Discontinuous Galerkin methods for the simulation of multiscale nanophotonic problems with application to light trapping in solar cells

Alexis Gobé

UNIVERSITÉ CÔTE D'AZUR

ÉCOLE DOCTORALE
SCIENCES FONDAMENTALES ET APPLIQUÉES

Méthodes Galerkin discontinues pour la simulation de problèmes multiéchelles en nanophotonique et applications au piégeage de la lumière dans des cellules solaires

Alexis Gobé

Inria Sophia Antipolis-Méditerranée

Présentée en vue de l'obtention du grade de docteur en sciences,
spécialité mathématiques appliquées
d' Université Côte d'Azur
Dirigée par : Stéphane Lanteri
Soutenue le : 17/02/2020

Devant le jury, composé de :
Christophe Geuzaine, Professeur, Université de Liège
Stéphanie Salmon, Professeur, Université de Reims
Stéphane Collin, Chargé de Recherche, C2N
Amélie Litman, Maître de Conférence, Institut Fresnel
Boniface Nkonga, Professeur, Université de Nice
Claire Scheid, Maître de Conférence, Université de Nice

Titre: Méthodes Galerkin discontinues pour la simulation de problèmes multiéchelles en nanophotonique et applications au piégeage de la lumière dans des cellules solaires

Résumé: L'objectif de cette thèse est l'étude numérique du piégeage de la lumière dans des cellules solaires nanostructurées. Le changement climatique est devenu une problématique majeure nécessitant une transition énergétique à court terme. Dans ce contexte, l'énergie solaire semble être une source énergétique idéale. Cette ressource est à la fois scalable à l'échelle planétaire et écologique. Afin de maximiser sa pénétration, des travaux visant à augmenter la quantité de lumière absorbée et à réduire les coûts liés à la conception des cellules sont nécessaires. Le piégeage de la lumière est une stratégie qui permet d'atteindre ces deux objectifs. Son principe consiste à utiliser des texturations nanométriques afin de focaliser la lumière dans les couches de semi-conducteur absorbantes. Dans ce travail, la méthode de Galerkine Discontinue en Domaine Temporel (DGTD) est introduite. Deux développements méthodologiques majeurs, permettant de mieux prendre en compte les caractéristiques des cellules solaires, sont présentés. Tout d'abord, l'utilisation d'un ordre d'approximation local est proposé , basé sur une stratégie de répartition particulière de l'ordre. Le deuxième développement est l'utilisation de maillage hybride mixant ses élements hexahédriques structurés et tétrahédriques non structurés. Des cas réalistes de cellules solaires issus de la littérature et de collaborations avec des physiciens du domaine du photovoltaïque permettent d'illustrer l' apport de ces développements. Un cas d'optimisation inverse de réseau de diffraction dans une cellule solaire est également présenté en couplant le solveur numérique avec un algorithme d'optimisation bayésienne. De plus, une étude approfondie des performances du solveur a également été réalisée avec des modifications méthodologiques pour contrer les problèmes de répartition de charge. Enfin, une méthode plus prospective, la méthode Multiéchelle Hybride-Mixte (MHM) spécialisée dans la résolution de problème très fortement multiéchelle est introduite. Un schéma en temps multiéchelle est présenté et sa stabilité prouvée.

Mots clés: Équations de Maxwell, Méthode de Galerkine discontinue, Matériaux dispersifs, Photovoltaique, Piégeage de la lumière, Nanophotonique, Maillage hybride, Maillage non conforme, Méthode d'ordre élevé, Méthode multi echelle, Optimisation, Calcul haute performance

Title: Discontinuous Galerkin methods for the simulation of multiscale nanophotonic problems with application to light trapping in solar cells

Abstract: The objective of this thesis is the numerical study of light trapping in nanostructured solar cells. Climate change has become a major issue requiring a short-term energy transition. In this context, solar energy seems to be an ideal energy source. This resource is both globally scalable and environmentally friendly. In order to maximize its penetration, it is needed to increase the amount of light absorbed and reduce the costs associated with cell design. Light trapping is a strategy that achieves both of these objectives. The principle is to use nanometric textures to focus the light in the absorbing semiconductor layers. In this work, the Discontinuous Galerkin Time-Domain (DGTD) method is introduced. Two major methodological developments are presented, allowing to better take into account the characteristics of solar cells. First, the use of a local approximation order is proposed, based on a particular order distribution strategy. The second development is the use of hybrid meshes mixing structured hexahedral and unstructured tetrahedral elements. Realistic cases of solar cells from the literature and collaborations with physicists in the field of photovoltaics illustrate the contribution of these developments. A case of inverse optimization of a diffraction grating in a solar cell is also presented by coupling the numerical solver with a Bayesian optimization algorithm. In addition, an in-depth study of the solver's performance has also been carried out with methodological modifications to counter load balancing problems. Finally, a more prospective method, the Multiscale Hybrid-Mixed method (MHM) specialized in solving very highly multiscale problems is introduced. A multiscale time scheme is presented and its stability is proven.

Keywords: Maxwell's equations, Discontinuous Galerkin method, Dispersive materials, Photovoltaics, Light trapping, Nanophotonics, Hybrid mesh, Non conform mesh, High order method, Multiscale method, optimization, High performance computing

Si une partie de ma thèse devait être écrite en français, c'est bien celle là. Au moment où j'écris ces mots, je suis exactement à un mois du point finale de cette aventure. Quelle aventure, quelle difficile et éprouvante aventure, quelle agréable aventure. Je remercie tout d'abord mon boss et directeur de thèse Stéphane, qui a toujours été là pour moi. On m'a souvent parlé de relations thésard/directeur de thèse difficiles, ou quasiment inexistantes. Pour moi ça a été tout l'inverse, et je ne changerai absolument rien à ces 3.7 années à avoir travaillé avec toi. Claire, merci pour tout. Tu as été indispensable durant ces 3 années. Tu as notamment réussi à me faire comprendre que la recherche, ce n'était pas seulement trouver, et que trouver quelque chose qui ne marche pas trop, c'était avoir trouver quelque chose! Merci à Stéphanie et Christophe pour avoir accepter de rapporter sur ce manuscrit, vos remarques m'ont été particulièrement utile pour sa finalisation. Merci à Amélie, Boniface, Claire et Stéphane d'avoir accepter d'être éxaminateurs. J'ai particulièrement apprécié les discussions scientifiques que nous avons eues avant, pendant et après la soutenance.
Jonathan, merci pour tout. Cette thèse n'aurait clairement pas la même saveur sans toi et ton goût pour l'esthétisme et pour tikz. Épargne moi tes "voleur" par ci, "voleur" par là en voyant mes planches de soutenance...! Niko, danke! Sehr gut café! C'était super d'être doctorant à tes côtés, toujours de supers discussions, que ce soit des discussions scientifiques ou des discussions sur les chars de combat... Docteur Léger, mon voisin mouginois du haut, merci beaucoup pour ce passage de témoin avec la MHM lors de tes derniers mois dans l'équipe. Je te serai éternellement reconnaissant pour ton accueil à Petropolis. C'est sûr que mon séjour au LNCC aurait été très très très différent sans toi. Ludo merci pour ton aide pendant TECSER, c'est grâce à toi que mon stage c'est si bien passé, ce qui m'a permis de commencer cette thèse. Merci Alexandra, surtout pour cette séance de sport ensemble (j'ai encore mal au jambe). Merci George, que ce soit pour le taxi, pour les burgers, pour les pâtisseries libanaises... Le prochain c'est toi! Je vais essayer de remercier également tous les Nachos/Atlantis que j'ai vu passer et avec qui j'ai passé de bons moments (Mahmoud, Nora, Vincent, Mohamed, Théo, Kun, Plum, Omid, Hao, Guillaume, Patrick, Théo, Yves, Stéphane D, Montserrat, Zakaria, Massimiliano, Thibault, Cédric, Leandro, Vincent, ...), mes collègues d'autres équipes (Mickael, Pascal, Radu, Quentin, Milica, Nicholas, Lorenzo, Émilie,...), et mes rencontres d'autre pays (Wesley, Frederic, Tadeu, ...).
Je remercie mes parents, sans qui je n'aurai été capable de rien. Merci d'avoir toujours été là, tout le temps depuis maintenant 28 ans. Je n'aurai jamais la possibilité de vous exprimer toute la reconnaissance que j'ai pour vous. Mémé, merci pour tout. J'ai tellement de chance de vous avoir. Quand j'y pense c'est difficile d'être à 1000 kilomètres de vous. Je vous dédis ces travaux. Merci également à ma famille au sens plus large, oncles, tantes, Audrey, Valérie, Hervé, ... Merci à mes deux grands amis, Clément et Maxence, pour être là depuis bientôt 20 ans. Je vous remercie tout particulièrement pour tous ces soirs où j'étais seul la semaine, et où nos soirées sur le pc à jouer et à parler ont permis à ces 3 années de passer bien plus vite que prévu. Merci également aux potes de la fac (Amélie, Flore, Léa, Pauline, ...) et à tous les autres que j'oublie.
Enfin, j'ai gardé le meilleur pour la fin, enfin la meilleure. Docteur Dupré... enfin pas encore mais c'est une question de jour maintenant! Sans toi, rien de tout ça ne se serait passé. Sans toi, c'est certain que je n'aurais jamais fait de thèse, ni de master de modélisation mathématique, ni de master de mathématiques d'ailleurs. C'est vraiment une chance de t'avoir eu sur ma route ce mois de mars 2012. On dira qu'on a survécu à ces 3 années à devoir attendre le vendredi soir pour se voir. Que c'était difficile. Merci d'avoir fait toutes ces concessions pour moi, pour nous... et surtout d'avoir fait exploser ton indice carbone, c'est plutôt cocasse quand on a fait une thèse sur une energie renouvelable.

Contents

1

Introduction

Mathematical models can describe a wide variety of phenomena, which can be physical, biological, economic, demographic, geological for example. There are a considerable number of models, from different families, sometimes established well before the birth of the first computers. Some are stochastic, *e.g.* with consideration of randomness, while others are deterministic, *e.g.* in the form of an ordinary differential equation (ODE) or a system of partial differential equations (PDE). Those models can be used to simulate realistic phenomena, which can thus play a predictive or explicative role. The study and exploitation of these models have always been essential for engineers. The importance of numerical modeling has become a more significant challenge with the increase in computer system capabilities, and in particular, with the advent of parallel computers. This was particularly the case for electromagnetism.

The physicist James Clerk Maxwell laid the foundations of modern electromagnetism in 1865 in [Max65]. For the first time, he formulated the classical theory of electromagneticm by bringing together electricity, magnetism, and light as various manifestations of the same phenomenon. Maxwell's equations have been studied for many decades, for different purposes. As for most of the computational models, computational electromagnetism can be used for validation and prototyping in a goal-oriented engineering setup. We can find electromagnetism applications everywhere and every day. Many of those are part of our daily life, such as wireless communications, electromagnetic braking, optical fibers, medical imaging, magnetic lift train, induction charging of batteries, or even in our computer's hard drives.

Photonics is one of the fields modeled by Maxwell's equations. It is the physical science of light, which is an electromagnetic radiation. One of the subfields of photonics, which is at the heart of this thesis work, is called nanophotonics.

Nanophotonics and photovoltaics Nanophotonics is the study of light and its interactions with matter at nanometric scales (or nanoscale). Nanophotonics incurs phenomena occurring at distances smaller or equivalent to the wavelength of the light. In the visible range of the electromagnetic spectrum (see figure 1.1), the wavelength is between 400 nm (which is the ultraviolet limit) and 700 nm (which is the infrared limit). When the studied structure is smaller or approaching this particular scale, the light-matter interactions are unveiling new phenomena, and general

optical physics is no longer appropriate. At this range, it is important to take into account the wavelength dependency of the speed of light. This phenomenon is called dispersion, and several models aim to simulate this material specificity, as the Drude model [Dru00], and the Generalized dispersive model [VLDC11, Viq18].

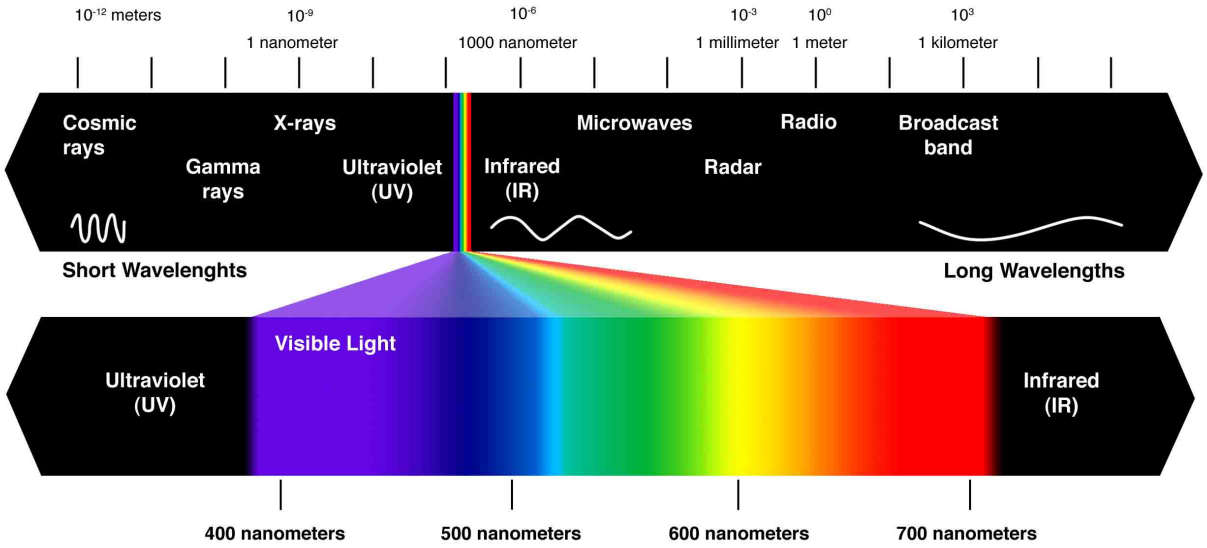


Figure 1.1 | General electromagnetic spectrum and zoom on the nanophotonics range of interest.

Over the past decades, nanophotonics research has led to astonishing technologies. To name some of the most important, the metasurfaces, which are thin sheet metal surfaces that allow the reflected and transmitted waves to be modulated in phase [GCA+17]. Thanks to optothermal heating [SSD+14], nanoparticles are now used in medicine in order to kill cancer cells. Advanced metasurfaces based on nanocube are now being used for nanoscale holography [GC15] and very precise molecule sensing [PCT+16]. Researchers have investigated a variety of nanophotonic methods in order to concentrate light in the optimal areas within a solar cell [FMA10]. Improvements in fabrication processes have permitted to reach high performances up to 44.7% [DGB+14].

In the context of climate change and energy transition, the development of renewable energies is essential. In the world, they represent a little more than 20% of the energy mix. If we put aside hydropower (which is a particular renewable energy because there is no intermittency), the first renewable resource in the world is wind power (9% in 2017) followed by solar energy (4% in 2017). However, according to the International Energy Agency (IEA), renewable energies will grow most rapidly in the electricity sector, providing nearly 30% of electricity demand by 2023, as shown. During this period, renewable energies are expected to account for more than 70% of global electricity generation growth, led by solar photovoltaics, followed by wind, hydro, and bioenergy. In particular, solar photovoltaics is expected to significantly increase all over the world, as shown in figure 1.2. Also, the IEA[1] predicts that photovoltaics could produce up to 16% of the world's electricity by 2050, thus becoming the first renewable resource.

With this in mind, a great amount of research is being carried out to improve efficiency and reduce the fabrication cost of solar cells.

[1]The International Energy Agency

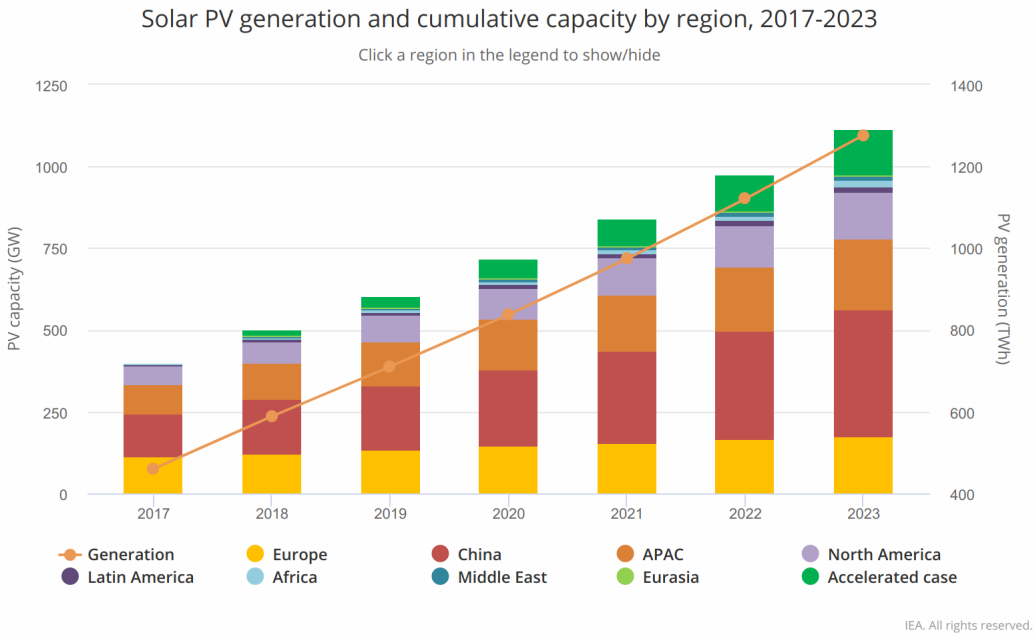


Figure 1.2 | Solar PV generation and cumulative capacity by region, 2017-2023. Source: IEA website: https://www.iea.org/topics/renewables/solar/

"To support global research efforts to produce photovoltaic modules with a >30% energy conversion efficiency for a <30 c$/Wp price by 2030: [30/30/30]" is the roadmap put forth by representatives of the leading solar photovoltaics research institute in France (IPVF [2]). Achieving this objective requires several major innovations. Already promising technologies, such as multijunction thin-films and crystalline silicon solar cells, are good candidates, but for the moment, they suffer from prohibitive manufacturing cost. The development of new experimental concepts based on more complicated physics, such as hot carriers are also considered. Finally, the use of nanophotonics in order to trap light inside the solar cell is an appealing possibility, mostly driven by the decrease in the thickness of solar cells to reduce materials costs [MC12]. Nanoscale light trapping allows the absorption to exceed the limits attainable with geometrical optics.

Light trapping Light trapping consists in capturing photons for use in several applications, such as solar cells. Light enters the different structures by refraction or scattering. Some light trapping structures allow incident light to be guided into the absorbent layer of a solar cell. The purpose is to keep the light trapped inside the device so that the absorption capacity can be improved beyond the absorption of a single pass of light through the device (see figure 1.3).

Different types of nanophotonic structures have been reported in the literature, such as gratings [WYL$^+$12], nanowires [EJD14] and random scattering [Loc12]. These structures are very difficult to choose a priori (before any measures or simulations). Physicists generally need to go through optimization phase, both parametric and geometric, to find the design that will lead to the best efficiency.

[2] Institut Photovoltaïque d'Île-de-France

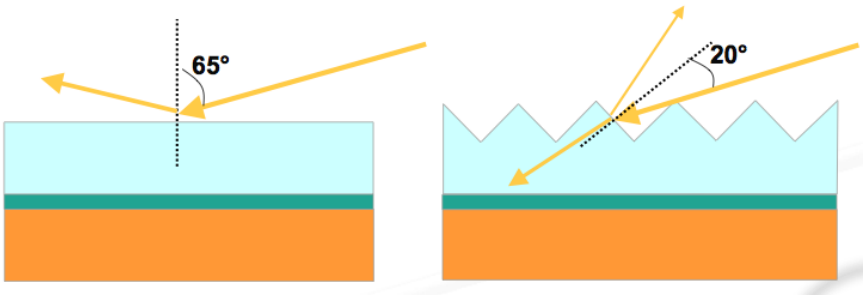


Figure 1.3 | Standard solar glass (left) versus light trapping (right). Source: Saint Gobain

Computational electromagnetism Nowadays, various numerical methods are well known and used to solve Maxwell's PDE system. These methods are especially useful and needed when the application to simulate cannot be handled with analytical solutions. Numerical methods make it possible to solve complex problems, quickly validate a corrective action, solve complex problems by splitting them into subproblems, optimize the sizing of installations, access non-measurable quantities, better understand the physical phenomena involved, but above all, reduce costs by solving problems before they occur. However, most of the realistic electromagnetic problems cannot be solved analytically. The complexity of the underlying PDE systems does not allow for analytical solutions except in a few simple geometrical settings. Therefore, the use of numerical methods is the only way to study complex and real-life problems.

The full-wave Maxwell's equations can be formulated in two settings: the time-domain (TD) and the frequency-domain (FD) formulations. Several numerical methods have been devised to solve both formulations. As often in numerical modeling, the choice of a method is driven by the application.

Frequency-domain methods Frequency-domain methods, also known as time-harmonic methods, are solving Maxwell's equations in frequency-domain. This means that the time variation is supposed to be harmonic. Frequency-domain methods are used when the considered propagation problem is monochromatic. The Finite Element Frequency-Domain (FEFD) is used to solve the frequency-domain boundary valued electromagnetic problems by using a variational form [Can]. It can be used with two- and three-dimensional canonical elements of differing shapes, allowing for a highly accurate discretization of the solution domain. The FEFD is often used in the frequency domain for computing the frequency field distribution in complex, closed regions such as cavities and waveguides. However, the solution domain must be truncated, making the FEFD unsuitable for radiation or scattering problems. The Method Of Moments (MOM) is a technique used to solve electromagnetic boundary or volume integral equations in the frequency-domain [BVF$^+$17]. Because the electromagnetic sources are the quantities of interest, the MOM is very useful in solving radiation and scattering problems. Rigorous Coupled-Wave Analysis (RCWA) method is a semi-analytical method. It is a Fourier-space method, so devices and fields are represented as a sum of spatial harmonics [MG81]. This method is used to improve diffraction structures for high efficiency solar cells. For not so complex 3D structure, RCWA is one of the reference methods. However, when the structure studied is varying in the propagation direction, RCWA needs to discretize it in a set of uniform (in the propagation direction) layer.

Time-domain methods Unlike the frequency-domain methods, time-domain methods with explicit time integration may not require to solve a large linear system of equations. Besides, it is possible to compute the solution over a broadband of wavelengths in a single simulation. For many years, the reference method for solving nanophotonic and PV broadband problems has been the Finite Difference Time-domain (FDTD) method or Yee's method [Yee66]-[TH05]-[GBM15]. The main reasons for the popularity of this method are its ease of implementation and its computational efficiency. However, to achieve the latter properties, the FDTD relies on a cartesian structured discretization of the computational domain [HS81]. When the underlying physics involves local rounded details, the structured grid becomes a significant limitation, which is manageable but makes the method more complex. Being naturally constructed for unstructured grids, the Finite Element method (FEM) is well suited for numerical solution of PDEs in complex geometries [CLM87]. Although initially developed for elliptic equations (boundary value problems), FEM have been further extended to hyperbolic equations (initial boundary value problems). In the time-domain and for the classic FEM, one generally needs to solve the sparse linear system associated to the mass matrix at each time-step, which makes the method very expensive and rarely used in practice. The Finite Volume method (FVM) relies on the evaluation of integrals over volume elements and facets of the mesh. This allows great geometrical flexibility, which is required for most electromagnetic problems [FBBV06]. However, this method lends itself to a piecewise constant approximation of the field and is thus facing accuracy issues. The Boundary Element Method (BEM) is an important alternative to the prevailing domain methods, such as the FDTD or the FETD. The basis of the method is to transform the original system of PDEs into an equivalent integral equation (or system) using the corresponding Green's representation formula (direct method), or in terms of a continuous distribution of singular solutions of the PDE over the boundaries of the problem (indirect method) [HV08]. Finally, about twenty years ago, the Discontinuous Galerkin for time-domain Maxwell's equations (DGTD) was introduced in [RF98].

Discontinuous Galerkin Time-Domain Among the previously presented methods, the Discontinuous Galerkin Time-Domain (DGTD) method has gradually emerged as a interesting alternative to the FDTD method. It is a discontinuous finite element type method that relies on a high-order interpolation of the electromagnetic fields within each cell of an unstructured mesh. Time integration can be achieved using an explicit scheme and, as a result of the discontinuity of the approximation, no global mass matrix inversion is required to advance the solution at each time-step, as it is the case with FETD methods working on unstructured meshes. However, the use of an explicit scheme restricts the time-step used, respectively to the Courant-Friedrichs-Lewy (CFL) condition. The resulting time-domain solver is well adapted to massively parallel computing. DGTD can also easily handle complex meshes thanks to the natural discontinuity of the solution at the cell's interface. The grid may be a conforming unstructured finite element mesh, a non-conforming one, or even a hybrid mesh made of various elements (tetrahedra, hexahedra) without the need of transition elements (prisms, pyramids) in contrast with continuous finite element methods. The DGTD method has been proven to work well with highly locally refined meshes [VL16]. This property makes this method more suitable for designing a *hp*-adaptive solution strategy (*i.e.* where the characteristic mesh size h and the interpolation degree p changes locally wherever it is needed). It is also flexible concerning the time integration scheme. The discontinuous Galerkin spatial discretization can be combined with any explicit or even globally or locally implicit time-stepping scheme.

Multiscale problems Many applications considered in nanophotonics and photovoltaics involve highly heterogeneous media or/and nanotextures embedded in larger domains. Such physical problems are particularly challenging for numerical methods. Indeed, whenever multiple spatial or temporal scales play an essential role in the problem of interest, several numerical difficulties must be addressed. In particular, discretization methods have to use very fine meshes to represent the smallest geometrical details. More specifically, the main difficulty from a numerical modeling point of view is to find parameters for constitutive relationships from material models that are valid at a smaller scale. Indeed, few standard algorithms are efficient, accurate, and robust enough to solve Maxwell's macroscopic equations when calculations involve multiple scales. In recent years, discretization methods suitable for multiscale problems have been studied extensively, such as Discontinuous Galerkin (DG) with strong heterogeneity of the electric permittivity as well as very small and curved geometries [Viq16, CL13]. While these two methods are quite general, homogenization techniques such as Heterogeneous Multiscale (HMM) method [HOV16], or the use of multiscale basis functions as in the Hybrid High-Order (HHO) method [DPT17] help dealing with both scales. The recently introduced MHM method, originally presented in [AHPV13] for elasticity and recently extended for Maxwell equations in time-domain [LPSV18], seems to be a promising approach, both in terms of flexibility and performance thanks to its divide-and-conquer framework totally suitable for massively parallel computer architectures.

In this thesis, we want to show the interest of the DGTD method for the simulation of photovoltaic problems. In particular, the trapping of light in solar cells involves both geometric and physical problems, which allows the use of possibilities of the DGTD method, which is rarely used. The MHM method, still very recent for Maxwell's equation in the time domain, seems very promising for solving photovoltaic problems on a larger scale.

Contributions and thesis outline The manuscript is structured as follow:

- In chapter 2, we introduce the photovoltaic topic, which is nowadays an essential concern in this context of energy transition. We present the main concepts of PV cells, and especially the optical part, which explains how photons are absorbed in the solar cell. As optics is directly related to electromagnetism, we recall the system of Maxwell's equations, as well as its extension to deal with the peculiarities of nanophotonics. Then, we present the spatial discretization of Maxwell's equations using a DG method. We briefly present the time-stepping strategy chosen for the time discretization. Additional details about domain truncation and source definition are also presented.

- In chapter 3, we present extensions to the standard Discontinuous Galerkin Time-domain (DGTD) method that we have studied in the context of the simulation of complex multiscale PV structures. We propose two numerical strategies, which are well-known for being especially well suited to DG methods, in order to deal with the specificities of solar cell structures, in particular, the periodic nanostructured stacking layer. A p-adaptivity strategy is proposed and compared on a realistic PV cell with the one initially present in [VL16]. We show nice improvements, both in terms of accuracy and computational time. We also present the DGTD method on hybrid structured-unstructured meshes, which allows us to improve the computational efficiency of realistic cases further.

- Chapter 7 is dedicated to the MHM for time-domain Maxwell's equations. The MHM is a two-level method, which requires the use of a separate numerical method at the second level. For the latter, we choose to use the DGTD method. This MHM method was initially introduced

in [LPSV18], and is extended here with a multiscale time-stepping scheme in order to get the most out of the MHM. The stability is theoretically proved, and 2D results are presented. For the first time, we show some numerical validations (energy preservation, convergence study), and then we show the capability of the method on a realistic nanophotonic case.

- Chapter 4 is concerned with realistic PV applications. We start with an introduction of state-of-the-art methods to deal with nanophotonic cases. A brief explanation of each method's advantages and drawbacks is proposed, and DG is placed in relation to them. Then, we define the main figures of merit used in photovoltaics and how they are computed in the DGTD framework. Finally, we present how realistic measures on solar cell nanotextures can be handled to generate high fidelity meshes. Two strategies are proposed in order to obtain biperiodic structures of complex shapes.

- In chapter 5, we concentrate on the optimization of PV cells. We investigate the coupling of an optimization method called efficient global optimization (EGO), with our DGTD solver. This optimization method relies on a metamodelisation. The metamodels are used to mimic the figure of merit, at a very cheap cost compared to the full-wave DGTD solver. Two realistic PV optimization problems are studied using the EGO methods. Efficiency and performance are presented, in particular with the parallel EGO (q-EGO), which allows parallel evaluation of the DGTD solver.

- Finally, in chapter 6, we study the high-performance computing aspects of our DGTD solver. We investigate the effect of traditional finite element methods parallelization, which is based on a mesh partition. We show that we have to face load balancing issues, mostly due to local computations inherent to realistic PV simulations. Advanced strategies are presented in order to handle really complex configurations and evaluated on realistic PV problems.

2

Physical and numerical context

Contents

In the context of climate change, the global energy market is turning increasingly into green energies since the end of last century. Renewable energies are scalable to world energy demand. Wind, solar, or hydro energies are the leading candidates to replace fossil energies like nuclear, coal, and gas. Several studies have aimed at estimating the potential of each renewable energy technology, and solar energy has been identified as the favorite [Fin18]. Solar energy is an abundant, clean, and available source of power. In 2017, more than 37% of power generating capacity installed was solar installations (see figure 2.1). This is nearly twice as much capacity as wind power [REN19]. With a potential of 1575 exajoule (EJ) per year, while world energy consumption was 606 EJ in 2015 [Gol01], solar energy could ensure the world's energy demand in the near future.

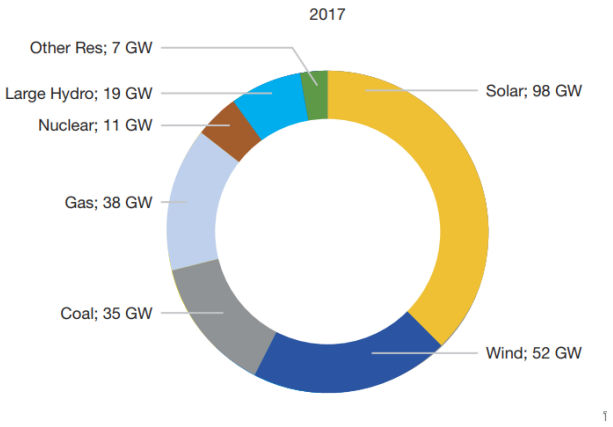


Figure 2.1 | Net power generating capacity added in 2017 by leading technology. Source: Frankfurt School-UNEP Centre and BNEF (2018) at SolarPower Europe 2018.

The physics of solar cells is based on the photovoltaic effect, discovered by Edmond Becquerel [Bec39] in the 19th century. The main idea behind a solar cell is to convert light into electrical power. Historically, space programs were the largest supporters of PV technology since photovoltaics was the best energy source for their satellites. The industry has since grown, and one can have seen PV systems almost everywhere in our daily life. For example, it is used to power cars, houses and commercial buildings.

PV cells are embedded in solar panels, which are also part of the so-called PV system (see figure 2.2). The generation of solar energy requires optimizing each part of a photovoltaic system,

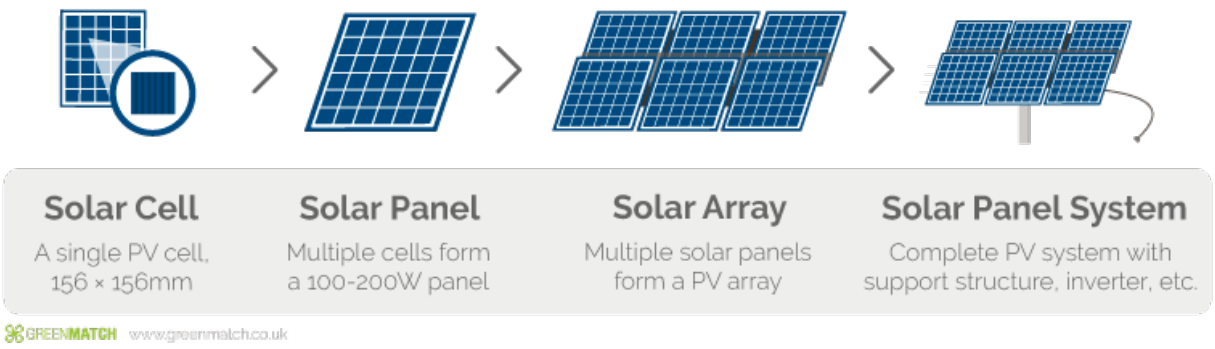


Figure 2.2 | Components of a solar panel system Source: GreenMatch.co.uk

namely collection optics, the photovoltaic array, switches, controllers, current inverters, storage devices, and tracking mechanics.

In order to continue this massive spread of PV around the world, it is necessary to keep improving the efficiency of solar cells while lowering manufacturing and material prices. One particularly well suited approach is to use nanophotonics and light trapping in order to maximize the absorption of the light inside the solar cell. This strategy allows to reduce the thickness of solar cells, which naturally decrease materials costs [MC12]. Most of the simulation of nanophotonics phenomena relies on the resolution of the system of Maxwell's equations.

Apart from basic devices (like rods or spheres) where analytical solutions are available (mostly thanks to Mie theories), realistic configurations are most of the time impossible to solve by hand. For that reason, numerical methods which solve Maxwell's equations have received considerable interest in the last decades. The family of DGTD methods seems promising. DGTD methods rely on a weak formulation and work very well on unstructured meshes. Their main specificity comes from the discontinuous approximation space used that decouples the neighbored mesh elements. Such an approach leads to discontinuities at the cell interfaces and hence to a global discontinuous solution. Moreover, the decoupling of the mesh elements makes the mass matrix block-diagonal, which can therefore be easily invert. This makes DGTD well-suited for explicit time integration schemes.

The first part of this chapter (section 2.1) is an introduction to photovoltaics. Since this thesis is not specifically about the physical aspects of solar light energy harvesting, we intentionally keep the discussion at a general level. This section includes a brief overview of the most used solar cell technologies. Although the numerical modeling of a full solar cell device required to deal with multiple physical models, i.e., it is a coupled multiphysics problem, our main concern in this thesis is the characterization of solar light interaction with a solar cell structures. The optical part of photovoltaics is modeled thanks to the system of Maxwell equations. Theses equations are recalled in section 2.2. Since we adopt a time-domain setting, we discuss in more detail the modeling of material physical dispersion and its role in the present context. The third section of this chapter is dedicated to the numerical modeling of light with dispersive media and detail the high-order DGTD method that constitutes the starting point of this thesis work (section 2.3).

2.1 Photovoltaics

2.1.1 Single junction and multiple junction

Multi-junction solar cells are p-n junction solar cells made of different semiconductor materials. The p- stands for a deficiency of electrons and an abundance of electron holes, while n- stands for the abundance of electrons and few electron holes The p-n junction of each material produces an electrical current in response to different wavelengths of light. Thus, the use of various semiconductor materials allows the absorption of a larger range of wavelengths. The efficiency of converting sunlight from the cell into electrical energy is therefore improved. Traditional single-junction cells have a maximum theoretical efficiency of 33.7%, while an infinite number of junctions would have a limit efficiency of 86.8% (theoretically) under highly concentrated sunlight [Gre01].

2.1.2 Semiconductor materials

Most of the PV cells are made of a stacking of layers of semiconductor materials. Semiconductor materials come from different parts of the periodic table, but they have some similarities. Their properties are related to their atomic characteristics and vary from one group to another. The atoms of a semiconductor can be materials from group IV of the periodic table, or a combination of material from group III and group V (called III-V semiconductors), or also combinations of material from group II and group VI (called II-VI semiconductors). Silicon is a group IV semiconductor material, and it is the most commonly used. One finds it in the basis of integrated circuit (IC) chips, which also make it the most advanced technology. Thus, most solar cells are made of silicon.

To (highly) schematize the PV phenomenon, the incoming light (photons) strikes the solar cell surface, and some of it is absorbed through the semiconductor, turning photons into electrons. The most famous PV cell is the single crystalline silicon, which consists of two differently doped silicon layers (p-n junction) [CFP54] as represented in figure 2.3. A solar cell can be seen as a battery, pushing electrons from the p- to the n- layer.

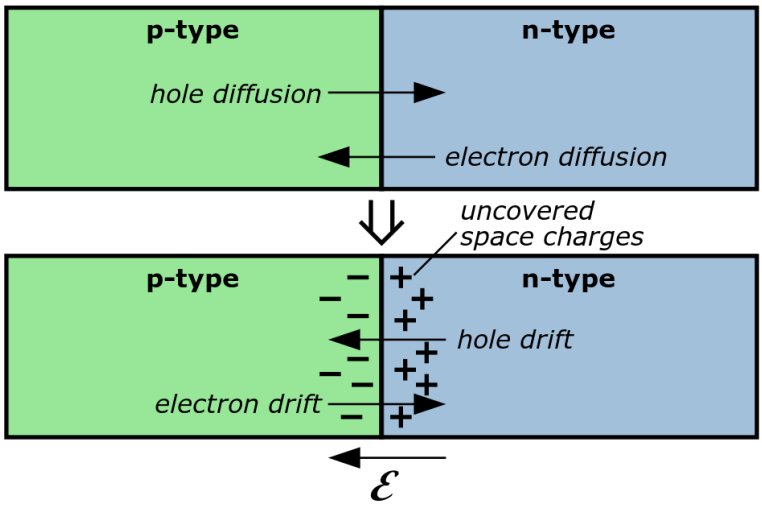
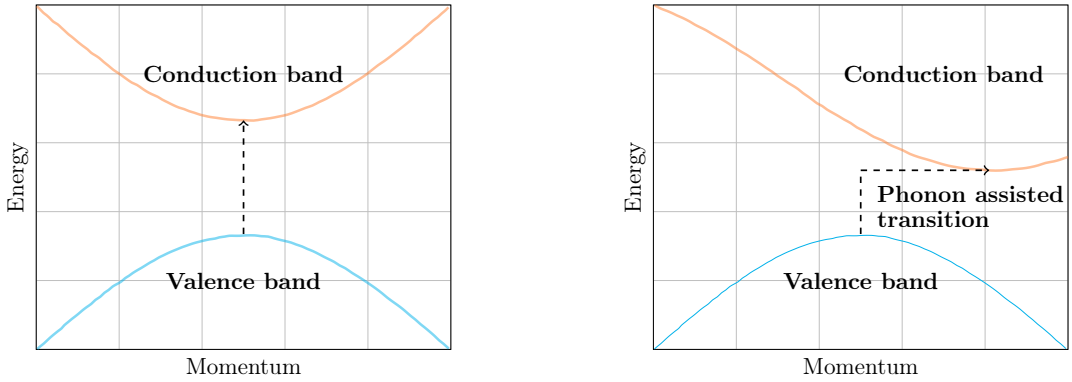


Figure 2.3 | Diagram of the diffusion across a p-n junction, with the resultant uncovered space charges, the electric field and the drift currents. Source: Wikibooks.org

2.1.3 Direct/indirect bandgap materials

The bandgap is the minimum amount of energy required for an electron to release itself from its bound state in order to participate in the conduction. The bandgap determines the amount of solar energy required for conduction, as well as the amount of energy produced. The bandgap of a semiconductor can be either a direct bandgap or an indirect bandgap. The difference between the two is based on the concepts of valence band and conduction band, as shown in figure 2.4. In non-metals, the valence band is the highest range of electron energies in which they are normally present at absolute zero, while the conduction band is the lowest range of free electron states.



(a) Direct bandgap

(b) Indirect bandgap

Figure 2.4 | Energy vs. crystal momentum for a semiconductor with a direct bandgap (a) and an indirect bandgap (b). In the case where the energy of the photon is higher than the bandgap, an electron-hole pair can be generated by exciting an electron of the valence in the conduction band. Any extra energy from the photon above the bandgap is lost due to thermalization.

A direct bandgap semiconductor is a semiconductor for which the maximum energy level of the valence band aligns with the minimum energy level of the conduction band, as shown in figure 2.4(a). Thus, a direct recombination takes place with the release of energy equal to the difference in energy between the recombinant particles. The probability of radiative recombination is high. The efficiency factor of such a semiconductor is higher than for the other bandgap. Thus, these semiconductors are always preferred over indirect bandgap semiconductors for the manufacture of optical sources. Among these semiconductors, one may find amorphous silicon and some III-V materials such as GaAs.

Within an indirect bandgap semiconductor, the maximum energy level of the valence band and the minimum energy level of the conduction band are misaligned, as shown in figure 2.4(b). In this case, due to a relative difference in the voltage pulse, the voltage pulse is first preserved by the release of energy, and only once the two momenta are aligned, a recombination occurs. The latter being accompanied by the release of energy. The probability of radiative recombination is relatively low, and the efficiency factor of such semiconductors is lower. These materials include crystalline silicon and Ge.

2.1.4 Efficiency and absorption

Absorption and efficiency of a PV system are two of the most essential properties of a solar cell. The efficiency of solar cells refers to the portion of sunlight energy which can be transformed by photovoltaics into electricity. This allows to determines the energy production of a PV system.

Absorption of light Three types of phenomena can occur when an incident photon impinges the surface of a semiconductor. The latter can be reflected from the top surface, it can be absorbed by the material, or it can be transmitted by the material. In the case of a solar cell, if the photon is reflected or transmitted, this will be considered a loss. Indeed, if the photon is not absorbed, it cannot produce energy. On the other hand, if it is absorbed, it can then excite an electron from the valence band to the conduction band. The response of the photon (absorption or transmission) is determined by its energy. If the photon does not have enough energy, it will not be able to excite the electron. The photon will therefore be transmitted. In order to be absorbed and excite the electron in the conduction band from the valence band, the incident photon must have sufficient energy. We can then define three groups according to the photon energy E_{ph} and the energy of the semiconductor bandgap E_{bg}:

- When the energy of the photons is lower than that of the material bandgap band $E_{ph} < E_{bg}$, the photons interact weakly with the semiconductor. In this case, they pass through it.
- When the energy of the photons is equal to that of the material bandgap $E_{ph} = E_{bg}$, the photons have just enough energy to create a pair of electron holes. In this case, photons are successfully absorbed.
- When the energy of the photons is higher than that of the material bandgap $E_{ph} > E_{bg}$, the photons are strongly absorbed. However, in the case of solar cells, the excess of energy is wasted due to the thermalization of electrons.

When a photon is absorbed, this results in the creation of a majority and a minority carrier. This production of charge carriers is the basis for PV energy production.

Absorption coefficient The absorption coefficient can quantify the ability of a material to absorb photons. This coefficient determines how far the light can penetrate the material before it is fully absorbed. It is noted α and is defined as follows:

$$\alpha = \frac{4\pi k}{\lambda} \tag{2.1}$$

where k is the extinction coefficient and λ is the wavelength.

When the absorption coefficient is low, this means that the material can weakly absorb the light. This coefficient therefore depends on the material, but also on the wavelength of the incident light. Figure 2.5 displays the absorption coefficient for three semiconductors: Gallium Arsenide (GaAs), Silicon (Si) and Amorphous Silicon (a-Si). The absorption coefficient is shown for the materials at 300° K depending on the wavelength of the light in vacuum. One can see a sharp edge in their absorption coefficient. Indeed, when the energy of the photons is lower than the bandgap as mentioned above, they cannot excite an electron from the valence band into the conduction band. Thus, electrons cannot be absorbed.

When the energy of the photons is close to the energy of the bandgap, only electrons at the edge of the valence band can interact with it, resulting in a low absorption. However, when photons have more energy, they can interact with more electrons, resulting in a larger absorption.

Absorption depth According to equation (2.1), one can see that the absorption coefficient depends on the wavelength. Consequently, light penetrates differently into a semiconductor depending on its wavelength, *i.e.* the light does not travel the same distance in the material before

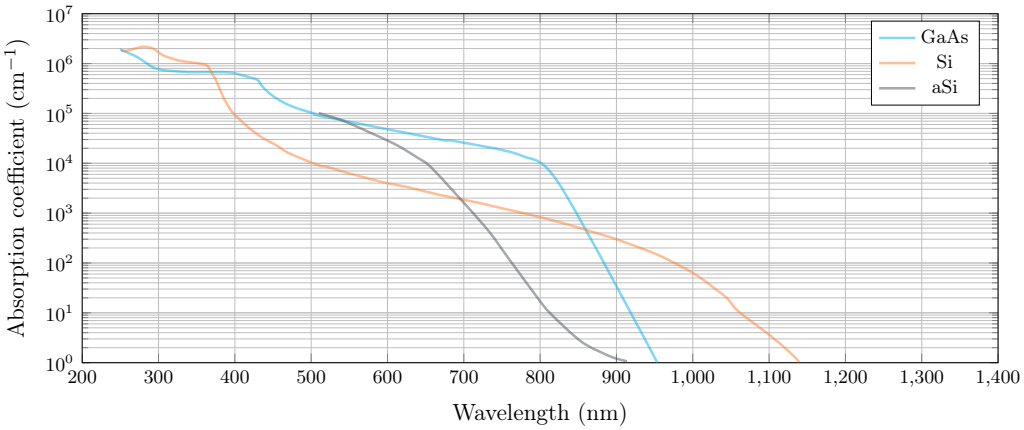


Figure 2.5 | The absorption coefficient, for three different materials (GaAs, Si and aSi) at 300K depending on the wavelength of the light in vacuum.

being absorbed. From the absorption coefficient, it is easy to determine the distance at which light intensity falls below 36% of its original one (or to be reduced by a factor $1/e$, with e the elementary charge). This distance is the absorption depth and corresponds to α^{-1}. It reveals that short wavelength light (high energy light) is absorbed over a short distance because of its higher absorption coefficient.

2.1.5 Shockley-Queisser limit

The Shockley-Queisser limit sets an upper bound for single-pass solar cell efficiency. It was calculated by William Shockley and Hans-Joachim Queisser in 1961 [SQE61]. This limit predicts the maximum photovoltaic conversion efficiency using a single bandgap semiconductor, which is around 33.7% for a single p-n junction cell under typical sunlight conditions. It is calculated by considering the amount of electrical energy extracted per incident photon. The general approach to compute the Shockley-Queisser limit is to calculate the absorption flux and the flux emitted from the solar cell. The difference between these two is the current from the solar cell.

2.1.6 Overview of solar cell technology

One way to maximize absorption of the whole solar spectrum is to stack layers of different materials, each absorbing its specific part of the spectrum. In this context, new types of solar cell designs have emerged.

Crystalline silicon ($c-Si$) is the most matured semiconductor technology used for the production of solar cells. It relies on a thick wafer of a few hundred micrometers in order to absorb sunlight. The state-of-art $c-Si$ solar cell efficiency is 26.7% [YYUA18] Thin-film technologies only use few micrometers layers to absorb sunlight thanks to direct bandgap materials. III-V semiconductors are one of the best technologies to achieve high-efficiency thin-film solar cells. Thanks to the Shockley-Queisser limit, GaAs (Gallium arsenide) is the most efficient material for single-junction solar cells, with an efficiency of up to 28.8%. One way to maximize absorption of the whole solar spectrum is to stack different materials, each absorbing its part of the spectrum. The bottleneck for III-V cells is their high cost due to slow crystal growth and expensive III-V crystalline substrates. These materials are mainly used for spatial engineering.

Crystalline silicon modules have reduced the market price below 1 USD per Watt since 2013. This is due to improved efficiency and thinner c-Si wafers (nowadays about $160 \approx 190\,\mu m$, in contrast to $300\,\mu m$ in the early 2000s). In 2017, the worldwide PV module production was estimated at around 97.5 GWp. 95% of the total production is from wafer-based silicon technology (62% multi-Si and 33% mono-Si) and 5% of the total production is from thin-film technologies (CdTe, CIGS, and a-Si). III-V multijunction and concentrating photovoltaic (CPV) systems still account for very few percentage in the terrestrial photovoltaic application. Figure 2.6 displays the efficiency of research solar cells classified by materials and types from 1975 to nowadays.

The majority of solar cells suffer from losses, meaning that the absorption of sunlight is incomplete over the whole spectrum. This can be caused by the physical and electrical properties of the PV material. For example, Amorphous Silicon (a-Si) is a bad absorber in the red part of the spectrum ($\Lambda \approx 700\,nm$). This loss mainly comes from thermalization loss. Another common reason is the high reflection of the front surface of the solar cell.

2.1.7 Efficiency/cost trade-off

In order to increase the efficiency of PV cells, different strategies are possible. For example, one can increase the amount of absorber inside the cell by thickening the absorbing layers. This possibility is not privileged nowadays because it can considerably increase the cost of the cell. A more elegant way is to use nano-texturing and light trapping techniques, so that the light can be focused in the absorber layers as in [DTM+17].

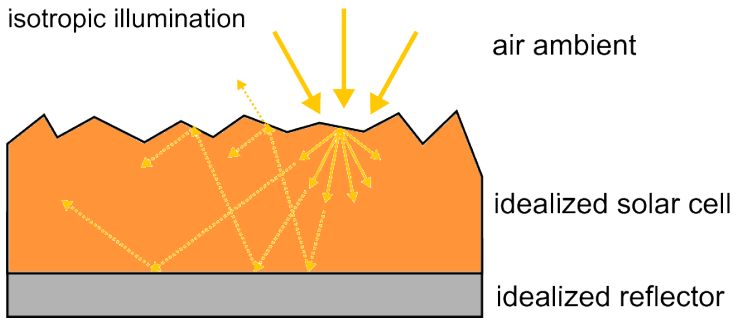


Figure 2.7 | Illustration of light trapping.

Light trapping structures allow incident light to be guided and focused in the absorber layer of a solar cell, as illustrated in figure 2.7. The absorption can be enhanced beyond the absorption of a single light pass through the device by keeping the light stuck inside the absorber. Such light trapping techniques have been possible thanks to the research on nano-optics and plasmonics during the last decades. Using metal nanostructures in dielectric layer stacks [MLD+08] or nano-gratings [DI16] can help in the building process of new solar cell designs. Different type of nano-texturing such as growth-induced textures [TLF+17], wet-chemical etching [PLD05], photonic crystals [ZYH+06] or plasmonic scattering at the front or back contact [CP08] and nano-imprint lithography [STZS+03] are nowadays available.

For such schemes, the standard theory developed by Eli Yablonovitch [Yab82], shows that the absorption enhancement factor has an upper limit of $4n^2$ (knows as the Yablonovitch limit) where n is the refractive index of the absorber. For nanophotonic films with thickness comparable or

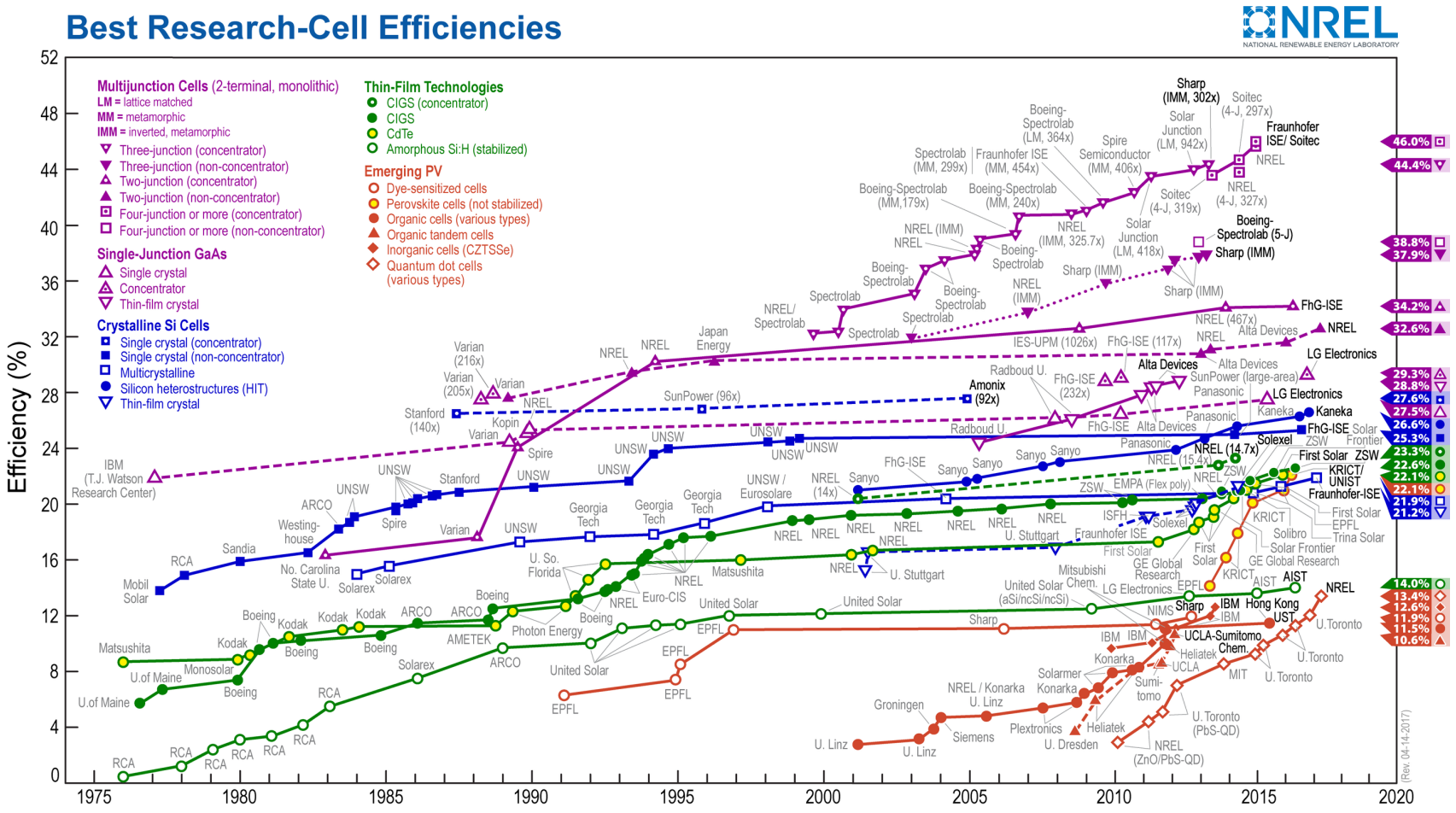


Figure 2.6 | Efficiency of research solar cells classified by materials and types from 1975 to nowadays. Source: National Renewable Energy Laboratory (2018)

even smaller than wavelength scale, ray optics picture, and some of the underlying assumptions in the conventional theory are no longer applicable. In that case, it can be shown that the absorption enhancement factor can go far beyond the Yablonovitch limit with proper design [MC12, BHS+12]. In order to model the absorption of light in solar cells, one needs to solve the system of Maxwell's equations (section 2.2).

2.2 Maxwell's equations

The complete set of Maxwell's equations in their modern form, that describe the spatio-temporal evolution of electromagnetic waves on a domain $\Omega \subset \mathbb{R}^3$, over a given time interval $[0, T]$,$T > 0$, are defined for all $(\mathbf{x}, t) \in \Omega \times [0, T]$ by

Maxwell's equations

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}, \tag{2.2}$$

$$\nabla \times \mathbf{H} = \frac{\partial \mathbf{D}}{\partial t} + \mathbf{J}, \tag{2.3}$$

$$\nabla \cdot \mathbf{D} = \rho, \tag{2.4}$$

$$\nabla \cdot \mathbf{B} = 0. \tag{2.5}$$

with $\mathbf{E}$ and $\mathbf{B}$ the electric field and the magnetic induction, which are vectors in $\mathbb{R}^3$. Most of the physics textbooks considered $\mathbf{E}$ and $\mathbf{B}$ as "fundamental fields". We also introduce the electric displacement $\mathbf{D}$, the magnetic field $\mathbf{H}$, as well as the density of free electric charges ρ, and the free electric current density $\mathbf{J}$. These quantities are all dependent on position $\mathbf{x} = {}^t(x, y, z)$ and time t.

2.2.1 Constitutive relations

To close the system (2.2 – 2.3), relations between $(\mathbf{E}, \mathbf{B})$ and $(\mathbf{D}, \mathbf{H})$ are required. In the general case, the constitutive relations are:

$$\mathbf{D} = \bar{\bar{\varepsilon}}\mathbf{E}, \qquad \mathbf{B} = \bar{\bar{\mu}}\mathbf{H}, \tag{2.6}$$

where $\bar{\bar{\varepsilon}}$ and $\bar{\bar{\mu}}$ are tensors depending on $\mathbf{x}$, t, $\mathbf{E}$ and $\mathbf{B}$. One can make some assumptions on the considered material to temporarily simplify this system. If the material is linear, $\bar{\bar{\varepsilon}}$ and $\bar{\bar{\mu}}$ are independent of $\mathbf{E}$ and $\mathbf{B}$; if it is isotropic, $\bar{\bar{\varepsilon}} \equiv \varepsilon \mathbb{I}_3$ and $\bar{\bar{\mu}} \equiv \mu \mathbb{I}_3$; if it is homogeneous ε and μ are constant. Assuming all those assumptions, (2.6) becomes:

$$\mathbf{D} = \varepsilon \mathbf{E}, \qquad \mathbf{B} = \mu \mathbf{H}.$$

It is standard to use relative permittivity ε_r and relative permeability μ_r, which are the normalization of ε and μ by ε_0 and μ_0, respectively the vacuum permittivity and permeability. Of course, in vacuum we have $\varepsilon_r = 1$ and $\mu_r = 1$. Hence, the constitutive relations are written as follows:

$$\mathbf{D} = \varepsilon_0 \varepsilon_r \mathbf{E}, \qquad \mathbf{B} = \mu_0 \mu_r \mathbf{H}. \tag{2.7}$$

and Maxwell's equations for linear, homogeneous, isotropic, non-dispersive materials:

$$\begin{aligned} \nabla \times \mathbf{E} &= -\mu_0 \mu_r \frac{\partial \mathbf{H}}{\partial t}, \\ \nabla \times \mathbf{H} &= \quad \varepsilon_0 \varepsilon_r \frac{\partial \mathbf{E}}{\partial t} + \mathbf{J}. \end{aligned} \tag{2.8}$$

Before being able to solve it, we need to add boundary and initial conditions to system (2.8). For numerical processing, we are adimensionning the system (2.8). The adimenssionated variables of $(\mathbf{E}, \mathbf{H}, \mathbf{J})$ are noted $(\widetilde{\mathbf{E}}, \widetilde{\mathbf{H}}, \widetilde{\mathbf{J}})$, and are computed as:

$$\begin{aligned} \widetilde{\mathbf{H}} &= Z_0 \mathbf{H}, \\ \widetilde{\mathbf{E}} &= \mathbf{E}, \\ \widetilde{t} &= c_0 t, \\ \widetilde{\mathbf{J}} &= Z_0 \mathbf{J}, \end{aligned}$$

with $Z_0 = \sqrt{\frac{\mu_0}{\varepsilon_0}}$ the vacuum impedance and $c_0 = \frac{1}{\sqrt{\varepsilon_0 \mu_0}}$ the speed of light in vacuum. Hence, the normalized Maxwell's system is:

$$\begin{aligned} \frac{\mu_0 c_0}{Z_0} \frac{\partial \widetilde{\mathbf{H}}}{\partial \widetilde{t}} &= -\frac{1}{\mu_r} \nabla \times \widetilde{\mathbf{E}}, \\ \varepsilon_0 c_0 Z_0 \frac{\partial \widetilde{\mathbf{E}}}{\partial \widetilde{t}} &= \quad \frac{1}{\varepsilon_r} \left(\nabla \times \widetilde{\mathbf{H}} - \widetilde{\mathbf{J}} \right), \end{aligned}$$

Given the definitions of c_0 and Z_0, one sees that $\frac{\mu_0 c_0}{Z_0} = \varepsilon_0 c_0 Z_0 = 1$. We can now drop the tilde notation to obtain the normalized Maxwell system, which will be used from now on.

Maxwell normalized system

$$\frac{\partial \mathbf{H}}{\partial t} = -\frac{1}{\mu_r} \nabla \times \mathbf{E}, \tag{2.9}$$

$$\frac{\partial \mathbf{E}}{\partial t} = \quad \frac{1}{\varepsilon_r} \left(\nabla \times \mathbf{H} - \mathbf{J} \right). \tag{2.10}$$

2.3 Dispersion models

Dispersion is a common phenomenon for all types of waves that travel through a medium. Wave's propagation is affected depending on the medium. For instance, for a polychromatic wave, all the wavelengths do not travel at the same speed through the medium. Among the numerous phenomena encountered in electromagnetics, many rely on the dispersive properties of materials. Dispersion is modeled by a frequency-dependent (wavelength-dependent) permittivity function $\varepsilon(\omega)$ ($\varepsilon(\lambda)$), often obtained from physical considerations. Regarding nanophotonics and photovoltaics applications, an accurate modeling of the ε for semiconductors in the visible spectrum is crucial. Several models aim to simulate this material specificity, as the Drude model [Dru00] and the Generalized dispersive model [VLDC11, Viq18].

2.3.1 Drude and Drude-Lorentz model

The Drude model is based on the kinetic theory of gases [Dru00]. This approximation consists of considering the metal as a static lattice of ions immersed in a free electrons gas. The interactions of these electrons with the ions are managed with collision frequency parameter γ_d. With regard to electron-electron interactions, they are totally neglected. For the electron gas, this leads to the following classical equation of motion:

$$\frac{\partial^2 \mathbf{x}}{\partial t^2} + \gamma_d \frac{\partial \mathbf{x}}{\partial t} = -\frac{e}{m_e}\mathbf{E}(t),$$

with m_e the electron mass and e the electronic charge. Considering a harmonic time-dependence of the form $\mathrm{e}^{-i\omega t}$, one obtains:

$$\widehat{\mathbf{x}} = \frac{e}{m_e}\frac{1}{\omega^2 + i\omega\gamma_d}\widehat{\mathbf{E}}.$$

By using the definition of the polarization $\widehat{\mathbf{P}} = -n_e e\widehat{\mathbf{x}}$, with n_e the electronic density, the latter equality can be rewritten as:

$$\widehat{\mathbf{P}} = -\varepsilon_0 \frac{n_e e^2}{m_e \varepsilon_0}\frac{1}{\omega^2 + i\omega\gamma_d}\widehat{\mathbf{E}},$$

and thus, the electric displacement becomes:

$$\widehat{\mathbf{D}} = \varepsilon_0 \left(1 - \frac{\omega_d^2}{\omega^2 + i\omega\gamma_d}\right)\widehat{\mathbf{E}},$$

where $\omega_d = \sqrt{\frac{n_e e^2}{m_e \varepsilon_0}}$ is called the plasma frequency of the electrons. It is common to include an additional parameter, χ_b, describing the contribution of the bound electrons at infinite frequency [Mai07]:

$$\widehat{\mathbf{D}} = \varepsilon_0 \left(1 + \chi_b - \frac{\omega_d^2}{\omega^2 + i\omega\gamma_d}\right)\widehat{\mathbf{E}}.$$

Then, the definition of the relative permittivity function is directly obtained by matching the previous expression with $\widehat{\mathbf{D}} = \varepsilon_0\varepsilon_r(\omega)\widehat{\mathbf{E}}$:

$$\varepsilon_{r,d}(\omega) = \varepsilon_\infty - \frac{\omega_d^2}{\omega^2 + i\omega\gamma_d}, \tag{2.11}$$

where $\varepsilon_\infty = 1 + \chi_b$ is the permittivity at infinite frequency. Hence, in the same classical approach as before, a spring term is added to the equation of motion:

$$\frac{\partial^2 \mathbf{x}}{\partial t^2} + \gamma_l \frac{\partial \mathbf{x}}{\partial t} + \omega_l^2 \mathbf{x} = -\frac{e}{m_e}\mathbf{E}(t).$$

Following the same development as for the Drude model, one obtains the expression of a Lorentz pole:

$$\varepsilon_{r,l}(\omega) = -\frac{\Delta\varepsilon\omega_l^2}{\omega^2 - \omega_l^2 + i\omega\gamma_l}.$$

The Drude-Lorentz model is then obtained by summing Drude and Lorentz terms:

$$\varepsilon_{r,dl}(\omega) = \varepsilon_\infty - \frac{\omega_d^2}{\omega^2 + i\omega\gamma_d} - \frac{\Delta\varepsilon\omega_l^2}{\omega^2 - \omega_l^2 + i\omega\gamma_l}. \tag{2.12}$$

with $\Delta\varepsilon$ a parameter representing the amplitude of the associated Lorentz pole.

2.3.2 Generalized dispersive model

Drude and Drude Lorentz models aims at modeling metals. For more specific materials, one usually cannot fit the dispersion profiles with the classical models. A more general approach is needed, allowing the fitting of any sort of optical measurements. A Padé style approximation is a simple analytical method for fitting an empirical set of points representing a material's permittivity function. The fundamental theorem of algebra allows to expand this approximation as a sum of a constant, one zero-order pole (ZOP), a set of first-order generalized poles (FOGP), and a set of second-order generalized poles (SOGP), as:

Generalized dispersive model

$$\varepsilon_{r,g}(\omega) = \varepsilon_\infty - \frac{\sigma}{i\omega} - \sum_{l\in L_1} \frac{a_l}{i\omega - b_l} - \sum_{l\in L_2} \frac{c_l - i\omega d_l}{\omega^2 - e_l + i\omega f_l}, \tag{2.13}$$

where $\varepsilon_\infty, \sigma, (a_l)_{l\in L_1}, (b_l)_{l\in L_1}, (c_l)_{l\in L_2}, (d_l)_{l\in L_2}, (e_l)_{l\in L_2}, (f_l)_{l\in L_2}$ are in $\mathbb{R}$, and L_1, L_2 are non-overlapping sets of indices. The constant ε_∞ represents the permittivity at infinite frequency, and σ the conductivity. The ZOP is represented by one parameter, each FOGP are represented by two parameters and each SOGP are represented by four parameters.

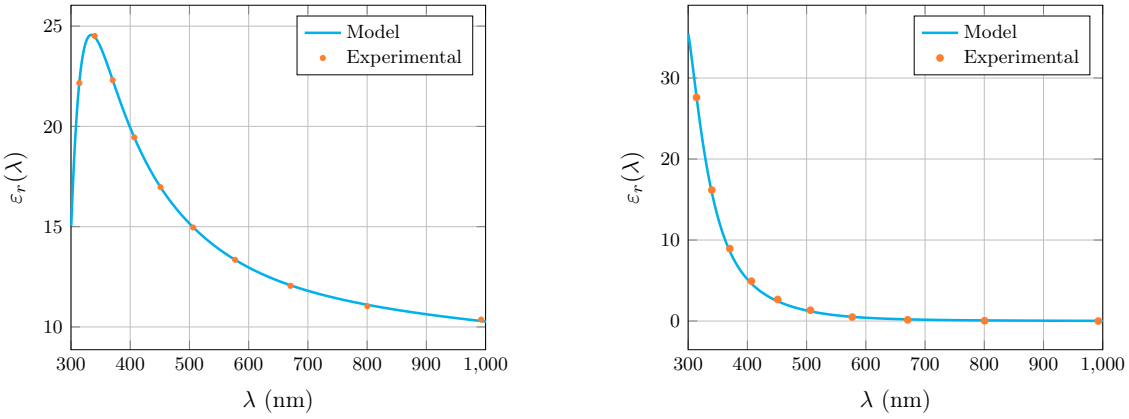
Most of the standard dispersion models are naturally includes in this single formulation (Debye, Drude and Drude-Lorentz, Sellmeier's law). This model has been studied in [Viq16], and was also used by Vial *et al* in [VLDC11] and by Han *et al* for the Complex-Conjugate Pole-Residue Pairs model (CCPRP) (see [HDF06]).

The strategy to fit the coefficients of (2.13) to experimental data is taken from [Viq18]. This fitting is a not trivial optimization problem. Especially for an increasing number of poles ($L1$ and $L2$), one can be left with a significant maximization problem, which may have many local maxima. Simulated Annealing (SA) methods have proved to be particularly well suited and efficient for finding global maxima in these situations ([KGV83]). In practice, a set of experimental data (measured by physicists or taken from well-known database such as [1]) is provided to the SA algorithm, as well as a number of FOGP and SOGP. The spectrum of interest is a crucial factor in the quality of the fitting, and it should be chosen in accordance with the part of the light considered. Indeed, the number of poles needs to depend on the behavior of the experimental permittivity function in the selected frequency range. In this manuscript, the spectrum of interest is in general set to $[300, 1000]$ nm, which constitutes a good representative range for PV applications. We show the fitting obtained for microcrystalline silicium (μc-Si) using 2 FOGP and 5 SOGP in figure 2.8. The optical measures are perfectly fitted over the whole range, with a relative error less than 1%. In chapter 4, a fit issue concerning bandgap modeling which is impacting the physical quantities specific to photovoltaics will be discussed.

2.3.3 Maxwell-GDM equations

In this section, the modified Maxwell's equations for dispersive media are derived by considering the previously introduced generalized dispersive model. One can derive the system of PDEs, following the steps described in [Viq16] for the Maxwell-Drude equations. The obtained system of PDEs, accounting for the generalized dispersive model (GDM) in time-domain is given by:

[1] https://refractiveindex.info/



(a) Real part

(b) Imaginary part

Figure 2.8 | Real and imaginary parts of the microcrystalline silicon relative permittivity predicted by the generalized dispersive model compared to experimental data.

Maxwell-Generalized dispersive model

$$
\begin{aligned}
&\frac{\partial \mathbf{H}}{\partial t} = -\nabla \times \mathbf{E}, \\
&\frac{\partial \mathbf{E}}{\partial t} = \frac{1}{\varepsilon_\infty}\left(\nabla \times \mathbf{H} - \mathbf{J}_s - \boldsymbol{\mathcal{J}}_0 - \sum_{l \in L_1} \boldsymbol{\mathcal{J}}_l - \sum_{l \in L_2} \boldsymbol{\mathcal{J}}_l\right), \\
&\boldsymbol{\mathcal{J}}_0 = (\sigma + \sum_{l \in L_2} d_l)\mathbf{E}, \\
&\boldsymbol{\mathcal{J}}_l = a_l \mathbf{E} - b_l \mathbf{P}_l \qquad \forall l \in L_1, \\
&\frac{\partial \mathbf{P}_l}{\partial t} = \boldsymbol{\mathcal{J}}_l \qquad \forall l \in L_1, \\
&\frac{\partial \boldsymbol{\mathcal{J}}_l}{\partial t} = (c_l - d_l f_l)\mathbf{E} - f_l \boldsymbol{\mathcal{J}}_l - e_l \mathbf{P}_l \qquad \forall l \in L_2, \\
&\frac{\partial \mathbf{P}_l}{\partial t} = d_l \mathbf{E} + \boldsymbol{\mathcal{J}}_l \qquad \forall l \in L_2.
\end{aligned}
\tag{2.14}
$$

We now switch to the numerical treatment of the previously stated systems of PDEs. The basic solver that we consider in this work combines a discontinuous Galerkin (DG) approach for the discretization in space (sections 2.4 and 2.3.3) with a fully explicit scheme for time integration scheme (section 2.5), yielding a so-called DGTD method.

2.4 Discontinous Galerkin formulation for Maxwell equations

In this section, we present the main principles and numerical ingredients of the DGTD method formulated on a conforming tetrahedral mesh and assuming an uniform interpolation order across all the elements of the mesh, which is essentially the outcome of the Ph.D. thesis of J. Viquerat

[Viq16]. In the next chapters of this manuscript, we will extend this method and explain in more detail our methodological contributions motivated by the study of light interaction with nano-structured materials for applications relevant to photovoltaics. We consider the case of Maxwell equations in non-dispersive materials for the presentation of the different aspects of the DG formulation in space (sections 2.4.1 to 2.6, and then elaborate on the main modifications for dealing with dispersive materials.

2.4.1 Weak formulation

Let $\Omega \subset \mathbb{R}^3$ be a bounded convex domain, and $\mathbf{n}$ the unitary outward normal to its boundary $\partial\Omega$. Let Ω_h be a discretization of Ω relying on a quasi-uniform triangulation $\mathcal{T}_h$ ($\exists \delta, \forall T_i \in \mathcal{T}_h, \forall k \in \mathcal{V}_i, h_k \leq \delta h_i$, where h_i is the size of the element T_i). This triangulation verifies $\mathcal{T}_h = \bigcup_{i=1}^{N} T_i$, where $N \in \mathbb{N}^*$ is the number of mesh elements and $(T_i)_{i\in[\![1,N]\!]}$ is the set of elements of $\mathcal{T}_h$. We denote by $a_{ik} = T_i \bigcap T_k$ the internal faces of between the adjacent cells T_i and T_k, and $\mathbf{n}_{ik}$ is the unit normal vector to a_{ik}, oriented from T_i to T_k. For each cell T_i, we define as $\mathcal{V}_i$ the set of indices $k \in [\![1, N]\!]$, such that T_i and T_k are sharing the face a_{ik}.

From now on, we can write the weak formulation of problem (2.9) locally in the cell T_i. Taking the L^2 scalar product of each term with a vector test function $\boldsymbol{\psi}$, we obtain the following variational problem:

Find $(\mathbf{E}, \mathbf{H}) \in H_0(\mathbf{curl}, \Omega_h) \times H(\mathbf{curl}, \Omega_h)$ such that $\forall\, \boldsymbol{\psi} \in H(\mathbf{curl}, \Omega_h)$,

$$\begin{aligned}
\int_{T_i} \mu_r \frac{\partial \mathbf{H}}{\partial t} \cdot \boldsymbol{\psi} + \int_{T_i} \nabla \times \mathbf{E} \cdot \boldsymbol{\psi} &= \mathbf{0}, \\
\int_{T_i} \varepsilon_r \frac{\partial \mathbf{E}}{\partial t} \cdot \boldsymbol{\psi} - \int_{T_i} \nabla \times \mathbf{H} \cdot \boldsymbol{\psi} &= -\int_{T_i} \mathbf{J} \cdot \boldsymbol{\psi}.
\end{aligned}$$

Then, using classical vectorial calculus and Green formulas gives:

$$\begin{aligned}
\int_{T_i} \mu_r \frac{\partial \mathbf{H}}{\partial t} \cdot \boldsymbol{\psi} + \int_{T_i} \mathbf{E} \cdot \nabla \times \boldsymbol{\psi} &= \int_{\partial T_i} (\boldsymbol{\psi} \times \mathbf{E}) \cdot \mathbf{n}_i, \\
\int_{T_i} \varepsilon_r \frac{\partial \mathbf{E}}{\partial t} \cdot \boldsymbol{\psi} - \int_{T_i} \mathbf{H} \cdot \nabla \times \boldsymbol{\psi} &= -\int_{T_i} \mathbf{J} \cdot \boldsymbol{\psi} - \int_{\partial T_i} (\boldsymbol{\psi} \times \mathbf{H}) \cdot \mathbf{n}_i.
\end{aligned}$$

One can notice that the previous equality only holds if the boundary terms $\int_{\partial T_i} (\boldsymbol{\psi} \times \mathbf{H}) \cdot \mathbf{n}_i$ exist. By using the properties of the mixed product, the previous system becomes:

$$(\boldsymbol{\psi} \times \mathbf{E}) \cdot \mathbf{n}_i = (\mathbf{E} \times \mathbf{n}_i) \cdot \boldsymbol{\psi}.$$

This implies that taking $\mathbf{E}$ in $H_0(\mathbf{curl}, \Omega_h)$ requires the existence of the trace of $\boldsymbol{\psi}$ on ∂T_i. Thus, we choose to take $\boldsymbol{\psi}$ in $H^1(\Omega_h)$. Therefore, $\forall T_i, \forall \boldsymbol{\psi} \in H^1(\Omega_h)$,

Local weak formulation

$$\begin{aligned}
\int_{T_i} \mu_r \frac{\partial \mathbf{H}}{\partial t} \cdot \boldsymbol{\psi} + \int_{T_i} \mathbf{E} \cdot \nabla \times \boldsymbol{\psi} &= \int_{\partial T_i} (\mathbf{E} \times \mathbf{n}_i) \cdot \boldsymbol{\psi}, \\
\int_{T_i} \varepsilon_r \frac{\partial \mathbf{E}}{\partial t} \cdot \boldsymbol{\psi} - \int_{T_i} \mathbf{H} \cdot \nabla \times \boldsymbol{\psi} &= -\int_{T_i} \mathbf{J} \cdot \boldsymbol{\psi} - \int_{\partial T_i} (\mathbf{H} \times \mathbf{n}_i) \cdot \boldsymbol{\psi}.
\end{aligned} \tag{2.15}$$

2.4.2 Discretization in space

We start by defining the approximation space V_h such that:

$$V_h = \left\{ v \in \left(L^2(\Omega)\right)^3, v_{|T_i} \in (\mathbb{P}_p(T_i))^3 \, \forall \, T_i \in \mathcal{T}_h \right\},$$

where $\mathbb{P}_p(T_i)$ is the space of polynomial functions of maximum degree p on T_i. We denote by $(\mathbf{H}_h, \mathbf{E}_h, \mathbf{J}_h)$ the semi-discrete fields seeked in space V_h, and there restrictions on the cell T_i as $(\mathbf{H}_i, \mathbf{E}_i, \mathbf{J}_i) = \left(\mathbf{H}_{h\,|T_i}, \mathbf{E}_{h\,|T_i}, \mathbf{J}_{h\,|T_i}\right)$. We also define a set of scalar basis functions $(\phi_{ik})_{1 \leqslant k \leqslant d_i}$ for each T_i, with d_i the number of degrees of freedom (d.o.f.) per dimension. Additionally, we associate the three vectors $\boldsymbol{\phi}^{\boldsymbol{v}}_{\boldsymbol{ik}}$ to each scalar basis function, such that:

$$\boldsymbol{\phi}^{\mathbf{1}}_{\boldsymbol{ik}} = \begin{bmatrix} \phi_{ik} \\ 0 \\ 0 \end{bmatrix}, \boldsymbol{\phi}^{\mathbf{2}}_{\boldsymbol{ik}} = \begin{bmatrix} 0 \\ \phi_{ik} \\ 0 \end{bmatrix}, \boldsymbol{\phi}^{\mathbf{3}}_{\boldsymbol{ik}} = \begin{bmatrix} 0 \\ 0 \\ \phi_{ik} \end{bmatrix}.$$

For a 3D system, we notice that $\mathbf{E}_i$ and $\mathbf{H}_i$ are two vectors of three components:

$$\mathbf{E}_i = \begin{bmatrix} E_i^x \\ E_i^y \\ E_i^z \end{bmatrix}, \quad \mathbf{H}_i = \begin{bmatrix} H_i^x \\ H_i^y \\ H_i^z \end{bmatrix},$$

which can both be locally expanded on the chosen set of basis functions:

$$E_i^v = \sum_{j=1}^{d_i} E_{ij}^v \phi_{ij}, \quad H_i^v = \sum_{j=1}^{d_i} H_{ij}^v \phi_{ij}, \quad v \in \{x, y, z\}. \tag{2.16}$$

Therefore, for practical purpose, we define the three vectors of d_i components:

$$\overline{\mathbf{E}}_i^{\,v} = \begin{bmatrix} E_{i1}^v \\ \vdots \\ E_{id_i}^v \end{bmatrix}, \quad \overline{\mathbf{H}}_i^{\,v} = \begin{bmatrix} H_{i1}^v \\ \vdots \\ H_{id_i}^v \end{bmatrix}, \quad v \in \{x, y, z\},$$

as well as the following $3d_i$ components vector:

$$\overline{\mathbf{E}}_{\,i} = \begin{bmatrix} \left(E_{ij}^x\right)_{1 \leqslant j \leqslant d_i} \\ \left(E_{ij}^y\right)_{1 \leqslant j \leqslant d_i} \\ \left(E_{ij}^z\right)_{1 \leqslant j \leqslant d_i} \end{bmatrix}, \quad \overline{\mathbf{H}}_{\,i} = \begin{bmatrix} \left(H_{ij}^x\right)_{1 \leqslant j \leqslant d_i} \\ \left(H_{ij}^y\right)_{1 \leqslant j \leqslant d_i} \\ \left(H_{ij}^z\right)_{1 \leqslant j \leqslant d_i} \end{bmatrix}.$$

Those vectors will be used to write the matrix-vector semi-discrete variational formulation of our system, analogously to (2.15). However, before that, the boundary terms in this formulation require some particular treatment before going further on in the spatial discretization process.

2.4.3 Numerical fluxes

Given that the test functions $\boldsymbol{\psi}$ can now be discontinuous at the interfaces a_{il} (between cells T_i and T_l), it is important to notice that the surface integrals, define as:

$$\int_{a_{il}} (\mathbf{E}_h \times \mathbf{n}_{il}) \cdot \boldsymbol{\psi}, \quad \text{and} \quad \int_{a_{il}} (\mathbf{H}_h \times \mathbf{n}_{il}) \cdot \boldsymbol{\psi}, \tag{2.17}$$

are not clearly define. Indeed, electromagnetic fields $\mathbf{E}_h$ and $\mathbf{H}_h$ can relate to either the field value in T_i or in T_l. We need to define the numerical fluxes (or traces), which will allow to recover a proper definition of the surface integrals. This will permit to reconnect the field values between neighbouring cells.

***Remark** : We specify that the numerical fluxes do not have a unique valid choice. In fact, different choices can lead to stable and convergent discrete schemes, with different numerical properties.*

We replace in equations (2.17) $\mathbf{E}_h$ and $\mathbf{H}_h$ by $\mathbf{E}_*$ and $\mathbf{H}_*$ such that:

$$\int_{a_{il}} (\mathbf{E}_* \times \mathbf{n}) \cdot \boldsymbol{\psi}, \quad \text{and} \quad \int_{a_{il}} (\mathbf{H}_* \times \mathbf{n}) \cdot \boldsymbol{\psi}.$$

It is well known that the flux can be seen as the solution of a Riemann problem at cell interfaces. In this thesis, we will exploit two very common flux choices. The first one is the centered flux, which is defined as:

$$\mathbf{E}_* = \frac{\mathbf{E}_i + \mathbf{E}_l}{2}, \quad \text{and} \quad \mathbf{H}_* = \frac{\mathbf{H}_i + \mathbf{H}_l}{2}. \tag{2.18}$$

This flux is non-dissipative and leads to an L^2 spatial convergence in h^p. When it is coupled to a non-dissipative time-integration scheme (for example Leap-Frog), this yields to a non-dissipative DGTD scheme [FLLP05]. The second numerical flux considered is the upwind flux, which is defined as:

$$\mathbf{E}_* = \frac{1}{Y_i + Y_l} \left(\{Y\mathbf{E}\}_{il} + \alpha \mathbf{n} \times [\![\mathbf{H}]\!]_{il} \right), \quad \text{and} \quad \mathbf{H}_* = \frac{1}{Z_i + Z_l} \left(\{Z\mathbf{H}\}_{il} - \alpha \mathbf{n} \times [\![\mathbf{E}]\!]_{il} \right), \tag{2.19}$$

where $\{A\}_{il} = A_i + A_l$ is twice the mean value of A at the interface, $[\![A]\!]_{il} = A_l - A_i$ is the jump of A at the interface, and $\alpha \in [0, 1]$ is a tunable parameter that allows to vary between the centered flux (2.18) taking $\alpha = 0$, to a fully upwind flux for $\alpha = 1$. The jump term of the upwind flux introduces dissipation in the DG scheme. Those dissipation can help damping nonphysical modes when instabilities occur [HW08]. It also leads to a better L^2 spatial convergence (as h^{p+1} against h^p for the centered flux).

2.4.4 DG matrices

In this part, we present the FE matrices which will allow us to write the matrix-vector form of system 2.14. We choose the test functions $\boldsymbol{\psi}$ such that they are the $3\,d_i$ vectors $\boldsymbol{\phi}^{\boldsymbol{v}}_{\boldsymbol{ik}}$, which constitutes the Galerkin choice:

$$\begin{aligned}
&\int_{T_i} \mu_r \frac{\partial \mathbf{H}_i}{\partial t} \cdot \boldsymbol{\phi}^{\boldsymbol{v}}_{\boldsymbol{ik}} + \int_{T_i} \mathbf{E}_i \cdot \nabla \times \boldsymbol{\phi}^{\boldsymbol{v}}_{\boldsymbol{ik}} = \sum_{l \in \mathcal{V}_i} \int_{a_{il}} (\mathbf{E}_* \times \mathbf{n}_{il}) \cdot \boldsymbol{\phi}^{\boldsymbol{v}}_{\boldsymbol{ik}}, \\
&\int_{T_i} \varepsilon_r \frac{\partial \mathbf{E}_i}{\partial t} \cdot \boldsymbol{\phi}^{\boldsymbol{v}}_{\boldsymbol{ik}} - \int_{T_i} \mathbf{H}_i \cdot \nabla \times \boldsymbol{\phi}^{\boldsymbol{v}}_{\boldsymbol{ik}} = -\sum_{l \in \mathcal{V}_i} \int_{a_{il}} (\mathbf{H}_* \times \mathbf{n}_{il}) \cdot \boldsymbol{\phi}^{\boldsymbol{v}}_{\boldsymbol{ik}} - \int_{T_i} \mathbf{J}_i \cdot \boldsymbol{\phi}^{\boldsymbol{v}}_{\boldsymbol{ik}}.
\end{aligned} \tag{2.20}$$

Mass matrix We start by considering the time-derivative term of the **E** evolutionary equation of system (2.20). The x component is:

$$\begin{aligned}\int_{T_i} \varepsilon_r \frac{\partial \mathbf{E}_i}{\partial t} \cdot \boldsymbol{\phi}_{\boldsymbol{ik}}^{\boldsymbol{x}} &= \int_{T_i} \varepsilon_r \frac{\partial E_i^x}{\partial t} \phi_{ik} \\ &= \sum_{j=1}^{d_i} \frac{\partial}{\partial t} E_{ij}^x \int_{T_i} \varepsilon_r \phi_{ij} \phi_{ik} \\ &= \left(\mathbb{M}_i^{\varepsilon_r} \frac{\partial \overline{\mathbf{E}}_i^{\,x}}{\partial t} \right)_k\end{aligned}$$

where $\mathbb{M}_i^{\varepsilon_r}$ is the mass matrix, of dimension $d_i \times d_i$:

$$\left(\mathbb{M}_i^{\varepsilon_r}\right)_{jk} = \int_{T_i} \varepsilon_r \phi_{ij} \phi_{ik},$$

with $(j,k) \in [\![1, d_i]\!]^2$.

Stiffness matrices We now focus on the curl integral of the **E** evolutionary equation of system 2.20. The x component is:

$$\begin{aligned}\int_{T_i} \mathbf{H}_i \cdot \nabla \times \boldsymbol{\phi}_{\boldsymbol{ik}}^{\boldsymbol{x}} &= \int_{T_i} \left(H_i^y \frac{\partial \phi_{ik}}{\partial z} - H_i^z \frac{\partial \phi_{ik}}{\partial y} \right) \\ &= \int_{T_i} \sum_{j=1}^{d_i} \left(H_{ij}^y \phi_{ij} \frac{\partial \phi_{ik}}{\partial z} - H_{ij}^z \phi_{ij} \frac{\partial \phi_{ik}}{\partial y} \right) \\ &= \sum_{j=1}^{d_i} H_{ij}^y \int_{T_i} \phi_{ij} \frac{\partial \phi_{ik}}{\partial z} - \sum_{j=1}^{d_i} H_{ij}^z \int_{T_i} \phi_{ij} \frac{\partial \phi_{ik}}{\partial y} \\ &= \left(\mathbb{K}_i^z \overline{\mathbf{H}}_i^{\,y} - \mathbb{K}_i^y \overline{\mathbf{H}}_i^{\,z} \right)_k \\ &= - \left(\overline{\mathbb{K}}_i \times \overline{\mathbf{H}}_i \right)_k^x .\end{aligned}$$

with the three stiffness matrices defined as:

$$\left(\mathbb{K}_i^v\right)_{jk} = \int_{T_i} \phi_{ij} \frac{\partial \phi_{ik}}{\partial v} \quad \text{for} \quad v \in \{x, y, z\},$$

with $(j,k) \in [\![1, d_i]\!]^2$. We can also define the global $3d_i \times d_i$ stiffness matrix that will be used in the matrix-vector form of the final system:

$$\overline{\mathbb{K}}_i = \begin{bmatrix} \mathbb{K}_i^x \\ \mathbb{K}_i^y \\ \mathbb{K}_i^z \end{bmatrix},$$

Flux matrix Finally, we consider the right member of system 2.20 that contains the flux contribution. Here we are using the centered flux (2.18), but the upwind case, as well as the generalization to other fluxes is straightforward. We also note that in the conforming case, the field expansion

defined on a_{il} over the basis functions of cells T_i or T_l is equivalent. We proceed as previously, focusing on the x component of the flux of the $\mathbf{E}$ evolutionary equation of system 2.20:

$$\begin{aligned}\int_{a_{il}} (\mathbf{H}_* \times \mathbf{n}_{il}) \cdot \boldsymbol{\phi}_{ik}^{\boldsymbol{x}} &= \int_{a_{il}} \left(H_*^y n_{il}^z - H_*^z n_{il}^y\right) \phi_{ik} \\ &= \int_{a_{il}} \left(\frac{H_i^y + H_l^y}{2} n_{il}^z - \frac{H_i^z + H_l^z}{2} n_{il}^y\right) \phi_{ik} \\ &= \frac{1}{2} \sum_j^{d_i} \left(\{H^y\}_{il} n_{il}^z - \{H^z\}_{il} n_{il}^y\right) \int_{a_{il}} \phi_{ij}\phi_{ik} \\ &= \left(\mathbb{S}_{il} \left(\overline{\mathbf{H}}_* \times \mathbf{n}_{il}\right)\right)_k^x\end{aligned}$$

where the flux matrices are:

$$\left(\mathbb{S}_{il}\right)_{jk} = \int_{a_{il}} \phi_{ij}\phi_{ik},$$

with $(j,k) \in [\![1, d_i]\!]^2$.

Remark *: In non-conforming cases (which includes hybrid meshes, non-conforming meshes and local approximation order), one should consider the proper expansions of the field over both cells respective bases. This is presented in details in the chapter 3.*

Semi-discrete formulation Thanks to the definition of the FE elementary matrices, we can define both global mass and flux matrices, which will allow to write the semi-discrete formulation in a compact form:

$$\overline{\mathbb{M}}_i^u = \begin{bmatrix} \mathbb{M}_i^u & \mathbf{0}_{d_i \times d_i} & \mathbf{0}_{d_i \times d_i} \\ \mathbf{0}_{d_i \times d_i} & \mathbb{M}_i^u & \mathbf{0}_{d_i \times d_i} \\ \mathbf{0}_{d_i \times d_i} & \mathbf{0}_{d_i \times d_i} & \mathbb{M}_i^u \end{bmatrix}, \quad \overline{\mathbb{S}}_{il} = \begin{bmatrix} \mathbb{S}_{il} & \mathbf{0}_{d_i \times d_i} & \mathbf{0}_{d_i \times d_i} \\ \mathbf{0}_{d_i \times d_i} & \mathbb{S}_{il} & \mathbf{0}_{d_i \times d_i} \\ \mathbf{0}_{d_i \times d_i} & \mathbf{0}_{d_i \times d_i} & \mathbb{S}_{il}, \end{bmatrix}.$$

which leads to the following matrix vector expression of the semi-discrete DG scheme for Maxwell's equations:

Semi-discrete scheme

$$\begin{aligned}\overline{\mathbb{M}}_i^{\mu_r} \frac{\partial \overline{\mathbf{H}}_i}{\partial t} &= -\overline{\mathbb{K}}_i \times \overline{\mathbf{E}}_i + \sum_{l \in \mathcal{V}_i} \overline{\mathbb{S}}_{il} \left(\overline{\mathbf{E}}_* \times \mathbf{n}_{il}\right), \\ \overline{\mathbb{M}}_i^{\varepsilon_r} \frac{\partial \overline{\mathbf{E}}_i}{\partial t} &= \quad \overline{\mathbb{K}}_i \times \overline{\mathbf{H}}_i - \sum_{l \in \mathcal{V}_i} \overline{\mathbb{S}}_{il} \left(\overline{\mathbf{H}}_* \times \mathbf{n}_{il}\right) - \overline{\mathbb{M}}_i \overline{\mathbf{J}}_i.\end{aligned} \tag{2.21}$$

2.4.5 Elements mapping

One of the strength of the DG method is that the FE matrices described in 2.4.4 are not stored for each element of $\mathcal{T}_h$, but are calculated only once for all on a reference element noted $\widehat{T}$ and then mapped on the considered physical tetrahedron T_i. Let $\widehat{T}$ be defined as follows in the $\boldsymbol{\xi} = (\xi, \eta, \zeta)$ coordinates system:

$$\widehat{T} = \left\{ (\xi, \eta, \zeta) \in \mathbb{R}^3_+, \xi + \eta + \zeta \leq 1 \right\}.$$

Then we define the physical tetrahedron in the $\mathbf{x} = (x, y, z)$ coordinates system as the image of $\widehat{T}$ by a mapping $\boldsymbol{\psi}_{T_i}$ (see figure 2.9) defined as:

$$\boldsymbol{\psi}_{T_i} : \widehat{T} \to T_i, \text{ such that, } \forall \boldsymbol{\xi} \in \widehat{T}, \ \mathbf{x} = \boldsymbol{\psi}_{T_i}(\boldsymbol{\xi}).$$

The vertices of $\widehat{T}$ are noted (A_1, A_2, A_3, A_4), and the vertices of T_i are noted (v_1, v_2, v_3, v_4). In this case, $\boldsymbol{\psi}$ is a linear combination of ξ, η and ζ defined as:

$$\boldsymbol{\psi}_{T_i}(\boldsymbol{\xi}) = v_1 + (v_2 - v_1)\xi + (v_3 - v_1)\eta + (v_4 - v_1)\zeta.$$

Let $(\phi_{ij})_{j=1..d_i}$ be the basis functions on T_i, and $\left(\widehat{\phi}_j\right)_{j=1..d_i}$ defined by $\widehat{\phi}_j = \phi_{ij} \circ \boldsymbol{\psi}_{T_i}$ on $\widehat{T}$. Then

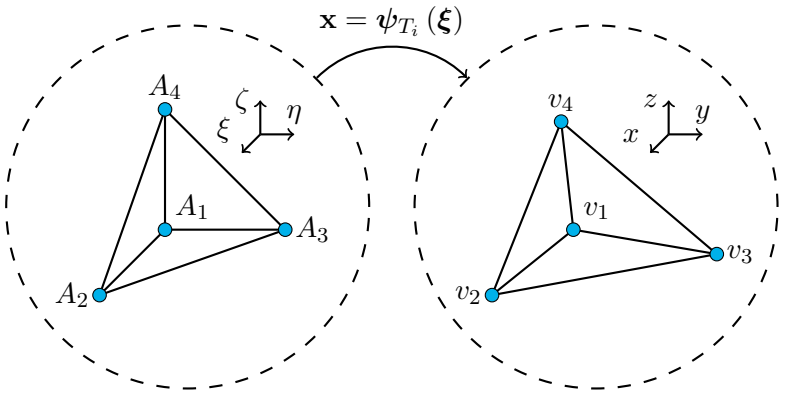


Figure 2.9 | Linear mapping from the reference element $\widehat{T}$ to the physical element T_i.

we can write the mass matrix on the element T_i as:

$$\begin{aligned}(\mathbb{M}_i)_{jk} &= \int_{T_i} \phi_{ij}(\mathbf{x})\phi_{ik}(\mathbf{x})d\mathbf{x} \\ &= \int_{\widehat{T}} \widehat{\phi}_j(\boldsymbol{\xi})\,\widehat{\phi}_k(\boldsymbol{\xi}) \left|\mathbf{J}_{\boldsymbol{\psi}_{T_i}}\right| d\boldsymbol{\xi},\end{aligned}$$

where $\mathbf{J}_{\boldsymbol{\psi}_{T_i}}(\boldsymbol{\xi})$ is the jacobian matrix of the mapping $\boldsymbol{\psi}$, defined as:

$$\left(\mathbf{J}_{\boldsymbol{\psi}_{T_i}}\right)_{jl} = \left(\frac{\partial \mathbf{x}_j}{\partial \boldsymbol{\xi}_l}\right)_{jl} = \begin{bmatrix} (v_2 - v_1)_x & (v_3 - v_1)_x & (v_4 - v_1)_x \\ (v_2 - v_1)_y & (v_3 - v_1)_y & (v_4 - v_1)_y \\ (v_2 - v_1)_z & (v_3 - v_1)_z & (v_4 - v_1)_z \end{bmatrix}.$$

In this case, the determinant $\left|\mathbf{J}_{\boldsymbol{\psi}_{T_i}}\right|$ is constant and only depend on (v_1, v_2, v_3, v_4). Hence, the mass matrix for each physical tetrahedron of $\mathcal{T}_h$ is a simple multiplication of the mass matrix calculated on the reference tetrahedron:

$$(\mathbb{M}_i)_{jk} = \left|\mathbf{J}_{\boldsymbol{\psi}_{T_i}}\right| \left(\widehat{\mathbb{M}}\right)_{jk}.$$

Similar situation occurs for stiffness and flux matrices by using the following change of variables (see [Mon03] for additional details):

$$
\begin{aligned}
\left(\mathbb{K}_i^v\right)_{jk} &= \int_{T_i} \left(\phi_{ij}(\mathbf{x}) \frac{\partial \phi_{ik}}{\partial v}(\mathbf{x}) \right) d\mathbf{x} \\
&= \int_{\widehat{T}} \left(\widehat{\phi}_j(\boldsymbol{\xi}) \left[\left| \mathbf{J}_{\boldsymbol{\psi}_{T_i}} \right| \mathbf{J}_{\boldsymbol{\psi}_{T_i}}^{-1} \nabla_{\boldsymbol{\xi}} \widehat{\phi}_k(\boldsymbol{\xi}) \right]_v \right) d\boldsymbol{\xi} \\
&= \left[\left| \mathbf{J}_{\boldsymbol{\psi}_{T_i}} \right| \mathbf{J}_{\boldsymbol{\psi}_{T_i}}^{-1} \int_{\widehat{T}} \widehat{\phi}_j(\boldsymbol{\xi}) \nabla_{\boldsymbol{\xi}} \widehat{\phi}_k(\boldsymbol{\xi}) d\boldsymbol{\xi} \right]_v \\
&= \sum_{m=1}^{3} \left[\left| \mathbf{J}_{\boldsymbol{\psi}_{T_i}} \right| \mathbf{J}_{\boldsymbol{\psi}_{T_i}}^{-1} \right]_{vm} \int_{\widehat{T}} \widehat{\phi}_j \frac{\partial \widehat{\phi}_k}{\partial \boldsymbol{\xi}_m} \\
&= \sum_{m=1}^{3} \left[\left| \mathbf{J}_{\boldsymbol{\psi}_{T_i}} \right| \mathbf{J}_{\boldsymbol{\psi}_{T_i}}^{-1} \right]_{vm} \left(\widehat{\mathbb{K}}^m \right)_{jk},
\end{aligned}
$$

where $\nabla_{\boldsymbol{\xi}}$ is the gradient operator in the $\widehat{T}$ basis. Therefore, the stiffness matrix on T_i can be built from the stiffness matrices $\left(\widehat{\mathbb{K}}^m\right)_{jk}$ calculated on the reference element. Finally, the flux matrix can be rewritten as:

$$
\begin{aligned}
\left(\mathbb{S}_{il}\right)_{jk} &= \int_{a_{il}} \phi_{ij} \phi_{lk} d\mathbf{s} \\
&= \int_{\widehat{t}} \widehat{\phi}_j \widehat{\phi}_k \left| \mathbf{J}_{\boldsymbol{\psi}_{T_i}} \right| \left| \mathbf{J}_{\boldsymbol{\psi}_{T_i}}^{-1} \widehat{\mathbf{n}} \right| d\widehat{\mathbf{s}} \\
&= \left| \mathbf{J}_{\boldsymbol{\psi}_{T_i}} \right| \left| \mathbf{J}_{\boldsymbol{\psi}_{T_i}}^{-1} \widehat{\mathbf{n}} \right| \left(\widehat{\mathbb{S}} \right)_{jk},
\end{aligned}
$$

with $\widehat{t}$ the reference triangle. As for the other matrices, these matrices can be deduced from the $\left(\widehat{\mathbb{S}}\right)_{jk}$. The unitary normals also transform as:

$$
\mathbf{n}_{il} = \frac{\mathbf{J}_{\boldsymbol{\psi}_{T_i}}^{-1} \widehat{\mathbf{n}}}{\left| \mathbf{J}_{\boldsymbol{\psi}_{T_i}}^{-1} \widehat{\mathbf{n}} \right|}.
$$

Therefore, even for the flux matrix, we only need to store a single (3×3) matrix instead of once per cell.

***Remark** : If one is using local polynomial orders, flux matrices must be computed for each possible order's jump. If the mesh is non-conform or hybrid, flux matrices must be computed for all the non-conformal interfaces.*

2.4.6 Polynomial expansion basis

In this part we discuss the choice of the basis function $\left(\widehat{\phi}_j\right)_{j=1..d_i}$. Over all the possible polynomial bases available, Lagrange polynomials are quite a common choice. They can be defined by a set of interpolation nodes distributed across the cell.

The Lagrange interpolants L_i are defined by the following property:

$$
L_i(\mathbf{x}_j) = \delta_{ij}, \forall (i,j) \in [\![1, d_i]\!]^2.
$$

There must be the same number of nodes and polynomials in order to define a complete basis. In a tetrahedron the number of Lagrande nodes needed to have a polynomial order p is equal to:

$$n(p) = \frac{(p+1)(p+2)(p+3)}{6},$$

and for each of its faces, it is:

$$s(p) = \frac{(p+1)(p+2)}{2}.$$

We choose to use Lagrange polynomials with equispaced nodes as it allows a simple integration of the FE matrices on the reference element since the nodes positions are known exactly.

***Remark** : In the case of an hexahedral cell, the number of Lagrange nodes requires for a polynomial order q is:*

$$n(q) = (q+1)^3 \quad and \quad s(q) = (q+1)^2.$$

2.4.7 DG operator spectrum

With DG, the global matrix of the spatial operator can be assemble such as FEM. Compared to the FEM global matrix, the DG one is of much larger dimension, even for the same number of cells, and the same interpolation order. This is due to the degree of freedom (d.o.f.) which are duplicated on the cell's interfaces. However, unlike the FEM global matrix, local matrices do not overlap. Indeed, the degrees of freedom exist independently at each cell interfaces on both sides. Eventually, we can rewrite the semi-discrete DG problem as the following generic form:

$$\frac{\partial \mathbf{W}_h}{\partial t}(t) = \mathbb{A}_h \mathbf{W}_h(t) + \mathbf{B}_h(t), \tag{2.22}$$

where $\mathbf{B}_h(t)$ represents the exterior sources, and the matrix $\mathbb{A}_h$ holds all the information relative to the spatial discretization.

2.5 Time discretization

In section 2.4, the semi-discrete scheme was obtained (2.21) by discretizing the spatial derivatives of Maxwell's equations. Similarly, the time derivatives need to be considered for the time discretization. Then, we consider the reduced problem of the 1D Maxwell's equations:

$$\begin{aligned} \mu_r \frac{\partial H_y}{\partial t} &= \frac{\partial E_z}{\partial x}, \\ \varepsilon_r \frac{\partial E_z}{\partial t} &= \frac{\partial H_y}{\partial x} + j(t). \end{aligned}$$

By dropping the spatial subscripts of the unknowns, the semi-discrete formulation associated to this system becomes:

$$\begin{aligned} \mathbb{M}_i^{\mu_r} \frac{\partial H_i}{\partial t} &= \mathbb{K}_i E_i + [E_*]_{x_{i-1}}^{x_i}, \\ \mathbb{M}_i^{\varepsilon_r} \frac{\partial E_i}{\partial t} &= \mathbb{K}_i H_i + [H_*]_{x_{i-1}}^{x_i} - \mathbb{M}_i J_i, \end{aligned} \tag{2.23}$$

with:

$$E_* = \frac{1}{Y_i + Y_l} \left(\{YE\}_{il} + \alpha \, [\![H]\!]_{il} \right), \quad H_* = \frac{1}{Z_i + Z_l} \left(\{ZH\}_{il} + \alpha \, [\![E]\!]_{il} \right).$$

In the previous equalities, subscript l designates $i-1$ or $i+1$. When trying to go from system (2.23) to system (2.22), a non-symmetric $\mathbb{A}$ matrix is obtained in the general case:

$$\mathbb{A} = \begin{bmatrix} \mathbb{A}_{\alpha,H} & \mathbb{A}_H \\ \mathbb{A}_E & \mathbb{A}_{\alpha,E} \end{bmatrix}.$$

The off-diagonal blocks represent both the centered part of the flow and the stiffness part, while the diagonal blocks represent the upwind contribution. Therefore, $\mathbb{A}$ is strictly anti-diagonal in the case of centered flows. Furthermore, the following properties are obtained in the case of a homogeneous medium ($Y_i = Y_l = Y$):

1. $\mathbb{A}_{\alpha,H}$ and $\mathbb{A}_{\alpha,E}$ are equal, *e.g.* $\mathbb{A}_{\alpha,H} = \mathbb{A}_{\alpha,E} \equiv \mathbb{A}_\alpha$;
2. $\mathbb{A}_H$ and $\mathbb{A}_E$ are multiples of each other, *e.g.* $\mathbb{A}_H = \frac{\varepsilon_r}{\mu_r} \mathbb{A}_E \equiv Y^2 \frac{\varepsilon_r}{\mu_r} \mathbb{A}_U$.

To begin with, we consider the simple case of vacuum ($\varepsilon_r = \mu_r = 1$). In this case, one has a symmetric $\mathbb{A}$ matrix and thus a diagonalizable system. Thus, one can just retain the corresponding formulation in the diagonalized basis, for the study of the time-stepping schemes. This has the effect of reducing to a system of ODEs of the form:

$$\frac{\partial \phi}{\partial t}(t) = \lambda \phi(t) + b(t) \equiv f(t, \phi(t)), \tag{2.24}$$

for each λ eigenvalue of $\mathbb{A}$. A number of time-integration techniques from the ODE community are appropriate to solve (2.24).

We can classified time-stepping methods into two main categories: the explicit time integration techniques, and the implicit methods. For the first category, the time state $\phi(t + \Delta t)$ is computed explicitly from $\phi(t)$. For the second category, the time-updated solution is obtained by solving an implicit expression of the form $g(\phi(t), \phi(t + \Delta t)) = 0$. It leads to the resolution of a linear system of equations at each time-step. As expected, the implicit method is more expensive than explicit technique. But, implicit methods are generally unconditionally stable. This means that any choice of Δt leads to a stable algorithm. For explicit methods, a numerical criterion on the time-step called the Courant–Friedrichs–Lewy (CFL) condition, must be respected. Otherwise, the resulting algorithm will be unstable and blow up. In conclusion, explicit time-stepping generaly requires much more time-steps but each time-step is significantly less numerically costly. Hereafter, solutions are investigated on intervals of the form $[0, T]$ with $T > 0$, discretized in time-steps of length Δt. In order to refer the discrete approximation of $\phi(t_n)$, with $t_n = n\Delta t$, we used the notation ϕ^n. A family of DG methods, called space-time DG methods [PFT00] [AAPG14], is designed to deal with time derivative similarly to space. The main disadvantage of these methods is that they usually lead to an implicit scheme. In the next part, we present the Runge-Kutta (RK) and Low- Storage Runge-Kutta (LSRK) schemes.

2.5.1 Classical RK schemes

Runge-Kutta time schemes are a class of multi-stage algorithms based on the multiple evaluations of the right side of (2.24) to evolve the system in time. Suppose that problem (2.24) is integrated between t and $t + \Delta t$:

$$\phi(t + \Delta t) = \phi(t) + \int_t^{t+\Delta t} f(u, \phi(u)) du. \tag{2.25}$$

We can approximate (2.25) using a quadrature formula with s terms:

$$\phi(t+\Delta t) \simeq \phi(t) + \Delta t \sum_{j=1}^{s} \beta_j f(t+\delta_j \Delta t, \phi(t+\delta_j \Delta t))du, \tag{2.26}$$

where $(\beta_j)_{j\in[\![1,s]\!]}$ and $(\delta_j)_{j\in[\![1,s]\!]}$ are constants depending of the choice of the quadrature formula. In order to evaluate the $\phi(t+\delta_j\Delta t)$ values, RK methods use a so-called prediction/correction technique. It builds on the previous guesses to calculate the next one. The n^{th} time-step with an s-stage RK algorithm is can be written as follow:

$$\begin{aligned}
\phi_1 &= f\left(t_n, \phi^n\right) \\
\phi_k &= f\left(t_n + \delta_k \Delta t, \phi^n + \Delta t \sum_{j=1}^{s} \alpha_{j,k}\phi_j\right) \quad \text{for} \quad k = 2, ..., s, \\
\phi^{n+1} &= \phi^n + \Delta t \sum_{j=1}^{s} \beta_j \phi_j,
\end{aligned} \tag{2.27}$$

where we suppose that $\phi_0 = \phi^n$. The system (2.27) is implicit in the general case. The summation in each intermediate stage extends to the maximum number of stages. For practical and efficacity reason, we choose to work with explicit RK schemes (*i.e.* $\alpha_{j,k} = 0 \,\forall k \geq j$). They can be written as:

Explicit RK schemes

$$\begin{aligned}
\phi_1 &= f\left(t_n, \phi^n\right) \\
\phi_k &= f\left(t_n + \delta_k \Delta t, \phi^n + \Delta t \sum_{j=1}^{k-1} \alpha_{j,k}\phi_j\right) \quad \text{for} \quad k = 2, ..., s, \\
\phi^{n+1} &= \phi^n + \Delta t \sum_{j=1}^{s} \beta_j \phi_j.
\end{aligned} \tag{2.28}$$

For larger number of stages, the principle of a RK algorithm remains similar to the two stages algorithm. At time-step n, the first stage of the method is:

$$\phi_1 = f\left(t_n, \phi^n\right) \equiv f_{t_n}, \tag{2.29}$$

and the second stage is:

$$\phi_2 = f\left(t_n + \delta_2 \Delta t, \phi^n + \Delta t \alpha_{1,2}\phi_1\right). \tag{2.30}$$

In (2.30), one can recognize $\phi^n + \Delta t\alpha_{1,2}\phi_1$ as a forward Euler method, that provides a first-order approximation of $\phi(t_n + \alpha_{1,2}\Delta t)$ by exploiting the approximate value of the slope at $t = t_n$ calculated during the stage (2.29). One can obtained ϕ_2 by setting $\delta_2 = \alpha_{1,2}$. The method is done

in two stages then, the solution at the next time-step is obtained as:

$$\begin{aligned}\phi^{n+1} &= \phi^n + \Delta t\left(\beta_1\phi_1 + \beta_2\phi_2\right)\\ &= \phi^n + \Delta t\left(\beta_1 f_{t_n} + \beta_2 f\left(t_n + \delta_2\Delta t, \phi^n + \Delta t\alpha_{1,2}\phi_1\right)\right)\\ &= \phi^n + \Delta t\left(\beta_1 f_{t_n} + \beta_2 f\left(t_n + \alpha_{1,2}\Delta t, \phi^n + \Delta t\alpha_{1,2}\phi_1\right)\right).\end{aligned}$$

Using Taylor expansion to the first order in Δt on the last term gives:

$$\phi^{n+1} = \phi^n + \Delta t\left(\beta_1+\beta_2\right) f_{t_n} + \beta_2\alpha_{1,2}\Delta t^2\left(\frac{\partial f}{\partial t}(t_n,\phi^n) + f_{t_n}\frac{\partial f}{\partial \phi}(t_n,\phi^n)\right) + O\left(\Delta t^3\right),$$

that can be matched with the Taylor expansion of ϕ, which gives:

$$\phi(t+\Delta t) = \phi(t) + \Delta t f_t + \frac{\Delta t^2}{2}\left(\frac{\partial f}{\partial t}(t,\phi) + f_t\frac{\partial f}{\partial \phi}(t,\phi)\right) + O\left(\Delta t^3\right),$$

where $f_t = f(t,\phi(t))$. In practice we drop the residual $O\left(\Delta t^3\right)$, that leads to the following system:

$$\begin{aligned}\beta_1 + \beta_2 &= 1,\\ \beta_2\alpha_{1,2} &= \frac{1}{2}.\end{aligned} \tag{2.31}$$

One can see RK schemes as a quadrature rule for which the sum of the weights is one, as recall in the first equation of (2.31). One also remark that (2.31) is underdetermined, which means that there is not a unique set of coefficients verifying (2.31). A common choice to satisfy the latter system is:

$$\begin{aligned}\alpha_{1,2} &= 1,\\ \beta_1 = \beta_2 &= \frac{1}{2}.\end{aligned} \tag{2.32}$$

This choice leads to a second-order RK method:

$$\begin{aligned}\phi_1 &= f\left(t_n, \phi^n\right)\\ \phi_2 &= f\left(t_n + \Delta t, \phi^n + \Delta t\phi_1\right)\\ \phi^{n+1} &= \phi^n + \frac{\Delta t}{2}\left(\phi_1 + \phi_2\right).\end{aligned} \tag{2.33}$$

Here we summarize the algorithm which is illustrated in figure 2.10. An estimate of $\frac{\partial\phi}{\partial t}(t_n)$ is given by f_{t_n} during the first step. Then, via a forward Euler step, we obtain a first-order approximation $\phi_*^{n+1} = \phi^n + \Delta t\phi_1$ of $\phi(t+\Delta t)$. In the second step, ϕ_*^{n+1} is used to estimate $\frac{\partial\phi}{\partial t}(t_{n+1})$. Finally, a second forward Euler step is used to obtain ϕ^{n+1}.

Similarly, one can obtained the four-step fourth-order RK algorithm by exploiting the average of four different evaluation of the slope (figure 2.11). The most classical version of this scheme is:

$$\begin{aligned}\phi_1 &= f\left(t_n, \phi^n\right)\\ \phi_2 &= f\left(t_n + \frac{\Delta t}{2}, \phi^n + \frac{\Delta t}{2}\phi_1\right)\\ \phi_3 &= f\left(t_n + \frac{\Delta t}{2}, \phi^n + \frac{\Delta t}{2}\phi_2\right)\\ \phi_4 &= f\left(t_n + \Delta t, \phi^n + \Delta t\phi_3\right)\\ \phi^{n+1} &= \phi^n + \frac{\Delta t}{6}\left(\phi_1 + 2\phi_2 + 2\phi_3 + \phi_4\right).\end{aligned} \tag{2.34}$$

Remark *: Obtaining an RK scheme of order p when $p > 4$ requires more than p stages ([But87, Lam91]).*

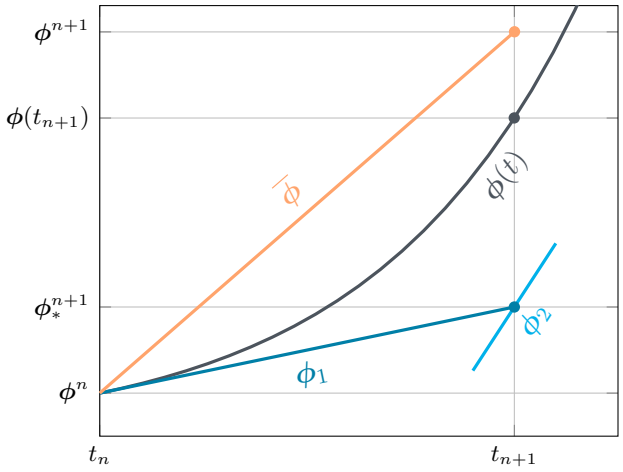


Figure 2.10 | Steps of the RK2 algorithm. ϕ_*^{n+1} is the estimate obtained by the forward Euler method. The slope of the solution between t_n and $t_n + \Delta t$ is approached by ϕ_1. The RK2 estimate is obtained by averaging ϕ_1 with a second approximation of the slope obtained from ϕ_*^{n+1} in $t_n + \Delta t$. $\overline{\phi}$ represents the average of the different slopes.

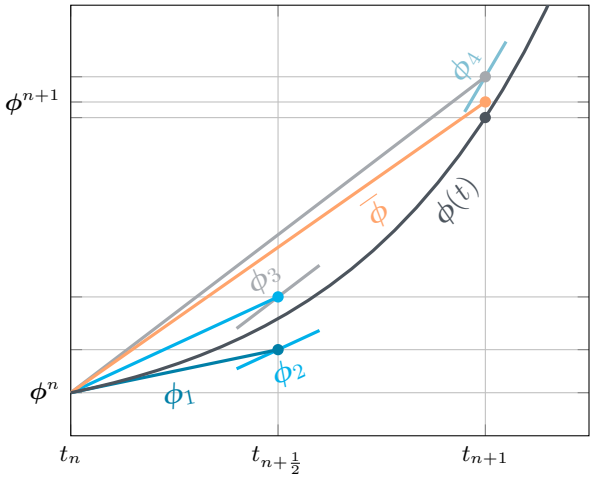


Figure 2.11 | Steps of the RK4 algorithm. $\overline{\phi}$ represents the average of the different slopes.

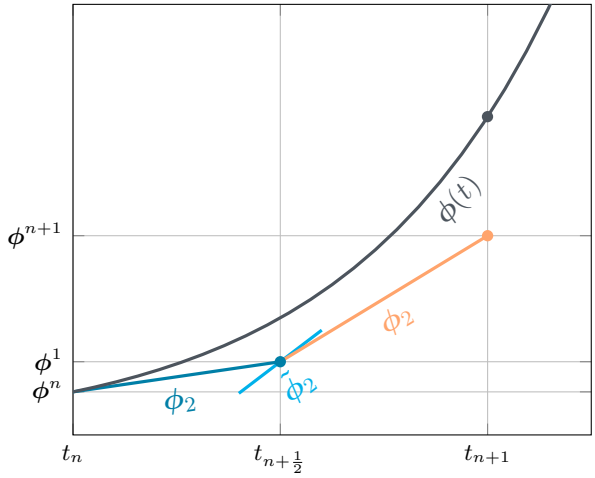


Figure 2.12 | Steps of the LSRK2 algorithm. Here, $\tilde{\phi}_2$ represents $\Delta t f\left(t_n + \frac{\Delta t}{2}, \phi^1\right)$ with 2 stages.

2.5.2 Low-storage RK schemes

Standard RK schemes can achieve high order accuracy. However, these algorithms needs to store all the s slope estimations in addition to the usual solutions. If this is not a major constraint for small problems and/or for a small number of substeps, this storage become a huge limitation for larger problems. Thus, a specific class of RK algorithms have been developed to overcome this issue. These algorithms require only two memory registers at every moment. They are called low-storage RK schemes, and they are usually presented under the Williamson formulation [Wil80]:

Low-storage RK schemes

$$\left.\begin{array}{rcl} \phi_1 & = & \phi^n \\ \phi_2 & = & a_k\phi_2 + \Delta t f\left(t_n + d_k\Delta t, \phi_1\right), \\ \phi_1 & = & \phi_1 + b_k\phi_2, \\ \phi^{n+1} & = & \phi_1. \end{array}\right\} \text{ for } k = 1, ..., s \tag{2.35}$$

Remark *: The convergence to order p is not anymore ensured in s substeps. At least 5 stages are necessary to achieve a fourth-order convergence as see in [CK94].*

The principle of LSRK schemes is slightly different from RK schemes. The final step is mixed within the slope estimation steps: starting from the initial point, the slope is approximated (with ϕ_2), and an Euler step is executed, giving an intermediate value of the solution (ϕ_1) at time station $t_n + b_1$. Then during the second stage, the slope stored in ϕ_2 receive a weighted average between the previous estimate and the new one (calculated from ϕ_1 at $t_n + b_1$). This is illustrated in figure 2.12.

2.5.3 Time-step and CFL condition

The main drawback of explicit methods is the CFL condition. This time-step restriction is computed from an energy-based stability study. In this thesis, theoretical results from [FLLP05] are

used. Consequently, for a space discretization with polynomial order p, the time-step is chosen as follows:

$$\Delta t_p = c_p \min_{T_i \in \mathcal{T}_h} \frac{V_{T_i}}{A_{T_i}}, \tag{2.36}$$

where V_{T_i} is the volume, A_{T_i} is the area of cell T_i, and c_p is an order-dependent constant. The maximal acceptable value for c_p, can be determined on a basic test case.

2.6 Domain truncation

2.6.1 Boundary conditions

Each computational setup needs to be closed by boundary conditions, which depends on the physics of the problem. Most of the time, photovoltaic problems are open. The considered physical domain must be artificially truncate due to the fact that computational domain cannot be infinite. In PV applications, some problems are peridiodic. This means they can be truncated using the so-called periodic boundary conditions (PBC). PBCs describe an infinite repetition of the same pattern. Periodicity can also be represented through symmetric domain, using perfect electric conductor condition (PEC) and perfect magnetic conductor condition (PMC).

In DGTD methods, the boundary conditions are imposed by adding a layer of cells called 'ghost cells'. This layer is outside the computational domain (see figure 2.13). Then, specific values of the fields are set in these ghost cells. Thus, the behavior of the solution on the boundary is controlled, without any special treatment, via numerical fluxes. Hereafter, the field inside the ghost cells are noted $\mathbf{E}_{gc}$ and $\mathbf{H}_{gc}$, while the fields in the boundary cells are denoted $\mathbf{E}_{bc}$ and $\mathbf{H}_{bc}$.

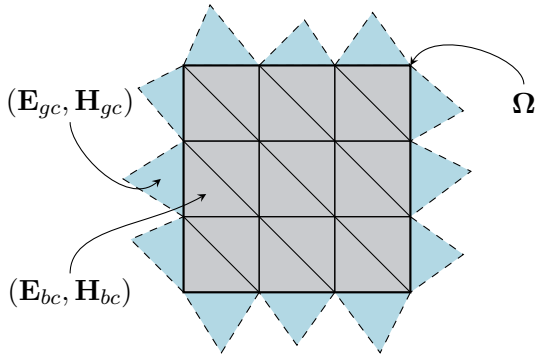


Figure 2.13 | Ghost cells layer on the computational domain boundary. Boundary conditions are naturally handle via numerical fluxes and ghost cells.

Perfect electric conductor condition Perfect electric conductor condition (PEC) is an idealized material which exhibits infinite electrical conductivity. A zero tangential electric field is enforced by setting the field values in the ghost cells on a PEC boundary as:

$$\mathbf{E}_{gc} = -\mathbf{E}_{bc} \text{ and } \mathbf{H}_{gc} = \mathbf{H}_{bc}.$$

Perfect magnetic conductor condition Perfect magnetic conductor condition (PMC) reprensents the reciprocal of the PEC condition. Most of the time, it is used to impose symmetry planes.

A zero tangential magnetic field is enforced by setting the field values in the ghost cells on a PMC boundary as:

$$\mathbf{E}_{gc} = \mathbf{E}_{bc} \text{ and } \mathbf{H}_{gc} = -\mathbf{H}_{bc}.$$

Absorbing boundary condition Absorbing boundary conditions (ABC) allow to partially absorb fields radiating out from the physical domain. It exists several forms of these conditions. The first-order Silver-Müller boundary condition is often used [Mon03]:

Silver-Müller boundary condition

$$\begin{aligned} \mathbf{n} \times (\mathbf{E} + Z\,(\mathbf{n} \times \mathbf{H})) &= \mathbf{0} \\ \mathbf{n} \times (\mathbf{H} - Y\,(\mathbf{n} \times \mathbf{E})) &= \mathbf{0} \end{aligned} \tag{2.37}$$

Imposing the Silver-Müller boundary condition is similar to set the incoming flux to zero on the boundary. For this reason, its expression depends on the upwinding factor α:

$$Y_{gc} = Y_{bc}, \quad Z_{gc} = Z_{bc}, \quad \mathbf{E}_{gc} = \mathbf{0} + \left(\frac{1-\alpha}{Y_{bc}}\right) \mathbf{n} \times \mathbf{H}_{bc} \text{ and } \mathbf{H}_{gc} = \mathbf{0} - \left(\frac{1-\alpha}{Z_{bc}}\right) \mathbf{n} \times \mathbf{E}_{bc}.$$

Remark *: The Silver-Müller condition perfectly absorbs normally-incident plane waves, when imposed on the boundary. However, when waves are incident at increasing angles, its performance decreases.*

Periodic boundary condition Periodic boundary conditions (PBC) can be used in order to simulate infinite mono- or bi-directional arrays, considering one elementary pattern. Cells from a periodic boundary face are matched with cells on the opposite boundary face of the domain. This way, every cell has a neighbor which is well-defined, and standard fluxes can be applied.

Remark *: A periodic mesh is a prerequisite to use PBC. In other words, opposite faces in the periodic direction must match.*

2.6.2 Perfectly matched layers

A novel numerical concept has been developed in 1994 by Bérenger in order to overcome the limitations of ABC. The objectives is to absorb the waves radiated from a system. To do so, Bérenger defined an artificial volume surrounding the physical domain in which a damping should occur progressively. Those artificial volumes are known as perfectly matched layers (PML). Figure 2.14 illustrates the functioning of PMLs. Outgoing wave propagates in the physical domain toward the PML, and crosses the interface. No reflection occurs, and the wave is progressively damped by the artificial medium while it continues to propagate in the PML. At one point, the wave will encounters the boundary of the computational domain, which is usually PEC or ABC. In both cases, the remaining of the wave will be totally (PEC) or partially (ABC) reflected toward the domain. In practice, the wave will travel a second time over the PML length, causing more damping. In general, when it enters again in the physical domain, the amplitude of the wave is attenuated by several orders of magnitude. For this reason, the error induced by PMLs is supposed

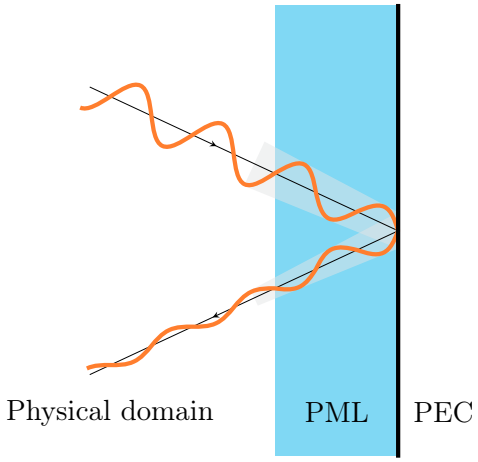


Figure 2.14 | General functioning of the PML. Outgoing wave propagates in the physical domain toward the PML, and crosses the interface. No reflection occurs, and the wave is progressively damped by the artificial medium and continues to propagate in the PML. At one point, the wave might encounters the edge of the computational domain, which is usually PEC or ABC. In both cases, the remaining of the wave will be totally (PEC) or partially (ABC) reflected toward the domain. In practice, the wave will travel a second time over the PML length, causing more damping. In general, when it enters again in the physical domain, the amplitude of the wave is attenuated by several orders of magnitude.

to be small enough not to lose the benefits of high-order methods. ABCs can be seen as "geometric" condition (*i.e.* it only becomes more efficient with a larger distance from the source), while PMLs take advantage of the high-order spatial discretization to allow higher levels of damping.

PMLs have evolved since Bérenger's implementation, and several varieties are now available [Bér07]. One can cite the uniaxial PML (UPML) [Viq16], and the complex frequency-shifted PML (CFS-PML), which are in use for Maxwell's equations. The last one is used in this work.

CFS-PML The complex frequency-shifted PML (CFS-PML) are based on a complex stretch of coordinates in the spatial operator. This approach was introduced by Kuzuoglu and Mittra [KM96]. However further developments showed that this method was found successful for wrong reasons [BPG02]. That being said, since [RG00] the method has been broadly used in the FDTD community. In 2011, a DGTD implementation of this PML has been proposed by König [Kön11]. The next paragraph is based on this implementation. The following change of variables imposes the complex stretching:

$$\frac{\partial}{\partial x} \to \frac{1}{s_x(\omega)}\frac{\partial}{\partial x}, \quad \frac{\partial}{\partial y} \to \frac{1}{s_y(\omega)}\frac{\partial}{\partial y}, \quad \frac{\partial}{\partial z} \to \frac{1}{s_z(\omega)}\frac{\partial}{\partial z},$$

with:

$$s_k(\omega) = \kappa_k - \frac{\sigma_k}{i\omega - \alpha_k}, \quad \text{with} \quad k \in \{x, y, z\},$$

where σ_k is the loss rate of the PML. Two new parameters are introduced: κ_k that represents a real stretching factor, which effect is only to artificially lengthen the PML; α_k which is the actual frequency shift, since it moves the pole of $s_k(\omega)$ from $\omega_p = 0$ to $\omega_p = i\omega$. Linearly-growing instabilities in long-time computations can arise, when $\alpha_k = 0$ [BPG02].

2.7 Sources

One of the major concern to simulate realistic configuration is a good control of the properties of incident fields. Indeed, the physical response of a nanophotonic system depends mainly on how it is excited. In this work, only plane waves are considered since they can model the solar light, even if several sources are available from realistic physical applications as laser beams [NH07], or dipoles.

2.7.1 Plane waves

The most simple kind of source are plane waves. Indeed, they are commonly used in numerical electromagnetics to determine the fundamental properties of a physical system even if these waves correspond to an asymptotic physical configuration (*i.e.* any radiating source propagating on a sufficiently large distance should look like a plane wave). Resonances and modes of a physical system can be excited depending on the spectral profile of the source. Monochromatic plane wave is the most basic kind of time dependence. In the temporal domain, its expression is:

$$\mathbf{E}(t) = \mathbf{E}_0 \sin(\omega_0 t). \tag{2.38}$$

One can notice that the Fourier transform of (2.38) is proportional to the Dirac function δ_{ω_0}. This type of source can be useful in several cases. However, running a whole time-domain simulation only to obtain the response of the system at one frequency may not be worth it, and a frequency-domain methods may be more suited for this type of source. In order to obtain a broader frequency spectrum, one can use:

$$\mathbf{E}(t) = \mathbf{E}_0 \sin(\omega_0 (t - t_0))\, \mathrm{e}^{-\frac{(t-t_0)^2}{2\sigma^2}}, \tag{2.39}$$

which is a Gaussian function of width σ centered around t_0, and modulated by a sine function. Its Fourier transform can be computed as follow:

$$\widehat{\mathbf{E}}(\omega) = \mathbf{E}_0 i\sigma \sqrt{\frac{\pi}{2}}\, \mathrm{e}^{i\omega t_0} \left(\mathrm{e}^{-\frac{\sigma^2(\omega-\omega_0)^2}{2}} - \mathrm{e}^{-\frac{\sigma^2(\omega+\omega_0)^2}{2}} \right), \tag{2.40}$$

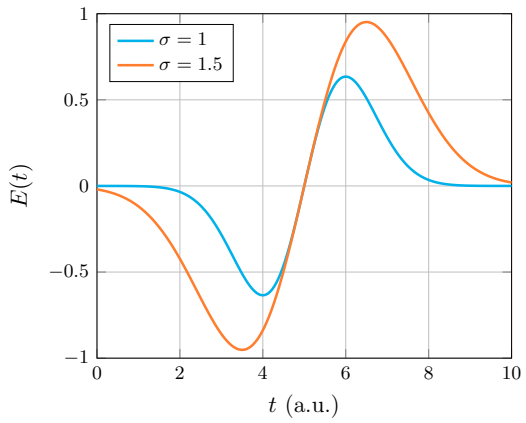
This means that such a pulse traveling through a structure will excite it on a wideband of wavelength, and not just on a single one. In our case, the idea is to use a pulse which contains the whole visible spectrum.

2.7.2 TF/SF formulation

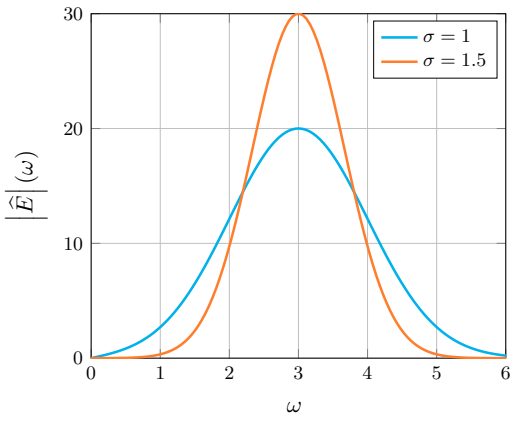
Different possibilities are available in order to impose the plane waves inside the physical domain. A first simple one is to use the ghost cells on the ABCs. This solution is no longer viable when PMLs are used (imposing fields in the PMLs will directly damp them). A possible alternative is to define an additional artificial contour inside the physical domain and between PMLs and the scatterer, on which the field could be imposed directly. For periodic domains, the artificial contour can be replaced with infinite artificial surfaces.

First we consider the splitting of the electric field in two parts:

$$\mathbf{E}_{\mathrm{tot}}(\mathbf{x}, t) = \mathbf{E}_{\mathrm{inc}}(\mathbf{x}, t) + \mathbf{E}_{\mathrm{sca}}(\mathbf{x}, t), \tag{2.41}$$



(a) Time-domain



(b) Frequency-domain

Figure 2.15 | Wideband pulse representation in time-domain and frequency-domain. Parameters are $E_0 = 20$ V.m^{-1}, $\omega_0 = 3$, $t_0 = 5$. The shorter the time-domain pulse, the wider the frequency-domain spectrum.

where $\mathbf{E}_{\text{tot}}$ is the total field, $\mathbf{E}_{\text{inc}}$ is the incident field, and $\mathbf{E}_{\text{sca}}$ is the scattered field. In our case, the incident field is a plane wave and it is known since it is imposed analytically. We consider now a splitting of the computational domain in two regions: the first one which in which the total field is computed, while in the second region the scattered field is computed. We call the interface between these two regions the total field/scattered field (TF/SF) interface. The DGTD formulation 2.21 is still valid in both region, and no modification is required, except for the computation of the fluxes between both regions. Consider a TF/SF interface between two cells, such that the local cell T_i is in the total field region and its neighbor cell T_l is the scattered field region. The upwind flux calculated for cell T_i is:

$$\mathbf{E}_{*,\text{tot}} = \frac{1}{Y_i + Y_l} \left(\{ Y \mathbf{E}_{\text{tot}} \}_{il} + \alpha \mathbf{n} \times [\![\mathbf{H}_{\text{tot}}]\!]_{il} \right),$$

with $\{ Y \mathbf{E}_{\text{tot}} \}_{il} = Y_i \mathbf{E}_{i,\text{tot}} + Y_l \mathbf{E}_{l,\text{tot}}$ and $[\![\mathbf{H}_{\text{tot}}]\!]_{il} = \mathbf{H}_{l,\text{tot}} - \mathbf{H}_{i,\text{tot}}$. However, the field values corresponding to cell T_l are not $\mathbf{E}_{l,\text{tot}}$ and $\mathbf{H}_{l,\text{tot}}$, but $\mathbf{E}_{l,\text{sca}}$ and $\mathbf{H}_{l,\text{sca}}$. Hence, the flux formulation must be modified to account for this difference. By considering (2.41), one easily shows that the right flux can be calculated as follows:

$$\mathbf{E}_{*,\text{tot}} = \underbrace{\frac{1}{Y_i + Y_l} \left(\{ Y \mathbf{E} \}_{il} + \alpha \mathbf{n} \times [\![\mathbf{H}]\!]_{il} \right)}_{\mathbf{E}_*} + \underbrace{\frac{1}{Y_i + Y_l} \left(Y_l \mathbf{E}_{\text{inc}} + \alpha \mathbf{n} \times \mathbf{H}_{\text{inc}} \right)}_{\mathbf{E}_{*,\text{inc}}}, \tag{2.42}$$

Symmetrically, the upwind flux for cell T_l in the scattered field region is:

$$\mathbf{E}_{*,\text{sca}} = \mathbf{E}_* - \mathbf{E}_{*,\text{inc}}. \tag{2.43}$$

Remark *: The fluxes computation at the TF/SF interface can be handled by a separate additional loop on the TF/SF faces during the fields update. causing a minimal overhead.*

The TF/SF decomposition are interesting for several reasons. In the presence of PMLs, they can be used as an "interior boundary" to impose incident fields. It is possible to observe both the total field in near of the scatterer and the scattered field far from it in a single computation, and

it helps to compute relevant scattering quantities, such as reflection, transmission and absorption (see chapter 4).

3

Non-conformal hybrid and locally adaptive DGTD method

Contents

In the previous chapter, we have presented a DG formulation using a uniform polynomial order across the whole computational domain, coupled with an explicit time-stepping scheme. However, when dealing with realistic and complex geometries, the resulting mesh can show significant variations in cell size. In such cases, the time-step is imposed by the smallest cells and depends on the polynomial order chosen, which can lead to prohibitive computational costs for high polynomial orders. Indeed, in such a scenario, high-order interpolation is only needed in the coarser cells,

while both the computational overhead and the smaller time-steps in the finer cells have negative impacts.

In order to overcome this limitation, several possibilities can be considered. For example, one can replace the explicit time scheme by an implicit time-stepping algorithm. This results in the resolution of a large linear system at each (larger) time-step [VB09]. Some hybrid implementations, called implicit-explicit (IMEX), were also proposed. For instance, in [Ver10] and [Moy13], the authors consider a hybrid formulation where the smallest cells are treated with an implicit scheme while using an explicit time integration for the bigger cells, thus limiting the time-step constraint. However, maintaining high-order time integration has proven to be difficult [Moy13]. Another possibility is to exploit local time-stepping (LTS). One of the main ideas is to divide the mesh into regions, which share the same time-step (see [Pip06] and [DG09]). Moreover, high-order convergence is preserved by this method as shown in [DG09].

A complementary strategy, which is naturally suitable to DG methods, relies on the use of local polynomial orders. The above-mentioned drawbacks can be managed by imposing low orders in the smaller cells, and high-orders in the coarser cells. Doing so significantly reduces both the global number of degrees of freedom and the time-step restriction with a minimal impact on the accuracy of the method. Strategies exploiting locally adaptive (LA) formulations usually combine both h- and p-adaptivity in order to concentrate the computational effort in the areas of high field variations. In this chapter, we want to show that, starting from a given mesh with a homogeneous $\mathbb{P}_k$ polynomial order, it is possible to exploit the LA strategy with all the polynomial orders $\mathbb{P}_p$ ($p \leq k$) and obtain a solution of similar accuracy for a reduced computational cost.

PV cells are made of an assembly of layers of materials. The difficulty in discretizing this type of domain lies in the interface between these layers, where highly detailed nanotextures are present. This area, which represents around 20 to 30% of the computational domain, is the main argument in favor of the use of an unstructured mesh. A strategy has already been proposed to drastically reduce the number of degrees of freedom as well as the computing time, notably for electromagnetism in [Dur13], and more precisely for nanophotonics in [LVD+14]. This strategy consists of using a structured hexahedral mesh in the homogeneous parts of the domain while using an unstructured mesh for the proper representation of geometry in the areas of importance.

In this chapter, we present two numerical strategies to deal with multiscale PV structures. First, a p-adaptive version of the DGTD method is studied in section 3.1. This is an extension of the one presented in [VL16]. Several strategies for the distribution of the interpolation orders across the mesh cells are also proposed. Then, to further extend this method, conforming tetrahedral meshes are replaced with non-conforming hybrid meshes. A modified DGTD formulation on hybrid tetrahedral/hexahedral meshes with p-adaptivity is presented in section 3.2. Then, we present the main details about the generation of hybrid meshes, especially for PV structures. Finally, in section 3.3, we show the impact of the mesh hybridization with p-adaptivity on the convergence of the scheme and on the computational cost of the simulations.

3.1 Locally adaptive DGTD method

In this section, the DG formulation presented in section 2.4 is modified to account for variable polynomial orders. Two distribution strategies are presented and compared. Resulting algorithms are tested for sequential executions, and the CPU and accuracy gains are evaluated on a realistic configuration relevant to photovoltaics.

3.1.1 Strategy based on the computed time-step per cell

We recall here the strategy initially introduced in [VL16], which is our starting and reference point. Starting with a given mesh, it seems evident that the final repartition of interpolation orders across the different cells will have a significant impact on the obtained accuracy, as well as on the CPU time required to obtain the numerical solution. Suppose that the solution is obtained on the given mesh with a homogeneous polynomial order $\mathbb{P}_k$. The point is here to see how, with a proper distribution of polynomial orders $\mathbb{P}_l$ with $l \leq k$, a solution with at least, a similar accuracy can be obtained at a lower computational effort. We start by defining the mesh size heterogeneity quantity as follow:

$$\overline{h} = \frac{h_{\max}}{h_{\min}}.$$

We also need to define the time-step in each cell T_i as Δt_i , which is computed following the formula given in section 2.5.3, *i.e.*

$$\Delta t_i = \mathrm{CFL}(p) h_i.$$

while Δt_i^p represents the effective time-step obtained if cell T_i is discretized with a $\mathbb{P}_p$ polynomial expansion:

$$\Delta t_i^p = \overline{\mathrm{CFL}}\,(p) \Delta t_i,$$

with the second CFL factor $\overline{\mathrm{CFL}}$ equal to:

$$\overline{\mathrm{CFL}}\,(p) = \frac{\mathrm{CFL}(p)}{\mathrm{CFL}(k)}$$

with k the initial interpolation order.

We also define the normalized time-step, which includes the computational load induced by the polynomial order, roughly estimated by the number of degrees of freedom (d.o.f.):

$$\overline{\Delta t}_i^p = \frac{\Delta t_i^p}{n(p)} = \frac{\overline{CFL}\,(p) \Delta t_i}{n(p)},$$

where $n(p)$ is the number of d.o.f. in a $\mathbb{P}_p$ cell. Finally, we define $p_{\min}$ and $p_{\max}$ the minimal and maximal user-authorized orders. In comparison with what was done in [VL16], we do not restrict the order jump between two adjacent cells (in [VL16], a $\mathbb{P}_k - \mathbb{P}_l$ interface needs to fulfill the condition $|k - l| = 1$), even if it means that we should store more matrices in memory.

The first step of the repartition procedure consists of computing all the local Δt_i, and sorting them by ascending order. The cell with a lower time-step receives the minimal order $p_{\min}$, and we compute its normalized time-step $\overline{\Delta t}_1^{p_{\min}}$. We define two temporary variables, p_{loc} and $\overline{\Delta t}_{\mathrm{loc}}$, which are respectively the current order assigned to elements and the current normalized time-step. For a given cell, the normalized time step for increased order $p_{\mathrm{loc}} + 1$ is compared to the current limiting normalized time-step $\overline{\Delta t}_{\mathrm{loc}}$. In the case where the first is higher than the second, switching to higher order is assumed to have a limited impact on the final time-step. Hence, p_{loc} is increased by one, and $\overline{\Delta t}_{\mathrm{loc}}$ is updated. The procedure is summarized in the algorithm 1.

Algorithm 1 Polynomial order distribution with the *time-step strategy*

for $i \leftarrow 1, n_t$ **do** ▷ Compute time-step for each cell
  Compute Δt_i
end for

Sort cells by ascending Δt_i
$p(1) \leftarrow p_{\min}$
$\overline{\Delta t}_{\text{loc}} \leftarrow \overline{\Delta t}_1^{p_{\min}}$
$p_{\text{loc}} \leftarrow p_{\min}$

for $i \leftarrow 2, n_t$ **do** ▷ Go over cells by ascending order of Δt_i
  if $p_{\text{loc}} + 1 > p_{\max}$ **then** ▷ Check that we do not exceed $p_{\max}$
    $p(i) \leftarrow p_{\max}$
  else
    Compute $\overline{\Delta t}_i^{p_{\text{loc}}+1}$
    if $\overline{\Delta t}_i^{p_{\text{loc}}+1} > \overline{\Delta t}_{\text{loc}}$ **then** ▷ Check if it is worth changing order
      $\overline{\Delta t}_{\text{loc}} \leftarrow \overline{\Delta t}_i^{p_{\text{loc}}+1}$ ▷ Update the limiting time-step
      $p(i) \leftarrow p_{\text{loc}} + 1$
      $p_{\text{loc}} \leftarrow p_{\text{loc}} + 1$ ▷ Update the current order
    else
      $p(i) \leftarrow p_{\text{loc}}$
    end if
  end if
end for

3.1.2 Strategy based on the number of points per wavelength

It is well known that a proper numerical approximation of waves by a numerical method is related to the number of points per wavelength. It is generally accepted that 6 to 10 points per wavelength are usually sufficient to reach an acceptable accuracy.

In the context of this thesis, the main application is PV and, in general, simulations consist of the illumination of a device by a polychromatic plane wave over the whole visible spectrum [300 nm, 1000 nm]. In addition to this broad spectrum, the heterogeneity of solar cells, coupled with the complexity of the material used plays an essential role in the light propagation. Of course, when light travels from one medium to another, its speed changes, and so does its wavelength. In order to obtain accurate results and capture all the resonances, some physical conditions have to be fulfilled.

We start by defining in each cell T_i, the metric $d^{\text{dof}}(i)$, which represents the maximum distance between two consecutive degrees of freedom. Then for the particular material in T_i, the minimum effective wavelength $\lambda(i)$ is computed. This is done by computing the wave's speed inside the material for all the wavelengths in the spectrum of interest. From there, it is possible to compute the minimum number of points per wavelength inside the tetrahedron T_i as:

$$n_{\text{ppw}}(i) = \frac{\lambda(i)}{d^{\text{dof}}(i)}. \tag{3.1}$$

Then, given $p_{\min}$ and $p_{\max}$ the minimal and maximal user-authorized orders, and $\overline{n}_{\text{ppw}}$ the minimum number of points per wavelength required, we are able to distribute the polynomial orders accross $\mathcal{T}_h$. We start by setting $p_{\min}$ in all cell T_i. Then we compute $d^{\text{dof}}(i)$ and $n_{\text{ppw}}(i)$. If $n_{\text{ppw}}(i) \leq \overline{n}_{\text{ppw}}(i)$ and $p(i) < p_{\max}$, we increase p_i. The procedure is summarized in algorithm 2.

Algorithm 2 Polynomial order distribution with the *Points per wavelength strategy*

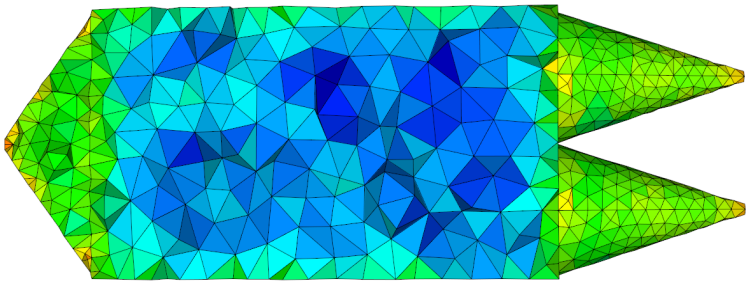
```
p(:) ← p_min                                   ▷ Each cell receives the minimal order
for i ← 1, n_t do                              ▷ Compute maximal distance between two d.o.f. for each cell
    Compute d^dof(i)
    Compute n_ppw(i)
    if n_ppw(i) > n̄_ppw then
        Exit                                   ▷ One can take a coarser mesh in this area
    end if
    if n_ppw(i) < n̄_ppw then
        if p(i) + 1 ≤ p_max then
            p(i) ← p(i) + 1
            Cycle
        else
            Exit                               ▷ One should refine the mesh here or increase p_max
        end if
    end if
end for
```

In the next part, we show on one example, the behavior of each strategy, and in particular that the second one performs better in the context of numerical simulation of PV cells.

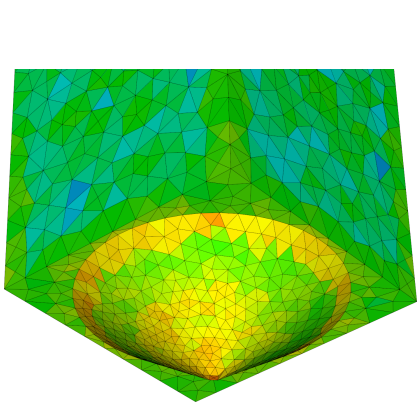
3.1.3 Performance assessment

The first algorithm has been assessed on several nanophotonic cases in [VL16]. In this work, we want to highlight the main differences between the two strategies, and especially the configurations in which the second one is more suitable. As a reminder, it is shown in [VL16] that for a similar accuracy, a computational speedup of 6.15 for a locally refined cubic cavity, 2.60 for a metallic nanolens, and 2.19 for a bowtie nanoantenna is achieved when using local order polynomials. The second strategy is more based on physics than on numerics. Of course, in the end, the objective is still the same, *i.e.* making the numerical simulation cheaper, both in terms of memory and of computational time. However, this should be done such that the obtained results are accurate enough. For example, we consider a nanostructured PV cell composed of two gratings [WYL$^+$12]. This structure is studied in details in section 4.5.1 of chapter 4. The structured layer is shown in figure 3.1 with colors representing a metric based on the volume of each cell (blue cells are the largest ones; red cells are the smallest ones).

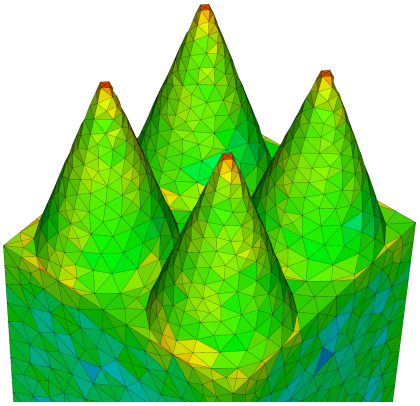
As one can see, some areas are critical, mostly at the top and bottom of the cones. The repartition of the 34544 cells is shown in figure 3.2(b), for the strategy based on the time-step and for the strategy based on the number of points per wavelength with $\overline{n}_{\text{ppw}} = 6$. We see that the first strategy in 3.2(a) is mostly leading to a $\mathbb{P}_2$ interpolation everywhere, with a few elements using $\mathbb{P}_1$ and $\mathbb{P}_3$ interpolations. Regarding the second strategy, the repartition is shown in 3.2(b) for $\overline{n}_{\text{ppw}} = 6$. We note that the repartition is not exactly as shown, *i.e.* proportions are correct, but elements with $\mathbb{P}_1$ interpolation are not only the smallest ones. Of course, the interpolation order



(a) Structured layer



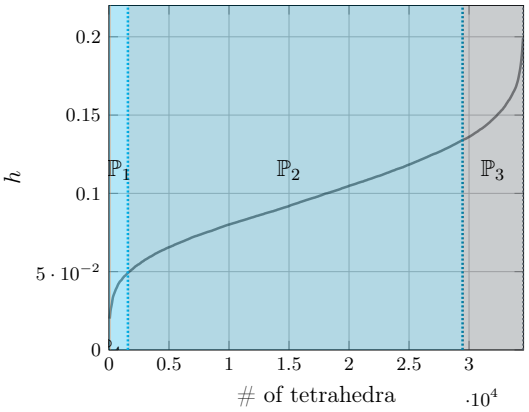
(b) Bottom cone



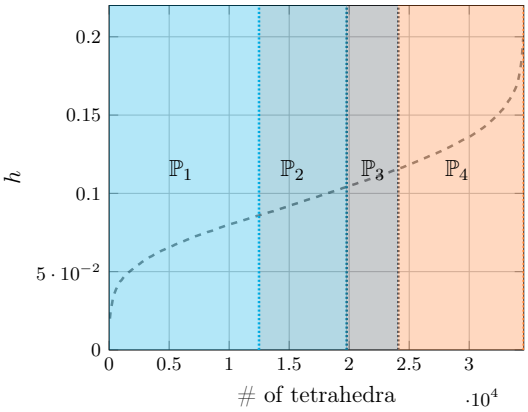
(c) Top cone

Figure 3.1 | Nanostructured mesh of the double grating solar cell: the color corresponds to the cell size.

assigned to a cell depends on the size of the cell, but also on the material and on the considered physical problem. We observe that most of the elements are rather using $\mathbb{P}_1$ and $\mathbb{P}_4$ interpolations.



(a) Time-step strategy

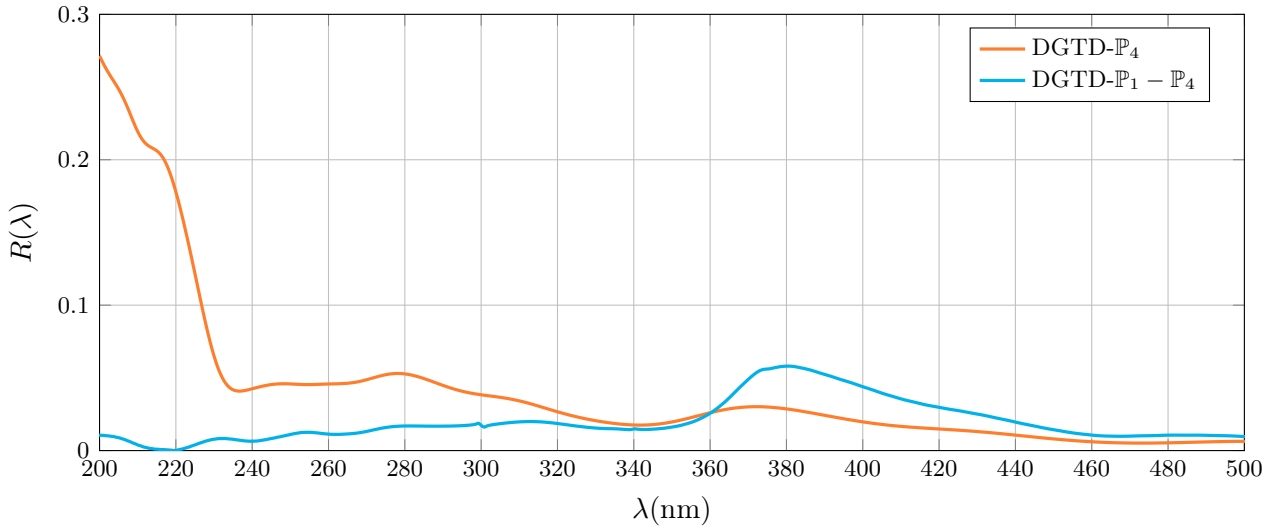


(b) Pts/wavelength strategy ($\overline{n}_{\text{ppw}} = 6$)

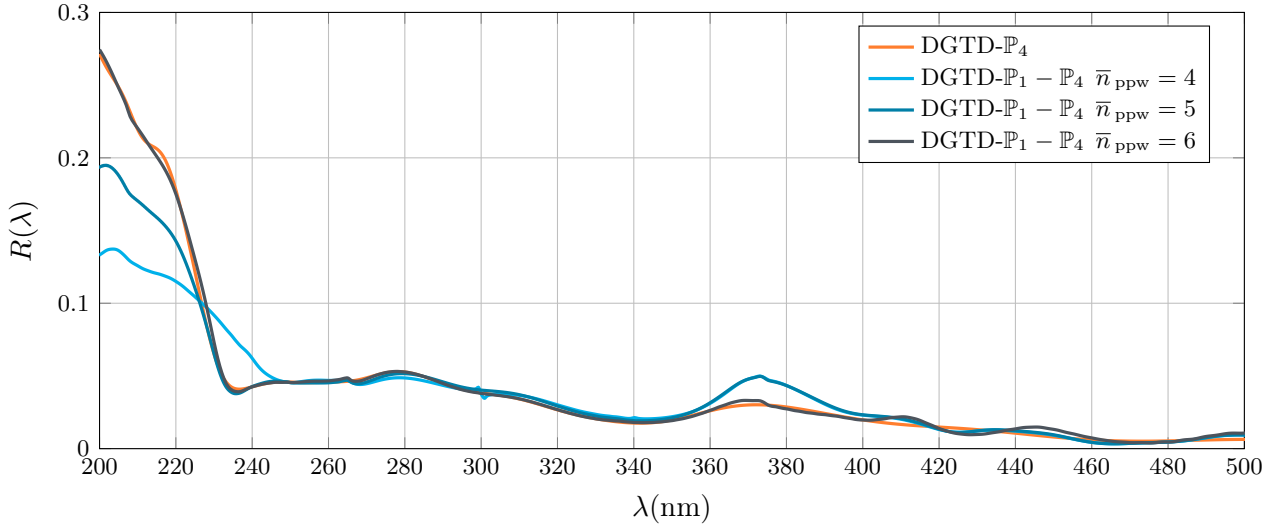
Figure 3.2 | Comparison between the two locally adaptative strategies for $\mathbb{P}_1 - \mathbb{P}_4$ case. The repartition in the right figure shows the amount of cell per order, which is not directly related to $h_{\min}$.

In figure 3.3, we compare both strategies on the computation of one important quantity of interest in the PV field: the reflection (noted R and defined in chapter 4). Both strategies are

plotted against a converged DGTD-$\mathbb{P}_4$ simulation. In figure 3.3(a), we observe a notable difference between DGTD-$\mathbb{P}_4$ and DGTD-$\mathbb{P}_1 - \mathbb{P}_4$ (actually DGTD-$\mathbb{P}_1 - \mathbb{P}_3$), especially on the first part of the spectrum [200 nm, 320 nm], which corresponds to the shortest wavelength. The results of the second strategy are shown in figure 3.3(b). Three different values of $\overline{n}_{\text{ppw}}$ are tested: $\overline{n}_{\text{ppw}} = \{4, 5, 6\}$. We observe a better behavior than the first strategy, even for 4 points per wavelength.



(a) Time-step strategy



(b) Pts/wavelength strategy

Figure 3.3 | Reflection coefficient for DGTD-$\mathbb{P}_4$ (reference) and $\mathbb{P}_1 - \mathbb{P}4$ strategies.

Finally, computation times are given in table 3.1. As expected, the time-step based strategy outperforms the other ones in terms of computational time, with a speedup of 4. However, the solution is not good. For the second strategy, the speedup is directly related to the number of points per wavelength required. We obtained a speedup which goes from 3 for $\overline{n}_{\text{ppw}} = 4$ to 1.95 for $\overline{n}_{\text{ppw}} = 6$.

	Computation time	Speedup
$\mathbb{P}_4$	26 h 00 mn	-
$\mathbb{P}_1 - \mathbb{P}_4$, Δt	6 h 24 mn	4.05
$\mathbb{P}_1 - \mathbb{P}_4$, $\overline{n}_{\text{ppw}} = 4$	8 h 36 mn	3.00
$\mathbb{P}_1 - \mathbb{P}_4$, $\overline{n}_{\text{ppw}} = 5$	10 h 48 mn	2.40
$\mathbb{P}_1 - \mathbb{P}_4$, $\overline{n}_{\text{ppw}} = 6$	13 h 12 mn	1.95

Table 3.1 | Computation time for each simulation set-up on a Intel(R) Xeon(R) Gold 6154 CPU 3.00GHz system.

To conclude this section, the strategy to be adopted depends mostly on the accuracy expected. When speedup is more important than accuracy, the first strategy and the second strategy with a small number of points per wavelength are preferred. When accuracy is still the main expectation, the second method with $\overline{n}_{\text{ppw}} \geq 5$ should be preferred.

3.2 DGTD on hybrid meshes

As seen in the previous section, in order to be able to reach a certain accuracy, one should refine the mesh or increase the polynomial interpolation order in suitable regions of the computational domain. Each of these possibilities drastically increases the computing resources required for the simulation. However, it is common knowledge that hexahedral meshes are economical in terms of space (one hexahedron roughly replaces six tetrahedra) and advantageous in terms of accuracy. Here, we adapt the strategy presented in [Dur13], which consists of combining a structured hexahedral mesh for the light propagation in untextured and homogeneous regions (here thick layers) with a fully unstructured tetrahedral mesh for the propagation in the nanotextured interfaces. Such meshes can be globally conforming or non-conforming and are naturally handled by a DG formulation.

3.2.1 Hybrid configuration

For continuous Finite Element Method (FEM), it is mandatory to use conforming meshes because of the continuity of the functions on the interfaces. Conforming hybrid meshes in 3D can be used by considering so-called transition elements like prisms or pyramids, which is not obvious from a meshing point of view. However DG methods, thanks to the absence of continuity of basis functions at the cells interfaces, can naturally handle hybrid and non-conforming meshes without a full modification of the core of the formulation presented in chapter 2.

For the sake of simplicity, here, the studied configurations are limited to a particular type of non-conformity. The use of non-conformal and hybrid faces is restrained to the case where neighboring hybrid faces match precisely, as shown in figure 3.4. In addition, the polynomial interpolation order in the parent cell have to be higher than the polynomial interpolation order of the small cells (see figure 3.5).

3.2.2 DGTD on hybrid meshes

To ease the reading, we consider the non-dispersive time-domain Maxwell equations in this section. However, the extension of the formulation is presented in equation (3.8) to the case of Maxwell's equations for dispersive media.

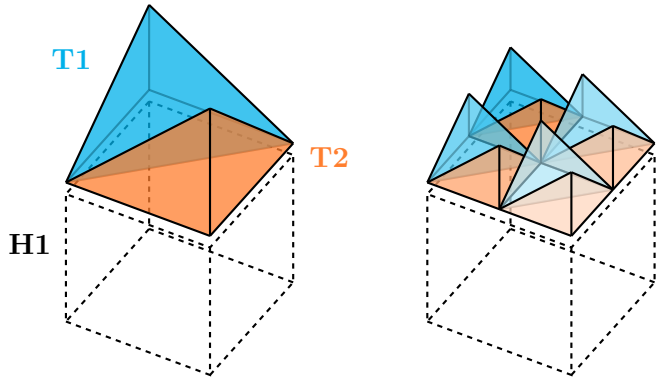


Figure 3.4 | Non-conformity considered. The union of hybrid faces from the children cells needs to match perfectly the face of the parent neighbor.

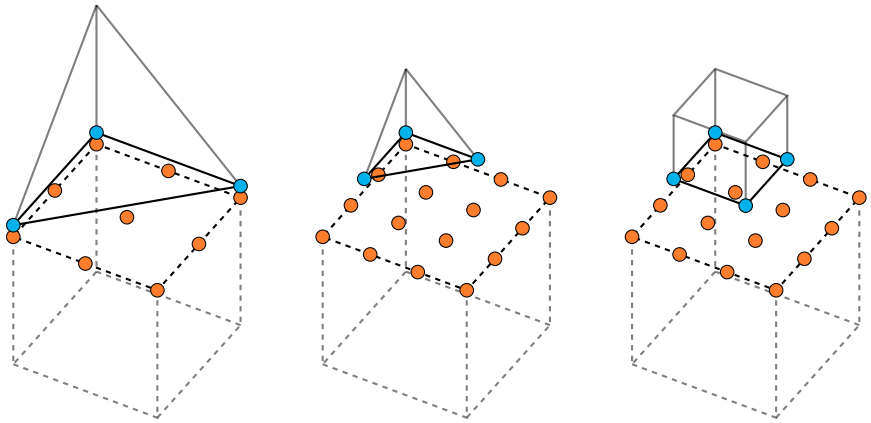


Figure 3.5 | Authorized hybrid interfaces. For the sake of precision, orders have to be higher on the parent element.

Let $\Omega \subset \mathbb{R}^3$ be a bounded convex domain, and $\mathbf{n}$ the unitary outward normal to its boundary $\partial\Omega$. The continuous, normalized Maxwell problem is: find $(\mathbf{E}, \mathbf{H})$ in $(L^2(\Omega))^3 \times (L^2(\Omega))^3)$

$$\begin{aligned} \mu_r \frac{\partial \mathbf{H}}{\partial t} &= -\nabla \times \mathbf{E}, \\ \varepsilon_r \frac{\partial \mathbf{E}}{\partial t} &= \quad \nabla \times \mathbf{H} - \mathbf{J}, \end{aligned}$$

We consider Ω_h as a discretization of Ω, relying on two quasi-uniform tesselations $\mathcal{T}_h$ and $\mathcal{H}_h$ verifying $\mathcal{T}_h = \bigcup_{i=1}^{N_\mathcal{T}} T_i$ and $\mathcal{H}_h = \bigcup_{i=1}^{N_\mathcal{H}} H_i$, where $N_\xi \in \mathbb{N}^*$ is the number of mesh elements in each part with $\xi \in \{\mathcal{H}, \mathcal{T}\}$, and $(T_i)_{i\in[\![1,N_\mathcal{T}]\!]}$ and $(H_i)_{i\in[\![1,N_\mathcal{H}]\!]}$ the set of simplices. For $i \in [\![1, N_\xi]\!]$, let C_i be a tetrahedron or a hexahedron of the hybrid mesh. The internal faces of the discretization are denoted $a_{ik} = C_i \bigcap C_k$ if C_i and C_k are neighboring cells, and $\mathbf{n}_{ik}$ is defined as the unit normal vector to the face a_{ik}, oriented from C_i toward C_k. For each cell C_i, $\mathcal{V}_i$ is the set of indices $\{k \in [\![1, N]\!] \,|\, C_i \bigcap C_k \text{ is a face}\}$.

Let us define V_h as:

$$V_h = \left\{ v \in \left(L^2(\Omega)\right)^3, \middle| \begin{array}{ll} v_{|T_i} \in (\mathbb{P}_p(T_i))^3 & \forall\, T_i \in \mathcal{T}_h \\ v_{|H_i} \in (\mathbb{Q}_q(H_i))^3 & \forall\, H_i \in \mathcal{H}_h \end{array} \right\}$$

where $\mathbb{P}_p(T_i)$ is the space of polynomials of maximum degree p on T_i and $\mathbb{Q}_q(H_i)$ is the space of polynomials of maximum degree q on H_j. The semi-discrete fields, sought in space V_h, are hereafter denoted $(\mathbf{H}_h, \mathbf{E}_h, \mathbf{J}_h)$, and on each cell T_i the restrictions $(\mathbf{H}_i, \mathbf{E}_i, \mathbf{J}_i) = \left(\mathbf{H}_h|_{T_i}, \mathbf{E}_h|_{T_i}, \mathbf{J}_h|_{T_i}\right)$ are defined. A set of scalar basis functions $(\phi_{ik})_{1\leqslant k\leqslant d_i}$ is defined for each T_i, where $d_i^{\mathcal{T}}$ is the number of degrees of freedom (d.o.f.) per dimension (respectively $(\psi_{jk})_{1\leqslant k\leqslant d_i^{\mathcal{H}}}$ for each H_i). Additionally, to each scalar basis function, the three vectors $\boldsymbol{\phi}_{\boldsymbol{ik}}^{\boldsymbol{v}}$ are associated (respectively $\boldsymbol{\psi}_{\boldsymbol{ik}}^{\boldsymbol{v}}$):

$$\boldsymbol{\phi}_{\boldsymbol{ik}}^{\boldsymbol{x}} = \begin{bmatrix} \phi_{ik} \\ 0 \\ 0 \end{bmatrix}, \boldsymbol{\phi}_{\boldsymbol{ik}}^{\boldsymbol{y}} = \begin{bmatrix} 0 \\ \phi_{ik} \\ 0 \end{bmatrix}, \boldsymbol{\phi}_{\boldsymbol{ik}}^{\boldsymbol{z}} = \begin{bmatrix} 0 \\ 0 \\ \phi_{ik} \end{bmatrix}, \boldsymbol{\psi}_{\boldsymbol{ik}}^{\boldsymbol{x}} = \begin{bmatrix} \psi_{ik} \\ 0 \\ 0 \end{bmatrix}, \boldsymbol{\psi}_{\boldsymbol{ik}}^{\boldsymbol{y}} = \begin{bmatrix} 0 \\ \psi_{ik} \\ 0 \end{bmatrix}, \boldsymbol{\psi}_{\boldsymbol{ik}}^{\boldsymbol{z}} = \begin{bmatrix} 0 \\ 0 \\ \psi_{ik} \end{bmatrix},$$

One now seeks the approximations $\mathbf{E}_h$ of $\mathbf{E}$ in space V_h (respectively $\mathbf{H}_h$ of $\mathbf{H}$). The contribution of each cell is therefore defined as $\mathbf{E}_i = \mathbf{E}_h|_{T_i}$. Here, one must notice that, for a 3D system, $\mathbf{E}_i$ is actually a vector that has 3 components:

$$\mathbf{E}_i = \begin{bmatrix} E_i^x \\ E_i^y \\ E_i^z \end{bmatrix},$$

each of which is locally expanded on the chosen set of basis functions:

$$E_i^v = \sum_{j=1}^{d_i} E_{ij}^v \phi_{ij}, v \in \{x, y, z\}. \tag{3.2}$$

Therefore, for practical purpose, one defines three vectors of d_i components:

$$\overline{\mathbf{E}}_i^{\,v} = \begin{bmatrix} E_{i1}^v \\ \vdots \\ E_{id_i}^v \end{bmatrix}, v \in \{x, y, z\},$$

as well as the following $3d_i$ components vector:

$$\overline{\mathbf{E}}_{\,i} = \begin{bmatrix} \left(E_{ij}^x\right)_{1\leqslant j\leqslant d_i} \\ \left(E_{ij}^y\right)_{1\leqslant j\leqslant d_i} \\ \left(E_{ij}^z\right)_{1\leqslant j\leqslant d_i} \end{bmatrix}.$$

The same four steps are done for $\mathbf{H}$ and for an element H_i, even if it is not detailed here.

We consider the weak formulation of Maxwell's equations without current:

$$\begin{aligned} \int_{C_i} \mu_r \frac{\partial \mathbf{H}}{\partial t} \cdot \boldsymbol{\phi}_{\boldsymbol{ik}}^{\boldsymbol{v}} + \int_{C_i} \mathbf{E} \cdot \nabla \times \boldsymbol{\phi}_{\boldsymbol{ik}}^{\boldsymbol{v}} = & \sum_{l\in\mathcal{V}_i} \int_{a_{il}} (\mathbf{E}_* \times \mathbf{n}_{il}) \cdot \boldsymbol{\phi}_{\boldsymbol{ik}}^{\boldsymbol{v}}, \\ \int_{C_i} \varepsilon_r \frac{\partial \mathbf{E}}{\partial t} \cdot \boldsymbol{\phi}_{\boldsymbol{ik}}^{\boldsymbol{v}} - \int_{C_i} \mathbf{H} \cdot \nabla \times \boldsymbol{\phi}_{\boldsymbol{ik}}^{\boldsymbol{v}} = - & \sum_{l\in\mathcal{V}_i} \int_{a_{il}} (\mathbf{H}_* \times \mathbf{n}_{il}) \cdot \boldsymbol{\phi}_{\boldsymbol{ik}}^{\boldsymbol{v}}. \end{aligned} \tag{3.3}$$

We can develop the first volumic integral using the local field expansions (3.2):

$$\int_{C_i} \varepsilon_r \frac{\partial \mathbf{E}_i}{\partial t} \cdot \boldsymbol{\phi}_{\boldsymbol{ik}}^{\boldsymbol{x}} = \left(\mathbb{M}_i^{\varepsilon_r} \frac{\partial \overline{\mathbf{E}}_i^{\,x}}{\partial t}\right)_k$$

as well as the curl integral:

$$\int_{C_i} \mathbf{H}_i \cdot \nabla \times \boldsymbol{\phi}^{\boldsymbol{x}}_{\boldsymbol{ik}} = -\left(\overline{\mathbb{K}}_i \times \overline{\mathbf{H}}_i\right)^x_k,$$

without any changes with respect to the formulation presented in chapter 2.

Indeed, the main changes will appear in the evaluation of the flux integral on hybrid faces. For the sake of simplicity, we consider the centered flux. The generalization to the upwind flux is straightforward. Consider a neighbor cell H_l of T_i with a $\mathbb{Q}_q$ polynomial expansion. The x-component of the flux integral on their common face a_{il} from the T_i side is:

$$\begin{aligned}\int_{a_{il}} (\mathbf{H}_* \times \mathbf{n}_{il}) \cdot \boldsymbol{\phi}^{\boldsymbol{x}}_{\boldsymbol{ik}} &= \int_{a_{il}} \left(H^y_* n^z_{il} - H^z_* n^y_{il}\right) \phi_{ik} \\ &= \int_{a_{il}} \left(\frac{H^y_i + H^y_l}{2} n^z_{il} - \frac{H^z_i + H^z_l}{2} n^y_{il}\right) \phi_{ik}\end{aligned} \tag{3.4}$$

Consider the following expansions for H^y_i and H^y_l, and the analogous relations for H^z_i and H^z_l:

$$H^y_i = \sum_{q=1}^{p} H^y_{iq} \phi_{iq} \quad \text{and} \quad H^y_l = \sum_{r=1}^{q} H^y_{lr} \psi_{lr}. \tag{3.5}$$

Plugging (3.5) in (3.4) leads to:

$$\begin{aligned}\int_{a_{il}} (\mathbf{H}_* \times \mathbf{n}_{il}) \cdot \boldsymbol{\phi}^{\boldsymbol{x}}_{\boldsymbol{ik}} &= \int_{a_{il}} \frac{n^z_{il}}{2} \left(\sum_{q=1}^{p} H^y_{iq} \phi_{iq} \phi_{ik} + \sum_{r=1}^{q} H^y_{lr} \psi_{lr} \phi_{ik}\right) \\ &\quad - \int_{a_{il}} \frac{n^y_{il}}{2} \left(\sum_{q=1}^{p} H^z_{iq} \phi_{iq} \phi_{ik} + \sum_{r=1}^{q} H^z_{lr} \psi_{lr} \phi_{ik}\right), \\ &= \frac{1}{2} \sum_{q=1}^{p} \left(H^y_{iq} n^z_{il} - H^z_{iq} n^y_{il}\right) \int_{a_{il}} \phi_{iq} \phi_{ik} \\ &\quad + \frac{1}{2} \sum_{r=1}^{q} \left(H^y_{lr} n^z_{il} - H^z_{lr} n^y_{il}\right) \int_{a_{il}} \psi_{lr} \phi_{ik}, \\ &= \frac{1}{2} \sum_{q=1}^{p} \mathbf{H}_{iq} \times \mathbf{n}_{il} \int_{a_{il}} \phi_{iq} \phi_{ik} \\ &\quad + \frac{1}{2} \sum_{r=1}^{q} \mathbf{H}_{lr} \times \mathbf{n}_{il} \int_{a_{il}} \psi_{lr} \phi_{ik}, \\ &= \left(\mathbb{S}_{il} \left(\overline{\mathbf{H}}_{*,i} \times \mathbf{n}_{il}\right)\right)^x_k + \left(\mathbb{S}^*_{il} . \left(\overline{\mathbf{H}}_{*,l} \times \mathbf{n}_{il}\right)\right)^x_k,\end{aligned} \tag{3.6}$$

where, in accordance with the definitions of chapter 2, the flux matrices are:

$$\left(\mathbb{S}_{il}\right)_{jk} = \int_{a_{il}} \phi_{ij} \phi_{ik} \quad \text{and} \quad \left(\mathbb{S}^*_{il}\right)_{rk} = \int_{a_{il}} \psi_{lr} \phi_{ik}. \tag{3.7}$$

To summarize, the flux integrals are cut in two parts: (i) the part corresponding to local contribution, which is integrated *via* the "regular" flux matrix, and (ii) the part corresponding to

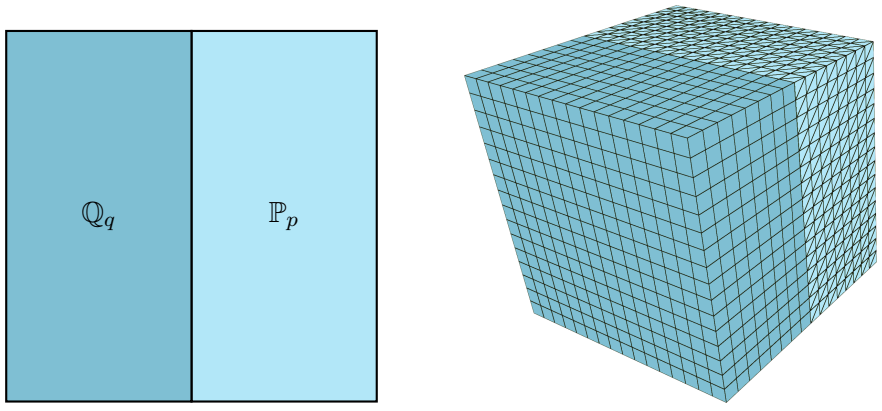
the neighbor contribution, which is integrated *via* non-conforming hybrid matrices. For a $\mathbb{P}_p - \mathbb{Q}_q$ interface, the flux matrix is rectangular, of size $s(p) \times s(q)$ (we recall that $s(p) = \frac{(p+1)(p+2)}{2}$ for a tetrahedron and $s(q) = (q+1)^2$ for a hexahedron). From this point, the rest of the derivation of the numerical form of the scheme is similar to the tetrahedral mesh case, and the reader is referred to section 2.4 for details. The final semi-discrete scheme is:

$$\begin{aligned}
\overline{\mathbb{M}}_i^{\mu_r} \frac{\partial \overline{\mathbf{H}}_i}{\partial t} &= -\overline{\mathbb{K}}_i \times \overline{\mathbf{E}}_i + \sum_{l \in \mathcal{V}_i} \overline{\mathbb{S}}_{il} \left(\overline{\mathbf{E}}_{*,i} \times \mathbf{n}_{il} \right) + \sum_{l \in \mathcal{V}_i} \overline{\mathbb{S}}_{il}^{*} \left(\overline{\mathbf{E}}_{*,l} \times \mathbf{n}_{il} \right), \\
\overline{\mathbb{M}}_i^{\varepsilon_r} \frac{\partial \overline{\mathbf{E}}_i}{\partial t} &= \quad \overline{\mathbb{K}}_i \times \overline{\mathbf{H}}_i - \sum_{l \in \mathcal{V}_i} \overline{\mathbb{S}}_{il} \left(\overline{\mathbf{H}}_{*,i} \times \mathbf{n}_{il} \right) - \sum_{l \in \mathcal{V}_i} \overline{\mathbb{S}}_{il}^{*} \left(\overline{\mathbf{H}}_{*,l} \times \mathbf{n}_{il} \right),
\end{aligned} \tag{3.8}$$

where the definition of $\overline{\mathbb{S}}_{il}^{*}$ is easily deduced from (3.7) and the analogous definition of $\overline{\mathbb{S}}_{il}$ given in section 2.4. Time integration still makes use of the schemes presented in section 2.5.

3.2.3 Numerical study of h-convergence

In this section, we study the numerical convergence of the DGTD method on hybrid meshes. For this purpose, we consider the cubic cavity case with PEC boundaries presented in chapter 2. Four hybrid meshes with increasing refinement are used. The cubic domain is split into two parts, as shown in figure 3.6, one for each element type. Upwind fluxes and the optimized 14 stages fourth-order low-storage Runge-Kutta time scheme developed in [NDB12] are used. The obtained results are given in table 3.2. As expected for a $\mathbb{P}_p - \mathbb{Q}_q$ configuration, the h-convergence rate (r) is limited by the lowest approximation order which is $min(p, q) + 1$.



(a) Test configuration for h-convergence

(b) **M4**

Figure 3.6 | Element distribution for the h-convergence validation. To test convergence, the domain is arbitrarily cut in two halves, each part meshed with a different element type.

3.2.4 Numerical study of hybrid interfaces

In a second step, we want to highlight the potential presence of parasitic reflection near the hybrid interfaces. For this purpose, we consider a cubic domain with absorbing boundary conditions, with the injection of a polychromatic incident plane wave in the x direction. This choice is motivated by

	M1		M2		M3		M4	
	$\|\mathbf{E}-\mathbf{E}_h\|$	r	$\|\mathbf{E}-\mathbf{E}_h\|$	r	$\|\mathbf{E}-\mathbf{E}_h\|$	r	$\|\mathbf{E}-\mathbf{E}_h\|$	r
$\mathbb{P}_1-\mathbb{Q}_1$	5.05×10^{-2}	–	1.28×10^{-2}	1.98	3.27×10^{-3}	1.97	1.47×10^{-3}	2.03
$\mathbb{P}_1-\mathbb{Q}_2$	6.90×10^{-2}	–	1.88×10^{-2}	1.87	4.82×10^{-3}	1.96	2.16×10^{-3}	2.05
$\mathbb{P}_1-\mathbb{Q}_3$	6.99×10^{-2}	–	1.89×10^{-2}	1.89	4.83×10^{-3}	1.97	2.16×10^{-3}	2.06
$\mathbb{P}_1-\mathbb{Q}_4$	6.69×10^{-2}	–	1.89×10^{-2}	1.89	4.83×10^{-3}	1.97	2.16×10^{-3}	2.06
$\mathbb{P}_2-\mathbb{Q}_2$	7.26×10^{-3}	–	9.82×10^{-4}	2.89	1.26×10^{-4}	2.97	3.74×10^{-5}	3.09
$\mathbb{P}_2-\mathbb{Q}_3$	7.07×10^{-3}	–	9.59×10^{-4}	2.88	1.23×10^{-4}	2.96	3.68×10^{-5}	3.08
$\mathbb{P}_2-\mathbb{Q}_4$	7.04×10^{-3}	–	9.67×10^{-4}	2.86	1.26×10^{-4}	2.94	3.79×10^{-5}	3.05
$\mathbb{P}_3-\mathbb{Q}_3$	6.62×10^{-4}	–	4.02×10^{-5}	4.04	2.54×10^{-6}	3.99	5.13×10^{-7}	4.07
$\mathbb{P}_3-\mathbb{Q}_4$	6.34×10^{-4}	–	4.07×10^{-5}	3.96	2.56×10^{-6}	3.99	5.18×10^{-7}	4.08

Table 3.2 | Error levels and convergence rates of the cubic cavity case for mixed orders of approximation on hybrid meshes of increasing refinement. All simulations were run with upwind fluxes and LSRK4 time integration.

the simulations of light trapping in PV cells, which in most cases uses this type of source. For the mesh, we construct a sample of possible non-conformity (see figure 3.7). All meshes are based on a cartesian grid of 216 hexahedra ($6\times6\times6$). Tetrahedral zones are obtained by splitting hexahedra in 6 tetrahedra. In figure 3.8(a), we compare the error on a tetrahedral mesh (DGTD-$\mathbb{P}_1$), with the two examples from the first configuration: DGTD-$\mathbb{P}_p-\mathbb{Q}_q$ means we inject the plane wave in the tetrahedral region ($\mathbb{P}$ first), while DGTD-$\mathbb{Q}_q-\mathbb{P}_p$ means we inject the plane wave in the hexahedral region ($\mathbb{Q}$ first). In figure 3.8(b), we consider the case where the hybrid interface is splitting the injection surface in half. For both graphs, the error is similar, with a slight increase for the second configuration.

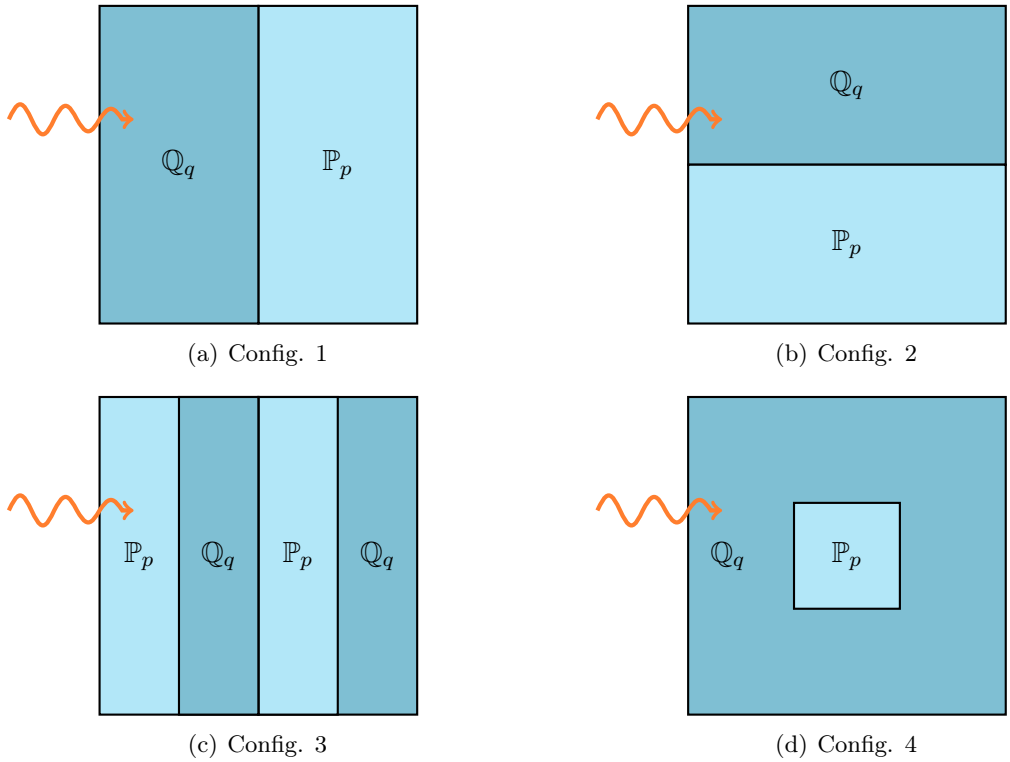
3.3 Realistic PV cells

PV structures present a large variety of configurations. However, most of them share common properties. Indeed, they are generally constituted of several stacked layers of materials, and the interface between each of them can be nanostructured for light trapping purposes [Pae12]-[WYL+12]. However, those nanotextures are only present at the junction between two layers and represent a very thin portion of the solar cell (a few nanometers compared to several hundreds, as shown in figure 3.9). Therefore, the use of hybrid meshes is relevant in this context.

The generation of hybrid meshes is possible by using the Gmsh meshing software [GR09][1]. Even if it is not really a proper feature from Gmsh, its ease and modularity make it possible to generate non-compliant and hybrid meshes of PV stacks quite easily. The main idea is to build the hybrid interfaces explicitly in the Gmsh script. This means that we must select in advance the regions that are meshed with hexahedra. This corresponds to having a set of intervals $[h^i_{\min}, h^i_{\max}]$, $i\in\mathbb{N}^*$, which represents the delimitation of each area.

The main restriction with this approach is that each hexahedral zone must have the form of a block. Our second possibility consists in using an intermediate tool, developed initially to study the radar cross section (RCS) of aircraft. This tool allows us to take as input a surface or volume, tetrahedral or hexahedral (or already hybrid) mesh, as well as a domain size ($l_x\times l_y\times l_z$), a max hexahedron size and a minimum hexahedron size. From these information, the software builds the

[1] http://gmsh.info/



(a) Config. 1 (b) Config. 2

(c) Config. 3 (d) Config. 4

Figure 3.7 | Hybrid configuration considered. Different hybrid interfaces to illustrate interface interferences.

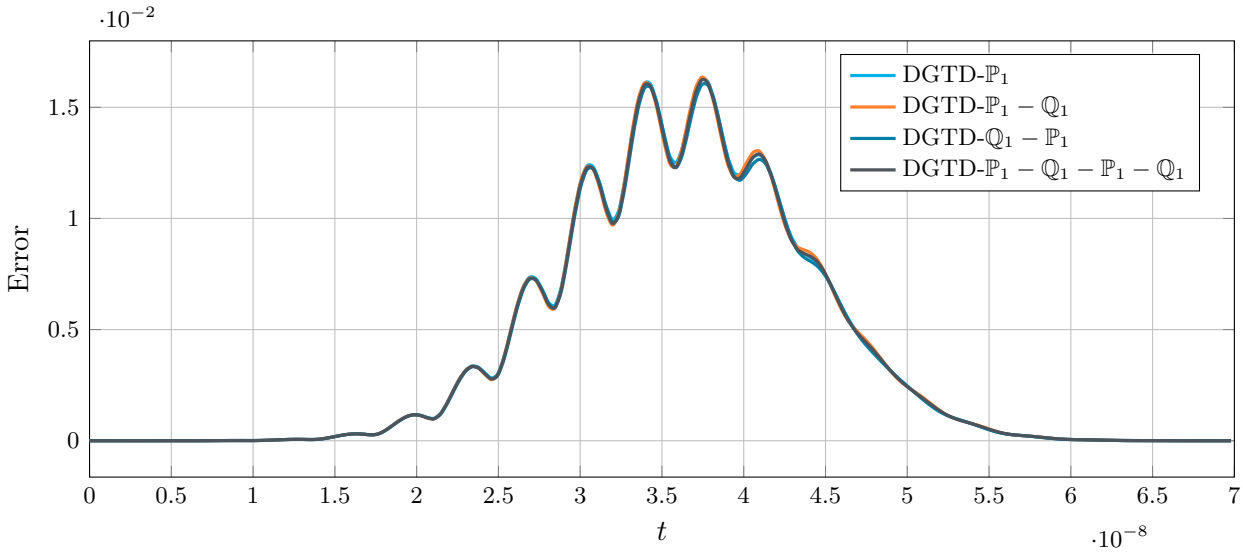
hybrid interface and the hexahedral zones around it. If the input meshes are surface, the Gmsh tool is used to fill the gap between interface and surface.

3.3.1 Numerical assessment for a double conical grating cell

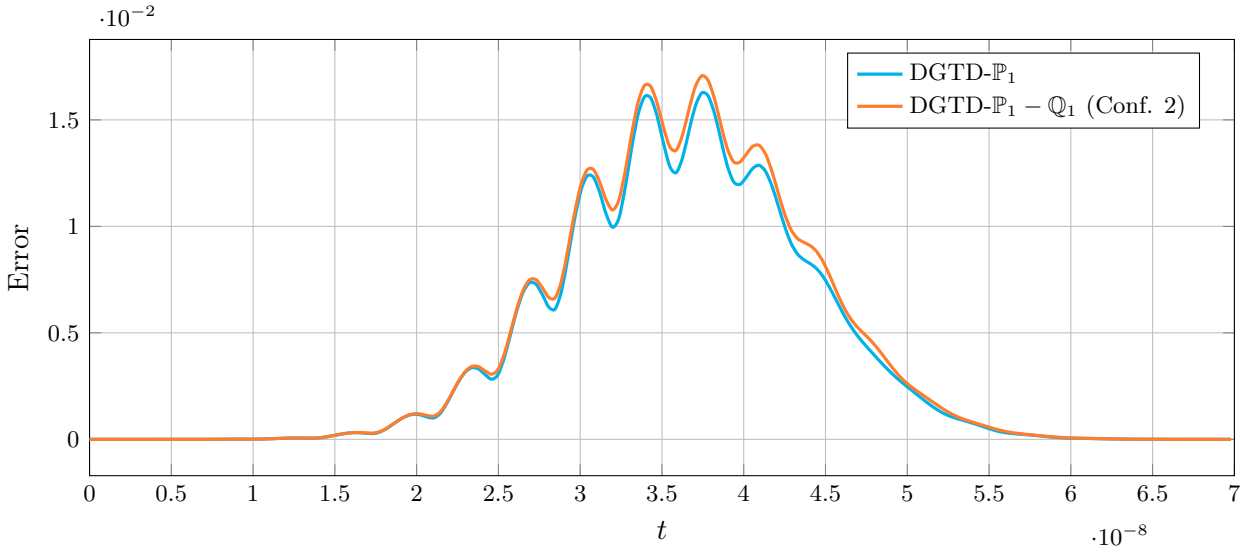
We consider the case of the nanostructured PV cell with the two conical gratings again [WYL+12]. The homogeneous regions are removed and replaced with layers of hexahedra. It concerns the silicon layer surrounded by the gratings, but also the artificial ones (PMLs and SF areas). One can see the hybrid mesh and its corresponding element's size in figure 3.10. For this setup, the same pattern as in section 3.1.3 is observed: the largest elements are hexahedra placed between the two gratings, while the finest ones are tetrahedra located at the extremity of cones.

Results are shown in figure 3.12. We start from the best working configuration using the point per wavelength strategy, as shown in part 3.1.3, *i.e.* using $\overline{n}_{\text{ppw}} = 6$. The results are matching on the range [230 nm, 500 nm] for both $\mathbb{P}_1 - \mathbb{P}_4 - \mathbb{Q}_3$ and $\mathbb{P}_1 - \mathbb{P}_4 - \mathbb{Q}_4$ interpolation orders. However, for the very first wavelengths, there is a lack of reflection when $\mathbb{Q}_3$ interpolation is used. We also plot in figure 3.11 the contour lines of the amplitude of the DFT of $\mathbf{E}$ at $\lambda = 450$ nm.

The computational times are shown in table 3.3. The results for the original DGTD-$\mathbb{P}_4$ method and the improved DGTD-$\mathbb{P}_1 - \mathbb{P}_4$ version using local interpolation order with $\overline{n}_{\text{ppw}} = 6$ are recalled, as well as the two hybrid configurations considered. The speedup in parentheses corresponds to the speedup from the DGTD-$\mathbb{P}_1 - \mathbb{P}_4$ simulation. The DGTD-$\mathbb{P}_1 - \mathbb{P}_4 - \mathbb{Q}_4$, $\overline{n}_{\text{ppw}} = 6$ presents a speedup of 1.40 as compared to the p-adaptive simulation. We do not see a real overhead due to the construction and storage of the specific hybrid matrices. In any case, this overhead is mainly present in the initialization phase, which is of less importance compared to the time loop.



(a) Config 1 and 3



(b) Config 2 and 4

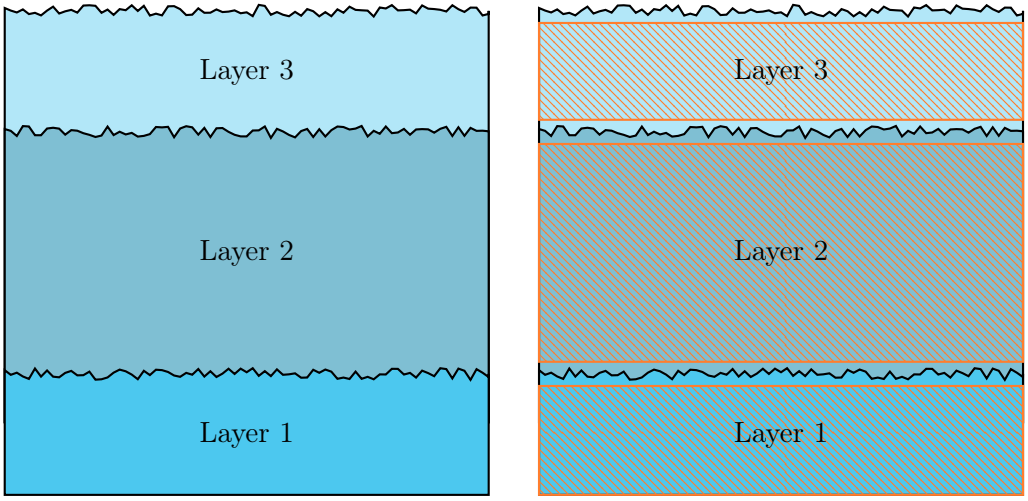
Figure 3.8 | Analysis of the error (L^2 in space, L^∞ in time) for different non conform interfaces (config 1 and config 3).

	Computation time	Speedup
$\mathbb{P}_4$	26 h 00 mn	-
$\mathbb{P}_1 - \mathbb{P}_4$, $\overline{n}_{\text{ppw}} = 6$	13 h 12 mn	1.95
$\mathbb{P}_1 - \mathbb{P}_4 - \mathbb{Q}_3$, $\overline{n}_{\text{ppw}} = 6$	4 h 26 mn	5.45 (3.00)
$\mathbb{P}_1 - \mathbb{P}_4 - \mathbb{Q}_4$, $\overline{n}_{\text{ppw}} = 6$	9 h 12 mn	2.85 (1.40)

Table 3.3 | Computation time for each simulation set-up on a Intel(R) Xeon(R) Gold 6154 CPU at 3.00GHz.

3.3.2 Numerical assessment for multilayer PV type nanostructures

Here we consider an idealistic PV stack composed of six layers. The interface geometry is manually defined using $a(x, y) = \cos\sqrt{y^2}$, which gives a bi-periodic domain. We consider five different



(a) PV stacks (b) PV stacks

Figure 3.9 | PV stack example with structured interface. The textured part represents a vary small percentage of the total domain.

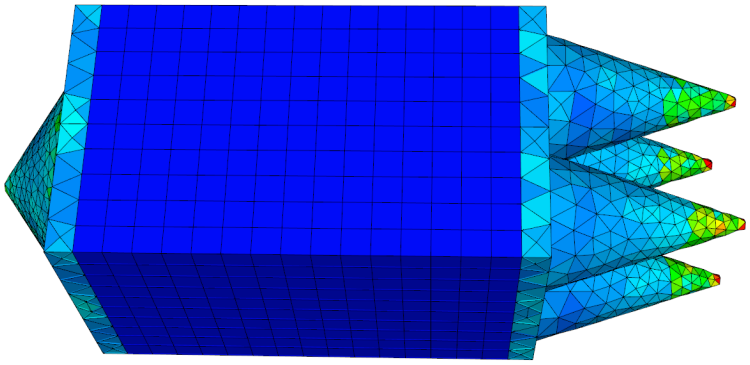


Figure 3.10 | Nanostructured mesh of the double grating solar cell: the color corresponds to the cell size.

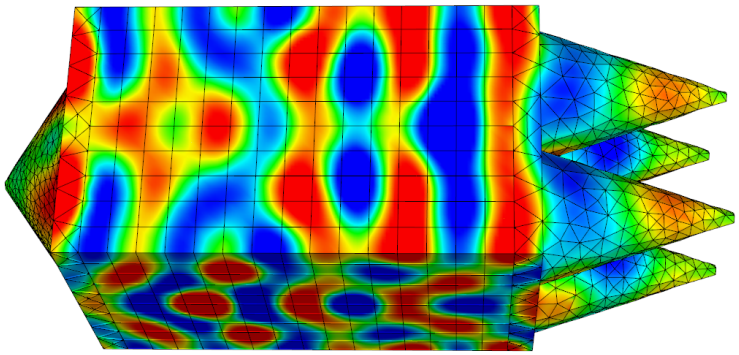


Figure 3.11 | Simulation of light trapping in a solar cell based on nanocone gratings. Simulations using hybrid mesh: contour lines of the module of the DFT of $\mathbf{E}$ for a wavelength $\lambda = 450$ nm.

configurations: M1 and M2 are hexahedral meshes, M3 and M4 are tetrahedral meshes, and M5 is a hybrid tetrahedral/hexahedral mesh (see the meshes in figure 3.13 and their characteristics in

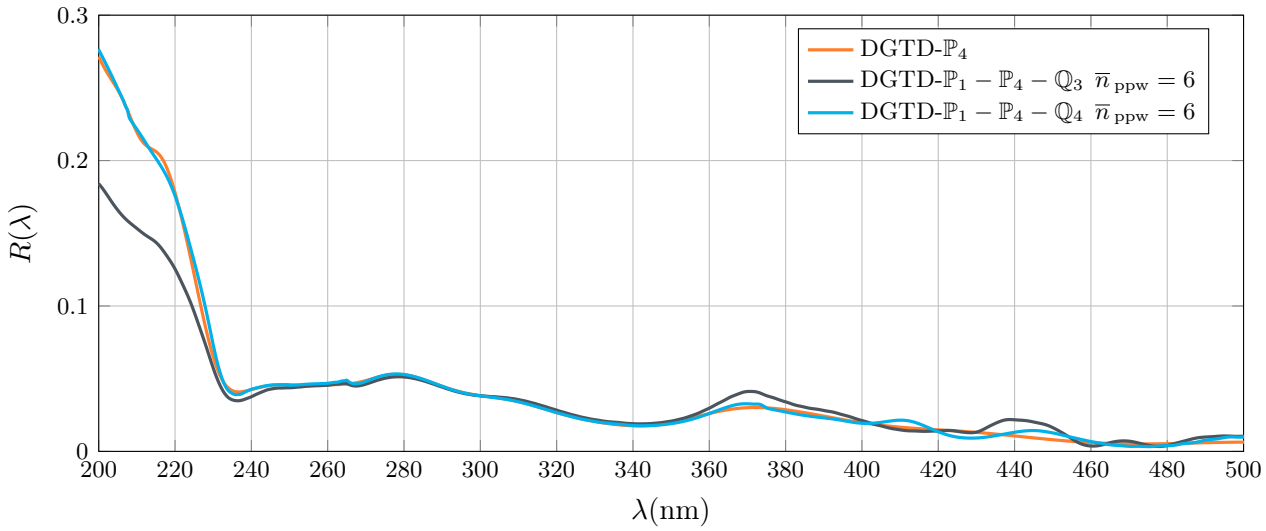


Figure 3.12 | Reflection coefficient for DGTD-$\mathbb{P}_4$ (reference) and two DGTD-$\mathbb{P}_1 - \mathbb{P}_4 - \mathbb{Q}_q$ hybrid simulations.

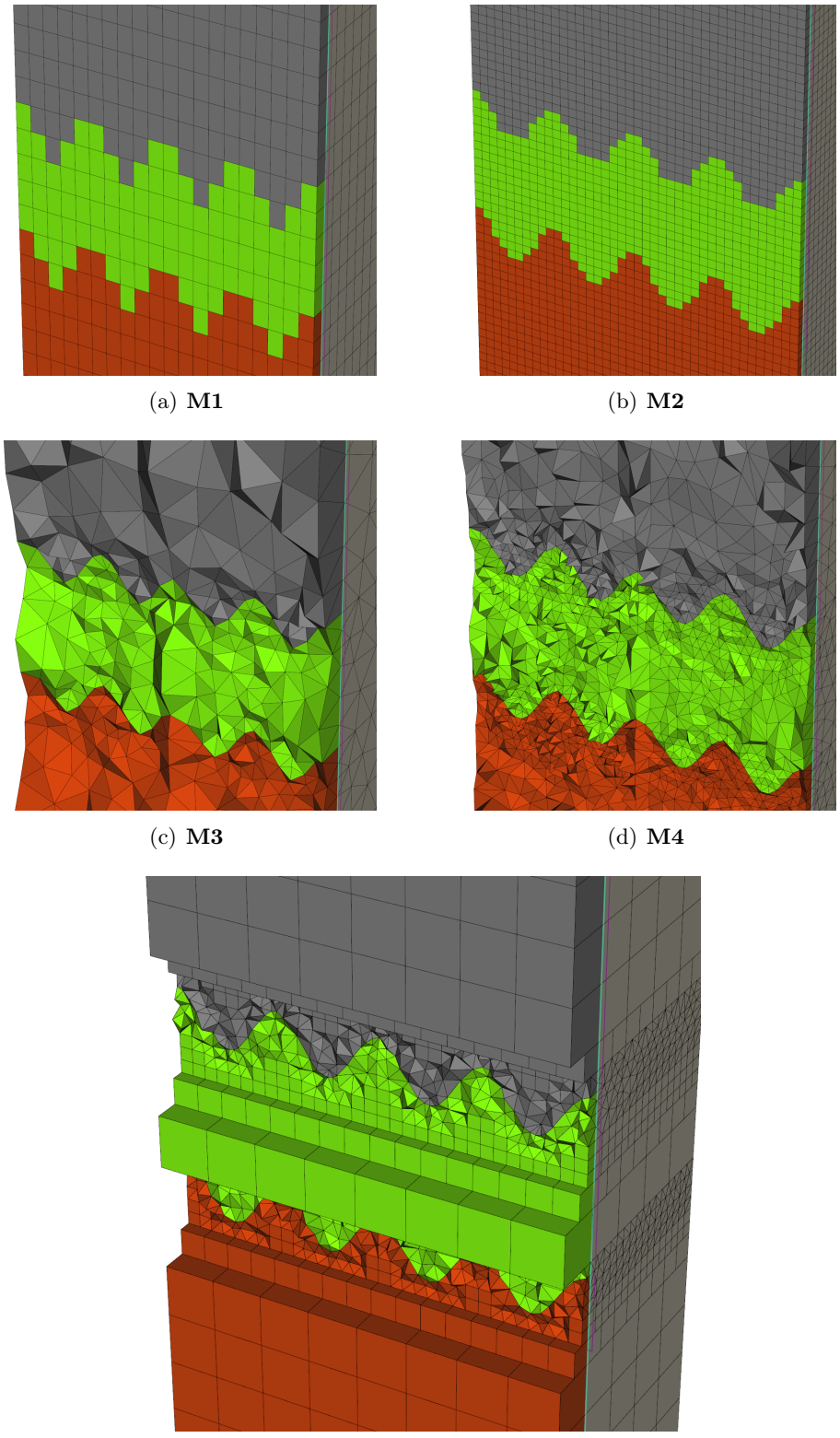
table 3.4).

	Number of cells	$h_{\min}\,(nm)$	$h_{\max}\,(nm)$
M1	24000 hexahedra	40	70
M2	320000 hexahedra	20	20
M3	51816 tetrahedra	11	25
M4	414528 tetrahedra	6	19
M5	5094 hexahedra	6	100
	+ 21380 tetrahedra		

Table 3.4 | Characteristics of different configuration meshes.

For this problem, the quantity of interest considered is the volumetric absorption (defined in chapter 4). A convergence analysis based on this quantity is done in figure 3.14. The volumetric absorption profile obtained with the coarse hexahedral mesh has a general appearance close to the reference solution. However, resonances near 500 nm are not well captured. On the contrary, the primary resonance is not in the same place. It is important to note that this anomaly does not come from a convergence concern but from a geometric representation error due to the staircasing effect. The refinement of the cartesian grid has been done in such a way as to obtain an equivalent solution. It results in a hexahedral mesh of 320000 elements. Mostly because of the simplicity of the textured interfaces, we obtained almost the same results for both tetrahedral meshes. The hybrid mesh generated is using two levels of hexahedron refinement in order to restrict the tetrahedra area. Due to the high ratio between the smaller and the lower, we obtained better results using a $\mathbb{P}_2$ interpolation for the tetrahedral region while using a mixed $\mathbb{Q}_2 - \mathbb{Q}_3$ for the hexahedral one. Subfigure 3.14(f) show the comparison of each converged results, which are all really close, except for the coarse hexahedral mesh.

Now that we see that M2, M3, M4 and M5 can achieve the same (good) result, we want to highlight the computation cost of all of them. The simulations have been performed sequentially on an Intel(R) E5-465L (2.60 GHz - 8 cores) with 512 go of RAM. We choose this workstation

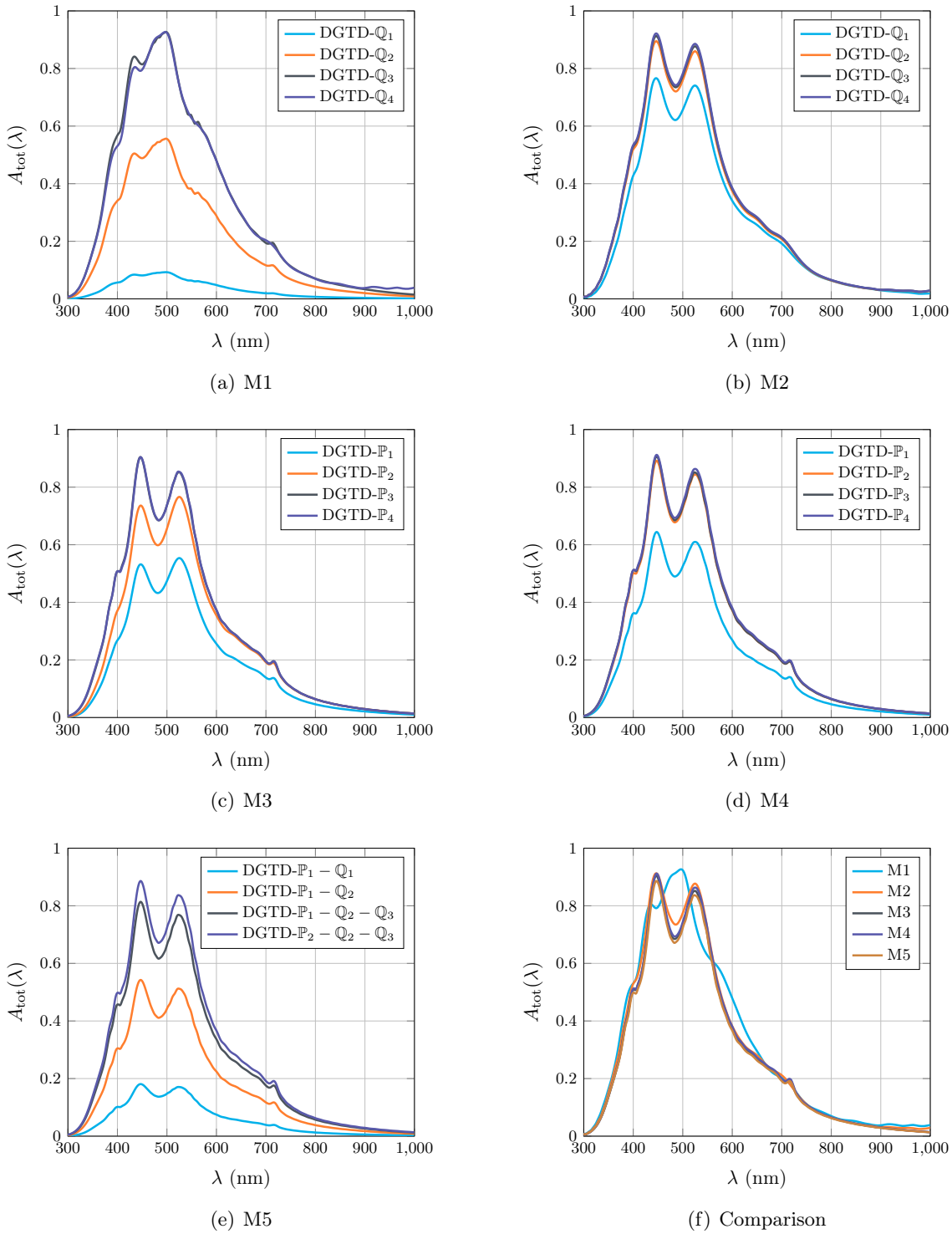


(a) **M1** (b) **M2**

(c) **M3** (d) **M4**

(e) **M5**

Figure 3.13 | Zoom on an interface region for the five considered meshes. Each color represents a specific material.



(a) M1 (b) M2

(c) M3 (d) M4

(e) M5 (f) Comparison

Figure 3.14 | Convergence study for meshes M1 to M5 on a quantity of interest (A). Subfigure (f) compares the results obtained for each mesh.

due to its huge memory, which allows us to compute our quantity of interest on M2 and M4. The computation times are shown in table 3.5.

		Number of time-steps	Computation time
M1	$\mathbb{Q}_3$	16840	1 h 55 mn
M2	$\mathbb{Q}_2$	16932	111 h 21 mn
M3	$\mathbb{P}_3$	38678	2 h 11 mn
M4	$\mathbb{P}_2$	103199	24 h 19 mn
M5	$\mathbb{P}_2 - \mathbb{Q}_2 - \mathbb{Q}_3$	18270	5 h 11 mn

Table 3.5 | Computation time for each meshes on an Intel(R) E5-465L (2.60 GHz - 8 cores)

3.4 Conclusion

In this part, we have presented two numerical strategies that enable a better resolution of complex geometrical problems. In the first part, we recalled some preliminary work done by Viquerat *et al* in [VL16] about local polynomial approximation. In this work, we extended the latter with a criterion based on the number of degrees of freedom per wavelength. In comparison with [Viq16], this approach allows us to have an a posteriori control of the behavior of the simulation from a starting mesh. Some interesting results have been shown, with a number of points per wavelength between 6 and 8. This allows an easier use of very various mesh sizes. In a second time, we presented the modifications required in the DGTD formulation in order to take into account hybrid meshes. A convergence analysis has been performed on a basic test case, and we investigated the behavior of several hybrid interfaces. From our examples, we did not experiment with any notable degradation with an incident normal polychromatic plane wave. Finally, in the last part, we introduced a multilayer PV structure. We highlighted the influence of the staircasing effect, which is one of the major drawbacks of the use of cartesian grids. Then, performances of the previous methods were compared, with a clear improvement when combining hybrid non conform meshes with local polynomial interpolation, both in computational time and in number of d.o.f..

4

Realistic simulation of light trapping in complex solar cell structures

Contents

Photovoltaics (PVs) is a very prominent field of research, with widespread awareness of renewable energy. One of the leading scientific concerns around PVs lies in the discovery of processes and strategies to improve the performance of solar cells, in order to enable them to become the reference green energy. Since the beginning of PVs, silicon has been the primary material used as an absorber, despite its significant defects. This is majorly caused by its low price. One of the main way to overcome these failures is to use light trapping to focus the light in the absorber. Light trapping consists in structuring one (or more) layers so that the light is trapped in the absorbing material. This structuring is the subject of much research [WYL+12],[EJD14],[Loc12]. Knowing that the costs and time required to manufacture the cell are not negligible, the use of numerical simulations seems to be an asset and a primary possibility for the testing and calibration of sophisticated PV devices.

The main criterion to evaluate and sort solar cells is the efficiency. It corresponds to the percentage of light received and transformed into electrical power. This transformation involves several physical processes, including optics and electricity. In the context of this thesis, we only consider the optical part, which consists in quantifying the proportion of incident solar light absorbed in a specific material, generally a semiconductor. We will not deal with the generation of electron hole pairs, recombination, and carrier transport, which are the other fundamental parts of solar cell.

Three possibilities occur when light reaches the solar cell: (i) the incident photons are reflected at the top surface, (ii) they are absorbed in the material, or (iii) the light is transmitted through the cell. For most of the PV devices, both reflection and transmission are considered as loss given that photons are not absorbed, and therefore, do not generate power. Both issues can be taken into account in different ways. Usually, reflection is handled using anti reflective coating [RGNR11]. This technology is not restrained to PV; it is also used in a wide variety of applications where low reflection is desired, *e.g.* on corrective lenses, and camera lens elements. To overcome the transmission, a mirror is often used at the back of the cell, reflecting the light initially transmitted and allowing a second pass through the absorber. This allows to exceed the single pass limit absorption. In recent research, those two strategies have been used in order to improve PV cell efficiencies. For example, recently in [CCDL+19], Chen *et al* used a nanostructured back mirror to reach the efficiency record for ultrathin solar cells (nearly 20%). The simulation of this part can be done by solving full-wave Maxwell's equations.

In this chapter, we first review the most commonly used numerical approaches for the simulation of light trapping in solar cell devices in section 4.1. In section 4.2, we recall the definition of the figures of merit that we will use for assessing the efficiency of such devices from the point of view of solar light absorption, such as the reflection and the transmission coefficients, External Quantum Efficiency (EQE) and the short circuit current density (Jsc). Then we present the similarities shared by PV geometries that we usually encounter, namely their bi-periodicity. We also present some methods to manage high definition measured textures data with noise. Those challenging data are preprocessed to create a good and usable mesh for our DGTD solver. We discuss both meshing aspects, as well as the observable of interest, specific to PV applications in section 4.3.

All these elements are illustrated with several realistic test cases from PV literature or studies undertaken in collaboration with physicists in the PV field in section 4.4, 4.5 and 4.6.

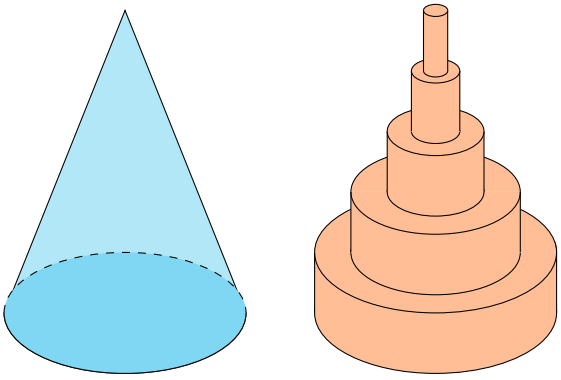
4.1 Numerical modeling of light trapping in PV devices

Solar cell physics is composed of several branches. The two main ones are: (i) optics, which characterize the interactions between solar light and the solar cell and, (ii) semiconductor physics, which deals with the generation of electric current from carrier generation and transport. In order to simulate each of those parts, it is necessary to use different methods, each adapted to model the physics of interest. For the semiconductor physics, one can solve the drift-diffusion model [ACM].

For the optical part, which is the physics we are considering in this thesis, several methods exist for the computation of solar cell properties. To cite a few of them, the Ray Tracing method is well adapted for textures with characteristic sizes that are larger than the relevant wavelength [BFM11]. This method literally consists in tracing a ray of light using simple geometrical optics based on the refractive index of each material. If layers are thin but still unstructured, it is possible to use the Transfert Matrix (TM) method. TM method is based on continuity conditions for the electric field between two media. Knowing the field at one side of a layer allows us to compute the field at the other side from a simple matrix operation. The Rigorous Coupled-Wave Analysis [MG81] is suitable for solving scattering from relatively small periodic structures. When texturization details are smaller or close to the relevant wavelength, it is important to take into account the optical wave effects. This is possible via discretization methods, that solve electromagnetic waves equations such as the FDTD [TH05] or the Finite Element Frequency-Domain (FEM) [Mon03] methods. Recently, some coupling methods have emerged. They proposed to couple different simulators (usually RCWA, ray tracing, and TM) in order to combine the benefits of each. The two main methods are OPTOS [TEK$^+$15] and the Coupled Modelling Approach (CMA) [TSL$^+$15].

4.1.1 Rigorous Coupled-Wave Analysis (RCWA)

The Rigorous Coupled-Wave Analysis, introduced in 1981 by Moharam [MG81], is a frequency-based, semi-analytical optical simulation method. It is mostly used to solve scattering from periodic dielectric structures. The RCWA method starts by discretizing the domain into layers in the z direction. Curved gratings need stairstep-approximated layers (see figure 4.1). Electromagnetic modes are calculated in each layer and then analytically propagated through the layers by matching boundary conditions at layer interfaces, in the same fashion as TM. These modes are expanded using Fourier series, which leads to an infinitely large system of Ordinary Diffential Equations (ODE). A finite-dimensional problem can be reached by truncating the Fourier expansions, at the cost of the accuracy of the method. RCWA method is very efficient for the calculation of far-field reflection and transmission coefficients, but the accurate computation of the near field at material boundaries remains challenging. RCWA has been used for many optical simulation of nanotextured solar cells [ABK$^+$19],[BEH$^+$16],[ACM]. The main drawback of the RCWA method is that it is not suited for non-cartesian geometries (see figure 4.1). Moreover, even if the method is dedicated to periodic structures, the number of modes necessary can become enormous. Indeed, it is related to the size of the elementary pattern and of the relevant wavelength. This generally leads to huge matrices to be inverted.

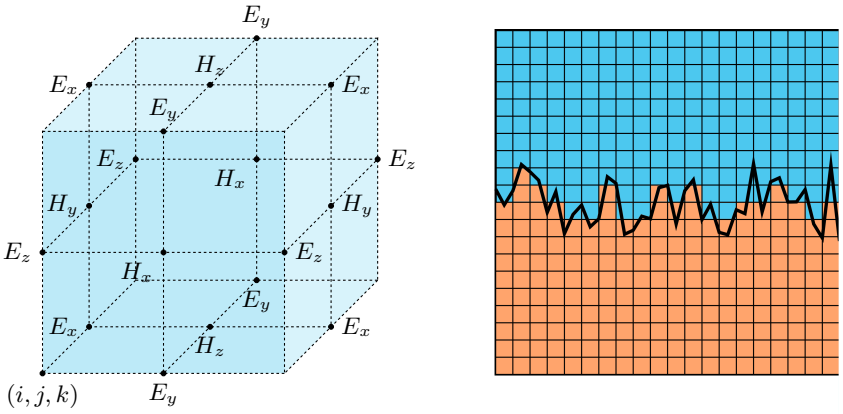


(a) Original shape (b) Stratified shape

Figure 4.1 | A very coarse stratification (b) of a cone (a). Non-cartesian geometries in the z-direction are split in a set of z parallel layers.

4.1.2 Finite Difference Time-domain (FDTD)

The Finite Difference Time-domain (FDTD) method is undoubtedly one of the most popular methods for time-domain computational nanophotonics. This method is also widely used for the study of light trapping in solar cell devices. The method relies on two Taylor expansions for both spatial and temporal derivatives. The popular version presented by K.S. Yee in 1966 [Yee66] relies on a particular discretization of the computational space. The spatial element used is now referred to as Yee cells (see figure 4.2(a)). In the original Yee scheme, the spatial derivatives are discretized using second-order central differences while time integration is achieved with a second-order leap-frog scheme. As of today, FDTD represents an easy to use and efficient method to solve electromagnetics problems, combining simple implementation and high computational efficiency.



(a) Yee cell (b) Staircasing effect on a complex interface

Figure 4.2 | Yee cell and staircasing effect. (a) The **H** field components are on the center of the faces, while the **E** ones are on the center of the edges. (b) Cartesian discretization of a textured interface. The materials are spread in cells according to the percentage of volume physically above or below the textured surface.

However, the FDTD method also presents some limitations. First, a smooth discretization

of curved geometries is impossible due to the fixed cartesian grid imposed by the Yee algorithm. This approximation leads to the well-known staircasing effect (see figure 4.2(b)), which is an important source of inaccuracy [DDH]. To overcome this pitfall, one can either use an extreme refinement of the grid, which leads to a drastic rise of the computational cost, or exploit one of the numerous possible modifications of the FD method that have been proposed for tackling the staircasing effect [HR98]. However, these modifications represent a tradeoff between the simplicity of the classical algorithm and the accuracy of the boundary description. An other source of inaccuracy in the FDTD method is faced when dealing in the case of heterogeneous problems. Indeed, since the electromagnetic fields are not smooth at the interface of different medium, the Taylor approximations used are no longer valid. More advanced FDTD methods aim to address this issue [TH05]. Unfortunately this is once again at the price of an increased complexity of the algorithm.

4.1.3 Finite Element Method (FEM)

Continuous FEM is used as an alternative method to the FDTD method. The FEM for CEM was introduced in 1969 by Silvester to solve waveguide problems [Sil69]. Unlike the FDTD, here the domain tessellation is an unstructured tetrahedral or hexahedral mesh. A discrete variational formulation is obtained from the continuous Maxwell's equations by approximating the unknowns in a finite-dimensional space. Then, the discretization of this weak form leads to a sparse system of equations in terms of the whole set of d.o.f.. This system is one of the drawbacks of FEM methods in time-domain since it must be solved at each time-step. Moreover, one should avoid the use of nodal basis functions as it was proved that they could lead to spurious modes, due to an ill representation of the curl kernel [SMYC95]. Edge finite elements (also known as Nédélec edge elements) were introduced in 1986 by Nédélec [Né80]. These elements display several interesting properties: (i) their divergence is zero, and (ii) each basis function associated with an edge has a constant tangential component on the latter, and a zero tangential component on the others (see figure 4.3). This leads to a naturally enforced tangential continuity of the electric field across the edges.

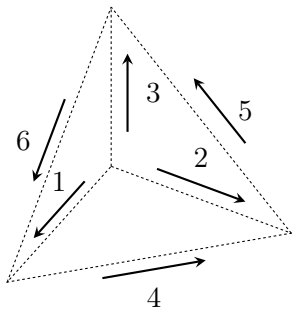


Figure 4.3 | The d.o.f. are associated to the edges of the element to naturally satisfy the continuity requirements for the approximated vector fields.

In order to adjust the accuracy of the simulation, FE methods can use either (i) a local refinement or coarsening of the mesh, (ii) a local or global increase of the order of the basis functions (if polynomial, it corresponds to an higher polynomial order), or (iii) a combination of both. However, these improvements lead to larger linear systems to solve at each time-step, which can make the

FE method impractical in time-domain simulations for huge systems. For this reason, FE methods are more often used in frequency-domain. However, a few references can be found exploiting time-domain FE for nanophotonics applications [HLY13].

4.2 Physical quantities of interest

When assessing the quality of a PV device, some fundamental quantities allow to measure the performances of different criteria such as its anti-reflection capability or the maximum current than can be generated from a solar cell. In this section, we introduce the figures of merit that we will use in our simulations.

4.2.1 Reflection, transmission and absorption

First of all, we need to introduce the Poynting vector. It is an appropriate quantity to describe the energy flux of a propagating electromagnetic wave. Its definition in time-domain is:

$$\boldsymbol{\pi}(\mathbf{x},t) = \mathbf{E}(\mathbf{x},t) \times \mathbf{H}(\mathbf{x},t). \tag{4.1}$$

More precisely, we are interested in the time-averaged Poynting vector which is define as:

$$\boldsymbol{\pi}(\mathbf{x},\lambda) = \frac{1}{2}\Re\left(\widehat{\mathbf{E}}(\mathbf{x},\lambda) \times \widehat{\mathbf{H}}^{*}(\mathbf{x},\lambda)\right), \tag{4.2}$$

with $\widehat{\mathbf{E}}$ and $\widehat{\mathbf{H}}$ the Fourier transform of respectively $\mathbf{E}$ and $\mathbf{H}$ and λ the wavelength. Given that definition, consider the situation of figure 4.4, where the scatterer is enclosed between two TF/SF surfaces. Between the two surfaces, *i.e.* in the total field region, the computed fields are $\mathbf{E}_{\text{tot}}$ and $\mathbf{H}_{\text{tot}}$ and one as:

$$\begin{aligned} \mathbf{E}_{\text{tot}} &= \mathbf{E}_{\text{inc}} + \mathbf{E}_{\text{sca}}, \\ \mathbf{H}_{\text{tot}} &= \mathbf{H}_{\text{inc}} + \mathbf{H}_{\text{sca}}. \end{aligned}$$

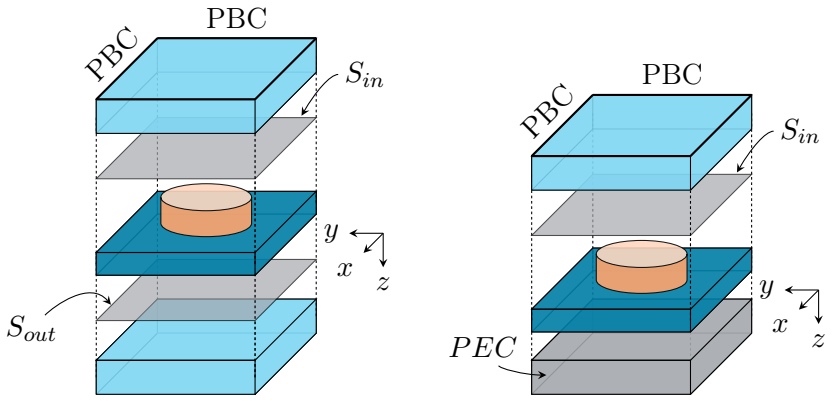
The incident field $(\mathbf{E}_{\text{inc}}, \mathbf{H}_{\text{inc}})$ is imposed on the TF/SF interface, and the computed fields in the scattered region are $\mathbf{E}_{\text{sca}}$ and $\mathbf{H}_{\text{sca}}$. Using these fields we can define $\boldsymbol{\pi}_{\text{inc}}$ and $\boldsymbol{\pi}_{\text{sca}}$ as:

$$\boldsymbol{\pi}_{\text{inc}}(\mathbf{x},\lambda) = \frac{1}{2}\Re\left(\widehat{\mathbf{E}}_{\text{inc}}(\mathbf{x},\lambda) \times \widehat{\mathbf{H}}^{*}_{\text{inc}}(\mathbf{x},\lambda)\right), \tag{4.3}$$

$$\boldsymbol{\pi}_{\text{sca}}(\mathbf{x},\lambda) = \frac{1}{2}\Re\left(\widehat{\mathbf{E}}_{\text{sca}}(\mathbf{x},\lambda) \times \widehat{\mathbf{H}}^{*}_{\text{sca}}(\mathbf{x},\lambda)\right), \tag{4.4}$$

which are the key ingredients for the calculation of the physical quantities of interest. Now we can define the reflection (R) and transmission (T) coefficients, which are two wavelength-dependent observables. When a scatterer is illuminated by an incident field, R represents the proportion of energy reflected to the injection plan, while T is the amount of energy that traveled through the system:

$$R(\lambda) = \frac{\displaystyle\int_{S_{in}} \boldsymbol{\pi}_{\text{sca}}(\cdot,\lambda)\cdot\mathbf{n}}{\displaystyle\int_{S_{in}} \boldsymbol{\pi}_{\text{inc}}(\cdot,\lambda)\cdot\mathbf{n}}, \tag{4.5}$$



(a) Device enclosed between two TF/SF infinite planes

(b) Device enclosed between a TF/SF infinite plane and an infinite mirror

Figure 4.4 | Periodic array of a textured layer. The TF/SF planes, S_{in} and S_{out}, are respectively above and below the layer (see the axes). The injection is done in the z direction and PML layers , while it is periodic in the x and y directions.

and:

$$T(\lambda) = \frac{\displaystyle\int_{S_{out}} \boldsymbol{\pi}_{\text{tot}}(\cdot,\lambda)\cdot\mathbf{n}}{\displaystyle\int_{S_{in}} \boldsymbol{\pi}_{\text{inc}}(\cdot,\lambda)\cdot\mathbf{n}}, \tag{4.6}$$

where S_{in} and S_{out} are the injection and output planes defined in figure 4.4 and $\mathbf{n}$ is the outgoing normal vector. Additionally, we can define the absorption (A) from R and T as:

$$A(\lambda) = 1 - T(\lambda) - R(\lambda). \tag{4.7}$$

which represents the amount of energy absorbed by the device. It is one of the most important quantity in the PV field as it represents the amount of energy absorbed by the cell, and so potentially transformed to electricity. The equation (4.7) point out two primary strategies to optimize solar cells absorption: (i) one can use an anti-reflection (AR) coating at the front surface of the cell to reduce R, (ii) or one can use a backside mirror, most of the time made of silver, to minimize T and lead to light trapping.

***Remark** : For non-periodic scattering problems, the absorption cross sections (C_{abs}) have to be used in order to compute the absorption [VL16].*

In the following, A computed from (4.7) will be noted A_{tot} since it corresponds to the total absorption in the PV cell.

4.2.2 Volumetric absorption

Absorption can also be computed with a volumetric method. It is possible to evaluate the Ohmic losses (W) inside a given material instead of computing the flux of the Poynting vector through surfaces. It is shown in [LL60] that the power absorbed by the scatterer as Ohmic losses is:

$$A_s(\lambda) = P_{Ohm}(\lambda) = \frac{\varepsilon_0 \lambda}{2} \int_{\Omega_s} \Im(\varepsilon_r(\lambda)) \left|\widehat{\mathbf{E}}(\mathbf{r},\lambda)\right|^2, \tag{4.8}$$

where Ω_s is the volume of the scatterer s. If the domain is composed of only one absorber, one can choose which quantity to compute, *i.e.* the volumic or the surfacic absorption. As discussed in [KM10], the computation of the absorption via the Poynting vector integration shows large errors when there is a strong scattering and can even produce unphysical negative absorption. These effects are well explained in the paper mentioned above. However, evaluation of the volumetric absorption is way more expensive as it requires to compute an integral over a volume rather than a surface.

4.2.3 Quantum Efficiency

Quantum Efficiency (QE) or Incident Photon to Converted Electron (IPCE) is a frequency-dependent quantity used to measure the efficiency of a PV device. It is one of the key quantities as it gives information on the current that a given cell will produce when illuminated by a particular wavelength. There are two quantum efficiency definitions used for the study of solar cell devices: the EQE, which is the ratio of the number of electrons generated to the number of incident photons per second and the Internal Quantum Efficiency (IQE), which is the ratio of the number of electrons generated to the number of incident photons absorbed by the cell per second as summarized in the following equations:

$$\mathrm{EQE} = \frac{\mathrm{electrons/sec}}{\mathrm{photons/sec}}, \tag{4.9}$$

$$\mathrm{IQE} = \frac{\mathrm{electrons/sec}}{\mathrm{absorbed\ photons/sec}}. \tag{4.10}$$

Considering that the DGTD solver used in this thesis is only dealing with optical effects, we can assume that all absorbed photons are collected at the junction without recombination. Then, the EQE is computed as:

$$\mathrm{EQE}(\lambda) = \frac{P_{Ohm}(\lambda)}{I_0 \cdot S_{\mathrm{dom}}} = A(\lambda), \tag{4.11}$$

where I_0 is the intensity of the incident light and S_{dom} is the surface area of the periodic pattern perpendicular to the incident light.

Remark *: Unlike the absorption calculated from R and T, volumetric absorption can be calculated in various materials. However, the volumic absorption in each layer must coincide with A_{tot}.*

4.2.4 Short circuit current and short circuit current density

One other important quantity is the short circuit current (Isc) (mA). It represents the current through the solar cell when the solar cell is short-circuited. The short-circuit current is due to light-generated carriers being generated and collected. It is therefore the largest current from the solar cell that can be drawn. It is more common to use the Jsc (mA / cm^2) which is the current produced per unit cell area, rather than the Isc because it removes the dependence of the solar cell area. Jsc mostly depends on two ingredients: the incident light spectrum or spectral irradiance, standardized to the AM1.5G spectrum for most of terrestrial solar cell measurements (see figure 4.5); and the EQE. From those two ingredients we can compute it as:

$$\mathrm{Jsc} = \int q\mathrm{EQE}(\lambda)\mathrm{AM1.5g}(\lambda)d\lambda, \tag{4.12}$$

with q is the elementary electronic charge (C).

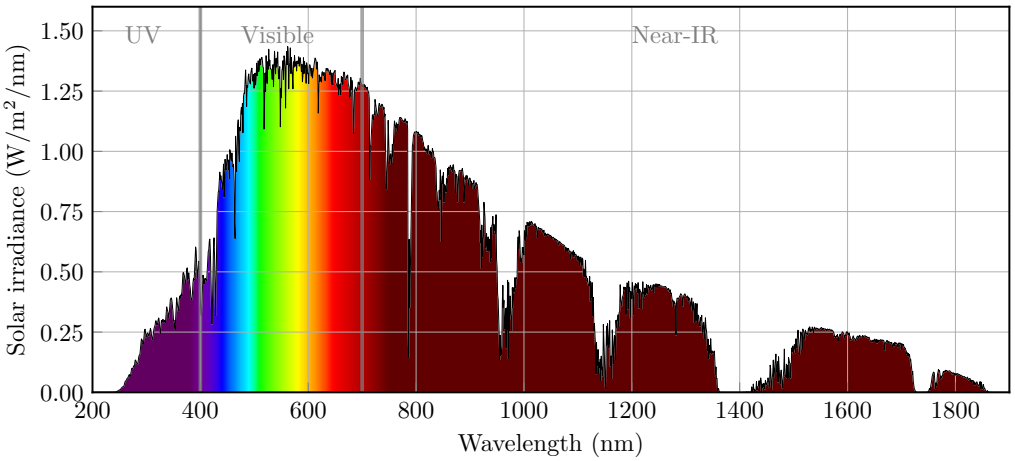


Figure 4.5 | Standardized AM1.5G spectrum. This spectrum is a good representation of the illumination conditions in the geographical mid-latitudes on a tilted PV flat plate array on a clear day without clouds [GME02].

Remark *: Some of the quantities defined above are sometimes referred to as near-field quantities and others as far-field quantities. It is important to notice that the numerical convergence of each field type requires different prerequisites, which will be the topic of a forthcoming part.*

4.3 Dealing with random textures

In this section, we explain the different steps required to move from random textures given as AFM (Atomic Force Microscopy) data, to a geometrical model that can be used for the simulations with our DGTD solver. Such a geometrical model consists of a fully unstructured tetrahedral or hexahedral/tetrahedral mesh, which is processed using available mesh generation tools. The two tools used in this thesis are Gmsh [GR09], a free 3D finite element mesh generator with a built-in CAD engine, and MeshGems [1], a suite of meshing software components, integrated in most of the CAD vendors.

4.3.1 Texturized structures from AFM

The main tool used to inspect the shape of textured layers is Atomic Force Microscopy (AFM). It is a very high resolution type microscopy capable of a resolution on the order of fractions of a nanometer. AFM outputs a discrete planar grid defined such that $(x, y) \in [0, L]^2$ with $x = n\delta h$, $y = m\delta h$, δh the resolution between two measured points and $(n, m) \in [0, N]^2$, $N = L/\delta h$. For all grid points, we also have a corresponding altitude noted $a(x, y) > 0$.

From there, we already have, in some ways, a quadrilateral surfacic mesh of the layer. However, most of the time, these data present abrupt jumps and spikes, which occur because of the way the AFM is used. In order to remove these defects, we choose to consider our set of points as an image, with the altitude a representing the grayscale of each point (x, y). Doing so, we can use image smoothing techniques coming from the digital image processing field to remove high-frequency

[1] http://www.meshgems.com/

content such as noise or edges. Two types of smoothing were tried here. First, we chose to use a box blur, also known as box linear filter. This method is easy to implement as it is basically a local average of each point. Mathematically, it consists of convolving the image with a normalized box filter:

$$a_{\text{post}}(x, y) = K * a(x, y), \tag{4.13}$$

with a_{post} the filtered image and K the filter kernel used here. The kernel K represents the shape and the size of the neighborhood taken into account. In this work we choose to take a square symmetric kernel:

$$K = \frac{1}{n^2} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & & 1 \\ \vdots & & \ddots & \vdots \\ 1 & \cdots & & 1 \end{bmatrix}_{n \times n}, \tag{4.14}$$

with $n > 0, n \in \mathbb{N}$. For example, we obtain the identity by taking $n = 1$. An illustration of the convolution for a 5×5 input dataset and a kernel with $n = 3$ is shown in figure 4.6.

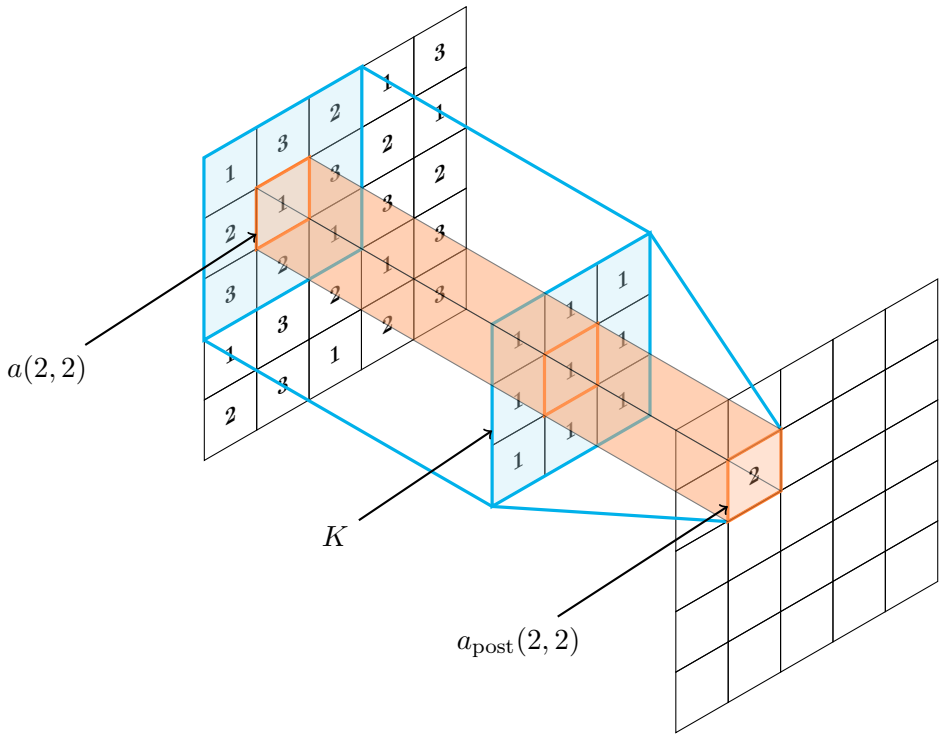


Figure 4.6 | Convolution of the input topography data with a moving-average filter. Center element of the kernel is placed over the source data, which is replaced by a average of itself and the nearby values.

By specifying that we only consider kernels with odd n, we can define $\overline{n} = \frac{n-1}{2}$. Then, we can develop equation (4.13) as:

$$a_{\text{post}}(x, y) = \sum_{k=-\overline{n}}^{\overline{n}} \sum_{l=-\overline{n}}^{\overline{n}} K(k + \overline{n}, l + \overline{n}) a(x + k\delta h, y + l\delta h) \tag{4.15}$$

Remark *: This implementation of discrete convolution requires some adjustment when the kernel moved to positions where it does not fit entirely within the image, e.g. at the edges and corners. It is possible to add some "ghost" pixels in order to normally apply the kernel.*

The second choice is called the Gaussian smoothing. The changes occurs at the kernel level, where the values $(1/N^2)$ are replaced with Gaussian values. The kernel is now represented as

$$K = (G(x_i, y_i))_{(i,j)\in\{1,\dots,N\}^2}, \tag{4.16}$$

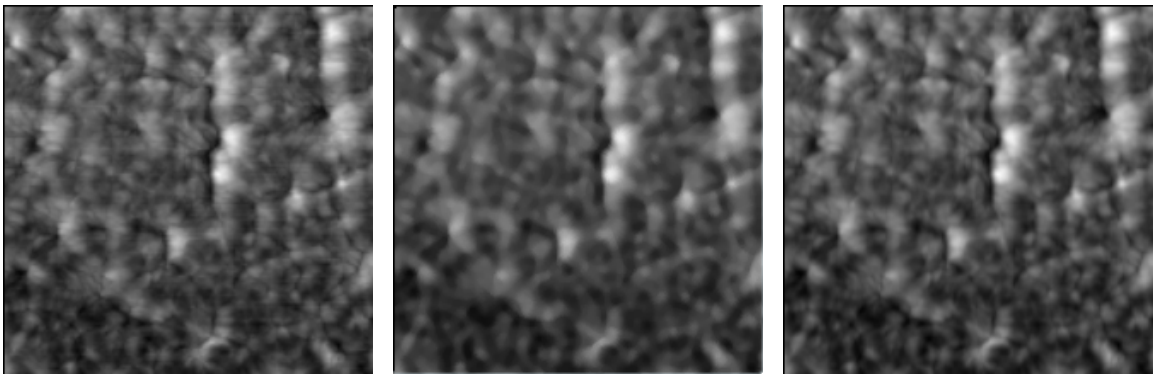
with

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \tag{4.17}$$

where σ represents the standard deviation. The advantage of this strategy, in comparison with the blur box, is the natural account of the distance between the source and the nearby values. Indeed, in this case, each pixel is still a weighted average of its neighboring values, but the weight is decreasing with the distance. The matrix 4.18 is an example of a 5×5 Gaussian kernel obtained using $\sigma = 1.0$.

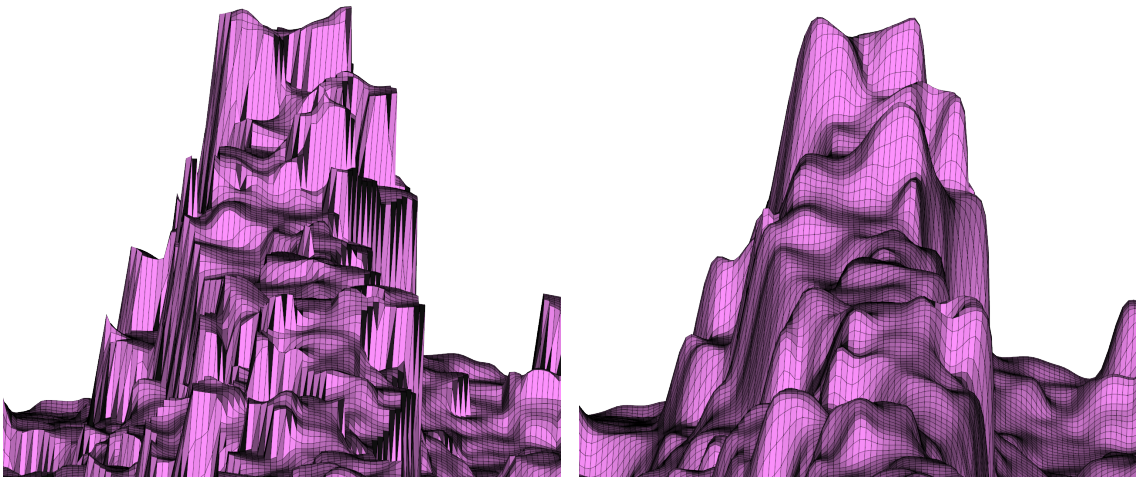
$$K = \frac{1}{273} \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}. \tag{4.18}$$

These filters have been assessed on a noisy set of points representative of PV random textures. We illustrate in figure 4.7 how Box blur strategy is way more invasive than the Gaussian smoothing. Of course, even if the blur box has succeeded in removing the unrealistic artefacts, it has also lost a large amount of geometric details, resulting in a very deep blur in figure 4.7(b). On the other hand, the Gaussian filter presents results much more consistent to the starting data. It allows us to remove the majority of defaults while keeping the texture details, as can be seen on the 3D topographic view in figure 4.8.



(a) Original image (b) Box blur applied (c) Gaussian smoothing applied

Figure 4.7 | Comparison between the box blur and the Gaussian smoothing. The original image (a) presents some horizontal strips (artifacts caused by the microscope measures), (b) shows a box blur results, using a 5×5 kernel. The resulting image is blurry, which means a loss in texture details; Gaussian smoothing (c) is much closer to the original image with an attenuation of the strips.



(a) Quadrangle mesh from AFM (b) Quadrangle mesh after Gaussian smoothing

Figure 4.8 | Textured surface pre- and post-treatment Realistic texturation of solar cell layer.

Obviously, the smoothing introduces discrepancies between the original model and the obtained surfacic mesh. It would be essential to check, as a future step, how we can control and minimize the impact of the smoothing concerning the standard deviation σ and the kernel size n. The first idea is to find a ratio between the resolution of the point cloud and the accuracy needed for the calculation.

4.3.2 Bi-periodic structures

Once the acquisition of surfacic meshes of each layer is made, we still lack the four lateral faces to have a closed domain. In particular, we have to impose Periodic Boundary Conditions (PBC) that allow to simulate artificially infinite mono- or bi-directional structures while considering only one elementary pattern. This is done by matching cells from a periodic boundary face with their neighbors on the opposite boundary of the domain. It is evident that the randomly textured nature of the geometries considered here does not naturally allow this matching to be applied.

In order to apply this principle to our realistic solar cell meshes, we explore two possibilities. The first one consists of building a mesh by gluing together four meshes obtained by symmetries. It is done in two steps, as shown in figure 4.9:

1. Considering a domain $\Omega = [x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}] \times [z_{\min}, z_{\max}]$, discretized in a mesh containing the vertices $v_i = (x_i, y_i, z_i)$ with $i \in \{1, ..., N\}$. Let P_x^+ be the plane defined by $x = x_{\max}$ and $p(v_i)$ be the orthogonal projection of v_i on P_x^+. From here, we can get a new mesh replacing the points with:

$$\widehat{v_i} = 2p(v_i) - v_i, \qquad i \in \{1, ..., N\}. \tag{4.19}$$

It should be noted that elements keep the same connectivity, which allows a very fast execution of the operation. From here we can merge the two meshes, with the only difficulty being to not duplicate the nodes on P_x^+.

2. The second step is straightforward. It consists in repeating the same operations from step 1 taking into account all the vertices of $\mathcal{T}_2 h$, discretization of $\Omega_2 = [x_{\min}, 2x_{\max} - x_{\min}] \times [y_{\min}, y_{\max}] \times [z_{\min}, z_{\max}]$, with the plane P_y^+ defined by $y = y_{\max}$.

This method presents two main drawbacks. The first and biggest one is that the obtained mesh is four times larger than the starting one, both in terms of computational domain size than in terms of elements. Secondly, one of the layers may present a hole or a bump at the edge of the domain. Joining both parts at this point can create problematic geometrical singularities, which can drastically affect the simulation results.

Remark *: One should have a look at the mesh to see if it is more interesting to use P_x^+ or P_x^- in (i), respectively P_y^+ or P_y^- in (ii).*

The second option is more intrusive and needs to be done before the volumic mesh generation. It uses a so-called Tukey window function on each textured layer to level out each edge, as proposed in [JLOZ15]. The main idea behind this procedure comes from the evident simplicity of matching rectangular faces. To obtain such rectangular faces in x and y directions, we apply the Tukey window function w to all of our point:

$$a_{tk}(x, y) = w(x) * w(y) * a_{\text{post}}, \tag{4.20}$$

In what follows, we use the function defined on $[0, 1]$ by:

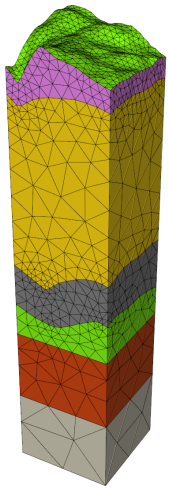
$$w(\xi) = \begin{cases} \frac{1}{2}\{1 + \cos[\pi(\frac{2\xi}{r} - 1)]\} & 0 \leq \xi \leq \frac{r}{2}, \\ 1 & \frac{r}{2} \leq \xi \leq 1 - \frac{r}{2} \\ \frac{1}{2}\{1 + \cos[\pi(\frac{2\xi}{r} - \frac{2}{r} - 1)]\} & 1 - \frac{r}{2} \leq \xi \leq 1, \end{cases}$$

with $\xi \in \{x, y\}$ and $r \in [0, 1]$ is a parameter managing how smooth the transition between the edges and the textures will be. One can see in figure 4.10, an analytical example which shows how the Tukey window works. We also show the obtained surface starting from a real dataset in figure 4.11.

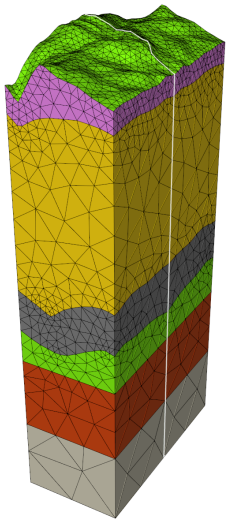
It is crucial to notice that the value chosen for r impacts the quality of the obtained mesh. Taking $r \approx 0$ will restrict the dumping area to a tiny area, creating sharp edges on the border, similar to the second drawback of the first method. On the other hand, having $r \approx 1$ will cause a loss of details on a large area of the domain. In addition to that, the flat areas located at the boundary can lead to the creation of cavity modes that should usually not exist. Once this is done, we can build a volumic mesh using a mesh generation tool, taking care to specify the need for bi-periodicity.

4.4 GaAs benchmark

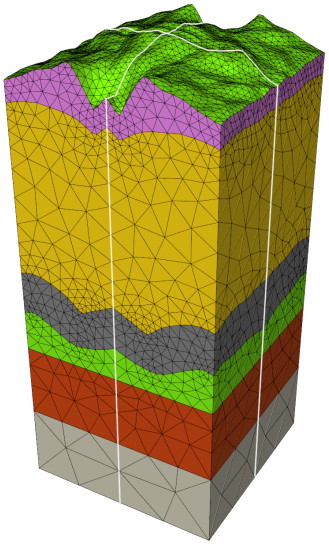
In order to assess the capabilities of the DGTD method introduced in chapter 2 for the simulation of light propagation in nanostructured solar cells, we participated in a benchmarking study proposed



(a) Starting from a given unit cell



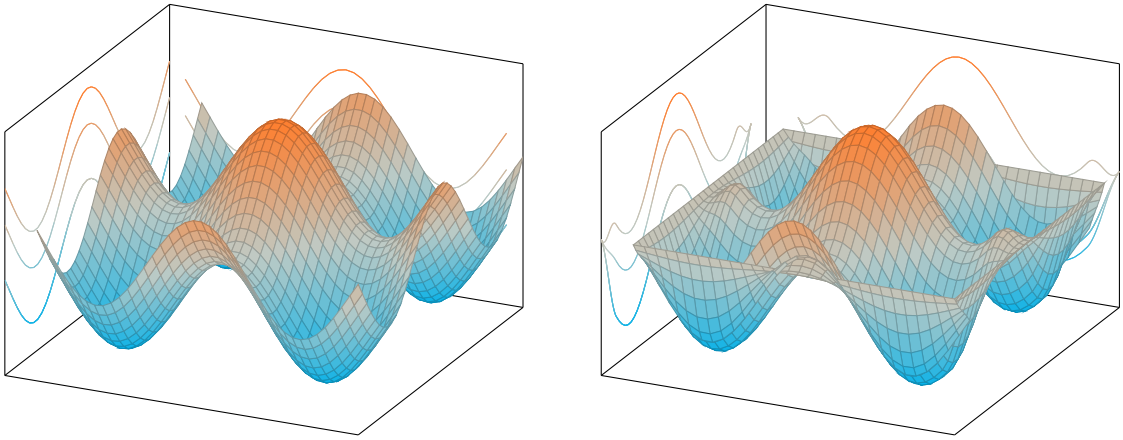
(b) Merge with its reflection symmetry from x



(c) Merge with its reflection symmetry from y

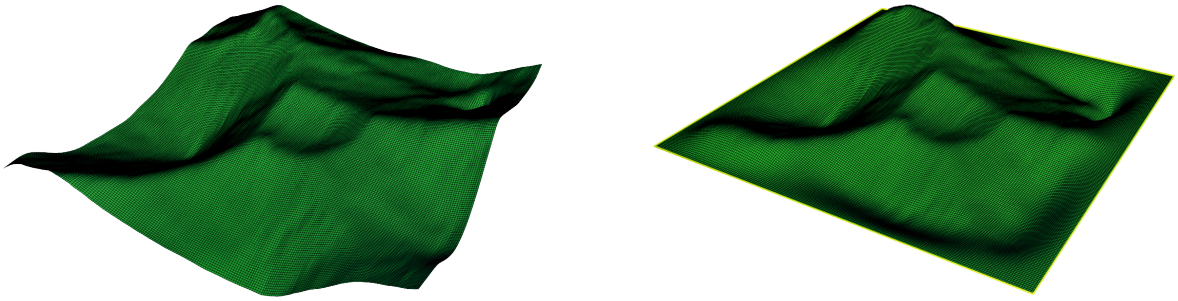
Figure 4.9 | Symmetric mesh strategy.

by Stéphane Collin at C2N. The main objective of this study is to compare the different numerical methods for the simulation of light absorption in nanostructured PV cells. Except for the DGTD method which is provided by us, the three other methods are the ones presented in section 4.1. For this purpose, a series of solar cell configurations based on Gallium Arsenide (GaAs) material of increased geometrical complexity has been specifically devised for this study. Solar cells considered are constituted of multiple slabs, and the silverback reflector is nanotextured with a non-symmetric pyramidal grating. The horizontal size of the elementary pattern is 600×600 nm^2. The solar cell is illuminated from above and a semi-infinite slab of silver is used at the bottom. PMLs are used



(a) Without periodicity (b) With periodicity

Figure 4.10 | Example of Tukey-window approach. As one can see in (b), the edges are all set-up to the mean value of $a(x, y)$.



(a) Original layer (b) Layer with PBC after applying Tukey window

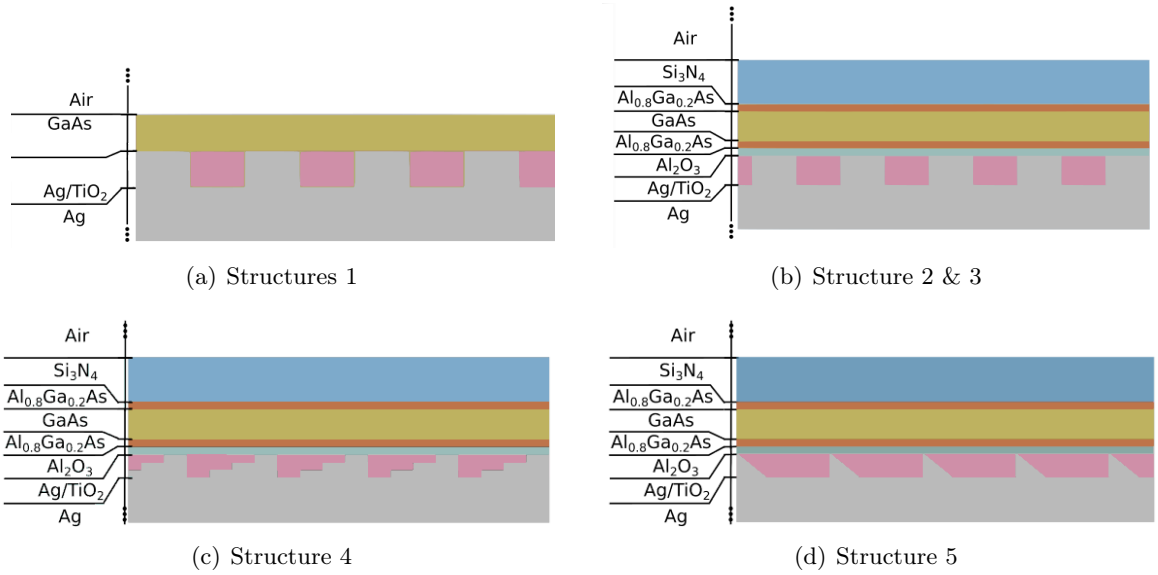
Figure 4.11 | Example of Tukey-window approach on a realistic textured layer with $r = 0.3$.

upstream of the injection surface to reduce parasitic reflection. Periodicity is imposed in x and y directions as we consider only one elementary pattern.

In order to highlight the difficulties encountered in simulating such a physical problem, five configurations have been defined. The first structure, referred to as Structure 1 in the following, does not include the anti-reflective coating layers at the top of the cell, and a rectangular block replaces the pyramid. Structure 2 includes an anti-reflective coating. Structure 3 is replacing the stud with a Cartesian staircase version of the pattern. This case triggers the breaking of the symmetry. Structure 4 introduces the invariance of the pyramid along the propagation direction z. Finally, Structure 5 uses the pyramid as the texturing pattern. The composition and the thicknesses of each structures is presented in figure 4.12 and table 4.1.

The geometries and the meshing part are both done with the Gmsh meshing software [GR09][2], which now includes a CAD engine. The meshes of Structure 1 to Structure 5 are represented in figure 4.13. Table 4.2 gives the characteristics of the tetrahedral meshes used in the numerical simulations.

[2]http://gmsh.info/



(a) Structures 1

(b) Structure 2 & 3

(c) Structure 4

(d) Structure 5

Figure 4.12 | Composition of each structure.

Material	Structure 1	Structure 2 & 3	Structure 4	Structure 5
Air	semi ∞	semi ∞	semi ∞	semi ∞
Si_3N_4	0	60	60	60
AlGaAs	0	10	10	10
GaAs	100	100	100	100
AlGaAs	0	10	10	10
Al_2O_3	0	10	10	10
TiO_2 (h)	100	100	100	210
Ag	semi ∞	semi ∞	semi ∞	semi ∞

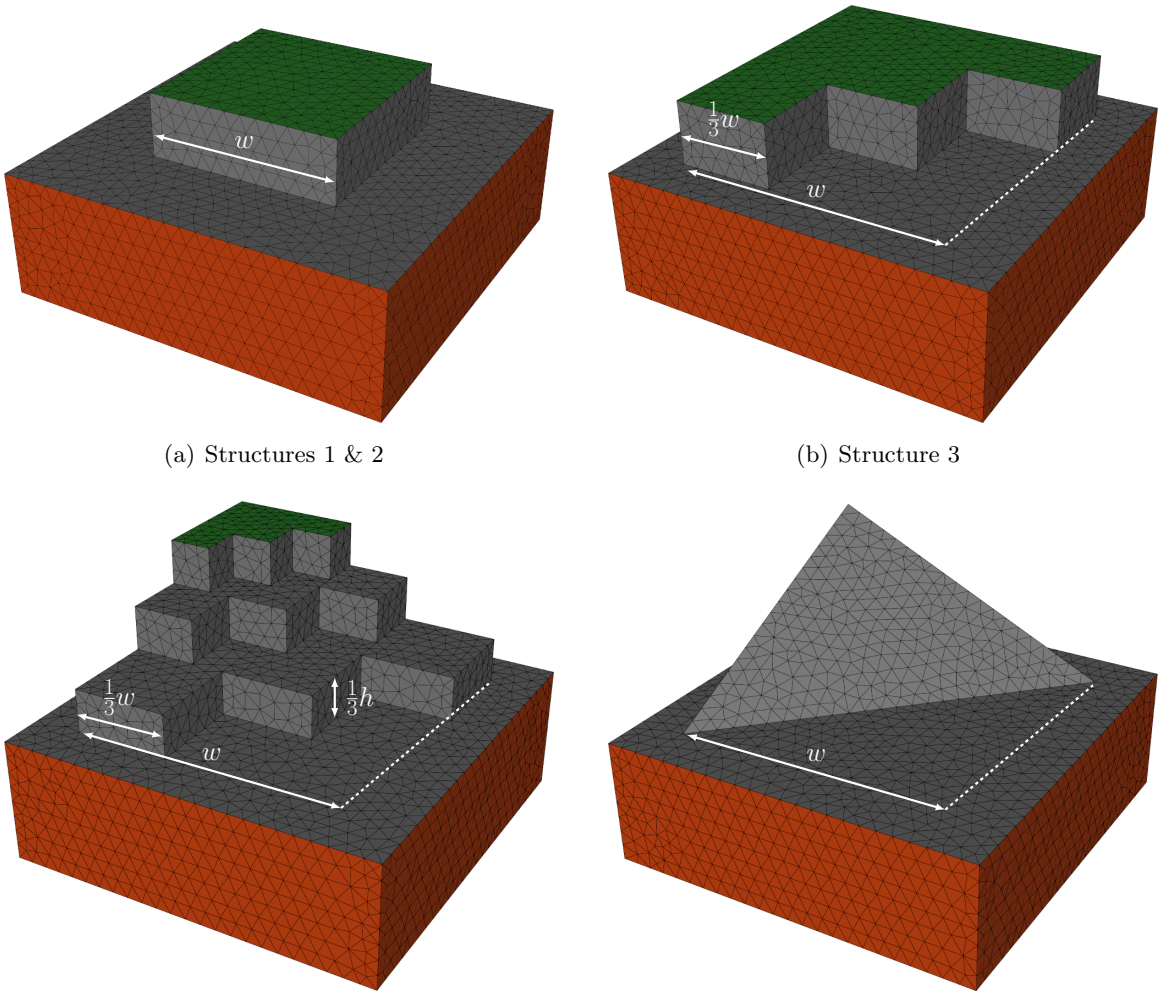
Table 4.1 | Thicknesses of the different materials for Structure 1 to 5 in nanometers.

Mesh	Tetrahedron	$h_{\min}$ (nm)	$h_{\max}$ (nm)	$\frac{h_{\max}}{h_{\min}}$
Structure 1 (very coarse)	314	100.0	327.7	3.28
Structure 1 (coarse)	3638	31.5	164.5	5.23
Structure 1 (fine)	75135	13.9	57.2	4.11
Structure 2	95052	10.0	54.4	5.43
Structure 3	95062	10.0	54.8	5.48
Structure 4	108102	10.0	57.1	5.71
Structure 5	106656	10.0	79.0	7.91

Table 4.2 | Characteristics of tetrahedral meshes of Structures 1 to 5.

4.4.1 Material models

The optical properties of the different materials that constitute the considered device structure have been fitted to the parameters of the generalized dispersion model [Viq18] presented in chapter 2. Different numbers of first-order poles and second-order poles have been tested in order to obtain



(a) Structures 1 & 2

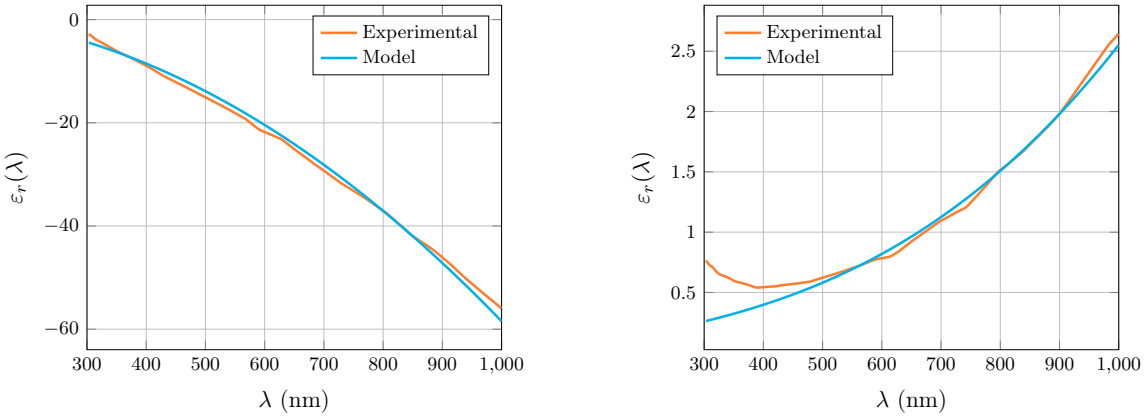
(b) Structure 3

(c) Structure 4

(d) Structure 5

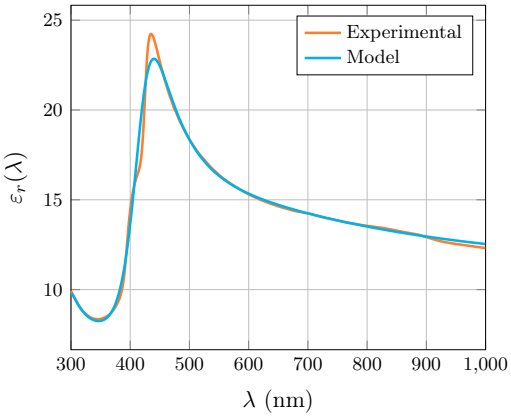
Figure 4.13 | Mesh of the textured layer with a complexification: (a) corresponds to the symmetric pattern ($w = 300$ nm), (b) is breaking this symmetry ($w = 450$ nm), (c) is considering an evolution of the pattern in the z direction ($w = 450$ nm), and (d) is considering a pyramid ($w = 450$ nm). The red surface corresponds to PBCs, the gray is the interface between the Ag nanostructure and the substrate, and the green is the interface between the GaAs and the Ag.

the best possible results. The fitted models obtained are ranked according to their error on the real part of the permittivity and on the imaginary part of the permittivity. From there, we choose a compromise between the number of poles and the accuracy of the model. Indeed, the addition of a pole is equivalent to adding an ODE (or two for a second-order pole) to the Maxwell-GDM equations. The obtained permittivity functions are plotted in figure 4.14. As can be seen, they are relatively well approximated in the wavelength range $\lambda = [300, 1000]$ nm. Note that the SiO_2 is here modelled with a constant permittivity and without losses, *i.e* $\varepsilon_r = 2.1025$ and $\varepsilon_i = 0$.

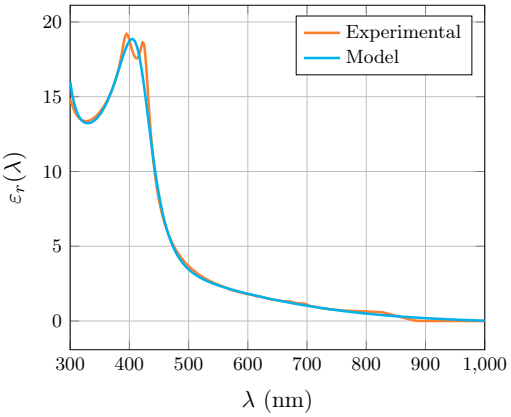


(a) Ag, real part

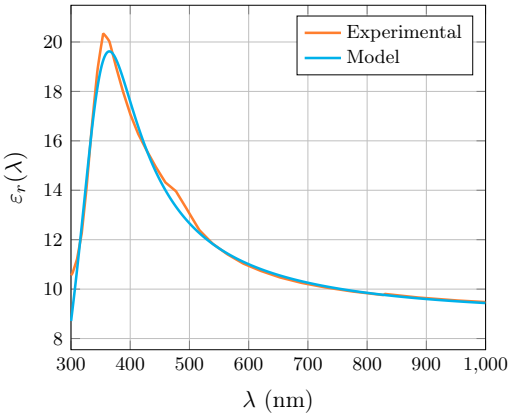
(b) Ag, imaginary part



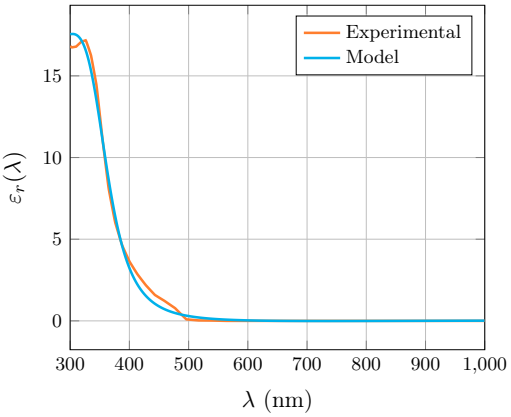
(c) GaAs, real part



(d) GaAs, imaginary part



(e) AlGaAs, real part



(f) AlGaAs, imaginary part

Figure 4.14 | Real and imaginary parts of the relative permittivity of Ag, GaAs and AlGaAs predicted by our dispersive model compared to experimental data.

4.4.2 Numerical convergence

Numerical convergence has been assessed for Structure 1 only, although in all cases we performed simulations with the DGTD method based on $\mathbb{P}_4$ interpolation in addition to $\mathbb{P}_3$ ones to ensure

there were no major variations in the obtained results. As an example, we plot in figure 4.15 the total absorption obtained for Structure 1, as a function of the polynomial order. As can be seen, $\mathbb{P}_3$ and $\mathbb{P}_4$ results are very close, except for some small discrepancies in the high wavelength region of the spectrum. For the sake of completeness, we also compare the spectra obtained using the three meshes shown in figure 4.16 for this structure. Results are shown in figure 4.17. As one can see, the total absorption is relatively well computed even on the coarse mesh.

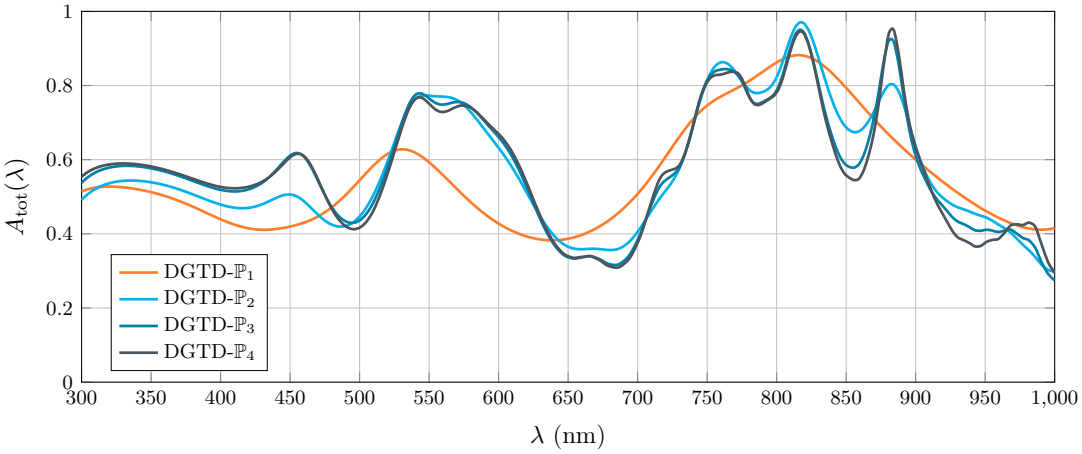
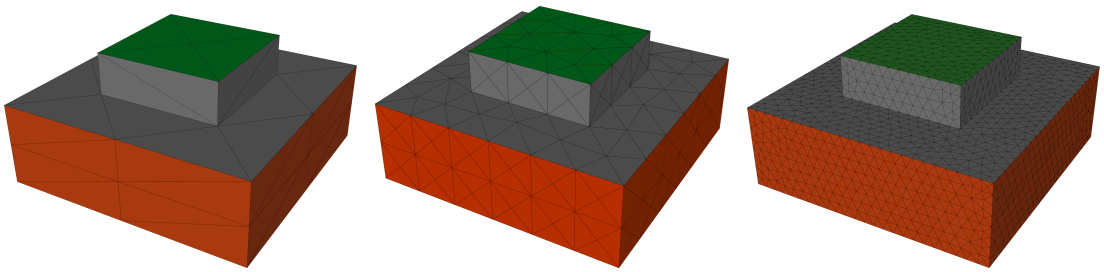


Figure 4.15 | A_{tot} for Structure 1 using coarse mesh with polynomial orders ranging from 1 to 4.



(a) Very coarse (b) Coarse (c) Fine

Figure 4.16 | Hierarchy of tetrahedral meshes for Structure 1.

In a second time, the same study has been conducted for the volumetric absorption in the GaAs. In figure 4.18, the convergence when increasing the polynomial order is shown. As one can see, the convergence of the volumetric absorption needs to use much higher polynomial order in order to converge, compared to the total absorption. Those results have a physical explanation. The volumetric absorption is a quantity computed inside the elements (via volumetric integration), which makes this observable a "near field" quantity, while on the other hand, the total absorption (as well as the transmission and the reflection) is computed on a surface (via surfacic integration) "far" from the nanotextures. This makes the variations in the near field much less impacting. In figure 4.19, we look at the convergence when refining the mesh. The results are well converged, even for the coarse mesh. The results on the very coarse mesh are even better than the results on the coarse mesh with a lower interpolation order.

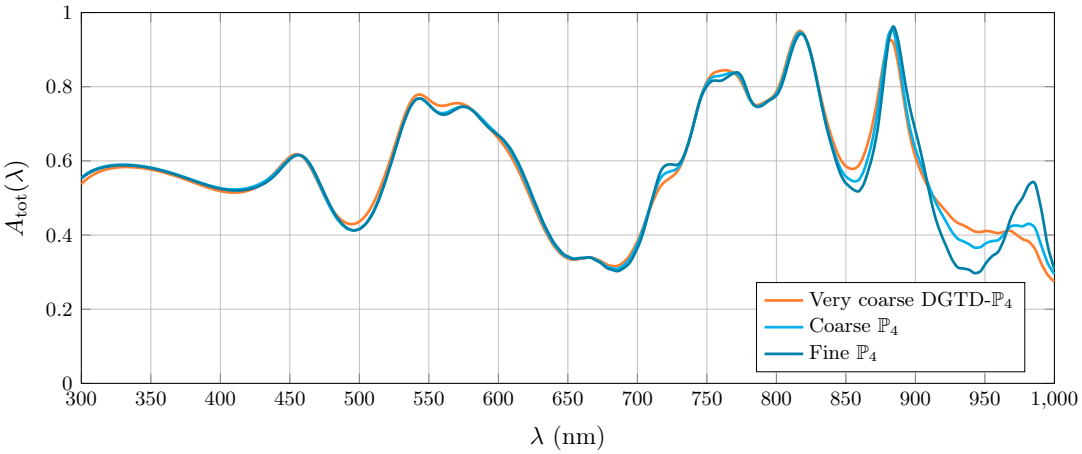


Figure 4.17 | A_{tot} for Structure 1 using three gradually refined meshes with $\mathbb{P}_4$ interpolation.

Remark *: The volumetric absorption is, by nature, more expensive to evaluate than the total absorption. This is even accentuated with the need for higher polynomial order.*

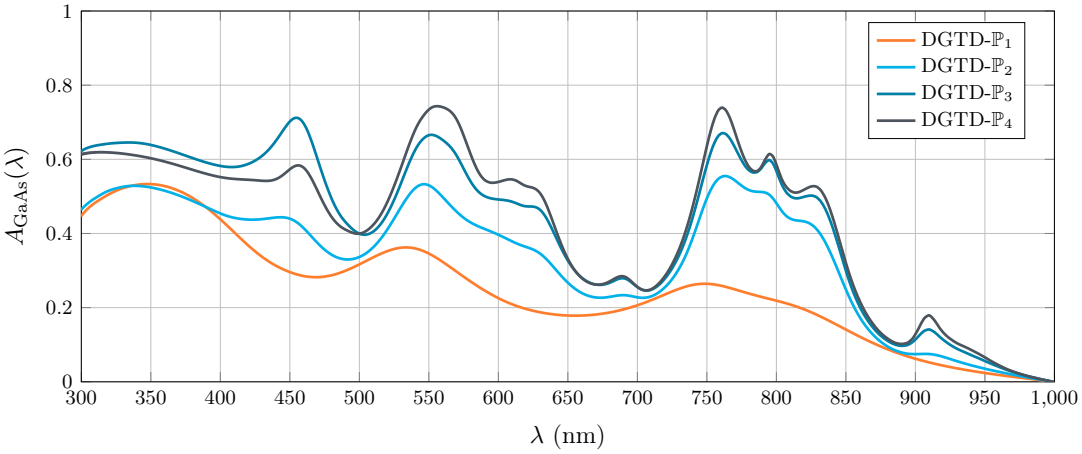


Figure 4.18 | A_{GaAs} for Structure 1 using coarse mesh with polynomial orders ranging from 1 to 4.

4.4.3 Results

All simulations have been run on 128 cores (16 Intel E5-2670 2.6 GHz processors, each with 8 cores). In figure 4.20, we plot A_{tot} obtained for the DGTD method based on $\mathbb{P}_3$ and $\mathbb{P}_4$ interpolations using a coarse mesh. As one can see, A_{tot} is quasi-converged for $\mathbb{P}_3$ on every structures. The same observation is made for the total absorptions in each of the structures, only a few resonances located near the GaAs bandgap, *i.e.* after 800 nm, are still presenting discrepancies. Regarding physics, a considerable improvement is observed between Structure 1 and the 4 others (from 300 nm to 700 nm). This corresponds to the adding of the anti-reflective coating. Structure 2 presents some absorption gaps in the upper part of the spectrum, which are gradually filled by breaking the symmetry (Structure 3) and adding a z-invariance in the nanostructure (Structure 4).

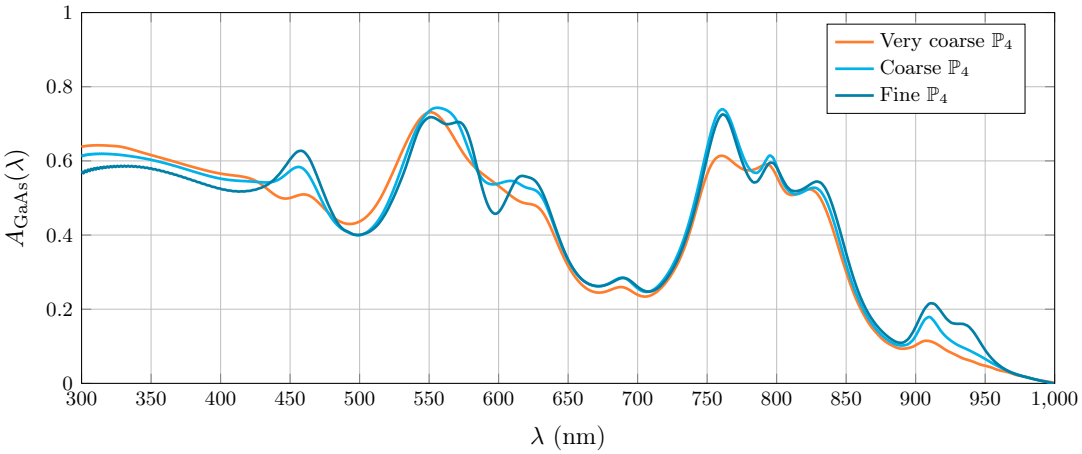
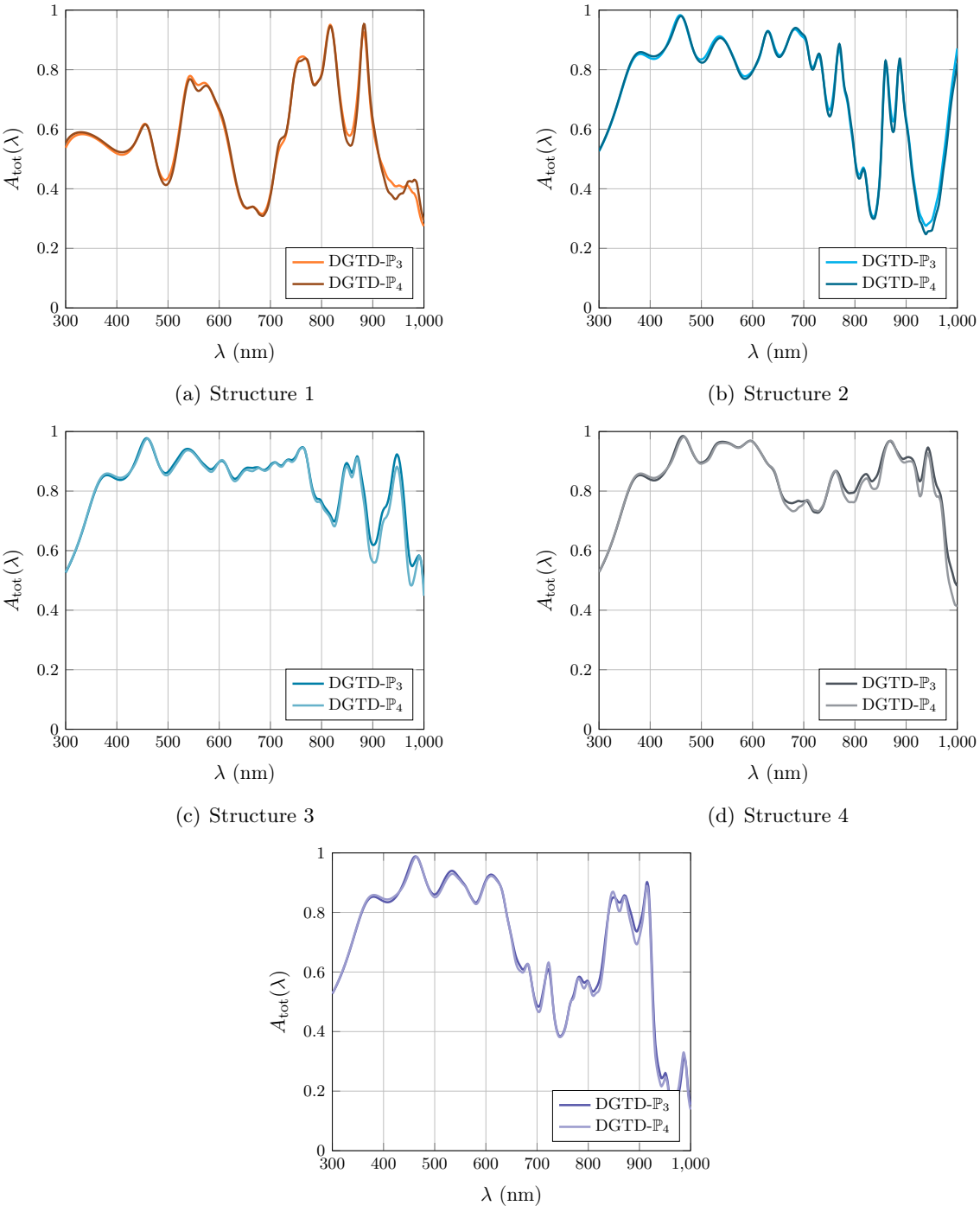


Figure 4.19 | A_{GaAS} for Structure 1 using three gradually refined meshes with $\mathbb{P}_4$ interpolation.

In figure 4.21, we plot A for each material, obtained for the DGTD method based on $\mathbb{P}_4$ interpolation using a fine mesh. The important physical quantity here is the absorption in the GaAs. As one can see, the same observation as for the total absorption can be made. However, we seem to see a decrease in the absorption spectrum of the GaAs between Structure 3 and Structure 4, but an increase of the absorption of the Ag. This is not the desired behavior, and in this case, we see that the volumetric absorption allows us to determine it, which was hidden by the total absorption.

Finally in figure 4.22, we compare our result on the total absorption with the other selected numerical methods, *i.e.* RCWA (Reticolo), FDTD (Lumerical FDTD) and FEFD. This study is still ongoing, which explains the absence of some methods for a few structures. The results observed are generally close. Differences are mainly focused on the upper part of the spectrum, the one that raised convergence problems for our method. The biggest differences are found in structure 5, where the two time-domain methods (FDTD and DGTD) are very close in comparison to the FEM. For this structure, RCWA results were not available due to the complexity of the nanostructure.



(a) Structure 1

(b) Structure 2

(c) Structure 3

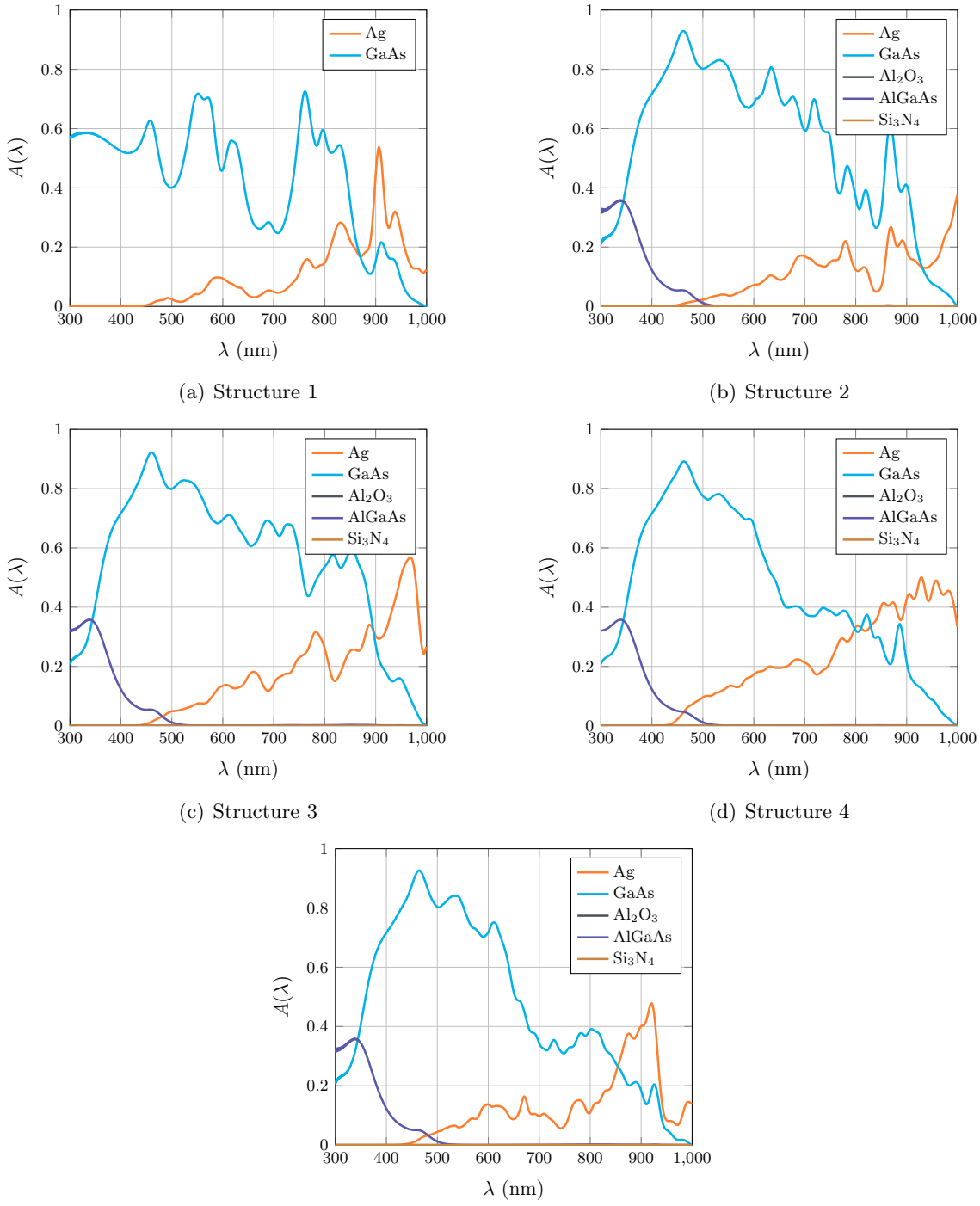
(d) Structure 4

(e) Structure 5

Figure 4.20 | A_{tot} for the different structures with $\mathbb{P}_3$ and $\mathbb{P}_4$ interpolations on the coarse mesh.

4.5 Conical texturation

In this section, we study two solar cell designs, which have been proposed in recent years by researchers in the field. These solar cell devices exploit gratings with nanocones in order to increase



(a) Structure 1

(b) Structure 2

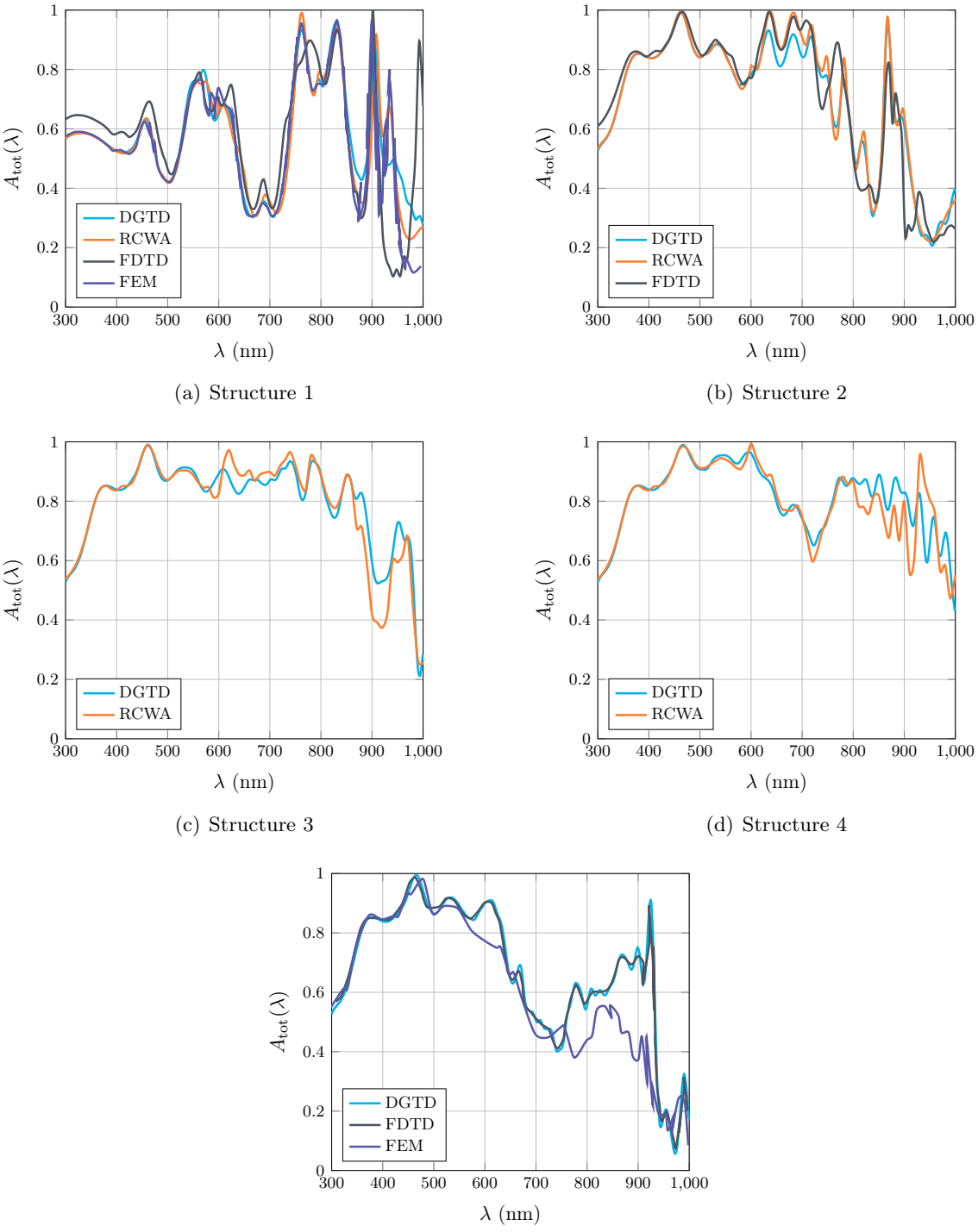
(c) Structure 3

(d) Structure 4

(e) Structure 5

Figure 4.21 | A for all layers in the 5 different structures obtained using DGTD-$\mathbb{P}_4$ on a fine mesh.

their light trapping capability. The first one is a double grating presents in [WYL+12]. The second one is a conical nanowire presents in [EJD14]. We use our high-order DGTD method, and the obtained results are also compared with results provided by the most widely used numerical



(a) Structure 1

(b) Structure 2

(c) Structure 3

(d) Structure 4

(e) Structure 5

Figure 4.22 | Comparison of DGTD, FDTD and RCWA methods for the computation of the total absorption in the 5 structures.

methods in the field.

4.5.1 Double grating

There is a significant recent interest in designing ultra-thin crystalline silicon solar cells with an active layer thickness of a few micrometers. Efficient light absorption in such thin-films requires both broadband anti-reflection coatings and effective light trapping techniques, which often have different design considerations. In [WYL$^+$12], the authors show that by employing a double-sided grating design, one can separately optimize the geometries for anti-reflection thanks to the top conical grating and also light trapping purposes, allowing broadband light absorption enhancement, thanks to the bottom conical grating. In the present study, we try to reproduce this test case. This structure contains a crystalline silicon thin-film with two silicon nanocone gratings. The nanocones form two-dimensional square lattices on both the front and the back surfaces. The film is placed on a perfect electric conductor (PEC) mirror. The design is introduced in [WYL$^+$12] as an optimal configuration, for a fixed quantity of absorber (corresponding to an equivalent flat layer of 2 μm c-Si). In the latter, the optimization is done using a RCWA solver with the aim of maximizing the Jsc.

4.5.1.1 Mesh

One can see in figure 4.23 the design of the cell. The two gratings are placed on each side of a large silicon layer of 1736 nm. The heterogeneity of element sizes can be seen directly. This is mostly explained by the geometric singularity at the top of the cone. In order to reduce this phenomenon, we choose to smooth out the extremity of all cones. Since we change the topology of the device, we readjust the equivalent volume of Si by increasing the size of the central layer according to the amount of volume truncated. As long as the cones are meshed with linear elements, Gmsh can be used.

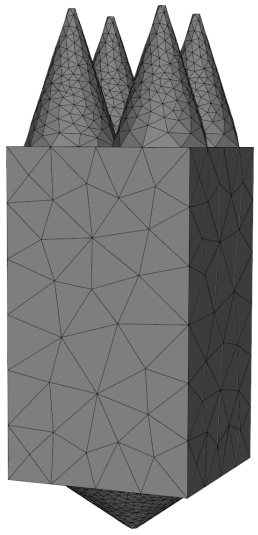


Figure 4.23 | Crystalline silicon structure with the cones extremity truncated.

4.5.1.2 Numerical convergence

We first conduct a series of simulations for the optimized structure obtained in [WYL$^+$12]. For the top nanocones, the period is 500 nm, the base radius is 250 nm, and the height is 710 nm. For the

bottom nanocones, the period is 1000 nm, the base radius is 475 nm, and the height is 330 nm. Two tetrahedral meshes have been constructed using the Gmsh software. Mesh M1 consists of 34,264 cells. The minimal, maximal and average length of a mesh edge in this mesh is respectively equal to 0.011 μm, 0.210 μm and 0.101 μm. Mesh M2 consists of 147,508 cells. The minimal, maximal, and average length of a mesh edge in this mesh is respectively equal to 0.0014 μm, 0.110 μm, and 0.064 μm. Results of simulations with mesh M1 and using the DGTD method with interpolation order 1 to 4 are shown in figure 4.24 (absorption spectra) and 4.27 (contour lines of the amplitude of the DFT of $\mathbf{E}$ at a particular wavelength). These results shed light on a potential difficulty that one may face when assessing the numerical convergence of simulations of light trapping in solar cells, *i.e.* the convergence of near field quantities.

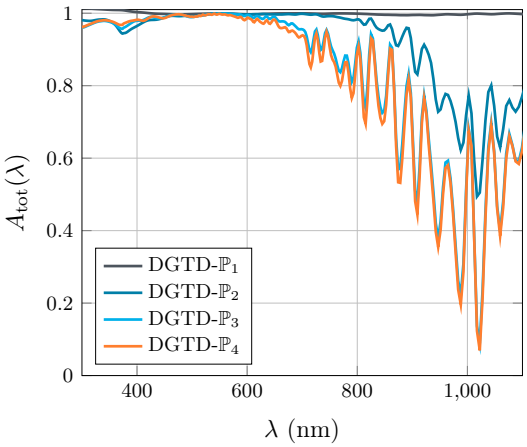


Figure 4.24 | Simulation using mesh M1: absorption spectra.

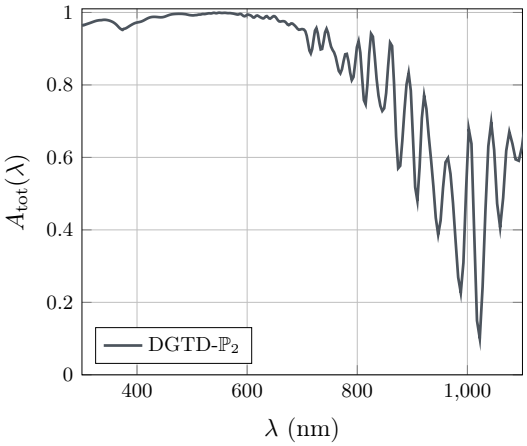


Figure 4.25 | Simulation using mesh M2: absorption spectrum.

4.5.1.3 Results

In many works, the quantity of interest (QoI) on which the numerical convergence analysis is based is the total absorption A_{tot}. When the FDTD method is used, the convergence is assessed by refining the underlying cartesian grid discretizing the solar cell structure. In the case of the

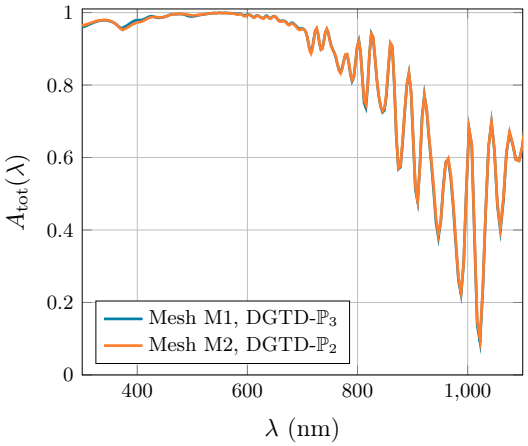
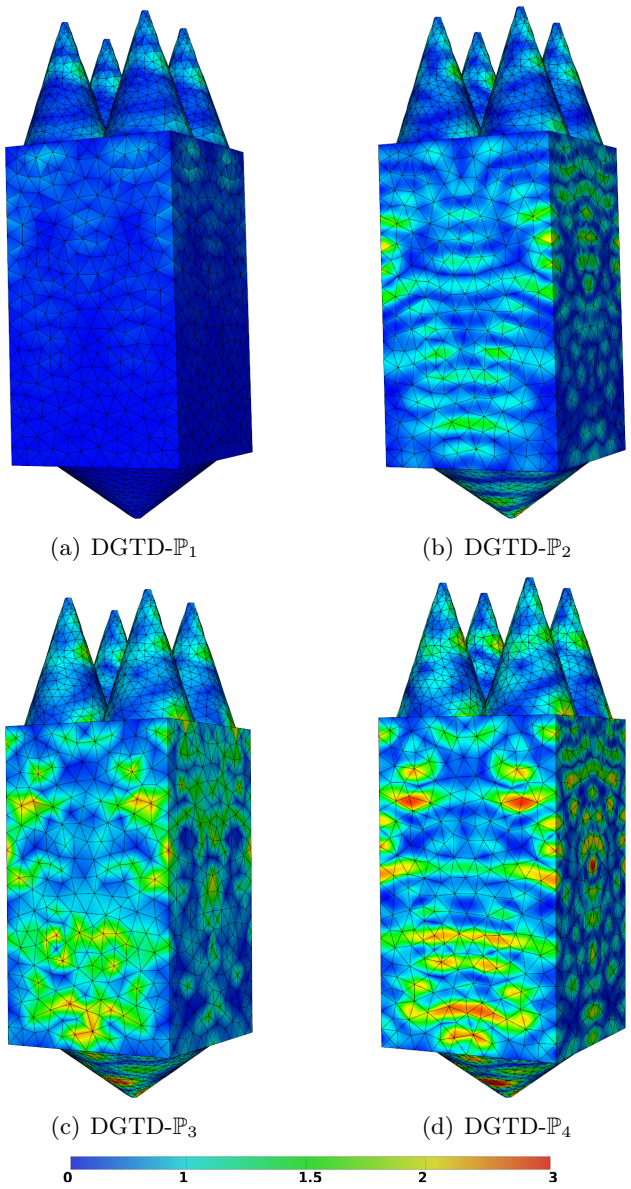


Figure 4.26 | Comparaison of results using meshes M1 and M2: absorption spectra.

DGTD method, two parameters can be considered separately or together to monitor the accuracy of a numerical solution: the discretization step referred to as h and taken as the maximum of the cell-wise sizes; the order of the polynomial interpolation of the components of the electromagnetic field within each cell referred to as p (we assume here, a uniform interpolation order). In figure 4.24, we fix the mesh and increase the interpolation order. We observe that the DGTD-$\mathbb{P}_3$ yields a converged solution using mesh M1 (DGTD-$\mathbb{P}_3$ and DGTD-$\mathbb{P}_4$ solutions are indistinguishable). However, the contour lines of the amplitude of the DFT of $\mathbf{E}$ in figure 4.27 clearly lead to a different picture as the DGTD-$\mathbb{P}_4$ solution shows patterns that are not well resolved when using the DGTD-$\mathbb{P}_3$ method. Now, we perform a single simulation with the DGTD-$\mathbb{P}_2$ method using mesh M2, see figures 4.25 and 4.28. Whereas the obtained absorption spectrum is almost identical to the one obtained with the DGTD-$\mathbb{P}_3$ method using mesh M1 (see figure 4.26), here again, we observe that the volumic field is more sensitive to the refinement of the mesh than the total absorption.

We conclude this series of results by assessing the influence of the physical simulation time t by comparing the absorption spectra obtained with the DGTD-$\mathbb{P}_2$ method using mesh M2, for different values of T (note that $T = 2.0 \cdot 10^{-13}$ seconds has been used in all the previously discussed simulations). In order to have a more generic stopping criterion, we numerically evaluate the residual energy present in the system. This energy $\mathcal{E}$ is calculated every n time-steps, $n \in \mathbb{N}^*$. Once the incident field has been fully injected, the maximum energy is recorded. In the course of the simulation, the energy in the system decreases to zero. We define the threshold for the absorption computation, which corresponds to a certain percentage of $\max_{n>0} \mathcal{E}$. For example, we usually used 0.01% as a suitable starting parameter. It can be noted that decreasing this threshold is equivalent to increasing T. The obtained absorption spectra for different values of the threshold are shown in figure 4.29. The simulation physical and computational times corresponding to the different energy levels are shown in table 4.3

We also distinguish a very different behavior between the small wavelengths and the large wavelengths. Even for the largest threshold (1%), the absorption for the lowest wavelength part is already converged. For the upper part of the spectrum, we observe stronger resonances that are still appearing when the threshold is strongly reduced (0.001% or a 200 longer physical time). In figure 4.30, we show the difference between two consecutive absorption spectra, which further confirms the difference in the behavior of the two parts of the full spectrum. This can be explained by the large extent of the spectrum of interest (200 nm to 1200 nm) that complicates the setup of



(a) DGTD-$\mathbb{P}_1$ (b) DGTD-$\mathbb{P}_2$

(c) DGTD-$\mathbb{P}_3$ (d) DGTD-$\mathbb{P}_4$

Figure 4.27 | Simulation of light trapping in a solar cell based on nanocone gratings. Simulations using mesh M1: contour lines of the module of the DFT of $\mathbf{E}$ for a wavelength $\lambda = 857$ nm.

the simulation. Indeed, it is well known that a good solution to wave problems requires a certain number of points per wavelength. However, for a time-domain solver, multiple wavelengths are handled synchronously. Here, the ratio $\lambda_{\max}/\lambda_{\min}$ is 6, which means that we should have roughly six times more degree of freedom for the lower part than for the upper one.

Therefore, we have taken a "frequency-domain" approach in order to reduce the overall computational time. We choose to split the simulation into two parts, one dealing with the small wavelengths, *i.e.* with a refined mesh, and high interpolation order, while the other can run on a coarser mesh with a lower interpolation order. Indeed, the physical simulation time required to capture the resonances of the upper part of the spectrum will remain the same, *i.e.* a threshold

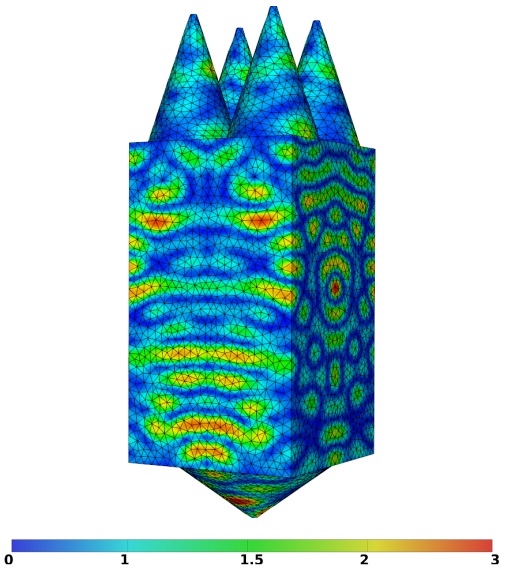


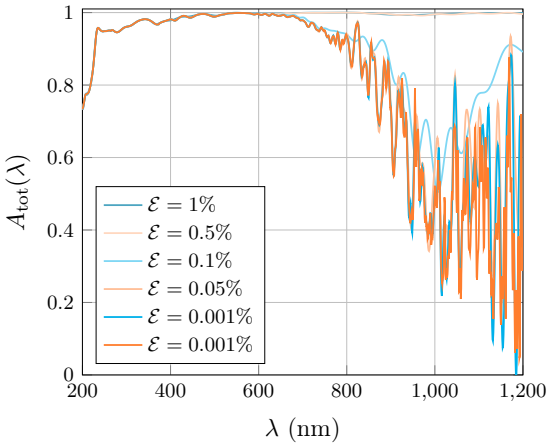
Figure 4.28 | Simulation using mesh M2 and the DGTD-$\mathbb{P}_2$ method: contour lines of the module of the DFT of $\mathbf{E}$ for a wavelength $\lambda = 857$ nm.

$\mathbb{P}$	Energy level	max $\lvert A_{i+1} - A_i \rvert$	Physical time T	Computation time	#Iterations
$\mathbb{P}_3 - \mathbb{P}_4$	5%	/	3.21e-14	2 h 10 mn	2600
	2%	1.28e-02	4.52e-14	3 h 03 mn	3660
	1%	1.50e-03	6.02e-14	4 h 04 mn	4880
	0.5%	3.00e-03	1.00e-13	6 h 45 mn	8100
$\mathbb{P}_2 - \mathbb{P}_3$	5%	/	3.82e-13	7 h 55 mn	28500
	2%	5.22e-01 (4.28e-01)	5.23e-13	10 h 50 mn	39000
	1%	3.36e-01 (5.82e-02)	6.52e-13	13 h 30 mn	48600
	0.5%	1.79e-01 (3.71e-02)	7.93e-13	16 h 25 mn	59100
	0.1%	3.94e-01 (4.92e-02)	1.20e-12	24 h 55 mn	89700
	0.05%	1.76e-01 (1.01e-02)	1.38e-12	28 h 32 mn	102720
	0.01%	1.90e-01 (8.25e-03)	1.93e-12	40 h 03 mn	144180
	0.005%	1.11e-01 (1.90e-03)	2.31e-12	47 h 56 mn	172560
	0.001%	8.69e-02 (1.46e-03)	3.10e-12	64 h 12 mn	231120

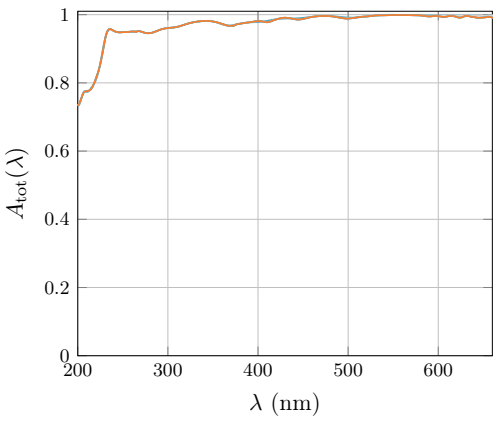
Table 4.3 | Correspondance between the energy levels and the physical time for both spectrum area. The error in the parenthesis is computed on absorption after the moving average.

lower than 0.005%. However, being able to use a much coarser mesh size, with a lower interpolation order will drastically decrease the computational load as well as increase the stable time-step. Also, in figures 4.31 and 4.32, we check the coupling between the absorption spectrum of the two simulations, as well as the concordance with the full run on M2. In figure 4.31, we superpose our obtained absorption spectrum with the single-pass absorption and the Yablonovitch limit, respectively given by $A_{\text{Single-pass}} = 1 - e^{-\alpha d}$ and $A_{\text{Yablonovitch}} = 1 - \frac{1}{1+4n^2\alpha d}$.

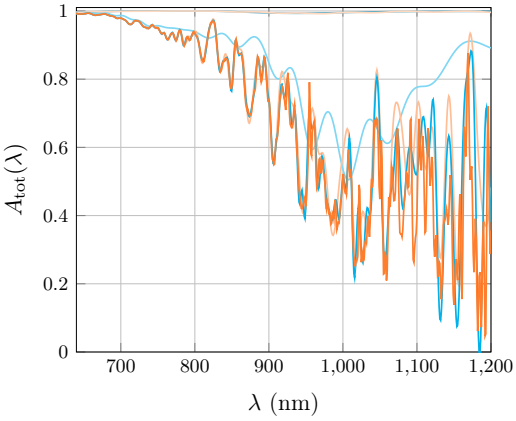
Finally, we have shown that the use of a time-domain method for "too much" broadband problems can pose complicated numerical problems. The use of a frequency-based strategy allows the simulation to be split into two parts, each of them being much cheaper in terms of computation



(a) Full spectrum



(b) From 200nm to 600nm



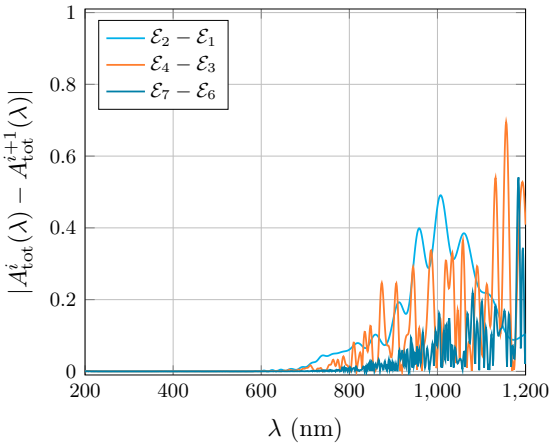
(c) From 600nm to 1200nm

Figure 4.29 | Computation of the total absorption for different values of the threshold on the residual energy. We clearly see that in the lower part of the spectrum, the absorption converges faster than in the upper part.

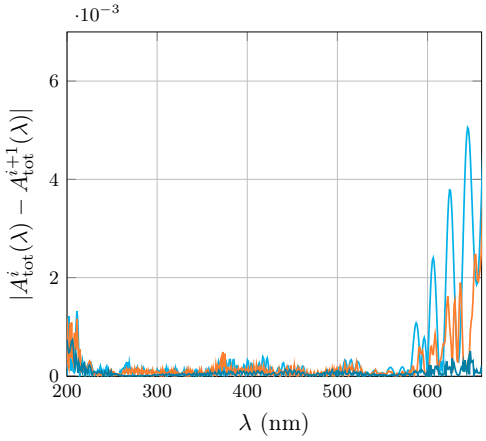
cost. The overlapping between the absorption spectrum is almost perfect.

4.5.2 Conical nanowire

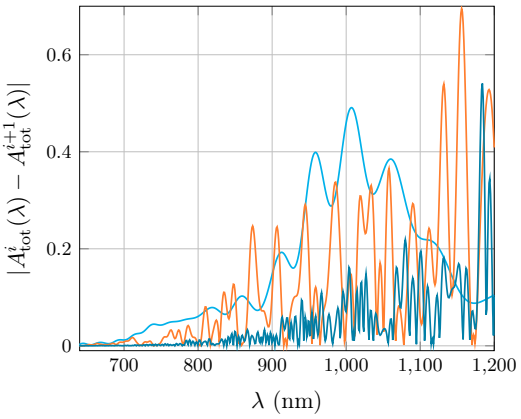
In [EJD14] the authors study light trapping in Gallium Arsenide (GaAs) solar cells based on thin-film photonic crystals using a FDTD method. Indeed, light trapping in thin-film photonic crystals is capable of surpassing the performance of conventional thick solar cells. This is made possible by a combination of improved solar absorption and more effective carrier collection on length scales smaller than the carrier diffusion length. Thus, thin-film photonic crystal active layers provide a unique opportunity for both photonic and electronic management in solar cells. The authors present designs that enable a significant increase in solar absorption in ultra-thin layers of GaAs. In the wavelength range [400 nm, 860 nm], 90–99.5% solar absorption is demonstrated depending on the photonic crystal architecture used and the nature of the packaging. It is shown that using only 200 nm equivalent bulk thickness of GaAs, forming a conical-pore photonic crystal (lattice constant 550 nm, pore diameter 600 nm and pore depth 290 nm) packaged with SiO2 and



(a) Full spectrum



(b) From 200nm to 600nm



(c) From 600nm to 1200nm

Figure 4.30 | Differences between two consecutive outputs of the absorption. One can see the two behaviors of the error in those graphs. In (a), three consecutive errors are plotted on the entire spectrum. In (b) and (c), we rescale the errors for the intervals [200 nm,650 nm] and [650 nm,1200 nm], respectively. There is three levels of magnitude between each.

deposited on a silver back-reflector, one can achieve 90% absorption of all available solar light in the wavelength range [400 nm, 860 nm]. We consider here the optimized setting studied by the authors, which is characterized by a 200 nm equivalent bulk thickness of GaAs (which represents a height of 4770 nm), a cone base radius of 100 nm, and a lattice constant of 500 nm. Such geometry is complicated to represent with structured mesh. In this part, we want to show the advantages of our DGTD method for simulating light absorption in such PV cell. One mesh of the structure is shown in figure 4.33.

4.5.2.1 Numerical convergence

The device geometry presents real challenges. Indeed, the combination of a high height ($h = 4.77\ \mu$m) and a small basis radius ($r = 100$ nm) for the GaAs cone requires the use of very

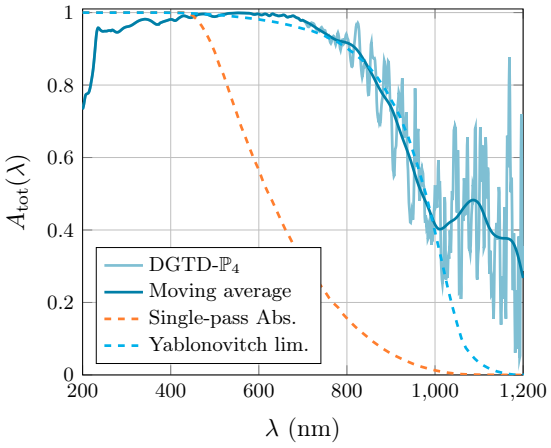
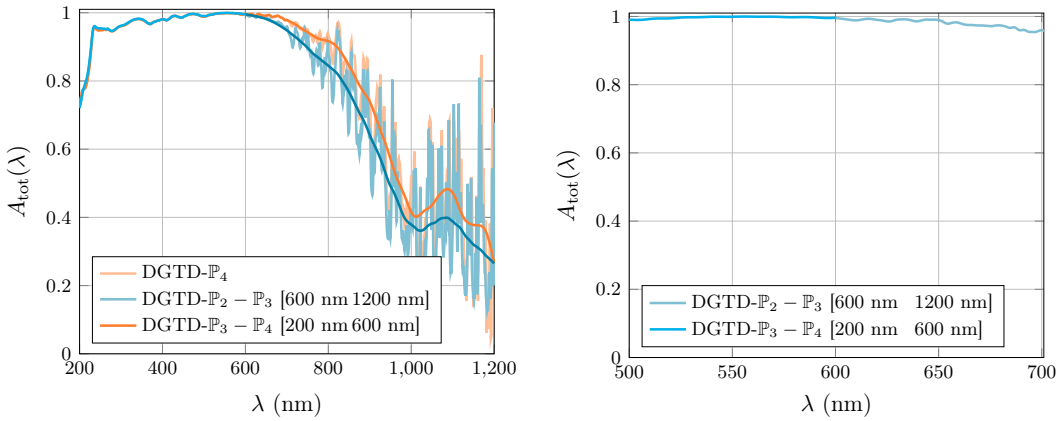


Figure 4.31 | Comparison of the obtained absorption with theoretical limits. A moving average with a window of 3 points is computed for easier readability. Theoretical limits are computed using optical parameters from literature.



(a) Full spectrum

(b) Zoom on the coupling area

Figure 4.32 | Absorption obtained by two simulations and compared with the single run solution. One can see in (b) that the connection of the two simulations is almost perfect.

small element at the top ($h_{\min} = 2$nm). Mainly due to meshing issues, we choose to simplify the extremity of the cone by truncating it. (see figure 4.34).

The convergence was tested on both fine and coarse meshes. The characteristics regarding the meshes are in table 4.4. We plot in figure 4.35 and 4.36 the total absorption spectrum obtained for polynomial orders going from 1 to 4 for the coarse and the fine mesh, respectively. As can be seen, for A_{tot}, $\mathbb{P}_3$ and $\mathbb{P}_4$ simulations are really close.

Mesh	Tetrahedron	$h_{\min}$ (nm)	$h_{\max}$ (nm)	$\frac{h_{\max}}{h_{\min}}$
Coarse	8914	3.86	500	129.53
Fine	57883	2.21	200	90.50

Table 4.4 | Characteristics of tetrahedral meshes of the conical nanowire .

Figure 4.33 | mesh of the GaAs cone with the injection region at the top and the PEC boundary at the bottom.

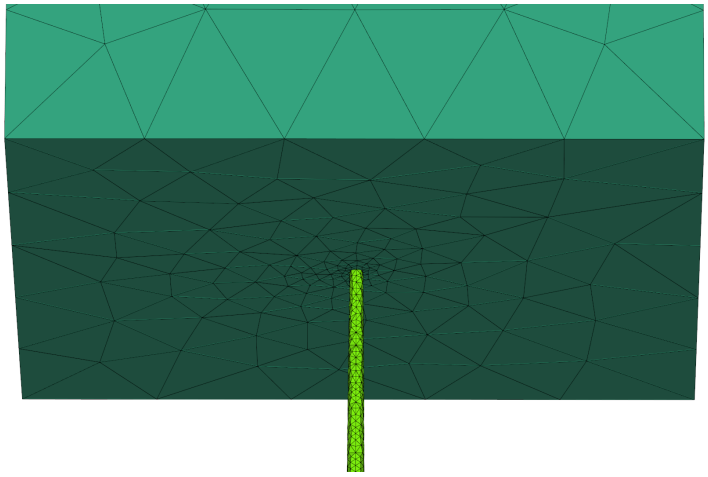


Figure 4.34 | Zoom on cone extremity. Here the extremity of the cone was truncated. It is represented by a disk of radius $r = 5$nm.

4.5.2.2 Results

We first discuss the numerical treatment of the optical characteristics of GaAs. This material exhibits a zero absorption above its bandgap which is around 870 nm. While this zero absorption

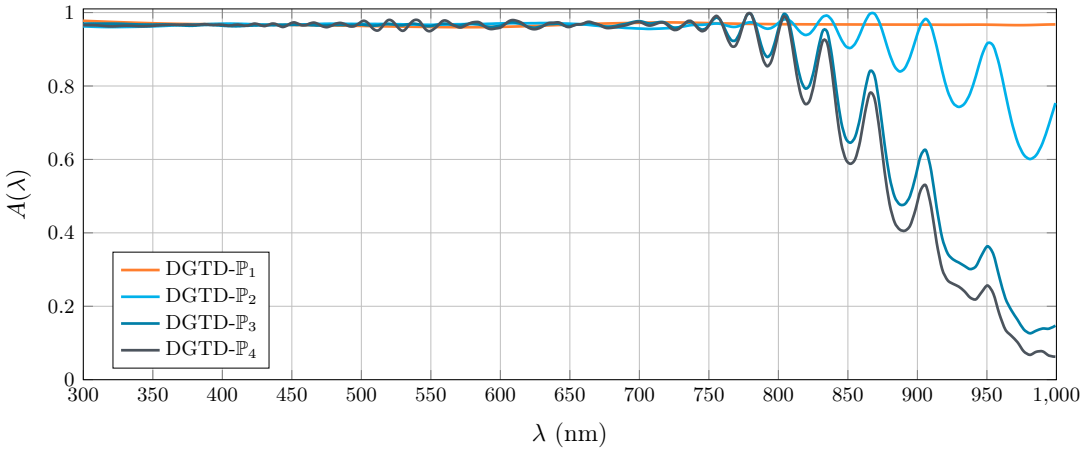


Figure 4.35 | A_{tot} in the GaAs photonic crystal on coarse mesh with polynomial orders from 1 to 4.

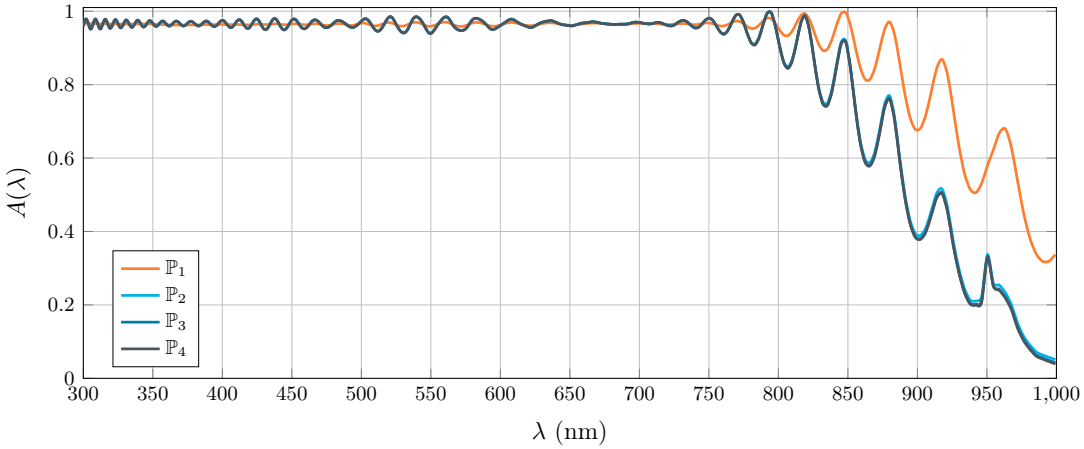


Figure 4.36 | A_{tot} in the GaAs photonic crystal on fine mesh with polynomial orders from 1 to 4.

feature is easily accounted for in the frequency-domain modeling setting by simply taking $\varepsilon_i(\lambda) = 0$ for $\lambda > 870$ nm, it represents a more challenging task in the time-domain case. The generalized dispersion model presented in details in [Viq18] involves several parameters. The process of fitting experimental optical data yields numerical values for these parameters through the resolution of an optimization problem. Simulated annealing is an appealing algorithm in this context, in particular, because it is a global optimization method. However, this algorithm works with several parameters, such as the number of neighbors used, the neighborhood size, or the temperatures, whose optimal values for a given problem and a target accuracy requirement in the solution of this problem, are in general difficult to infer. We have considered two parameter configurations (referred to as original and new in the following) and run the simulated annealing algorithm for fitting the real and imaginary parts of the electric permittivity of GaAs. The solution with our initial guess (option original), which works well for many other materials, is shown in figure 4.37. At first glance, the obtained result seems satisfactory, although, by zooming in the bandgap region, we clearly observe a mismatch in the fitting of the imaginary part. The left plot of figure 4.40 shows the result of a simulation with the DGTD-$\mathbb{P}_2$ method in terms of the total absorption spectrum. We note that the time-domain simulation fails to accurately model the zero absorption above 870 nm, which fits with

the mismatch of ε_i. Indeed, the accuracy of the simulation of light absorption in the studied GaAs solar cell appears to be highly sensitive to the quality of the fitting, especially near the bandgap. We tried to tune the simulated annealing algorithm in order to improve the quality of the fitting, by adding some constraints on the bandgap region in the optimization part. One can see the clear improvement in figure 4.38. We can see a comparison in log-scale in figure 4.39, which clearly highlights the fitting issue. The simulation performed with the second fit confirms the importance of the modelization of the imaginary part of the permittivity, which produced almost three times more unphysical absorption (see figure 4.40).

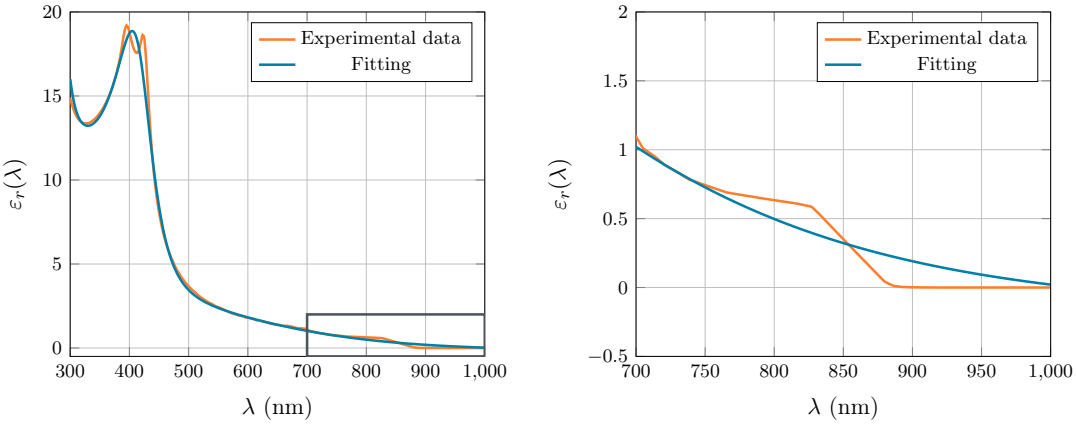


Figure 4.37 | Original fitting of imaginary part of the GaAs electric permittivity.

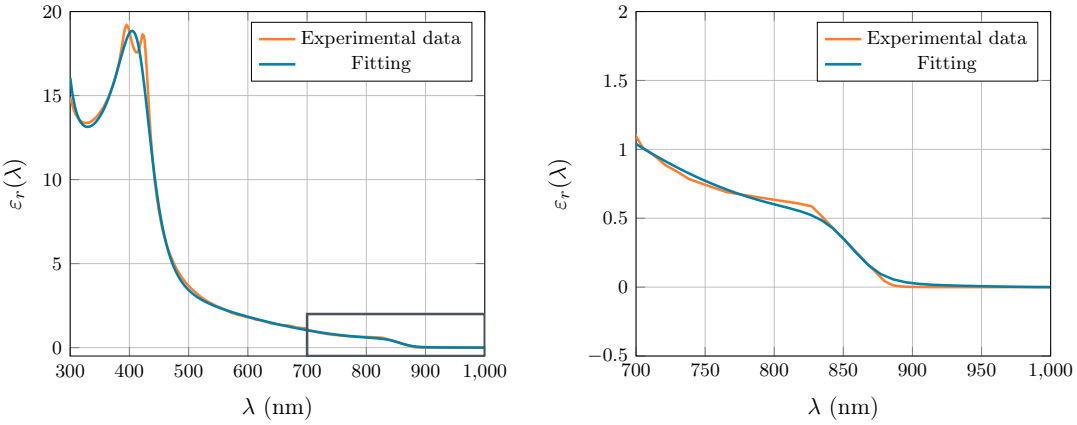


Figure 4.38 | Optimized fitting of imaginary part of the GaAs electric permittivity. We observe a clear improvement near the band gap region.

In this section we have shown the importance of the material model used, which has a huge influence on the final results. The bandgap's fit problem is a known problem in the time-domain field: in [EJD14], the absorption is plotted from 400 nm to 860 nm.

4.6 Thin-film solar cells

In this part, we present a joint work realized in interaction with physicists from IEK-5 Photovoltaic, Forschungszentrum Julich (FZJ). This work is part of the EoCoE project [LGAB18]. We

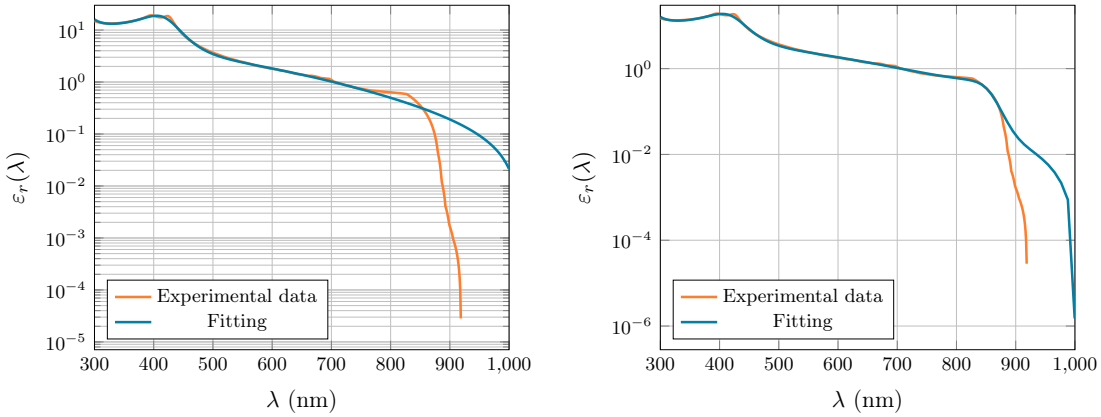


Figure 4.39 | Fitting of imaginary part of the GaAs electric permittivity in semi-log scale. (Left) is the original fit while (Right) is the optimal fit.

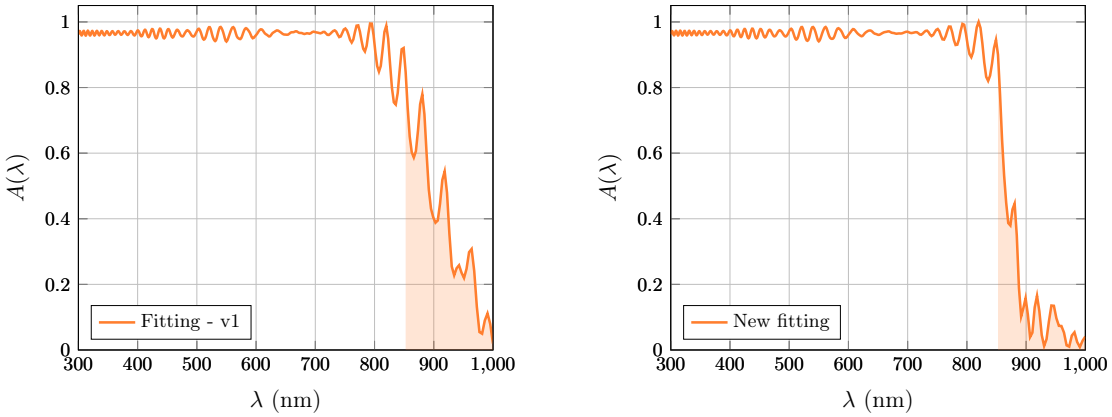


Figure 4.40 | Simulation using DGTD-$\mathbb{P}_2$ method. (Left) is based on the original fit while (Right) is based on the optimal fit. The obtained non physical absorption is divided by almost 3.

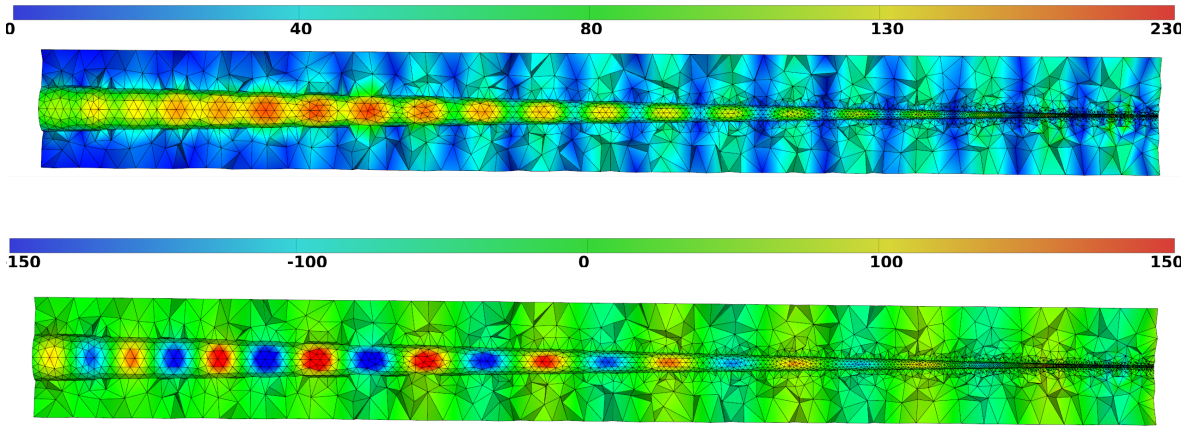


Figure 4.41 | Simulations using DGTD-$\mathbb{P}_2$ method and second option for the fitting: contour lines of the module of the DFT of **E** (top) and real part of E_x (bottom) for a wavelength $\lambda = 800$ nm.

study light trapping in a silicon-based thin-film solar cell setup that consists of several randomly textured layers (see figure 4.42). The focus is on amorphous and microcrystalline silicon (a-Si:H and μc-Si:H), which belong to the family of disordered semiconductors. The main characteristics of those materials is the structural disorder, which affects the optical and electronic properties significantly.

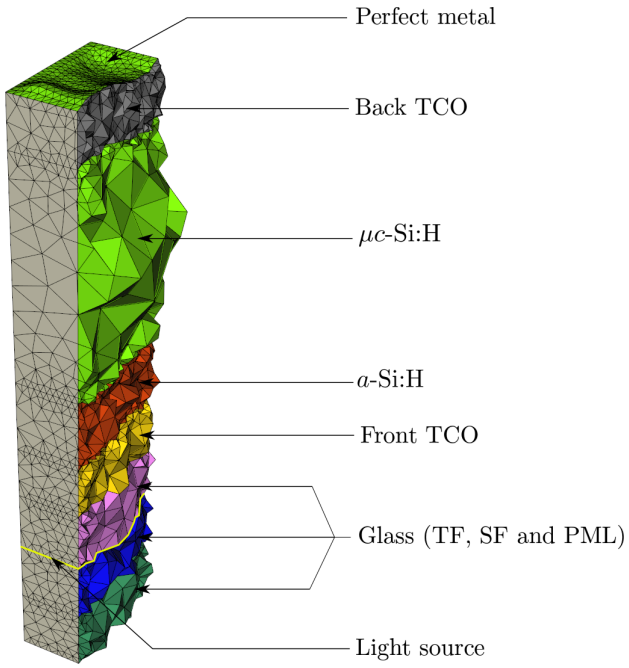


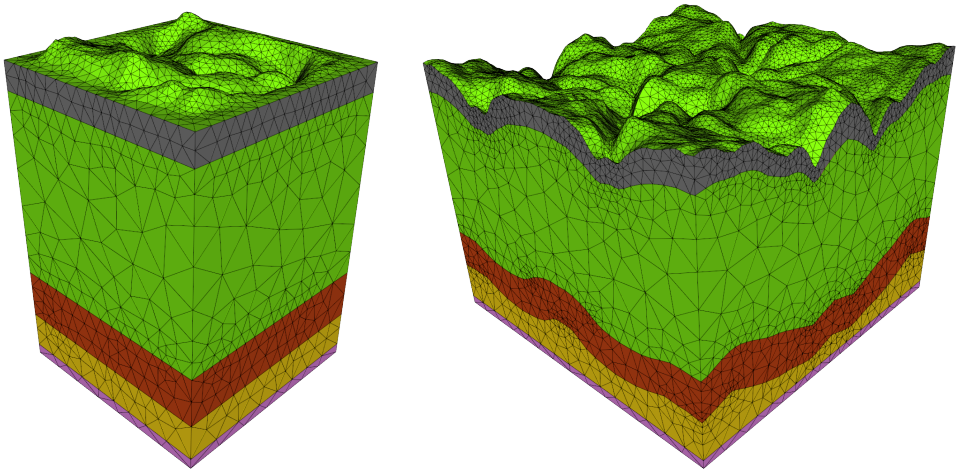
Figure 4.42 | Composition of the thin-film solar cell device.

4.6.1 Geometrical models

Researchers from FZJ provide us with topographic data of each layer obtained from AFM. In order to generate a geometrical model, we apply the different strategies presented in section 4.3. Partial views of generated tetrahedral meshes are shown in figure 4.43. These meshes correspond to a subset of the full model, which has been used for preliminary tests and comparison of the different strategies for periodic boundaries. It is obtained by taking only 1×1 μm^2 subset of each layer data instead of the full 10×10 μm^2 points. The mesh in figure 4.44 is obtained for the full model using the Tukey window approach. Indeed, it is way too costly to use the alternative method for imposing periodicity as it yields a tetrahedral mesh with 5 million elements.

4.6.2 Material models

The optical properties of the different materials that constitute the considered solar cell structure have been fitted to the parameters of our generalized dispersion model, which was originally intended for metals. The obtained permittivity functions are plotted in figure 4.45. As can be seen, all materials are relatively well approximated in the range $\lambda = [300, 1000]$ nm. Regarding glass, a constant permittivity $\varepsilon_r = 2.25$ is used.



(a) Modified mesh using Tukey window approach

(b) Modified mesh using symmetric mesh

Figure 4.43 | Submodel meshes obtained with the two different strategies for imposing periodicity.

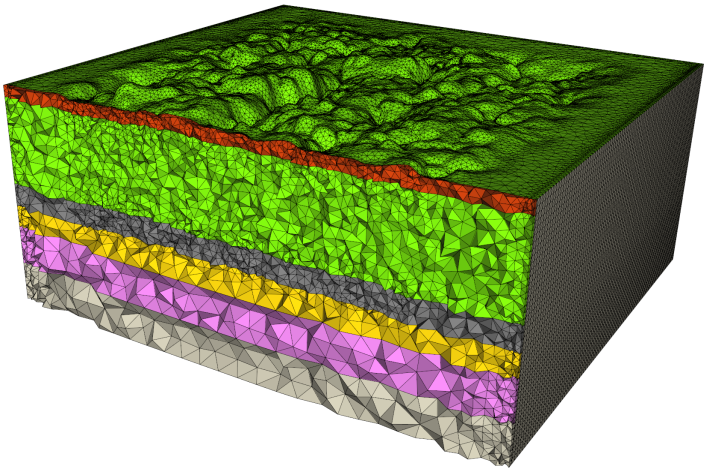
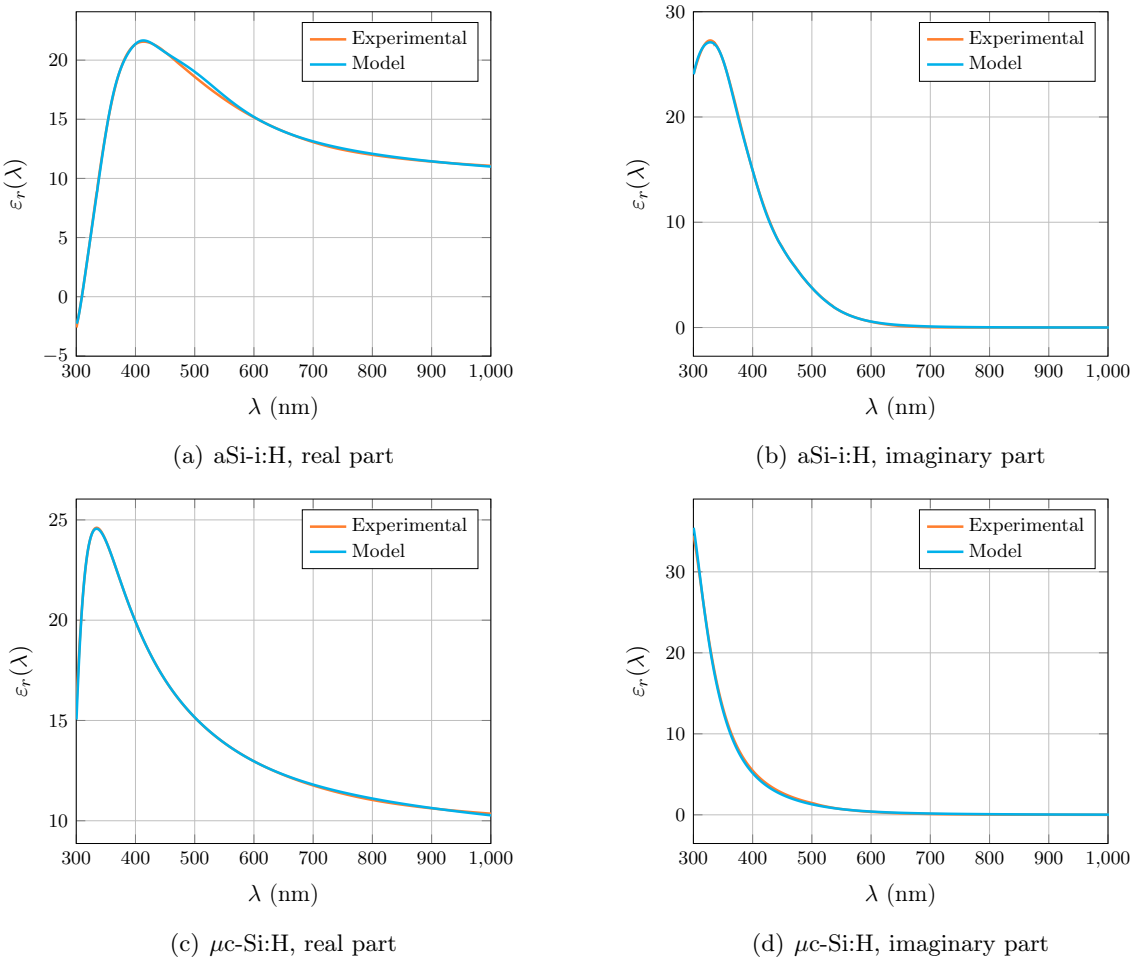


Figure 4.44 | Full cell using the Tukey window approach with $r = 0.3$.

4.6.3 Numerical convergence

Numerical convergence has been assessed by considering the small submodel of the solar cell structure on the one hand, and the full-size model, on the other hand. The first mesh (M1) is used as a reference: the nanotextures are removed and the interfaces are flattened. Two tetrahedral meshes (M2 and M3) have been constructed on the realistic topography data, using the procedure described in section 4.3. The characteristics of the meshes used in this study are summarized in table 4.5. In this table, h_{min} and h_{max} respectively stand for the minimum and maximum length of a mesh edge. We plot in figure 4.46 to 4.48 the EQE, for both materials, obtained for polynomial orders ranging from 1 to 4 in the DGTD method. As can be seen, the DGTD-$\mathbb{P}_3$ and



(a) aSi-i:H, real part

(b) aSi-i:H, imaginary part

(c) μc-Si:H, real part

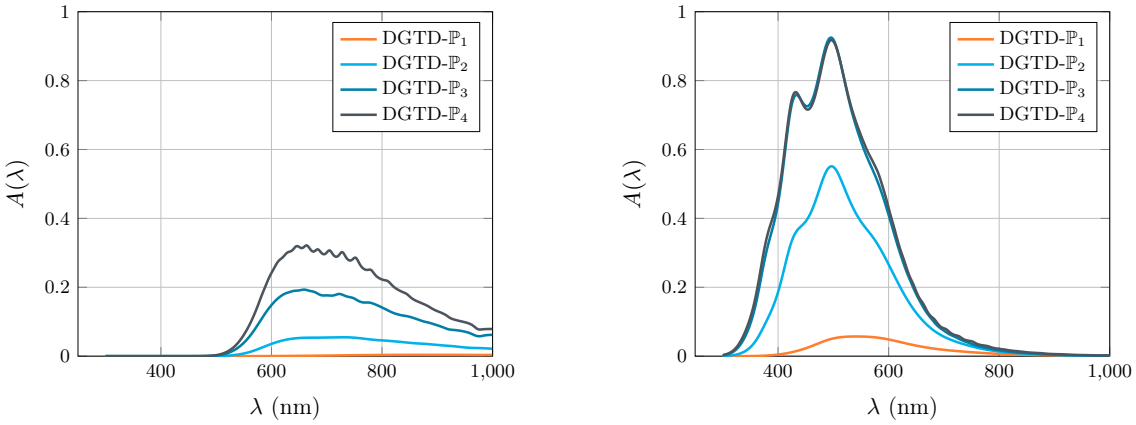
(d) μc-Si:H, imaginary part

Figure 4.45 | Real and imaginary parts of the relative permittivity of front TCO, a-Si:H, μc-Si:H and back TCO predicted by our dispersive model compared to experimental data.

DGTD-$\mathbb{P}_4$ methods yield almost identical results when considering the submodel for the a-Si:H layer. Moreover, we observe a smoothing of the plots from M1 to M3. Indeed in figure 4.46, the EQE on M1 presents some resonances for both the μc-S:H (between 600 nm and 800 nm), and for a-Si:H (at 430 nm and 500 nm). We seem to be observing some residuals of this with the submodel in figure 4.47, while the EQE profiles in figure 4.48 are smoothed. From those statements, we can say that flat interfaces cause those resonances. It is also visible on the submodel results because of the Tukey window, which is flattening a large portion of the surface.

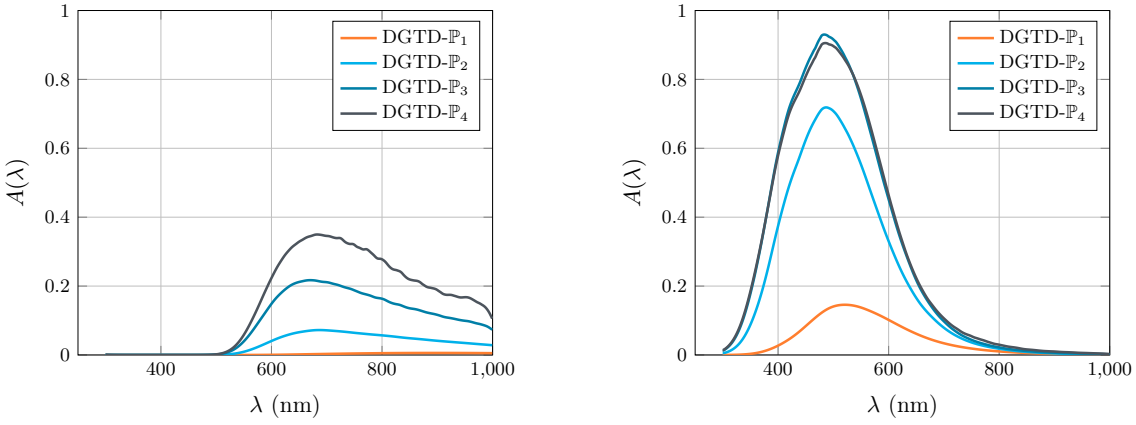
Mesh	# Tetrahedra	$h_{\min}$ (nm)	$h_{\max}$ (nm)	$\frac{h_{\max}}{h_{\min}}$
M1: $1 \times 1\,\mu\text{m}^2$	29461	29.2	465.7	15.9
M2: $1 \times 1\,\mu\text{m}^2$	41387	9.9	482.6	48.7
M3: $10 \times 10\,\mu\text{m}^2$	1151793	6.7	917.8	137.0

Table 4.5 | Characteristics of the tetrahedral meshes for the $1 \times 1\,\mu\text{m}^2$ and $10 \times 10\,\mu\text{m}^2$ models.



(a) μc-Si:H (b) a-Si:H

Figure 4.46 | EQE for $1 \times 1\,\mu\mathrm{m}^2$ flat submodel (mesh M1) with polynomial orders from 1 to 4.

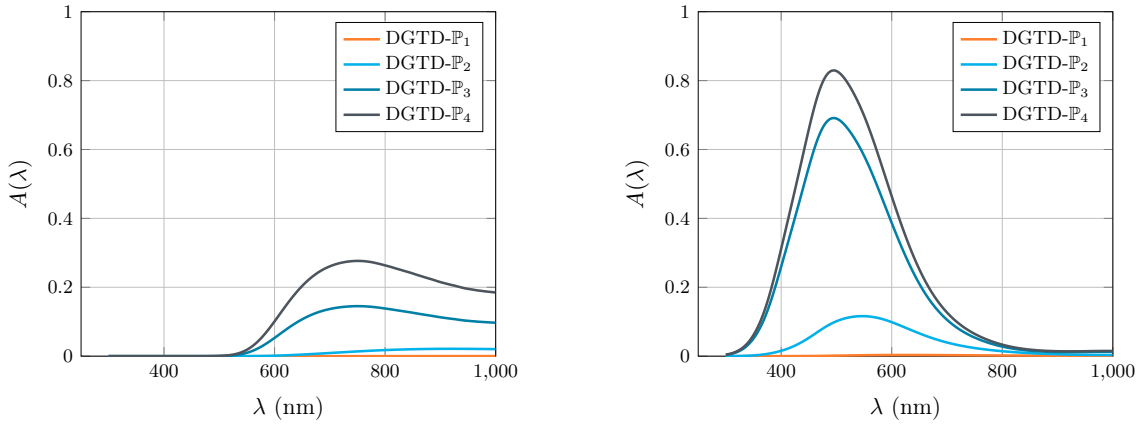


(a) μc-Si:H (b) a-Si:H

Figure 4.47 | EQE for $1 \times 1\,\mu\mathrm{m}^2$ submodel (mesh M2) with polynomial orders from 1 to 4.

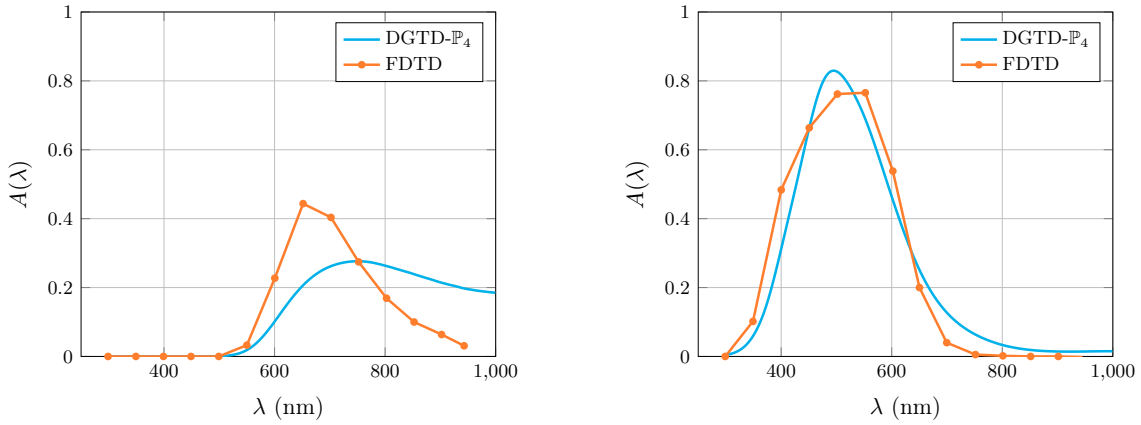
4.6.4 Comparison with FDTD simulations

A comparison with results from FDTD simulations performed at IEK-5 is shown in figure 4.49. As can be seen, the results are in relatively good agreement, especially for the a-Si:H. The observed discrepancies can be attributed to two causes: (i) the use of a Cartesian grid in the FDTD method and (ii) the number of frequency points for the evaluation of the observable. Concerning the first point, the results of the FDTD simulations presented here have been obtained for a Cartesian grid that is refined in order to minimize the staircasing effect inherent to this type of discretization, while keeping the simulation time to a reasonable level. A more accurate comparison would require to refine the underlying Cartesian grid further, therefore, increasing the FDTD simulation time. Regarding the second point, the FDTD results have been obtained by considering a monochromatic plane wave for the incident field and sweeping over the target frequency range. Because of the computational overhead, the number of FDTD simulations has been kept low, as can be seen in figure 4.49.



(a) μc-Si:H (b) a-Si:H

Figure 4.48 | EQE for $10 \times 10\,\mu\mathbf{m}^2$ model (mesh M3) with polynomial orders from 1 to 4.



(a) μc-Si:H (b) a-Si:H

Figure 4.49 | Comparison of EQE from simulations with the FDTD and DGTD solvers, for microcrystalline silicon (μc-Si:H) and amorphous silicon (a-Si:H) using the full model (mesh M2).

Shannon theorem for frequency-dependent observables Since we are considering a time-domain modeling setting, the computation of the observable quantities discussed previously requires to perform a discrete Fourier transform of the electromagnetic field on the fly. Therefore, in order to reduce the computational overhead of computing the volumetric absorption, we have exploited Shannon's theorem for sampling of frequency-dependent quantities. In fact, setting:

$$\Delta t_{\text{obs}} = \frac{1}{2 f_{max}},$$

allows to perfectly sample the spectrum of the $\mathcal{A}_{\text{layer}}$ operator, by evaluating equation (4.11) at certain Δt_{obs} time-steps. By doing so, we can reduce the CPU time up to 400% for the full model as one can see in table 4.6. These simulations have been performed on an in-house cluster system with 128 cores (16 Intel E5-2670 @ 2.6 GHz nodes each with 8 cores).

Mesh	Order	Improved time	Original time	Gain (%)
$1 \times 1\,\mu\text{m}^2$	$\mathbb{P}_1$	12 mn 38 s	13 mn 58 s	10.6
	$\mathbb{P}_2$	19 mn 01 s	23 mn 39 s	24.4
	$\mathbb{P}_3$	46 mn 58 s	1 h 01 mn 08 s	30.2
	$\mathbb{P}_4$	3 h 25 mn 34 s	3 h 45 mn 06 s	9.5
$10 \times 10\,\mu\text{m}^2$	$\mathbb{P}_1$	3 h 02 mn 16 s	15 h 54 mn 14 s	423.5
	$\mathbb{P}_2$	9 h 29 mn 52 s	49 h 46 mn 51 s	424.1
	$\mathbb{P}_3$	32 h 43 mn 57 s	126 h 12 mn 00 s	285.5

Table 4.6 | CPU times with and without applying Shannon theorem (full model, mesh M2).

4.7 Conclusion

In this chapter, we started by reviewing the three major numerical methods used in the PV field to deal with the optical part of PV problems. Then, we have presented the main quantities of interest used when accessing the quality of solar devices, precisely the external quantum efficiency (EQE) and the short circuit current (Jsc). The procedure to build a realistic mesh starting from a topography measured, generally obtained by atomic force microscopy (AFM), has been presented. Some geometry processing has been explained, in order to get usable meshes, mostly in terms of periodicity. Then we showed different realistic solar cells simulation, from several collaborations with physicists. We provided convergence analysis for all the cases, and we compared our results with the state of art solvers. Our methods obtained relevant results that are close to the other methods. In future work, it would be interesting to validate our results with experimental results. Also, a comparison in terms of computation time is mandatory in order to highlight which methods are the most profitable depending on the considered case.

5

Design optimization

Contents

Optimization is an essential tool for making decisions. It focuses on finding the best solution among the set of all feasible solutions. It is used in many science and engineering fields. When coupled with numerical solvers, it allows prototyping at virtually no cost. For this reason, optimization is widely used in photovoltaics. Most of the time, the quantity to maximize is the absorption of solar light by the cell. One of the main challenges and research axis consists in increasing the light trapping in the absorber to exceed the best possible efficiency [GY10]. In [GMY13], Ganapati *et al* use optimization in order to design nanoscale textures for light trapping in subwavelength thin-films. The optimization is based on an adjoint gradient method to search for a local optimum,

and is coupled with FDTD optical simulations. The authors represent the surface texture as a truncated Fourier series, and the optimization parameters are the Fourier coefficients. In [DS13], Düring *et al* use topological optimization on the geometry of the surface structuring for both plasmonic and dielectric devices. The merit function is the extraordinary optical absorption (EOA) at one specific wavelength. They show that the topological optimization can propose non-intuitive surface designs. In [LYE+17], Lee *et al* study the quasi-random features that can be imposed on an amorphous silicon surface by wrinkle lithography. They use Fourier-based inverse design and show an enhancement of light trapping capability by a factor of five over the $[400\text{ nm}, 1200\text{ nm}]$ region. In [ACM], Anderson *et al* are optimizing the efficiency of thin-film PV solar cells. The optimization algorithm is based on a gradient-free optimization algorithm, which is well suited for maximizing an objective function over a high-dimensional parameter space, coupling with an HDG optoelectric solver.

Most of the optimization problems in science and engineering are black box optimization problems, and are characterized by an objective function $f(\mathbf{x}) \in \mathbb{R}$ that does not have an analytic expression. The function f is depending on the parameter $\mathbf{x}$, which is searched in a space called the admissible design space, denoted by $\mathbf{W} \subset \mathbb{R}^d$. Evaluating f can be cheap, which allows a sampling of many points in $\mathbf{W}$, *e.g.* , via random search, or it can be expensive, such as when it requires the simulation of a complex physical problem in three space dimensions, or when the data required to evaluate the objective function come from laboratory experiments. In such situations, one has to select an optimization technique that minimizes the number of evaluations of the objective function. Moreover, the problem at hand may exhibit several local minima, thus motivating the use of a technique that can deal efficiently with this issue and provide a global optimal solution. In this context, Bayesian Global Optimization (BGO) techniques are particularly relevant [JSW98].

In this chapter, we consider an optimization strategy based on an adaptive statistical learning approach. The method used is called Efficient Global Optimization (EGO). It was initially introduced in [JSW98], and it is now one of the most popular BGO techniques. This method relies on the construction of metamodels or surrogate models. Historically, linear regressions were used as metamodels, while today Gaussian Process (GP) models are used. It has been initially developed for optimization studies concerning fluid flow problems [DC12, DLG16] and fluid-structure interaction problems [SHD+17]. We briefly review the design and working principles of this optimization approach. GP based optimization methods have become very popular in recent years for the global optimization of complex systems characterized by high computational costs in the evaluation of the objective function. GP models have also been found especially interesting for optimization, particularly in the framework of the EGO method. One of the main ideas of the EGO is to rely on an acquisition function, usually, the Expected Improvement (EI) [Moc89], to find new promising designs. During the iterative process, the new point obtained is tested in the objective function, and its result is used to update and improve the accuracy of the metamodel.

The chapter starts with a presentation of the EGO algorithm in section 5.1. The optimization notations are explained, and a simple example is provided to illustrate the method. Section 5.2 deals with the study of a textured PV glass. The main objective is to reduce the reddish reflection caused by the silicon while enhancing the total absorption. To do so, we optimize the pyramidal elementary pattern, which is replicated into all layers. Section 5.3 deals with a hot carrier solar cell. It is using a thin layer of indium gallium arsenide (InGaAs). The main idea is to work on the nanostructured back mirror to increase light trapping. This case is optimized using a parallel EGO (also named q-EGO), which allows to accelerate the iterative part of the EGO algorithm by evaluating multiple objective functions in parallel.

5.1 Efficient Global Optimization (EGO)

Bayesian Global Optimization (BGO) provides efficient and effective techniques based on Bayes Theorem to direct the search of a global optimization problem. To do so, it starts by building a probabilistic model of the objective function, called the metamodel or surrogate model. The specific BGO technique used in this work is called Efficient Global Optimization (EGO). EGO is used when the evaluation of the objective function is an expensive task, which is precisely the case in our context in which we resort to a full-wave electromagnetic solver in three space dimensions for each cost function evaluation. EGO consists of two steps. The first one is called the *learning step*. It relies on a set of values of the objective function at several points in the parameter space. Those values are used to build a surrogate model. It is usually called the Design of Experiments (DoE).

5.1.1 Design of Experiments

The initial metamodel is constructed from a design of experiment (DoE) denoted by $\mathbf{X}_{\text{DoE}} = \{\mathbf{x}^1, ..., \mathbf{x}^n\} \in (\mathbf{W})^n$, where $n \in \mathbb{N}$, $n > 0$ is the number of points in the DoE. Usually, $\mathbf{W} = \mathbb{R}^d$. The set $\mathbf{X}_{\text{DoE}}$ must be chosen depending on the complexity of both f and $\mathbf{W}$ as the accuracy of the resulting metamodel depends on it. Different possibilities are available to construct the design, such as uniform sampling, random sampling, orthogonal sampling, or the reference method, which is used in this work, the Latin Hypercube Sampling (LHS) [SWMW89]. LHS works as follows: it positions each design in the DoE database such that it does not have common coordinates with the other previously positioned designs, as illustrated in figure 5.1. The main advantage of this strategy is the minimization of the interaction of the variables, as shown in [Ste87]. Then, one

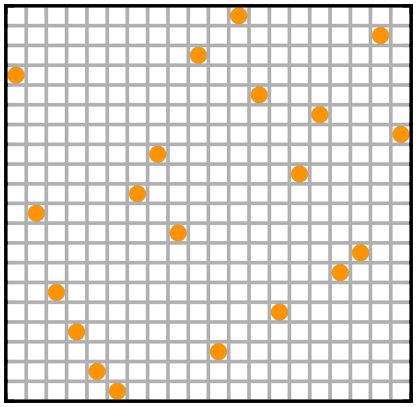


Figure 5.1 | Illustration of LHS sampling for 2 discrete parameters with $n = 20$.

evaluates the objective function for each $\mathbf{x}^i \in \mathbf{X}_{\text{DoE}}$ to get the set of data $(\mathbf{X}_{\text{DoE}}, f(\mathbf{X}_{\text{DoE}}))$. This allows to construct a surrogate model that aims at mimicking precisely the behavior of f while being computationally cheaper to evaluate.

5.1.2 Surrogate model

A popular surrogate model for Bayesian optimization is a Gaussian process (GP). GP modeling is a statistical method to approximate functions from a finite set of possibly noisy observations at ar-

bitrary points. An initial surrogate is constructed using the dataset from the DoE. This metamodel is then used to predict at any promising design $\mathbf{x}^t$ where t, $t > n$ represents the index of the design. Then one evaluates $f(\mathbf{x}^t)$, which is added to the dataset $\mathbf{X} = \{(\mathbf{X}_{\text{DoE}}, f(\mathbf{X}_{\text{DoE}}), (\mathbf{x}^t, f(\mathbf{x}^t)\}$, and used again to update the surrogate model. GPs are used for both their cheap evaluations and their interesting statistical information. Indeed, one can predict the objective function at points $\mathbf{x}^t$ thanks to the GP model mean, as well as an estimation of the error in terms of the GP model variance.

5.1.3 Acquisition functions

The new designs used to update the surrogate are obtained via an acquisition function, which must be defined in such a way that it allows both exploitation and exploration. Exploitation uses the predictions of the surrogate model to propose a new design maximizing or minimizing the objective function, while exploration proposes a new design in the area of the design space where the uncertainty of the model is high. More formally, the new design $\mathbf{x}^t$ is defined by $\mathbf{x}^t = \text{argmax}_{\mathbf{x}\in\mathbf{W}}\, g(\mathbf{x}|\mathbf{X}_{1:t-1})$ where g is the acquisition function and $\mathbf{X}_{1:t-1} = \{(\mathbf{x}^1, f(\mathbf{x}^1)), ..., (\mathbf{x}^{t-1}, f(\mathbf{x}^{t-1}))\}$ are the $t-1$ samples, computed from f so far with $t > n$. The most commonly used acquisition function, and also the one considered in this work, is the Expected Improvement (EI). To give an idea of what is the expected improvement, we place ourselves in the case where we have evaluated $t-1$ designs. The goal is to maximize f on $\mathbf{W}$, and our current largest evaluation is $f^+_{1:t-1} = \max_{\mathbf{x}\in\mathbf{X}_{1:t-1}} f(\mathbf{x})$. We suppose that we have one additional evaluation $\mathbf{x}^t$ to perform. After this evaluation, the new optimum will either be $f^+_{1:t-1}$ if $f^+_{1:t-1} > f(\mathbf{x}^t)$, or $f(\mathbf{x}^t)$ if $f^+_{1:t-1} < f(\mathbf{x}^t)$. One can quantify the improvement over this additional evaluation as $\max(f(\mathbf{x}) - f^+_{1:t-1}, 0)$. As $f(\mathbf{x})$ is unknown until its evaluation, one can take its prediction from the surrogate model and so choose $\mathbf{x}^t$. More formally, the EI is defined as

$$\text{EI}(\mathbf{x}) = \mathbb{E}\max(f(\mathbf{x}) - f^+_{1:t-1}), 0), \tag{5.1}$$

which can be evaluated analytically under the GP model as

$$\text{EI}(\mathbf{x}) = \begin{cases} (\mu(\mathbf{x}) - f^+_{1:t-1})\Phi(Z) + \sigma(\mathbf{x})\phi(Z) & \text{if } \sigma(\mathbf{x}) > 0 \\ 0, & \text{if } \sigma(\mathbf{x}) = 0, \end{cases} \tag{5.2}$$

where

$$Z = \begin{cases} \dfrac{\mu(\mathbf{x}) - f^+_{1:t-1} - \xi}{\sigma(\mathbf{x})} & \text{if } \sigma(\mathbf{x}) > 0, \\ 0 & \text{if } \sigma(\mathbf{x}) = 0, \end{cases} \tag{5.3}$$

with $\mu(\mathbf{x})$ and $\sigma(\mathbf{x})$ are respectively the mean and the standard deviation of the GP prediction at $\mathbf{x}$. The functions Φ and ϕ are respectively, the cumulative distribution function (CDF) and the probability density function (PDF) of the standard normal distribution. The first term in equation (5.2) is the exploitation term, and the second term is the exploration term.

5.1.4 EGO workflow and illustrative example

The DoE phase is carried out to construct an initial model. Then, an iterative and adaptive strategy is used to enrich the model and locate the most interesting values of the cost functional. In this perspective, an internal optimization problem is solved to determine a set of new promising designs to be evaluated. The new cost functional evaluations are then employed to update the

model. This iterative enrichment step is conducted until a stopping criterion is verified. It can be an a priori number of iterations N, a budget time constraint *e.g.* the optimization should take less than 10 hours, or even a convergence threshold, *e.g.* based on the merit function as EI $< \epsilon$. We illustrate in figure 5.2 the main steps of EGO.

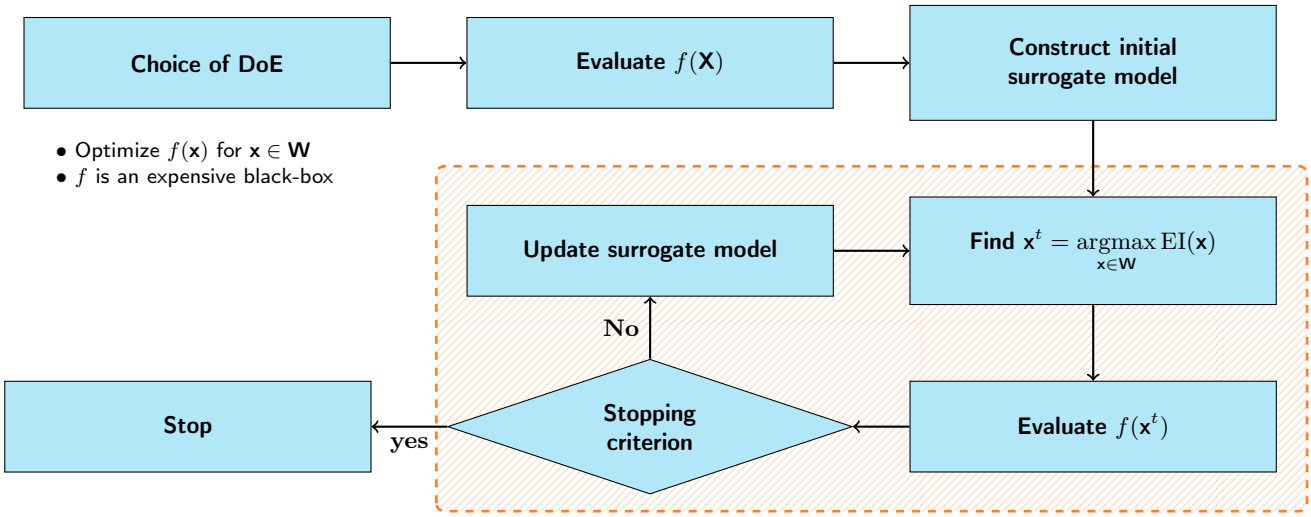
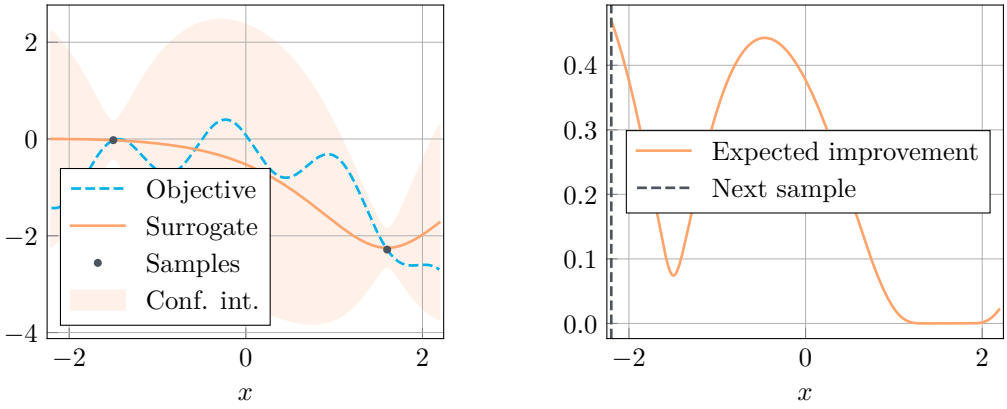


Figure 5.2 | EGO optimization flow diagram. The orange part represents the iterative part of the EGO.

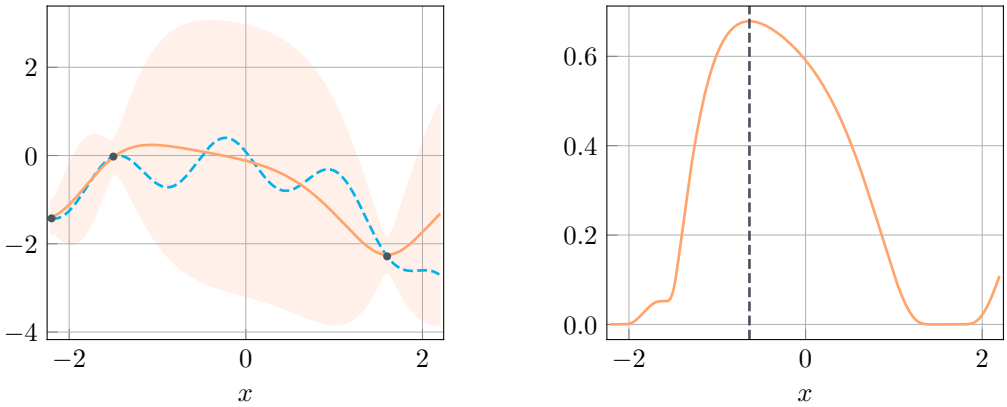
The operation of the EGO algorithm is now illustrated using a 1D example, which consists of maximizing an analytic objective function given by

$$f(x) = -\cos 3x \, \cos 2(x+1) - \frac{x^2}{3}, \quad x \in [-2.2, 2.2]. \tag{5.4}$$

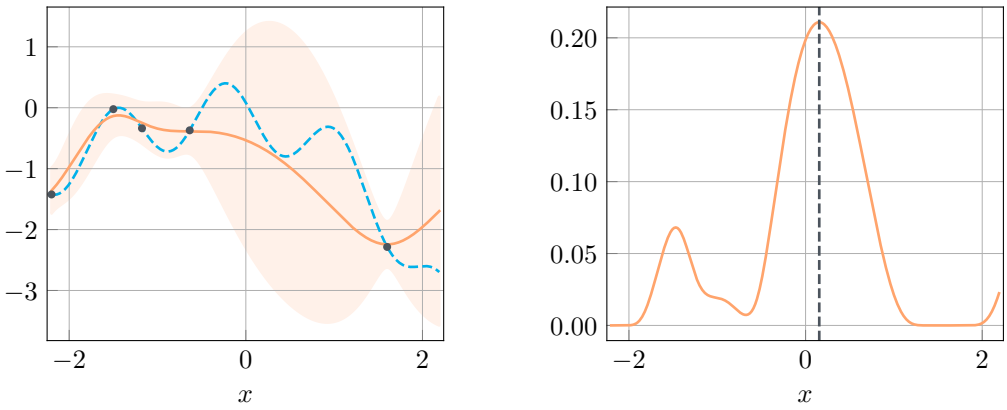
Let us choose a DoE composed of two points, *i.e.* $\mathbf{X}_{\mathrm{DoE}} = \{-1.6, 1.6\}$. We represent four steps of the EGO procedure in figure 5.3. On the left column, one can see the objective function (dashed blue line), the measures (black dots), and the surrogate model represented by its mean (orange curve) and its confidence interval (light orange surface). In the right column, one can see the expected improvement in orange along with $\mathbf{x}^+$. We represent in figure 5.4 (a) the distance between two consecutive points predicted from EI, and in figure 5.4 (b), the current maximum value. It is interesting to see how the surrogate is evolving: at first, it seeks and searches in uncertain regions (from iteration 1 to 5), and then it starts focusing the refinement close to the actual maximum. This is one of the very interesting property of EGO, which allows to reach global optimums without being stuck in a local one.



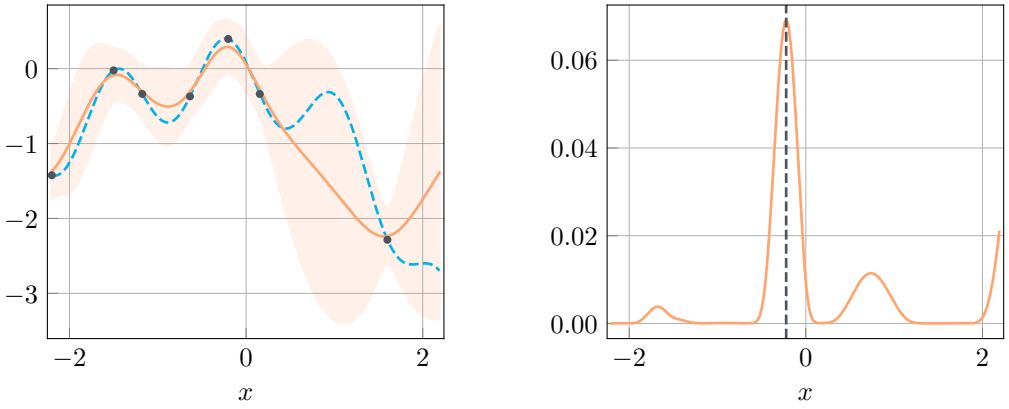
(a) Iteration 1.



(b) Iteration 2.

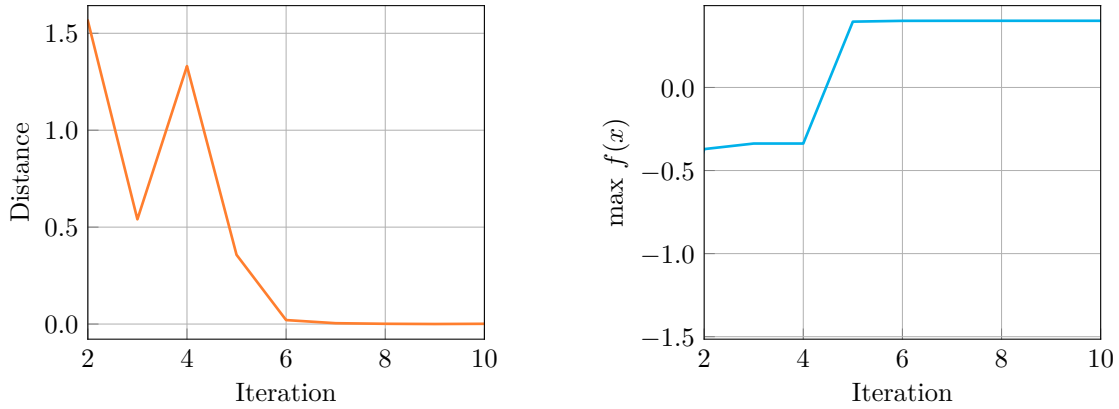


(c) Iteration 4.



(d) Iteration 6.

Figure 5.3 | Simple 1D example of the EGO. Superposition of the analytical objective function with the surrogate for several iterations of the EGO algorithm (Left). Plot of the expected improvement in orange and of the future sampling point (Right).



(a) Distance between consecutive points.

(b) Current maximum.

Figure 5.4 | Simple 1D example of the EGO. In the left we can see the distance between two consecutive $\mathbf{x}^+$. The iterations 1 to 5 correspond to the exploration part. In the right we plot the current maximum.

In the following, we apply the EGO method to the design of nanostructured solar cells for optimizing their light trapping properties. We consider two solar cell configurations corresponding to two different forms of nanostructure. Moreover, we consider EGO implementations from two different software frameworks: the FAMOSA software suite and the DiceOptim package for R.

FAMOSA The Full and Adaptive Multi-Level Optimum-Shape Algorithm[1] (FAMOSA) is an optimization platform developed at Inria Sophia Antipolis - Méditerranée by the Acumes project team. It is devoted to multidisciplinary design optimization. This software must be coupled to a solver that provides a merit function. FAMOSA can be coupled to any solver and, therefore, can

[1] http://famosa.gforge.inria.fr/

be used for several purposes, ranging from numerical to multidisciplinary design optimization. It has been developed initially for aerodynamic design.

DiceOptim DiceKriging and DiceOptim[2] are R packages available on the official R(CRAN) repository. They have been developed in the frame of the DICE (Deep Inside Computer Experiments) consortium. DICE gathered major French companies and public institutions, having a high research and development interest in computer experiments. The main objective was to put in common industrial and academic problems in order to foster research and transfer in the field of design and analysis of computer experiments.

5.2 Surface texturing for enhanced color rendering

This study has been done in collaboration with Sunpartner Technologies, a company designing and manufacturing PV cells embedded in mainstream technologies. In this context, we consider a silicon based PV glass. This glass is actually a multilayered structure in which each layer of material contributes to the efficiency of the whole device. This effectiveness is assessed as the capacity of the structure to reflect specific colors of the visible spectrum in variable ambient lighting conditions, and to absorb further colors that affect user comfort. A difficulty that Sunpartner Technologies is currently facing in the development of its PV glass is the inadequate absorption of red tints, as shown in figure 5.5 left, which cause visible red reflections. Those reflections come from the absorption coefficient of the silicon. Indeed, it is significantly higher in the blue part than in the red part, which results in a significant penetration depth of nearly fifty microns. Minimizing the reflection of the red part of the spectrum and, at the same time, improving the absorption properties of the device is achievable from two main strategies. The first one is to increase the thickness of the silicon layer to reach its penetration depth. The second one consists of embedding the cell with anti-reflection coating and light trapping textures, which are reducing reflections, and increasing light trapping, respectively. From an industrial point of view, the first option may not necessarily be economically viable. The alternative option is to use thin layers. However, when the PV cell structure consists of an association of thin-films of some hundreds of nanometer for the thickest, there may be creation of colors by constructive interference during the light propagation in the multilayer structure.

The surface texturing is the most promising solution to deal with both issues: it can limit the effects of these interference phenomena on one hand, and increase the poor absorption of predominantly red colors on the other hand. Figure 5.5 right illustrates three strategies for texturing a typical structure, representative of a PV glass. This structure is made of three materials (glass, silicon (Si) and Transparent conductive oxid (TCO)), possibly with an anti-reflective gel coating. In the following, we propose to study the impact of textures by modeling and numerically simulating the propagation of solar light in this type of PV glass.

5.2.1 Objectives of the study

In this study, we limit ourselves to an idealized PV glass structure based on an average topography of the nanotexturing of the different layers, as shown in figure 5.6 for a classical solar cell stack. For the PV glass structure considered in this study, Sunpartner Technologies has provided an average topography representative of actual built structures, with three examples shown in figure 5.7. The

[2]https://cran.r-project.org/web/packages/DiceOptim/index.html

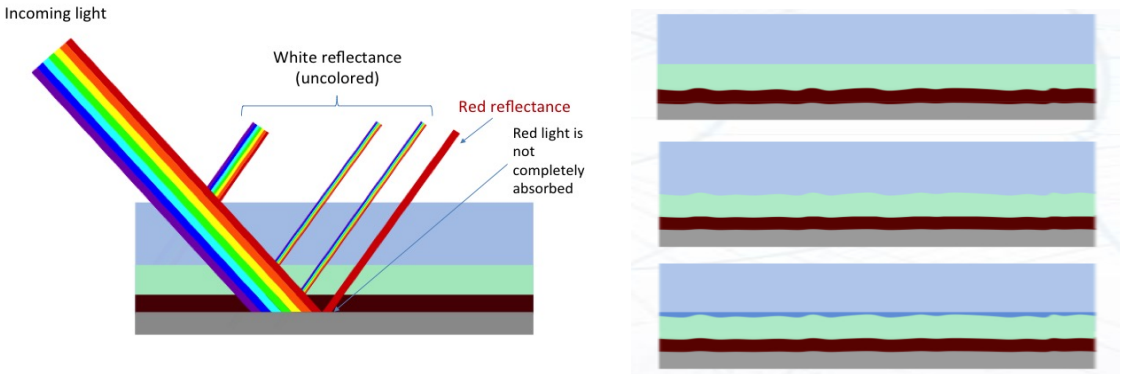


Figure 5.5 | Illustration of the problem of poor absorption of red tints (Left). Examples of texturations of a PV glass structure (Right) - Gray: back refelector (metal) - Brown: TCO (Transparent Conducting Oxide) - Green: glass - Dark blue: anti-reflective gel coating - Light blue: ambient medium.

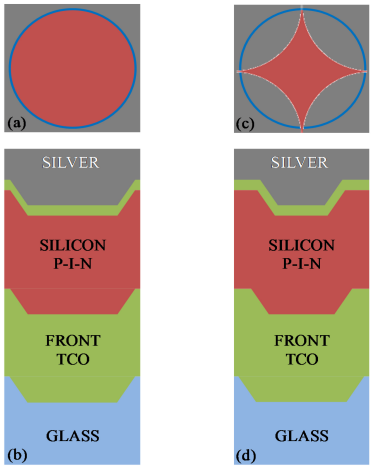


Figure 5.6 | Structure of a classical solar cell stack: top and transverse views.

use of a PV glass structure based on an average topography allows us to apply a hypothesis of periodicity more conveniently, as shown in chapter 4, reducing the size of geometric models and, thus, the computation cost of the full-wave solver. In section 5.2.4, a parametric study is first carried out in order to obtain a preliminary picture of the role of the texturing on the absorption of red tints and the creation of colors by constructive interference in the layers. Then, in section 5.2.5, an inverse design analysis is conducted using EGO in order to obtain the optimal nanotexturing for a given objective.

5.2.2 Cell composition

We consider a PV structure made of 7 layers (see figure 5.8). Layers characteristics are summarized in table 5.1. We can see that this cell is alterning very thick layers of hundreds of nanometers with very thin layers of few nanometers.

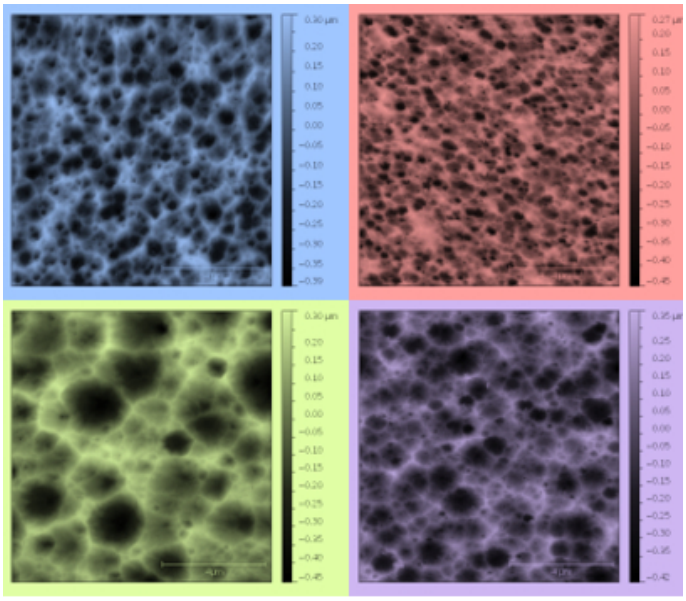


Figure 5.7 | Several nanostructurations of a ZnO layer surface (top view).

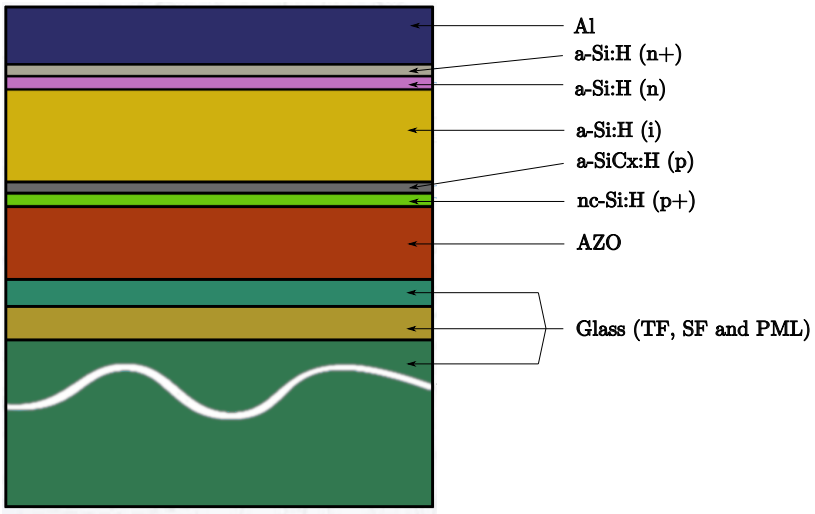


Figure 5.8 | Composition of the PV cell stack.

5.2.3 Construction of geometrical models

In this study, instead of considering realistic texturing of the PV layers as depicted in figure 5.7, we adopt a homogenized texturing. An analysis of the texturing of several tens of cells allowed to choose the pattern used, which is a periodic pyramidal PV pattern. Pyramid parameters such as heights and bases, are deduced from surface roughness data given by Atomic Force Microscopy (AFM) (defined in chapter 4). A geometrical model of this structure is shown in figure 5.9. Such simplified textures are useful for numerical simulations because they naturally allow the use of PBC, thus reducing the size of the computational domain drastically. To do so, we need to construct the elementary pattern of the structure, as shown in figure 5.10(a). It is also possible to reduce

Material	Thickness (nm)
AZO	800
nc-Si:H(p+)	8
a-SiCx:H(p)	12
a-Si:H(i)	400
a-Si:H(n)	12
a-Si:H(n+)	12
Al	500

Table 5.1 | Layers thicknesses. Glass thickness is 500 μm.

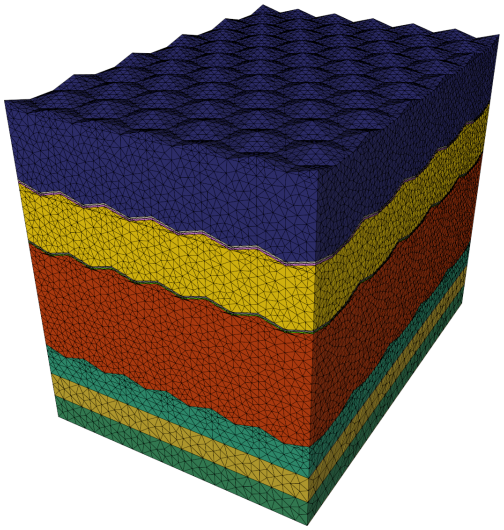


Figure 5.9 | Geometrical model of the PV stack with a texturing of the layers based on a pyramidal grating. This corresponds to a 5×4 pyramidal array. For the computation, we consider one pyramid with PBCs.

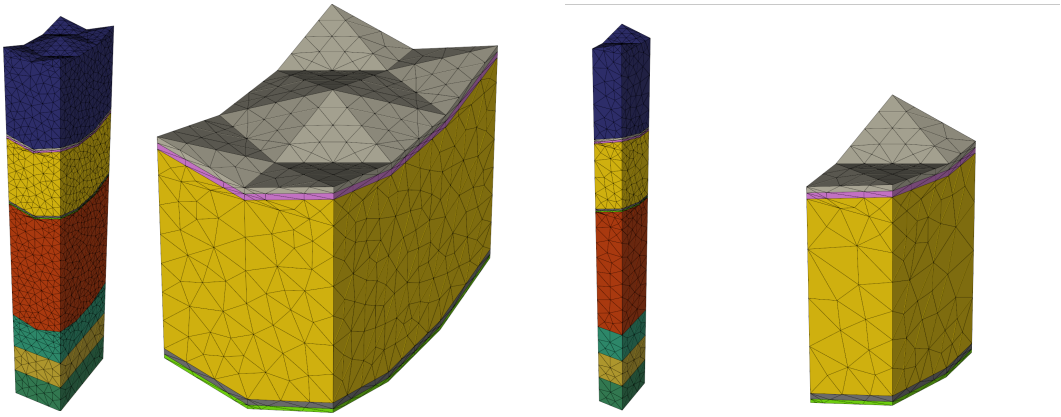
even more the size of the computational domain to one-quarter of the elementary pattern if normal incidence is assumed, see figure 5.10(b).

Remark *: Those geometrical simplifications are possible since we are not considering the realistic topography, as in figure 5.7. However, it would be interesting to validate the reliability of the simplified model with the realistic one, which will be the subject of a future study.*

Meshes are obtained using the Gmsh[3] CAD modeler and mesh generator. The elementary pattern that we consider in this study is modeled as the union of five pyramids.

Remark *: Such a geometrical model is not trivial to create, especially when the Computer-Aided Design (CAD) has to be modified and rebuilt in the framework of an optimization process. Indeed, in this context, the parameters are supposed to change almost freely. However here, if pyramid height parameters are bigger than the upper layer starting altitude, pyramids can intersect pyramids from the upper one, changing most of the underlying geometrical topology. Such issues are generally hard to handle and often require manual modifications.*

[3] `http://gmsh.info/`



(a) Elementary pattern, (b) Quarter of elementary pattern,

Figure 5.10 | Unit pattern with an isolation of the very thin Si-based layers.

5.2.4 Influence of the geometry of the pyramids

Based on the results of a preliminary study made by Sunpartner Technologies, which was concerned with the measurement of texture roughness of the PV glass samples, a particular texture has been selected for the purpose of this study (see figure 5.11). The characteristics of this texture are:

- `Ra` (nm): 47.8 ± 3.7 (average roughness);
- `Rq` (nm): 63.6 ± 8.4 (average quadratic roughness);
- `Sal` (nm): 165.1 (auto-correlation length).

The parameter `Sal` corresponds to the distance over the surface such that two points have minimal correlation. This value is used to set the periodicity of the simplified geometry. The parameters `Ra` and `Rq` correspond to the roughness of the texture. Those three parameters are used to determine the general texture specification (see table 5.2). In term of numerical and geometrical model,

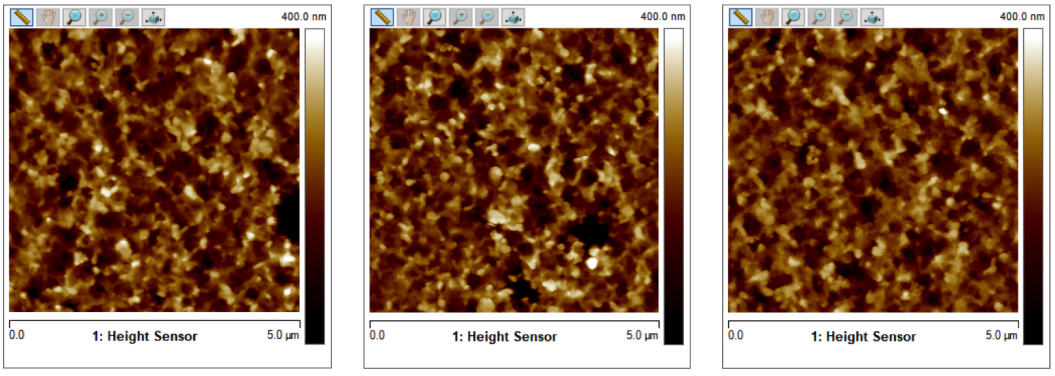


Figure 5.11 | Images of the nanotexturing of a AZO layer surface (top view) used in the definition of the selected texture.

we use a layer of Perfect Match Layer (PML) at the bottom of the cell to remove the parasitic

Material	h (nm)	r (nm)
AZO	47.8	165.1

Table 5.2 | Texturation of the AZO layer: geometrical characteristics of pyramids.

reflection. The incident light is propagating from the bottom to the top. In case of the periodic set-up, PBCs are used in both $x-y$ directions. For the quarter domain, PEC and PMC are used, with PEC in the polarization direction. The top of the cell corresponds to the Al back reflector, which is completed with a PEC boundary.

We first demonstrate the influence of the interpolation order in the DGTD method on the convergence of the absorption spectrum on a chosen geometry (h and r are fixed). For this purpose, we use a coarse tetrahedral mesh with only 1001 cells, and we consider the geometrical characteristics of pyramids given in table 5.2. The absorption spectra resulting from simulations with the DGTD-$\mathbb{P}_2$ and DGTD-$\mathbb{P}_3$ methods are shown in figure 5.12. Simulations have been performed on a multi-core server equipped with Intel Xeon Gold 6154 CPU@3.0 GHz. Using four cores the simulation time is respectively equal to 385 sec (DGTD-$\mathbb{P}_2$ method) and 1181 sec (DGTD-$\mathbb{P}_3$ method). We see that the solution resulting from the DGTD-$\mathbb{P}_2$ run is already sufficiently converged everywhere. An absorption above 60% is observed from 300 nm to 650 nm. The problem concerning the red part is clearly illustrated here: the absorption drops below 40% around 700 nm to 750 nm. Nevertheless, we choose to adopt the DGTD-$\mathbb{P}_3$ method in the sequel in order to be more confident with the convergence of our results.

In a second step, we perform a preliminary parametric study in order to assess the influence of the geometrical parameters h and r. Various configurations have been tested and are summarized in table 5.3. The corresponding absorption spectra are plotted in figure 5.13, 5.14 and 5.15. From figure 5.13, we restrict the modification to the pyramid radius. We observe a small improvement for the absorption around 650 nm. It even fills a gap at 670 nm. However, it did not bring any improvement in the upper part of the spectrum. In figure 5.14, we do the counterpart by changing only the height of the pyramid. We see some interesting improvement on the upper region of the spectrum with a slightly decay for the lower part. For the results in figure 5.15, we take two other couples of (h, r), both higher than the starting ones. Combining the two options seems to further increase the absorption in the upper red part, with again, a small decay for the resonances before 600 nm. From those statements, we are confident with the obtained results which confirm the effectiveness of the optimization strategy.

h (nm)	r (nm)	# elem	$h_{\max}/h_{\min}$
47.8	214.6	1053	19.4
47.8	165.1	1001	18.3
47.8	82.5	790	25.1
95.6	198.1	1236	24.6
95.6	165.1	1129	24.6
115.6	165.1	1091	26.1
119.5	198.1	1253	26.8
125.6	165.1	1149	27.0
125.6	185.1	1198	27.0

Table 5.3 | Mesh characteristics. Here $h_{\max}/h_{\min}$ is the ratio between the maximal and minimal length of mesh edges.

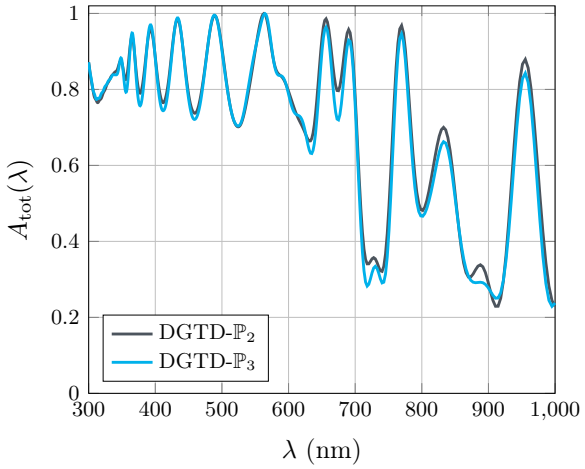


Figure 5.12 | Absorption coefficient convergence for the starting-point texture with h=47.8 nm and r=165.1 nm.

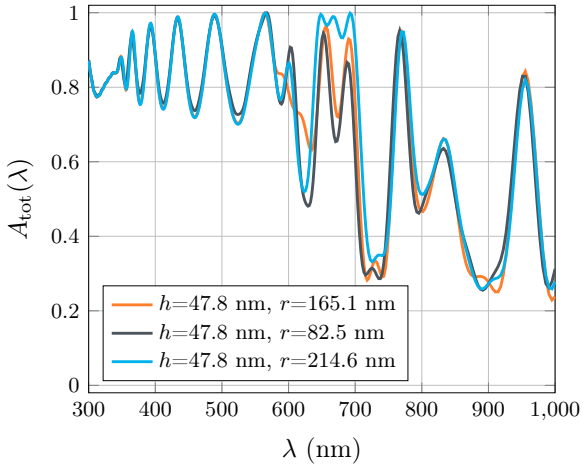


Figure 5.13 | Absorption coefficient for various textures obtained with DGTD-$\mathbb{P}_3$ method. Computation times for each configuration are respectively 1181 sec, 1360 sec and 1356 sec.

5.2.5 Optimization of the geometry of the pyramids

Now that we have a preliminary picture of the influence of the two parameters h (nm) and r (nm) on the absorption profile, we realize an inverse design analysis. In this part, we make use of the EGO method from the Famosa library. Starting from the geometrical characteristics of the average pyramidal grating given in table 5.2, we investigate here the possibility of improving further the performance of the PV glass structure by using a numerical optimization strategy. In order to reduce the red reflections while increasing the total absorption, we choose to minimize the total reflectance over the spectrum of interest, *i.e.* $\lambda \in [300\text{ nm}, 1000\text{ nm}]$, *i.e.*

$$\max_{\substack{h\in[40\text{ nm},180\text{ nm}]\\ r\in[120\text{ nm},300\text{ nm}]}} \int_{300\text{ nm}}^{1000\text{ nm}} A(\lambda)d\lambda \tag{5.5}$$

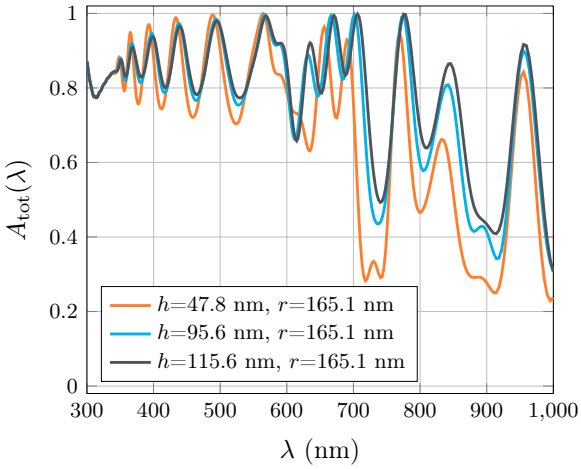


Figure 5.14 | Absorption coefficient for various textures with DGTD-$\mathbb{P}_3$ method. Computation times for each configuration are respectively 1181 sec, 1693 sec, 2017 sec.

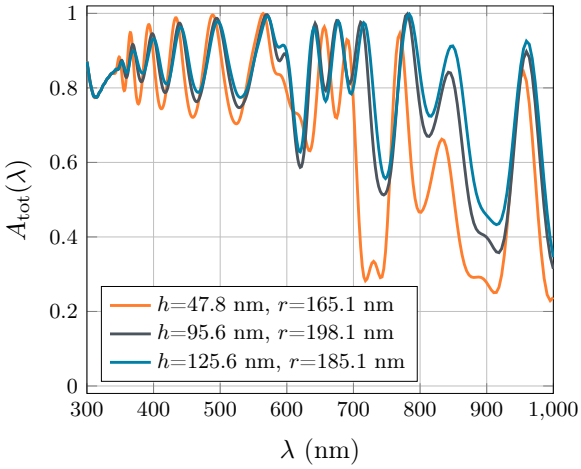
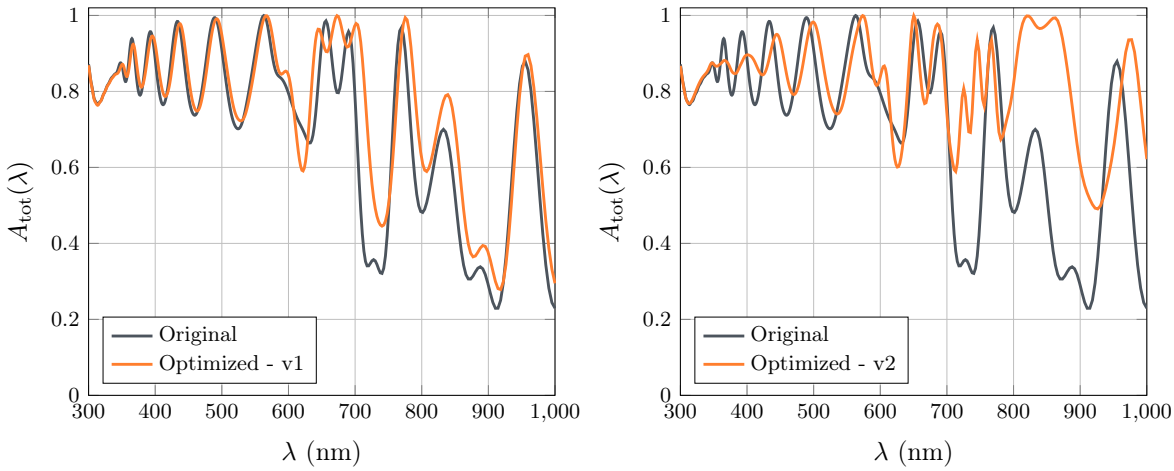


Figure 5.15 | Absorption coefficient for various textures with DGTD-$\mathbb{P}_3$ method. Computation times for each configuration are respectively 1181 sec, 3062 sec and 2572 sec.

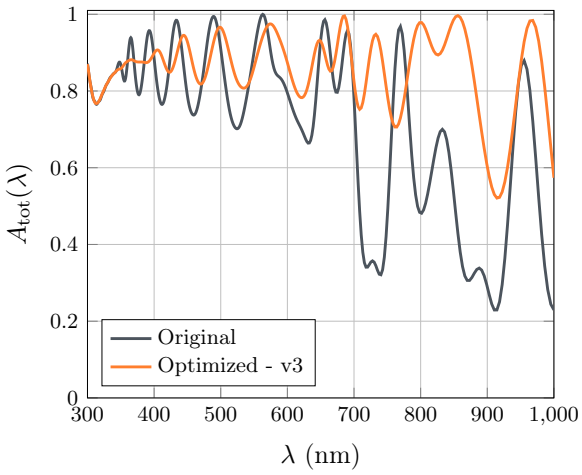
More precisely, we minimize the Euclidean norm of 500 equally spaced discrete values of the absorption in this wavelength range. The inverse design analysis has been conducted on a server equipped with 8 Intel Xeon Gold 6154 CPUs running at 3.00 GHz. The DGTD-$\mathbb{P}_3$ method has been used for all the simulations. Given the number of parameters (two), and the size of the spaces for these two parameters, the DoE database is made of 20 points. For the same reasons, the EGO algorithm is set up for 20 iterations. Three optimization scenarios are considered, each with its own accessible design spaces. We have the scenario V1 with $h \in$ [40 nm , 80 nm] and $r \in$ [120 nm , 200 nm], scenario V2 with $h \in$ [40 nm , 120 nm] and $r \in$ [120 nm , 300 nm] and scenario V3 with $h \in$ [40 nm , 180 nm] and $r \in$ [120 nm , 300 nm]. The parameter ranges are considered as constraints in the EGO algorithm. The results of these three optimization scenarios are shown in figure 5.16 (a) to 5.16 (c). For scenario V3, the simulation time for the DoE phase is 5 h 24 mn, while for the EGO phase, it is equal to 6 h 36 mn.

Remark *: The DoE phase is computed sequentially, which is not the optimal way. Indeed, one can compute all the DoE points in parallel, and then start the iterative part of the EGO. However, in our case, since the full optimization (DoE + iterations) is submitted entirely on 8 cores, it is easier to have the same behavior for the two parts, i.e. a sequential optimization process, while fullwave DGTD simulations are distributed on 8 cores.*



(a) V1: h=72.6 nm, r=198.8 nm

(b) V2: h=109.8 nm, r=265.1 nm



(c) V3: h=175.8 nm, r=192.3 nm.

Figure 5.16 | Absorption coefficient for the starting-point texture with h=47.8 nm and r=165.1 nm, and the 3 optimized textures for DGTD-$\mathbb{P}_3$ method.

For V2 and V3, we observe a significant improvement for the absorption of the red color ($\approx$ 700 nm). In the range [700 nm, 1000 nm], an improvement of almost 60% is noticed. In addition, there are no more wavelengths for which the absorption is lower than 50%. The contour lines of the amplitude of the DFT of $\mathbf{E}$ at two particular wavelengths are shown in figure 5.17. We clearly see what range of frequency each layer is targeting.

In conclusion of this study, using the EGO method from the FAMOSA library has allowed us to find better configurations both to reduce the reflection in the [700 nm, 900 nm] range and increase

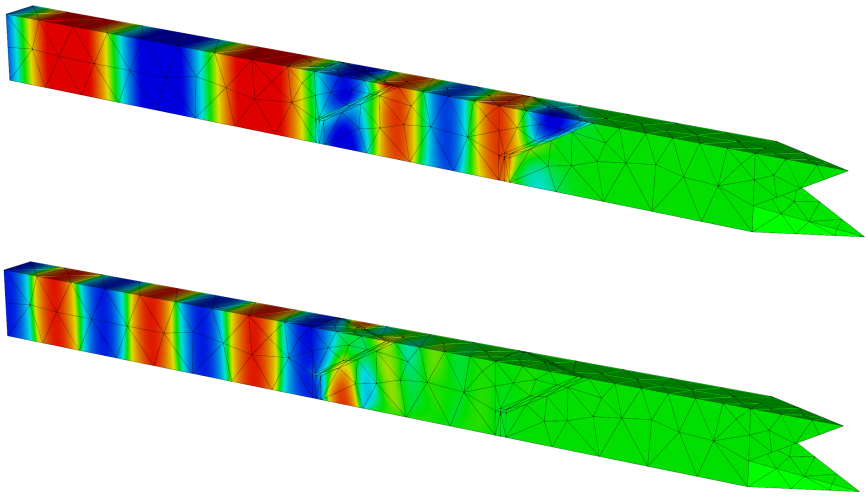


Figure 5.17 | Simulations using DGTD-$\mathbb{P}_3$ method on the optimal configuration: contour lines of the real part of E_x for a wavelength $\lambda = 850$ nm (top) and $\lambda = 400$ nm (bottom).

the overall absorption of the glass. Indeed, our figure of merit was not actually focusing on the reduction of the reflection. It could be interesting to consider this problem as a multi-objective optimization, with the second objective being the reduction of R in the range $[700\text{ nm}, 800\text{ nm}]$. Besides, a validation of the obtained optimal configuration is necessary. As we did in chapter 4 with the thin-film tandem solar cell, we could consider the exact topography for each layer, which would drastically increase the size and the complexity of the resulting geometric model. However, this comparison is necessary in order to show the reliability of the pyramidal-based simplified model.

5.3 Hot carrier pyramidal nanostructured solar cell

In the field of PV solar cells, many strategies are investigated for overcoming the Shockley-Queisser limit (defined in chapter 4). Multijunction solar cells are intensively studied for their capability of each individual junction to convert a fraction of the solar spectrum and give the record power conversion efficiency of 46%. However, producing cells reaching such values is extremely expensive, mainly due to the use of III-V materials. To reduce this cost, many studies have been performed on tandem solar cells. Hot carrier solar cell is an advanced concept which aims at overcoming the Shockley-Queisser limit with a single junction device where dissipation of the carrier energy via heat is avoided.

In this section, we consider a hot carrier solar cell proposed by researchers at C2N. The PV nanostructure considered is composed of several layers. From top to bottom: we have a 90 nm layer of Magnesium Fluoride (MgF2), then a 40 nm Zinc Sulfide (ZnS) layer. Just below, we have a 10 nm layer of constant refractive index $n = 3.3$, then the absorber layer of Indium Gallium Arsenide (InGaAs) of thickness 50 nm, another 10 nm layer of constant index. Below, we found pyramidal nanostructure, with a constant index $n = 1.5$. These pyramids are in a matrix of constant refractive index $n = 3.3$, and finally, we have a semi-infinite Ag layer at the bottom. A sketch of the cell structure is shown in figure 5.18.

Particularly in this structure, the thickness of the absorber plays a decisive role in the mechanism of the solar cell. Indeed, in order to avoid the thermalization rate, which reduces the cell

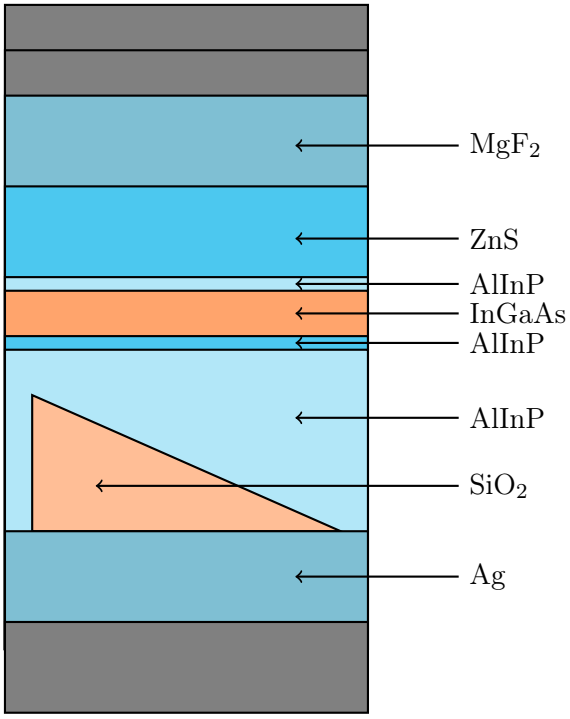


Figure 5.18 | Composition of the PV cell. Gray area corresponds to artificial volumes (PML and SF region). Orange area corresponds to both the absorber (InGaAs layer) and the nanotexture (SiO_2 pyramid).

efficiency, and improve the fast carrier collection rate, which increases the cell efficiency, it is necessary to use an ultra-thin absorber layer (less than 100 nm thick). In comparison, the previous cells (section 5.2) was composed by, at least, 400 nm absorber. However, such a thin layer drastically decreases the absorption of the cell. The main direction for overcoming this issue is to rely on an efficient light trapping strategy. In this work, we consider a back mirror nanotextured made of pyramids. This geometric model was chosen for its asymmetry, which promotes the light trapping. Indeed, in [CH15], it is proven that both the level of absorption and the number of resonances increase by symmetry breaking. The goal here is to find the perfect pyramidal configuration, such that it increases the Jsc of the solar cell. To do so, we optimize the design of pyramid's grating, which is characterized by three parameters: the height of the pyramids h, the diameter d, and the gap between two elementary patterns g. The parameters are represented in figure 5.19. In comparison with the previous study, here the asymmetry of the pattern requires the use of PBC.

5.3.1 Material models

Concerning the optical characteristics of the constituting materials, for simplicity, we only consider the absorption in the InGaAs and the parasitic absorption in Ag. The absorption in the MgF_2 and the ZnS are neglected. All the fits are represented in figure 5.20. Concerning the imaginary part of the permittivity function of the InGaAs, we can see in figure 5.21, which represents the imaginary part of the permittivity in log-scale, that the representation of the bangap is problematic. As the fitting of silicium in the previous chapter, we tried to work on the simulated annealing algorithm to improve the accuracy near the bandgap. However for this one, it was not possible to improve

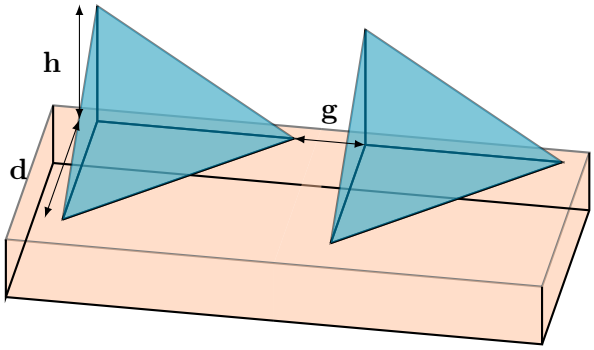


Figure 5.19 | Illustration of the three design parameters. The gap is the smallest distance between two successive pyramids.

it. It is certainly due to the very large spectrum of interest [300 nm,1600 nm], which is harder to take completely into account.

5.3.2 Comparison with FDTD results

Thanks to a preliminary multi-parametric study using a FDTD fullwave solver (Lumerical FDTD[4]), our physicist partners have proposed two potential designs: (D1) $h = 390$ nm, $d = 690$ nm and $g = 0$ nm; (D2) $h = 390$ nm, $d = 690$ nm and $g = 8$ nm. D1 and D2 correspond to the best designs obtained via the multi-parametric study.

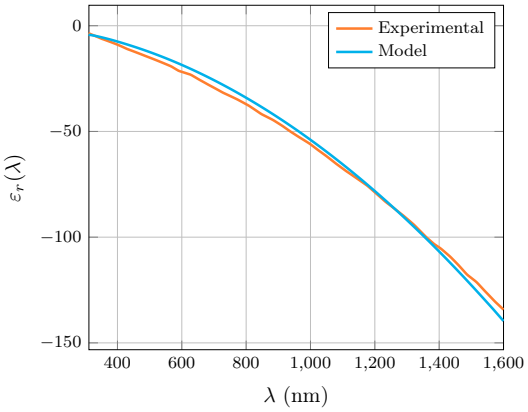
In a first step, we reproduce the FDTD results for configurations D1 and D2 with our DGTD method. We performed a convergence analysis for both configurations (see figure 5.22).

The overall absorption profile obtained with both methods matches. However, the FDTD solution is closer to the DGTD-$\mathbb{P}_2$ simulation, which is not converged. The difference between the short circuit current density (Jsc) computed with DGTD-$\mathbb{P}_2$ and DGTD-$\mathbb{P}_3$ is 12% (from 39.8 mA / cm^2 to 34.3 mA / cm^2). The other computed Jsc are in the table 5.4. We also plot two theoretical limits, the single-pass absorption and the Yablonovitch limit (defined in chapter 2) in figure 5.23.

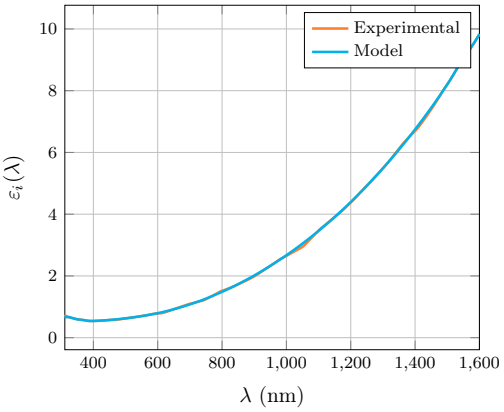
5.3.3 Parallel EGO

From now on, we only consider the PV cell geometry without a gap. One limitation of the EGO from the computational efficiency perspective is the sequential evaluation of f in the optimization step, caused by the single points given by the EI figure of merit. Ginsbourger *et al* proposed in [GLRC08] a generalized acquisition function well-suited for parallel computing. This generalization is called multi-points expected improvement, or q-EI, as it generates q design points at each iteration of the EGO algorithm. This acquisition function allows us to perform the most expensive steps of the EGO, which is the objective function evaluation in a parallel way. Assuming that a minimum number of designs are necessary to converge towards the global optimum, the use of q-EI allows us to treat q points in parallel, and thus, to update the Gaussian model with q observations rather than one. Doing so, we can speedup the enrichment phases by evaluating multiple good candidates at once, and thus significantly improve the model in one iteration. An EGO diagram with q-EI is illustrated in figure 5.24.

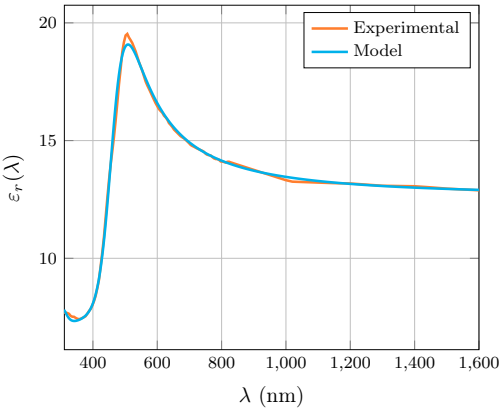
[4]https://www.lumerical.com/products/fdtd/



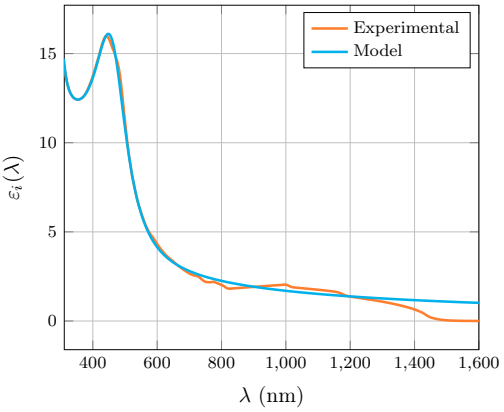
(a) Ag real part,



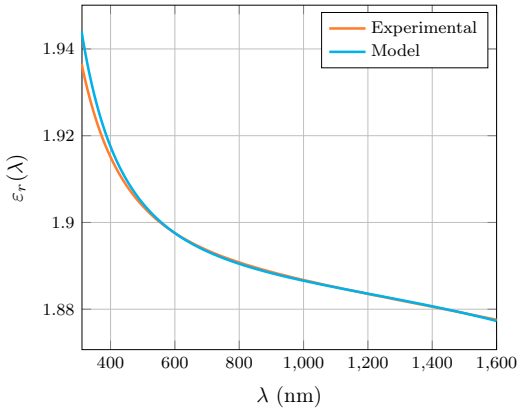
(b) Ag imaginary part,



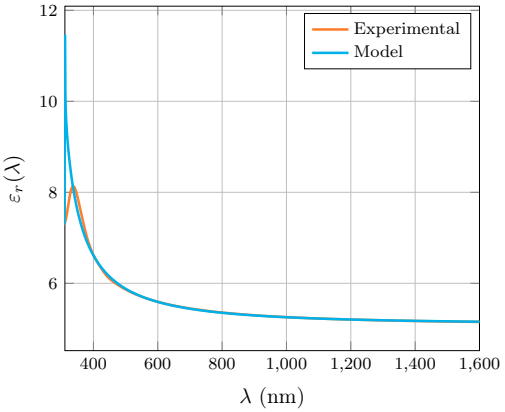
(c) InGaAs real part,



(d) InGaAs imaginary part



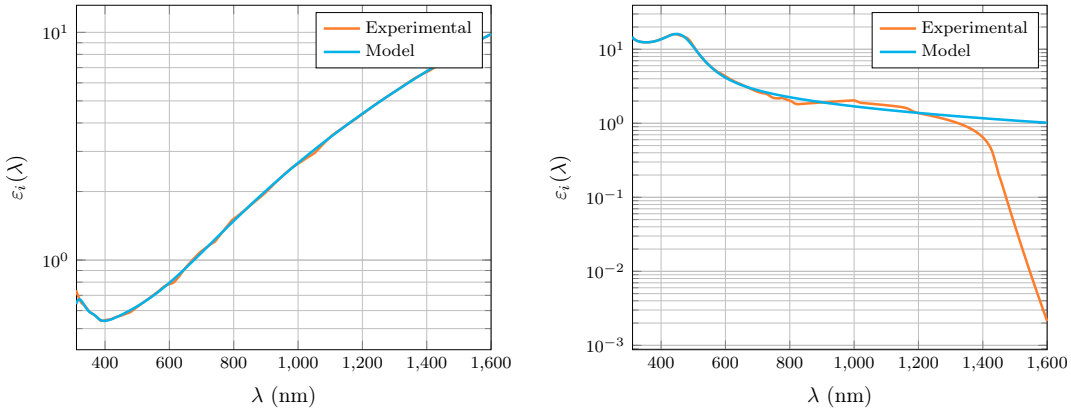
(e) MgF_2 real part,



(f) ZnS real part,

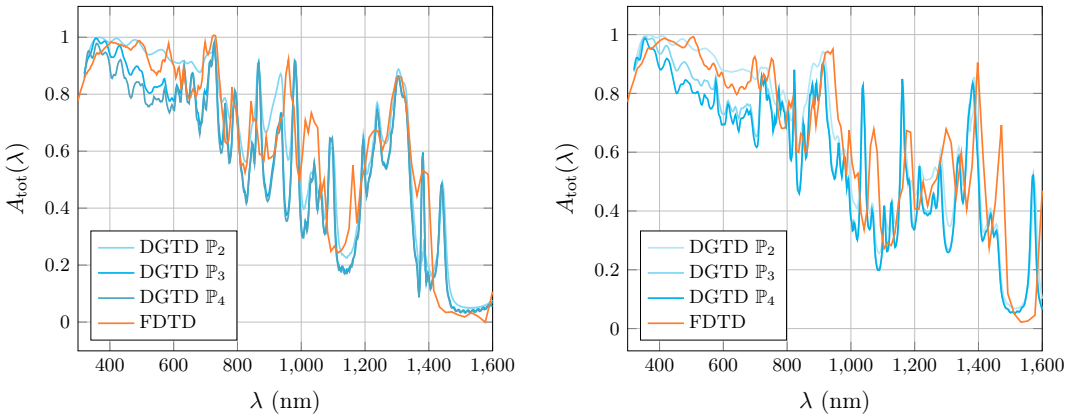
Figure 5.20 | Real and imaginary parts of the relative permittivity of Ag, InGaAs, MgF_2 and ZnS predicted by our dispersive model compared to experimental data.

We have done the optimization of the pyramidal grating again, this time using the q-EI, also available in the DiceOptim library. The difficulty in this situation is the wrapper around the DGTD solver runs. Indeed, launching several instances of the DGTD solver in parallel, knowing that the



(a) Ag imaginary part in log scale, (b) InGaAs imaginary part in semi-log scale

Figure 5.21 | Real and imaginary parts of the relative permittivity predicted by our dispersive model compared to experimental data in semi-log scale.



(a) Configuration 1 (with gap) (b) Configuration 2 (without gap)

Figure 5.22 | Convergence study on configurations D1 and D2 compared with FDTD.

solver is itself parallel, requires a careful management and distribution of computational resources.

The following test was performed on a cluster of Intel Xeon Gold 6154 CPUs running at 3.00 GHz. We start from a DoE database with 16 points, and we choose to predict and evaluate four points per iteration, over five iterations. As a comparison point, we also perform a sequential optimization in order to compare the obtained results for both methods. In order to also compare the optimization time, we take the same computation power per DGTD solver run (32 cores).

The results of different steps of the parallel EGO are shown in figure 5.25. Those maps represent the prediction of the metamodel at different updates of the model. For this structure, the maximum Jsc obtained for the optimized parameters $d = 943$ nm and $h = 877$ nm is 42.43 mA / cm^2 compared to 34.32 mA / cm^2 for the initial configuration. It corresponds to an improvement of 23%, as seen in the absorption spectra in figure 5.26. We also plot the contour lines of the amplitude of the DFT of $\mathbf{E}$ at wavelength $\lambda = 1060$ nm in figure 5.27. However, we note a nice improvement in the higher part of the spectra, which corresponds to the lack of accuracy of our InGaAs dispersion model.

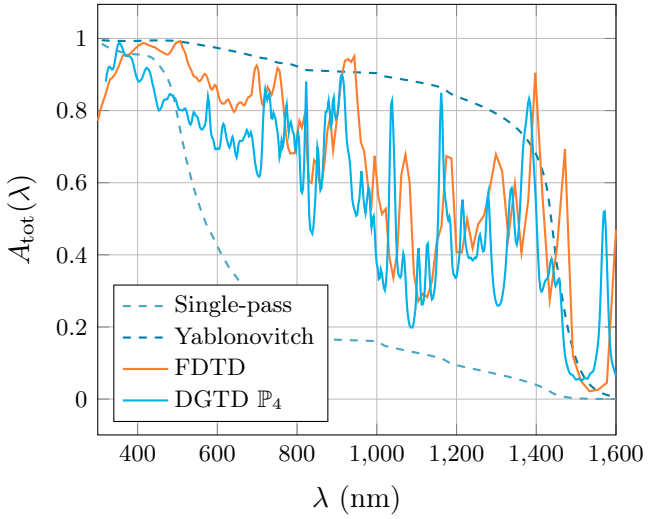


Figure 5.23 | Superposition of our results with theoritical limits.

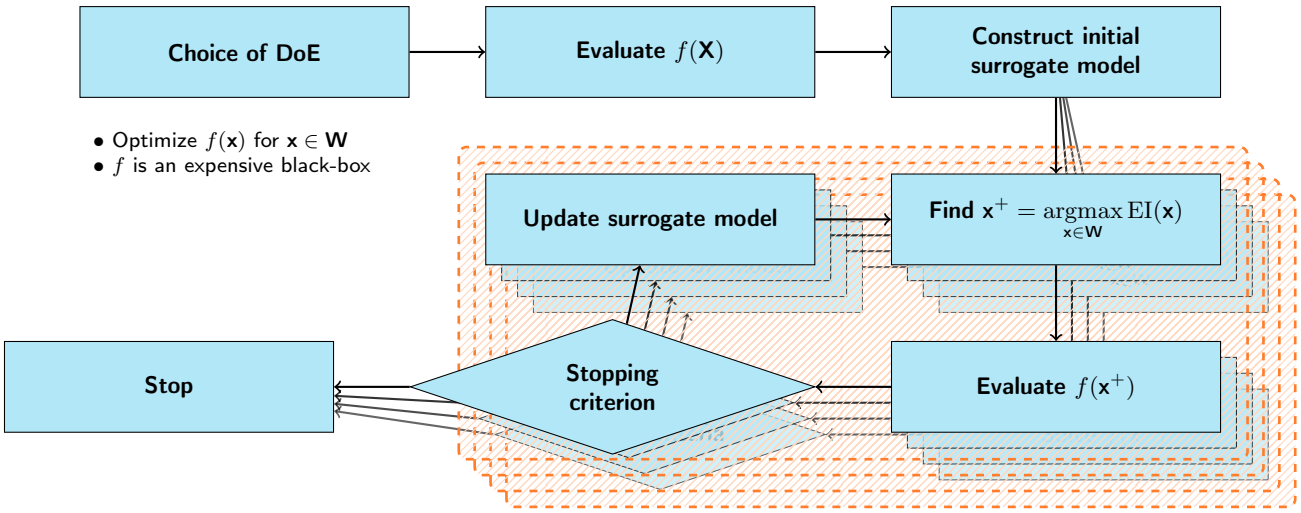
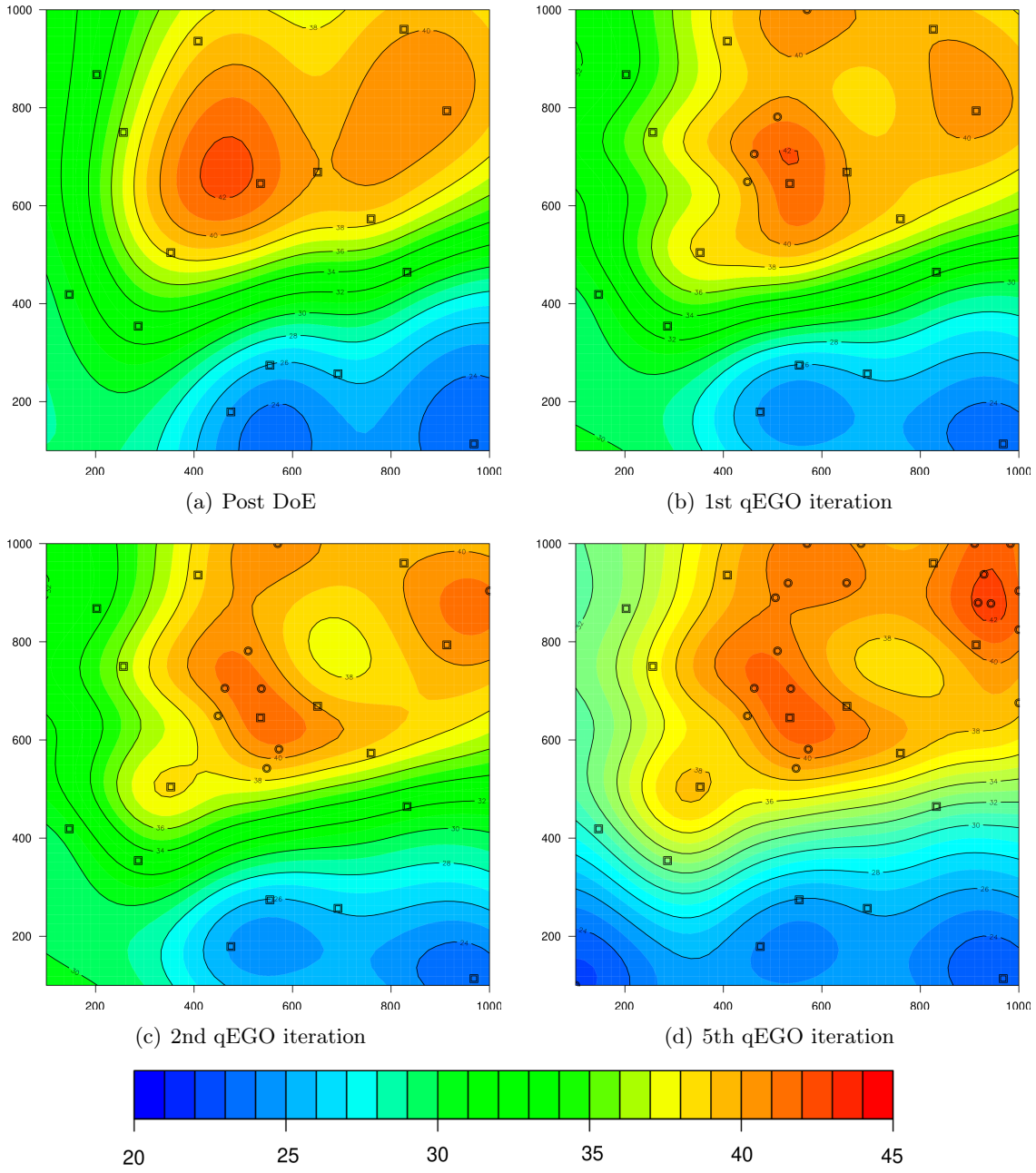


Figure 5.24 | q-EGO optimization flow diagram. The orange part represent the iterative iterative portion of the EGO method.

5.3.4 Performance

A comparison has been made between parallel and sequential optimization workflows. As we use four times more resources, we expect to have a computation time four times less important, assuming that there is no extra cost on the metamodelization side. We obtain a CPU time of 21 hours for the parallel version and 62 hours for the sequential one, which corresponds to a speedup of almost 3. We can explain this lower acceleration by a load balancing issue between concurrent evaluations of the solver. Indeed, designs produced by q-EI at each iteration can show substantial differences, resulting in entirely different computational domain sizes. The space of accessible designs being large, there can be a factor of 1000 between the volume of two structures proposed, *i.e.* $(h_{\text{pyr}}, d_{\text{pyr}}) = (100, 100)$ and $(h_{\text{pyr}}, d_{\text{pyr}}) = (1000, 1000)$. We show in figure 5.28 two designs evaluated during the last iteration. Design (a) corresponds to an exploration where



(a) Post DoE

(b) 1st qEGO iteration

(c) 2nd qEGO iteration

(d) 5th qEGO iteration

Figure 5.25 | Prediction of the Jsc from the surrogate model after DoE step and during several iterations of the parallel EGO algorithm. Squares represents the DoE samples and the circles are the point selectionned by the algorithm.

Case	Solver	Jsc (mA / cm^2)
Single-pass		16.86
Yablonovitch		47.99
Total		55.57
D1	FDTD	39.20
	DGTD-$\mathbb{P}_2$	40.30
	DGTD-$\mathbb{P}_3$	35.87
	DGTD-$\mathbb{P}_4$	34.80
D2	FDTD	38.94
	DGTD-$\mathbb{P}_2$	39.79
	DGTD-$\mathbb{P}_3$	35.46
	DGTD-$\mathbb{P}_4$	34.32

Table 5.4 | Computed Jsc over [320-1600] nm.

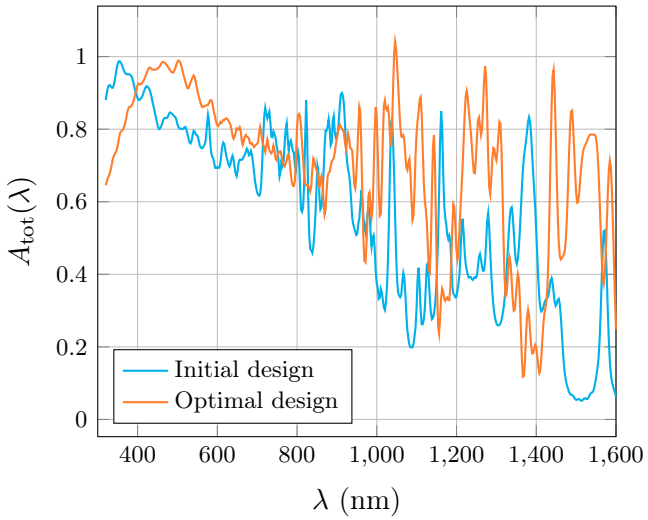


Figure 5.26 | Comparison of the initial design and the optimal one.

there has not been much research, while design (b) exploits the metamodel by searching in a potentially interesting area. The two simulation times associated with these problems are 33 sec and 11783 sec, respectively. An intelligent way to manage these problems would be to allocate computational resources not uniformly across simulations, but rather according to the complexity of the corresponding problem. Another way would be to use an asynchronous EGO algorithm, which updates the metamodel each time an evaluation of the cost function is completed.

5.4 Conclusion

In this chapter, we have presented an optimization method particularly well adapted to the typical objective function obtained by solving long simulations of DIOGENeS-DGTD. We processed two realistic PV cases with two different software environments (Famosa and DiceOptim). The first

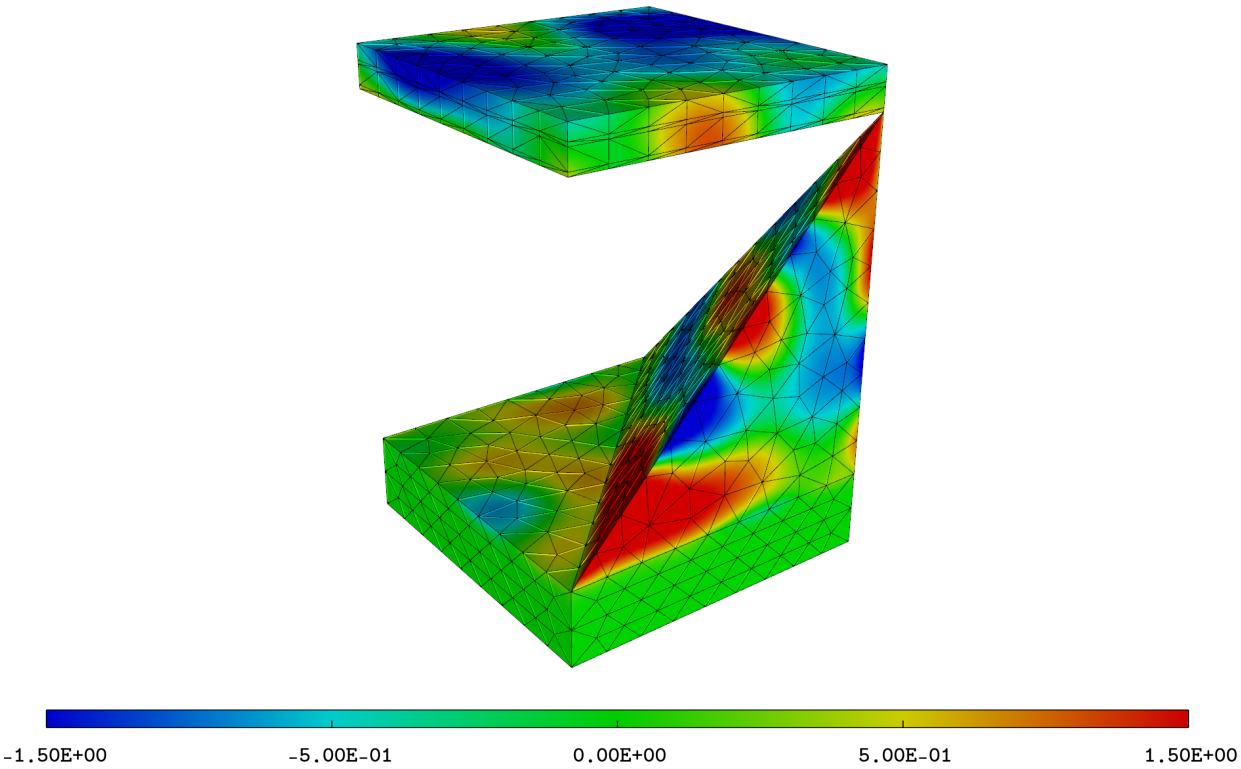
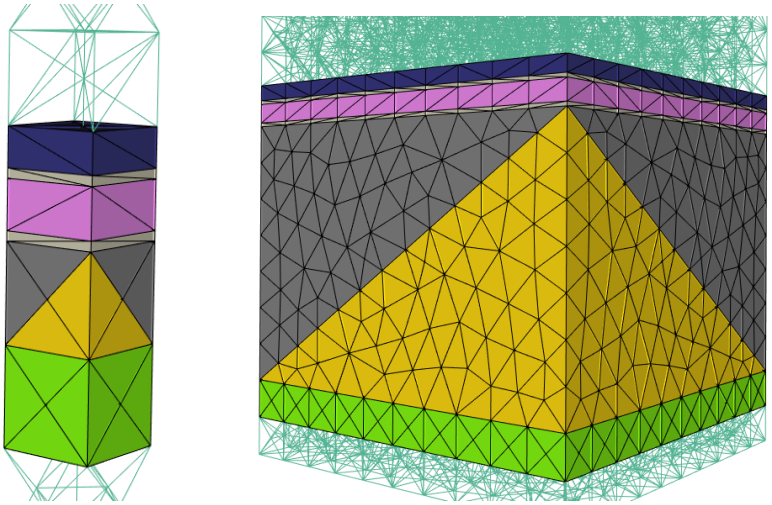


Figure 5.27 | Simulations using DGTD-$\mathbb{P}_4$ method on the optimal configuration: contour lines of the real part of E_x (bottom) for a wavelength $\lambda = 1290$ nm.



(a) Design 1 (b) Design 2

Figure 5.28 | Two designs proposed at one iteration. Design 1: $h_{\text{pyr}} = 100$ nm, $d_{\text{pyr}} = 100$ nm, 299 cells, CPU time: 31 sec. Design 2: $h_{\text{pyr}} = 679$ nm, $d_{\text{pyr}} = 1000$ nm, 14867 cells, CPU time: 11783 sec.

case consisted of a PV glass. Such technology requires a good efficiency with good aesthetic

properties. That is why it is crucial to reduce the parasitic and reddish reflections. Using the Euclidean standard as a merit function, we observed a significant improvement in the red part of the absorption spectrum. Furthermore, the general aspect of the absorption was smoother, which corresponds to an attenuation of parasitic reflections. The second case consisted of a hot carrier solar cell. To increase the absorption of the cell with respect to the Jsc, a nanotexturing tetrahedral pyramid grating was placed at the bottom of the cell for light trapping purposes. A parallel EGO method was then used to optimize the grating. This parallel method, contrary to the sequential version presented, allowed updating the metamodel through multiple evaluations at each iteration. A sub-optimal scalability was observed, which can be explained by the couples of design given by the acquisition function. One way to overcome this can be to use an asynchronous EGO algorithm or to modify the distribution of computing resources by performing a prior analysis of the complexity of the design to be simulated, for example, by looking at the number of cells for the given mesh.

6

High performance computing aspects

Contents

A recurrent concern with the numerical simulation of realistic 3D problems is the time to solution, necessary to obtain sufficiently accurate results. In many cases, the time required to obtain these solutions is not problematic. In other cases, a budget of time and computing resources can be assigned to the simulation, thereby limiting the simulation parameters (mesh size, interpolation order, physical simulation time) in order to match the budget. For example, optimization problems (as those discussed in chapter 5) need multiple full-wave solver evaluations, therefore, increasing further the wasted time in case of poor solver performances.

The aspect related to the study of the performances of numerical software, as well as its modification to improve its computational efficiency, is part of what is called high performance computing (HPC). HPC stands at the crossroads of different scientific disciplines and skillsets. For example, the discretization of a PDE system by FEM yields the resolution of linear systems. Developing efficient and scalable linear system solvers requires ingredients from numerical analysis, linear algebra, and computer science.

In general, a numerical solver is evaluated by its efficiency. There are two types of efficiencies: the first one is whether the obtained solution is accurate enough, which is most of the time, the

principal (and only) concern for scientists; the second one is the computation time required to obtain the solution, which can be utterly prohibitive in some cases. Using poorly adapted methods and very low-performance software can make some problems almost unsolvable.

Parallel computing is nowadays ubiquitous in large-scale numerical simulations of complex physical problems. As its name suggests, a parallel software is able to run on many processors in parallel. This makes it possible to use the largest and most powerful computers, often called *supercomputers*. These are usually based on a cluster of computing nodes, which are basically multiple computers connected together. Those machines can make their Central Processing Units (CPUs) work on the same problem at the same time. Multiple forms of parallelism exist in a modern supercomputer, which materialize at different levels of the software. The latter are implemented by different parallelism protocols, the most renown being MPI (for Message Passing Interface), which was intended to distributed memory systems, and OpenMP for multithreading and shared memory programming. One other benefit of MPI programming is to remove the impossibility to run an application on one system because of the memory limitation by splitting this memory across several CPUs. For instance, this can be very helpful when large linear systems of equations have to be solved.

Parallel softwares introduce a new criterion to measure software performances. Indeed, a good parallelization should, in theory, make it possible to reduce the computation time proportionally to the number of tasks used. The main issue that can appear is referred to as load imbalance. A proper load balancing consists of distributing approximately equal amounts of work among tasks so that all tasks are kept busy at the same moment. It is primordial for parallel programs when synchronization is needed. Of course, if all tasks need to communicate at one point, this operation will be performed when every task is ready, *i.e.* when the slowest one has finished its assigned work. As a result, having a perfect speedup is not straightforward, and some implementation adjustments are required to balance the computational load between each CPU. Besides, we should not neglect the overhead introduced with the parallelism, *e.g.* the communications introduced by MPI. Two types of parallelism regions can be defined depending on the ratio between communication operations and local computation: the fine grain parallelism, which corresponds to piece of codes doing small amounts of computation between communication operations, and the coarse grain parallelism, which represents part of code where the computational work is significant compared to communications.

In this chapter, we focus on the performance of the DGTD solver developed in this thesis, called DIOGENeS-DGTD, which is a component of the DIOGENeS software suite [1]. In section 6.1, we study the parallel performances of the coarse grain parallelization of DIOGENeS-DGTD, which is based on SPMD strategy combining a partitioning of the underlying mesh and a message passing programming with MPI. We highlight some load balancing issues in the DGTD time loop in the case of a realistic solar cell simulation. We also show that those load balancing issues are even more prominent for simulations on hybrid meshes. We propose a strategy to reduce these load balancing issues, which consists in an effective partitioning method based on an iterative algorithm over simulations. This allows us to find the most expensive area without any a priori knowledge. In the second part of this chapter, we present a modification of the DIOGENeS-DGTD solver to include hybrid parallelization in order to relax the constraint on the mesh partitioning when using a large number of tasks.

[1] https://diogenes.inria.fr/

6.1 Coarse grain parallelism

In the vast majority of realistic studies of light trapping in solar cells, numerical simulations cannot be performed in a reasonable time sequentially. This is due to the geometric complexity yielding computational meshes with many cells on the one hand, and to a large number of resonant modes in the structure requiring a very long physical time in order for their accurate resolution on the other hand. One efficient parallelism paradigm for discontinuous Galerkin methods is built on top of graph theory. Indeed, in order to distribute the problem on several computational nodes, the involved data structures, which are typically restrained to the mesh $\mathcal{T}_h$ for DG methods, need to be dispatched across the available resources. This is possible thanks to the partitioning of the mesh. DG discretization is known for being particularly well suited for distributed memory parallelism. This is due to the compactness of the discretization, which produces a limited amount of information exchanged between mesh partitions. Of course, the only information to be communicated between neighboring partitions is linked to the d.o.f. of the boundary cells.

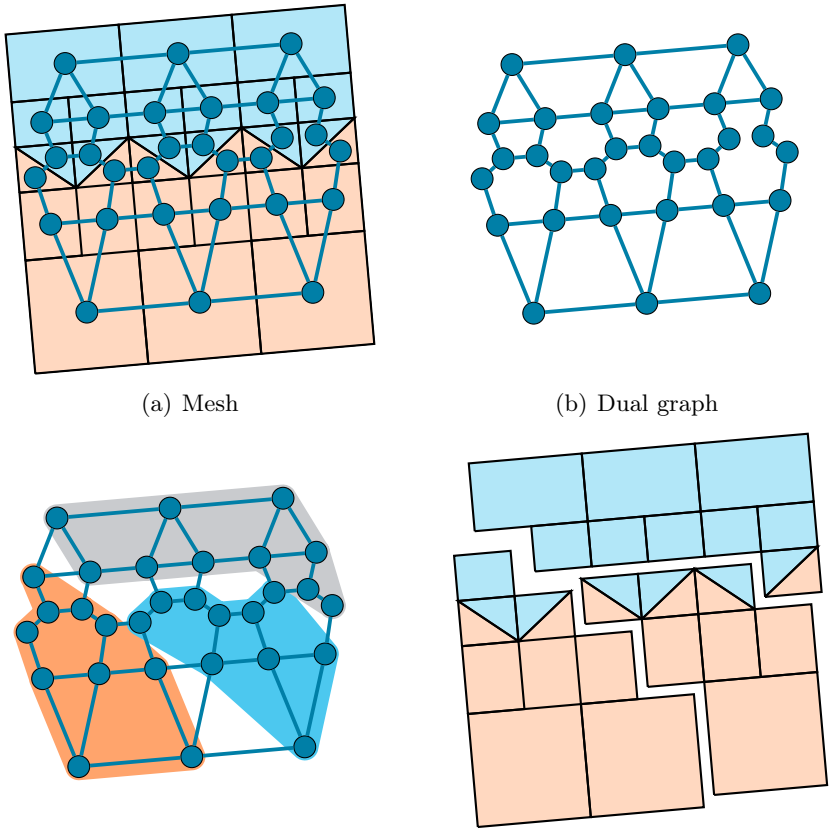
6.1.1 Mesh partitioning

We start with some notations. As in all the previous chapters, we denote the considered mesh by $\mathcal{T}_h$. Then, this mesh is split into $n, \in \mathbb{N}, n \geq 1$ submeshes, denoted by $\mathcal{T}_h^i$, $i \in \{1, ..., n\}$. In the parallel computing context, n also represents the number of cores. The number of neighbors of each $\mathcal{T}_h^i$, $i \in \{1, ..., n\}$ corresponds to the amount of inter-partition communications required to share information from one submesh to another. So it makes sense to try to get partitions that are almost equally weighted (*i.e.* with the same number of cells) while minimizing the number of neighbors. It is also important to minimize the size of the neighborhood region (which represents the memory size of the communication). Of course, an MPI parallelization is, by nature, artificially increasing the amount of d.o.f. of the problem due to more ghost cells at the inter subdomain boundaries. So minimizing the number of partition cuts is essential in order to have good performances. The main metric of parallel efficiency is the speedup. It defines how the computational time is reduced with the number of CPUs, for a fixed total problem size, *i.e.* the ratio $\frac{t \text{ on } n \text{ cores}}{t \text{ on 1 core}}$.

To do so, we start by representing $\mathcal{T}_h$ as a graph, with its elements K being the graph nodes while the faces of K are graph edges. This graph needs to be optimally partitioned, that is with an equal number of nodes within each subgraph and minimum number of partition cuts. In order to do so, we interface DIOGENeS with the MeTiS library [ARK06], which is a collection of programs that can be used for partitioning unstructured graphs both on serial as well as on parallel computers. The partitioning workflow is represented in figure 6.1.

6.1.2 Parallel load balancing

One major issue faced while dealing with realistic problems is the complexity and heterogeneity of the domain. Additionally, one simulation may involve various modeling and numerical features (*e.g.* hybrid cells, locally adapted interpolation order, PML, dispersive materials) and may also need additional computations (quantity of interests, injection of sources). Each of these ingredients represents a specific computational cost, both in terms of memory consumption and in arithmetic operations. Considering the aforementioned possible peculiarities, the risk of having some load unbalancing in the computation is high. Different strategies are possible to handle the load imbalance. The first one considered here consists of adapting the partitioning strategy. In this section, we profile DIOGENeS-DGTD on several test cases presented in chapter 4.



(a) Mesh (b) Dual graph

(c) Graph partitionning (d) Sub-meshes

Figure 6.1 | Mesh partitioning workflow. Starting from a mesh (a), its corresponding dual graph is computed. Then some partitioning algorithms are used to obtain partitions of this dual graph (c). From here, the submeshes are recovered.

We first illustrate a typical load balancing problem in our applications. In order to approach a realistic situation, we enable the computation of the reflection coefficient using the Shannon-Nyquist sampling theorem. The simulation is run for 100 iterations using a DGTD method with a $\mathbb{P}_4$ interpolation. The mesh is made of 2976 elements. It is partitioned into eight subdomains using a uniform distribution of weights associated with the nodes of the graph. The partitioned mesh is represented in figure 6.2. The subdomains have from 1 to 4 neighbors, and the number of cells deviation from the bigger submesh to the smaller submesh is 1.02. Those statistics are convincing at first.

The profiling of DIOGENeS-DGTD is done using two tools developed at the Barcelona Supercomputing Center (BSC). The first one, Extrae[2], generates traces of the execution of the program. Tracing allows us to collect low-level information, directly on the execution of functions and also communications and idle times. The other tool is Paraver[3], which is used to analyze these traces.

We focus on the time loop, which is the outer iterative part of the solver. In a realistic simulation, this loop represents 90% of the running time. We represent on figure 6.3 the 14 steps

[2]https://tools.bsc.es/extrae
[3]https://tools.bsc.es/paraver

present in an iteration of the LSRK 14-5 scheme, presented in [NDB12]. The iteration considered is an iteration for which the observed quantity is updated. As explained in chapter 4, this quantity is computed on the TF/SF interface, *i.e.* in the 6th subdomain. Since R is a frequency-dependent quantity, it does not need to be updated at each time-step according to Shannon-Nyquist sampling theorem, which states that a broadband time signal can be sampled and entirely reconstructed from its samples if the wave is sampled at least twice as fast as its highest frequency component [Sha49].

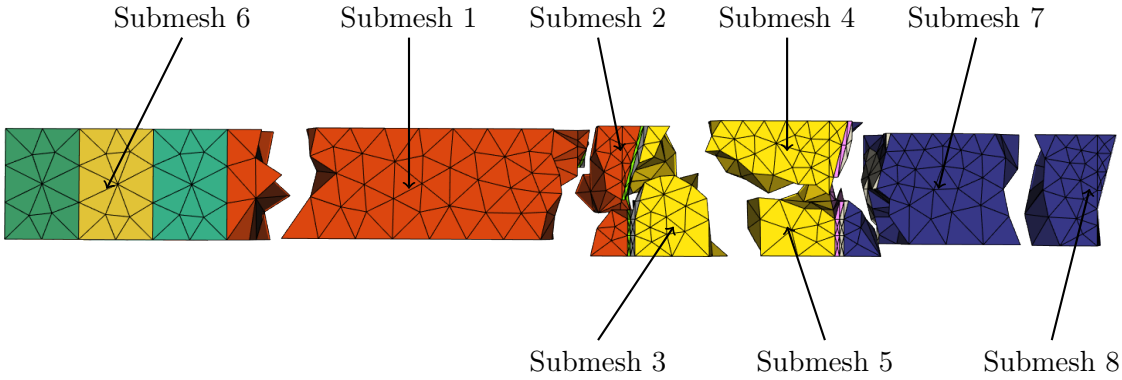


Figure 6.2 | View of partitions of the PV glass presented in secion 5.2.

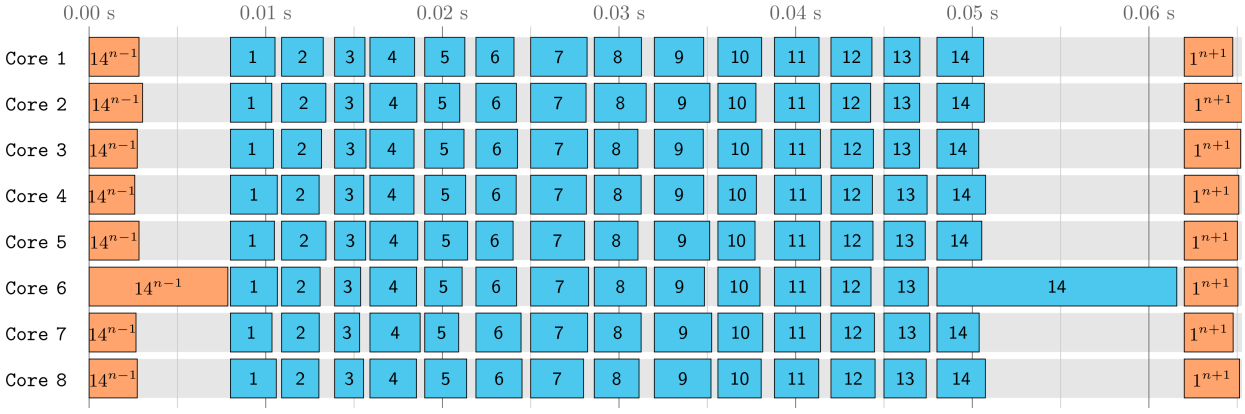


Figure 6.3 | DIOGENeS-DGTD traces: focus on the main time loop. Isolation of one iteration of DIOGENeS-DGTD with LRSK-14-5 time scheme on 8 cores. The highlighted iteration corresponds to the update of the reflection observable.

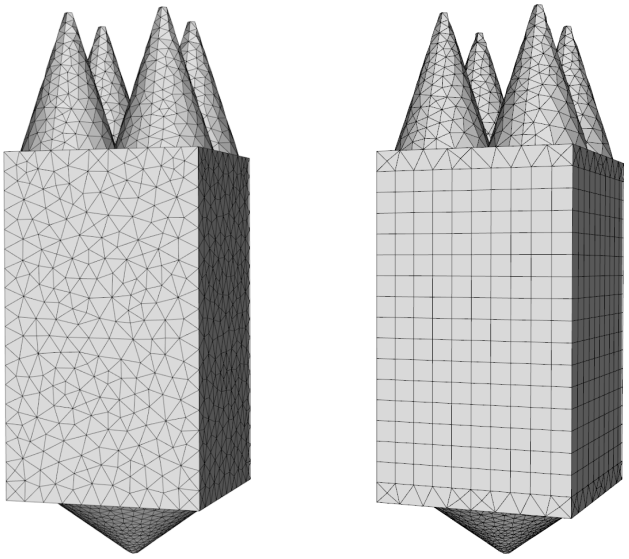
We purposely removed the communications between cores while keeping the gap between each step, which is representative of the idle time. We clearly see the synchronization at each step. We observe an imbalance in the computational time for the 6th CPU, at the last step. This is relevant to the placement of the TF/SF interface, knowing that the observable is updated on this surface in the mesh. Furthermore, we see that the iteration responsible for the observable updates (blue) is even more imbalanced.

Those load balancing issues are even more problematic when dealing with hybrid meshes. We remind some results from chapter 3, where we have shown a gain of almost 40% for the

computational time using hybrid meshes in sequential execution mode. Here we consider a smaller example from chapter 4. We consider a solar cell with the back and front texturing. In figure 6.4, we show the two meshes used to compare the scaling of DIOGENeS-DGTD on tetrahedral and on hybrid meshes. The two meshes are constituted of 33200 cells and 23000 cells, respectively.

The partitions are here built without any constraint. The parallel speedups plotted in figure 6.5 are satisfying at the beginning, with a downfall for the hybrid case. We even see that at some point, the DGTD solver on hybrid meshes is less effective than on tetrahedral meshes. Two main reasons are explaining this behavior. First, the tetrahedral mesh involves twice as many degrees of freedom as the hybrid mesh, thus improving the ratio of computing time to communication time. Secondly, communications and computations caused by hybrid faces are considerably more expensive. If the partitioning is done in this region, those heavy computations are distributed over the cores, causing a significant overhead.

It is relevant that the hybrid interfaces introduce extra computations. As seen in chapter 3, the elementary matrices must be computed on all the hybrid cells (a hybrid cell has at least one hybrid face; a hybrid face is shared by two cells from a different type). In the case of some hybrid artificial interface, extra information needs to be communicated, resulting in an extended buffer. In figure 6.6, we present performance figures for simulations based on constrained partitions, which have been obtained by setting a high penalty for the hybrid faces weights. In this way, configurations with cutting on non-conforming faces are prevented. We observe a clear improvement of the scaling, which keeps more advantageous the use of a hybrid mesh than the tetrahedral mesh.



(a) Tetrahedral mesh M1

(b) Hybrid mesh M2

Figure 6.4 | Tetrahedral (left) and hybrid cubic-tetrahedral (right) meshes of a nanostructured solar cell.

In summary, we have shown that, first, the use of hybrid meshes offers a practical interest as it allows us to accurately solve the neighborhood of the scattering objects while reducing the number of degrees of freedom in the rest of the domain, thus saving computational resources without sacrificing numerical precision. Second, the use of a non-conforming multi-element mesh

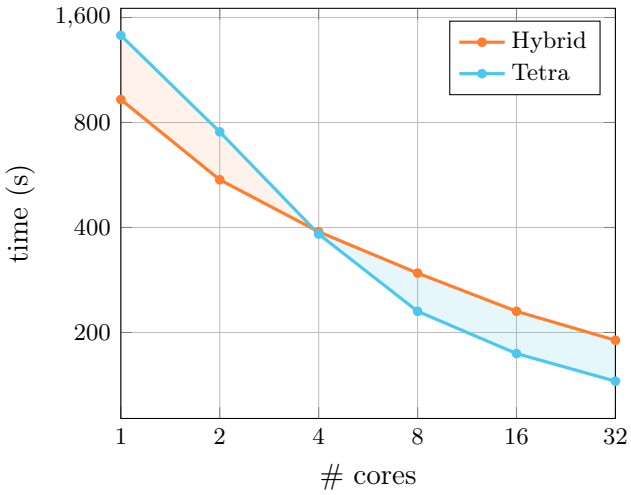
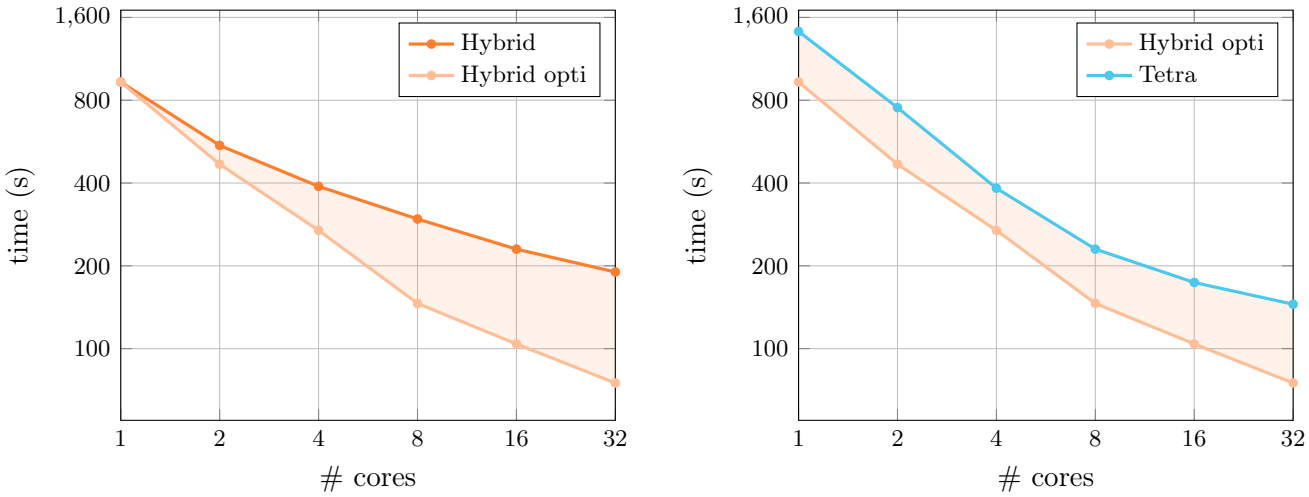


Figure 6.5 | Weak scalability comparison between a DGTD simulation using the hybrid mesh M1 and the tetrahedral mesh M2. The sequential simulation is 60% faster with the hybrid mesh, however, the parallel speedup is much worse to the point of being less interesting when the number of subdomains exceeds 4.



(a) Comparison of hybrid partitions,

(b) Comparison hybrid optimized and tetrahedal

Figure 6.6 | Weak scalability comparison between a DGTD simulation using the hybrid mesh M1 and the tetrahedral mesh M2. Impact of the optimization of the mesh partitioning.

does yield a notable degradation of the parallel performances as compared to a fully tetrahedral mesh methodology. However, the addition of a constraint at the partitioner level allows us to limit this degradation and to take advantage of the hybrid gains on parallel runs.

6.1.3 Iterative approach with advanced weighting

In the previous section, we have shown that the modification of node and edge weights can lead to improved partitions and better parallel performances. Of course, it seems logical to attach to each mesh cell a metric of computational cost, which depends on the considered physical setting. Having the perfect weight of each cell, which leads to an almost perfectly balanced partition, is not realistic. This problem can be seen as an optimization problem, with as many parameters as the mesh cells.

In this section, we focus on reducing the computational load imbalance of a DIOGENeS-DGTD simulation by modifying the cell's weight. To do so, we present an iterative approach that can enhance the repartition of weights. At first, all cells and all faces are receiving a weight. This weight can be the same everywhere, or it is possible to set an a priori weight value depending on the simulation settings, as seen in [Viq16]. From here, the mesh is transcripted to its dual graph, as explained in section 6.1.1, and partitioned using MeTiS. Then DIOGENeS-DGTD is run for a few iterations in order to get information concerning the time loop. Local timers are set to recover the computational time of each MPI process. From those times, we can calculate an average simulation time, as well as the deviation from the mean of each process. This allows us to highlight which processes were the most expensive and, as a result, in which subdomain the costly cells are located. This deviation is the key ingredient in the optimization process here. Each node of the graph, representing each element T_i, is provided with a weight noted w_i. In the partitioning step, the weights are taken into account such that the total weight of the different subdomains are as close as possible. We propose the following weight, which is compatible with any initial weight:

$$w_i^{n+1} = w_i^n + m_s^n, \tag{6.1}$$

where $m_s^n \in \mathbb{R}$ is a metric representing the time deviation in the subdomain s at the iteration n. The methodology is illustrated in figures 6.7 and 6.8.

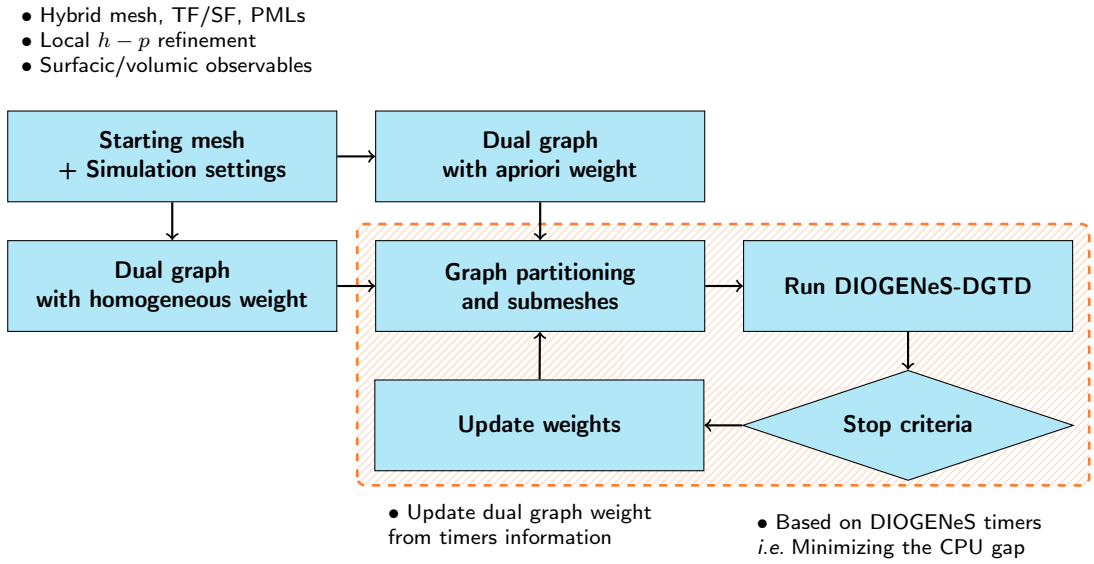
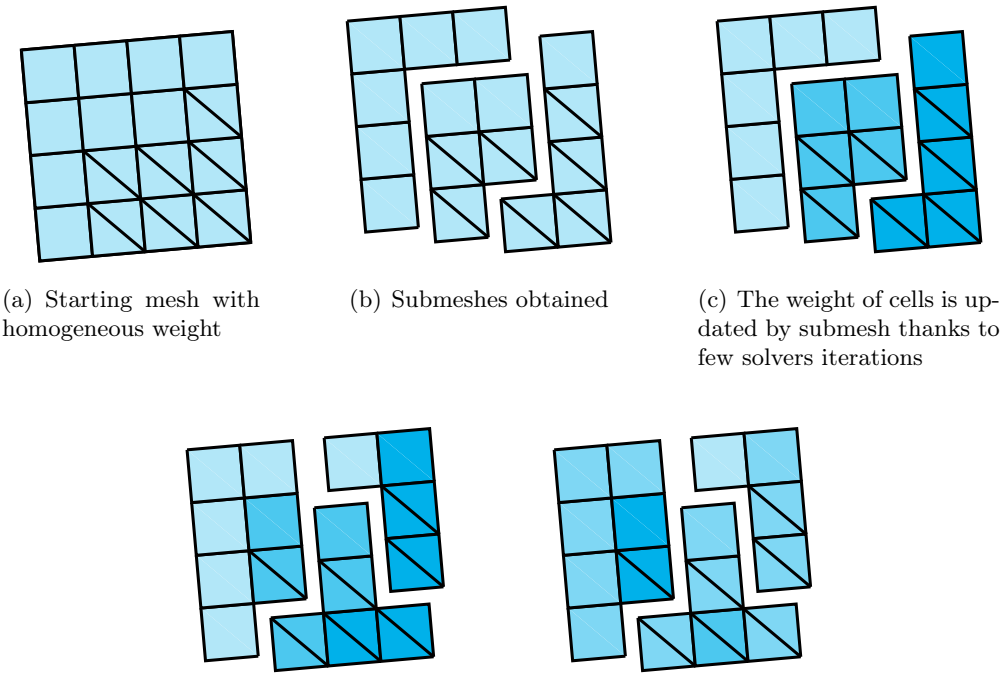


Figure 6.7 | Workflow of the iterative partitioning method.

We have applied this weighting strategy to the DIOGENeS-DGTD simulation for the solar cell with the back and front texturing, using the hybrid mesh M2 and running the solver for 100



(a) Starting mesh with homogeneous weight

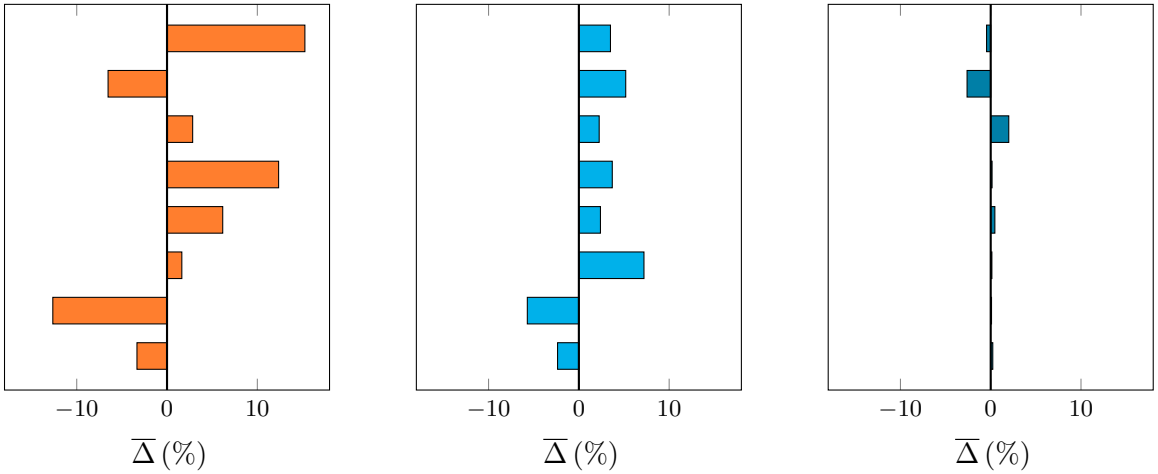
(b) Submeshes obtained

(c) The weight of cells is updated by submesh thanks to few solvers iterations

(d) New submeshes

(e) The weight of cells is updated

Figure 6.8 | Illustration of the iterative partition steps. The gradient of blue represent the weight associate to the underlying element.



(a) Original partition (b) Fifth iteration (c) Final partition

Figure 6.9 | Correction of the load imbalance using the iterative partitioning approach. Each bar corresponds to the relative imbalance of a single CPU to the average value computed over all processors. As can be seen, the balance of the CPU loads is restore over iterations, with at the end a maximum imbalance of 3%.

time iterations. We consider the case of 8 partitions. For a simpler comparison of the different partitions, we are using the relative deviation to the mean CPU time $\overline{\Delta}$ (in %) on each process. The effect of weighting is assessed by analyzing the evolution of the relative deviations from the first iteration to the tenth. The results are shown in figure 6.9. As can be seen, the load balancing is improving (from 15% to 2%).

Then, we have used this strategy on the case shown in section 4.5.1, which exhibits notable load imbalance caused by the TF/SF interface included in one subdomain. Using this strategy over 20 time iterations allows us to restore an acceptable level of load imbalance. As one can see in figure 6.10, the new partition is splitting the TF/SF interface. The function traces in figure 6.11 highlight the better overall load balancing, even if the two aforementioned partitions are still the most expensive ones. An improvement of 10% is observed for this problem.

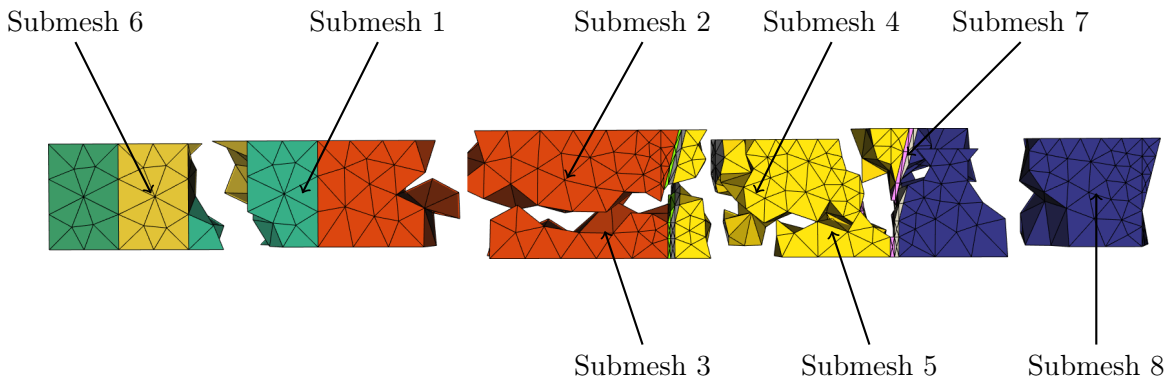


Figure 6.10 | Exploded view of partitionss of the PV glass presented in section 5.2.

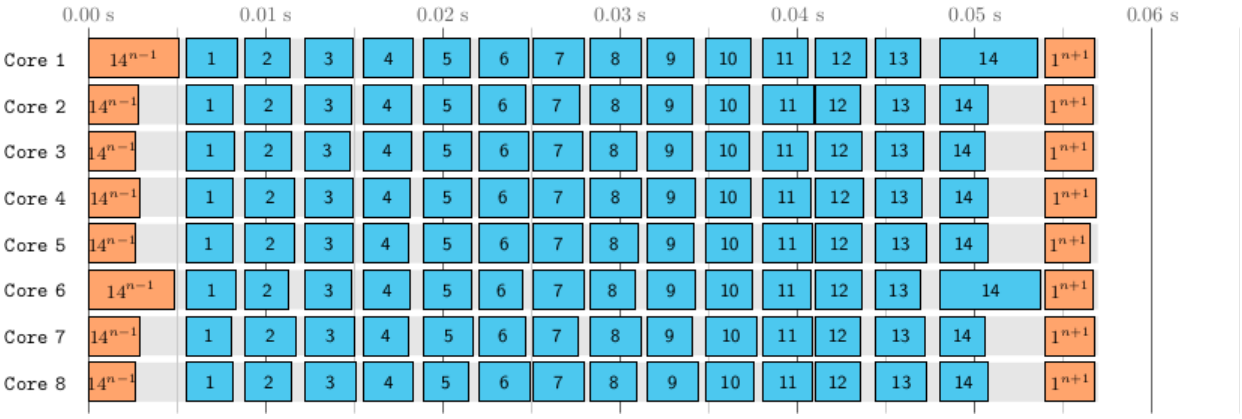


Figure 6.11 | DIOGENeS-DGTD traces: focus on the main time loop. Isolation of one iteration of DIOGENeS-DGTD with LRSK-14-5 time scheme on 8 cores. The highlighted iteration corresponds to the update of the reflection observable.

6.1.4 Strong scalability assessment

We have also performed a strong scalability analysis of the realistic thin-film solar cell presented in chapter 4 by applying the proposed DGTD solver with a tetrahedral mesh of the full solar cell model. This mesh is composed of 1,689,764 elements. This performance analysis is conducted on the Occigen PRACE system hosted by CINES in Montpellier. Each node of this system consists of two Intel Haswell E5-2690@2.6 GHz CPU, each with 12 cores. The parallel speed-up is evaluated for 1000 time iterations of the DGTD-$\mathbb{P}_k$ solver using a fourth-order low-storage Runge-Kutta time scheme. Performance results are presented in table 6.1 and 6.2. where "Elapsed" is the elapsed time, which is used for the evaluation of the parallel speed-up relatively to the first figure given for each configuration of the DGTD-$\mathbb{P}k$ method. In the first table, the timings include the calculation of an important observable quantity for the considered problem, which is the volumic absorption. The computation of this quantity requires sweep over frequencies in a target spectrum and compute on the fly during the time evolution a discrete Fourier transform of the electrical field, *i.e.* for each degree of freedom of the DG approximation of the electrical field. The later computation can be performed in a fully parallel way for each element of the mesh. However, the evaluation of the volume absorption is limited to a subvolume, *i.e.* a layer of the multilayer solar cell model; in the present case, the a-Si:H layer is selected. In figure 6.1, we have used mesh partitions with a uniform weight attached to graph nodes. Thus, we do not take into account the computational load balancing issues raised by this localization of the computation of the volume absorption. Then, the obtained parallel speed-up in table 6.1 is suboptimal and degrades when the interpolation degree k is increased. In table 6.3, we used the iterative partitioning strategy with 20 iterations. The parallel speed-up performed way better, with an improvement from 3.30 to 3.90 for the $\mathbb{P}_2$ interpolation. In addition, the overall computation time is 30% reduced. On the contrary, when the evaluation of the volume absorption is deactivated, the parallel performances are quasi-optimal, *i.e.* a linear speed-up is observed (see table 6.2). It is interesting to see how the computation of this figure of merit makes the computation way more expensive.

Solver	# cores	Elapsed	Speedup
DGTD-$\mathbb{P}_1$	96	2681 sec	1.00 (1.0)
	192	1365 sec	1.95 (2.0)
	384	768 sec	3.50 (4.0)
DGTD-$\mathbb{P}_2$	96	4364 sec	1.00 (1.0)
	192	2254 sec	1.95 (2.0)
	384	1332 sec	3.30 (4.0)
DGTD-$\mathbb{P}_3$	192	3678 sec	1.00 (1.0)
	384	2232 sec	1.65 (2.0)

Table 6.1 | Strong scalability analysis of the DGTD-$\mathbb{P}_k$ solver on the Occigen system. Tetrahedral mesh with 305,265 vertices and 1,689,764 elements. Timings for 1000 time iterations including the evaluation of the volume absorption in the Silicon layer. Execution mode: 1 MPI process per core.

Solver	# cores	Elapsed	Speedup
DGTD-$\mathbb{P}_1$	96	584 sec	1.00 (1.0)
	192	292 sec	2.00 (2.0)
	384	146 sec	4.00 (4.0)
DGTD-$\mathbb{P}_2$	96	974 sec	1.00 (1.0)
	192	490 sec	2.00 (2.0)
	384	246 sec	3.95 (4.0)
DGTD-$\mathbb{P}_3$	192	808 sec	1.00 (1.0)
	384	418 sec	1.95 (2.0)

Table 6.2 | Strong scalability analysis of the DGTD-$\mathbb{P}_k$ solver on the Occigen system. Mesh M1 (full model) with 305,265 vertices and 1,689,764 elements. Timings for 1000 time iterations excluding the evaluation of the volume absorption in the Silicon layer. Execution mode: 1 MPI process per core.

Solver	# cores	Elapsed	Speedup
DGTD-$\mathbb{P}_1$	96	2217 sec	1.00 (1.0)
	192	1114 sec	2.00 (2.0)
	384	559 sec	3.95 (4.0)
DGTD-$\mathbb{P}_2$	96	3836 sec	1.00 (1.0)
	192	1880 sec	1.95 (2.0)
	384	937 sec	3.90 (4.0)

Table 6.3 | Strong scalability analysis of the DGTD-$\mathbb{P}_k$ solver with an optimized partition. Tetrahedral mesh with 305,265 vertices and 1,689,764 elements. Timings for 1000 time iterations including the evaluation of the volume absorption in the Silicon layer. Execution mode: 1 MPI process per core.

6.2 Hybrid parallelization

Nowadays, modern computers and clusters are based on multi-core CPUs and/or on multi CPU architecture. Such computing node are not totally suited to message passing, as they work on a shared memory. Using MPI to distribute a problem on those cores adds an overhead, mainly due to inter core communications. Such architecture is way more suited to a shared-memory parallelism, which is generally implemented using the OpenMP standard. OpenMP is limited to shared memory systems. However, it offers some advantages. The main one is the incremental programming approach. Indeed, a first parallel version can be obtained by using a few OpenMP

directives in the most expensive kernels of the solver. Then, this preliminary parallelization can be incrementally extended to the remaining parts of the solver. Secondly, no communication is done at a multi-core level. Combining OpenMP with MPI can remove both the non-optimal communication on multi-core nodes, and reduce the number of mesh partitions, which naturally mitigate the risk of load imbalance.

In this section, we present preliminary results of such a hybrid parallelization, exploiting both MPI and OpenMP. Our shared memory parallelization within a multi-core node exploits OpenMP tasking, and we have decided to focus on the use of the `TASKLOOP` construct. OpenMP tasking constructs have been inserted in the most time consuming numerical kernels of the DIOGENeS-DGTD solver. These numerical kernels are concerned with:

- The computation of the right-hand side (RHS) of the system of ordinary differential equations resulting from the DG discretization in the space of system (equation (2.14));
- The update of the physical fields through time-stepping with a Low Storage Runge-Kutta (LSRK) scheme.

The corresponding module components, *i.e.* subroutines or functions, implement loops over mesh cells with a lot of local matrix-vector products.

Besides the numerical kernels for the computation of the RHS terms and the update of the physical fields through time-stepping with a LSRK scheme, the central computing loop of the DIOGENeS-DGTD solver also involves several auxiliary kernels that may account for a non-negligible part of the overall computing time of a simulation. Those kernels are concerned with the on-the-fly computation of physical observables and other figures of merit that are required for assessing the results of the simulation. They do not influence the numerical solution, but they cannot be considered as post-processing operations, as most of them need to be updated through the whole simulation. For example, although the DGTD solver computes time-varying distributions of the EM field, most of the physical observables are based on frequency-domain data. Therefore, one needs to compute a discrete Fourier transform of the EM field and related quantities. More importantly, these physical observables are most often associated with particular subvolumes or internal surfaces of the computational domain Ω. Therefore, their computation inherently introduces load balancing issues. All these auxiliary kernels have also been adapted to allow for OpenMP task parallelization.

6.2.1 Performance results

For validating and assessing the performances of the OpenMP-enabled version of the DIOGENeS-DGTD solver, we have selected two test problems of increasing size and complexity: a basic vacuum filled cubic cavity, and a thin-film solar cell. In the following tables, "DGTD-$\mathbb{P}_k$" stands for the DGTD solver relying on a uniform polynomial order of degree k. All the numerical experiments reported here have been performed on a single compute node of MareNostrum4[4], which is equipped with two sockets Intel Xeon Platinum 8160 CPU with 24 cores, each is running at 2.1 GHz. As a general rule, the number of OpenMP tasks is set to be equal to the number of threads, i.e., of the number of cores since we provide timing measures for runs using between 1 to 32 cores. Our first goal is to assess the achievable scalability with our task-based parallelization of DIOGENeS-DGTD using the OpenMP `TASKLOOP` construct.

[4]Main supercomputer in the BSC

6.2.1.1 Vacuum-filled cubic cavity.

We make use of a tetrahedral mesh with 20,500 cells. The simulated physical time is set to 10^{-8} sec. We recall that for a given mesh when the polynomial interpolation order in the DG formulation is increased, the allowable time-step size, which is subjected to a CFL-like stability condition, decreases. This is why in table 6.4, the number of time-steps increases with the interpolation order. For this model test problem, we obtain speed-up figures that are almost linear up to 16 tasks. As can be expected from the socket architecture of a compute node, the speed-up is lower when we use 32 cores.

Method	# time-steps	# OpenMP tasks	Wall time (sec)	Speedup
DGTD-$\mathbb{P}_2$	129	1	312.0	1.00
-	-	4	79.2	3.94
-	-	8	42.4	7.36
-	-	16	23.4	13.34
-	-	32	13.9	22.45
DGTD-$\mathbb{P}_3$	170	1	763.0	1.00
-	-	4	196.8	3.88
-	-	8	97.3	7.84
-	-	16	50.7	15.05
-	-	32	30.3	25.18
DGTD-$\mathbb{P}_4$	246	1	2135.0	1.00
-	-	4	541.8	3.94
-	-	8	274.1	7.79
-	-	16	141.3	15.10
-	-	32	84.3	25.32

Table 6.4 | Performance results for the vacuum-filled cubic cavity test problem.

6.2.1.2 Light trapping in a solar cell device.

We make use of a tetrahedral mesh with 41,387 cells. The simulated physical time is set to 10^{-15} sec. For this test problem, the computation of the volumic absorption in two of the six constituting material layers of the structure is impacting the achievable speed-up negatively, although the obtained figures are almost perfect up to 8 tasks, as seen in 6.5.

6.3 Conclusion

In this chapter, we have presented some profiling of the solver used in this thesis, and some improvements techniques concerning the partitions of the mesh. First, we have shown some imbalances caused by the highly longitudinal aspects of the considered mesh. The extensively expensive surfaces are usually regrouped in the same subdomains due to the common mesh partitioning strategy. This can be improved using weighted graphs, however, this results in a longer global time because of the more expensive communication. The case of hybrid meshes was also considered as it is a complexification, specifically at the non-conforming interfaces. Very bad performances were shown on parallel run with usual partitions. We showed that by avoiding a hybrid interface, splitting

Method	# time-steps	# OpenMP tasks	Wall time (sec)	Speedup
DGTD-$\mathbb{P}_1$	80	1	412.0	1.00
-	-	4	107.0	3.85
DGTD-$\mathbb{P}_2$	110	1	936.0	1.00
-	-	4	235.5	3.98
-	-	8	122.1	7.67
-	-	16	69.5	13.47
DGTD-$\mathbb{P}_3$	138	4	483.8	1.00
-	-	8	253.2	1.91
-	-	16	143.0	3.38
DGTD-$\mathbb{P}_4$	179	4	1124.5	1.00
-	-	8	568.1	1.98
-	-	16	323.5	3.48
-	-	32	221.0	5.10

Table 6.5 | Performance results for the solar cell device test problem.

results in an almost perfect speed-up. We also gave some information about the simulation of the realistic PV case from chapter 4, and in particular, times of each simulation. In the last part, we showed some results using a hybrid parallelization. This allowed to reduce the stress of the mesh due to much partition. Some tests need to be done for more complex cases, with hybrid meshes and p adaptivity. This work was granted access to the HPC resources of CINES under the allocation 2019-A0060610263 attributed by GENCI (Grand Equipement National de Calcul Intensif)

7

Multiscale Hybrid–Mixed method for nanophotonics

Contents

Most of the cases considered in nanophotonics and photovoltaics involve highly heterogeneous materials, with very fine geometric details compared to the size of the domain of interest. Such real-life configurations represent a big challenge for numerical methods because of computational simulations. In a broad sense, discretization methods have to use very fine meshes to represent the smallest geometric details. Moreover, very localized phenomena can lead to high-frequency field distribution which requires the use of refined meshes with also high-order approximation.

In recent years, discretization methods suitable for multiscale problems have been studied extensively. As we have shown in chapter 2 and 3, DG methods are flexible, both from the meshing and from the high-order approximation point of view. It makes it possible to take into account strong heterogeneity of the physical parameters (classically the permittivity ε), as well as very small and curved geometries [VL16]. From the same family of methods, there is the Hybrid Discontinuous Galerkin (HDG) method presented in [CDL] for Maxwell. One of the key specificity of the HDG method is the splitting of the problem into a global-local formulation. The reason to do so is to make the method globally less expensive by decoupling the volumic degrees of freedom. The counterpart to this is that it requires to solve a linear system at each time-step. Other methods built on top of homogenization techniques such as HMM [HOV16] can solve complex heterogeneous problems on coarse meshes. We can also refer to the HHO method [DPT17], which shares much similitude in the construction phase with the MHM method studied in the following.

In this chapter, we focus on the MHM method, first presented in [AHPV13] for an elliptic model and extented for Darcy's equation in [HMV16]. It was recently proposed for Maxwell equations in time-domain in [LPSV18]. The fundamental idea of the MHM is to find a way to recover information lost through a rough meshing of the domain. The key ingredients to do so are: (i) the splitting of the initial problem into a global-local formulation; (ii) the construction of the so-called multiscale basis functions that upscale the information from a fine-scale to a coarse scale. Those ingredients together make the MHM method effective, easily parallelizable, and robust since everything is established on a coarse partition.

The efficiency of explicit methods is always suffering when the geometry of the structures requires a high mesh refinement. One possibility is to use local time-step (LTS) methods to split our problems in different regions, each of them using its ideal time-step. The MHM naturally handles this decomposition step. To benefit from it, we develop a new formulation of the MHM method for Maxwell's equation that incorporates the possibility of multiscale time-step (MTS). We extend the results presented in [LPSV18]. In the first part, we present the model and the formulation with a MTS scheme. Then we establish and study energy preservation. Finally, we made two-dimensional numerical tests with comparisons to a standard DGTD method.

7.1 Model and preliminaries

In this chapter, we are still considering the system of Maxwell's equations in 3D introduced in chapter 1. We will now start by defining the notations, some of which are specific to the MHM method.

7.1.1 Spatial discretization

The MHM method starts from a discretization of the given spatial domain Ω. We denote by $\mathcal{T}_H$ the partition of Ω, formed by elements K of maximal size $H > 0$. From now on, we will denote by

$\mathcal{T}_H$ the coarse mesh or first level mesh and the elements K are denoted by *macro element*. This subtlety will be important in the second part of this chapter. We also note by $\partial\mathcal{T}_H$ the skeleton of $\mathcal{T}_H$. One can find an illustration of those definitions in figure 7.1.

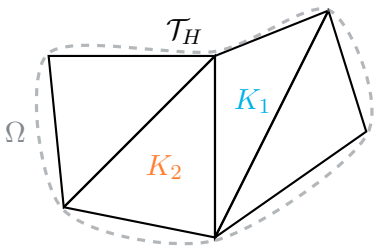


Figure 7.1 | Space discretization. First level space discretization. The domain Ω is split into coarse macroelements K.

Remark *: This partition does not need to fit the geometric details present in* Ω.

7.1.2 Time discretization

The second step consists of the coarse time discretization of the time interval $[0, T]$. For that we define $\mathcal{T}_{\Delta t}$ the partition made of the N intervals I_n with $I_n := [t_{n-1}, t_n] \in \mathcal{T}_{\Delta t}$, where $N \in \mathbb{N}^*$ and $n \in \{1, ..., N\}$. We also define the coarse time-step associated to the interval I_n as $\Delta t^n = t_n - t_{n-1} > 0$.

Remark *: This partition may be non-uniform, that is why we keep the exponent* n *on the time-step* Δt^n.

One can find an illustration of those definitions in figure 7.2

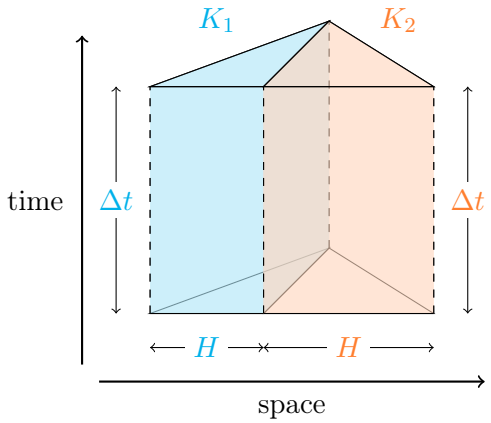


Figure 7.2 | Time discretization. First level time discretization. The time interval $[0, T]$ is split in several timeslab I_n. The blue H is the maximum size of K_1, the orange H is the maximum size of K_2.

We denote by $\mathbf{V}$ the broken $\mathbf{H}^1$ space defined as

$$\mathbf{V} := \left\{ \mathbf{v} \in \mathbf{L}^2(\Omega) : \mathbf{v}|_K \in \mathbf{H}^1(K) \text{ for all } K \in \mathcal{T}_H \right\} \tag{7.1}$$

with the following scalar product

$$(\mathbf{u},\mathbf{v})_{\mathbf{v}} := \sum_{K\in\mathcal{T}_H}(\mathbf{u},\mathbf{v})_K + \sum_{K\in\mathcal{T}_H}(\nabla\mathbf{u},\nabla\mathbf{v})_K \tag{7.2}$$

From here we can write the classical system of Maxwell's equations in a local weak formulation as follows.
For each $n\in\{1,...,N\}$, find $(\mathbf{E}^n,\mathbf{H}^n)\in\mathcal{C}^1(I_n,\mathbf{V})\times\mathcal{C}^1(I_n,\mathbf{V})$ such that $\forall t\in I_n$:

$$\begin{cases} (\varepsilon\partial_t\mathbf{E}^n,\mathbf{v})_K - (\nabla\times\mathbf{H}^n,\mathbf{v})_K & = (\mathbf{J},\mathbf{v})_K, & \forall\mathbf{v}\in\mathbf{V} \\ (\mu\partial_t\mathbf{H}^n,\mathbf{w})_K + (\nabla\times\mathbf{E}^n,\mathbf{w})_K & = 0, & \forall\mathbf{w}\in\mathbf{V} \\ \mathbf{E}^n(t_{n-1}) & = \mathbf{E}^{n-1}(t_{n-1}) & \\ \mathbf{H}^n(t_{n-1}) & = \mathbf{H}^{n-1}(t_{n-1}) & \end{cases} \tag{7.3}$$

with the initial conditions $(\mathbf{E}^0(0),\mathbf{H}^0(0)) = (\mathbf{E}^0,\mathbf{H}^0)$. We refer to [LPSV18] for the motivation of the use of this space $\mathbf{V}$.

Remark *: Please note that $\forall n\in\{1,...,N\}$, $(\boldsymbol{E}^n,\boldsymbol{H}^n)$ defined in (7.3) are smooth functions of time.*

7.1.3 Hybridization and global-local formulation

Next, the continuity of the variables $(\mathbf{E},\mathbf{H})$ is relaxed on the set of boundaries ∂K for all $K\in\mathcal{T}_H$. Instead, a weak continuity of the tangential component of the electric field $\mathbf{E}$ is imposed on $\partial\mathcal{T}_H$. We propose the following hybrid weak formulation, originally proposed by Raviart and Thomas in [RT77] and already used in [LPSV18].
Here we define Λ as the space of the restriction of the tangential component of functions in $H(curl;\Omega)$ to the boundaries ∂K *i.e.*

$$\boldsymbol{\Lambda} := \left\{\mathbf{v}\times\mathbf{n}^K|_{\partial K}\in\mathbf{H}^{-\frac{1}{2}}(\partial K)\,\text{for all}\,K\in\mathcal{T}_H : \mathbf{v}\in H(\text{curl};\Omega)\right\} \tag{7.4}$$

with $\mathbf{n}^K$ the outward normal vector to K. For each $I_n\in\mathcal{T}_{\Delta t}$, $n\in\{1,...,N\}$, find $(\mathbf{E}^n,\mathbf{H}^n,\boldsymbol{\lambda}^n)\in\mathcal{C}^1(I_n;\mathbf{V})\times\mathcal{C}^1(I_n;\mathbf{V})\times\mathcal{C}^0(I_n;\boldsymbol{\Lambda})$ such that $\forall t\in I_n$:

$$\begin{cases} (\varepsilon\partial_t\mathbf{E}^n,\mathbf{v})_K - (\mathbf{H}^n,\nabla\times\mathbf{v})_K & = (\mathbf{J},\mathbf{v})_K - (\boldsymbol{\lambda}^n,\mathbf{v})_{\partial K}, & \forall\mathbf{v}\in\mathbf{V} \\ (\mu\partial_t\mathbf{H}^n,\mathbf{w})_K + (\nabla\times\mathbf{E}^n,\mathbf{w})_K & = 0, & \forall\mathbf{w}\in\mathbf{V} \\ (\boldsymbol{\nu},\mathbf{E}^n)_{\partial\mathcal{T}_H} & = 0, & \forall\boldsymbol{\nu}\in\boldsymbol{\Lambda} \end{cases} \tag{7.5}$$

with the strong continuity in time $\mathbf{E}^n(t_{n-1}) = \mathbf{E}^{n-1}(t_{n-1})$ and $\mathbf{H}^n(t_{n-1}) = \mathbf{H}^{n-1}(t_{n-1})$. We call λ^n the hybrid variable on I_n. We see that the first two equations of (7.5) are defined locally on each macro element K while the third one is defined on the entire mesh skeleton $\partial\mathcal{T}_H$.

Remark *: This separation into the global part, and the local part is one of the key aspects of the MHM method. The computational benefits of the method result from this step.*

7.1.4 Splitting of electromagnetics fields

We use the linearity of Maxwell's operator to decompose the electromagnetic field $(\mathbf{E},\mathbf{H})$ in two parts: one coming from the external current $\mathbf{J}$, and the other one from the hybrid variable λ. We thus write for all $n\in\{1,...,N\}$

$$\mathbf{E}^n = \mathbf{E}^{n,\mathbf{J}} + \mathbf{E}^{n,\boldsymbol{\lambda}} \qquad \text{and} \qquad \mathbf{H}^n = \mathbf{H}^{n,\mathbf{J}} + \mathbf{H}^{n,\boldsymbol{\lambda}} \tag{7.6}$$

where, $(\mathbf{E}^{n,\mathbf{J}}, \mathbf{H}^{n,\mathbf{J}})$ and $(\mathbf{E}^{n,\boldsymbol{\lambda}}, \mathbf{H}^{n,\boldsymbol{\lambda}})$ are respectively solutions of the following local Maxwell problems $\forall t \in I_n$:

$$\begin{cases} \left(\varepsilon \partial_t \mathbf{E}^{n,\mathbf{J}}, \mathbf{v}\right)_K - \left(\mathbf{H}^{n,\mathbf{J}}, \nabla \times \mathbf{v}\right)_K & = (\mathbf{J}, \mathbf{v})_K, \qquad \forall \mathbf{v} \in \mathbf{V} \\ \left(\mu \partial_t \mathbf{H}^{n,\mathbf{J}}, \mathbf{w}\right)_K + \left(\nabla \times \mathbf{E}^{n,\mathbf{J}}, \mathbf{w}\right)_K & = 0, \qquad \forall \mathbf{w} \in \mathbf{V} \\ \mathbf{E}^{n,\mathbf{J}}(t_{n-1}) & = \mathbf{E}^{n-1}(t_{n-1}) \\ \mathbf{H}^{n,\mathbf{J}}(t_{n-1}) & = \mathbf{H}^{n-1}(t_{n-1}) \end{cases} \tag{7.7}$$

with $\mathbf{E}^{n,\mathbf{J}}(t_{n-1}) = \mathbf{E}^{n-1}(t_{n-1})$ and $\mathbf{H}^{n,\mathbf{J}}(t_{n-1}) = \mathbf{H}^{n-1}(t_{n-1})$ and

$$\begin{cases} \left(\varepsilon \partial_t \mathbf{E}^{n,\boldsymbol{\lambda}}, \mathbf{v}\right)_K - \left(\mathbf{H}^{n,\boldsymbol{\lambda}}, \nabla \times \mathbf{v}\right)_K & = -(\boldsymbol{\lambda}^n, \mathbf{v})_{\partial K}, \quad \forall \mathbf{v} \in \mathbf{V} \\ \left(\mu \partial_t \mathbf{H}^{n,\boldsymbol{\lambda}}, \mathbf{w}\right)_K + \left(\nabla \times \mathbf{E}^{n,\boldsymbol{\lambda}}, \mathbf{w}\right)_K & = 0, \qquad \forall \mathbf{w} \in \mathbf{V} \\ \mathbf{E}^{n,\boldsymbol{\lambda}}(t_{n-1}) & = 0 \\ \mathbf{H}^{n,\boldsymbol{\lambda}}(t_{n-1}) & = 0 \end{cases} \tag{7.8}$$

With respectively the following initial conditions:

$$\begin{array}{llll} \mathbf{E}^{0,\mathbf{J}}(0) = \mathbf{E}^0 & \text{and} & \mathbf{H}^{0,\mathbf{J}}(0) = \mathbf{H}^0 \\ \mathbf{E}^{0,\boldsymbol{\lambda}}(0) = 0 & \text{and} & \mathbf{H}^{0,\boldsymbol{\lambda}}(0) = 0 \end{array} . \tag{7.9}$$

Remark *: We note that the choice of putting all the contributions from the previous I_n in the $\boldsymbol{J}$ part as seen in equation* (7.9) *will have a significant impact on the computational efficiency of the method as it allows to compute problem* (7.8) *once for all.*

Owing to this splitting, the third equation becomes a one-field face formulation for the Lagrange multiplier $\boldsymbol{\lambda}$ as follows:
For each $I_n \in \mathcal{T}_{\Delta t}$, $n \in \{1, ..., N\}$, find $\boldsymbol{\lambda}^n \in \boldsymbol{\Lambda}$ such that $\forall t \in I_n$:

$$\left(\boldsymbol{\nu}, \mathbf{E}^{n,\boldsymbol{\lambda}}\right)_{\partial \mathcal{T}_H} = -\left(\boldsymbol{\nu}, \mathbf{E}^{n,\mathbf{J}}\right)_{\partial \mathcal{T}_H}, \qquad \forall \boldsymbol{\nu} \in \boldsymbol{\Lambda}, \tag{7.10}$$

We precise that when $\boldsymbol{\lambda}$ is used in the exponent of $\mathbf{E}^{n,\boldsymbol{\lambda}}$, one should see $\mathbf{E}^{n,\boldsymbol{\lambda}^n}$. The additional exponent is omitted here for the sake of readability.

7.2 One-level MHM discretization

We start with the space-time discretization of the Lagrange multiplier $\boldsymbol{\lambda}$. We called this step the one-level or first-level discretization.

7.2.1 Discretization of the hybrid variable

We choose a finite dimensional space to approximate the variables in $\mathcal{C}^0(I_n; \boldsymbol{\Lambda})$. We are especially looking for $\boldsymbol{\lambda}_H(t)$ in such a space satisfying (7.10). In this work, we choose to take the Lagrange multipliers $\boldsymbol{\lambda}_H$ constant in time per time-slab I_n *i.e.* $\boldsymbol{\lambda}_H \in \mathbb{P}_0(I_n), n \in \{1, ..., N\}$ and polynomial on $\mathcal{T}_H$, *i.e.* $\boldsymbol{\lambda}_H \in \mathbb{P}_0(I_n) \otimes \boldsymbol{\Lambda}_H$ with:

$$\boldsymbol{\Lambda}_H := \left\{ \mathbf{v}_H \in \boldsymbol{\Lambda} : \mathbf{v}_H|_F \in \mathbb{P}^m(F)^3 \text{ for all } F \in \partial\mathcal{T}_H \right\}, \quad m \in \mathbb{N}, \tag{7.11}$$

and look for $\boldsymbol{\lambda}_H^n$ in $\boldsymbol{\Lambda}_H$. Similarly, equation (7.6) becomes a discrete decomposition as:

$$\mathbf{E}_H^n = \mathbf{E}^{n,\mathbf{J}} + \mathbf{E}^{n,\boldsymbol{\lambda}_H} \qquad \text{and} \qquad \mathbf{H}_H^n = \mathbf{H}^{n,\mathbf{J}} + \mathbf{H}^{n,\boldsymbol{\lambda}_H} \tag{7.12}$$

From here equation (7.8) becomes: Find $(\mathbf{E}^{n,\boldsymbol{\lambda}_H}, \mathbf{H}^{n,\boldsymbol{\lambda}_H}, \boldsymbol{\lambda}_H^n)$, $n \in \{1, ..., N\}$ in $(\mathbf{V}, \mathbf{V}, \boldsymbol{\Lambda}_H)$ such that $\forall t \in I_n$:

$$\begin{cases} \left(\varepsilon \partial_t \mathbf{E}^{n,\boldsymbol{\lambda}_H}, \mathbf{v}\right)_K - \left(\mathbf{H}^{n,\boldsymbol{\lambda}_H}, \nabla \times \mathbf{v}\right)_K & = -\left(\boldsymbol{\lambda}_H^n, \mathbf{v}\right)_{\partial K}, \quad \forall \mathbf{v} \in \mathbf{V} \\ \left(\mu \partial_t \mathbf{H}^{n,\boldsymbol{\lambda}_H}, \mathbf{w}\right)_K + \left(\nabla \times \mathbf{E}^{n,\boldsymbol{\lambda}_H}, \mathbf{w}\right)_K & = 0, \quad \forall \mathbf{w} \in \mathbf{V} \\ \mathbf{E}^{n,\boldsymbol{\lambda}_H}(t_{n-1}) & = 0 \\ \mathbf{H}^{n,\boldsymbol{\lambda}_H}(t_{n-1}) & = 0 \end{cases} \tag{7.13}$$

with the initial conditions:

$$\mathbf{E}^{0,\boldsymbol{\lambda}_H} = 0 \qquad \text{and} \qquad \mathbf{H}^{0,\boldsymbol{\lambda}_H} = 0 \tag{7.14}$$

and the global problem: For each $I_n \in \mathcal{T}_{\Delta t}$, $n \in \{1, ..., N\}$, find $\boldsymbol{\lambda}^n \in \boldsymbol{\Lambda}$ such that $\forall t \in I_n$:

$$\left(\boldsymbol{\nu}, \mathbf{E}^{n,\boldsymbol{\lambda}_H}\right)_{\partial\mathcal{T}_H} = -\left(\boldsymbol{\nu}, \mathbf{E}^{n,\mathbf{J}}\right)_{\partial\mathcal{T}_H}, \qquad \forall \boldsymbol{\nu} \in \boldsymbol{\Lambda}_H, \tag{7.15}$$

7.2.2 Introduction of the multiscale basis functions

Let $\boldsymbol{\psi}_i$, with $i \in \{1, ..., \dim \boldsymbol{\Lambda}_H\}$, be a basis for $\boldsymbol{\Lambda}_H$. Let $\beta_i^n \in \mathbb{R}$, $n \in \{1, ..., N\}$ are the degrees of freedom associated to $\boldsymbol{\lambda}_H^n$ in $\boldsymbol{\Lambda}_H$, *i.e.* :

$$\lambda_H^n = \sum_{i=1}^{\dim \Lambda_H} \beta_i^n \psi_i \tag{7.16}$$

Now we replace $\boldsymbol{\lambda}_H^n$ with $\boldsymbol{\psi}_i$, with $i \in \{1, ..., \dim \boldsymbol{\Lambda}_H\}$ in (7.13) and we define the set $(\boldsymbol{\eta}_i^{n,\mathbf{E}}, \boldsymbol{\eta}_i^{n,\mathbf{H}})$, $n \in \{1, ..., N\}$, for all $K \times I_n \in \mathcal{T}_H \times \mathcal{T}_{\Delta t}$, as the solution of the following problem:

$$\begin{cases} \left(\varepsilon \partial_t \boldsymbol{\eta}_i^{n,\mathbf{E}}, \mathbf{v}\right)_K - \left(\boldsymbol{\eta}_i^{n,\mathbf{H}}, \nabla \times \mathbf{v}\right)_K & = -\left(\boldsymbol{\psi}_i \mathbf{n} \cdot \mathbf{n}^K, \mathbf{v}\right)_{\partial K}, \quad \forall \mathbf{v} \in \mathbf{V}(K) \\ \left(\mu \partial_t \boldsymbol{\eta}_i^{n,\mathbf{H}}, \mathbf{w}\right)_K + \left(\nabla \times \boldsymbol{\eta}_i^{n,\mathbf{E}}, \mathbf{w}\right)_K & = 0, \quad \forall \mathbf{w} \in \mathbf{V}(K) \\ \boldsymbol{\eta}_i^{n,\mathbf{E}}(t_{n-1}) & = 0 \\ \boldsymbol{\eta}_i^{n,\mathbf{H}}(t_{n-1}) & = 0 \end{cases} \tag{7.17}$$

with $\mathbf{V}(K)$ the space of functions in V restricted to K. These functions are called the multiscale basis function. They serve as a basis to reconstruct $\mathbf{E}^{n,\boldsymbol{\lambda}_H}$ and $\mathbf{H}^{n,\boldsymbol{\lambda}_H}$.

Multiplying each equation by β_i^n, we obtain:

$$\begin{cases} \left(\varepsilon \beta_i^n \partial_t \boldsymbol{\eta}_i^{n,\mathbf{E}}, \mathbf{v}\right)_K - \left(\beta_i^n \boldsymbol{\eta}_i^{n,\mathbf{H}}, \nabla \times \mathbf{v}\right)_K & = -\left(\beta_i^n \boldsymbol{\psi}_i \mathbf{n} \cdot \mathbf{n}^K, \mathbf{v}\right)_{\partial K}, \quad \forall \mathbf{v} \in \mathbf{V}(K) \\ \left(\mu \beta_i^n \partial_t \boldsymbol{\eta}_i^{n,\mathbf{H}}, \mathbf{w}\right)_K + \left(\nabla \times \beta_i^n \boldsymbol{\eta}_i^{n,\mathbf{E}}, \mathbf{w}\right)_K & = 0, \quad \forall \mathbf{w} \in \mathbf{V}(K) \\ \boldsymbol{\eta}_i^{n,\mathbf{E}}(t_{n-1}) & = 0 \\ \boldsymbol{\eta}_i^{n,\mathbf{H}}(t_{n-1}) & = 0 \end{cases} \tag{7.18}$$

and we sum for each i in $\{1, ..., \dim \boldsymbol{\Lambda}_H\}$:

$$\begin{cases} \left(\varepsilon \sum_{i=1}^{\dim\Lambda_H} \beta_i^n \partial_t \boldsymbol{\eta}_i^{n,\mathbf{E}}, \mathbf{v}\right)_K - \left(\sum_{i=1}^{\dim\Lambda_H} \beta_i^n \boldsymbol{\eta}_i^{n,\mathbf{H}}, \nabla \times \mathbf{v}\right)_K & = -\left(\sum_{i=1}^{\dim\Lambda_H} \beta_i^n \boldsymbol{\psi}_i \mathbf{n} \cdot \mathbf{n}^K, \mathbf{v}\right)_{\partial K}, \quad \forall \mathbf{v} \in \mathbf{V}(K) \\ \left(\mu \sum_{i=1}^{\dim\Lambda_H} \beta_i^n \partial_t \boldsymbol{\eta}_i^{n,\mathbf{H}}, \mathbf{w}\right)_K + \left(\nabla \times \sum_{i=1}^{\dim\Lambda_H} \beta_i^n \boldsymbol{\eta}_i^{n,\mathbf{E}}, \mathbf{w}\right)_K & = 0, \qquad \forall \mathbf{w} \in \mathbf{V}(K) \\ \boldsymbol{\eta}_i^{n,\mathbf{E}}(t_{n-1}) & = 0 \\ \boldsymbol{\eta}_i^{n,\mathbf{H}}(t_{n-1}) & = 0 \end{cases} \tag{7.19}$$

From here and using equations (7.16) and (7.13) we can identify $\forall t \in I_n$:

$$\mathbf{E}^{n,\lambda_H} = \sum_{i=1}^{\dim\Lambda_H} \beta_i^n \boldsymbol{\eta}_i^{n,\mathbf{E}} \quad \text{and} \quad \mathbf{H}^{n,\lambda_H} = \sum_{i=1}^{\dim\Lambda_H} \beta_i^n \boldsymbol{\eta}_i^{n,\mathbf{H}}, \tag{7.20}$$

with the initial conditions $\boldsymbol{\eta}_i^{n,\mathbf{E}}(t_{n-1}) = 0$ and $\boldsymbol{\eta}_i^{n,\mathbf{H}}(t_{n-1}) = 0$.

Using equation (7.20) in (7.12), we obtain $\forall t \in I_n$:

$$\mathbf{E}_H^n = \mathbf{E}^{n,\mathbf{J}} + \sum_{i=1}^{\dim\boldsymbol{\Lambda}_H} \beta_i^n \boldsymbol{\eta}_i^{n,\mathbf{E}} \qquad \text{and} \qquad \mathbf{H}_H^n = \mathbf{H}^{n,\mathbf{J}} + \sum_{i=1}^{\dim\boldsymbol{\Lambda}_H} \beta_i^n \boldsymbol{\eta}_i^{n,\mathbf{H}} \tag{7.21}$$

From there we note the set of $(\boldsymbol{\eta}_i^{n,\mathbf{E}}, \boldsymbol{\eta}_i^{n,\mathbf{H}})$ with $i \in \{1, ..., \dim \boldsymbol{\Lambda}_H\}$, the multiscale basis functions. They are upscaling local information from the problems (7.17) to the coarse mesh. One can see an example in figure 7.3. Let us point out that the set of β_i^n is a fundamental part of the resulting algorithm. Indeed, (β_i^n) are the only globally coupled d.o.f. in the MHM method and are solution of a linear system defined on the coarse mesh skeleton $\partial\mathcal{T}_H$. This linear system results from (7.10) and decomposition (7.20) as for each $n \in \{1, ..., N\}$ find β_i^n such that $\forall t \in I_n$:

$$\left(\boldsymbol{\nu}_H, \sum_{i=1}^{\dim\Lambda_H} \beta_i^n \boldsymbol{\eta}_i^{n,\mathbf{E}}\right)_{\partial\mathcal{T}_H} = -\left(\boldsymbol{\nu}_H, \mathbf{E}^{n,\mathbf{J}}\right)_{\partial\mathcal{T}_H}, \qquad \forall \boldsymbol{\nu}_H \in \boldsymbol{\Lambda}_H, \tag{7.22}$$

given $\boldsymbol{\eta}_i^{n,\mathbf{E}}$ and $\mathbf{E}^{n,\mathbf{J}}$.

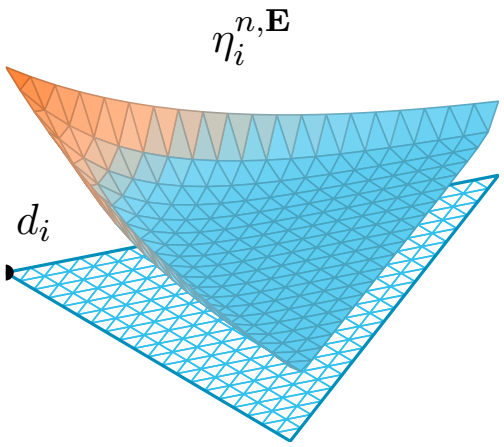


Figure 7.3 | Multiscale basis function. Example of $\eta_i^{n,\mathbf{E}}$ basis function on a given macroelement K.

If one assumes that close analytical formula are available for all $n \in \{1, ..., N\}$ for $(\boldsymbol{\eta}^{n,\mathbf{E}}, \boldsymbol{\eta}^{n,\mathbf{H}})$ and $(\mathbf{E}^{n,\mathbf{J}}, \mathbf{H}^{n,\mathbf{J}})$, it is still necessary to obtain the d.o.f. β_i^n to compute the solution from system (7.22). However, for real application test case, we do not have access to analytical formulas, meaning a second level of discretization is necessary to make the method practical. This is the subject of the next section.

7.3 Two-level MHM method

As stated at the end of the first section, analytical formulas for both $\boldsymbol{\eta}^{n,\boldsymbol{\xi}}$ and $\boldsymbol{\xi}^{n,\mathbf{J}}$, $\boldsymbol{\xi} \in \{\mathbf{E}, \mathbf{H}\}$ are not available for realistic cases. In order to solve the second level problems defined in (7.7) and (7.13), we choose to use a spatially centered DG scheme coupled with a second order Leap-Frog (LF) time scheme.

7.3.1 DG on each macro element K

The second level of space discretization will be called local mesh or fine mesh, unlike the global mesh or coarse mesh $\mathcal{T}_H$. More precisely, in each macro element K of $\mathcal{T}_H$ we build a fine partition noted $\mathcal{T}_h^K$, formed by elements κ of characteristic length $h^K > 0$. We precise that when h is used in the indices of $\mathcal{T}_h^K$, one should see $\mathcal{T}_{h^K}^K$. The additional exponent is omitted here for the sake of readability. One can find an illustration of this definition in figure 7.4. We also define the set of faces of the local mesh $\mathcal{T}_h^K$ by $\mathcal{F}_h^K$ as well as the set of faces on the boundary of K as $\mathcal{F}_0^K$.

Remark *: It is important to notice that no continuity at the junction between macro elements is mandatory, as illustrated in figure 7.4.*

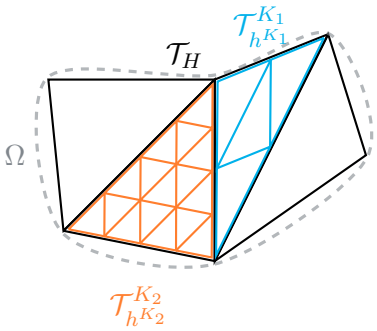


Figure 7.4 | Space discretization. Second level space discretization. Each macroelement K are discretized in submeshes $\mathcal{T}_{h^K}^K$.

We discretize local problems equations (7.7) and (7.17) with respect to the space variable by selecting a local finite dimensional space $\mathbf{V}_h(K)$ whose functions are defined over $\{\mathcal{T}_h^K\}_{h>0}$. The two discrete problems read as follows:

1. Find for $n \in \{1, ..., N\}$, $(\mathbf{E}_h^{n,\mathbf{J}}, \mathbf{H}_h^{n,\mathbf{J}})$ in $\mathbf{V}_h(K) \times \mathbf{V}_h(K)$ such that $\forall t \in I_n$:

$$\begin{cases} \left(\varepsilon\partial_t \mathbf{E}_h^{n,\mathbf{J}}, \mathbf{v}_h\right)_K - \left(\mathbf{H}_h^{n,\mathbf{J}}, \nabla\times\mathbf{v}_h\right)_K & = (\mathbf{J}, \mathbf{v}_h)_K, \qquad \forall \mathbf{v}_h \in \mathbf{V}_h(K) \\ \left(\mu\partial_t \mathbf{H}_h^{n,\mathbf{J}}, \mathbf{w}_h\right)_K + \left(\nabla\times\mathbf{E}_h^{n,\mathbf{J}}, \mathbf{w}_h\right)_K & = 0, \qquad \forall \mathbf{w}_h \in \mathbf{V}_h(K) \\ \mathbf{E}_h^{n,\mathbf{J}}(t_{n-1}) & = \mathbf{E}_h^{n-1}(t_{n-1}) \\ \mathbf{H}_h^{n,\mathbf{J}}(t_{n-1}) & = \mathbf{H}_h^{n-1}(t_{n-1}), \end{cases} \tag{7.23}$$

with $\mathbf{E}_h^{0,\mathbf{J}} = \mathbf{E}^0$ and $\mathbf{H}_h^{0,\mathbf{J}} = \mathbf{H}^0$.

2. Find for $n \in \{1, ..., N\}$, $(\boldsymbol{\eta}_{i,h}^{n,\mathbf{E}}, \boldsymbol{\eta}_{i,h}^{n,\mathbf{H}})$ in $\mathbf{V}_h(K) \times \mathbf{V}_h(K)$, such that for all $i \in \{1, ..., \dim\boldsymbol{\Lambda}_H\}$ such that $\forall t \in I_n$:

$$\begin{cases} \left(\varepsilon\partial_t \boldsymbol{\eta}_{i,h}^{n,\mathbf{E}}, \mathbf{v}_h\right)_K - \left(\boldsymbol{\eta}_{i,h}^{n,\mathbf{H}}, \nabla\times\mathbf{v}_h\right)_K & = -\left(\boldsymbol{\psi}_i \mathbf{n}\cdot\mathbf{n}^K, \mathbf{v}_h\right)_{\partial K}, \quad \forall \mathbf{v}_h \in \mathbf{V}_h(K) \\ \left(\mu\partial_t \boldsymbol{\eta}_{i,h}^{n,\mathbf{H}}, \mathbf{w}_h\right)_K + \left(\nabla\times\boldsymbol{\eta}_{i,h}^{n,\mathbf{E}}, \mathbf{w}_h\right)_K & = 0, \qquad \forall \mathbf{w}_h \in \mathbf{V}_h(K) \\ \boldsymbol{\eta}_{i,h}^{n,\mathbf{E}}(t_{n-1}) & = 0 \\ \boldsymbol{\eta}_{i,h}^{n,\mathbf{H}}(t_{n-1}) & = 0. \end{cases} \tag{7.24}$$

3. With the global problem which links the two aforementioned ones: Find for $n \in \{1, ..., N\}$, β_i^n, such that for all $i \in \{1, ..., \dim\boldsymbol{\Lambda}_H\}$ such that $\forall t \in I_n$:

$$\left(\boldsymbol{\nu}_h, \sum_{i=1}^{\dim\Lambda_H} \beta_i^n \boldsymbol{\eta}_{i,h}^{n,\mathbf{E}}\right)_{\partial\mathcal{T}_H} = -\left(\boldsymbol{\nu}_h, \mathbf{E}_h^{n,\mathbf{J}}\right)_{\partial\mathcal{T}_H}, \qquad \forall \boldsymbol{\nu}_h \in \boldsymbol{\Lambda}_H, \tag{7.25}$$

From here we can reconstruct our fields:

$$\mathbf{E}_{H,h}^n = \mathbf{E}_h^{n,\mathbf{J}} + \sum_{i=1}^{\dim\boldsymbol{\Lambda}_H} \beta_i^n \boldsymbol{\eta}_{i,h}^{n,\mathbf{E}} \qquad \text{and} \qquad \mathbf{H}_{H,h}^n = \mathbf{H}_h^{n,\mathbf{J}} + \sum_{i=1}^{\dim\boldsymbol{\Lambda}_H} \beta_i^n \boldsymbol{\eta}_{i,h}^{n,\mathbf{H}} \tag{7.26}$$

7.3.2 LF2 time integration on each time-slab I_n

In this part, we devise the fully discrete two-level MHM method by setting up a time-marching scheme for the local problems equations (7.23) and (7.24). A second-order Leap-Frog scheme with a multiscale time-stepping (MTS) strategy is proposed.
The second level time discretization comes from a decomposition of each I^n in a uniform partition of time-step $\Delta\tau^{n,K} > 0$ on each $K \in \mathcal{T}_H$, i.e. $t^n = t^{n,0} < t^{n,1} < .... < t^{n,M^{n,K}} = t^{n+1}$, with $M^{n,K} \in \mathbb{N}$, where $t^{n,k} = t^{n,0} + k\Delta\tau^{n,K}$, $k \in \{1, ..., M^{n,K}\}$. In the same way we call micro time-slab or fine time-slab the interval $I^{n,m} = [t^{n,m}, t^{n,m+1}]$ and $\Delta\tau^{n,K}$ the micro time-step associated to $I^{n,m}$.

Remark *: Note that this partition can be different in each element $K \in \mathcal{T}_h$. Strictly speaking, we should have one more exponent related to the macro element since the fine discretizations are independent from one K to another.*

Next, we focus on the intrinsic MTS of the MHM method within a simple framework by assuming that $\mathcal{T}_{\Delta t}$ is uniform with characteristic length $\Delta t^n = \Delta t$, $n \in \{1, ..., N\}$. We choose to divide each $I^n \in \mathcal{T}_{\Delta t}$ of each K into M^K intervals $t^n = t^{n,0} < t^{n,1} < \ldots < t^{n,M^K-1} < t^{n,M^K} = t^{n+1}$, so

that we can omit the time exponent from $\Delta\tau^{n,K}$ *i.e.* $\Delta\tau^{n,K} = \Delta\tau^K$, $n \in \{1, ..., N\}$. On K, we denote by $\mathbf{E}_{H,h}^{n,m}$ (resp. $\mathbf{H}_{H,h}^{n,m}$) the field that approximate $\mathbf{E}_{H,h}^{n}$ (resp. $\mathbf{H}_{H,h}^{n}$) at the discrete local time $t^{n,m}$, $m \in \{0, ..., M^K\}$. Since $t^{n,M^K} = t^{n,0} + M^K\Delta\tau^K = t^{n,0} + \Delta t^n = t^{n+1,0}$, we keep the continuity of $\mathbf{E}_{H,h}^{n,M} = \mathbf{E}_{H,h}^{n+1,0}$ and $\mathbf{H}_{H,h}^{n,M} = \mathbf{H}_{H,h}^{n+1,0}$ between each macro time-slab as illustrated in figure 7.5.

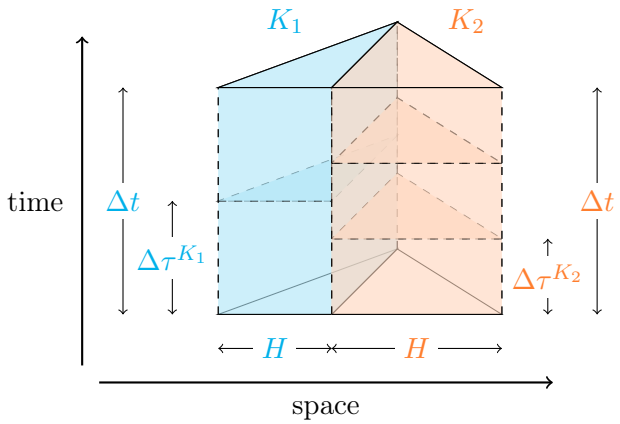


Figure 7.5 | Time discretization. Two level time discretization. The timeslab Δt is split in micro time-steps $\Delta\tau$

Owing to these definitions, the centered DG method applied to equation (7.24) reads as follows. Let $K \in \mathcal{T}_H$, find for each $m \in \{1, \dots, M^K\}$, $(\mathbf{E}_h^{n,m,\lambda_H}, \mathbf{H}_h^{n,m,\lambda_H}) \in \mathcal{C}^1(I_n, \mathbf{V}_h(K)) \times \mathcal{C}^1(I^n, \mathbf{V}_h(K))$, for all $I^n \in \mathcal{T}_{\Delta t}$, such that

$$\begin{cases} \dfrac{1}{\Delta\tau^K}\big(\varepsilon\boldsymbol{\eta}_{i,h}^{m+\frac{1}{2},\mathbf{E}}, \boldsymbol{\phi}_k^K\big)_K = \dfrac{1}{\Delta\tau^K}\big(\varepsilon\boldsymbol{\eta}_{i,h}^{m-\frac{1}{2},\mathbf{E}}, \boldsymbol{\phi}_k^K\big)_K + \big(\boldsymbol{\eta}_{i,h}^{m,\mathbf{H}}, \nabla\times\boldsymbol{\phi}_k^K\big)_K \\ \qquad\qquad - \big(\left\{\boldsymbol{\eta}_{i,h}^{m,\mathbf{H}}\right\}, [\![\boldsymbol{\phi}_k^K]\!]\big)_{\mathcal{F}_0^K} - (\boldsymbol{\psi}_i, \boldsymbol{\phi}_k^K)_{\partial K}\,, \\ \dfrac{1}{\Delta\tau^K}\big(\mu\boldsymbol{\eta}_{i,h}^{m+1,\mathbf{H}}, \boldsymbol{\phi}_k^K\big)_K = \dfrac{1}{\Delta\tau^K}\big(\mu\boldsymbol{\eta}_{i,h}^{m,\mathbf{H}}, \boldsymbol{\phi}_k^K\big)_K - \big(\boldsymbol{\eta}_{i,h}^{m+\frac{1}{2},\mathbf{E}}, \nabla\times\boldsymbol{\phi}_k^K\big)_K \\ \qquad\qquad + \big(\left\{\boldsymbol{\eta}_{i,h}^{m+\frac{1}{2},\mathbf{E}}\right\}, [\![\boldsymbol{\phi}_k^K]\!]\big)_{\mathcal{F}_h^K}\,, \end{cases} \tag{7.27}$$

with $\boldsymbol{\eta}_{i,h}^{\frac{1}{2},\mathbf{E}} = 0$ and $\boldsymbol{\eta}_{i,h}^{0,\mathbf{H}} = 0$, for all $(\mathbf{v}_h, \mathbf{w}_h) \in \mathbf{V}_h(K) \times \mathbf{V}_h(K)$. One can retrieve the definitions of the DG operators $\{\cdot\}$ and $[\![\cdot]\!]$ in chapter 2.

When applied to (7.7), it becomes: For each $K \in \mathcal{T}_H$, $m \in \{0, \dots, M^K\}$ find $(\mathbf{E}_h^{n,m,\mathbf{J}}, \mathbf{H}_h^{n,m,\mathbf{J}}) \in \mathbf{V}_h(K) \times \mathbf{V}_h(K)$, for all $I^n \in \mathcal{T}_{\Delta t}$, such that

$$\begin{cases} \Big(\varepsilon\,\dfrac{\mathbf{E}_h^{n,m+\frac{1}{2},\mathbf{J}} - \mathbf{E}_h^{n,m-\frac{1}{2},\mathbf{J}}}{\Delta\tau^K}, \mathbf{v}_h\Big)_{\mathcal{T}_h^K} - \big(\mathbf{H}_h^{n,m,\mathbf{J}}, \nabla\times\mathbf{v}_h\big)_{\mathcal{T}_h^K} + \big(\left\{\mathbf{H}_h^{n,m,\mathbf{J}}\right\}, [\![\mathbf{v}_h]\!]\big)_{\mathcal{F}_0^K} = (\mathbf{J}^{n,m}, \mathbf{v}_h)_{\mathcal{T}_h^K}\,, \\ \Big(\mu\,\dfrac{\mathbf{H}_h^{n,m+1,\mathbf{J}} - \mathbf{H}_h^{n,m,\mathbf{J}}}{\Delta\tau^K}, \mathbf{w}_h\Big)_{\mathcal{T}_h^K} + \big(\mathbf{E}_h^{n,m+\frac{1}{2},\mathbf{J}}, \nabla\times\mathbf{w}_h\big)_{\mathcal{T}_h^K} - \big(\left\{\mathbf{E}_h^{n,m+\frac{1}{2},\mathbf{J}}\right\}, [\![\mathbf{w}_h]\!]\big)_{\mathcal{F}_h^K} = 0\,, \end{cases} \tag{7.28}$$

given $\mathbf{E}_h^{n,\frac{1}{2},\mathbf{J}} = \mathbf{E}_{H,h}^{n-1}$ and $\mathbf{H}_h^{n,0,\mathbf{J}} = \mathbf{H}_{H,h}^{n-1}$ with $\mathbf{E}_h^{0,\frac{1}{2},\mathbf{J}} = \mathbf{E}^0$ and $\mathbf{H}_h^{0,0,\mathbf{J}} = \mathbf{H}^0$, for all $(\mathbf{v}_h, \mathbf{w}_h) \in \mathbf{V}_h(K) \times \mathbf{V}_h(K)$.

Remark *: As one can see, there is no more exponent n in the multiscale basis problem* (7.27)*. This invariance with respect to the macro time-slabs comes from two points: (i) the right-hand side of the problem* (7.27)*, which only depends on the basis function* $\boldsymbol{\psi}$*; (ii) the macro time-slab initial conditions are* $(0,0)$*. From there, one can solve the problem* (7.27) *once for all* I_n *i.e.* $I_K^{n,m} = [t_K^{n,m}, t_K^{n,m+1}]$*. We choose to not write explicitly this dependence if there is no ambiguity.*

With regard to the global problem (7.25), a choice must me made for its resolution. In [LPSV18], where no MTS strategy is envisaged, the global problem is solved once per time-slab I_n, $n \in \{1, ..., N\}$, more precisely at t^{n,M^K}. In our case, we envisage the evolution through one macrotime slab. In order to take into account history inside each I_n, we propose the following equivalent formulation: For all $I_n \in \mathcal{T}_{\Delta t}$, find $\lambda_H^n \in \Lambda_H$ such that

$$\left(\boldsymbol{\nu}_H \,, \int_{I_n} \sum_{i=1}^{\dim\Lambda_H} \beta_i^n \boldsymbol{\eta}_{i,h}^{n,\mathbf{E}}\right)_{\partial\mathcal{T}_H} = -\left(\boldsymbol{\nu}_H \,, \int_{I_n} \mathbf{E}_h^{n,\mathbf{J}}\right)_{\partial\mathcal{T}_H}, \qquad \forall \boldsymbol{\nu}_H \in \boldsymbol{\Lambda}_H, \tag{7.29}$$

Remark *: For the moment, equation* (7.29) *is not discretized in time. Different integration quadrature are possible. We show in the following that the choice of the integration quadrature affects the stability of the scheme.*

7.3.3 Energy preservation

The global problem is approximated for all $n \in \{1, ..., N\}$ by:

$$\Delta\tau^K \big(\sum_{m=1}^{M} \frac{\mathbf{E}_{H,h}^{n,m+\frac{1}{2}} + \mathbf{E}_{H,h}^{n,m-\frac{1}{2}}}{2}, \boldsymbol{\mu}_H\big)_{\partial\mathcal{T}_H} = 0 \tag{7.30}$$

which corresponds to a trapezoidal quadrature rule on the integral $\displaystyle\int_{I^n} \big(\mathbf{E}_{H,h}, \boldsymbol{\mu}_H\big)_{\partial\mathcal{T}_H} = 0$. This choice comes directly from the theorem below.

Theorem 7.3.1. *Let us define the second local level energy,* $\forall n \in \{1, ..., N\}$, $\forall m \in \{1, ..., M^K\}$

$$\mathcal{E}_{H,h}^{n,m+1/2}(K) = \frac{\varepsilon}{2} \| \boldsymbol{E}_{H,h}^{n,m+\frac{1}{2}} \|_{0,\mathcal{T}_h^K}^2 + \frac{1}{2} \big(\mu \boldsymbol{H}_{H,h}^{n,m}, \boldsymbol{H}_{H,h}^{n,m+1}\big)_{\mathcal{T}_h^K}. \tag{7.31}$$

The first level energy given on I^n *at discrete time* t^n *by*

$$\mathcal{E}_{H,h}^{n} = \sum_{K\in\mathcal{T}_H} \mathcal{E}_{H,h}^{n,M^K+1/2}(K) \tag{7.32}$$

One has the following energy , $\forall n \in \{1, ..., N\}$*:*

$$\mathcal{E}_{H,h}^{n} \;=\; \sum_{K\in\mathcal{T}_H} \mathcal{E}_{H,h}^{0,1/2}(K) + \sum_{K\in\mathcal{T}_H} \sum_{k=0}^{n} \sum_{m=1}^{M^K} (\boldsymbol{J}^{k,m}, \frac{\boldsymbol{E}^{k,m+\frac{1}{2}} + \boldsymbol{E}^{k,m-\frac{1}{2}}}{2})_{\mathcal{T}_H},$$

So that, if one has a vanishing current $\boldsymbol{J}$*, the energy is preserved over discrete times.*

Let $n \in \{0,..,N\}$. We write the local problems on each $I^n \times K$ and use the equivalent discrete equations resulting from (7.27), (7.28) and (7.29). Given $(\mathbf{E}_{H,h}^{n,\frac{1}{2},\lambda_H}, \mathbf{H}_{H,h}^{n,0,\lambda_H})$, find for each $m \in \{1,\dots,M\}$, $(\mathbf{E}_{H,h}^{n,m+1/2}, \mathbf{H}_{H,h}^{n,m}) \in \mathcal{C}^1(I^n, \mathbf{V}_h(K)) \times \mathcal{C}^1(I^n, \mathbf{V}_h(K))$ such that

$$\left\{\begin{array}{rcl} \big(\varepsilon \dfrac{\mathbf{E}_{H,h}^{n,m+\frac{1}{2}} - \mathbf{E}_{H,h}^{n,m-\frac{1}{2}}}{\Delta\tau^K}, \mathbf{v}_h\big)_{\mathcal{T}_h^K} - \big(\mathbf{H}_{H,h}^{n,m}, \nabla\times\mathbf{v}_h\big)_{\mathcal{T}_h^K} + (\Big\{\mathbf{H}_{H,h}^{n,m}\Big\}, [\![\mathbf{v}_h]\!])_{\mathcal{F}_0^K} & = & -(\boldsymbol{\lambda}_H^n, \mathbf{v}_h)_{\partial K}, \\ & & +(\mathbf{J}^{n,m}, \mathbf{v}_h)_{\mathcal{T}_h^K}, \\ \big(\mu \dfrac{\mathbf{H}_{H,h}^{n,m+1} - \mathbf{H}_{H,h}^{n,m}}{\Delta\tau^K}, \mathbf{w}_h\big)_{\mathcal{T}_h^K} + \big(\mathbf{E}_{H,h}^{n,m+\frac{1}{2}}, \nabla\times\mathbf{w}_h\big)_{\mathcal{T}_h^K} - (\Big\{\mathbf{E}_{H,h}^{n,m+\frac{1}{2}}\Big\}, [\![\mathbf{w}_h]\!])_{\mathcal{F}_h^K} & = & 0\,, \end{array}\right. \tag{7.33}$$

with $\mathbf{E}_{H,h}^{n,\frac{1}{2}} = \mathbf{E}_{H,h}^{n-1,M^K+\frac{1}{2}}$, $\mathbf{H}_{H,h}^{n,0} = \mathbf{H}_{H,h}^{n-1,M^K}$ and $\mathbf{J}^{n,m}$ the external current computed at time $t^{n,m}$.

Remark *: For the second equation, on faces of $\mathcal{F}_h^K \setminus \mathcal{F}_0^K$ i.e. the internal faces, we use*

$$\Big\{\boldsymbol{E}_{H,h}^{n,m+\frac{1}{2}}\Big\} = \boldsymbol{E}_{H,h}^{n,m+\frac{1}{2}}$$

and

$$[\![\boldsymbol{w}_h]\!] = \boldsymbol{w}_h.$$

Proof. One has for $n \in \{1,...,N\}$ and $m \in \{1,...,M^K\}$

$$\begin{array}{rcl} \mathcal{E}_{H,h}^{n,m+1/2}(K) - \mathcal{E}_{H,h}^{n,m-1/2}(K) & = & \dfrac{\varepsilon}{2}\| \mathbf{E}_{H,h}^{n,m+\frac{1}{2}}\|_{0,\mathcal{T}_h^K}^2 + \dfrac{1}{2}\big(\mu\mathbf{H}_{H,h}^{n,m}, \mathbf{H}_{H,h}^{n,m+1}\big)_{\mathcal{T}_h^K} \\ & & -\dfrac{\varepsilon}{2}\| \mathbf{E}_{H,h}^{n,m-\frac{1}{2}}\|_{0,\mathcal{T}_h^K}^2 - \dfrac{1}{2}\big(\mu\mathbf{H}_{H,h}^{n,m-1}, \mathbf{H}_{H,h}^{n,m}\big)_{\mathcal{T}_h^K} \\ & = & \dfrac{1}{2}\big(\varepsilon\,(\mathbf{E}_{H,h}^{n,m+\frac{1}{2}} - \mathbf{E}_{H,h}^{n,m-\frac{1}{2}}), \mathbf{E}_{H,h}^{n,m+\frac{1}{2}} + \mathbf{E}_{H,h}^{n,m-\frac{1}{2}}\big)_{\mathcal{T}_h^K} \\ & & +\dfrac{1}{2}\big(\mu(\mathbf{H}_{H,h}^{n,m+1} - \mathbf{H}_{H,h}^{n,m-1}), \mathbf{H}_{H,h}^{n,m}\big)_{\mathcal{T}_h^K}. \end{array}$$

Adding the equations (7.28) to its $\boldsymbol{\lambda}$ counterpart gives us this complete form of system (7.28):

$$\left\{\begin{array}{rcl} \big(\varepsilon \dfrac{\mathbf{E}_{H,h}^{n,m+\frac{1}{2}} - \mathbf{E}_{H,h}^{n,m-\frac{1}{2}}}{\Delta\tau^K}, \mathbf{v}_h\big)_{\mathcal{T}_h^K} & = & \big(\mathbf{H}_{H,h}^{n,m}, \nabla\times\mathbf{v}_h\big)_{\mathcal{T}_h^K} - (\Big\{\mathbf{H}_{H,h}^{n,m}\Big\}, [\![\mathbf{v}_h]\!])_{\mathcal{F}_0^K} \\ & & -(\boldsymbol{\lambda}_H^n, \mathbf{v}_h)_{\partial K} + (\mathbf{J}^{n,m}, \mathbf{v}_h)_{\mathcal{T}_h^K}, \\ \big(\mu \dfrac{\mathbf{H}_{H,h}^{n,m+1} - \mathbf{H}_{H,h}^{n,m}}{\Delta\tau^K}, \mathbf{w}_h\big)_{\mathcal{T}_h^K} & = & -\big(\mathbf{E}_{H,h}^{n,m+\frac{1}{2}}, \nabla\times\mathbf{w}_h\big)_{\mathcal{T}_h^K} + (\Big\{\mathbf{E}_{H,h}^{n,m+\frac{1}{2}}\Big\}, \Big[\!\Big[\mathbf{w}_{H,h}^{n,m}\Big]\!\Big])_{\mathcal{F}_h^K}, \end{array}\right. \tag{7.34}$$

Testing with respectively the fields $\frac{\mathbf{E}_{H,h}^{n,m+\frac{1}{2}}+\mathbf{E}_{H,h}^{n,m-\frac{1}{2}}}{2}$ and $\mathbf{H}_{H,h}^{n,m}$, and the second equation of (7.34) at discrete step $m-1$, with $\mathbf{H}_{H,h}^{n,m}$, one obtains the three equations:

$$\left\{\begin{array}{rcl}
\big(\varepsilon\,\dfrac{\mathbf{E}_{H,h}^{n,m+\frac{1}{2}}-\mathbf{E}_{H,h}^{n,m-\frac{1}{2}}}{\Delta\tau^K},\dfrac{\mathbf{E}^{n,m+\frac{1}{2}}+\mathbf{E}^{n,m-\frac{1}{2}}}{2}\big)_{\mathcal{T}_h^K} & = & \big(\mathbf{H}_{H,h}^{n,m},\nabla\times\dfrac{\mathbf{E}^{n,m+\frac{1}{2}}+\mathbf{E}^{n,m-\frac{1}{2}}}{2}\big)_{\mathcal{T}_h^K} \\
 & & -\big(\Big\{\mathbf{H}_{H,h}^{n,m}\Big\},\Big[\!\Big[\dfrac{\mathbf{E}^{n,m+\frac{1}{2}}+\mathbf{E}^{n,m-\frac{1}{2}}}{2}\Big]\!\Big]\big)_{\mathcal{F}_0^K} \\
 & & -(\boldsymbol{\lambda}_H^n,\dfrac{\mathbf{E}^{n,m+\frac{1}{2}}+\mathbf{E}^{n,m-\frac{1}{2}}}{2})_{\partial K} \\
 & & +(\mathbf{J}^{n,m},\dfrac{\mathbf{E}^{n,m+\frac{1}{2}}+\mathbf{E}^{n,m-\frac{1}{2}}}{2})_{\mathcal{T}_h^K}\,, \\
\big(\mu\,\dfrac{\mathbf{H}_{H,h}^{n,m+1}-\mathbf{H}_{H,h}^{n,m}}{\Delta\tau^K},\mathbf{H}_{H,h}^{n,m}\big)_{\mathcal{T}_h^K} & = & -\big(\mathbf{E}_{H,h}^{n,m+\frac{1}{2}},\nabla\times\mathbf{H}_{H,h}^{n,m}\big)_{\mathcal{T}_h^K} \\
 & & +\big(\Big\{\mathbf{E}_{H,h}^{n,m+\frac{1}{2}}\Big\},\Big[\!\Big[\mathbf{H}_{H,h}^{n,m}\Big]\!\Big]\big)_{\mathcal{F}_h^K}\,, \\
\big(\mu\,\dfrac{\mathbf{H}_{H,h}^{n,m}-\mathbf{H}_{H,h}^{n,m-1}}{\Delta\tau^K},\mathbf{H}_{H,h}^{n,m}\big)_{\mathcal{T}_h^K} & = & -\big(\mathbf{E}_{H,h}^{n,m-\frac{1}{2}},\nabla\times\mathbf{H}_{H,h}^{n,m}\big)_{\mathcal{T}_h^K} \\
 & & +\big(\Big\{\mathbf{E}_{H,h}^{n,m-\frac{1}{2}}\Big\},\Big[\!\Big[\mathbf{H}_{H,h}^{n,m}\Big]\!\Big]\big)_{\mathcal{F}_h^K}\,,
\end{array}\right. \tag{7.35}$$

This gives,

$$\left\{\begin{array}{rcl}
\big(\varepsilon\,\dfrac{\mathbf{E}_{H,h}^{n,m+\frac{1}{2}}-\mathbf{E}_{H,h}^{n,m-\frac{1}{2}}}{\Delta\tau^K},\dfrac{\mathbf{E}^{n,m+\frac{1}{2}}+\mathbf{E}^{n,m-\frac{1}{2}}}{2}\big)_{\mathcal{T}_h^K} & = & \big(\mathbf{H}_{H,h}^{n,m},\nabla\times\dfrac{\mathbf{E}^{n,m+\frac{1}{2}}+\mathbf{E}^{n,m-\frac{1}{2}}}{2}\big)_{\mathcal{T}_h^K} \\
 & & -\big(\Big\{\mathbf{H}_{H,h}^{n,m}\Big\},\Big[\!\Big[\dfrac{\mathbf{E}^{n,m+\frac{1}{2}}+\mathbf{E}^{n,m-\frac{1}{2}}}{2}\Big]\!\Big]\big)_{\mathcal{F}_0^K} \\
 & & -(\boldsymbol{\lambda}_H^n,\dfrac{\mathbf{E}^{n,m+\frac{1}{2}}+\mathbf{E}^{n,m-\frac{1}{2}}}{2})_{\partial K} \\
 & & +(\mathbf{J}^{n,m},\dfrac{\mathbf{E}^{n,m+\frac{1}{2}}+\mathbf{E}^{n,m-\frac{1}{2}}}{2})_{\mathcal{T}_h^K}\,, \\
\big(\mu\,\dfrac{\mathbf{H}_{H,h}^{n,m+1}-\mathbf{H}_{H,h}^{n,m-1}}{\Delta\tau^K},\mathbf{H}_{H,h}^{n,m}\big)_{\mathcal{T}_h^K} & = & -\big(\mathbf{E}_{H,h}^{n,m+\frac{1}{2}}+\mathbf{E}_{H,h}^{n,m-\frac{1}{2}},\nabla\times\mathbf{H}_{H,h}^{n,m}\big)_{\mathcal{T}_h^K} \\
 & & +\big(\Big\{\mathbf{E}_{H,h}^{n,m+\frac{1}{2}}+\mathbf{E}_{H,h}^{n,m-\frac{1}{2}}\Big\},\Big[\!\Big[\mathbf{H}_{H,h}^{n,m}\Big]\!\Big]\big)_{\mathcal{F}_h^K}\,,
\end{array}\right. \tag{7.36}$$

Thus, one can estimate the variation of the fully discrete energy over $I^{n,m}$ as:

$$
\begin{aligned}
\frac{\mathcal{E}_{H,h}^{n,m+1/2}(K)-\mathcal{E}_{H,h}^{n,m-1/2}(K)}{\Delta\tau^K} &= \big(\mathbf{H}_{H,h}^{n,m},\nabla\times\frac{\mathbf{E}^{n,m+\frac{1}{2}}+\mathbf{E}^{n,m-\frac{1}{2}}}{2}\big)_{\mathcal{T}_h^K}\\
&\quad-\big(\Big\{\mathbf{H}_{H,h}^{n,m}\Big\},\Big[\!\Big[\frac{\mathbf{E}^{n,m+\frac{1}{2}}+\mathbf{E}^{n,m-\frac{1}{2}}}{2}\Big]\!\Big]\big)_{\mathcal{F}_0^K}\\
&\quad-(\boldsymbol{\lambda}_H^n,\frac{\mathbf{E}^{n,m+\frac{1}{2}}+\mathbf{E}^{n,m-\frac{1}{2}}}{2})_{\partial K}\\
&\quad+(\mathbf{J}^{n,m},\frac{\mathbf{E}^{n,m+\frac{1}{2}}+\mathbf{E}^{n,m-\frac{1}{2}}}{2})_{\mathcal{T}_h^K},\\
&\quad-\big(\mathbf{E}_{H,h}^{n,m+\frac{1}{2}}+\mathbf{E}_{H,h}^{n,m-\frac{1}{2}},\nabla\times\mathbf{H}_{H,h}^{n,m}\big)_{\mathcal{T}_h^K}\\
&\quad+\big(\Big\{\mathbf{E}_{H,h}^{n,m+\frac{1}{2}}+\mathbf{E}_{H,h}^{n,m-\frac{1}{2}}\Big\},\Big[\!\Big[\mathbf{H}_{H,h}^{n,m}\Big]\!\Big]\big)_{\mathcal{F}_h^K}
\end{aligned}
\tag{7.37}
$$

Summing (7.37) for $m\in\{1,...,M\}$ make a lot of terms telescoping , which at the end gives us:

$$
\begin{aligned}
\mathcal{E}_{H,h}^{n}-\mathcal{E}_{H,h}^{n-1} &= \sum_{K\in\mathcal{T}_H}\sum_{m=1}^{M^K}\Delta\tau^K(\mathcal{E}_{H,h}^{n,m+1/2}(K)-\mathcal{E}_{H,h}^{n,m-1/2}(K))\\
&= \sum_{K\in\mathcal{T}_H}\sum_{m=1}^{M^K}-\Delta\tau^K(\boldsymbol{\lambda}_H^n,\frac{\mathbf{E}^{n,m+\frac{1}{2}}+\mathbf{E}^{n,m-\frac{1}{2}}}{2})_{\partial K}\\
&\quad+\sum_{K\in\mathcal{T}_H}\sum_{m=1}^{M^K}\Delta\tau^K(\mathbf{J}^{n,m},\frac{\mathbf{E}^{n,m+\frac{1}{2}}+\mathbf{E}^{n,m-\frac{1}{2}}}{2})_{\mathcal{T}_H^K},
\end{aligned}
\tag{7.38}
$$

Finally, if we write the global evolution of the discrete energy over $[0,t^n]$, we obtain,

$$
\begin{aligned}
\mathcal{E}_{H,h}^{n}-\mathcal{E}_{H,h}^{0} &= \sum_{k=0}^{n}\mathcal{E}_{H,h}^{k+1}-\mathcal{E}_{H,h}^{k}\\
&= \sum_{k=0}^{n}\sum_{K\in\mathcal{T}_H}\sum_{m=1}^{M^K}-\Delta\tau^K(\boldsymbol{\lambda}_H^n,\frac{\mathbf{E}^{k,m+\frac{1}{2}}+\mathbf{E}^{k,m-\frac{1}{2}}}{2})_{\partial K}\\
&\quad+\sum_{k=0}^{n}\sum_{K\in\mathcal{T}_H}\sum_{m=1}^{M^K}\Delta\tau^K(\mathbf{J}^{k,m},\frac{\mathbf{E}^{k,m+\frac{1}{2}}+\mathbf{E}^{k,m-\frac{1}{2}}}{2})_{\mathcal{T}_H^K},
\end{aligned}
\tag{7.39}
$$

where we have used the continuity of the field on the boundary of I^n.

We now use the global problem (7.30) on each I^n with test function $\boldsymbol{\mu}_H=\boldsymbol{\lambda}_H^n$. We find:

$$
\mathcal{E}_{H,h}^{n+1}-\mathcal{E}_{H,h}^{0} = \sum_{K\in\mathcal{T}_H}\sum_{k=0}^{n}\sum_{m=1}^{M^K}\Delta\tau^K(\mathbf{J}^{k,m},\frac{\mathbf{E}^{k,m+\frac{1}{2}}+\mathbf{E}^{k,m-\frac{1}{2}}}{2})_{\mathcal{T}_H^K},
$$

□

7.3.4 Stability

In order to prove that our scheme is stable, we study if $\mathcal{E}_{H,h}^{n,M+1/2}$ is a positive definite quadratic form of $\mathbf{E}^{n,m+\frac{1}{2}}$ and $\mathbf{H}^{n,m}$. We suppose that each $\mathcal{T}_h^K$ is a quasi uniform mesh, i.e. $\mathcal{T}_h^K$ is shape regular and $\exists \zeta_K > 0 \, \forall \tau_i \in \mathcal{T}_h^K$, $h_{\tau_i} \geq \zeta_K h_K$ with $h_K = \max_{\tau_i \in \mathcal{T}_h^K} h_{\tau_i}^K$. We obtain the following result.

Theorem 7.3.2. *For all $K \in \mathcal{T}_H$, the local energy $\mathcal{E}_{H,h}^{n,m+1/2}(K)$ is a positive form of $\boldsymbol{E}_{H,h}^{n,m+\frac{1}{2}}$ and $\boldsymbol{H}_{H,h}^{n,m}$ under the following CFL condition per macro element K*

$$\Delta\tau^K < \frac{h_K}{2C_K} \tag{7.40}$$

with C_K a given constant independent of $(\Delta\tau^K, h_K)$;

Proof. We start from the fully discrete local energy, and by using the second equation of (7.36) to replace $\mathbf{H}_{H,h}^{n,m+1}$, we get:

$$\begin{aligned}
\mathcal{E}_{H,h}^{n,m+1/2}(K) &= \frac{\varepsilon}{2} \| \mathbf{E}_{H,h}^{n,m+\frac{1}{2}} \|_{0,\mathcal{T}_h^K}^2 + \frac{1}{2} \big(\mu \mathbf{H}_{H,h}^{n,m}, \mathbf{H}_{H,h}^{n,m+1} \big)_{\mathcal{T}_h^K} \\
&= \frac{\varepsilon}{2} \| \mathbf{E}_{H,h}^{n,m+\frac{1}{2}} \|_{0,\mathcal{T}_h^K}^2 + \frac{\mu}{2} \| \mathbf{H}_{H,h}^{n,m} \|_{0,\mathcal{T}_h^K}^2 \\
&\quad - \Delta\tau^K \big(\mathbf{E}_{H,h}^{n,m+\frac{1}{2}}, \nabla \times \mathbf{H}_{H,h}^{n,m} \big)_{\mathcal{T}_h^K} \\
&\quad + \Delta\tau^K \Big(\Big\{ \mathbf{E}_{H,h}^{n,m+\frac{1}{2}} \Big\}, \Big[\Big[\mathbf{H}_{H,h}^{n,m} \Big] \Big] \Big)_{\mathcal{F}_h^K}
\end{aligned}$$

We recall here inverse inequalities [Cia78, EG06]. We can assume that there exists a constant $C > 0$ such that for all $p \in \mathbb{P}_l(\mathcal{T}_h^K)$,

$$\begin{aligned}
\| \nabla \times p \|_{0,\tau}^2 &\leq C h_\tau^{-1} \| p \|_{0,\tau}^2 \\
\| \nabla \times p \|_{0,\mathcal{T}_h^K}^2 &= \sum_{\tau \in \mathcal{T}_h^K} \| \nabla \times p \|_{0,\tau}^2 \\
&\leq \sum_{\tau \in \mathcal{T}_h^K} C h_\tau^{-1} \| p \|_{0,\tau}^2 \, ,
\end{aligned}$$

and

$$\begin{aligned}
\| p \|_{0,\partial\tau}^2 &\leq C h_\tau^{1} \| p \|_{0,\tau}^2 \\
\| p \|_{0,\mathcal{F}_h^K}^2 &= \sum_{\tau \in \mathcal{T}_h^K} \| p \|_{0,\partial\tau}^2 \\
&\leq \sum_{\tau \in \mathcal{T}_h^K} C h_\tau^{-1} \| p \|_{0,\tau}^2 \, ,
\end{aligned}$$

Thus,

$$\begin{aligned}
\Delta\tau^K\big(\mathbf{E}_{H,h}^{n,m+\frac{1}{2}},\nabla\times\mathbf{H}_{H,h}^{n,m}\big)_{\mathcal{T}_h^K} &\leq \Delta\tau^K C_1\sum_{\tau\in\mathcal{T}_h^K}\frac{1}{h_\tau}\|\mathbf{E}_{H,h}^{n,m+\frac{1}{2}}\|_{0,\tau}\|\mathbf{H}_{H,h}^{n,m}\|_{0,\tau}\\
|\Delta\tau^K\big(\Big\{\mathbf{E}_{H,h}^{n,m+\frac{1}{2}}\Big\},\Big[\!\Big[\mathbf{H}_{H,h}^{n,m}\Big]\!\Big]\big)_{\mathcal{F}_h^K}| &\leq \Delta\tau^K C_2C_3\sum_{\tau_i\in\mathcal{T}_h^K}\sum_{\tau_j\in\mathcal{V}_i}\frac{1}{2h_{\tau_i}^{\frac{1}{2}}h_{\tau_i}^{\frac{1}{2}}}\|\mathbf{E}_{H,h}^{n,m+\frac{1}{2}}\|_{0,\tau_i}\|\mathbf{H}_{H,h}^{n,m}\|_{0,\tau_i}\\
&\quad+\Delta\tau^K C_3C_4\sum_{\tau_i\in\mathcal{T}_h^K}\sum_{\tau_j\in\mathcal{V}_i}\frac{1}{2h_{\tau_i}^{\frac{1}{2}}h_{\tau_j}^{\frac{1}{2}}}\|\mathbf{E}_{H,h}^{n,m+\frac{1}{2}}\|_{0,\tau_j}\|\mathbf{H}_{H,h}^{n,m}\|_{0,\tau_i},
\end{aligned}$$

moreover, using quasi-uniformity, one as:

$$\begin{aligned}
\Delta\tau^K\big(\mathbf{E}_{H,h}^{n,m+\frac{1}{2}},\nabla\times\mathbf{H}_{H,h}^{n,m}\big)_{\mathcal{T}_h^K} &\leq \frac{\Delta\tau^K C_1}{\zeta_K h_K}\sum_{\tau\in\mathcal{T}_h^K}\|\mathbf{E}_{H,h}^{n,m+\frac{1}{2}}\|_{0,\tau}\|\mathbf{H}_{H,h}^{n,m}\|_{0,\tau}\\
|\Delta\tau^K\big(\Big\{\mathbf{E}_{H,h}^{n,m+\frac{1}{2}}\Big\},\Big[\!\Big[\mathbf{H}_{H,h}^{n,m}\Big]\!\Big]\big)_{\mathcal{T}_h^K}| &\leq \frac{\Delta\tau^K C_2C_3}{2\zeta_K h_K}\sum_{\tau_i\in\mathcal{T}_h^K}\sum_{\tau_j\in\mathcal{V}_i}\|\mathbf{E}_{H,h}^{n,m+\frac{1}{2}}\|_{0,\tau_i}\|\mathbf{H}_{H,h}^{n,m}\|_{0,\tau_i}\\
&\quad+\frac{\Delta\tau^K C_3C_4}{2\zeta_K h_K}\sum_{\tau_i\in\mathcal{T}_h^K}\sum_{\tau_j\in\mathcal{V}_i}\|\mathbf{E}_{H,h}^{n,m+\frac{1}{2}}\|_{0,\tau_j}\|\mathbf{H}_{H,h}^{n,m}\|_{0,\tau_i},
\end{aligned}$$

Then we can give a lower bound for $\mathcal{E}_{H,h}^{n,m+1/2}(K)$:

$$\begin{aligned}
\mathcal{E}_{H,h}^{n,m+1/2}(K) &\geq \sum_{\tau\in\mathcal{T}_h^K}\big(\frac{\varepsilon}{2}\|\mathbf{E}_{H,h}^{n,m+\frac{1}{2}}\|_{0,\tau}^2+\frac{\mu}{2}\|\mathbf{H}_{H,h}^{n,m}\|_{0,\tau}^2\big)\\
&\quad-\frac{\Delta\tau^K C_1}{\zeta_K h_K}\sum_{\tau\in\mathcal{T}_h^K}\|\mathbf{E}_{H,h}^{n,m+\frac{1}{2}}\|_{0,\tau}\|\mathbf{H}_{H,h}^{n,m}\|_{0,\tau}\\
&\quad-\frac{\Delta\tau^K C_2C_3}{2\zeta_K h_K}\sum_{\tau_i\in\mathcal{T}_h^K}\sum_{\tau_j\in\mathcal{V}_i}\|\mathbf{E}_{H,h}^{n,m+\frac{1}{2}}\|_{0,\tau_i}\|\mathbf{H}_{H,h}^{n,m}\|_{0,\tau_i}\\
&\quad-\frac{\Delta\tau^K C_3C_4}{2\zeta_K h_K}\sum_{\tau_i\in\mathcal{T}_h^K}\sum_{\tau_j\in\mathcal{V}_i}\|\mathbf{E}_{H,h}^{n,m+\frac{1}{2}}\|_{0,\tau_j}\|\mathbf{H}_{H,h}^{n,m}\|_{0,\tau_i},
\end{aligned}$$

Noticing that:

$$\begin{aligned}
\|\mathbf{E}_{H,h}^{n,m+\frac{1}{2}}\|_{0,\tau}\|\mathbf{H}_{H,h}^{n,m}\|_{0,\tau} &= C\sqrt{\varepsilon}\|\mathbf{E}_{H,h}^{n,m+\frac{1}{2}}\|_{0,\tau}\sqrt{\mu}\|\mathbf{H}_{H,h}^{n,m}\|_{0,\tau}\\
&\leq \frac{C}{2}\big(\varepsilon\|\mathbf{E}_{H,h}^{n,m+\frac{1}{2}}\|_{0,\tau}^2+\mu\|\mathbf{H}_{H,h}^{n,m}\|_{0,\tau}^2\big),
\end{aligned}$$

We obtain:

$$
\begin{aligned}
\mathcal{E}_{H,h}^{n,m+1/2}(K) \;\geq\; & \sum_{\tau\in\mathcal{T}_h^K}\big(\frac{\varepsilon}{2}\|\mathbf{E}_{H,h}^{n,m+\frac{1}{2}}\|_{0,\tau}^2+\frac{\mu}{2}\|\mathbf{H}_{H,h}^{n,m}\|_{0,\tau}^2\big)\\
& -\frac{\Delta\tau^K C_1C_5}{\zeta_K h_K}\sum_{\tau\in\mathcal{T}_h^K}\big(\frac{\varepsilon}{2}\|\mathbf{E}_{H,h}^{n,m+\frac{1}{2}}\|_{0,\tau}^2+\frac{\mu}{2}\|\mathbf{H}_{H,h}^{n,m}\|_{0,\tau}^2\big)\\
& -\frac{\Delta\tau^K C_2C_3C_5}{2\zeta_K h_K}\sum_{\tau_i\in\mathcal{T}_h^K}\sum_{\tau_j\in\mathcal{V}_i}\big(\frac{\varepsilon}{2}\|\mathbf{E}_{H,h}^{n,m+\frac{1}{2}}\|_{0,\tau_i}^2+\frac{\mu}{2}\|\mathbf{H}_{H,h}^{n,m}\|_{0,\tau_i}^2\big)\\
& -\frac{\Delta\tau^K C_3C_4C_5}{2\zeta_K h_K}\sum_{\tau_i\in\mathcal{T}_h^K}\sum_{\tau_j\in\mathcal{V}_i}\big(\frac{\varepsilon}{2}\|\mathbf{E}_{H,h}^{n,m+\frac{1}{2}}\|_{0,\tau_j}^2+\frac{\mu}{2}\|\mathbf{H}_{H,h}^{n,m}\|_{0,\tau_i}^2\big)\\
= \;& \big(\frac{1}{2}-\frac{\Delta\tau^K\tilde{C}_1}{2\zeta_K h_K}-\frac{\Delta\tau^K\tilde{C}_2}{4\zeta_K h_K}-\frac{\Delta\tau^K\tilde{C}_3}{4\zeta_K h_K}\big)\sum_{\tau\in\mathcal{T}_h^K}\big(\varepsilon\|\mathbf{E}_{H,h}^{n,m+\frac{1}{2}}\|_{0,\tau}^2+\mu\|\mathbf{H}_{H,h}^{n,m}\|_{0,\tau}^2\big)\\
= \;& \big(\frac{1}{2}-\frac{\Delta\tau^K}{h_K}(\frac{2\tilde{C}_1+\tilde{C}_2+\tilde{C}_3}{4\zeta_K})\big)\sum_{\tau\in\mathcal{T}_h^K}\big(\varepsilon\|\mathbf{E}_{H,h}^{n,m+\frac{1}{2}}\|_{0,\tau}^2+\mu\|\mathbf{H}_{H,h}^{n,m}\|_{0,\tau}^2\big)\\
= \;& \big(\frac{1}{2}-\frac{\Delta\tau^K}{h_K}\frac{C}{\zeta_K}\big)\sum_{\tau\in\mathcal{T}_h^K}\big(\varepsilon\|\mathbf{E}_{H,h}^{n,m+\frac{1}{2}}\|_{0,\tau}^2+\mu\|\mathbf{H}_{H,h}^{n,m}\|_{0,\tau}^2\big)
\end{aligned}
$$

Then, for all macro element K, the discrete electromagnetic energy $\mathcal{E}_{H,h}^{n,m+1/2}(K)$ is a positive form of $\mathbf{E}_{H,h}^{n,m+\frac{1}{2}}$ and $\mathbf{H}_{H,h}^{n,m}$ under the condition:

$$\Delta\tau^K < \frac{h_K}{2C}\zeta_K \tag{7.41}$$

Therefore, the global energy $\mathcal{E}_{H,h}^n = \sum_{K\in\mathcal{T}_H}\sum_{m=1}^{M^K}\mathcal{E}_{H,h}^{n,m+1/2}(K)$ is a positive form of all variables under a local CFL condition for each $K\in\mathcal{T}_H$. □

As a result, we have shown that the energy is a bounded positive quadratic form which proves the stability.

7.4 Numerical results

This section presents preliminary results obtained with a MHM-DGTD code with MTS strategy. The Fortran90 code solves the Maxwell's equations in 2D.

7.4.1 Algorithm

First of all, even if the results presented in this part are obtained using a non parallel implementation of the MHM-DGTD method, we want to highlight all the potential of the method in the following description of the algorithm steps of algorithm 3 (see below).

Algorithm 3 Summary of the MHM method with MTS algorithm

1: Set $\mathbf{E}_K^0$ and $\mathbf{H}_K^0$, $\forall K \in \mathcal{T}_H$ from initial conditions
2: **for** $n = 1 \to N$ **do** Set $\mathbf{E}_K^{n,\mathbf{J}} = \mathbf{E}_K^{n-1}$ and $\mathbf{H}_K^{n,\mathbf{J}} = \mathbf{H}_K^{n-1}$, $\forall K \in \mathcal{T}_H$
3: **for** $k \in \mathcal{T}_H$ **do** ▷ Loop on macro elements
4: **for** $m = 1 \to M$ **do** ▷ Loop on local time-steps
5: **if** $n = 1$ **then**
6: Determine $\boldsymbol{\eta}^{\boldsymbol{m},\mathbf{H}}$ from $\boldsymbol{\eta}^{\boldsymbol{m-\frac{1}{2}},\mathbf{E}}$ and $\boldsymbol{\eta}^{\boldsymbol{m-1},\mathbf{H}}$
7: Determine $\boldsymbol{\eta}^{\boldsymbol{m+\frac{1}{2}},\mathbf{E}}$ from $\boldsymbol{\eta}^{\boldsymbol{m-\frac{1}{2}},\mathbf{E}}$ and $\boldsymbol{\eta}^{\boldsymbol{m},\mathbf{H}}$
8: **end if**
9: Determine $\mathbf{H}_K^{n,m,\mathbf{J}}$ from $\mathbf{H}_K^{n,m-1,\mathbf{J}}$ and $\mathbf{E}_K^{n,m-\frac{1}{2},\mathbf{J}}$
10: Determine $\mathbf{E}_K^{n,m+\frac{1}{2},\mathbf{J}}$ from $\mathbf{E}_K^{n,m-\frac{1}{2},\mathbf{J}}$, $\mathbf{H}_K^{n,m,\mathbf{J}}$ and $\mathbf{J}_K^{n,m}$
11: Assemble $\mathbf{E}_K^{n,m+\frac{1}{2},\mathbf{J}}$ to the right-hand side of the global problem
12: **end for**
13: **end for**
14: Solve the global problem and update $\mathbf{E}_K^{n,0}$ and $\mathbf{H}_K^{n,0}$, $\forall K \in \mathcal{T}_H$
15: **end for**

Compared to DG methods for which all the communications between cells are done via the numerical fluxes, the MHM method is almost entirely local if we neglect the cheap global problem in line 14. Each current problem defined in equation (7.28) is associated with a macro element K. They are all considered as single independent problems with their own initial and boundary conditions changing over the time-slabs. As a result, we expect an almost perfect scalability with a parallelization of the local problems. Indeed, tasks 6-7 and 9-10, which represent respectively local problems (7.27) and (7.28), are embarrassingly parallel, *i.e.* that there is no dependency between tasks.

The aforementioned parallel strategy is the intuitive way of thinking the parallelism of the MHM. However, more advanced possibilities are considerable, as for example, a two-level parallelization or a hybrid parallelization could allow to accelerate computationally expensive local problems.

Since the MHM method involves two levels of discretization, all the standard numerical parameters are doubled. The two mesh sizes H and h, as well as the interpolation order of the basis functions, both for the first level space $\boldsymbol{\lambda}_H$, denote by p^e, and for the second level space $\mathbf{V}_h$ denoted by p^k. As shown by Harder *et al* in [HMV16] for elasticity, in the case where $H = h$, the couple (p^e, p^k) must verify $p^k = p^e + 1$ if p^e is odd and $p^k = p^e + 2$ if it is even. We will see that this will influence our results as well.

7.4.2 Convergence of the degenerated scheme

In this subsection, we start with a convergence study of the degenerated time scheme, *i.e.* when there is no second level in time ($\Delta\tau^K = \Delta t$, $\forall K \in \mathcal{T}_H$). For this purpose, we consider the analytical solution of the cavity mode given on a domain $\Omega = [0, 1]^2$ with Perfect Electric Conductor (PEC)

boundary condition on $\partial\Omega$ given by:

$$H^x(t,x,y) = -\frac{\sqrt{2}}{2}\sin(2\sqrt{2}\pi t)\sin(2\pi x)\cos(2\pi y)$$
$$H^y(t,x,y) = \frac{\sqrt{2}}{2}\sin(2\sqrt{2}\pi t)\sin(2\pi x)\cos(2\pi y)$$
$$E^z(t,x,y) = \cos(2\sqrt{2}\pi t)\sin(2\pi x)\cos(2\pi y)$$

with $t \in [0, 2\pi]$ and $(x, y) \in [0, 1] \times [0, 1]$. Those tests are realized with $H = h$, and $\Delta t = \text{CFL}H$. The convergence is measured using the following discrete norm:

$$\max_{n\in\{1,\dots,N\}} \left\|\mathbf{E}^n_{H,h} - \mathbf{E}^n_{\text{ex}}, \mathbf{H}^n_{H,h} - \mathbf{H}^n_{\text{ex}}\right\|_{L^2}. \tag{7.42}$$

We find in figure 7.6 that the numerical solutions converge as $O(h^p)$ in the $||\cdot||_{L^\infty}$ norm, where $p = \min(p^e + 1, p^k, 2)$. The second order Leap-Frog time scheme will restrict the space-time convergence order to 2. As one can see, some points are missing for the case $p^e = 4, p^K = 1$. It comes from a problem with the resolution of the global problem, probably because of the non-compliance with the condition stated above ($p^k = p^e + 2$).

7.4.3 Numerical study of the MTS scheme

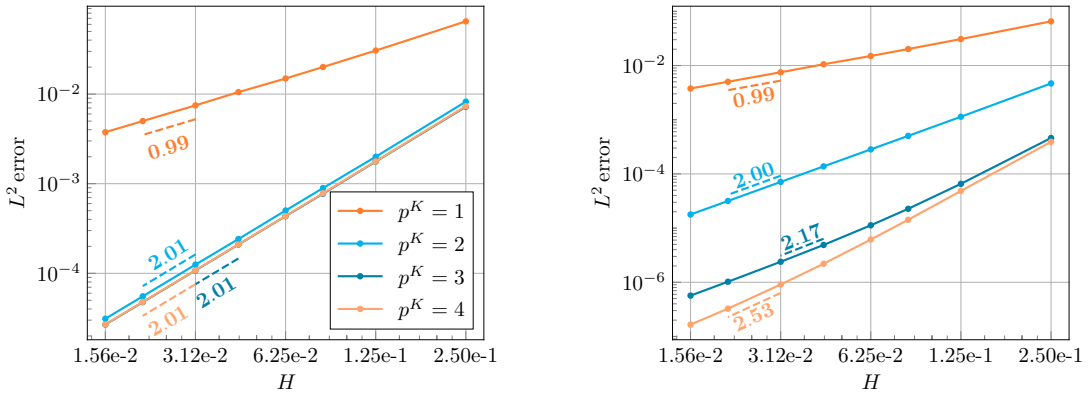
Stability We illustrate the preservation of energy (with $J = 0$) as proved in Theorem 7.3.1. We show in figure 7.7 that energy is numerically preserved when a trapezoidal quadrature rule is used to evaluate the global problem. We also tested the choice of rectangular quadrature to approximate the global problem and even did not observe stability (see orange curve in figure 7.3.1).

Numerical convergence Since the theoretical analysis of convergence has not yet been done, the majority of this part consists of assumptions, factual observations and postulates. From classical analysis, one could naturally expect an error estimate as

$$O(H^{p^e+1} + h^{p^k} + \Delta\tau^k + \Delta t^e)$$

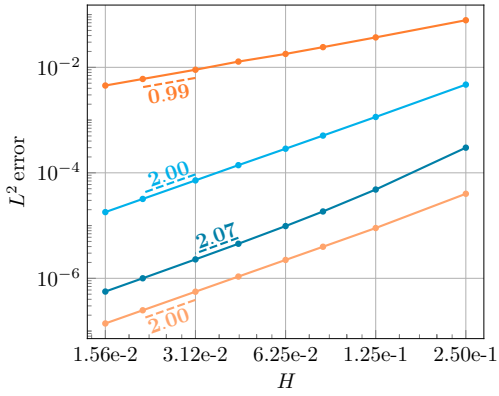
The expectation concerning the H^{p^e+1} comes from [HMV16]. Our first intuition, especially following the degenerated tests and the use of the LF2 time scheme, is that $\min(k, e) = 2$. The following tests aim at showing the behavior of the error when using the MTS time scheme, *i.e.* $\Delta t \neq \Delta\tau^K$ for at least one K.

- Error analysis with respect to Δt (figure 7.8)
  First, we choose $(H, h^K, \Delta\tau^K)$ for all $K \in \mathcal{T}_H$, small enough so as to decouple the error analysis and provide an error analysis with respect to Δt. To do this, we start by choosing the pair of orders $(p^e, p^K) = (1, 3)$ such that the local problems are well posed following [HMV16] ($p^k = p^e + 2$) without refinement at the second level, *i.e.* $H = h^K$, $\forall K \in \mathcal{T}_H$. We also set $\Delta\tau$ as one tenth of the optimal micro time-step which is defined as $\min_{K\in\mathcal{T}_H} CFL\, h^K$. We start from the degenerated case *i.e.* $\Delta t = \Delta\tau^K$, $\forall K \in \mathcal{T}_H$, and then we gradually increase the Δt. It was done for two different levels of spatial refinement, *i.e.* $H/h^K = 1$ for the orange plot, and $H/h^K = 2$ for the blue one. A second-order slope is observed for the bigger time-steps, followed by a saturation of a magnitude close to H^2. This first observation confirms our intuitions that we also have a second order in Δt *i.e.* $e = 2$.

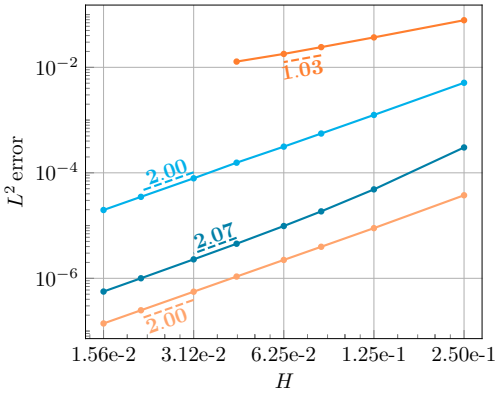


(a) $p^e = 1$

(b) $p^e = 2$



(c) $p^e = 3$



(d) $p^e = 4$

Figure 7.6 | Convergence study of MHM-DGTD method for all the order couples (p^e, p^K) with p^e from 1 (a) to 4 (d) and p^K from 1 (orange) to 4 (light orange).

- Global convergence analysis (figures 7.9)
  Here, we look at the classical convergence, *i.e.* when $(H, h^K, \Delta t, \Delta\tau^K) \to 0$ simultaneously. The pair of order used here is still ($p^e = 1$, $p^k = 3$), with a spatial refinement $\frac{H}{h}$ going from 1 to 8. All the four parameters are linked to each other from the following relations: i) $\frac{H}{h} = \{1, 2, 4, 8\}$ from subfigures (a) to (d); ii) $\Delta\tau^K = \min_{K\in\mathcal{T}_H} CFL\, h^K$; iii) $\Delta t = \alpha\Delta\tau^K$ with $\alpha \in \{1, 5, 19, 15, 20\}$, represented by the different color lines. Thus setting only one of the four parameters to 0 amounts setting all of them to 0. As previously, we observe an order 2 for the degenerated case *i.e.* $\Delta t = \Delta\tau$. Focusing first on the subfigure (a), we observe an order 1. However, as H decreases we remark a change of order from 2 to 1. This does not seem to happen in subfigure (d) but this might be due to the fact that $\frac{H}{h} = 8$ and we might observe the same behavior for H smaller than the values achieved (here $1.56e^{-2}$). From this, one would infer $\min(p^e + 1, p^K, e, k) = 1$.

  Remark *: It is difficult to conclude if it is a loss of order or a limitation of the scheme as we are expecting at best a second order.*

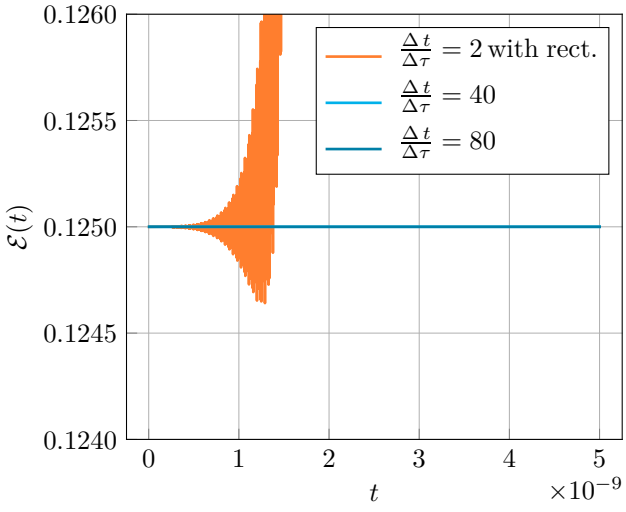


Figure 7.7 | Energy preservation of MHM-DGTD method with LTS and two different integration in time for $\Delta\tau$ constant and increasing Δt.

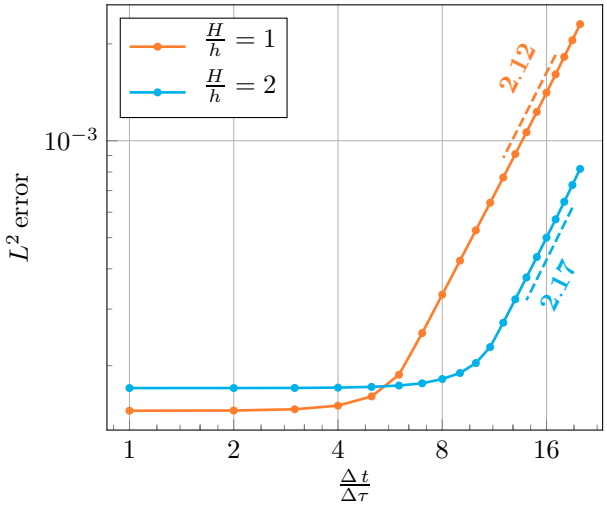
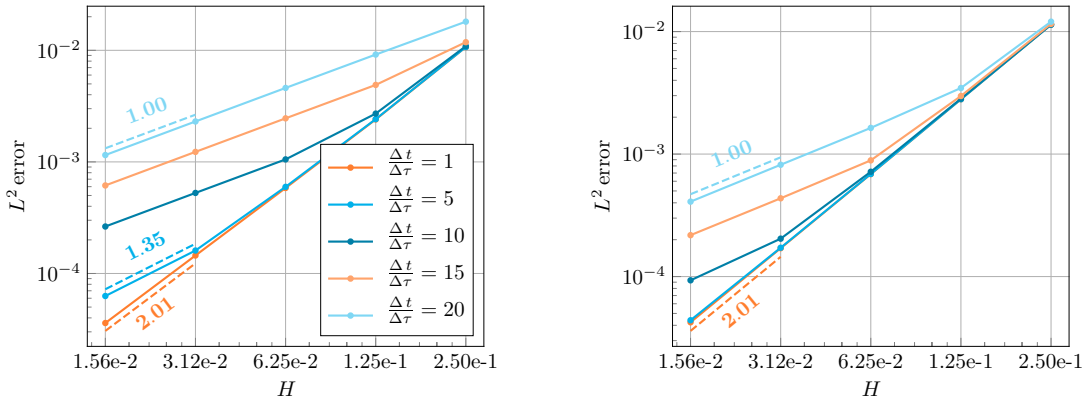


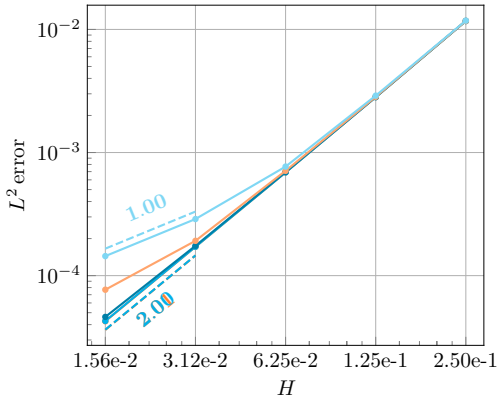
Figure 7.8 | Convergence study of MHM-DGTD method with LTS with $(p^e, p^K) = (1, 3)$, for fixed "small enough" $(h, \Delta\tau, H)$ with $H =$ 3.12e-2, $\Delta\tau = \frac{Ch}{10}$ and $\Delta t \to 0$ (right to left).

- Convergence analysis in terms of the global parameters $(\Delta t, H)$ (figure 7.10) The third case is motivated by the two levels of the MHM method. One can ask how the first level behaves when the error of the second level is negligible. For this case we study the convergence in $(\Delta t, H)$ by decoupling the errors using $(\Delta\tau^K, h^K)$ small enough so as not to pollute the errors in $(\Delta t, H)$. We look at different ratios $\frac{\Delta t}{H}$ in subfigures (a),(b) and (c). We use a couple of orders $(p^e, p^K) = (1, 2)$ as we avoid the case $h^K = H$. The different lines represent several values of $(\Delta\tau^K, h^K)$ with $\Delta\tau^K = CFL\, h$ for each of these ratios. In every tests the order obtained is 2 (or even 2.5). We infer that $\min(e, p^e + 1) = 2$, so $e \leq 2$.

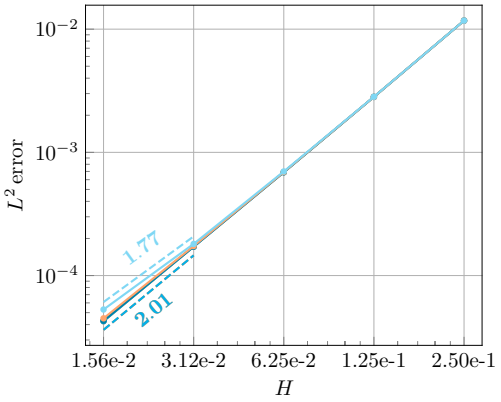


(a) $\frac{H}{h} = 1$

(b) $\frac{H}{h} = 2$



(c) $\frac{H}{h} = 4$



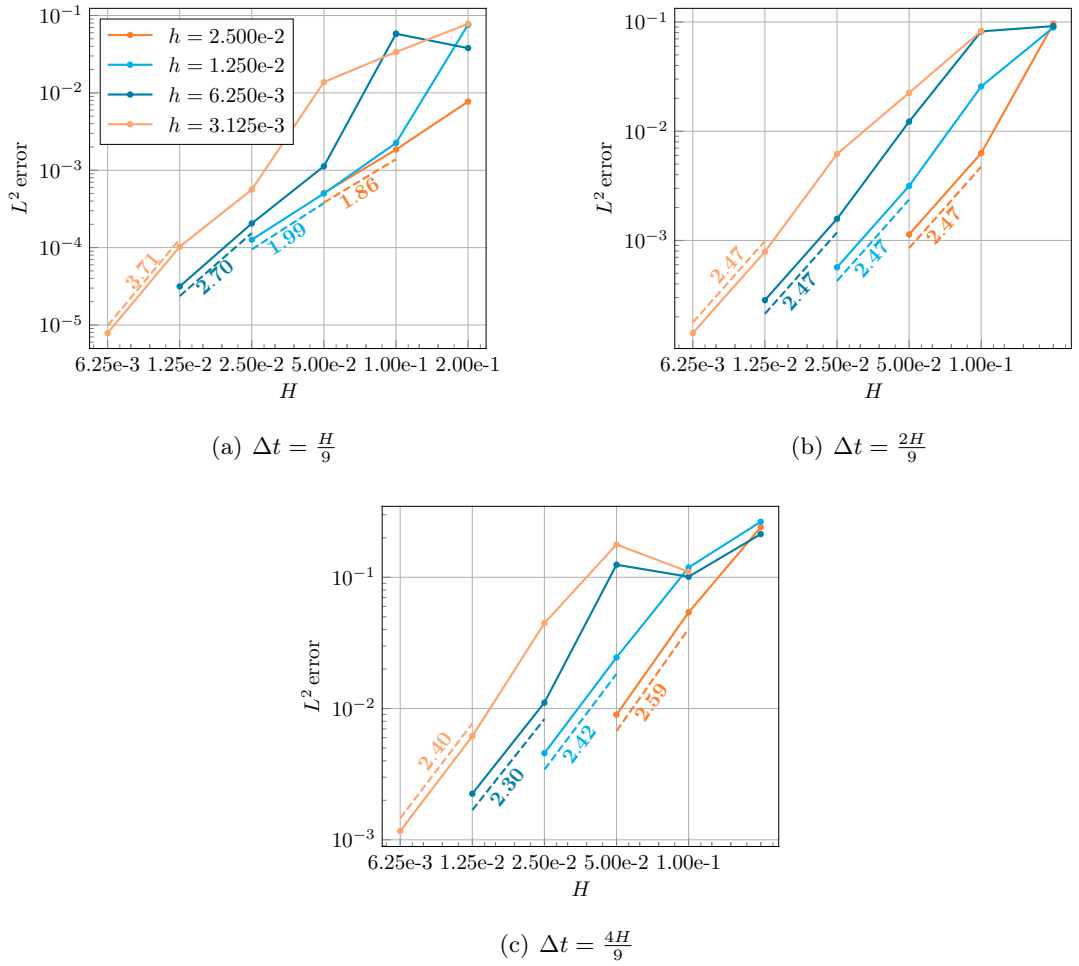
(d) $\frac{H}{h} = 8$

Figure 7.9 | Convergence study of MHM-DGTD method with LTS with $(p^e, p^K) = (1, 3)$ with $(\Delta t, \Delta \tau, H, h) \to 0$ for spatial refinement from 1 to 8 ($h = \frac{H}{\alpha}$ for $\alpha \in [1, 8]$).

In conclusion, we have shown numerically that the MHM method with MTS scheme converges. First of all, we confirmed its unconditional stability numerically with respect to the macro time and space parameters, by showing the energy conservation and its positivity for any $(\Delta t, H)$. Then we tried to investigate the convergence order of the respective parameter.

However, one test shows a decrease from 2 to 1 in the convergence order for specific parameters, especially when the number of local time-steps per global time-step increases. The increasing complexity of local problems can explain this. Indeed, the increase of Δt amounts to increasing the final time of each local problems. We know that doing so increases the error, especially the dispersion one. Therefore it makes the local problems more difficult to solve. To keep the same accuracy, one should refine the local problems. This is a comportment we saw in figure 7.9.

Since for the moment, we cannot rely on a rigorous error analysis, we can only infer that the error estimate might be more intricate than $O(H^{p^e+1} + h^{p^k} + \Delta\tau^k + \Delta t^e)$. Indeed, figure 7.9 might show that a complex relationship between the two levels must be respected to reach the optimal convergence, and even maybe even a super-convergence as in figure 7.10.



(a) $\Delta t = \frac{H}{9}$

(b) $\Delta t = \frac{2H}{9}$

(c) $\Delta t = \frac{4H}{9}$

Figure 7.10 | Convergence study of MHM-DGTD method with LTS for fixed small enough local problems $K(h, \Delta\tau)$ with p^e from 1 (a) to 4 (d) and p^K from 1 (orange) to 4 (light orange).

7.4.4 Comparison with DGTD

7.4.4.1 Photonic structure

In this part, we aimed at comparing the MHM method and the DGTD method on a realistic multiscale nanophotonic case. To do so, we select a prototype problem of a photonic crystal structure tested in [LPSV18]. The nanophotonic device is composed of a silica chip filled with 15 round silicon holes of radius 0.3125. For this study, we place the device in a domain $\Omega = [0, 10] \times [0, 10]$ made of air as shown in figure 7.11. A monochromatic plane wave of frequency 0.15 (dimensionless frequency) is coming from the left of the structure. The permittivity used for the 3 materials are $\varepsilon = 1$ for the surrounding air, $\varepsilon = 1.5$ for the silica chip and $\varepsilon = 3.14$ for the silicon holes. Absorbing Boundary Condition (ABC) are used on the whole boundary $\partial\Omega$.

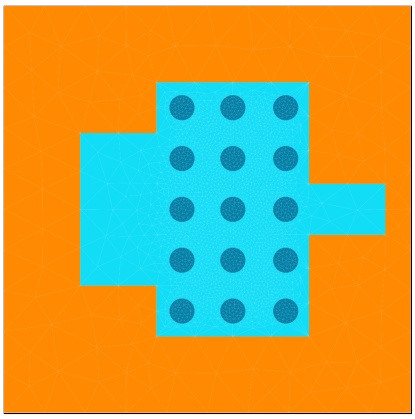


Figure 7.11 | Geometry of the nanowaveguide considered. The surrounding is made of air (orange), the chip is composed of silica (blue) with holes of silicium (dark blue).

7.4.4.2 Numerical features

To be able to compute a numerical error for this realistic case, we start by computing one reference solution. To do so, we use DGTD method on a highly refined mesh with $\mathbb{P}_2$ basis functions. The reference solution is plot in figure 7.14(a). We choose to compute the error on nine fixed points, visible in the figure 7.12, instead of projecting the solutions on the same mesh.

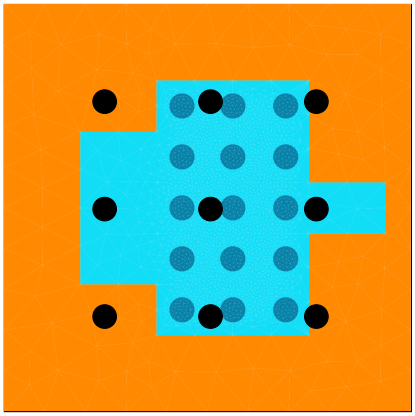
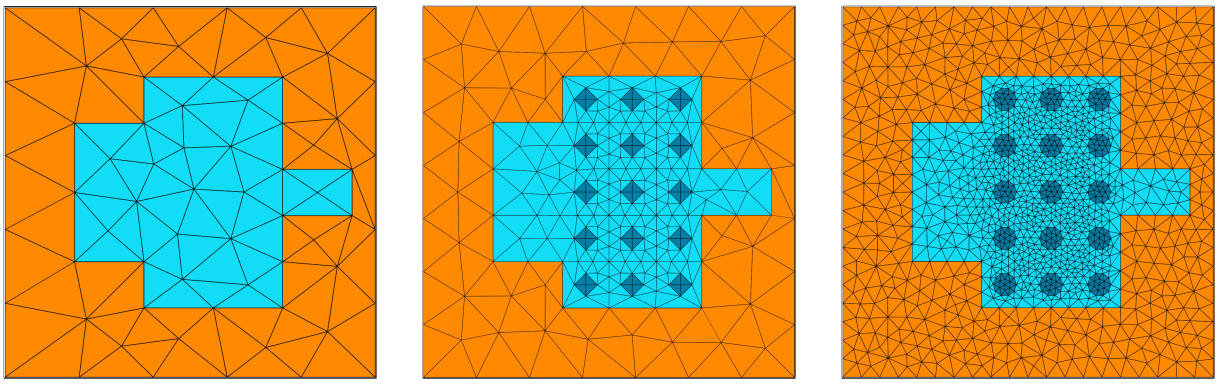


Figure 7.12 | Points on which the errors is computed. We selected 9 points uniformly distributed in order to compute the error between a reference solution and the bench runs.

The study is carried out as follows: i) We take a very coarse mesh M1, not fitting to the geometry (see in figure 7.13(a) that M1 is not meshing the spheres). We show that the MHM on this mesh obtains a correct error. ii) More and more refined meshes are used for the DGTD method until the MHM error is more or less reached. We used three meshes as shown in figure 7.13.

7.4.4.3 Results

The information related to the three mesh sizes used is presented in the table 7.1 as well as the maximum error at the control points. One can see that on the very coarse mesh, the MHM method



(a) Coarser mesh M1 (b) Coarse mesh with meshed spheres M2 (c) Fine mesh M3

Figure 7.13 | Meshes used for the methods comparison. All meshes are aligned with the silicium device but M1 is not conform with the spheres.

is performing way better than the DGTD method. This is predictable given the absence of the silica holes in M1. However, to reach a solution close to the MHM results, we use a mesh with nearly 40 times more d.o.f.. Of course, we omit the second level discretization deliberately. However, we can see from the accuracy of the visualization of figure 7.14(b), that the structure is thinly discretized.

Table 7.1 | Comparison between MHM and DGTD on several meshes.

	Mesh	Number of d.o.f.	$\max \lvert \mathbf{E} - \mathbf{E}_{ref} \rvert$
MHM	M1	2450	4.82×10^{-2}
DGTD-$\mathbb{P}_4$	M1	7230	2.27
	M2	21690	8.82×10^{-1}
	M3	107730	2.79×10^{-1}
	M4	184260	5.15×10^{-2}

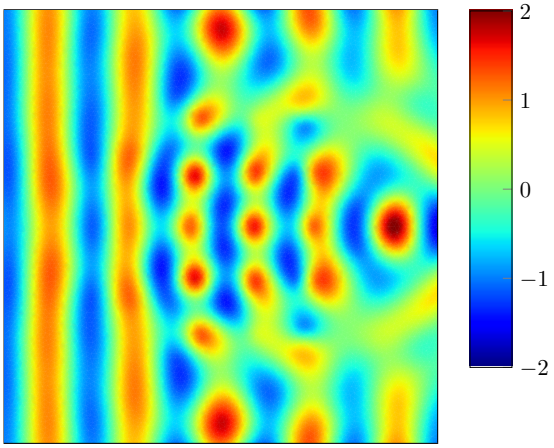
7.4.5 Computation times

Now that we have shown that the MHM can precisely solve a complex problem on a very coarse mesh, we will look at the performance of the method in terms of computation times.

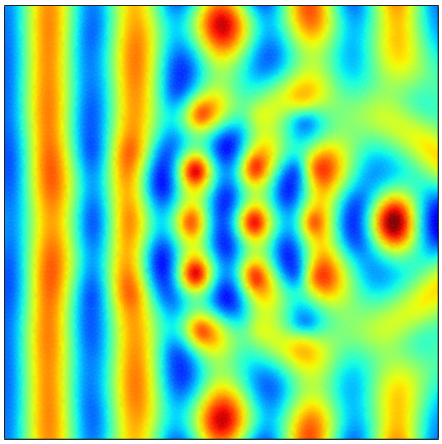
For each simulation, we retrieve the times of the different computation phases in both solvers, *i.e.* the computation of the multiscale basis functions $\boldsymbol{\eta}^{\boldsymbol{\xi}}$ before the time-loop, the resolution of local problems to retrieve $(\mathbf{E}^J, \mathbf{H}^J)$ at each time-step, the resolution of the global problem, and the summation $\mathbf{E} = \mathbf{E}^J + \mathbf{E}^{\boldsymbol{\lambda}}$. The two firsts are referred to as local times while the two lasts are referred to as the global times.

In figure 7.15, we present the comportment of the MHM method in regards to the macro time-step used. Fine time-steps are set thanks to the micro CFL, and we increase Δt such that the ratio $\Delta t / \Delta \tau^K$ goes from 1 to 8. We observe a quasi-optimal scalability in terms of global times.

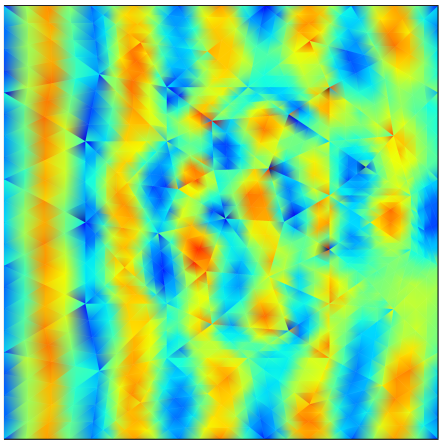
Remark *: We do not compare computational times between MHM and DGTD. Indeed: (i) the MHM targets in particular potentially unsolvable problems for the DGTD method and should com-*



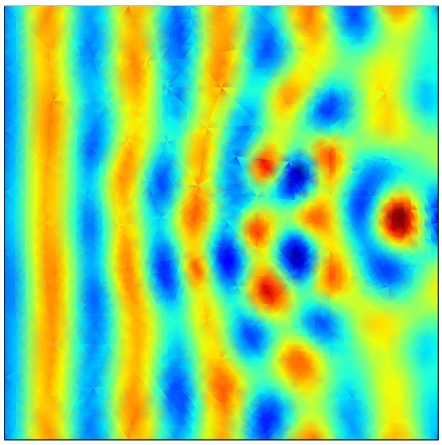
(a) Reference solution



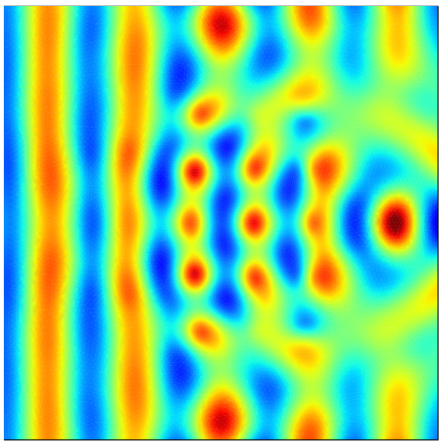
(b) MHM solution on M1 (1225 d.o.f.)



(c) DGTD solution on M1 (7230 d.o.f.)



(d) DGTD solution on M2 (21690 d.o.f.)



(e) DGTD solution on M3 (107730 d.o.f.)

Figure 7.14 | Solution (E_z) for the propagating monochromatic plane wave in a silicium nano-waveguide. The results was obtained for 3 different meshes. The number of d.o.f. in (a) only takes into account the first level problem.

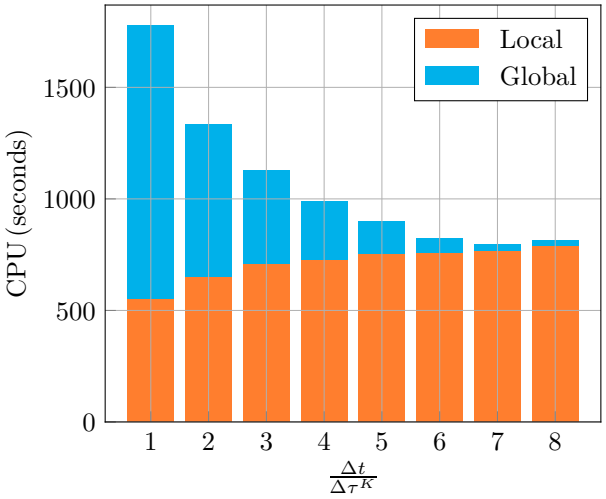


Figure 7.15 | Representation of the computational times evolution when increasing Δt with the MTS time scheme.

pete on those type problems; (ii) the MHM needs to be parallelized in order to compete; and (iii) it is necessary to understand the interplay between the parameters $(\mathcal{T}_H, \mathcal{T}_{\Delta t}, p^e, p^K)$ *in order to use the method as efficiently as possible.*

7.5 Conclusion

In this chapter we presented a MHM method using a DGTD method at the second level. We proposed a MTS time scheme to benefit from the independence of local problems. We proved the preservation of the energy and the stability of the method using a trapezoidal rule for the computation of the global problem, and we illustrated it numerically. However, a theoretical study of the convergence should help to understand the behavior of the error with MTS scheme. Moreover, the use of higher-order time scheme is also an objective for the future of the explicit MHM.

We also showed the capacity of the MHM method to solve accurately complex, realistic problems on very coarse meshes. However, for the moment, the usefulness of the method is limited. Indeed, its high computational cost requires an adapted parallelism to be competitive or at least reasonable in terms of computation time. Several levels of parallelism can be considered, with the most logical first approach being the parallelization of local problems. One can also think about using parallel DGTD solver for the local problems.

We also think that the method can be well suited for implicit time scheme and for frequency-domain problems [CFV19]. Indeed, the bottleneck of these methods for real test cases is, most of the time, the amount of memory available to store the linear system. However, in the same way, that HDG method did, MHM method is spreading out the computation load in the local problems whereas the global linear system to solve involves only the skeleton of the coarse mesh.

8

Conclusion

Contents

The purpose of this last chapter is to summarize the content of this manuscript and to identify possible extensions.

8.1 Summary

This thesis aimed at proposing a 3D discontinuous Galerkin time-domain method able to simulate complex and realistic PV devices. First, an introduction to the context of photovoltaics, and its link with electromagnetism has been made. A presentation of Maxwell's equations, and its coupling with the generalized dispersive model was done. This dispersion model can represent accurately a wide diversity of dispersive materials on a broad band, and especially semiconductors which are one of the main components in solar cells. Then we have presented the DGTD method with its spatial discretization. Time discretization using RK and LSRK schemes has also been introduced with some words on stability. The imposition of a plane wave in the computational domain has been discussed, as well as the decomposition of the domain in a total field region and a scattered field region. Finally, the domain truncature has been discussed, with the introduction of PMLs, PBCs and the other classical boundary conditions.

In the third chapter, we have presented two methodological contributions that aimed to improve the performance and the accuracy of the DGTD method in the particular for solar cell problems. In a first part, the possibility to use local approximation order was investigated. The selected

order repartition is based on the number of points per wavelength, which is required to obtain a correct approximation of wave phenomena. This strategy was applied on a realistic silicium solar cell. Both accuracy of the approximation ($< 1\%$ relative error) and sequential speed-up (≈ 2) was interesting. The second contribution consist of using hybrid non-conforming meshes. This strategy allowed the use of a refined unstructured mesh at the material junctions while having a structured coarse grid in the homogeneous regions. This strategy has been tested and validated on simple test cases and on the realistic solar cell. Coupling both strategies showed an even bigger speed-up (≈ 3).

In the fourth chapter, we tried to demonstrate the capabilities of our DGTD solver on realistic photovoltaic cases. In the first part we presented state-of-the-art methods to deal with such applications. The most important quantities of interest were presented (Absorption, EQE, Jsc). Then, we showed how to deal with geometrical measures obtained in laboratory. Different pre- and post-processing were presented (smoothing raw measures, building periodic meshes from random textures) using methods taken from digital image processing. After that, we have presented the simulations of different type of solar cells using DGTD. The first one comes from a collaboration with C2N and IPVF on GaAs based solar cells. We compared the results obtained from different method. The second case dealt with solar cells based on conical gratings. Those particular cells enhanced a lot of resonances, which required a long physical time to catch. The last one consists of a very large solar cell with random textures obtained by AFM. We showed the capability of the DGTD solver to deal with large geometries ($10\mu m^2$) and big meshes (more than one million cells).

In the fifth chapter, we coupled our DGTD solver with an optimization method called Efficient Global Optimization (EGO). This method is based on a metamodelling, which aims to replicate the objective function. This method is particularly well adapted to optimize expensive and unknown (black-box) objective function. Several optimizations were conducted. The first one is the enhancement of color rendering of a solar cell, for which we were able to provide a nanotexturation that reduces and unifies the parasitic reflections, especially in the red / near infrared region. The second case is the optimization of a pyramid based nanotexturation. The main idea here was to maximize the density of short-circuit current while keeping a fixed quantity of absorber (InGaAs). The q-EGO method was used for this case, which allow parallel evaluation of the objective function per optimization step. The optimized design here increased the short-circuit current density by 23%. Load unbalances were observed during the optimization loop, especially when the configuration present huge differences.

In the sixth chapter, the parallel performances of our code has been assessed. In a first part we presented the parallel implementation of our solver, which relied on a partition of the mesh. A numerical experiment with the Metis mesh partitioning tool and an uniform weighting, speedups and efficiencies of our parallel MPI implementation were assessed on a realistic solar cell case. Bothh hybrid and tetrahedral meshes were tested and compared. Results showed that this simple weighting worked better for tetrahedral meshes. For a good scaling with hybrid meshes, we added a constraint for the partitioning of hybrid faces, which allowed us to solve this issue. We also proposed an iterative partitioning algorithm, which used ongoing parallel simulation to weight the different mesh cells. This algorithm allows to balance the load on each CPU, especially when the considered case has complex features (PMLs, TFSF, observable computations). On a realistic solar cell, the load unbalance went from 15% to 2%. Preliminary results on hybrid MPI/openMP parallelization were presented.

In the last chapter, we have presented of a recent method. The MHM is particularly well adapted to deal with multiscale problems with geometric details on coarse mesh. To be used for realistic applications, the MHM needs a second numerical method to solve local problems in the

element of the coarse partition. We chose to use DGTD with centered flux. We also proposed a particular time-stepping algorithm, which allowed us to completely decoupled local problems in time. The stability has been proven, with numerical experiments in accordance with the theory. Preliminary results concerning the convergence of the methods regarding both time and space variables have been conducted.

8.2 Perspectives

We envision the following possible extensions and perspectives for the different topics we addressed in this thesis.

8.2.1 Material models

Some of the simulation of direct bandgap materials in this work presented an absorption above the bandgap, which is unphysical. Indeed, this region is characterized by an exponential decrease of the imaginary part of the permittivity towards zero called exponential Urbach tail. This characteristic is not easily taken into account in the generalized dispersion model. Indeed, we showed some improvement in this region with very fine tuning of the simulated annealing algorithm, but there is still parasitic absorption. The modeling of such behavior requires the proposal of an other dispersion model, which can represent the Urbach tail. Exponential models seems to handle this features and could be interesting, but a preliminary study about its time-domain writing needs to be done.

8.2.2 Solution accuracy

The methodological contributions presented in this thesis were relatively simple, in the sense that they were based on simple criteria. The tuning of the different parameters generally relies on multiple experiments. Those strategies could be coupled with an *a posteriori* error estimate, which will allow an adaptation of the polynomial order as well as the refinement of the mesh in an automatic computation framework.

8.2.3 Photovoltaics and semiconductor physics

In this thesis, we treated the optical part of the photovoltaic. To simulate in a broader sense the absorption in solar cells, it would be interesting to consider, in addition to the optical model which characterizes the absorption of photons, an electrical model for the simulation of charge carriers transport. The electronic step can be represented threw a drift-diffusion electronic (DDE) model to calculate the current curve of the cell, and so its efficiency. This has been done in [ACM] by coupling a RCWA solver to a HDG solver.

8.2.4 Multiscale Hybrid-Mixed method

Several extensions are needed for the MHM method to compete and be fully compared to other numerical methods. At first, to deal with classical solar cell configurations, we should couple the method with the general dispersion model, and see how it is impacting both locals and global problems. In terms of performance, it would be interesting to work on different parallel versions of the method. Indeed, the divide-and-conquer framework of the method seemed to allow the use of several levels of parallelism: each local problems can be solved on its computer node, and

the DGTD solver can also be run in parallel. Finally, an implementation of the method for 3D Maxwell's equations is expected in order to reproduce different realistic results of this thesis.

Bibliography

[AAPG14] L. D. Angulo, J. Alvarez, M. F. Pantoja, and S. G. Garcia. An explicit nodal space-time discontinuous galerkin method for maxwell's equations. IEEE Microwave and Wireless Components Letters, 24:827–829, 2014.

[ABK+19] L. C. Andreani, A. Bozzola, P. Kowalczewski, M. Liscidini, and L. Redorici. Silicon solar cells: toward the efficiency limits. Advances in Physics: X, 4(1):1548305, 2019.

[ACM] T. H. Anderson, B. J. Civiletti, and P. B. Monk. Coupled Optoelectronic Simulation and Optimization of Thin-Film Photovoltaic Solar Cells. Technical report.

[AHPV13] R. Araya, C. Harder, D. Paredes, and F. Valentin. Multiscale Hybrid-Mixed Method. SIAM Journal on Numerical Analysis, 51(6):3505–3531, 2013.

[ARK06] A. Abou-Rjeili and G. Karypis. Multilevel algorithms for partitioning power-law graphs. In 20th International Parallel and Distributed Processing Symposium, IPDPS 2006, volume 2006. IEEE Computer Society, 2006.

[Bec39] M. E. Becquerel. Memoire sur les effets electriques produits sous l'influence des rayons solaires. Comptes Rendus Hebdomadaires des Seances de L'Academie des Sciences, 9:561–567, 1839.

[BEH+16] A. J. Bett, J. Eisenlohr, O. Höhn, P. Repo, H. Savin, B. Bläsi, and J. C. Goldschmidt. Wave optical simulation of the light trapping properties of black silicon surface textures. Optics Express, 24(6):A434, mar 2016.

[Bér07] J.-P. Bérenger. Perfectly matched layer for computational electromagnetics. Morgan & Claypool, first edition, 2007.

[BFM11] S. C. Baker-Finch and K. R. McIntosh. Reflection of normally incident light from silicon solar cells with pyramidal texture. Progress in Photovoltaics: Research and Applications, 19(4):406–416, jun 2011.

[BHS+12] C. Battaglia, C. M. Hsu, K. Söderström, J. Escarré, F. J. Haug, M. Charrière, M. Boccard, M. Despeisse, D. T. Alexander, M. Cantoni, Y. Cui, and C. Ballif. Light trapping in solar cells: can periodic beat random? ACS Nano, 6(3):2790–2797, 2012.

[BPG02] E. Bécache, P. G. Petropoulous, and S. D. Gedney. On the long-time behavior of unsplit perfectly matched layers. Technical report, INRIA, 2002.

[But87] J. C. Butcher. The Numerical Analysis of Ordinary Differential Equations: Runge-Kutta and General Linear Methods. Wiley, 1987.

[BVF+17] H. Brüns, A. Vogt, C. Findeklee, A. Schröder, M. Magdowski, M. P. Robinson, F. Heidler, and C. Schuster. Modeling challenging emc problems using the method of moments. Electromagnetic Compatibility Magazine, IEEE, March 2017.

[Can] A. C. Cangellaris. Frequency-Domain Finite Element Methods for Electromagnetic Field Simulation: Fundamentals, State of the Art, and Applications to EMI/EMC Analysis.

[CCDL+19] H.-L. Chen, A. Cattoni, R. De Lépinau, A. W. Walker, O. Höhn, D. Lackner, G. Siefer, M. Faustini, N. Vandamme, J. Goffard, B. Behaghel, C. Dupuis, N. Bardou, F. Dimroth, and S. Collin. A 19.9%-efficient ultrathin solar cell based on a 205-nm-thick GaAs absorber and a silver nanostructured back mirror. Nature Energy, aug 2019.

[CDL] A. Christophe, S. Descombes, and S. Lanteri. An implicit hybridized discontinuous Galerkin method for the 3D time-domain Maxwell equations.

[CFP54] D. M. Chapin, C. S. Fuller, and G. L. Pearson. A new silicon p-n junction photocell for converting solar radiation into electrical power. Journal of Applied Physics, 25(5):676–677, 1954.

[CFV19] T. Chaumont-Frelet and F. Valentin. A multiscale hybrid-mixed method for the Helmholtz equation in heterogeneous domains. Technical report, 2019.

[CH15] T. Cai and S. E. Han. Effect of symmetry in periodic nanostructures on light trapping in thin film solar cells. Journal of the Optical Society of America B, 32(11):2264, nov 2015.

[Cia78] P. G. Ciarlet. The finite element method for elliptic problems. North-Holland Pub. Co., 1978.

[CK94] M. H. Carpenter and C. A. Kennedy. Fourth-order 2n-storage runge-kutta schemes. Technical report, National Aeronautics and Space Administration, 1994.

[CL13] J. Chen and Q. H. Liu. Discontinuous Galerkin time-domain methods for multiscale electromagnetic simulations: A review, 2013.

[CLM87] A. C. Cangellaris, C. C. Lin, and K. K. Mei. Point-Matched Time Domain Finite Element Methods for Electromagnetic Radiation and Scattering. IEEE Transactions on Antennas and Propagation, 35(10):1160–1173, 1987.

[CP08] K.R. Catchpole and A. Polman. Plasmonic solar cells. Optics Express, 16(26):21793–21800, Dec 2008.

[DC12] R. Duvigneau and P. Chandrashekar. Kriging-based optimization applied to flow control. International Journal for Numerical Methods in Fluids, 69:1701–1714, 2012.

[DDH] A. Ditkowski, K. Dridi, and J. S. Hesthaven. Convergent cartesian grid methods for Maxwell's equations in complex geometries. Journal of Computational Physics.

[DG09] J. Diaz and M. Grote. Energy conserving explicit local time stepping for second-order wave equations. SIAM Journal of Scientific Computing, 31:1985–2014, 2009.

[DGB+14] F. Dimroth, M. Grave, P. Beutel, U. Fiedeler, C. Karcher, T. N. D. Tibbits, E. Oliva, G. Siefer, M. Schachtner, W. Alexander, A. W. Bett, R. Krause, M. Piccin, N. Blanc, C. Drazek, E. Guiot, B. Ghyselen, T. Salvetat, A. Tauzin, T. Signamarcheix, A. Dobrich, T. Hannappel, and K. Schwarzburg. Waferbondedfour-junction GaInP/GaAs//GaInAsP/GaInAs concentrator solar cells with 44.7% efficiency. Progress in photovoltaics: research and applications, 22:277–282, 2014.

[DI16] N. Das and S. Islam. Design and analysis of nano-structured gratings for conversion efficiency improvement in gaas solar cells. Energies, 9:690, 08 2016.

[DLG16] R. Duvigneau, J. Labroquère, and E. Guilmineau. Comparison of turbulence closures for optimized active control. Computers & Fluids, 124:67–77, 2016.

[DPT17] D. A. Di Pietro and R. Tittarelli. An introduction to Hybrid High-Order methods. mar 2017.

[Dru00] P. Drude. Zur elektronentheorie der metalle. Annalen der Physik, 306:566–613, 1900.

[DS13] M. B. Dühring and O. Sigmund. Optimization of extraordinary optical absorption in plasmonic and dielectric structures. Journal of the Optical Society of America B, 30(5):1154–1160, 2013.

[DTM+17] V. Depauw, C. Trompoukis, I. Massiot, W. Chen, A. Dmitriev, P. R. Cabarrocas, I. Gordon, and J. Poortmans. Sunlight-thin nanophotonic monocrystalline silicon solar cells. Nano Futures, 1(2):021001, aug 2017.

[Dur13] C. Durochat. Méthode de type galerkin discontinu en maillages multi-éléments pour la résolution numérique des équations de maxwell instationnaires. 2013.

[EG06] A. Ern and J.-L. Guermond. Evaluation of the condition number in linear systems arising in finite element approximations. ESAIM: Mathematical Modelling and Numerical Analysis, 40(1):29–48, 2006.

[EJD14] S. Eyderman, S. John, and A. Deinega. Near perfect solar absorption in ultra-thin-film {GaAs} phtonic crystals. Journal of Materials Chemistry A, 2:761–769, 2014.

[FBBV06] C. Fumeaux, D. Baumann, P. Bonnet, and R. Vahldieck. Developments of finite-volume techniques for electromagnetic modeling in unstructured meshes. In 17th Int. Zurich Symp. Electromagn. Compat. 2006, volume 2006, pages 5–8, 2006.

[Fin18] Bloomberg New Energy Finance. Global trends in renewable energy investment 2018. 2018.

[FLLP05] L. Fezoui, S. Lanteri, S. Lohrengel, and S. Piperno. Convergence and stability of a discontinuous galerkin time-domain method for the 3d heterogeneous maxwell equation on unstructured meshes. ESAIM : Mathematical Modelling and Numerical Analysis, 39:1149–1176, 2005.

[FMA10] V. E. Ferry, J. N. Munday, and H. A. Atwater. Design Considerations for Plasmonic Photovoltaics. Advanced Materials, 22:4794–4808, 2010.

[GBM15] B. Gallinet, J. Butet, and O. J.F. Martin. Numerical methods for nanophotonics: Standard problems and future challenges. Laser and Photonics Reviews, 9(6):577–603, 2015.

[GC15] P. Genevet and F. Capasso. Holographic optical metasurfaces: a review of current progress. Reports on Progress in Physics, 78(2):024401, 2015.

[GCA+17] P. Genevet, F. Capasso, F. Aieta, M. Khorasaninejad, and R. Devlin. Recent advances in planar optics: from plasmonic to dielectric metasurfaces. Optica, 4(1):139, 2017.

[GLRC08] D. Ginsbourger, R. Le Riche, and L. Carraro. A Multi-points Criterion for Deterministic Parallel Global Optimization based on Gaussian Processes A Multi-points Criterion for Deterministic Parallel Global Optimization based on Gaussian Processes A Multi-points Criterion for Deterministic Parallel Global Optimization based on Gaussian Processes. Technical report, 2008.

[GME02] C. A. Gueymard, D. Myers, and K. Emery. Proposed reference irradiance spectra for solar energy systems testing. Solar Energy, 73(6):443–467, 2002.

[GMY13] V. Ganapati, O. D. Miller, and E. Yablonovitch. Light Trapping Textures Designed by Electromagnetic Optimization for Sub-Wavelength Thick Solar Cells. 4(1):175–182, 2013.

[Gol01] J. Goldemberg. World energy assessment. energy and the challenge of sustainability. Sep 2001.

[GR09] C. Geuzaine and J.-F. Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre-and post-processing facilities. Technical report, 2009.

[Gre01] M. A. Green. Third generation photovoltaics: Ultra-high conversion efficiency at low cost. Progress in Photovoltaics: Research and Applications, 9(2):123–135, 2001.

[GY10] E. Garnett and P. Yang. Light Trapping in Silicon Nanowire Solar Cells. Nano Letters, 10(3):1082–1087, mar 2010.

[HDF06] M. Han, R. W. Dutton, and S. Fan. Model dispersive media in fdtd method with complex-conjugate pole-residue pairs. Microwave and Wireless Components Letters, 16:119–121, 2006.

[HLY13] Y. Huang, J. Li, and W. Yang. Modeling backward wave propagation in metamaterials by the finite element time-domain method. SIAM Journal of Scientific Computing, 35:248 – 274, 2013.

[HMV16] C. Harder, A. L. Madureira, and F. Valentin. A hybrid-mixed method for elasticity. ESAIM: Mathematical Modelling and Numerical Analysis, 50(2):311–336, 2016.

[HOV16] P. Henning, M. Ohlberger, and B. Verfürth. A new heterogeneous multiscale method for time-harmonic Maxwell's equations. SIAM Journal on Numerical Analysis, 54(6):3493–3522, 2016.

[HR98] Y. Hao and J. C. Railton. Analyzing electromagnetic structures with curved boundaries on cartesian (fdtd) meshes. IEEE Transactions on Antennas Propagation, 46:82 – 88, 1998.

[HS81] R. Holland and L. Simpson. Finite-Difference Analysis of EMP Coupling to Thin Struts and Wires. IEEE Transactions on Electromagnetic Compatibility, EMC-23(2):88–97, 1981.

[HV08] D. Huybrechs and S. Vandewalle. An efficient implementation of boundary element methods for computationally expensive Green's functions. Engineering Analysis with Boundary Elements, 32(8):621–632, aug 2008.

[HW08] J. S. Hesthaven and T. Warburton. Nodal discontinuous Galerkin methods. Springer, 2008.

[JLOZ15] K. Jäger, D. N.P. Linssen, I. Olindo, and M. Zeman. Ambiguities in optical simulations of nanotextured thin-film solar cells using the finite-element method. Optics Express, 23(19):A1060, 2015.

[JSW98] D. R. Jones, M. Schonlau, and W. J. Welch. Efficient Global Optimization of Expensive Black-Box Functions. Technical report, 1998.

[KGV83] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. Science, 220:671–680, 1983.

[KM96] M. Kuzuoglu and R. Mittra. Frequency dependence of the constitutive parameters of causal perfectly matched layers. IEEE Microwave and Guided Wave Letters, 6:447–449, 1996.

[KM10] Andreas M. K. and Olivier J. F. M. Pitfalls in the Determination of Optical Cross Sections From Surface Integral Equation Simulations. IEEE Transactions on Antennas and Propagation, 58(6):576–585, 2010.

[Kön11] M. König. Discontinuous Galerkin Methods in Nanophotonics. PhD thesis, Fakultät für Physik des Karlsruher Instituts für Technologie, 2011.

[Lam91] J. D. Lambert. Numerical Methods for Ordinary Differential Systems. Wiley, 1991.

[LGAB18] S. Lanteri, A. Gobé, U. Aeberhard, and K. Bittkau. Rigorous modeling of light absorption in nanostructured materials using a parallel high order finite element time-domain technique. In Computational Optics 2018, Francfort, Germany, May 2018.

[LL60] L. D. Landau and E. M. Lifshitz. Electrodynamics of Continuous Media. Pergamon Press, 1960.

[Loc12] D. Lockau. Optical modeling of thin film silicon solar cells with random and periodic light management textures. Http://Opus.Kobv.De/Tuberlin/Volltexte/2013/3875/, page 179, 2012.

[LPSV18] S. Lanteri, D. Paredes, C. Scheid, and F. Valentin. The multiscale hybrid-mixed method for the maxwell equations in heterogeneous media. Multiscale Modeling and Simulation: A SIAM Interdisciplinary Journal, December 2018.

[LVD+14] R. Léger, J. Viquerat, C. Durochat, C. Scheid, and S. Lanteri. A parallel non-conforming multi-element DGTD method for the simulation of electromagnetic wave interaction with metallic nanoparticles. Journal of Computational and Applied Mathematics, 270:330–342, 2014.

[LYE+17] W. K. Lee, S. Yu, C. J. Engel, T. Reese, D. Rhee, W. Chen, and T. W. Odom. Concurrent design of quasi-random photonic nanostructures. Proceedings of the National Academy of Sciences of the United States of America, 114(33):8734–8739, aug 2017.

[Mai07] S. A. Maier. Plasmonics: Fundamentals and Applications. Springer, 2007.

[Max65] J. C. Maxwell. A Dynamical Theory of the Electromagnetic Field Email alerting service. Philosophical Transactions of the Royal Society A, 1865.

[MC12] S. Mokkapati and K. R. Catchpole. Nanophotonic light trapping in solar cells. Journal of Applied Physics, 112(10):101101, 2012.

[MG81] M. G. Moharam and T. K. Gaylord. Rigorous coupled-wave analysis of planar-grating diffraction. Journal of the Optical Society of America, 71(7):811, jul 1981.

[MLD+08] P. Matheu, S. H. Lim, D. Derkacs, C. McPheeters, and E. T. Yu. Metal and dielectric nanoparticle scattering for improved optical absorption in photovoltaic devices. Applied Physics Letters, 93(11):113108, 2008.

[Moc89] J. Mockus. Global Optimization and the Bayesian Approach. pages 1–3. 1989.

[Mon03] P. Monk. Finite Element Methods for Maxwell's Equations. Oxford Science Publications, 2003.

[Moy13] L. Moya. Locally implicit discontinuous Galerkin time-domain methods for electromagnetic wave propagation in biological tissues. PhD thesis, Université Nice Sophia-Antipolis, 2013.

[Néd80] J. C. Nédélec. Mixed finite elements in $\mathbb{R}^3$. Numerische Mathematik, 35:315 – 341, 1980.

[NDB12] J. Niegemann, R. Diehl, and K. Busch. Efficient low-storage runge-kutta schemes with optimized stability regions. Journal of Computational Physics, 231:364–372, 2012.

[NH07] B. Novotny and L. Hecht. Principles of nano-optics. Cambridge University Press, first edition, 2007.

[Pae12] U. W. Paetzold. Light trapping with plasmonic back contacts in thin- film silicon solar cells. Technical report, 2012.

[PCT+16] A. W. Powell, D. M. Coles, R. A. Taylor, A. A. R. Watt, H. E. Assender, and J. M. Smith. Plasmonic Gas Sensing Using Nanocube Patch Antennas. Advanced Optical Materials, 4(4):634–642, 2016.

[PFT00] S. Petersen, C. Farhat, and R. Tezaur. A space-time discontinuous galerkin method for the solution of the wave equation in the time-domain. International Journal for Numerical Methods in Engineering, 0:1–6, 2000.

[Pip06] S. Piperno. Symplectic local time-stepping in non-dissipative dgtd methods applied to wave propagation problems. ESAIM: Mathematical Modelling and Numerical Analysis, 40:815–841, 2006.

[PLD05] P. Panek, M. Lipiński, and J. Dutkiewicz. Texturization of multicrystalline silicon by wet chemical etching for silicon solar cells. Journal of Materials Science, 40(6):1459–1463, Mar 2005.

[REN19] REN21. Renewables 2019 global status report. 2019.

[RF98] M. Remaki and L. Fezoui. Une méthode de Galerkin discontinu pour la résolution des équations de Maxwell en milieu héterogene. Technical report, Inria Sophia Antipolis, Project-team Caïman, 1998.

[RG00] J. A. Roden and S. D. Gedney. Convolutional pml (cpml): An efficient fdtd implementation of the cfs-pml for arbitrary media. Microwave and Optical Technology Letters, 27:334–339, 2000.

[RGNR11] H. K. Raut, V. A. Ganesh, A. S. Nair, and S. Ramakrishna. Anti-reflective coatings: A critical, in-depth review, oct 2011.

[RT77] P. A. Raviart and J. M. Thomas. Primal Hybrid Finite Element Methods for 2nd Order Elliptic Equations. Technical report, 1977.

[Sha49] C. E. Shannon. Communication in the presence of noise. Proceedings of the IEEE, 37(1):10–21, 1949.

[SHD+17] M. Sacher, F. Hauville, R. Duvigneau, O. Le Maître, N. Aubib, and M. Durand. Efficient optimization procedure in non-linear fluid-structure interaction problem: application to mainsail trimming in upwind conditions. Journal of Fluids and Structures, 69:209–231, 2017.

[Sil69] P. Silvester. Finite element solution of homogeneous waveguide problems. Alta Frequenza, 38:313 – 317, 1969.

[SMYC95] D. Sun, J. Manges, X. Yuan, and Z. Cendes. Spurious modes in finite element methods. IEEE Antennas and Propagation Magazine, 37:12 – 24, 1995.

[SQE61] W. Shockley, H. J. Queisser, and R. Ell. Detailed Balance Limit of Efficiency of pn Junction Solar Cells Additional information. Journal of Applied Physics, 32:510, 1961.

[SSD+14] G. A. Sotiriou, F. Starsich, A. Dasargyri, M. C. Wurnig, F. Krumeich, A. Boss, J.-C. Leroux, and S. E. Pratsinis. Photothermal killing of cancer cells by the controlled plasmonic coupling of silica-coated Au/Fe2O3 nanoaggregates. Advanced Functional Materials, 24(19):2818–2827, 2014.

[Ste87] M. Stein. Large sample properties of simulations using latin hypercube sampling. Technometrics, 29(2):143–151, 1987.

[STZS+03] C. M. Sotomayor Torres, S. Zankovych, J. Seekamp, A. P. Kam, C. C. Cedeño, T. Hoffmann, J. Ahopelto, F. Reuther, K. Pfeiffer, G. Bleidiessel, G. Gruetzner, M. V. Maximov, and B. Heidari. Nanoimprint lithography: an alternative nanofabrication approach. Materials Science and Engineering: C, 23(1):23 – 31, 2003. Current Trends in Nanotechnologies: From Materials to Systems. Proceedings of Symposium Q, E-MRS Spring Meeting 2002. June 18-21 2002.

[SWMW89] J. Sacks, W. Welch, J. T. Mitchell, and H. Wynn. Design and analysis of computer experiments. With comments and a rejoinder by the authors, volume 4. 1989.

[TEK+15] N. Tucher, J. Eisenlohr, P. Kiefel, O. Höhn, H. Hauser, M. Peters, C. Müller, J. C. Goldschmidt, and B. Bläsi. 3D optical simulation formalism OPTOS for textured silicon solar cells. Optics Express, 23(24):A1720–34, nov 2015.

[TH05] A. Taflove and S. Hagness. Computational Electrodynamics: The Finite-Difference Time- Domain Method. Artech House, Boston, third edition, 2005.

[TLF+17] C.-Y. Tsai, J.-D. Lai, S.-W. Feng, C.-J. Huang, Ch.-H. Chen, F.-W. Yang, H.-C. Wang, and L.-W. Tu. Growth and characterization of textured well-faceted zno on planar Si(100), planar Si(111), and textured Si(100) substrates for solar cell applications. Beilstein J Nanotechnol., 8:1939–1945, 2017.

[TSL+15] M. Topič, M. Sever, B. Lipovšek, A. Čampa, and J. Krč. Approaches and challenges in optical modelling and simulation of thin-film solar cells. Solar Energy Materials and Solar Cells, 135:57–66, 2015.

[VB09] J. G. Verwer and M. A. Botchev. Unconditionaly stable integration of maxwell's equations. Linear Algebra and Applications, 431:300–317, 2009.

[Ver10] J. G. Verwer. Component splitting for semi-discrete maxwell's equations. BIT Numerical Mathematics, 51:427–445, 2010.

[Viq16] J. Viquerat. Simulation of electromagnetic waves propagation in nano-optics with a high-order discontinuous galerkin time-domain method. 2016.

[Viq18] J. Viquerat. Fitting experimental dispersion data with a simulated annealing method for nano-optics applications. Journal of Nanophotonics, 12(3):036014, 2018.

[VL16] J. Viquerat and S. Lanteri. Simulation of near-field plasmonic interactions with a local approximation order discontinuous Galerkin time-domain method. Photonics and Nanostructures - Fundamentals and Applications, 18:43 – 58, 2016.

[VLDC11] A. Vial, T. Laroche, M. Dridi, and L. L. Cunff. A new model of dispersion for metals leading to a more accurate modeling of plasmonic structures using the fdtd method. Applied Physics A, 203:849–853, 2011.

[Wil80] J. H. Williamson. Low-storage runge-kutta scheme. Journal of Computational Physics, 35:48–56, 1980.

[WYL+12] K. X. Wang, Z. Yu, V. Liu, Y. Cui, and S. Fan. Absorption enhancement in ultrathin crystalline silicon solar cells with antireflection and light-trapping nanocone gratings. Nano Letters, 12:1616–1619, 2012.

[Yab82] E. Yablonovitch. Statistical ray optics. Journal of the Optical Society of America, 72(7):899–907, Jul 1982.

[Yee66] K. Yee. Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media. IEEE Transactions on Antennas and Propagation, 14:302 – 307, 1966.

[YYUA18] K. Yamamoto, K. Yoshikawa, H. Uzu, and D. Adachi. High-efficiency heterojunction crystalline si solar cells. Japanese Journal of Applied Physics, 57(8S3):08RB20, jul 2018.

[ZYH+06] L. Zeng, Y. Yi, C. Hong, J. Liu, N. Feng, X. Duan, L. C. Kimerling, and B. A. Alamariu. Efficiency enhancement in si solar cells by textured photonic crystal back reflector. Applied Physics Letters, 89(11):111111, 2006.

Acronyms

a-Si	Amorphous Silicon.
BEM	Boundary Element Method.
DG	Discontinuous Galerkin.
DGTD	Discontinuous Galerkin Time-domain.
FDTD	Finite Difference Time-domain.
FEFD	Finite Element Frequency-Domain.
HDG	Hybrid Discontinuous Galerkin.
HHO	Hybrid High-Order.
HMM	Heterogeneous Multiscale.
MHM	Multiscale Hybrid-mixed Method.
MIMD	Multiple Instructions on Multiple Data.
MOM	Method Of Moments.
PV	Photovoltaic.
RCWA	Rigorous Coupled-Wave Analysis.
SIMD	Single Instruction on Multiple Data.