) is known as Faraday's law of induction, which describes the interaction of the magnetic field with its surroundings [15]. Equation (2.1c) is known as Gauss' law of Magnetism and states that the magnetic field is a solenoidal vector field [16]. Equation (2.1d) is Ampere's circuit law combined with Maxwell's addition, which relates the magnetic field to the electric current through a loop [14].

We assume the medium as the domain $\Omega \subset \mathbb{R}^3$. We consider this medium as the homogeneous dielectric material, where the dielectric permittivity is a function of position vector r. The structure does not vary with time, also there are no free charges. We use these approximations to relate the displacement field D and the electric field E. We do the same for the magnetic induction field B and magnetic field H [12].

Assuming the field strengths small enough, such that we have linear correlations, and the dielectric material being macroscopic and isotropic, i.e. $E(r,\omega)$ is related to $D(r,\omega)$ via a combination of the vacuum permittivity ϵ_0 and the relative permittivity $\epsilon(r,\omega)$, we get $D(r) = \epsilon_0\epsilon(r,\omega)E(r)$. Similarly, the definition of the relation between the magnetic induction field and the magnetic field is given by $B(r) = \mu_0\mu(r,\omega)H(r)$, with μ_0 the magnetic permeability of a vacuum and $\mu(r,\omega)$ the relative magnetic permeability

[4]. We implement these relations in the Maxwell's equations, (2.1), to obtain

$$\nabla \cdot \epsilon E(r,t) = \rho(r,t), \tag{2.2a}$$

$$\nabla \times E(r,t) = -\mu \frac{\partial H(r,t)}{\partial t}, \tag{2.2b}$$

$$\nabla \cdot \mu H(r,t) = 0, \tag{2.2c}$$

$$\nabla \times H(r,t) = J(r,t) + \epsilon \frac{\partial E(r,t)}{\partial t}, \tag{2.2d}$$

in the domain of the medium, i.e. $\Omega \subset \mathbb{R}^3$. Here, we take $\epsilon = \epsilon_0 \epsilon(r,\omega)$ describing the electric permittivity, and $\mu = \mu_0 \mu(r,\omega)$ describing the magnetic permeability. Also, $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})^T$ is the gradient operator. Note, if equation (2.2c) holds for the initial conditions and $\nabla \cdot J = 0$, then (2.2b) and (2.2d) will ensure that (2.2c) will always be satisfied. Therefore, we drop equation (2.2c) from the derivations, but keep it in mind at places where the consistency might be broken [7]. Since we assumed that there are no free charges, the free charge density, ρ, is set to zero. Hence, we simplify equation (2.2a), i.e., Gauss' Law, to

$$\nabla \cdot \epsilon E(r,t) = 0. \tag{2.3}$$

Since we assume the light propagation in the linear regime of the Maxwell equations, it gives us the possibility to separate time and space dependency in functions E, H, and J. We separate time and space dependency by expanding the fields into a set of harmonic modes. These modes represent the field patterns that vary harmonically, i.e. sinusoidally, with time [4]. We take the real part to obtain the physical fields and the complex part to construct the time dependent part of the functions. We write a harmonic mode as a spatial pattern times a complex exponential, i.e.

$$E(r,t) = \bar{E}(r) e^{-i\omega t}, \tag{2.4a}$$

$$H(r,t) = \bar{H}(r) e^{-i\omega t}, \tag{2.4b}$$

$$J(r,t) = \bar{J}(r) e^{-i\omega t}, \tag{2.4c}$$

where $i = \sqrt{(-1)}$, and ω is the angular frequency. Combining equations (2.3) and (2.4) with (2.2), we get

$$\nabla \cdot \epsilon \bar{E}(r) = 0, \tag{2.5a}$$

$$\nabla \times \bar{E}(r) = i\omega\mu \bar{H}(r), \tag{2.5b}$$

$$\nabla \times \bar{H}(r) = \bar{J}(r) - i\omega\epsilon \bar{E}(r). \tag{2.5c}$$

With equations (2.5b) and (2.5c) we make a combination between the electric field $\bar{E}(r)$ and magnetic field $\bar{H}(r)$. We divide equation (2.5b) by μ, such that we have

$$\mu^{-1} \nabla \times \bar{E}(r) = i\omega \bar{H}(r). \tag{2.6}$$

We take the curl on both sides of equation (2.6) and take iw in front of the curl at the right hand side, such that we get

$$\nabla \times (\mu^{-1} \nabla \times \bar{E}(r)) = i\omega(\nabla \times \bar{H}(r)). \tag{2.7}$$

We insert equation (2.5c) into (2.7), to obtain

$$\nabla \times (\mu^{-1} \nabla \times \bar{E}(r)) = i\omega \bar{J}(r) + \omega^2 \epsilon \bar{E}(r)). \tag{2.8}$$

We now have a function with the electric field $\bar{E}(r)$ as a free variable. To simplify, we take $j = i\omega \bar{J}(r)$ and get rid of the overhead bar. This gives us

$$\nabla \times (\mu^{-1} \nabla \times E(r)) - \omega^2 \epsilon E(r) = j. \tag{2.9}$$

Equation (2.9) formulates the electric field $E(r)$. We assume the domain Ω to have a perfectly conduction boundary. We apply a known current source j to calculate a harmonic solution of equation (2.9) [12]. Since we assume a perfectly conduction boundary, we have

$$n \times E = g \text{ at } \Gamma = \partial\Omega, \tag{2.10}$$

where $n \in \mathbb{R}^3$ denotes the outward normal unit vector at $\partial\Omega$, the boundary of the domain Ω. We denote the tangential trace of the electric field E by $g \in \mathbb{R}^3$, which is equal to zero in an ideal case.

Similar to the derivation of the electric field, a derivation of the magnetic field can be made, in which case we have a function with the magnetic field as a free variable. We only take the derivation of the electric field.

2.2 Function Spaces

In mathematics, function spaces describe the characteristics of a set of functions. In this report, we use multiple notations and function spaces. These are listed and defined in this section. We take an open domain Ω in $\mathbb{R}^2$ or $\mathbb{R}^3$ and denote the Hilbertian Sobolev space of real or complex scalar- or vector-valued functions by $H^s(\Omega)^d$, with $d = 1, 2$ or 3, and the regularity exponent $s \geq 0$. If $s = 0$, $H^s(\Omega)$ is given as $L^2(\Omega)$. We denote the standard inner product in $L^2(\Omega)$ as $(\cdot, \cdot)_\Omega$. The norm for the space $H^s(\Omega)^d, d = 1, 2$ or 3, is given by $|| \cdot ||_{s,\Omega}$ [12]. Furthermore, we use the next notations and function spaces

$$\begin{aligned}
H^1(\Omega) :=& \{v \in L^2(\Omega) : |\nabla v| \in L^2(\Omega)\}, && \text{(2.11a)}\\
H^1_0(\Omega) :=& \{v \in H^1(\Omega) : v = 0 \text{ at } \delta\Omega\}, && \text{(2.11b)}\\
H(\text{div}; \Omega) :=& \{v \in L^2(\Omega)^3 : \nabla \cdot v \in L^2(\Omega)\}, && \text{(2.11c)}\\
H_0(\text{div}; \Omega) :=& \{v \in H(\text{div}; \Omega) : v \cdot n = 0 \text{ at } \delta\Omega)\}, && \text{(2.11d)}\\
H(\text{div}^0; \Omega) :=& \{v \in H(\text{div}; \Omega) : \nabla \cdot v = 0 \in \Omega\}, && \text{(2.11e)}\\
H(\text{curl}; \Omega) :=& \{v \in L^2(\Omega^3) : \nabla \times v \in L^2(\Omega^3)\}, \text{ and} && \text{(2.11f)}\\
H_0(\text{curl}; \Omega) :=& \{v \in H(\text{curl}; \Omega) : n \times v = 0 \text{ at } \delta\Omega\}. && \text{(2.11g)}
\end{aligned}$$

2.3 Mixed Formulation

We consider the domain $\Omega \subseteq \mathbb{R}^3$ with the boundary Γ. We assume that the electric field $E \in H(\text{curl};\Omega) \cap H(\text{div}^0;\Omega) \oplus H_0^1(\Omega)$. Since we assume that the electric field is sufficiently smooth, we consider the $L^2(\Omega)^3$-orthogonal Helmholtz decomposition. The electric field function is written as

$$E = u + \nabla\phi, \tag{2.12}$$

with $u \in H(\text{curl};\Omega) \cap H(\text{div}^0;\Omega)$ and $\phi \in H_0^1(\Omega)$, where u represents the vector potential and ϕ represents the scalar potential of the system [12, 18, 19, 20, 21, 26]. The Helmholtz decomposition is orthogonal in $L^2(\Omega)^3$, which implies that $(u, \nabla\phi) = 0$ for all $\phi \in H_0^1(\Omega)$ and $u \in H(\text{curl};\Omega) \cap H(\text{div}^0;\Omega)$. We define $p := \omega^2\phi$, to simplify equation (2.9) to

$$\begin{aligned} j &= \nabla \times (\mu^{-1}\nabla \times \bar{E}(x)) - \omega^2\epsilon\bar{E}(x), \\ &= \nabla \times (\mu^{-1}\nabla \times (u + \nabla\phi)) - \omega^2\epsilon(u + \nabla\phi), \\ &= \nabla \times (\mu^{-1}\nabla \times u) - \omega^2\epsilon u - \epsilon\nabla p. \end{aligned}$$

Since $\phi \in H_0^1(\Omega)$, we reduce the perfectly conducting boundary condition at Γ, given by equation (2.10), to

$$\begin{aligned} g &= n \times (u + \nabla\phi) \\ &= n \times u. \end{aligned}$$

Similarly, we rewrite Gauss' Law to

$$\begin{aligned} 0 &= \nabla \cdot (\epsilon\bar{E}) && (2.13) \\ &= \nabla \cdot (\epsilon(u + \nabla\phi)) && (2.14) \\ &= \nabla \cdot (\epsilon u) + \nabla \cdot \nabla\phi. && (2.15) \end{aligned}$$

We use $\phi \in H_0^1(\Omega) \Rightarrow \phi = 0$ at Γ, to deduce that the scalar potential ϕ is also equal to zero in the domain Ω, since $\phi = 0$ is the unique solution of

$$\nabla \cdot \nabla\phi = \Delta p = 0 \text{ in } \Omega,$$

and

$$p = 0 \text{ at } \Gamma.$$

Assuming that $\nabla \cdot j = 0$ is valid, the modified Maxwell equations give us a model problem, which is to find a pair of (u, p), such that

$$\begin{aligned} \nabla \times (\mu^{-1}\nabla \times u) - \omega^2\epsilon u - \epsilon\nabla p &= j && \text{in } \Omega, && (2.16a) \\ \nabla \cdot (\epsilon u) &= 0 && \text{in } \Omega, && (2.16b) \\ n \times u &= g && \text{at } \Gamma, && (2.16c) \\ p &= 0 && \text{at } \Gamma. && (2.16d) \end{aligned}$$

2.4 Discontinuous Galerkin discretization

To accurate model the propagation of light, we discretize the medium into many smaller parts, called elements, such that the sum of these parts is the whole medium. We indicate the elements by K and take the permittivity ϵ to be constant inside each element. If we use more elements, the medium can be better approximated. However, calculations with more elements are computationally more expensive.

We employ the Discontinuous Galerkin Finite Element Method to do calculations with unstructured meshes generated with the Centaur mesh generator [12, 35]. We take a finite element mesh $\mathcal{T}_h$ of the domain Ω subdivided in tetrahedral elements. The granularity of the shape regular mesh $\mathcal{T}_h$ is indicated by h, i.e. $h = \max_{K\in\mathcal{T}_h}(h_K)$, where $h_K = \text{diameter}(K)$ for all $K \in \mathcal{T}_h$. We assume that the meshes are aligned with the discontinuities in the coefficients μ and ϵ. The set $\mathcal{F}_h^i$ represents all internal faces in $\mathcal{T}_h$, and $\mathcal{F}_h^b$ represents all boundary faces in $\mathcal{T}_h$. Therefore, the set of all faces is represented as $\mathcal{F}_h = \mathcal{F}_h^i \cup \mathcal{F}_h^b$ [12].

We approximate the scalar- and vector-valued functions p and u, respectively, in the discontinuous finite element spaces Q_h and V_h, where

$$Q_h := \{q \in L^2(\Omega) : q|_K \in P^l(K), \forall K \in \mathcal{T}_h\}, \tag{2.17}$$

$$V_h := \{v \in L^2(\Omega)^3 : v|_K \in S^l(K), \forall K \in \mathcal{T}_h\}, \tag{2.18}$$

for approximating order $l \geq 1$, with $P^l(K)$ the space of Lagrange polynomials of degree at most l on K, and $S^l(K)$ the space of Nédélec elements of first family [12, 25]. For the piecewise smooth vector- and scalar-valued functions $v \in V_h$ and $q \in Q_h$, we introduce the trace operators to extend the notion of restriction of the function to the boundary of the domain [17]. We state that $F \in \mathcal{F}_h^i$ is an internal face shared by two elements K^L and K^R. Here the upper script notation L indicates the element to the left, and R indicates the element to the right of face F. We define n^L as the unit outward normal vector of element K^L at a face F and n^R as the unit outward normal vector of element K^R at a face F. Note that $n^L = -n^R$. We take the tangential and normal jump of a vector-valued function v at $F \in \mathcal{F}_h^i$ by

$$[[v]]_T \equiv n^L \times v^L + n^R \times v^R,$$
$$[[v]]_N \equiv n^L \cdot v^L + n^R \cdot v^R,$$

where the tangential jump is indicated by subscript T and the normal jump is indicated by subscript N. The normal jump of a scalar-valued function q is defined by

$$[[q]]_N \equiv n^L q^L + n^R q^R.$$

Similarly, we specify the averages of v and q by

$$\{\{v\}\} \equiv \frac{v^L + v^R}{2},$$
$$\{\{q\}\} \equiv \frac{q^L + q^R}{2}.$$

For the boundary faces $F \in \mathcal{F}_h^b$, we define the tangential jump and normal jump of the vector-valued function v by the element to the left. Hence, we have

$$\begin{aligned} [[v]]_T &\equiv n^L \times v^L, \\ [[v]]_N &\equiv n^L \cdot v^L. \end{aligned}$$

Similarly, we define the jumps in the same way, as the averages. For a scalar-valued function q, we define the normal jump to be

$$[[q]]_N \equiv n^L q^L,$$

and we define the averages on the boundary as

$$\begin{aligned} \{\{v\}\} &\equiv v, \\ \{\{q\}\} &\equiv q. \end{aligned}$$

Later on we will eliminate the auxiliary variables in the discontinuous Galerkin discretization using lifting operators [12, 26, 27]. We define the lifting operators as

$$(L(u), w)_\Omega \equiv \sum_{F \in F_h^i} \int_F \{\{u\}\} \cdot [[w]]_T dS, \tag{2.19}$$

and

$$(S(u), w)_\Omega \equiv \sum_{F \in F_h} \int_F [[u]]_T \cdot \{\{w\}\} dS. \tag{2.20}$$

The operator ∇_h denotes the elementwise spatial derivative ∇ operator.

2.5 Weak Formulation

We will introduce the weak form of our problem to obtain the discontinuous Galerkin Discretization [12, 20, 22]. With a weak formulation the continuity requirements are reduced on the basis functions, which allows the use of simplified polynomials. We use the Ainsworth-Coyle basis functions or Nédélec basis functions [7, 5, 24, 25]. With the weak formulation, we approximate the exact results accurately, when the right mesh refinement is used. To obtain the weak formulation for the set of equations (2.16), an auxiliary variable M is introduced, such that

$$\nabla \times M - \omega^2 \epsilon u - \epsilon \nabla p = j \text{ in } \Omega, \tag{2.21}$$

with the assumption that

$$M - \mu^{-1} \nabla \times u = 0 \text{ in } \Omega. \tag{2.22}$$

With the introduction of test functions $v, w \in V_h$, we elaborate the weak formulation. Multiplying test function v with equation (2.21) and test function w with (2.22), replacing M with $M_h \in V_h$, u with $u_h \in V_h$, and p with $p_h \in Q_h$, and integrating over the domain

Ω, we obtain

$$(j, v)_\Omega = \sum_{K \in \tau_h} \int_K ((\nabla \times M_h) \cdot v - \omega^2 \epsilon u_h \cdot v - \epsilon \nabla p_h \cdot v) dx, \tag{2.23}$$

and

$$(M_h, w)_\Omega = \sum_{K \in \tau_h} \int_K (\mu^{-1}(\nabla \times u_h)) \cdot w dx, \tag{2.24}$$

where $(.,.)_\Omega$ denotes the standard $L^2(\Omega)$ inner product.

Next, we use the following identity for a matrix A

$$(A\vec{b}) \cdot \vec{c} = (A\vec{b})^T \vec{c} = \vec{b}^T A^T \vec{c} = \vec{b} \cdot (A^T \vec{c}), \tag{2.25}$$

where the upperscript T indicates the transpose of a vector or matrix. If A is symmetric, which is the case for μ^{-1}, this identity results in

$$(A\vec{b}) \cdot \vec{c} = \vec{b} \cdot (A\vec{c}). \tag{2.26}$$

We also consider the next identity, with B and C as vector quantities,

$$(\nabla \times B) \cdot C = \nabla \cdot (B \times C) + (\nabla \times C) \cdot B. \tag{2.27}$$

Finally, we introduce the divergence theorem

$$\int_V (\nabla \cdot D) dV = \int_S (D \cdot n) dS, \tag{2.28}$$

where V is the volume over which we integrate, D represents a continuously differentiable vector field, n is the outward pointing unit normal field of the boundary of the volume, indicated by S

We use identities given in equations (2.26) and (2.27), to rewrite (2.24) to

$$(M_h, w)_\Omega = \sum_{K \in \tau_h} \int_K (\nabla \cdot (u_h \times \mu^{-1} w) + (\nabla \times (\mu^{-1} w)) \cdot u_h) dx, \tag{2.29}$$

and we use the divergence theorem, given in (2.28), to obtain

$$(M_h, w)_\Omega = \sum_{K \in \tau_h} \int_K (\nabla \times (\mu^{-1} w) \cdot u_h) + \sum_{K \in \tau_h} \int_\Gamma (u_h^* \times \mu^{-1} w) \cdot n dS, \tag{2.30}$$

where u_h^* denotes the numerical flux for u_h. The numerical flux u_h^* is introduced since $u_h \in V_h$ has a multivalued trace at the element faces. Using the identity, given by equation

(2.27), (2.30) can be rewritten to

$$\begin{aligned}(M_h, w)_\Omega = &\sum_{K\in\tau_h} \int_K ((\nabla \times u_h) \cdot \mu^{-1} w - \nabla \cdot (u_h \times \mu^{-1} w))dx \\ &+ \sum_{K\in\tau_h} \int_\Gamma (u_h^* \times \mu^{-1} w) \cdot n dS.\end{aligned} \tag{2.31}$$

Firstly, we apply the divergence theorem on the second term. Then we use the identity given in equation (2.27) on the right hand side. Using the fact that the test function w is zero outside element K by definition of a test function, we get

$$(M_h, w)_\Omega = \sum_{K\in\tau_h} \int_K (\nabla \times u_h) \cdot \mu^{-1} w dx + \sum_{K\in\tau_h} \int_\Gamma ((u_h^* - u_h) \times \mu^{-1} w) \cdot n dS \tag{2.32}$$

$$\Rightarrow (M_h, w)_\Omega = \sum_{K\in\tau_h} \int_K (\nabla \times u_h) \cdot \mu^{-1} w dx + \sum_{K\in\tau_h} \int_\Gamma (n \times (u_h^* - u_h)) \cdot \mu^{-1} w dS. \tag{2.33}$$

We will now introduce two more mathematical identities, which will be used several times in this chapter. Firstly, we introduce the expressions for the average and tangential jump, as defined in Section 2.4, into the right hand side of equation (2.33), such that we have

$$\begin{aligned}\sum_{K\in\tau_h} \int_\Gamma (n \times u) \cdot v dS = &\sum_{F\in F_h^i} \int_F ((n^L \times u_h^L) \cdot v_h^L + (n^R \times u^R) \cdot v_h^R) dS \\ &+ \sum_{F\in F_h^b} \int_F (n^L \times u_h^L) \cdot v_h^L dS.\end{aligned} \tag{2.34}$$

Since each internal face appears twice in the summation over all the faces ∂K and using $n^R = -n^L$, we immediately obtain our first identity: For any $u_h, v_h \in V_h$, we have

$$\begin{aligned}\sum_{K\in\tau_h} \int_\Gamma (n \times u) \cdot v dS = &\sum_{F\in F_h^i} \int_F ([[u]]_T \cdot \{\{v\}\} - [[v]]_T \cdot \{\{u\}\}) dS \\ &+ \sum_{F\in F_h^b} \int_F [[u]]_T \cdot \{\{v\}\} dS.\end{aligned} \tag{2.35}$$

Similarly, the second relation is derived by introducing the jumps and averages for vector- and scalar-valued functions, defined in Section 2.4, at the right hand side, and by using $n^R = -n^L$. This relation is used to simplify the discontinuous Galerkin discretization: $\forall u_h \in V_h, q_h \in Q_h$, we have

$$\begin{aligned}\sum_{K\in\tau_h} \int_\Gamma q_h u_h \cdot n dS = &\sum_{F\in F_h^i} \int_F ([[q_h]]_N \cdot \{\{u_h\}\} + [[u_h]]_N \cdot \{\{q_h\}\}) dS \\ &+ \sum_{F\in F_h^b} \int_F [[q_h]]_N \cdot \{\{u_h\}\} dS.\end{aligned} \tag{2.36}$$

Implementing the first identity in equation (2.35) into (2.33), we obtain

$$\begin{aligned}(M_h, w)_\Omega = &\sum_{K\in\tau_h}\int_K (\nabla\times u_h)\cdot\mu^{-1}w dx \\ &+\sum_{F\in F_h^i}\int_F ([[u_h^* - u_h]]_T\cdot\{\{\mu^{-1}w\}\} - [[\mu^{-1}w]]_T\cdot\{\{u_h^* - u_h\}\})dS \\ &+\sum_{F\in F_h^b}\int_F [[u_h^* - u_h]]_T\cdot\{\{\mu^{-1}w\}\}dS.\end{aligned} \tag{2.37}$$

With use of lifting operators, defined in equations (2.19) and (2.20), we rewrite (2.37) to

$$(M, w)_\Omega = \sum_{K\in\tau_h}\int_K (\nabla\times u)\cdot\mu^{-1}w dx + (S(u^* - u), \mu^{-1}w)_\Omega - (L(u^* - u), \mu^{-1}w)_\Omega \tag{2.38}$$

$$\Rightarrow (M, w)_\Omega = (\mu^{-1}\nabla_h\times u, w) + (\mu^{-1}S(u^* - u), w)_\Omega - (\mu^{-1}L(u^* - u), w)_\Omega. \tag{2.39}$$

Since $w\in V_h$ is an arbitrary test function, we deduce the expression to

$$M_h = \mu^{-1}\nabla_h\times u_h + \mu^{-1}S(u_h^* - u_h) - \mu^{-1}L(u_h^* - u_h), \tag{2.40}$$

almost everywhere in Ω. Next, using the identity given in equation (2.27), we rewrite (2.23) to

$$(j_h, v)_\Omega = \sum_{K\in\tau_h}\int_K (\nabla\cdot(M_h\times v) + M_h\cdot(\nabla\times v) - \omega^2\epsilon u_h\cdot v - \epsilon\nabla p_h\cdot v)dx. \tag{2.41}$$

We rewrite the first term on the right hand side, using the divergence theorem, applying the identity in equation (2.27), and using the fact that the test function v is zero outside element K by definition of a test function, we obtain

$$(j_h, v)_\Omega = \sum_{K\in\tau_h}\int_K (M_h\cdot(\nabla\times v) - \omega^2\epsilon u_h\cdot v - \epsilon\nabla p_h\cdot v)dx + \sum_{K\in\tau_h}\int_\Gamma (n\times M_h^*)\cdot v dS, \tag{2.42}$$

with M_h^* the numerical flux for M_h. Now, using the first identity, we can express the contribution from the element boundaries as a sum over all faces in $\mathcal{T}_H$.

$$\begin{aligned}(j_h, v)_\Omega = &(M_h, \nabla_h\times v)_\Omega - (\omega^2\epsilon u_h, v)_\Omega - (\nabla_h p_h, \epsilon v)_\Omega \\ &+\sum_{F\in F_h^i}\int_F ([[M_h^*]]_T\cdot\{\{v\}\} - [[v]]_T\cdot\{\{M_h^*\}\})dS \\ &+\sum_{F\in F_h^b}\int_F [[M_h^*]]_T\cdot\{\{v\}\}dS.\end{aligned} \tag{2.43}$$

Substituting the value of M_h, given in equation (2.39), into (2.43), we obtain

$$\begin{aligned}(j_h, v)_\Omega =&(\mu^{-1}\nabla_h \times u_h, \nabla_h \times v)_\Omega - (\omega^2 \epsilon u_h, v)_\Omega - (\nabla_h p_h, \epsilon v)_\Omega \\ &+(S(u_h^* - u_h), \mu^{-1}\nabla_h \times v)_\Omega - (L(u_h^* - u_h), \mu^{-1}\nabla_h \times v)_\Omega \\ &+ \sum_{F\in F_h^i} \int_F ([[M_h^*]]_T \cdot \{\{v\}\} - [[v]]_T \cdot \{\{M_h^*\}\})dS \\ &+ \sum_{F\in F_h^b} \int_F [[M_h^*]]_T \cdot \{\{v\}\} dS.\end{aligned} \tag{2.44}$$

Using the definition of the lifting operators in equations (2.19) and (2.20), we get

$$\begin{aligned}(j_h, v)_\Omega =&(\mu^{-1}\nabla_h \times u_h, \nabla_h \times v)_\Omega - (\omega^2 \epsilon u_h, v)_\Omega - (\nabla_h p_h, \epsilon v)_\Omega \\ &+ \sum_{F\in F_h} \int_F [[u_h^* - u_h]]_T \cdot \{\{\mu^{-1}\nabla_h \times v\}\} dS - \sum_{F\in F_h^i} \int_F \{\{u_h^* - u_h\}\} \cdot [[\mu^{-1}\nabla_h \times v]] dS \\ &+ \sum_{F\in F_h^i} \int_F ([[M_h^*]]_T \cdot \{\{v\}\} - [[v]]_T \cdot \{\{M_h^*\}\})dS + \sum_{F\in F_h^b} \int_F [[M_h^*]]_T \cdot \{\{v\}\}\end{aligned}$$

$$\begin{aligned}\Rightarrow (j_h, v)_\Omega =&(\mu^{-1}\nabla_h \times u_h, \nabla_h \times v)_\Omega - (\omega^2 \epsilon u_h, v)_\Omega - (\nabla_h p_h, \epsilon v)_\Omega \\ &+ \sum_{F\in F_h^i} \int_F ([[M_h^*]]_T \cdot \{\{v\}\} - [[v]]_T \cdot \{\{M_h^*\}\} \\ &+[[u_h^* - u_h]]_T \cdot \{\{\mu^{-1}\nabla_h \times v\}\} - \{\{u_h^* - u_h\}\} \cdot [[\mu^{-1}\nabla_h \times v]])dS \\ &+ \sum_{F\in F_h^b} \int_F ([[M_h^*]]_T \cdot \{\{v\}\} + [[u_h^* - u_h]]_T \cdot \{\{\mu^{-1}\nabla_h \times v\}\})dS.\end{aligned} \tag{2.45}$$

Using integration by parts for the following

$$\begin{aligned}-(\nabla_h p_h, \epsilon v)_\Omega &= -\sum_{K\in\tau_h} (\nabla p_h) \cdot \epsilon v dx \\ &= -\sum_{K\in\tau_h} \int_K (\nabla \cdot (p_h \epsilon v) - p_h \nabla \cdot (\epsilon v))dx \\ &= -\sum_{K\in\tau_h} \int_\Gamma p_h^* n \cdot \epsilon v dS + \sum_{K\in\tau_h} \int_K p_h \nabla \cdot (\epsilon v) dx,\end{aligned}$$

and using the second identity in equation (2.36), we get

$$\begin{aligned}(\nabla_h p_h, \epsilon v)_\Omega =& \sum_{F\in F_h^i} \int_F ([[p_h^*]] \cdot \{\{\epsilon v\}\} + [[\epsilon v]]_N \{\{p_h^*\}\})dS \\ &+ \sum_{F\in F_h^b} \int_F [[p_h^*]] \cdot \{\{\epsilon v\}\} dS - (p_h, \nabla_h \cdot (\epsilon v))_\Omega.\end{aligned} \tag{2.46}$$

Substituting this back into equation (2.45), we obtain

$$\begin{aligned}(j_h, v)_\Omega =&(\mu^{-1}\nabla_h \times u_h, \nabla_h \times v)_\Omega - (\omega^2 \epsilon u_h, v)_\Omega + (p_h, \nabla_h \cdot (\epsilon v))_\Omega \\ &- \sum_{F\in F_h^i} \int_F ([[p_h^*]]_N \cdot \{\{\epsilon v\}\} + [[\epsilon v]]_N \{\{p_h^*\}\}) dS \\ &- \sum_{F\in F_h^b} \int_F [[p_h^*]]_N \cdot \{\{\epsilon v\}\} dS + \sum_{F\in F_h^i} \int_F ([[M_h^*]]_T \cdot \{\{v\}\} - [[v]]_T \cdot \{\{M_h^*\}\} \\ &+[[u_h^* - u_h]]_T \cdot \{\{\mu^{-1}\nabla_h \times v\}\} - \{\{u_h^* - u_h\}\} \cdot [[\mu^{-1}\nabla_h \times v]]) dS \\ &+ \sum_{F\in F_h^b} \int_F ([[M_h^*]]_T \cdot \{\{v\}\} + [[u_h^* - u_h]]_T \cdot \{\{\mu^{-1}\nabla_h \times v\}\}) dS.\end{aligned} \tag{2.47}$$

Again, using the integration by parts for the terms consisting parameter p, we get

$$\begin{aligned}(j_h, v)_\Omega =&(\mu^{-1}\nabla_h \times u_h, \nabla_h \times v)_\Omega - (\omega^2 \epsilon u_h, v)_\Omega - (\nabla_h p_h, \epsilon v)_\Omega \\ &- \sum_{F\in F_h^i} \int_F ([[p_h^* - p_h]]_N \cdot \{\{\epsilon v\}\} + [[\epsilon v]]_N \{\{p_h^* - p_h\}\}) dS \\ &- \sum_{F\in F_h^b} \int_F [[p_h^* - p_h]]_N \cdot \{\{\epsilon v\}\} dS + \sum_{F\in F_h^i} \int_F ([[M_h^*]]_T \cdot \{\{v\}\} - [[v]]_T \cdot \{\{M_h^*\}\} \\ &+[[u_h^* - u_h]]_T \cdot \{\{\mu^{-1}\nabla_h \times v\}\} - \{\{u_h^* - u_h\}\} \cdot [[\mu^{-1}\nabla_h \times v]]) dS \\ &+ \sum_{F\in F_h^b} \int_F ([[M_h^*]]_T \cdot \{\{v\}\} + [[u_h^* - u_h]]_T \cdot \{\{\mu^{-1}\nabla_h \times v\}\}) dS.\end{aligned} \tag{2.48}$$

With this, we obtain a weak formulation of equation (2.16a) in Ω, as the first part of the system we want to solve given in (2.16).

For the divergence constraint, i.e. equation (2.16b),

$$\nabla \cdot (\epsilon u) = 0 \text{ in } \Omega,$$

the discontinuous Galerkin formulation can be obtained in a similar way. Multiplying equation (2.16b) with arbitrary test functions $q \in Q_h$, integrating over each element $K \in \mathcal{T}_h$ and using the second identity (2.36), gives

$$\begin{aligned}0 =& - \sum_{K\in\mathcal{T}_h} \int_K \nabla q \cdot (\epsilon u_h) dx + \sum_{F\in F_h^i} \int_F ([[q]]_N \cdot \{\{(\epsilon u_h^*)\}\} + [[(\epsilon u_h^*)]]_N \{\{q\}\}) dS \\ &+ \sum_{F\in F_h^b} \int_F [[q]]_N \cdot \{\{(\epsilon u_h^*)\}\} dS,\end{aligned} \tag{2.49}$$

in Ω.

For the remaining part of the system, i.e. equations (2.16c) and (2.16d), the discontinuous Galerkin formulation is given by solely replacing u and p by their discretized numerical fluxes u_h^* and p_h^*, respectively, and replacing g by its discretized form g_h, obtaining

$$n \times u_h^* = g_h, \tag{2.50}$$
$$p_h^* = 0, \tag{2.51}$$

at Γ.

2.6 Interior Penalty Flux

Our modified model problem, with the assumption that $\nabla \cdot j$ holds, where we needed to find a pair (u, p) such that the equations given in (2.16) hold, is in weak form given by finding a pair (u_h, p_h) such that (2.48), (2.49), (2.50) and (2.51) hold. We now define the numerical fluxes u_h^*, p_h^*, M_h^*. This can be done in many different ways, however we choose to use the Interior Penalty (IP) numerical fluxes defined in [12, 23, 21, 26, 22, 29]. For interior faces, i.e. $F \in F_h^i$, we have

$$u_h^* = \{\{u_h\}\}, \tag{2.52a}$$
$$M_h^* = \{\{\mu^{-1}\nabla_h \times u_h\}\} - a_F[[u_h]]_T, \tag{2.52b}$$
$$p_h^* = \{\{p_h\}\} - b_F[[\epsilon u_h]]_N, \tag{2.52c}$$
$$\epsilon u_h^* = \{\{\epsilon u_h\}\} - c_F[[p_h]]_N, \tag{2.52d}$$

with $a_F, b_F, c_F \in \mathbb{R}^+$ as penalty coefficients. For boundary faces, i.e. $F \in F_h^b$, we have

$$n \times u_h^* = g_h, \tag{2.53a}$$
$$M_h^* = \{\{\mu^{-1}\nabla_h \times u_h\}\} + a_F g_h - a_F[[u_h]]_T, \tag{2.53b}$$
$$p_h^* = 0, \tag{2.53c}$$
$$\epsilon u_h^* = \{\{\epsilon u_h\}\} - c_F[[p_h]]_N, \tag{2.53d}$$

with $a_F, c_F \in \mathbb{R}^+$ as penalty coefficients and $g_h \in V_h$ as the L_2 approximation of the boundary value at Γ.

Inserting equations (2.52) and (2.53) into (2.48), and using the relations for the jumps and averages, e.g. $[[\{\{u_h\}\}]] = 0$, and using $[[M_h^*]]_T \cdot \{\{v\}\} = -[[v]]_T \cdot \{\{M_h^*\}\}$ for

$F \in F_h^b$, we get

$$
\begin{aligned}
(j_h, v)_\Omega =& (\mu^{-1}\nabla_h \times u_h, \nabla_h \times v)_\Omega - (\omega^2 \epsilon u_h, v)_\Omega - (\nabla_h p_h, \epsilon v)_\Omega \\
&+ \sum_{F \in F_h^i} \int_F ([[p_h]]_N \cdot \{\{\epsilon v\}\} + b_F [[\epsilon v]]_N [[\epsilon v]]_N) dS \\
&+ \sum_{F \in F_h^i} \int_F (-[[v]]_T \cdot \{\{\mu^{-1}\nabla_h \times u_h\}\} + [[v]]_T \cdot a_F [[u_h]]_T - [[u_h]]_T \cdot \{\{\mu^{-1}\nabla_h \times v\}\}) dS \\
&+ \sum_{F \in F_h^b} \int_F (-[[v]]_T \cdot (\{\{\mu^{-1}\nabla_h \times u_h\}\} + a_F g_h - a_F [[u_h]]_T) \\
&+ (g_h - n \times u_h) \cdot \{\{\mu^{-1}\nabla_h \times v\}\}) dS
\end{aligned}
$$

$$
\begin{aligned}
\Rightarrow (j_h, v)_\Omega =& (\mu^{-1}\nabla_h \times u_h, \nabla_h \times v)_\Omega - (\omega^2 \epsilon u_h, v)_\Omega - (\nabla_h p_h, \epsilon v)_\Omega \\
&+ \sum_{F \in F_h^i} \int_F ([[p_h]]_N \cdot \{\{\epsilon v\}\} + b_F [[\epsilon v]]_N [[\epsilon v]]_N) dS \\
&+ \sum_{F \in F_h^i} \int_F (-\{\{\mu^{-1}\nabla_h \times u_h\}\} \cdot [[v]]_T - \{\{\mu^{-1}\nabla_h \times v\}\} \cdot [[u_h]]_T + a_F [[u_h]]_T \cdot [[v]]_T) dS \\
&+ \sum_{F \in F_h^b} \int_F (-\{\{\mu^{-1}\nabla_h \times u_h\}\} \cdot [[v]]_T - \{\{\mu^{-1}\nabla_h \times v\}\} \cdot [[u_h]]_T + a_F [[u_h]]_T \cdot [[v]]_T) dS \\
&+ \sum_{F \in F_h^b} \int_F (-a_F [[v]]_T \cdot g_h + g_h \cdot \{\{\mu^{-1}\nabla_h \times v\}\}) dS,
\end{aligned}
\tag{2.54}
$$

in Ω.

Applying the Interior Penalty flux for the divergence constraint on the faces $F \in F_h$ and eliminating double jumps and average operators, we get

$$
0 = -(\epsilon u_h, \nabla_h q)_\Omega + \sum_{F \in F_h} \int_F (\{\{(\epsilon u_h)\}\} \cdot [[q]]_N - c_F [[p_h]] \cdot [[q]]_N) dS, \tag{2.55}
$$

in Ω.

For the remaining two equations of the modified model, given in equations (2.50) and (2.51), we get

$$
n \times u_h = g_h, \tag{2.56}
$$

$$
p_h^* = 0, \tag{2.57}
$$

at Γ.

Therefore, a weak formulation of the time-harmonic mixed Maxwell equations with the divergence constraint can be written as follows: Find $(u_h, p_h) \in V_h \times Q_h$, such that

$\forall(v, q) \in V_h \times Q_h$

$$a_h(u_h, v) - (\omega^2 \epsilon u_h, v)_\Omega + b_h(v, p_h) = (j_h, v)_\Omega + d_h(g_h, v), \tag{2.58a}$$

$$b_h(u_h, q) - c_h(p_h, q) = 0, \tag{2.58b}$$

where we have

$$\begin{aligned} a_h(u_h, v) = &(\mu^{-1}\nabla_h \times u_h, \nabla_h \times v)_\Omega \\ &- \sum_{F \in F_h} \int_F (\{\{\mu^{-1}\nabla \times u_h\}\} \cdot [[v]]_T + \{\{\mu^{-1}\nabla \times v\}\} \cdot [[u_h]]_T) dS \\ &+ \sum_{F \in F_h} \int_F a_F [[u_h]]_T \cdot [[v]]_T dS + \sum_{F \in F_h^i} \int_F b_F [[\epsilon v]]_N [[\epsilon u_h]]_N dS, \end{aligned} \tag{2.59a}$$

$$b_h(v, p_h) = -(\nabla_h p_h, \epsilon v)_\Omega + \sum_{F \in F_h} \int_F [[p_h]]_N \cdot \{\{\epsilon v\}\} dS, \tag{2.59b}$$

$$c_h(p_h, q) = \sum_{F \in F_h} \int_F c_F [[p_h]]_N \cdot [[q]]_N dS, \tag{2.59c}$$

$$d_h(g_h, v) = \sum_{F \in F_h^b} \int_F (a_F [[v]]_T \cdot g_h - g_h \cdot \{\{\mu^{-1}\nabla \times v\}\}) dS. \tag{2.59d}$$

As described before, ∇_h denotes the elementwise ∇ operator. We note that the bilinear form a_h corresponds to the interior penalty discretization of the curl-curl operator with an additional normal jump term. The bilinear form b_h discretizes the divergence constraint in the mixed Maxwell formulation using a discontinuous Galerkin scheme. The bilinear form c_h is a stabilization term, which controls the jumps in the scalar potential p. The parameters a_F, b_F and c_F are positive stabilization parameters, which depend on the mesh size and the polynomial order of the basis functions.

2.7 Eigenvalue Problem Formulation

In the remainder, we consider light propagation in an infinite periodic domain Ω without external current sources, i.e. $j = 0$. Equations (2.59) give us the generalized eigenvalue problem in ω^2 as: Find $(u_h, p_h, \omega) \in V_h \times Q_h \times \mathbb{C}$, such that $\forall(v, q) \in V_h \times Q_h$,

$$a_h(u_h, v) + b_h(v, p_h) = (\omega_h^2 \epsilon u_h, v)_\Omega, \tag{2.60a}$$

$$b_h(u_h, q) - c_h(p_h, q) = 0. \tag{2.60b}$$

Reference [22] presents the complete convergence theory for the eigenvalue problem given by equations (2.60). To study a unit cell, infinitely repeated in every direction, we employ Bloch-Floquet periodic boundaries in the positive and negative X, Y, and Z directions for the domain Ω to describe an infinite periodic crystal and introduce the Bloch mode expansion

$$u_h(r) = e^{ik\cdot r}\tilde{u}_h(r), \tag{2.61}$$

$$p_h(r) = e^{ik\cdot r}\tilde{p}_h(r), \tag{2.62}$$

where the wave vectors $\tilde{u}$ and $\tilde{p}$ are periodic, such that the period fits in the domain Ω. Hence, we have

$$\tilde{u}_h(r) = \tilde{u}_h(r+R), \tag{2.63}$$

$$\tilde{p}_h(r) = \tilde{p}_h(r+R), \tag{2.64}$$

for all lattice vectors R. Introducing the Bloch mode expansion into equation (2.60), we obtain the eigenvalue problem: Find $(\tilde{u}_h, \tilde{p}_h, \omega_h) \in V_h \times Q_h \times \mathbb{C}$, such that $\forall (v,q) \in V_h \times Q_h$,

$$\begin{aligned} a_h(\tilde{u}_h, v) + b_h(v, \tilde{p}_h) &= (\omega_h^2 \epsilon \tilde{u}_h, v)_\Omega, \\ b_h(\tilde{u}_h, q) - c_h(\tilde{p}_h, q) &= 0, \end{aligned} \tag{2.65}$$

where

$$\begin{aligned} a_{k,h}(\tilde{u}_h, v) = {} & (\mu^{-1}\nabla_{k,h} \times \tilde{u}_h, \nabla_{k,h} \times v)_\Omega \\ & - \sum_{F\in F_h} \int_F (\{\{\mu^{-1}\nabla_k \times \tilde{u}_h\}\} \cdot [[v]]_T + \{\{\mu^{-1}\nabla_k \times v\}\} \cdot [[\tilde{u}_h]]_T) dS \\ & + \sum_{F\in F_h} \int_F a_F [[\tilde{u}_h]]_T \cdot [[v]]_T dS + \sum_{F\in F_h^i} \int_F b_F [[\epsilon v]]_N [[\epsilon \tilde{u}_h]]_N dS, \end{aligned} \tag{2.66a}$$

$$b_{k,h}(v, \tilde{p}_h) = -(\nabla_{k,h}\tilde{p}_h, \epsilon v)_\Omega + \sum_{F\in F_h} \int_F [[\tilde{p}_h]]_N \cdot \{\{\epsilon v\}\} dS, \tag{2.66b}$$

$$c_{k,h}(\tilde{p}_h, q) = \sum_{F\in F_h} \int_F c_F [[\tilde{p}_h]]_N \cdot [[q]]_N dS, \tag{2.66c}$$

with $\nabla_k = \nabla + ik$. Equation (2.65) represents the model problem to calculate a photonic band structure. The discontinuous Galerkin discretization of equation (2.65) results in the following matrices:

$$\begin{aligned} a_{k,h}(v_j, v_i) &\to A_{ij}, \\ b_{k,h}(v_i, q_j) &\to B_{ij}, \\ c_{k,h}(q_j, q_i) &\to C_{ij}, \\ (v_j, \epsilon v_i) &\to M_{ij}. \end{aligned}$$

Hence, the generalized eigenvalue problem, in terms of matrices, is given by

$$\begin{pmatrix} A & B \\ B^T & C \end{pmatrix} \begin{pmatrix} u_h \\ p_h \end{pmatrix} = \omega^2 \begin{pmatrix} M & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} u_h \\ p_h \end{pmatrix}, \tag{2.67}$$

which finds the smallest eigenvalue ω_h.

For an efficient computation of the photonic bandstructure, References [12, 22] report that it is beneficial to rewrite the generalized eigenvalue problem, given in equation (2.67), as

$$\begin{pmatrix} M & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} u_h \\ p_h \end{pmatrix} = \tilde{\omega}^2 \begin{pmatrix} A & B \\ B^T & C \end{pmatrix} \begin{pmatrix} u_h \\ p_h \end{pmatrix}, \tag{2.68}$$

with $\omega_h^2 = \frac{1}{\tilde{\omega}_h^2}$. The benefit of this formulation is the more efficiently computations of the largest eigenvalues of a matrix, than the non-zero eigenvalues close to zero as is generally done in photonic crystal band gap computations. This possibility arises due to the mixed formulation satisfying the divergence constraint, eliminating the large null space of the curl-curl operator. Hence, we will only obtain non-zero eigenvalues.

2.8 Implicit Divergence Constraint

A simplification of the eigenvalue problem derived in the previous subsections is achieved by implicitly enforcing the divergence constraint. We rather assume that $\nabla \cdot (\epsilon \bar{E})$ is equal to zero. Given a divergence free initial condition, equation (2.1a) will automatically be satisfied at all times at the continuous level. Hence, we neglect equation (2.1a) in the same way as we neglected (2.1c) in (2.1) [7]. Therefore, the modified model is reduced to equations (2.9) and (2.10), i.e.

$$\nabla \times (\mu^{-1} \nabla \times E) - \omega^2 \epsilon E = j \qquad \text{in } \Omega, \tag{2.69a}$$

$$n \times E = g \qquad \text{at } \Gamma. \tag{2.69b}$$

Similar to the derivation in Subsection 2.5, we introduce an auxiliary variable M such that equation (2.69a) is rewritten to

$$\nabla \times M - \omega^2 \epsilon E = j \text{ in } \Omega, \tag{2.70}$$

with the assumption that

$$M - \mu^{-1}(\nabla \times E) = 0. \tag{2.71}$$

Multiplying test function v with equation (2.70) and test function w with (2.71), replacing M with $M_h \in V_h$ and E with $E_h \in V_h$, and integrating over domain Ω, we obtain

$$(j, v)_\Omega = \sum_{K \in \mathcal{T}_h} \int_K (\nabla \times M_h) \cdot v - \omega^2 \epsilon E_h \cdot v dx, \tag{2.72}$$

and

$$(M_h, w)_\Omega = \sum_{K \in \mathcal{T}_h} \int_K (\mu^{-1}(\nabla \times E_h)) \cdot w dx, \tag{2.73}$$

where $(.,.)_\Omega$ denotes the standard $L^2(\Omega)$ inner product. We apply the same derivation as in equation (2.24), with only u_h replaced by E_h. Therefore, with the same definitions as given in Subsection 2.4, the auxiliary variable is rewritten to

$$M_h = \mu^{-1}\nabla_h \times E_h + \mu^{-1}S(E_h^* - E_h) - \mu^{-1}L(E_h^* - E_h). \tag{2.74}$$

We employ the method of Subsection 2.5 on equation (2.72). We use the divergence theorem and the identities given in Subsection 2.4. We substitute M_h and the Lifting operators back into the equation. Finally, we use integration by parts, to obtain

$$\begin{aligned}(j_h, v)_\Omega =&(\mu^{-1}\nabla_h \times E_h, \nabla_h \times v)_\Omega - (\omega^2\epsilon E_h, v)_\Omega \\ &+ \sum_{F\in F_h^i}\int_F([[M_h^*]]_T \cdot \{\{v\}\} - [[v]]_T \cdot \{\{M_h^*\}\} \\ &+[[E_h^* - E_h]]_T \cdot \{\{\mu^{-1}\nabla_h \times v\}\} - \{\{E_h^* - E_h\}\} \cdot [[\mu^{-1}\nabla_h \times v]])dS \\ &+ \sum_{F\in F_h^b}\int_F([[M_h^*]]_T \cdot \{\{v\}\} + [[E_h^* - E_h]]_T \cdot \{\{\mu^{-1}\nabla_h \times v\}\})dS.\end{aligned} \tag{2.75}$$

Next, we set the definition of the numerical fluxes [12, 27]. For interior faces $F \in F_h^i$, we have

$$E^* = \{\{E\}\}, \tag{2.76a}$$

$$M_h^* = \{\{\mu^{-1}\nabla E_h\}\} - a_F[[E_h]]_T, \tag{2.76b}$$

with $a_F \in \mathbb{R}^+$ as penalty coefficient. For boundary faces $F \in F_h^b$, we have

$$n \times E_h^* = g_h, \tag{2.77a}$$

$$M_h^* = \{\{\mu^{-1}\nabla E_h\}\} + a_F g_h - a_F[[E_h]]_T. \tag{2.77b}$$

Inserting these numerical fluxes into equation (2.75) to obtain the space discretization according to the interior penalty numerical flux, we get

$$\begin{aligned}(j_h, v)_\Omega =&(\mu^{-1}\nabla_h \times E_h, \nabla_h \times v)_\Omega - (\omega^2\epsilon E_h, v)_\Omega \\ &+ \sum_{F\in F_h^i}\int_F(-\{\{\mu^{-1}\nabla_h \times E_h\}\} \cdot [[v]]_T - \{\{\mu^{-1}\nabla_h \times v\}\} \cdot [[E_h]]_T + a_F[[E_h]]_T \cdot [[v]]_T)dS \\ &+ \sum_{F\in F_h^b}\int_F(-\{\{\mu^{-1}\nabla_h \times E_h\}\} \cdot [[v]]_T - \{\{\mu^{-1}\nabla_h \times v\}\} \cdot [[E_h]]_T + a_F[[_h]]_T \cdot [[v]]_T)dS \\ &+ \sum_{F\in F_h^b}\int_F(-a_F[[v]]_T \cdot g_h + g_h \cdot \{\{\mu^{-1}\nabla_h \times v\}\})dS.\end{aligned} \tag{2.78}$$

With this equation, we have a weak formulation of the time-harmonic mixed Maxwell equations, with the implicit divergence constraint, which can be written as: Find $E_h \in V_h$, such that $\forall v \in V_h$,

$$a_h(E_h, v) - (\omega^2\epsilon E_h, v)_\Omega = (j_h, v)_\Omega + d_h(g_h, v), \tag{2.79}$$

with

$$\begin{aligned} a_h(E_h, v) = {} & (\mu^{-1}\nabla_h \times E_h, \nabla_h \times v)_\Omega \\ & - \sum_{F\in F_h} \int_F (\{\{\mu^{-1}\nabla \times E_h\}\} \cdot [[v]]_T + \{\{\mu^{-1}\nabla \times v\}\} \cdot [[E_h]]_T) dS \\ & + \sum_{F\in F_h} \int_F a_F [[E_h]]_T \cdot [[v]]_T dS + \sum_{F\in F_h^i} \int_F b_F [[\epsilon v]]_N [[\epsilon E_h]]_N dS, \end{aligned} \tag{2.80a}$$

$$d_h(g_h, v) = \sum_{F\in F_h^b} \int_F (a_F [[v]]_T \cdot g_h - g_h \cdot \{\{\mu^{-1}\nabla \times v\}\}) dS. \tag{2.80b}$$

where ∇_h denotes the elementwise ∇ operator and $a_F, b_F \in \mathbb{R}^+$ are the penalty coefficients, which are positive stabilization parameters depending on the mesh size and polynomial order of the basis functions. In this special case, we again note that the bilinear form a_h corresponds to the interior penalty discretization of the curl-curl operator with an additional normal jump term.

Similar to Subsection 2.7, we formulate the eigenvalue problem, by assuming there is no external source, such that $j = 0$ and $g_h = 0$ on an infinite periodic domain Ω. Also, we similarly introduce the Bloch-Floquet periodic boundaries in the positive and negative X, Y, and Z directions for the domain Ω in order to consider a unit cell, which is infinitely many times repeated in all directions. Introducing the Bloch mode expansion, we get

$$E_h(r) = e^{ik\cdot r}\tilde{E}_h(r), \tag{2.81}$$

with wave vector k. Here $\tilde{E}_h$ is periodic, such that the period fits in the domain Ω. Hence, we have

$$\tilde{E}_h(r) = \tilde{E}_h(r + R), \tag{2.82}$$

for all lattice vectors R. Implementing this into equation (2.79) gives the generalized eigenvalue problem in ω_h^2 as: Find $(\tilde{E}_h, \omega_h) \in V_h \times \mathbb{C}$, such that $\forall v \in V_h$,

$$a_h(\tilde{E}_h, v) = (\omega_h^2 \epsilon \tilde{E}_h, v)_\Omega, \tag{2.83}$$

where

$$\begin{aligned} a_{k,h}(\tilde{E}_h, v) = {} & (\mu^{-1}\nabla_{k,h} \times \tilde{E}_h, \nabla_{k,h} \times v)_\Omega \\ & - \sum_{F\in F_h} \int_F (\{\{\mu^{-1}\nabla_k \times \tilde{E}_h\}\} \cdot [[v]]_T + \{\{\mu^{-1}\nabla_k \times v\}\} \cdot [[\tilde{E}_h]]_T) dS \\ & + \sum_{F\in F_h} \int_F a_F [[\tilde{E}_h]]_T \cdot [[v]]_T dS + \sum_{F\in F_h^i} \int_F b_F [[\epsilon v]]_N [[\epsilon \tilde{E}_h]]_N dS, \end{aligned} \tag{2.84}$$

with $\nabla_k = \nabla + ik$. Equation (2.83) represents the model problem to calculate a photonic band structure. The discontinuous Galerkin discretization of equation (2.69) results in the following matrices:

$$\begin{aligned} a_{k,h}(v_j, v_i) &\to A_{ij}, \\ (v_j, \epsilon v_i) &\to M_{ij}. \end{aligned}$$

Therefore, the generalized eigenvalue problem, in terms of matrices, is given to be

$$AE_h = \omega_h^2 M E_h, \tag{2.85}$$

which finds the eigenvalues ω_h.

3 Numerical Computations of the Generalized Eigenvalue Problem Implementation in hpGEM

We aim to solve a generalized eigenvalue problem $\mathcal{M}x = \lambda\mathcal{S}x$, as given in equation (2.85). We test different photonic crystal structures, where the geometries of these structures are defined by the dielectric permittivity function varying in space. To perform numerical computations, we use the software package hpGEM [28], which is able to solve the generalized eigenvalue problem. Besides hpGEM, multiple toolkits are used. The applications are elaborated in this section.

3.1 Software Package hpGEM

We solve the generalized eigenvalue problem $\mathcal{M}x = \lambda\mathcal{S}x$ using the C++ software package hpGEM [28]. The hpGEM software package is developed with multiple applications, e.g., DG-Max, to apply the discontinuous Galerkin methods to a variety of physical problems. It is specialized to solve partial differential equations for fluid mechanics and electromagnetism, among which we find the Maxwell equations [5, 7]. The hpGEM package uses toolkits PETSc [31] and SLEPc [32] to solve linear systems and eigenvalue problems, respectively [31, 32]. The hpGEM package provides a structured mesh generator. Additionally, it provides a Centaur[35] mesh reader to enable computations on complex structures with unstructured meshes. We use the commercial packages Rhinoceros [36] and Centaur [35] to create complex structures and to generate unstructured meshes, respectively. The hpGEM package also provides a wide range of features, such as:

- handling mesh geometries in one, two and three dimensions, with a variety of element shapes with homogeneous or periodic boundary conditions
- creating structured meshes
- reading unstructured Centaur meshes
- providing a variety of basis functions
- giving the possibility of h- and p- adaptation
- computing Jacobians and coordinate transformations from the reference element to the physical element for internally and externally generated meshes
- computing integrals, e.g., Gauss' integration rules up to at least order seven for all supported geometries are included in support of higher-order finite element discretizations
- including a global algebraic system assembly

– providing automated-routines to write a computed solution to Tecplot and Paraview files

The DG-Max application of the software package hpGEM is able to calculate the eigenvalue problem, the time-dependent Maxwell system, and the time-harmonic Maxwell system. Additionally, DG-Max can handle perfectly conducting and periodic boundary conditions, the Interior Penalty and Brezzi [27] flux, and geometries with a piecewise-constant permittivity ϵ.

3.2 Software Application DG-Max

We use the DG-Max solver to apply the discontinuous Galerkin methods to the Maxwell equations. A computation with the software application DG-Max starts with initializing all mesh data, a timer starts, and the input parameters, such as type of solver and basis functions, are read. DG-Max constructs a structured mesh with boundary conditions via the hpGEM kernel or DG-Max reads an unstructured mesh and boundary conditions from a pre-generated Centaur file. PETSc is initialised and the global matrices $\mathcal{M}$ and $\mathcal{S}$ are defined. A variable is initialized which indicates if the inverse of $\mathcal{M}$ needs to be calculated. Depending on the type of solver, the initial solution, the derivative of the initial solution, and the source term are defined. The next step is to compute the element matrices. If the inverse mass matrix is used, also the Jacobian and its derivatives are computed. For each element, the local mass matrices (or inverse mass matrices) and stiffness matrices are initialized and the actual values of the local matrices are computed using the integration routines provided by hpGEM. The same is done for all boundary and internal faces. With the use of a pre-defined solver, the linear system is solved and output is created.

To correctly use the DG-Max code, we use mainly three files to implement the settings. These three files are DG-Max.cpp, BaseExtended.cpp, and ElementInfos.cpp. DG-Max.cpp is the main file, where we initialise the mesh and the boundary conditions, call the element- and face integrations, indicate the type of basis functions and set the type of solver. In BaseExtended.cpp, we describe the solvers and also calculate the element- and face integrands. In ElementInfos.cpp, we define the dielectric function and the penalty parameter.

DG-Max.cpp - Starting in DG-Max.cpp, one states if an internal mesh must be generated or that an external mesh is used. The internal mesh is a structured mesh generated depending on the input parameter n with which the function is called[1]. The domain $\Omega = (0,1)^3$ is subdivided in $n * n * n$ sub-cubes, which are then split into five tetrahedra, four of which are congruent [7]. In total, there are $5n^3$ elements generated. The boundary conditions need to be set manually, before generating the mesh. In case of an external mesh, a '.hyb' file needs to be called. A build-in mesh reader[2] is used

---

[1] Changes made in DG-Max.cpp: MyMeshManipulator()

[2] Changes made in MeshManipulator_Impl.h: readCentaurMesh3d()

to read all the data from the given file. When the mesh is generated or read, the basis functions are initiated. In DG-Max, two basis functions are implemented, the Ainsworth Coyle basisfunctions and the Nédélec elements [25, 5]. The final setting in DG-Max.cpp is the type of problem which needs to be solved. DG-Max consists of three different solvers, namely the time-dependent, harmonic, and eigenvalue solver. Whereas the time-dependent solver is mainly used to check the code and can later on be used for different applications, the harmonic solver can be used to check different errors in the harmonic solutions, e.g., the $L^2(\Omega)^3$ error. Since we are interested in the eigenvalues of different nanophotonic structures, we use the eigenvalue solver. In addition, we use the harmonic solver to check if the code is working correctly.

BaseExtended.cpp - Three solvers are implemented in BaseExtended.cpp and all the corresponding computations are done to obtain the results. At the start of the eigenvalue solver function, the degrees of freedom per element are determined using the basis functions. In case of periodic boundary conditions, boundary blocks[3] are located in the global matrix to take into account the elements adjacent to the periodic boundaries. The mass matrix $\mathcal{M}$, stiffness matrix $\mathcal{S}$, and the local matrices are computed[4]. The eigenvalue problem is initialised using SLEPc[5]. If needed, the inverted mass matrix $\mathcal{M}^{-1}$ is assembled, to initialize the product with stifness matrix $\mathcal{M}^{-1}S$. If the implementation of DG-Max is used, rather than DivDG-Max, i.e. the divergence constraint is neglected, a target is set to find the eigenvalues in the neighbourhood of this target. If no target is used, it is possible to find solely zero eigenvalues, if we search for the smallest eigenvalues. The number of desired eigenvalues is adjustable. Since we use a finite mesh resolution and therefore higher eigenvalues have many nodes and peaks and will not be approximated accurately, we put it the number of desired eigenvalues to 24. A significant fraction of the spectrum consists of zero eigenvalues, therefore it may be necessary to calculate many more eigenvalues than the few non-zero eigenvalues we are interested in. The number of iterations depends on the number of different wave vector k-values we use. If we use one k-value, only one iteration is needed. If we use more k-values, the system will be recalculated per k-value. In case of the eigenvalue solver, we are interested in the eigenvalues throughout the Brillouin zone in the reciprocal space, i.e. the smallest primitive unit cell of the structure of interest [4]. We take up to 60 k-values to implement the k shift. We start with k equal to $(0, 0, 0)$, then 'shift' in the ΓX direction up to $(\pi, 0, 0)$ in 20 steps, after that we shift in the XS direction in 20 steps up to $(\pi, \pi, 0)$. Finally, we go straight up along the SR direction and increase k up to (π, π, π), again in 20 steps. This is illustrated in Fig. 3. At every iteration, local and global matrices are recalculated with the corresponding k-value. The output of the eigenvalue solver is a vector with all the calculated eigenvalues. In case of 60 iterations, this yields $1440 (= 24 * 60)$ eigenvalues.

ElementInfos.cpp - DG-Max handles geometries with a piecewise-constant dielectric function ϵ. We take the piecewise dielectric function ϵ constant in each element. The piecewise dielectric constant can differ strongly per element. Depending on the dielectric

---

[3] Changes made in BaseExtended.cpp: FindBoundaryBlocks()

[4] Changes made in BaseExtended.cpp: elementMassIntegrand(), elementStiffnessIntegrand(), elementSpaceIntegrand(), initialConditionsIntegrand(), initialConditionsDerivIntegrand(), faceStifnessIntegrand(), faceStiffnessIPIntegrand(), faceSpaceIntegrandIP(), faceStifnessIntegrandBR()

[5] Changes made in BaseExtended: SolveEigenvalues()

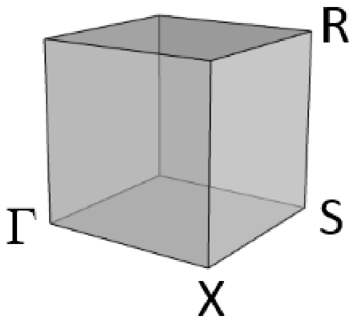


Figure 3: Visualization of the shift in wave vector k in reciprocal space. We start in Γ and shift to R via X and S, respectively.

function, we need mesh refinements to do proper calculations. We build the structures in such manner, that the grid is refined at the jumps in dielectric constants. We describe the piecewise dielectric function in ElementInfos.ccp. Here, the value of the dielectric constant is set per physical element[6]. In ElementInfos.cpp, we also calculate the penalty parameter. We adjust the penalty parameter using the input parameters. In case we use hpGEM to internally generate a structured mesh, the penalty parameter is set automatically to $n(p+1)(p+3)$, with n the number of sub-cubes in one direction and p the polynomial order of the basis function. However, if an external unstructured mesh is used, the penalty parameter must be set manually, accordingly to the smallest element in the mesh.

3.3 Additional Toolkits

The software package hpGEM only handles a cubic domain tessellated with tetrahedra, creating a structured mesh. This is easy to use for simple structures. However, it is less accurate when corners are involved or a local mesh refinement is needed. In the case of hpGEM mesh generation all the elements have the size of the smallest needed element size. An unstructured grid is a solution to efficiently deal with general geometries and local mesh refinement. The toolkits Rhinoceros[36] and Centaur[35] are commercial toolkits, which we use to create more complex structures and meshes.

3.3.1 Structure Modeling

We use Rhinoceros to create the structure of the nanophotonic crystal with a cubic unit cell of interest. We employ commands such as ‘Box’, ‘Circle’, ’Array’, and ‘MakeHole’. If structures consist of pores, we need to create an overlap of these pores to form the edge of the domain. This is necessary to apply the boundary conditions. If we do not create an overlap, it will be taken as empty space. We create a new outer structure and use the command ‘split’ to easily create these overlaps. We remove these double panels so that

---

[6] Changes made in ElementInfos.cpp: ElementInfos()

Centaur does not have to handle this, which takes more time and easily causes errors. The file needs to be saved as .igs file for use in Centaur. It needs to be saved as .3dm file if reopening in Rhinoceros is considered.

3.3.2 Mesh Generation

In Centaur the structure, created in Rhinoceros, needs to be loaded and a CAD cleaner is used to create the edges (panels) and boundaries of the edges (curves) of the data points of the structure. The boundary conditions are set. In case of periodic boundary conditions, Centaur uses periodic and shadow panels, which need to be combined. In case of multiple dielectric regions, an extra region needs to be created. The overlapping curves of the different regions need to be combined by use of interfaces. If done correctly, a mesh is generated. The input parameters for the mesh generation need to be set, such that refinement is performed correctly. It is necessary to put the number of pyramids to zero, such that solely tetrahedra are created. The output of the mesh generator consists of a large number of files and folders, but for the computations with DG-Max, only the .hyb file is used. This file is loaded into DG-Max as described in the previous Section. In case of an unstructured Centaur mesh the penalty parameter needs to be set manually using the input parameters. The penalty parameter is set to $n(p+1)(p+3)$, with p the polynomial order of the basis functions and n the multiplicity of the smallest element fitting in one direction.

4 Results

We present computations using DG-Max for various tests cases in this section. We perform a series of test calculations. We solve the eigenvalue problem given in equation (2.85) for the reduced frequency $\tilde{\omega}^2$, with $\tilde{\omega} = \omega a/2\pi c$, where ω is the frequency, a the lattice parameter, and c the speed of light. We express $\tilde{\omega}$ in units of (a/λ_w), with λ_w the wavelength. We are mainly interested in the smallest non-zero eigenvalues. Note, that we solve the problem for $\tilde{\omega}^2$ in (2.85), whereas the DG-Max eigenvalue solver solves the general eigenvalue problem in λ. Since $\tilde{\omega}^2 = \lambda$, we take the square root of λ in order to find $\tilde{\omega}$. In case periodic boundary conditions are considered, an additional division by 2π is needed to find the correct eigenvalues.

First, we analytically validate the code of DG-Max. After that, we perform convergence tests for the Vacuum Crystal, Bragg Stack, Cylindrical Crystal, Cuboid Crystal, and the Inverse Woodpile crystal. The Bragg Stack, Cylindrical, Cuboid, and Inverse Woodpile Crystal cases are compared to the MPB plane-wave expansion method [6].We compare the numerical results of the Vacuum crystal to the analytical solution. We calculate the convergence order of the Inverse Woodpile crystal and apply the Richardson extrapolation.

4.1 Harmonic Solver

The first validation of the software application DG-Max is performed using the harmonic solver of DG-Max. Given an exact solution, the source term is calculated by hand. With the source term as input, the exact solution is numerically calculated. The output of the harmonic solver is the error of the calculated solution compared to the exact solution in $L^2(\Omega)$, $H(\text{curl})$, and DG-norm. We define the $H(\text{curl})$ and DG-norm as

$$||E||_{\text{curl}}^2 = ||E||_0^2 + ||\nabla \times E||_0^2, \tag{4.1}$$

$$||E||_{\text{DG}}^2 = ||E||_0^2 + ||\nabla_h \times E||_0^2 + ||h^{-\frac{1}{2}}[[E]]||_{0,F_h}^2, \tag{4.2}$$

where $||\cdot||_0$ denotes the $L^2(\Omega)$-norm, $||\cdot||_{0,F_h}$ the $L^2(F)$ norm, and $h(x) = h_F$, which is the diameter of face F, containing x [29]. In References [5, 7, 29, 30] multiple tests for structured meshes are given. To validate the code, we repeat these tests of which the results are presented in the appendix, see Appendix A. We conclude that the DG-Max code works properly using the harmonic solver.

4.2 Eigenvalue Solver and Internal Meshes

We obtain the results of the eigenvalue problem using the eigenvalue solver of DG-Max. In order to validate the eigenvalue solver of DG-Max, we calculate the eigenvalues for common geometries. We consider the eigenvalue problem with the implicit divergence constraint, given in Section 2.8 by equation (2.85). Hence, we solve the generalized eigenvalue problem in $\tilde{\omega}^2$ for different dielectric permittivity functions ϵ.

4.2.1 Vacuum Crystal

The Vacuum Crystal is considered the simplest test case, since also an analytical solution can be calculated. We consider a homogeneous cube consisting of air with a constant dielectric permittivity $\epsilon = 1$ of which we calculate the 3D photonic band structures. The homogeneous cube is periodically repeated in all three directions using periodic boundary conditions. For validation, we analyse it by an analytical method and two numerical methods.

We calculate the analytical solution by taking a cube with dimensions $a \times a \times a$, with a the lattice parameter, where ϵ is constant throughout medium. Replacing ω with $\tilde{\omega}$, taking $\mu = 1$, assuming that E is constant and by taking k orthogonal to E, equation (2.85) is reduced to

$$ik \times ik \times E = \tilde{\omega}^2 E, \tag{4.3}$$

such that we find $||k||^2 = \tilde{\omega}^2$. Considering the Bloch-Floquet theory, we see that at least $||k + 2\pi l|| = \tilde{\omega}^2, \forall l \in \mathbb{Z}^3$, are eigenvalues of (4.3).

To find eigenvalues with DG-Max, we employ a structured mesh of 2560 tetrahedra, generated in hpGEM. We approximate the electric field E using first order hierarchic basis functions of $H(\text{curl})$ conforming finite elements as given by Ainsworth and Coyle [24]. We set the penalty coefficients to $a_F = 64$ and $b_F = 0$. In order to find the lowest eigenvalues and not solely zero eigenvalues, we set the target of the eigenvalue solver to $\lambda = 40$, to find eigenvalues in the neighbourhood of $\tilde{\omega} = 1.01$. Finally, the well-known MPB solver [6] for photonic band structures is used for comparison. Using the MPB plane-wave expansion method with a spatial resolution of $32 \times 32 \times 32 = 32768$ grid points, additional approximations for the photonic band structures are calculated [12].

In Fig. 4 the results of the analytical method, the DG-Max eigenvalue solver, and the MPB solver are shown for the 3D photonic band structure of the Vacuum Crystal between $\tilde{\omega} = 0$ and $\tilde{\omega} = 1.4$. The results of the DG-Max eigenvalues solver match with the analytical solution. However, at higher frequencies, there is a difference in numerically calculated eigenvalues and analytically calculated eigenvalues. These differences are not statistical errors, since they follow a systematic trend. Due to the inverse relation of wavelength and frequency, wavelength decreases with increasing frequency. To accurately resolve these higher frequency waves, the DG-Max eigenvalue solver needs a higher mesh resolution and the MPB eigenvalue solver needs a higher spatial resolution in order to reduce this systematic error. If the divergence constraint is not explicitly enforced, about 60% of the eigenvalues in the band structure are zero eigenvalues. DG-Max frequently finds these zero eigenvalues, which cause a major waste of computational time. We note that the MPB method computes several non-physical eigenvalues, which are not calculated in the analytical solution. These spurious eigenvalues are not present at low frequencies when the homogeneous cube is taken as a fictitious Bragg stack with alternating layers of dielectric permittivity $\epsilon_1 = 1$ and $\epsilon_2 = 1.0001$. However, the spurious eigenvalues are still present at higher frequencies [12].

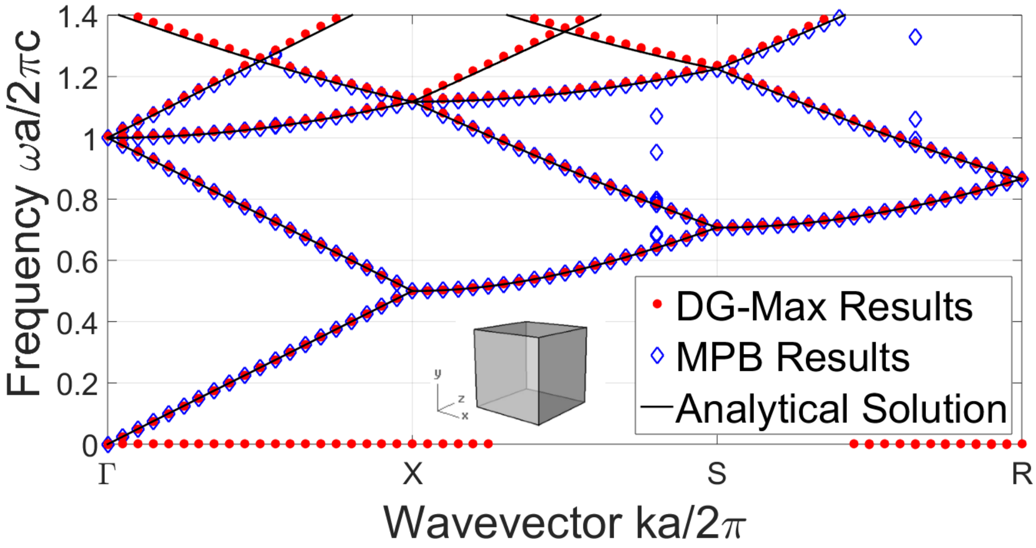


Figure 4: The 3D photonic band structure for a Vacuum Crystal with dielectric permittivity of $\epsilon = 1$. We computed the eigenvalues of the Vacuum Crystal case using the DG-Max solver with a structured mesh of 2560 elements, the MPB solver using a resolution of $32 \times 32 \times 32$ grid points, and the analytical solution. The red dots are the numerically calculated eigenvalues by the DG-Max eigenvalue solver. The open blue diamonds represent the numerically calculated eigenvalues by the MPB solver. The black solid line represents the solution to the analytical solution. A unit cell of the Vacuum Crystal is also presented.

4.2.2 Bragg Stack

To study the accuracy of the eigenvalue computations in the presence of material interfaces, we calculate the 3D photonic band structure of a Bragg stack, which is a cube consisting of alternating slabs, which differ in dielectric permittivity. We take one layer with width $0.5a$, with a being the lattice parameter, consisting of air with dielectric permittivity $\epsilon_1 = 1$. The other layer, with the same width, consists of a high-index material with dielectric permittivity $\epsilon_2 = 13$. Both layers are periodically repeated in the x-direction and are infinitely long in the other two directions y and z, using periodic boundary conditions.

We employ a structured mesh of 2560 tetrahedra, generated by hpGEM, in case of the DG-Max eigenvalue solver. The electric field E is again approximated using the first order of hierarchic basis functions of $H(\text{curl})$ conforming finite elements proposed by Ainsworth and Coyle [24]. The penalty coefficients are set to $a_F = 64$ and $b_F = 0$. The target of the eigenvalue solver is set to $\lambda = 5$, so that we find eigenvalues in the neighbourhood of $\tilde{\omega} = 0.36$. For comparison, the approximations of the 3D photonic band structures are also calculated using MPB plane-wave expansion method with a spatial resolution of $32 \times 32 \times 32 = 32768$ grid points.

Figure 5 shows the 3D photonic band structure between $\tilde{\omega} = 0$ and $\tilde{\omega} = 0.4$, obtained using the DG-Max eigenvalue solver and MPB solver. We observe that the band structures computed by the DG-Max solver match very well with the ones computed by the MPB method. However, we observe small differences at higher regions in frequency. This is an expected result, since a higher mesh resolution is required to resolve a solution with a smaller wavelength. We observe that the DG-Max computes many zero eigenvalues, since the divergence constraint is not explicitly enforced and hence, wastes a lot of computing time. The highlighted region in the ΓX direction of the reciprocal space indicates the stop gap of the Bragg Stack, see [4]. In the stop gap region, from $\tilde{\omega} = 0.15$ up to $\tilde{\omega} = 0.26$, no modes exist with a wave vector along the ΓX direction. Therefore all light with a frequency in this region is completely reflected if it falls upon the Bragg Stack structure in the ΓX direction.

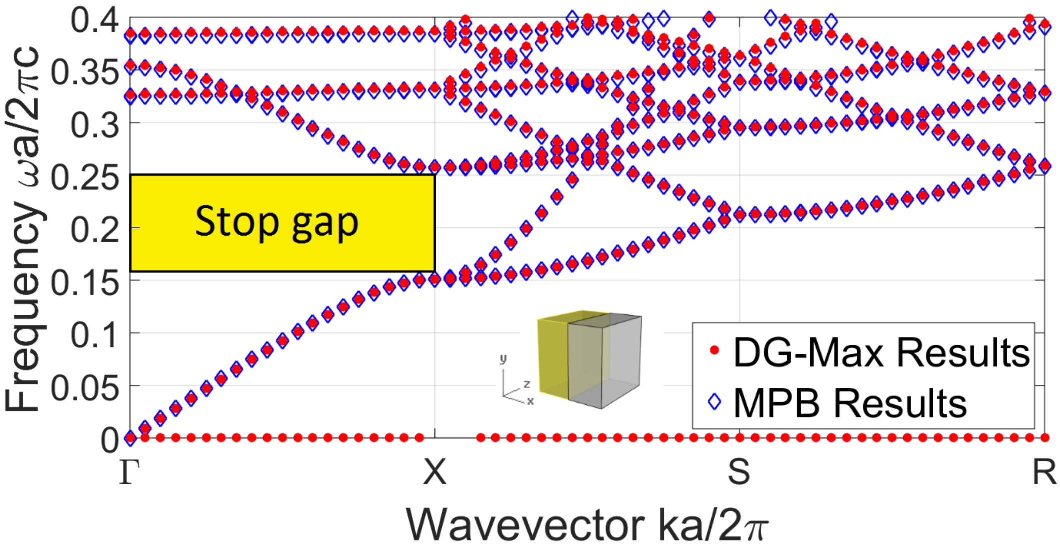


Figure 5: The 3D photonic band structure for the Bragg Stack with dielectric permittivity of $\epsilon_1 = 1$ and $\epsilon_2 = 13$. We computed the eigenvalues of the Bragg Stack using the DG-Max solver with a structured mesh of 2560 elements and the MPB solver using a resolution of $32 \times 32 \times 32$ grid points. The red dots are the numerically calculated eigenvalues by the DG-Max eigenvalue solver. The open blue diamonds represent the numerically calculated eigenvalues by the MPB solver. A unit cell of the Bragg Stack is also presented, with the yellow part indicating ϵ_2 and the grey part indicating ϵ_1.

4.3 Eigenvalue Solver and External Meshes

More complex structures are used to characterize the effect of the presence of a more complex material interface. We created complex structures using Rhinoceros and Centaur, as elaborated in Section 3.3. The externally created meshes are unstructured, in contrast with the internally generated structured meshes. We load the externally generated meshes in DG-Max, where a build-in mesh reader is used to read the mesh with the boundary conditions set in Centaur. All the settings used in Centaur, to generate the unstructured meshes, can be found in the appendix, see Appendix B. Besides the employed Vacuum Crystal and Bragg Stack, we test three more geometries. A cubic structure with centred cylinder as simplest testcase with an external mesh, a cubic cell with centred cuboid to employ the singularities at the corners of the cuboid inside the cube, and the Inverse Woodpile, since it can have a 3D band gap with a large relative band width [34].

4.3.1 Cylindrical Crystal

We compute the 3D photonic band structure, where the dielectric permittivity of $\epsilon_2 = 13$ is centred in a rod shape in the centre of the cube, with radius $0.2a$, with a the lattice constant. The surroundings of the rod is taken as air, therefore the dielectric permittivity is $\epsilon_1 = 1$. This cube is periodically repeated in the x- and y-direction and elongated in the z-direction, using periodic boundary conditions.

We employ an unstructured mesh of 3349 tetrahedra, generated using Centaur, in case of the DG-Max eigenvalue solver. The electric field E is approximated using the first order of hierarchic basis functions of $H(\text{curl})$ conforming finite elements proposed by Ainsworth and Coyle [12]. The penalty coefficients are set to $a_F = 640$ and $b_F = 0$. The target of the eigenvalue solver is set to $\lambda = 12$, so that we find eigenvalues in the neighbourhood of $\tilde{\omega} = 0.55$. The 3D photonic band structures are also calculated using the MPB plane-wave expansion method with a spatial resolution of $32 \times 32 \times 32 = 32768$ grid points.

The results of the 3D photonic band structure, between $\tilde{\omega} = 0$ and $\tilde{\omega} = 0.78$, of the square lattice of dielectric columns are shown in Fig. 6. These results are obtained using the DG-Max eigenvalue solver and MPB solver. We observe that the band structure computed by the DG-Max eigenvalue solver and MPB method match well. However, we observe small errors at higher regions in frequency. This conforms with our expectation that an eigenvalue solver needs a higher spatial or mesh resolution at higher frequencies. We observe that the DG-Max computes many zero eigenvalues, since the divergence constraint is not explicitly enforced.

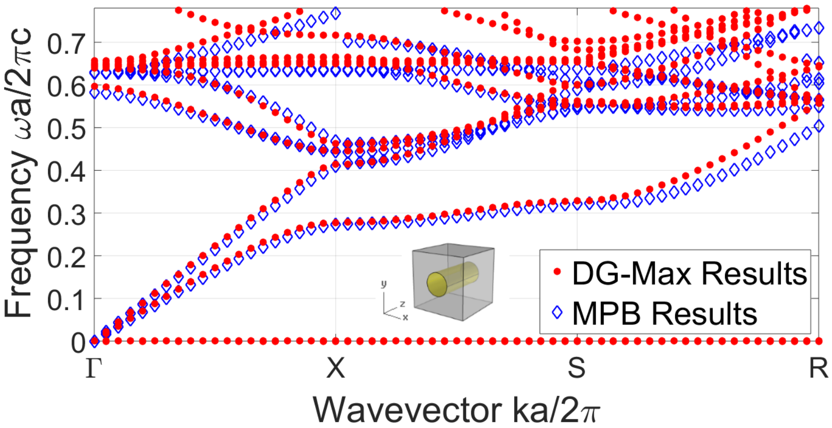


Figure 6: The 3D photonic band structure for the Cylindrical Crystal case with dielectric permittivity of $\epsilon_1 = 1$ and $\epsilon_2 = 13$. We computed the eigenvalues of the Cylindrical Crystal using the DG-Max solver with a unstructured mesh of 3349 elements and the MPB solver using a resolution of $32 \times 32 \times 32$ grid points. The red dots are the numerically calculated eigenvalues by the DG-Max eigenvalue solver. The open blue diamonds represent the numerically calculated eigenvalues by the MPB solver. A unit cell of the Cylindrical Crystal is also presented, where the yellow and grey parts indicate ϵ_2 and ϵ_1, respectively.

4.3.2 Cuboid Crystal

In the Cuboid Crystal case, we compute the 3D photonic band structure, where the dielectric permittivity of $\epsilon_2 = 13$ is centred square-like inside a cube. The length and width of the square in the x- and y-direction is $0.8a$, such that the surroundings have a width of $0.1a$ in both directions, where a is the lattice parameter. The surrounding is considered air and thus has dielectric permittivity $\epsilon_1 = 1$. This cube with inner square is periodically repeated in the x- and y-direction and elongated in the z-direction, using periodic boundary conditions. Due to the elongation in the z-direction, we thus have a cuboid inside a cube.

We employ an unstructured mesh of 3278 tetrahedra, generated using Centaur, in case of the DG-Max eigenvalue solver. The electric field E is approximated using the first order of hierarchic basis functions of $H(\text{curl})$ conforming finite elements proposed by Ainsworth and Coyle [12]. The penalty coefficients are set to $a = 80$ and $b = 0$. The target of the eigenvalue solver is set to $\lambda = 6$, so that we find eigenvalues in the neighbourhood of $\tilde{\omega} = 0.39$. The 3D photonic band structures are also calculated using MPB plane-wave expansion method with a spatial resolution of $32 \times 32 \times 32 = 32768$ grid points.

Figure 7 shows the results of the 3D photonic band structure of the Cuboid Crystal between $\tilde{\omega} = 0$ and $\tilde{\omega} = 0.34$. The results are obtained using the DG-Max eigenvalue solver and the MPB solver. We observe that the band structure computed by the MPB method match with the ones computed by the DG-Max solver. However, we observe errors at higher regions in frequency. This in agreement with our expectation that an eigenvalue solver needs a higher spatial or mesh resolution at higher frequencies in comparison with the MPB solver. We observe that the DG-Max solver computes many zero eigenvalues, since the divergence constraint is not explicitly enforced.

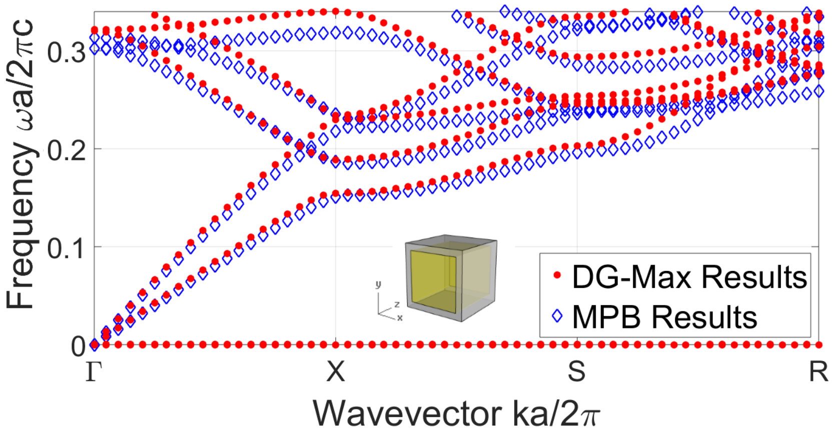


Figure 7: The 3D photonic band structure for the Cuboid Crystal case with dielectric permittivity of $\epsilon_1 = 1$ and $\epsilon_2 = 13$. We computed the eigenvalues of the Cuboid Crystal using the DG-Max solver with a structured mesh of 2560 elements and the MPB solver using a resolution of $32 \times 32 \times 32$ grid points. The red dots are the numerically calculated eigenvalues by the DG-Max eigenvalue solver. The open blue diamonds represent the numerically calculated eigenvalues by the MPB solver. A unit cell of the Cuboid Crystal is also presented, where the yellow part indicates ϵ_2 and the grey part indicates ϵ_1.

4.4 Convergence plots

To verify that the refinement of the mesh results in a more accurate eigenvalue calculation, we consider the convergence of the results obtained from the calculations by the DG-Max eigenvalue solver and the MPB solver. We study the convergence of a specific eigenvalue. We compute the eigenvalues corresponding to wave vector $k = (0, 0, 0)$ for different geometries. We test the same geometry using a sequence of increasingly refined meshes. We plot the obtained results against h_{est}, which is an estimation of the size of one element. Using the total number of elements N, we estimate the size of an element by

$$h_{est} = \frac{1}{\sqrt[3]{N}}. \tag{4.4}$$

We expect that for larger N, the calculated eigenvalues will be more accurate. Note, however, that the structure of each unstructured mesh for each value of N is different, which can affect the monotonicity of the convergence rate.

In this section we study all cases using external *unstructured* meshes generated by Centaur. In order to test the efficiency of the DG-Max eigenvalue solver, we also test to see how far we can increase the total number of elements N. We make a comparison with the MPB solver for the Cylinder, Cuboid, and Inverse Woodpile crystal. We compare the convergence order of both solvers for the Inverse Woodpile crystal using Richardson extrapolation.

4.4.1 Vacuum Crystal

In the Vacuum Crystal case, where we take a homogeneous cube consisting of air with a constant dielectric permittivity $\epsilon = 1$, we use mesh sizes of $N = 375$, 707, 3107, 6273, 11663, and 28351. Hence, we obtain $h_{est} = 0.139, 0.112, 0.069, 0.054, 0.044$, and 0.033, respectively. The target is set to $\lambda = 40$ for the DG-Max eigenvalue solver to find eigenvalues in the neighbourhood of $\tilde{\omega} = 1.01$.

We find a single eigenvalue with a multiplicity of 12, as well as several zero eigenvalues. These 12 eigenvalues are the approximation of the first eigenvalue $\tilde{\omega} = 1$ for the Vacuum Crystal. To compare the convergence of the eigenvalues, we take the largest computed eigenvalue for each different mesh size. Note that the largest computed eigenvalue is the worst approximation of the first eigenvalue $\tilde{\omega} = 1$.

In Fig. 8 the results of the largest computed eigenvalue of the Vacuum Crystal are obtained for different sized unstructured meshes. Note that we plot by h_{est}. The specific eigenvalue monotonically decreases to the first exact eigenvalue as the number of elements is increased. Hence, Fig. 8 shows a clear convergence with increasing mesh sizes. The increase in mesh refinement, resulting in up to more than 28000 elements, is handled correctly by the DG-Max solver. The same number of non-zero eigenvalues is calculated for different mesh sizes.

The analytical solution of the vacuum case is known, therefore we make a comparison between the approximated eigenvalues λ_h and the exact solution λ. We calculate an exact eigenvalue of $\lambda = 4\pi^2$, hence we find $\tilde{\omega} = 1$. We calculate the error E_h by

$$E_h = \frac{|\lambda - \lambda_h|}{\lambda}. \tag{4.5}$$

Figure 9 shows the error in numerical solution E_h against h_{est}. The error is decreasing for increasing total number of elements. Since the error decreases to zero, a better approximation of the exact solution is calculated using a refined mesh of the Vacuum Crystal. Note that this is the worst computed approximation of the exact eigenvalue.

We use the error and h_{est} to observe the convergence in logarithmic scale for comparison. In Fig. 10 the results of the error in numerical solution in logarithmic scale is presented.

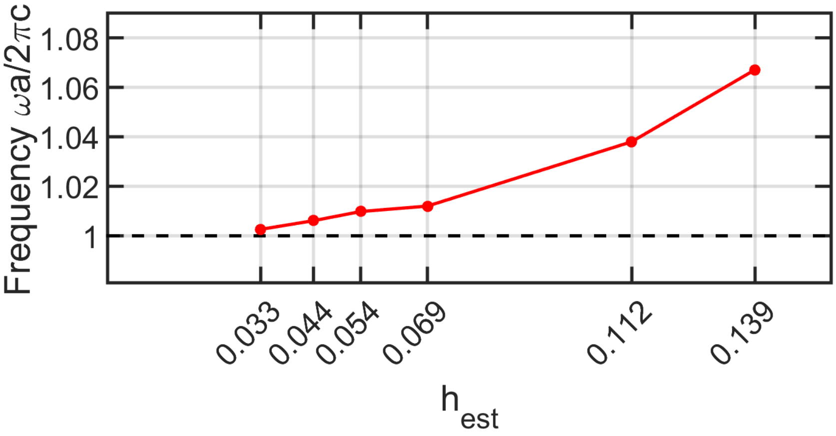


Figure 8: The largest computed eigenvalue by the DG-Max eigenvalue solver of the Vacuum Crystal for wave vector $k = (0,0,0)$. We set the target of the DG-Max solver to $\tilde{\omega} = 1$. We set the dielectric permittivity to $\epsilon = 1$. We use different sized unstructured Centaur meshes ranging from 375 elements up to 28351 elements. The estimation of the size of one element h_{est} is expressed as $1/\sqrt[3]{N}$. The red dots indicate the computed eigenvalues. The solid red line indicates the convergence. The dashed black line indicates the first exact eigenvalue $\tilde{\omega} = 1$ of the Vacuum Crystal.

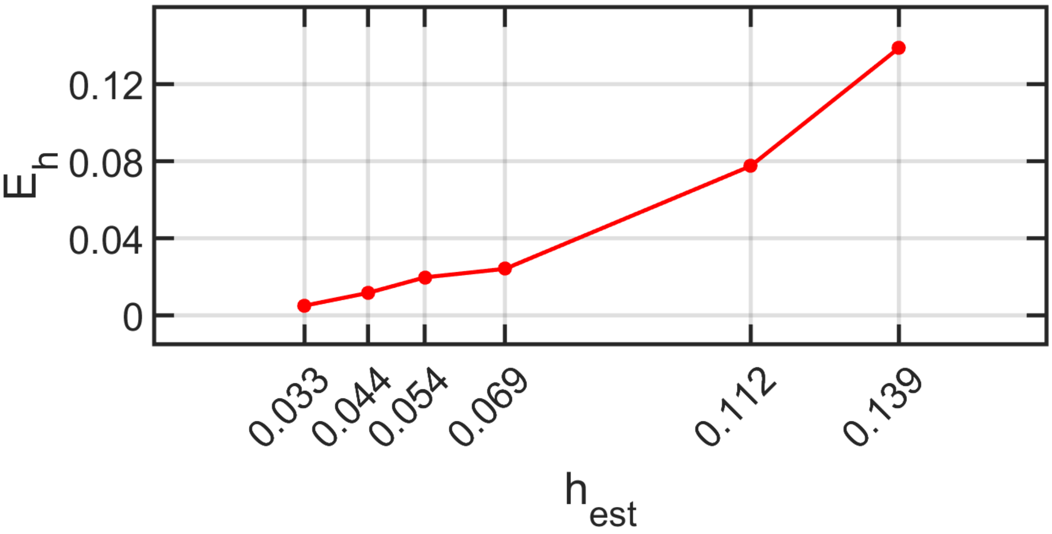


Figure 9: The error of the largest computed eigenvalue by the DG-Max solver of the Vacuum Crystal for wave vector $k = (0, 0, 0)$. We set the target of the DG-Max solver to $\lambda = 40$ to obtain eigenvalues in the neighbourhood of $\tilde{\omega} = 1$. We set the dielectric permittivity to $\epsilon = 1$. We use different sized unstructured Centaur meshes ranging from 375 elements up to 28351 elements. The error E_h is given by $|\lambda - \lambda_h|/\lambda$, with λ the analytical calculated eigenvalue and λ_h the numerically computed eigenvalue of the Vacuum Crystal. The estimation of the size of one element h_{est} is expressed as $1/\sqrt[3]{N}$. The red dots indicate the computed eigenvalues. The error E_h goes to zero for increased mesh sizes, indicated by the red solid line, therefore the computed eigenvalue λ_h converges to the correct eigenvalue λ.

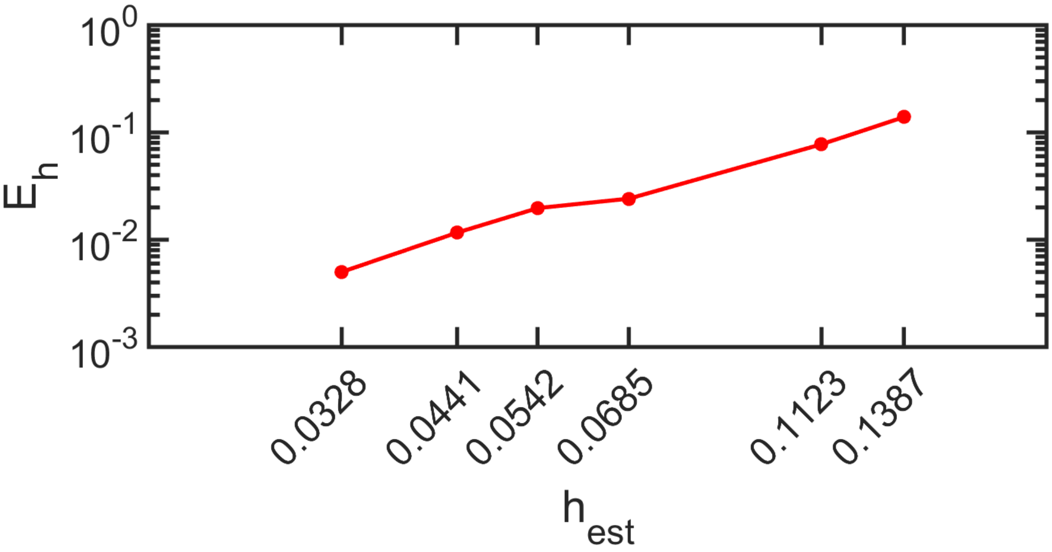


Figure 10: The error of the last computed non-zero eigenvalue by the DG-Max solver of the Vacuum Crystal for wave vector $k = (0, 0, 0)$ in logarithmic scale. We set the dielectric permittivity to $\epsilon = 1$. We use different sized unstructured Centaur meshes ranging from 375 elements up to 28351 elements. The error E_h is given by $|\lambda - \lambda_h|/\lambda$, with λ the analytically calculated eigenvalue and λ_h the numerically computed eigenvalue of the Vacuum Crystal. The estimation of the size of one element h_{est} is expressed as $1/\sqrt[3]{N}$. The red dots indicate the computed eigenvalues. The error E_h goes to zero for increased mesh sizes, indicated by the red solid line, therefore the computed eigenvalue λ_h converges to the correct eigenvalue λ.

4.4.2 Bragg Stack

We observe the convergence of the Bragg Stack for a single wave vector $k = (0, 0, 0)$. We take a cube consisting of alternating slabs which differ in dielectric permittivity. We take one layer with width $0.5a$, with a the lattice parameter, consisting of air with dielectric permittivity $\epsilon_1 = 1$. The other layer, with same width, consists of a high-index material with dielectric permittivity $\epsilon_2 = 13$. We use meshes with sizes $N = 375$, 707, 3107, 6273, 11663, and 28351 and thus have $h_{est} = 0.139, 0.112, 0.069, 0.054, 0.044$, and 0.033 for the Bragg Stack. Hence, we use the same number of elements as in the Vacuum Crystal case, only now with a different dielectric function ϵ. We set the target of the DG-Max solver to $\lambda = 5$ to obtain eigenvalues in the neighbourhood of $\tilde{\omega} = 0.36$.

The computations for the Bragg Stack with the eigenvalue solver result in diverse non-zero eigenvalues ranging from $\tilde{\omega} = 0.326$ to $\tilde{\omega} = 0.495$, rather than one single eigenvalue with a certain multiplicity. Also, several zero eigenvalues are found. For different mesh sizes, we find a different number of non-zero and zero eigenvalues, since the various externally generated unstructured meshes have a different element distribution. Computations done with meshes $N = 707, 6273, 11663$, and 28351 result in 14 non-zero eigenvalues, with $N = 3107$ we find 22 non-zero eigenvalues and for $N = 375$, we find 12 non-zero eigenvalues. We take the largest computed eigenvalue in each case for different mesh sizes for comparison. Since we find 8 additional non-zero eigenvalues in case of $N = 3107$, we neglect these 8 non-zero eigenvalues and take the largest eigenvalue of the remaining 14 non-zero eigenvalues. This specific eigenvalue is around $\tilde{\omega} = 0.44$. The convergence of the results obtained by the DG-Max solver is compared to the largest computed eigenvalue $\tilde{\omega} = 0.4369$ of the MPB solver using the wave vector $k = (0, 0, 0)$ and space resolution of $32 \times 32 \times 32$ grid points.

In Fig. 11 we show the convergence plot for a specific eigenvalue of the Bragg Stack. A decrease of the largest computed eigenvalue is obtained for an increase in total number of elements. Hence, we have convergence for an increase in total number of elements. The increase in mesh refinement, resulting in up to more than 28000 elements, is handled correctly by the DG-Max solver for the Bragg Stack.

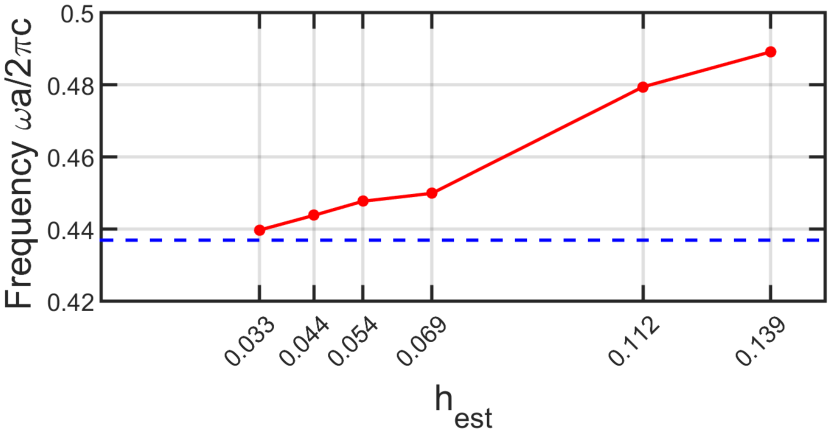


Figure 11: The largest computed eigenvalue by the DG-Max eigenvalue solver of the Bragg Stack for wave vector $k = (0, 0, 0)$. We set the target of the DG-Max solver to $\lambda =$ to obtain eigenvalues in the neighbourhood of $\tilde{\omega} = 0.36$. We set the dielectric permittivities to $\epsilon_1 = 1$ and $\epsilon_2 = 13$. We use different sized unstructured Centaur meshes ranging from 375 elements up to 28351 elements. The estimation of the size of one element h_{est} is expressed as $1/\sqrt[3]{N}$. The red dots indicate the computed eigenvalues. The solid red line indicates the convergence. The blue dashed line indicates the last computed eigenvalue by the MPB solver using $32 \times 32 \times 32$ grid points.

4.4.3 Cylindrical Crystal

We compute the photonic band structure of the Cylindrical Crystal case for a single wave vector $k = (0, 0, 0)$ using the DG-Max eigenvalue solver and the MPB plane-wave expansion method. We take the dielectric permittivity of $\epsilon_2 = 13$ centred in a rod shape in the centre of the cube, with radius $0.2a$, with a the lattice constant. We consider the surroundings of the rod to be air, therefore the dielectric permittivity is given by $\epsilon_1 = 1$. To study the convergence of the results obtained with the DG-Max solver, we generate multiple unstructured meshes of different sizes using Centaur. We use meshes with the total number of elements $N = 1404, 5791, 10446, 14944, 36478$, 80555, and 177587. This results in $h_{est} = 0.089, 0.056, 0.046, 0.041, 0.023$, and 0.018, respectively. Figure 12 shows two cross sections of the unit cell of the Cylindrical Crystal of an unstructured mesh with 80555 elements. To study the convergence of the calculated eigenvalues of the MPB solver, we take grids with $8 \times 8 \times 8$, $16 \times 16 \times 16$ and $32 \times 32 \times 32$ grid points, resulting in $h_{est} = 0.125, 0.0625$, and 0.03125, respectively. We set the target of the DG-Max solver to $\lambda = 12$ to find eigenvalues in the neighbourhood of $\tilde{\omega} = 0.55$.

Similarly to the Bragg Stack, in the Cylindrical Crystal case we do not find one single eigenvalue multiple times for both solvers. We find more diverse non-zero eigenvalues in a range of $\tilde{\omega} = 0.532$ up to $\tilde{\omega} = 0.768$ and several zero eigenvalues. The number of calculated eigenvalues by DG-Max differs per mesh size, which all have a different element distribution. In case of mesh sizes $N = 5791, 10446, 80555$, and 177587 we find 16 non-zero eigenvalues, with mesh size $N = 14944$ we find 17 non-zero eigenvalues, and if $N = 1404$ we find 10 non-zero eigenvalues. Using the MPB solver, we obtain 8 non-zero eigenvalues and several zero eigenvalues. In order to compare the convergence of both solvers, we take the 6^{th} computed eigenvalue for each different mesh size. We find 17 computed non-zero eigenvalues for the mesh size $N = 14944$, which is one more

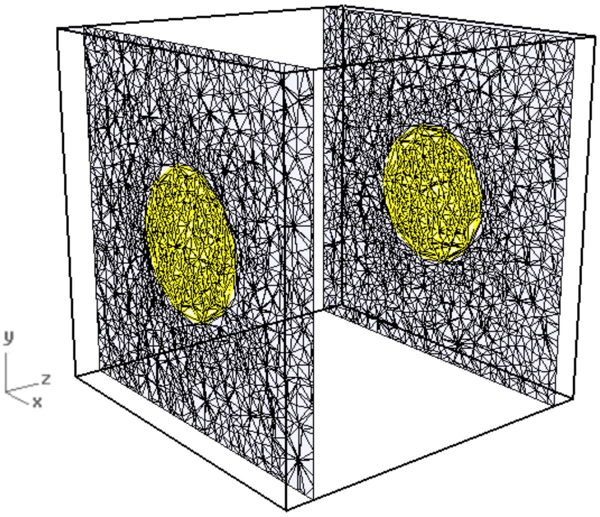


Figure 12: Two cross sections in the xy-plane of the unit cell of the Cylindrical Crystal. The unstructured mesh contains 80555 elements. The mesh is generated using toolkit Centaur [35]. The yellow area indicates the highly-indexed material with dielectric permittivity of $\epsilon_2 = 13$. The grey area indicates the low-indexed material with dielectric permittivity of $\epsilon_1 = 1$.

calculated eigenvalue in comparison to the largest mesh sizes for the DG-Max solver, therefore we take the 7th computed eigenvalue in this case.

Figure 13 shows the convergence for the Cylindrical Crystal of the specific eigenvalue computed using the DG-Max and MPB solver. We plot the obtained eigenvalues $\tilde{\omega}$ against h_{est}. An increase in total number of elements N results in a decrease in the specific $\tilde{\omega}$, for both of the solvers. At smaller mesh sizes the differences in computed eigenvalues by the different solvers is larger in comparison to the computed eigenvalues using larger mesh sizes. We speculate that this is related to the divergence constraint, which is not enforced at the discrete level in DG-Max, whereas it is enforced at the discrete level in the MPB solver [4]. With increasing mesh refinement, the results are approximately equal. The increase in mesh refinement for the meshes created with Centaur, resulting in up to more than 177000 elements, is handled correctly by DG-Max for the Cylindrical Crystal.

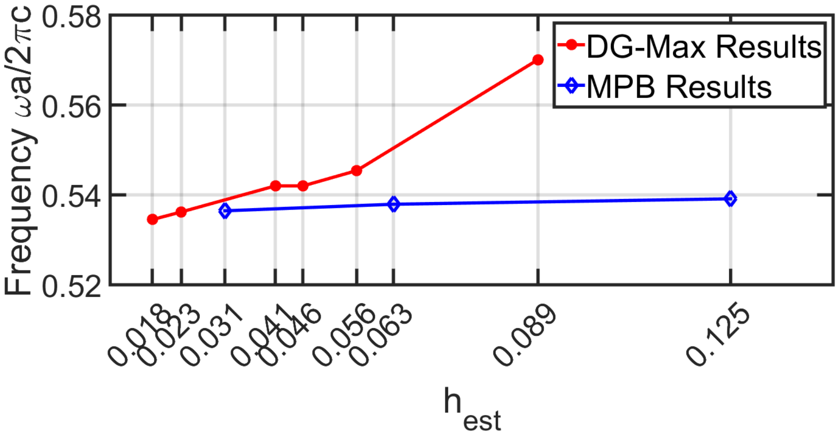


Figure 13: The 6th computed non-zero eigenvalue by the DG-Max eigenvalue solver and the MPB solver of the Cylindrical Crystal for wave vector $k = (0, 0, 0)$. We set the target of the DG-Max solver to $\lambda = 12$ to obtain eigenvalues in the neighbourhood of $\tilde{\omega} = 0.55$. We set the dielectric permittivities to $\epsilon_1 = 1$ and $\epsilon_2 = 13$. We use different sized unstructured Centaur meshes ranging from 1404 elements up to 177587 elements for the DG-Max solver. In case of the MPB solver uses mesh resolutions of 512 up to 32768. The estimation of the size of one element h_{est} is expressed as $1/\sqrt[3]{N}$. The red dots indicate the eigenvalues computed using DG-Max. The open blue diamonds represent the computed eigenvalue using the MPB solver. The solid red and blue line indicate the convergence of both solvers.

4.4.4 Cuboid Crystal

We employ the geometry for the Cuboid Crystal, where we have a square with length and width $0.8a$ centred in a cube with a dielectric permittivity $\epsilon_2 = 13$, which is elongated in the z-direction. We surround the square with air, which has a dielectric permittivity of $\epsilon_1 = 1$, with a width of $0.1a$ in both x- and y-direction. The cube with inner square is periodically repeated in the x- and y-direction using periodic boundary conditions. We compute the photonic band structure for a single wave vector $k = (0, 0, 0)$ for different sized meshes using the DG-Max eigenvalue solver and the MPB Solver. In case of the DG-Max eigenvalue solver, we use $N = 1603, 3278, 5838, 10570, 13810, 23222$, and 35077, which result in $h_{est} = 0.086, 0.067, 0.056, 0.046, 0.41, 0.035$, and 0.031, respectively. Figure 14 shows two cross sections of the unit cell of the Cuboid Crystal of the unstructured mesh with 35077 elements. We employ the MPB solver using resolution spaces 512, 4096, and 32768, resulting in $h_{est} = 0.125, 0.063$. We set the target of the DG-Max solver to $\lambda = 6$ to obtain eigenvalues in the neighbourhood of $\tilde{\omega} = 0.39$.

Both solvers do not find one single eigenvalue with certain multiplicity for one wave vector k, but find more diverse eigenvalues in a range of $\tilde{\omega} = 0.3$ up to $\tilde{\omega} = 0.5$. The number of calculated eigenvalues by DG-Max differs per mesh size, which all have a different element distribution. In case of $N = 5838, 10570, 13810$, we find 15 non-zero eigenvalues, for $N = 35077$ we find 14 non-zero eigenvalues, for $N = 23222$ we find 11 non-zero eigenvalues, with $N = 3278$ 12 non-zero eigenvalues, and for $N = 1603$ we find 17 non-zero eigenvalues. Using the MPB solver, we obtain 13 non-zero eigenvalues. In order to compare the convergence of both solvers, we take the 8th computed eigenvalue for each different mesh sizes. We find less eigenvalues with DG-Max in the cases of $N = 23222$ and $N = 3278$, therefore we take the 5th calculated eigenvalue in these cases.

In Fig. 15 the convergence is shown of the specific eigenvalue computed by the DG-Max eigenvalue solver and MPB solver for increasing total number of elements N. The approximations by the DG-Max solver are worse than the MPB solver, in case of a coarse mesh. However, this is compensated with the faster convergence of the DG-Max solver in comparison with the MPB solver. With increasing mesh refinement we see that the approximations are about equal for both solvers. The increase in total number of elements for Centaur meshes, resulting in a mesh over 35000 elements, is handled correctly by DG-Max.

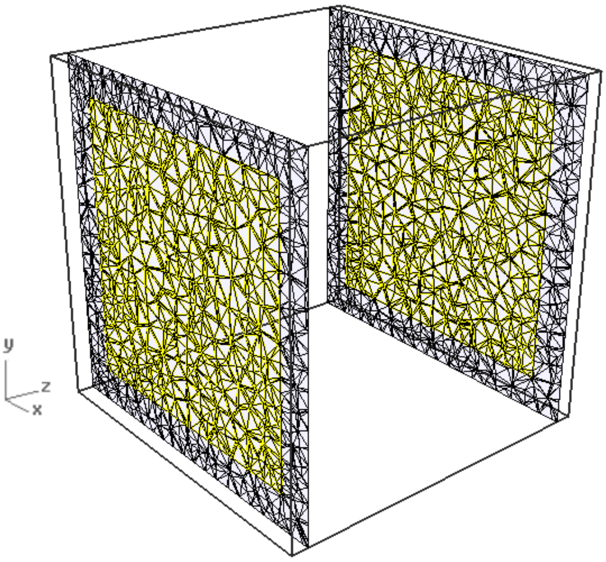


Figure 14: Two cross sections in the xy-plane of the unit cell of the Cuboid Crystal. The unstructured mesh consists of 35077 elements. The mesh is generated using toolkit Centaur [35]. The yellow area indicates the highly-indexed material with dielectric permittivity of $\epsilon_2 = 13$. The grey area indicates the low-indexed material with dielectric permittivity of $\epsilon_1 = 1$.

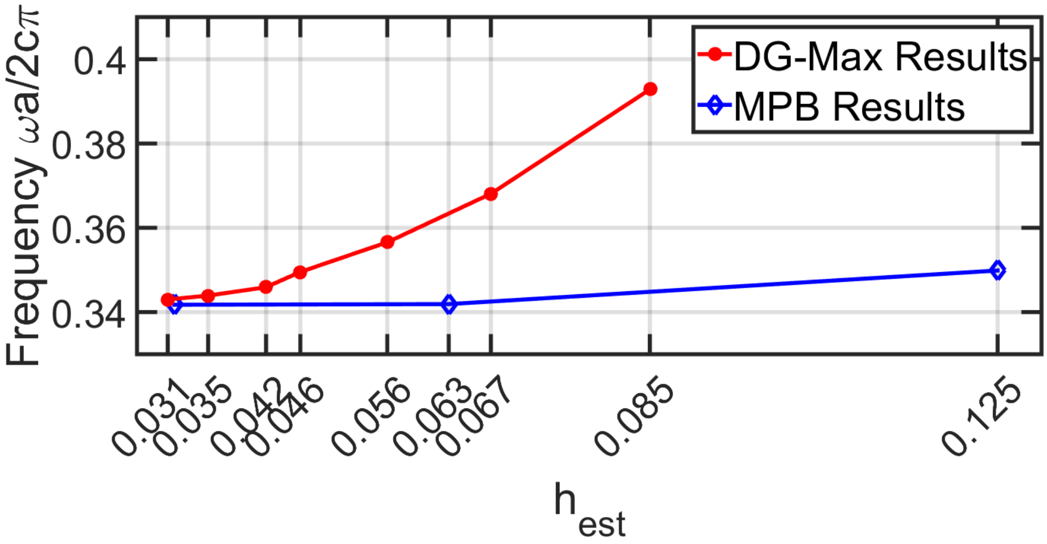


Figure 15: The 8$^{\text{th}}$ computed non-zero eigenvalue by the DG-Max eigenvalue solver and the MPB solver of the Cuboid Crystal for wave vector $k = (0, 0, 0)$. We set the target of the DG-Max solver to $\lambda = 6$ to obtain eigenvalues in the neighbourhood of $\tilde{\omega} = 0.39$. We set the dielectric permittivities to $\epsilon_1 = 1$ and $\epsilon_2 = 13$. We use different sized unstructured Centaur meshes ranging from 1603 elements up to 35077 elements for the DG-Max solver. In case of the MPB solver uses mesh resolutions of 512 up to 32768 gird points. The estimation of the size of one element h_{est} is expressed as $1/\sqrt[3]{N}$. The red dots indicate the eigenvalues computed using DG-Max. The open blue diamonds represent the computed eigenvalue using the MPB solver. The solid red and blue line indicate the convergence of both solvers.

4.4.5 Inverse Woodpile

Next, we test the Inverse Woodpile crystal. We take the two 2D arrays of identical pores running in the ΓX and XS direction. The pores have radius $r = 0.19a$, with a the lattice parameter. We take the unit cell with lattice parameters a in the XS and c in the directions ΓX and SR. To approximate the ideal ratio $\frac{a}{c} = \sqrt{2}$ for cubic symmetry, we use $c = 0.707a$ to create the structure in Rhinoceros for use in DG-Max. In MPB we take the ratio up to 6 decimals of $\sqrt{(}2)$. In Fig. 2 the Inverse Woodpile is depicted with x, y, and z directions corresponding to reciprocal space with $\Gamma X, XS$, and SR directions, respectively. We take the pores to be high-indexed materials with a dielectric permittivity of $\epsilon_2 = 13$. We consider the surroundings to consist of air, which has a dielectric permittivity of $\epsilon_1 = 1$.

The 3D photonic band gap structure is calculated using the DG-Max eigenvalue solver and the MPB solver. For the DG-Max solver, we create meshes using Centaur with total number of elements $N = 11199, 14903, 18568, 28342$, and 56695. Hence, we use $h_{est} = 0.045, 0.041, 0.038, 0.033$, and 0.026, for scaling purposes. Figures 16 and 17 show cross sections of the unit cell of the Inverse Woodpile Crystal of the unstructured mesh with 56695 elements. In case of the MPB solver we use resolutions $35 \times 50 \times 35, 70 \times 100 \times 70$, and $140 \times 200 \times 140$ grid points, resulting in a total number of grid points $N = 61250, 490000$, and 3920000, and we thus have $h_{est} = 0.025, 0.013$, and 0.006, respectively. We set the target of the DG-Max solver to $\lambda = 10$ to obtain eigenvalues in the neighbourhood of $\tilde{\omega} = 0.50$

Both solvers calculate more diverse eigenvalues. The results of the DG-Max solver are in the range of $\tilde{\omega} = 0.33$ up to $\tilde{\omega} = 0.54$, whereas the MPB solver has eigenvalues in a range from $\tilde{\omega} = 0.38$ up to $\tilde{\omega} = 0.75$. The number of computed non-zero eigenvalues by the DG-Max eigenvalue solver are 6, 12, 16, 19, and 17 for the mesh sizes, in order from smallest to largest mesh. The output of the MPB solver consists of 25 non-zero eigenvalues. In order to compare the solvers, we study the influence of the resolution on the calculated eigenvalue around $\tilde{\omega} = 0.495$. Therefore, we take the first calculated eigenvalue by DG-Max, and the 8^{th} calculated eigenvalue by MPB for all different mesh refinements. The results of both solvers are shown in two different figures.

Figure 18 shows the convergence of a specific eigenvalue for the Inverse Woodpile with increasing mesh refinement. The calculated eigenvalue decreases monotonically as the number of elements increases. The increase in total number of elements for Centaur meshes, resulting in a mesh of 56000 elements, is handled correctly by DG-Max, for this complex structure. The number of non-zero eigenvalues even increases with increasing total number of elements. A very small mesh refinement in combination with the complex structure of the Inverse Woodpile crystal results in only six non-zero eigenvalues, whereas in the case of large mesh refinements up to nineteen non-zero eigenvalues are calculated.

In Fig. 19 the convergence of a specific eigenvalue for the Inverse Woodpile with increase in mesh resolution using the MPB solver is shown. We obtain a slightly different eigenvalue around $\tilde{\omega} = 0.495$ in comparison with the DG-Max solver, for reasons that are currently not clear. The solutions of the MPB solver also show a convergence for increasing total number of elements.

In order to obtain the order of convergence and to make use of the Richardson extrapolation we want to use mesh sizes with $N, N/2$, and $N/4$. To approximate the convergence order of the DG-Max solver we employ the Centaur meshes with total number of elements $N = 11199, 28342$, and 56695. In case of the MPB solver we use the given

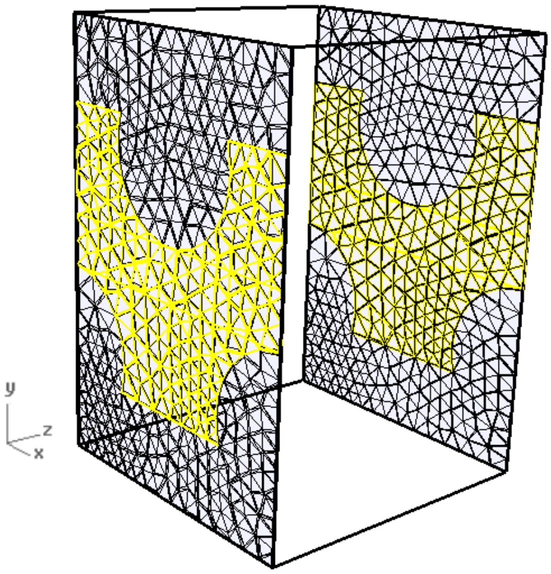


Figure 16: Two cross sections in the xy-plane of the unit cell of the Inverse Woodpile crystal. The total number of unstructured elements is 56695. The mesh is generated using toolkit Centaur [35]. The yellow area indicates the highly-indexed material with dielectric permittivity of $\epsilon_2 = 13$. The grey area indicates the low-indexed material with dielectric permittivity of $\epsilon_1 = 1$.

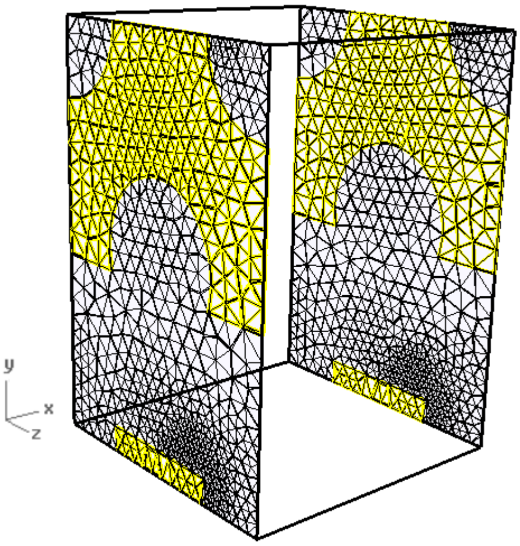


Figure 17: Two cross sections in the yz-plane of the unit cell of the Inverse Woodpile crystal. The total number of unstructured elements is 56695. The mesh is generated using toolkit Centaur [35]. The yellow area indicates the highly-indexed material with dielectric permittivity of $\epsilon_2 = 13$. The grey area indicates the low-indexed material with dielectric permittivity of $\epsilon_1 = 1$.

three meshes with resolutions $35 \times 50 \times 35, 70 \times 100 \times 70$, and $140 \times 200 \times 140$. The convergence order is calculated using

$$\frac{(\tilde{\omega}_{N/4} - \tilde{\omega}_{N/2})}{(\tilde{\omega}_{N/2} - \tilde{\omega}_{N})} = 2^q, \tag{4.6}$$

with $\tilde{\omega}$ the results of the Inverse Woodpile crystal computation with mesh size indicated by the subscript [33]. To calculate the order of convergence directly, we solve

$$q = \frac{\log(\frac{(\tilde{\omega}_{N/4} - \tilde{\omega}_{N/2})}{(\tilde{\omega}_{N/2} - \tilde{\omega}_{N})})}{\log(2)}. \tag{4.7}$$

Figure 20 shows the results of the convergence orders for both the solvers. We only see six band indexes for the DG-Max solver, since the solver only computed six non-zero eigenvalues with the mesh of $N = 11199$. For the MPB solver we have 20 different band indexes and therefore calculate 20 convergence orders. The mean of the convergence orders of both solvers is calculated and plotted. The dashed upper line is the mean of the DG-Max solver and is equal to $q = 2.24$, whereas the lower straight line is the mean of the MPB solver, which is equal to $q = 1.42$. Note that the convergence order of the DG-Max solver is calculated using the finest mesh, which contains about 12000 elements, rather then $N/4 \approx 14000$. We therefore expect that the true convergence order is about $q = 2$ for the DG-Max solver. The convergence order of the MPB solver is approximately $q = 1.5$. The convergence order of DG-Max is higher than the convergence order of the MPB solver.

Using the estimated convergence order of the DG-Max eigenvalue solver we use the Richardson extrapolation to approximate the exact outcome at $h_{est} = 0$ [33]. We approximate the specific eigenvalue using $N = 56695$ and $N/2 = 28342$, using

$$\tilde{\omega} = \tilde{\omega}_N + \frac{(\tilde{\omega}_N - \tilde{\omega}_{N/2})}{(2^q - 1)}, \tag{4.8}$$

where we take an estimated $q = 2$ on basis of the calculated convergence order of the DG-Max solver. This results in an approximated eigenvalue of $\tilde{\omega} = 0.4979$, as shown in Fig. 18.

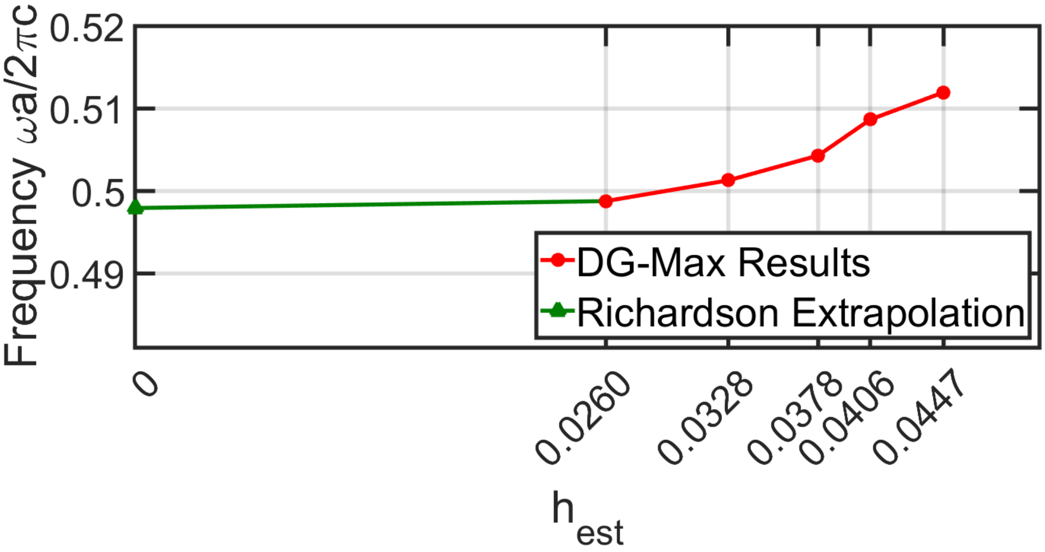


Figure 18: The first computed non-zero eigenvalue by the DG-Max eigenvalue solver of the Cuboid Crystal for wave vector $k = (0, 0, 0)$. We set the target of the DG-Max solver to $\lambda = 10$ to obtain eigenvalues in the neighbourhood of $\tilde{\omega} \approx 0.50$. We set the dielectric permittivities to $\epsilon_1 = 1$ and $\epsilon_2 = 13$. We use different sized unstructured Centaur meshes ranging from 11199 elements up to 56695 elements. The estimation of the size of one element h_{est} is expressed as $1/\sqrt[3]{N}$. The red dots indicate the eigenvalues computed using DG-Max. The solid red line indicates the convergence of the solver. The Richardson Extrapolation is represented by the solid green triangle. The solid green line indicates the convergence according to the Richardson Extrapolation. The extrapolated eigenvalue has a frequency of $\tilde{\omega} = 0.4975$.

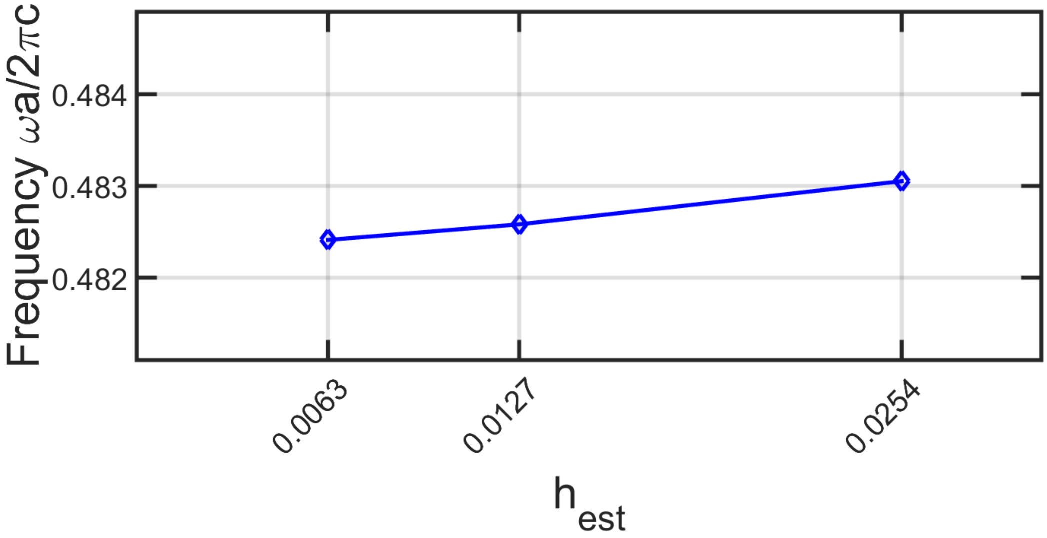


Figure 19: The 8th computed non-zero eigenvalue using the MPB solver of the Cuboid Crystal for wave vector $k = (0, 0, 0)$. We set the dielectric permittivities to $\epsilon_1 = 1$ and $\epsilon_2 = 13$. We use different sized meshes having resolutions of 61250 up to 3920000 grid points. The estimation of the size of one element h_{est} is expressed as $1/\sqrt[3]{N}$. The open blue diamonds indicate the eigenvalues computed by the MPB solver. The solid blue line indicates the convergence of the solver.

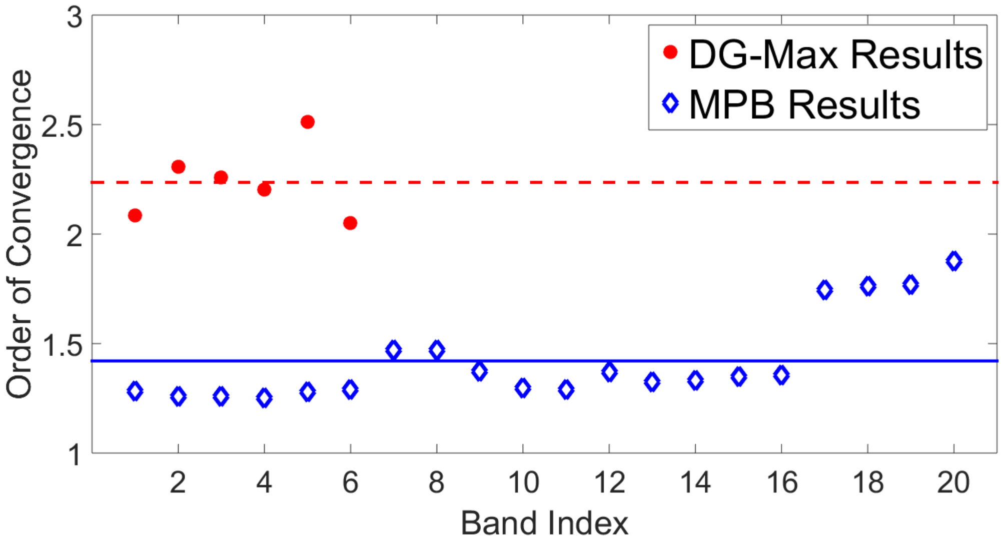


Figure 20: The convergence rates of the Inverse Woodpile crystal for the DG-Max solver and the MPB solver. In case of the DG-Max eigenvalue solver we used mesh sizes of 11199, 28342, and 56695 elements. In case of the MPB solver we used mesh resolutions of 61250, 490000, and 3920000 grid points. The red dots represent the calculated convergence orders using the computed eigenvalues with the DG-Max solver. The upper dashed red line indicates the mean of the order of convergence of the DG-Max results. The open blue diamonds represent the calculated convergence orders of the computed eigenvalues using MPB. The bottom straight blue line indicates the mean of order of convergence of the MPB results.

5 Conclusion and Outlook

In this work, we tested the DG-Max solver and the included algorithms by performing a series of computations on real photonic crystals. The DG-Max solver is validated using multiple test cases for the harmonic and eigenvalue solver. We solved the time-harmonic Maxwell Equations using the Harmonic solver for the Vacuum Crystal. The eigenvalue problem with implicit divergence constraint was validated using the DG-Max eigenvalue solver for the Vacuum Crystal, Bragg Stack, Cylindrical Crystal, and Cuboid Crystal. We solved the eigenvalue problem using the Ainsworth-Coyle basisfunctions with polynomial order $p = 1$. The Vacuum Crystal and Bragg Stack were tested using internally generated structured meshes. We used externally generated unstructured meshes for the Cylindrical and Cuboid Crystal. In order to verify the capabilities of the DG-Max solver, we solved the eigenvalue problem with implicit divergence constraint for a single wave vector $k = (0, 0, 0)$. We were able to perform computations for the Cylindrical Crystal with a mesh consisting of approximately 177000 elements, for the Cuboid Crystal with over 35000 elements, and for the Inverse Woodpile with a mesh of approximately 56000 elements. The convergence rate of the approximations for the eigenvalue solver is very promising. The convergence order of $q = 2$ was achieved for the Inverse Woodpile photonic crystal, which is in accordance with the expected convergence order of $O(h^{2p})$. The Richardson Extrapolation was applied to estimate the eigenvalue of the Inverse Woodpile photonic crystal.

The eigenvalue problem was solved using the implicit divergence constraint, however if the divergence constraint is applied, the null space of the curl-curl operator in the eigenvalue problem is removed. This removal of the null space makes it possible to invert the eigenproblem, which is expected to lead to a much larger convergence speed of the Krylov subspace method. With the divergence constraint, we expect to be able to do calculations with even larger mesh refinements.

Alternatively, applying a preconditioner could lead to faster calculations. We could also study whether using a different iterative solver could enhance the convergence speed.

The DG-Max solver only handles cubic unit cells. Implementing the option of differently shaped unit cells would lead to a wider range of possibilities to study photonic crystals with different crystal latices. The results could also be extended in the case of parallel computations where the wave vector k is not equal to zero.

It is a significant challenge to calculate the optical properties of crystals with disorder, and of crystals with finite size. We anticipate that DG-Max could be easily extended to handle crystals with disorder, by incorporating the disorder into the unstructured mesh. To calculate the optical properties associated with finite size, such as the reflectivity, it is necessary to implement non-reflecting boundary conditions in the harmonic solver.

A user interface of the different DG-Max solvers could contribute to easily switching between test cases. Adjusting settings for the solver in the user interface, instead of making changes to the code, will provide a better usability.

6 Appendix A

To validate the software application DG-Max, we use the Harmonic solver. We solve the time-harmonic Maxwell Equations given in equation (2.85) on the domain $\Omega = [0,1]^3$. We calculate the source term using a given electric field. Using this source term, we recalculate the electric field using the harmonic solver and find the error between the exact electric field and the calculated electric field. We use the Interior Penalty flux with penalty parameters $a_F = n(p+1)(p+3)$ and $b_F = 0$, with n the number of cubes in each coordinate direction and p the polynomial order of the basis functions. Both Ainsworth-Coyle and Nédélec basisfunctions are considered. Note that the total number of elements N in the mesh is equal to $5n^3$. For the validation using the Harmonic solver, we use the simplest test case, i.e., we employ the Vacuum Crystal, and hence the dielectric function is constant throughout the medium. The dielectric constant is set to $\epsilon = 1$. According to the theoretical analysis [22], the order of convergence should be equal to p and $p+1$ for the DG-norm and L^2-norm, respectively. We test the Harmonic solver for two different boundary properties, namely the Homogeneous Boundary Condition and the Periodic Boundary Condition.

6.1 Homogeneous Boundary Condition

We use the boundary condition

$$n \times E = 0$$

with n the outward pointing normal and E the electric field. We use the next exact solution and corresponding source term

$$\begin{aligned} E(r) &= \begin{pmatrix} \sin(\pi y)\sin(\pi z) \\ \sin(\pi x)\sin(\pi z) \\ \sin(\pi x)\sin(\pi y) \end{pmatrix} \\ j(r) &= \begin{pmatrix} \sin(\pi y)\sin(\pi z) \\ \sin(\pi x)\sin(\pi z) \\ \sin(\pi x)\sin(\pi y) \end{pmatrix} (2\pi^2 - 1). \end{aligned}$$

We show the results of the error for the harmonic problem with homogeneous boundary conditions in Table 1. Based on these results we conclude that the implementations for the Ainsworth-Coyle basisfunctions and the Nédélec basisfunctions are correct. The results are in correspondence with References [7, 5, 30, 29].

	Ainsworth-Coyle		**Nedelec**		
	$\|\|E - E_h\|\|_{L^2}$	$\|\|E - E_h\|\|_{DG}$	$\|\|E - E_h\|\|_{L^2}$	$\|\|E - E_h\|\|_{H(\text{curl})}$	$\|\|E - E_h\|\|_{DG}$
p = 1					
N = 5	9.28261E-01	1.62854E-00	6.5833E-01	7.63698E-01	1.27529E-00
N = 40	2.32551E-01	2.019003E-00	4.02097E-01	2.00626E-00	2.02443E-00
N = 320	3.83959E-02	9.55292E-01	2.29614E-01	1.01503E-00	1.01777E-00
N = 2560	8.25675E-03	4.73758E-01	1.18945E-01	5.11162E-01	5.11686E-01
N = 20480	1.99599E-03	2.36097E-01	5.99649E-02	2.5652E-01	2.56639E-01
N = 163840	4.97621E-04	1.17865E-01	3.0041E-02	1.2848E-01	1.28509E-01
p = 2					
N = 5	2.89629E-01	2.61006E-00	3.48656E-01	2.4495E-00	2.52811E-00
N = 40	2.76453E-02	6.3689E-01	1.04756E-01	6.32939E-01	6.39719E-01
N = 320	3.10859E-03	1.53192E-01	2.8631E-02	1.55768E-01	1.56455E-01
N = 2560	3.60829E-04	3.65245E-02	7.2919E-03	3.80809E-02	3.81546E-02
N = 20480	4.39965E-05	8.92042E-03	1.83137E-03	9.3973E-03	9.40574E-03
p = 3					
N = 5	8.04395E-02	1.89736E-00	1.11536E-01	1.66911E-00	1.70135E-00
N = 40	5.00262E-03	1.70872E-01	1.69895E-02	1.55559E-01	1.56825E-01
N = 320	1.98414E-04	1.56358E-02	2.23184E-03	1.55275E-02	1.557E-02
N = 2560	1.05E-05	1.77921E-03	2.82453E-04	1.81169E-03	1.8136E-03
p = 4					
N = 5	2.38314E-02	3.37751E-01	3.37801E-02	3.17005E-01	3.21806E-01
N = 40	4.46275E-04	2.09013E-02	2.13739E-03	2.07168E-02	2.08152E-02
N = 320	9.12398E-05	3.32809E-03	1.51583E-04	2.46929E-03	2.47722E-03

Table 1: Error in the $L^2(\Omega)$, H(curl,Ω), and DG-norm for the harmonic problem with homogeneous boundary conditions for different sized structured meshes and different polynomial orders of the basis functions. We use the Ainsworth-Coyle and Nédélec basis functions.

6.2 Periodic Boundary Condition

In this case we consider periodic boundary conditions. The exact solution and corresponding source term are given by

$$E(r) = \begin{pmatrix} \sin(\pi y)\sin(\pi z) \\ \sin(\pi x)\sin(\pi z) \\ \sin(\pi x)\sin(\pi y) \end{pmatrix}$$

$$j(r) = \begin{pmatrix} \sin(\pi y)\sin(\pi z) \\ \sin(\pi x)\sin(\pi z) \\ \sin(\pi x)\sin(\pi y) \end{pmatrix} (8\pi^2 - 1).$$

The results of the harmonic problem with periodic boundary conditions are given in Table 2. Based on these results we conclude that the implementations for the Ainsworth-Coyle and Nédélec basisfunctions are correct. The results are corresponding with References [5, 7, 29, 30].

	Ainsworth-Coyle		Nedelec		
	$\|\|E - E_h\|\|_{L^2}$	$\|\|E - E_h\|\|_{DG}$	$\|\|E - E_h\|\|_{L^2}$	$\|\|E - E_h\|\|_{H(\text{curl})}$	$\|\|E - E_h\|\|_{DG}$
p = 1					
N = 320	7.75575E-01	3.73966E-00	4.14402E-01	3.84946E-00	3.85994E-00
N = 2560	1.04547E-01	1.881E-00	2.31215E-01	1.99779E-00	1.99991E-00
N = 20480	1.48174E-02	9.42424E-01	1.19048E-01	1.00605E-00	1.00652E-00
N = 163840	2.519E-03	4.71155E-01	5.99695E-02	5.03804E-01	5.03918E-01
p = 2					
N = 320	7.45153E-02	1.03894E-00	1.07071E-01	1.06844E-00	1.0717E-00
N = 2560	5.42294E-03	2.74011E-01	2.86335E-02	2.85039E-01	2.85511E-01
N = 20480	4.54879E-04	6.94452E-02	7.29111E-03	7.25098E-02	7.2571E-02
p = 3					
N = 320	1.25244E-02	2.0532E-01	1.70813E-02	2.09727E-01	2.10098E-01
N = 2560	4.28822E-04	2.64982E-02	2.23521E-03	2.71281E-02	2.71509E-02
p = 4					
N = 320	1.24048E-03	3.22842E-02	2.16354E-03	3.29134E-02	3.29484E-02

Table 2: Error in the $L^2(\Omega)$, H(curl,Ω), and DG-norm for the harmonic problem with periodic boundary conditions for different sized structured meshes and different polynomial orders. We use the Ainsworth-Coyle and Nédélec basis functions.

7 Appendix B

In this appendix we elaborate on the settings used for the toolkit Centaur [35]. We employ different geometries for which we generate different sized meshes. To ensure the reproducability of the results, we give the settings for all these meshes.

After we load the structure in Centaur and set the boundary conditions, we can start generating the mesh. If we click the mesh generation button of Centaur, a window pops up. In this window multiple settings can be manually changed to obtain the preferred mesh refinement. We proceed to customize the input parameters, rather then using the defaults settings. We configure the input parameters which opens a new window containing 6 tabs. We will only change settings in the tabs Surface, Prism/Hex, and Tetrahedra. Once we changed the parameters to the preferred settings, we can proceed with the mesh generation. We do not change advanced settings. The casename is manually changed, according to the test case.

Surface contains 7 parameters of which we will change 3 to the preferred settings. We set the stretching ratio (SR) to control the rate of change of element size from one surface face to its neighbours. We keep the scaling factor, which controls the overall size of the grid elements, equal to default value of 1.0, at all times. We change the maximum length scale $\mathrm{L}_{\mathrm{max}}$ to control the maximum size of surface elements. We change the minimum length scale $\mathrm{L}_{\mathrm{min}}$ to control the minimum size of surface elements. We keep the factor for curvature clustering, proximity clustering, and CAD clustering to the default settings. These factors control the amount of clustering based on the local curvature of the geometry, based on the occasion if two surfaces are close to each other, and based on near small CAD features in the geometry, respectively. We show the changed settings in tables 3, 4, 5, and 6, for different geometries with accordingly mesh sizes.

Prism/Hex contains 5 parameters, however we only change the setting of the number of prism and hex layers we want to use. We set this value to zero in all cases, since we solely want to use tetrahedra.

Tetrahedra shows 5 parameters. We change the stretching ratio (SR) to control the rate of change of element size from one element to its neighbours. We change the interface length scale blending factor (BF) to control which length scale the tetrahedra need to match. We use the interface thickness ratio (ITR) to set the ratio between size of the tetrahedra and the thickness of the last layer of mesh. We use the tetrahedra grid quality (TGQ) to control the tetrahedral quality improvement process. Its value sets the maximum allowable sliver, skewness, and volume ratio. Additionally, the limit maximum tetrahedral size can be set. For the generation of the meshes of the Vacuum Crystal and Bragg Stack, we allowed a maximum scale of 0.125 for all tetrahedra. In all other cases, we did not manually limit the size of all tetrahedra.

	Surface			**Tetrahedra**			
	SR	L_{max}	L_{min}	SR	BF	ITR	TGQ
N = 375	2.0	1.6	$5e^{-2}$	1.5	0.8	2.0	5
N = 707	1.4	1.0	$3e^{-3}$	1.5	0.8	2.0	5
N = 3107	1.35	1.6	$3e^{-3}$	2.0	0.8	2.0	5
N = 6273	1.7	0.02	$1e^{-6}$	1.6	0.8	1.6	5
N = 11663	1.7	0.01	$1e^{-6}$	1.6	0.8	1.6	5
N = 28351	1.8	0.004	$1e^{-4}$	1.6	1.0	1.6	10

Table 3: Input parameters for the customized mesh generation of the Vacuum Crystal and Bragg Stack using Centaur. The abbreviation ST stands for stretching ratio, L_{max} for maximum length scale, L_{min} for minimum length scale, BF for interface length scale blending factor, ITR for the interface thicness ratio, and TGQ for the tetrahedra grid quality.

	Surface			**Tetrahedra**			
	SR	L_{max}	L_{min}	SR	BF	ITR	TGQ
N = 1404	2.2	0.2222	0.005	1.9	0.8	1.9	5
N = 3349	2.0	0.2222	0.020	1.8	0.6	1.8	5
N = 5791	2.0	0.2222	0.005	2.0	0.8	2.0	5
N = 10446	2.0	0.2222	0.005	1.9	0.8	1.9	5
N = 14944	2.0	0.2222	0.005	1.8	0.8	1.8	5
N = 36479	2.0	0.2222	0.005	1.6	0.8	1.6	5
N = 80555	1.5	0.2222	0.005	1.7	0.8	1.7	8
N = 177587	1.5	0.2222	0.0005	1.5	0.8	1.5	8

Table 4: Input parameters for the customized mesh generation of the Cylindrical Crystal using Centaur. The abbreviation ST stands for stretching ratio, L_{max} for maximum length scale, L_{min} for minimum length scale, BF for interface length scale blending factor, ITR for the interface thickness ratio, and TGQ for the tetrahedra grid quality.

	Surface			**Tetrahedra**			
	SR	L_{max}	L_{min}	SR	BF	ITR	TGQ
N = 1603	2.8	0.4	0.03	2.8	0.5	2.8	5
N = 3278	2.5	0.4	0.02	2.6	0.6	2.6	5
N = 5838	2.2	0.4	0.014	2.4	0.8	2.4	5
N = 10570	1.2	0.4	0.005	1.9	0.5	1.9	5
N = 13810	1.6	0.4	0.01	1.8	0.5	1.8	5
N = 23222	1.8	0.4	0.00833	1.7	0.8	1.7	5
N = 35077	1.8	0.4	0.00833	1.6	0.8	1.6	5

Table 5: Input parameters for the customized mesh generation of the Cuboid Crystal using Centaur. The abbreviation ST stands for stretching ratio, L_{max} for maximum length scale, L_{min} for minimum length scale, BF for interface length scale blending factor, ITR for the interface thickness ratio, and TGQ for the tetrahedra grid quality.

	Surface			**Tetrahedra**			
	SR	L_{max}	L_{min}	SR	BF	ITR	TGQ
N = 11199	1.8	0.02828	0.002	2.2	0.7	2.2	5
N = 14903	1.8	0.05656	0.002	2.1	0.6	2.1	5
N = 18568	1.8	0.05656	0.001	2.0	0.8	2.0	5
N = 28341	1.8	0.05656	0.002	1.9	0.7	1.9	8
N = 56695	1.8	0.05656	0.002	1.8	0.7	1.8	8

Table 6: Input parameters for the customized mesh generation of the Inverse Woodpile crystal using Centaur. The abbreviation ST stands for stretching ratio, L_{max} for maximum length scale, L_{min} for minimum length scale, BF for interface length scale blending factor, ITR for the interface thickness ratio, and TGQ for the tetrahedra grid quality.

8 References

[1] M. Merklein, B. Stiller, K. Vu, S. J. Madden, and B. J. Eggleton, “A chip-integrated coherent photonic-phononic memory,” Nature Communications **8**, 574 (2017).

[2] N. Gisin, G. Ribody, W. Tittel, and H. Zbinden, “Quantum cryptography,” Rev. Mod. Phys., **74**, 145 (2002).

[3] T. B. H. Tentrup, W. M. Luiten, P. Hooijschuur, R. van der Meer, and P. W. H. Pinkse, “Two high-dimensional cartesian bases for quantum key distribution,” Information Optics (WIO) **16**, 1 (2017).

[4] J. D. Joannopoulos, S. G. Johnson, J. N. Winn, and R. D. Meade, *“Photonic crystals: Molding the flow of light,”* (Princeton University Press, Princeton NJ, 2008).

[5] M. Metzger, “Accurate Discontinuous Galerkin Finite Element Methods for Computing Light Propagation in Photonic Crystals,” Master Thesis, University of Twente (2015).

[6] S. G. Johnson and J. D. Joannopoulos, “Block-iterative frequncey-domain methods for Maxwell’s equations in a planewave basis,” Opt. Express **8**, 3 (2011).

[7] F. Brink, “Discontinuous Galerkin Finite Element Methods for Photonic Crystals,” Master Thesis, University of Twente (2013).

[8] J.C. Knight, “Photonic crystal fibres,” Nature **424**, 847 (2003).

[9] B. D. Wilts, B. A. Zubiri, M. A. Klatt, B. Butz, M. G. Fischer, S. T. Kelly, E. Spiecker, U. Steiner, and G. E. Schröder-Turk, “Butterfly gyroid nanostructures as a time-frozen glimpse of intracellular membrane development,” Science Advances **3**, e1603119 (2017).

[10] L. Kassa-Baghdouche, T. Boumaza, and M. Bouchemat, “Optical properties of point-defect nanocavity implemented in planar photonic crystal with various low refractive index cladding materials,” M. Appl. Phys. B **121**, 297 (2015).

[11] T. Baba, “Slow light in photonic crystals,” Nature Photonics **2**, 465 (2008).

[12] D. Devashish, “3D Periodic photonic nanostructures with disrupted symmetries,” Phd Thesis, University of Twente (2017).

[13] M. Kerker, *“The Scattering of Light and Other Electromagnetic Radiation,”* (Academic Press, New York and London, 1969).

[14] D. J. Griffiths, *“Introduction to electrodynamics,”* (Prentice Hall, Upper Saddle River NJ, 1999).

[15] M. Sadiku, *“Elements of Electromagnetics,”* (Oxford University Press, Oxford NY, 2007).

[16] T. Chow, *"Electromagnetic Theory: A modern perspective,"* (Jones and Bartlett, 2006).

[17] L. Evans, *"Partial Differential Equations,"* (American Mathematical Society, 1998).

[18] S. Caorsi, P. Fernandes, and M. Raffetto, "On the convergence of Galerkin finite element approximations of electromagnetic eigenproblems," SIAM J. Numer. Anal. **38**, 580 (2000).

[19] L. Demkowicz and L. Vardapetyan, "Modeling of electromagnetic absorption/scattering problems using hp-adaptive finite elements," Comput. Methods Appl. Mech. Eng. **152**, 103 (1998).

[20] L. Vardapetyan and L. Demkowicz, "hp-adaptive finite elements in electromagnetics," Comput. Methods Appl. Mech. Eng. **169**, 331 (1999).

[21] I. Perugia, D. Schötzau, and P. Monk, "Stablized interior penalty methods for the time-harmonic Maxwell equations," Comput. Methods Appl. Mech. Eng. **191**, 4675 (2002).

[22] Z. Lu, A. Cesmelioglu, J. J. W. van der Vegt, and Y. Xu, "Discontinusous Galerkin approximations for computing electromagnetic Bloch modes in photonic crystals," J. Sci. Comput. **44**, 219 (2010).

[23] D. Arnold, "An interior penalty finite element method with discontinuous elements," SIAM J. Numer. Anal. **19**, 742 (1982).

[24] M. Ainsworth and J. Coyle, "Hierarchic finite element bases on unstructured tetrahedral meshes," Int. J. Numer. Meth. Eng. **58**, 2103 (2003).

[25] J. C. Nédélec, "Mixed finite elements in $\mathbb{R}^3$," Numer. Math. **35**, 315 (1980).

[26] P. Houston, I. Perugia, and D. Schötzau, "Mixed discontinuous Galerkin Approximation of the Maxwell Operator," SIAM J. Numer. Anal. **42**, 434 (2004).

[27] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini, "Unified analysis of discontinuous Galerkin methods for elliptic problems," SIAM J. Numer. Anal. **39**, 1749 (2002).

[28] L. Pesch, A. Bell, W. Solie, V. Ambati, B. O., and J. van der Vegt, "hpGEM — A sofware framework for discontinuous Galerkin finite element Methods," ACM Transactions on Mathematical Software **33.4** (2007).

[29] D. Sàrmàny, F. Izsàk, and J. J. W. van der Vegt, "Optimal penalty parameters for symmetric discontinuous Galerkin discretisations of the time-harmonic Maxwell equations," J. Sci. Comput. **44**, 219 (2010).

[30] I. Denissen, "Discontinuous Galerkin Finite Element Methods for the time-harmonic Maxwell equations in periodic media," Master Thesis, University of Twente (2013).

[31] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith, "Efficient Management of Parallelism in Object Oriented Numerical Software Libraries," in "Modern Software Tools in Scientific Computing," edited by E. Arge, A. M. Bruaset, and H. P. Langtangen, p163–202 (Birkhäuser Press, Switzerland, 1997).

[32] V. Hernandez, J.E. Roman, and V. Vidal, "SLEPc: A scalable and flexible toolkit for the solution of eigenvalue problems," ACM Trans. Math. Softw. **31**, 351 (2005).

[33] L.F. Richardson, "The approximate arithmetical solution by finite differences of physical problems including differential equations, with an application to the stresses in a masonry dam," Philosophical Transactions of the Royal Society A. **210**, 307 (1911).

[34] S. R. Huisman, R. V. Nair, L. A. Woldering, M. D. Leistikow, A. P. Mosk, and W. L. Vos, "Signature of a three-dimensional photonic band gap observed on silicon inverse woodpile photonic crystals," Phys. Rev. B **83**, 205313 (2011).

[35] 2017. centaursoft.com.

[36] R. McNeel et al., 2017. rhino3d.com.