HAL
archives-ouvertes.fr

Simulation of electromagnetic waves propagation in nano-optics with a high-order discontinuous Galerkin time-domain method

Jonathan Viquerat

▶ To cite this version:

Jonathan Viquerat. Simulation of electromagnetic waves propagation in nano-optics with a high-order discontinuous Galerkin time-domain method. Other. Université Nice Sophia Antipolis, 2015. English. NNT : 2015NICE4109 . tel-01272010v10

HAL Id: tel-01272010
https://hal.archives-ouvertes.fr/tel-01272010v10

Submitted on 29 Jan 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Université Nice Sophia Antipolis

Inria

Université Nice Sophia Antipolis - UFR Sciences

École Doctorale Sciences Fondamentales et Appliquées

Thèse présentée pour obtenir le titre de

Docteur en Sciences

de l'Université de Nice Sophia Antipolis

Spécialité Mathématiques Appliquées

présentée et soutenue par

Jonathan VIQUERAT

Simulation de la propagation d'ondes électromagnétiques en nano-optique par une méthode Galerkine discontinue d'ordre élevé

Simulation of electromagnetic waves propagation in nano-optics with a high-order discontinuous Galerkin time-domain method

Thèse dirigée par Stéphane LANTERI & Claire SCHEID

soutenue le 10 décembre 2015

Jury			
M. BUSCH, Kurt	Professeur	Institut für Physik, Berlin	Rapporteur
M. CIARLET, Patrick	Professeur	ENSTA ParisTech	Rapporteur
M. REMACLE, Jean-François	Professeur	Université Catholique de Louvain	Examinateur
M. POULIGUEN, Philippe	Responsable scientifique	Direction générale de l'armement	Examinateur
M. VIAL, Alexandre	Professeur	Institut Charles Delaunay, Troyes	Examinateur
M. MOREAU, Antoine	Maître de Conférences	Institut Pascal, Clermont-Ferrand	Invité
M. LANTERI, Stéphane	Directeur de recherche	INRIA Sophia Antipolis	Directeur de thèse
Mme SCHEID, Claire	Maître de Conférences	Laboratoire J. A. Dieudonné, Nice	Co-Directrice de thèse

"La vie ce n'est pas d'attendre que les orages passent, c'est d'apprendre à danser sous la pluie"

— Sénèque

Acknowledgements

Over the past three years, I had the good fortune to be surrounded by exceptional people, as much on a scientific level as on a personal level. These thanks are just a mere attempt to make it clear how indebted I am to all of them.

First, I would like to thank Stéphane Lanteri and Claire Scheid for their patience, their kindness and their trust over these three years. Stéphane and Claire, it has been a real pleasure to work under your guidance. I particularly enjoyed the considerable freedom you gave me, and I am delighted to spend two more years in your company.

My sincere thanks go to Kurt Busch and Patrick Ciarlet for accepting the role of referee, and for their careful reading of the present manuscript. I would like to take advantage of this opportunity to express how grateful I am to Patrick Ciarlet, for I may not have undertaken a PhD without his advice.

I deeply thank Jean-François Remacle, Philippe Pouliguen, Alexandre Vial and Antoine Moreau for the honor they made me of accepting to be part of the jury. They turned the defense into a moment of scientific exchange that I greatly appreciated. I would also like to acknowledge the Direction Générale de l'Armement for the partial funding of this thesis.

This manuscript would have lacked its physical touch without the precious help and numerous discussions we had with Antoine Moreau, Maciej Klemm, Wilfried Blanc, as well as our interlocutors at Cea Grenoble. I thank them here for their time and patience, and I am looking forward to pursue our collaborations. I also take the opportunity to thank Jens Niegemann for the countless questions he kindly answered during this PhD.

I deeply thank all the members of the Nachos project-team in which I had the pleasure to spend the past three years. Thank you to Ludovic Moya, Clément Durochat and Fabien Peyrusse for their warm welcome in the team when I arrived. I would like to thank the senior researchers of the team, and more particularly Nathalie Glinsky for her kindness and all the nice discussions, and Stéphane Descombes for his experience about time-integration techniques, and for proof-reading section 3.2 of this manuscript. I owe a lot to Raphaël Léger for the countless hours of help on my projects, his friendship and his sharp sense of humour. Thank you to Nikolai Schmitt for all the scientific discussions, for feeding me from time to time, and for the nice climbing sessions; to Alexandra Christophe for her exceptional mood and her kindness; to Colin Vo Cong Tri for all the good laughs and for his driving skills. The list of people I would like to thank is still long: Tristan Cabel, Montserrat Argente, Frédéric Valentin, Marie Bonnasse, Paul Loriot, Julien Coulet, Thomas Frachon, Hugo Flachat, Jules Fauque, Rebecca Contal, Geoffroy Fouillade, Quentin Brissaud, Samira Amraoui. To all of you, thank you for the unique atmosphere you brought to the team.

A special thank goes to Emmanuel Cieren, for without him I would certainly be working on fluid dynamics, and to Julien Servais, for the great years the three of us spent together. More broadly speaking, thank you to all the Ensta 2012. I would also like to thank Émilien Kofman for being such a great friend, and for tolerating my poor climbing level on a weekly basis.

I could never thank my family enough for their continuous support, their love, and for keeping me from my natural tendency to feel overwhelmed by unexpected things. A whole manuscript would not be enough to tell you how lucky I feel to have you. My thoughts also go to Damien and my grandparents.

My deepest thanks go to Camille for coping with me during the ups and downs of this PhD. Your love and your good mood are the best remedies to my grumpiness. My greatest certainty is that having you beside me is a treasure.

Contents

1 Introduction **1**
- 1.1 A bit of History . . . 1
- 1.2 Nano-optics . . . 1
- 1.3 Computational electromagnetics in time-domain . . . 2
- 1.4 The Discontinuous Galerkin Time-Domain method . . . 4
- 1.5 Outline . . . 5

2 Classical electromagnetics **7**
- 2.1 Maxwell's equations . . . 7
  - 2.1.1 Constitutive relations . . . 8
  - 2.1.2 Adimensionning . . . 9
  - 2.1.3 Material interfaces . . . 10
  - 2.1.4 Some analytical solutions to Maxwell's equations . . . 12
- 2.2 Dispersive models . . . 17
  - 2.2.1 Underlying physics . . . 17
  - 2.2.2 Drude and Drude-Lorentz models . . . 18
  - 2.2.3 Generalized model . . . 20
  - 2.2.4 Maxwell's equations in dispersive materials . . . 22
  - 2.2.5 An illustration: a metallic sphere . . . 24
  - 2.2.6 A digression on non-local models . . . 26
  - 2.2.7 Causality principle . . . 28

3 The DGTD method **31**
- 3.1 DG method for Maxwell equations . . . 31
  - 3.1.1 Weak formulation . . . 31
  - 3.1.2 Space discretization . . . 32
  - 3.1.3 Numerical fluxes . . . 33
  - 3.1.4 DG matrices . . . 34
  - 3.1.5 Mapping from a reference element . . . 36
  - 3.1.6 Polynomial expansion basis . . . 38
  - 3.1.7 Boundary conditions . . . 39
  - 3.1.8 Derivation of the flux formulation . . . 41
  - 3.1.9 Spectrum of the DG operator . . . 45
- 3.2 Time discretization . . . 47
  - 3.2.1 A quick overview of time integration schemes . . . 48

3.2.2 Leap-Frog schemes . . . 48
3.2.3 Runge-Kutta schemes . . . 53
3.2.4 Low-storage schemes . . . 57
3.2.5 Timestep choice and the CFL condition . . . 60
3.3 Validation and numerical experiments . . . 60
3.3.1 PEC cubic cavity mode . . . 60
3.3.2 Convergence with centered and upwind fluxes . . . 60
3.3.3 Flux weightings . . . 62
3.3.4 Fluxes for anisotropic materials . . . 62
3.4 The ADE method for dispersive materials . . . 64
3.4.1 ADE formulation in the DGTD framework . . . 64
3.4.2 Validation . . . 64
3.4.3 On the necessity of a good description of dispersive materials . . . 64
3.5 Theoretical results . . . 67
3.5.1 Stability . . . 67
3.5.2 Convergence . . . 68

4 Technicalities **71**
4.1 Domain truncation . . . 71
4.1.1 Absorbing boundary conditions . . . 71
4.1.2 Perfectly matched layers . . . 73
4.1.3 Properties of PMLs . . . 75
4.1.4 CFS-PML for Maxwell's equations . . . 76
4.1.5 Performance assessment in the DGTD framework . . . 78
4.2 Sources and TF/SF formulation . . . 79
4.2.1 Sources . . . 79
4.2.2 TF/SF formulation . . . 85
4.3 Fourier transform . . . 86
4.4 Relevant quantities in electromagnetic scattering . . . 88
4.4.1 Cross-sections . . . 88
4.4.2 Volumetric absorption . . . 89
4.4.3 Reflection and transmission . . . 89
4.4.4 Far field and radar cross-section . . . 90

5 Curvilinear elements **93**
5.1 Formulation for curved elements . . . 94
5.1.1 Intrinsic limitation of linear elements . . . 94
5.1.2 High-order mapping . . . 96
5.1.3 Numerical integration . . . 102
5.1.4 CFL condition . . . 104
5.2 Validation on a spherical PEC cavity . . . 104
5.2.1 Results . . . 106
5.2.2 Visualization issues . . . 107
5.3 Plasmonic resonances of isolated and coupled gold nanospheres . . . 107
5.3.1 Isolated nanosphere . . . 107
5.3.2 Coupled nanospheres . . . 110
5.4 Realistically-rounded metallic nanocubes . . . 113

6 Locally adaptive DGTD method **115**
6.1 DG formulation . . . 116
6.2 h-convergence . . . 118
6.3 Order distribution strategy . . . 118
6.4 Performance assessment . . . 121
6.4.1 Sequential speedup . . . 121
6.4.2 Parallel load balance . . . 123
6.5 Plasmonic nanolens . . . 125
6.6 Bowtie nanoantenna . . . 127
6.7 Conclusion . . . 132

7 Improving performances **133**
7.1 Reverse Cuthill-McKee renumbering . . . 133
7.2 Performances of a non-blocking MPI implementation . . . 134
7.2.1 Mesh partitioning . . . 135
7.2.2 Strong scaling . . . 135
7.2.3 Parallel balance . . . 137
7.2.4 Conclusion . . . 139

8 Realistic nano-optics computations **141**
8.1 Electron energy loss spectroscopy . . . 141
8.1.1 Introduction . . . 141
8.1.2 EELS spectrum of an aluminium nanosphere . . . 142
8.2 Gap-plasmon confinement with gold nanocubes . . . 145
8.2.1 Physical parameters and quantities . . . 145
8.2.2 Influence of the rounding . . . 146
8.2.3 Absorption and scattering regimes . . . 146
8.2.4 Numerical discussion . . . 148
8.3 Dielectric reflectarrays . . . 152
8.3.1 Physical parameters and quantities . . . 152
8.3.2 Influence of lithography defects . . . 153
8.3.3 1D dielectric reflectarray . . . 153
8.3.4 2D dielectric reflectarray . . . 154
8.3.5 Numerical considerations . . . 154

9 Outlook **161**
9.1 Summary . . . 161
9.2 Future works . . . 162
9.2.1 Physics and material models . . . 162
9.2.2 Numerical improvements . . . 163
9.2.3 High-performance computing . . . 163

A Dispersion parameters **165**

B Non-conforming $\mathbb{P}_p - \mathbb{P}_m$ matrices **169**

C Another gold permittivity function **173**

List of Figures

1.1 Photonic crystal structures 2
1.2 Staggered unknowns discretization in a Yee cell 3
1.3 Comparison between finite elements, finite volumes and discontinuous Galerkin 5

2.1 Integration domains for jump relations 11
2.2 Spatial representation of a linearly polarized plane wave 13
2.3 Dielectric slab illuminated with a plane wave 14
2.4 Silver permittivity predicted by the Drude model compared to experimental data 19
2.5 Silver permittivity predicted by the Drude-Lorentz model compared to experimental data 20
2.6 Gold permittivity predicted by the 4SOGP model compared to experimental data 22
2.7 Nickel permittivity predicted by the Drude and the 1SOGP model compared to experimental data 23
2.8 Plasmon oscillation in a metallic sphere due to an exterior electric field 25
2.9 Plasmonic resonance of a gold nanosphere 25
2.10 Configurations of relevance for the non-local model 27
2.11 Non-local resonance of a gold nanosphere 29

3.1 Linear mapping from the reference element $\widehat{T}$ to the physical element T_i 37
3.2 Lagrange nodes on triangles and tetrahedra 39
3.3 Second order Lagrange polynomials on triangles 40
3.4 Ghost cells layer on the computational domain boundary 41
3.5 Structure of the Riemann problem at cells interfaces 44
3.6 Eigenvalues of the discrete DG operator with PBCs for various upwinding factors 46
3.7 Stability regions, phase and amplitude errors induced by Leap-Frog schemes 52
3.8 Steps of the RK2 algorithm 56
3.9 Steps of the RK4 algorithm 57
3.10 Stability contour and phase error induced by Runge-Kutta schemes 58
3.11 Steps of the LSRK2 algorithm 59
3.12 Stability contours of two low-storage Runge Kutta schemes 59
3.13 Convergence rate for DG $\mathbb{P}_3$ method with LSRK4 algorithm, for various values of the unwinding factor α 62
3.14 Impact of flux weighting on accuracy for increasing jumps of ε 63
3.15 Validation of the generalized dispersive model implementation 65
3.16 **E** near-field solution and scattering cross-section of a silica/gold nanoshell device 66

4.1 Reflection coefficients for first to fourth-order absorbing boundary conditions 73
4.2 General functioning of the PML 74
4.3 Reflection coefficient of the CFS-PML for traveling waves 76
4.4 PML configuration for a cubic domain 78
4.5 Performance of the CFS-PML 80
4.6 Performance of the Silver-Muller condition compared to the CFS-PML 81
4.7 Wideband pulse representation in time-domain and frequency-domain 82
4.8 Composition of a step-index optical fiber 83
4.9 Solutions of the mode equation for step-index optical fibers 84
4.10 Examples of electric field map in fiber modes 85
4.11 Mesh configuration including a scatterer, a TF/SF interface, and a PML layer 86
4.12 TF/SF interface with modified fluxes 87
4.13 Periodic array of scatterers on a metallic slab 90
4.14 Spherical coordinates system for the RCS computation 91

5.1 Polygonal approximation of a domain with curved boundaries 94
5.2 Geometrical distance between a polygonal and a polynomial boundary 95
5.3 Second order mapping from the reference element $\hat{T}$ to the physical element T_i 96
5.4 Proper and improper (n, p) combinations for a quadratic physical surface 99
5.5 Curvilinear elements selection on a nanocube mesh with rounded corners 100
5.6 Number of integrations points required by Gauss-Legendre and [ZCL09] rules on triangle and tetrahedron 103
5.7 Coarse mesh of a PEC cavity 105
5.8 $\mathbb{P}_2$ numerical solution of the spherical cavity mode on a coarse mesh 107
5.9 Total time required for the numerical integration of the FE matrices 108
5.10 Projection of a $\mathbb{P}_2$ curvilinear solution as a $\mathbb{P}_1$ solution on a refined mesh 108
5.11 Coarse mesh of a nanosphere for a cross-section calculation 109
5.12 Scattering cross-section of a metallic sphere 111
5.13 Near-field visualization of the electric field around a gold nanosphere dimer 112
5.14 Absorption cross-section of a gold nanosphere dimer 112
5.15 Nanocube mesh with rounded corners 113
5.16 Absorption cross-section of a rounded silver nanocube on a gold film 114

6.1 Order distribution for the h-convergence validation of LA-DGTD method 119
6.2 Authorized interfaces in the LA-DGTD implementation 120
6.3 Meshes with local refinements for the cubic cavity mode 121
6.4 Compared timestep distribution in locally refined meshes 123
6.5 Weighted vs non-weighted parallel load balance for the LA-DGTD method 124
6.6 Nanolens composed of three metallic spheres 125
6.7 Mesh setup for a metallic nanolens 126
6.8 Polynomial order repartition for the nanolens mesh 126
6.9 Field enhancement in the vicinity of the smallest sphere of a self-similar nanolens 127
6.10 E_y field map in a nanolens device 128
6.11 Bowtie nanoantenna with rounded edges 129
6.12 Mesh setup for a bowtie nanoantenna 129
6.13 Polynomial order repartition for the bowtie mesh 130
6.14 Extinction cross-section of a bowtie nanoantenna 130

6.15 $|\mathbf{E}|$ field map in a bowtie device 131

7.1 Nanosphere mesh for RCM study 134
7.2 Impact of the RCM renumbering on the connectivity matrix 134
7.3 Mesh to graph conversion 135
7.4 Sub-domains of a Metis-partitioned mesh 136
7.5 Statistics of Metis-partitioned meshes 137
7.6 Variable data exchange procedures between cores 138
7.7 MPI speedup and efficiency for the PEC cavity case 138
7.8 MPI time over CPU time ratio 139
7.9 CPU time balance over sub-domains 140

8.1 Mesh setup for an aluminium nanosphere EELS spectrum computation 143
8.2 EELS spectrum of a single aluminium nanosphere 144
8.3 E_z field map during an EELS experiment 144
8.4 Random arrangement of chemically-produced nanocubes 145
8.5 Realistic metallic nanocube on a dielectric-coated metallic slab 146
8.6 Meshes of rounded nanocubes 147
8.7 Absorption cross-section of a nanocube device for various edge roundings 148
8.8 Resonance frequencies and absorption efficiencies of gold nanocubes 149
8.9 $\widehat{\mathbf{H}}$ field modulus for different nanocube configurations 150
8.10 Unit cell of a realistic monodimensional dielectric reflectarray 152
8.11 6-element dielectric reflectarray 153
8.12 Reflection coefficient of a single dielectric resonator with lithography defect 154
8.13 Ideal and realistic 1D dielectric reflectarray meshes 155
8.14 Radar cross-section of ideal and realistic 1D dielectric reflectarrays 155
8.15 Time-domain snapshot for ideal and realistic 1D dielectric reflectarrays 156
8.16 Ideal 2D dielectric reflectarray mesh 157
8.17 Radar cross-section of an ideal 2D dielectric reflectarray 157
8.18 Time-domain snapshot for an ideal 2D dielectric reflectarray 158

A.1 Silver and gold permittivities predicted by the 4-SOGP model compared to experimental data 166

C.1 Gold permittivity predicted by a 3-SOGP model for gold nanocubes 174

List of Tables

2.1 Units and numerical values of electromagnetic constants 9
2.2 Units of the original and the normalized Maxwell systems 10
2.3 Quality of the fit obtained by various dispersion models for silver and gold 21
2.4 Quality of the fit obtained by Drude and 1SOGP models for nickel 22

3.1 Minimal number of RK stages to obtain p^{th}-order convergence 56
3.2 Meshes characteristics for a cubic PEC cavity 61
3.3 Error levels and convergence rates of the cubic cavity case for different approximation orders, fluxes and time schemes with meshes of increasing refinement 61
3.4 Convergence rates for the anisotropic PEC cavity with meshes of increasing refinement . 63

5.1 Number of integration points required for exact integration on the triangle 103
5.2 Number of integration points required for exact integration on the tetrahedron 103
5.3 Meshes characteristics for the spherical PEC cavity case 104
5.4 Convergence rates of the spherical cavity case for different approximation orders and fluxes, with rectilinear and curvilinear meshes of increasing refinement 106
5.5 Mesh characteristics for the metallic sphere cross-section computation case 110

6.1 Error levels and convergence rates of the cubic cavity case for mixed orders of approximation on meshes of increasing refinement 119
6.2 Characteristics of the locally refined cubic cavity meshes 121
6.3 CPU times, memory consumption and error levels for mixed orders of approximation on locally refined meshes 122

7.1 Sequential speedup with the RCM algorithm 135

A.1 Coefficients of various dispersive models for gold 167
A.2 Coefficients of various dispersive models for silver 168

C.1 Coefficients of 3-SOGP model for gold nanocubes 174

1

Introduction

1.1 A bit of History

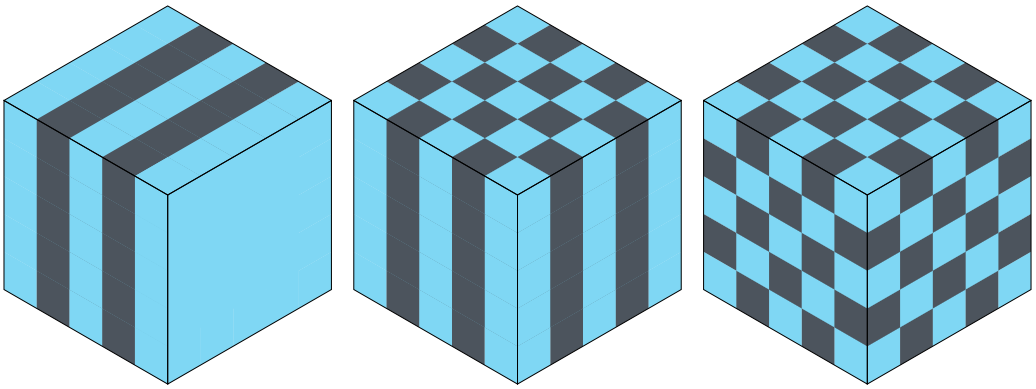
The first observations of electric and magnetic phenomena by man date back to 600 B.C., when Thales of Miletus observed the property of amber to attract light objects, such as fabric, after being rubbed with fur. In the same period, he also reported the existing attraction between lodestone and iron. Three centuries later, Euclid threw together the basis of geometrical optics in *Optica*, describing the laws of reflection and postulating that light travels in straight lines. From this point, the studies of electromagnetism and light followed parallel paths, until the XIX$^{\text{th}}$ century. In 1848 and 1850, Hippolyte Fizeau and Léon Foucault measured the speed of light respectively at 3.14×10^8 and 2.98×10^8 m.s^{-1}. In 1855, Wilhelm Eduard Weber and Rudolf Kohlrausch found out through an experimentation that the ratio of the electromagnetic to the electrostatic unit charge was close to 3.107×10^8 m.s^{-1}. Although the values from Fizeau and Foucault were known at that time, they did not notice the alikeness of the results [Kei98].

It is only in 1861 that James Clerk Maxwell, looking at Weber and Kohlrausch's results, established the existing link between light propagation and electromagnetic phenomena. In [Max65], he concludes : "The agreement of the results seems to show that light and magnetism are affections of the same substance, and that light is an electromagnetic disturbance propagated through the field according to electromagnetic laws". At that stage, Maxwell's theory of electromagnetism is regrouped in a set of twenty unknowns and equations, that will then be converted into modern notations by a concurrent work of Olivier Heaviside, Josiah Willard Gibbs and Heinrich Hertz in 1884. It should be noted that the 1861 formulation of Maxwell still relies on the existence of the luminiferous aether, a postulated medium necessary to the propagation of light. For more than forty years, the latter will be a source of conflict, his properties being very difficult to accept in the physical paradigm of that time. In 1905, Einstein's special theory of relativity finally provided a framework that did not require the presence of aether anymore.

1.2 Nano-optics

Maxwell's equations in their modern form have been studied for many decades, resulting in an extremely wide range of applications. Many of those are now part of our everyday life, such as wireless communications of all forms, optical fibers, medical imaging, ... In order to control electromagnetic wave propagation,

most of these devices rely on tailored geometries and materials. During the last decades, the evolution of lithography techniques allowed the creation of geometrical structures at the nanometer scale, thus unveiling a variety of new phenomena arising from light-matter interactions at such levels. These effects usually occur when the device is of comparable size or (much) smaller than the wavelength of the incident field. Periodic mono- or multi-dimensional arrangements of sub-wavelength dielectric patterns, known as photonic crystals (see figure 1.1), give rise to allowed and forbidden wavelengths regions in certain directions [JJ07]. These so-called band gaps can be tuned by slight modifications of the periodicity, allowing physicists to create a full range of light-control devices from photonic crystals. Periodic arrays of dielectric resonators can also be used to achieve non-cartesian reflection of plane waves, which is a highly promising step toward on-chip wireless optical communications [ZWS+13].



(a) 1D cristal (b) 2D cristal (c) 3D cristal

Figure 1.1 | Photonic crystal structures in one, two and three dimensions. The blue and gray areas represent the alternance of high and low permittivity materials.

Metallic nanostructures can also demonstrate stunning effects when excited in the optical regime. The key feature of these effects is the coupling of the electromagnetic field to the electron gas of the metal, resulting in an oscillation phenomenon called plasmon. One usually differentiates the bulk plasmons, that take place in the volume, from the surface plasmons (SP), that arise at the interface between the metal and a dielectric. SPs can be propagative along a metal/dielectric interface, or non-propagative, in which case they are called localized surface plasmons (LSPs). The proper excitation of LSPs can lead to very intense resonances (meaning that the field is enhanced). Thanks to metallic tips exploiting this strong localization, optical microscopy beyond the diffraction limit [NH07] is possible. The high sensitivity of resonant metallic nanostructures also allows to create very accurate biosensors [CLS+11]. In the medical field, attempts have been made to develop cancer therapies based on the localized heating produced with resonating nano particles [SSD+14]. As for dielectrics, periodic arrays of metallic patterns can lead to new devices with non-natural behaviors at larger scales. These structures are usually gathered under the root word metamaterials, which then designates an effective medium composed of an arrangement of nanostructures, and displaying uncommon properties. Negative refractive index materials [DWSL07] or optical cloaking [CCKS07] are some of the most common examples.

1.3 Computational electromagnetics in time-domain

The large variety of phenomena displayed by nano-optic systems, their dependance upon a large number of parameters (geometry, materials, sources, ...), as well as the complexity of most fabrication processes

prevent physicists from relying on experiments only. However, apart from very specific cases involving simple geometries, and for which electromagnetic fields can be expressed as closed-forms, solutions to Maxwell's equations are out of reach of hand calculations. Hence, numerical simulation seems to be the appropriate complementary tool to physical experiments, and can be exploited in various ways. Indeed, it can be used to rapidly scan a large number of configurations, in order to identify the most efficient set of parameters. This scanning can be done "blindly" by hand if a small number of parameters is involved, or by combining a direct numerical method to an iterative optimization algorithm when the dimension of the parameters space becomes large [Pav13]. Numerical tools also allow a deeper understanding of the physical phenomena observed in real devices, since they allow the experimentalist to obtain information about any quantity out of the simulation, which is not possible in most physical experiments. Additionally, various physical models can be easily assessed and their effects compared, in order to verify their applicability in given configurations. Various techniques are available to solve nano-optics problems: some are specialized algorithms, that were developed for the fast-solving of specific configurations at low computational cost (for example the Discrete Dipole Approximation (DDA) [DF94] or the Rigorous Coupled-Wave Analysis (RCWA) [MG81]). However, these can hardly or not at all handle other applications. On the other hand, more general methods exist that are well suited to solve a very large set of problems. In the remaining of this section, we focus on the major time-domain techniques.

The Finite-Difference Time-Domain (FDTD) method is certainly the most spread of all. As early as 1928, Courant, Friedrichs and Lewy published an article presenting a finite-difference scheme for the second order wave equation in 1D and 2D, as well as the well-known CFL stability condition involved for explicit time-domain schemes [RFH28]. In 1966, Yee introduced a staggered grid in space (see figure 1.2) to solve the curl formulation of Maxwell's equations [Yee66]. The method relies on a combination of Taylor expansions to express the spatial derivatives, and on a centered Leap-Frog (LF) scheme in time. As of today, FD represent a particularly simple method to solve electromagnetics problems, combining simple implementation and high computational efficiency. They were applied successfully to numerous nano-optics configurations [SCG10].

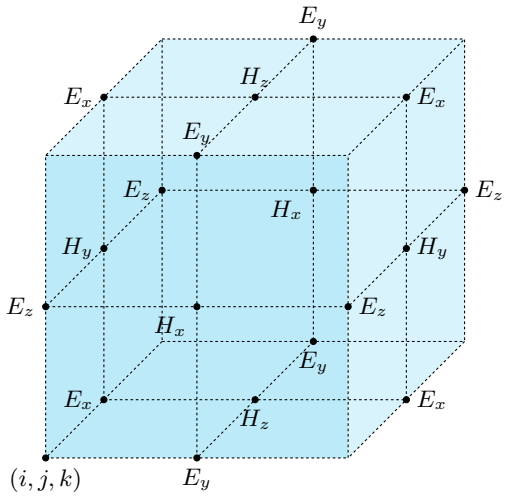


Figure 1.2 | Staggered unknowns discretization in a Yee cell. The **H** field components are on the center of the faces, while the **E** ones are on the center of the edges.

However, FD algorithms suffer from serious drawbacks. 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, which is an important source of inaccuracy [DDH01]. To over-

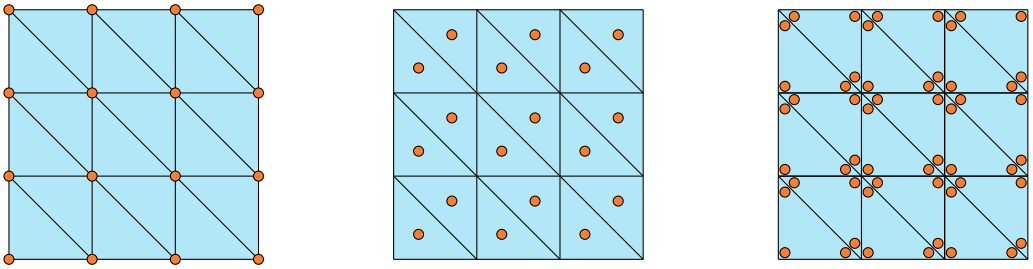
come this pitfall, the user can either use an extreme refinement of the grid, which leads to a serious rise in computational cost, or exploit one of the numerous possible modifications of the FD method that have been proposed for tackling the staircasing effect [HR98]. However, all the latter available modifications represent a tradeoff between the simplicity of the classical algorithm and the accuracy of the boundary description. The second main source of inaccuracy in the FDTD method arises in the case of heterogeneous problems. In this case, the Taylor approximation used is no longer valid, since the electromagnetic fields are not smooth across the interface. The consequence is that higher-order FD schemes in space are usually reduced to second-order. Advanced FDTD methods were developed to tackle this problem [TH05], at the price of an increased complexity of the algorithm. Moreover, there is no theoretical convergence proof for FDTD algorithms outside the uniform grid case.

Finite Elements (FE) were introduced in 1969 by Silvester to solve waveguide problems [Sil69]. This method does not rely on a grid, but on a tessellation of the geometry of the problem. Starting from the continuous equations, a discrete variational form is obtained by approximating the unknowns in a finite dimensional space. Then, its discretization leads to a sparse matrix-vector system that has to be solved at each timestep. In the specific case of electromagnetism, the use of nodal basis functions, *e.g.* such as their value is unity at a given vertex and zero on every other, is subject to caution. Indeed, it was proved that they can lead to spurious oscillations, due to an ill representation of the curl kernel [SMYC95]. To overcome this issue, Nédélec introduced a new family of vector finite elements in 1986 [Né80], named Nédélec finite elements, or edge finite elements. These elements display several interesting properties: (i) their divergence is zero, and (ii) each basis function associated to an edge has a constant tangential component on the latter, and a zero tangential component on the others. Hence, the tangential continuity of the electric field across the edge is naturally enforced.

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, or (iii) a combination of both. However, these improvements lead to larger linear systems to solve at each timestep, which can make the FE method impractical in time-domain simulations for very large systems. For this reason, in nano-optics, FE methods are more often used in frequency-domain. However, a few references can be found exploiting time-domain FE for nanophotonics applications [HLY13].

1.4 The Discontinuous Galerkin Time-Domain method

Discontinuous Galerkin (DG) methods were originally introduced in 1973 by Reed and Hill [RH73], and have been widely used since in the computational fluid dynamics field. However, their application to the time-domain Maxwell equations is more recent [RF98]. DG methods can be seen as classical finite element methods for which the global continuity of the approximation is lifted. In the same fashion as FE methods, the unknowns are approximated on a finite set of basis functions. However, for DG, the support of basis functions are restrained to a single discretization cell. Hence, the solution produced by a DG method is discontinuous (similarly to finite volumes), and multiple different field values are stored for each element/element interface degree of freedom (see figure 1.3). The three main consequences are that (i) DG methods naturally handle material and field discontinuities, (ii) the weak formulation is local to an element, implying no large mass matrix inversion in the solving process, and (iii) the order of polynomial approximation in space can be made arbitrarily high by adding more degrees of freedom inside the elements. However, this also means that DG methods have higher memory requirements than standard FE methods. Afterward, connexion between the cells is restored by the use of a numerical flux, in the fashion of finite volume methods. The choice of the numerical flux has a great influence on the mathematical properties of the DG discretization, as energy preservation, for example.



(a) Finite elements : continuous, non-constant-per-cell solution

(b) Finite volumes : discontinuous, constant-per-cell solution

(c) Discontinuous Galerkin : discontinuous, non-constant-per-cell solution

Figure 1.3 | Concept comparison between FE, FV and DG. The triangles represent the cells of the mesh, while the orange dots represent the degrees of freedom. For FE, the whole problem is considered at once, and the obtained numerical solution is continuous across cell interfaces. For FV, a local problem is considered in each cell, leading to a discontinuous, constant-per-cell solution. For DG, the method is analog to FV, but the solution is not restrained to a constant per cell. In this case, a first-order polynomial approximation is used for the DG discretization.

The discontinuity of the approximation makes room for numerous methodological improvements, such as efficient parallelization ([Die12], [BFLP06]) or the use of non-conforming [FL10] and hybrid meshes [LVD+14]. Recent studies in the DG framework include local timestepping [Pip05] as well as locally implicit formulations [Moy12]. Also, a wide choice of time-integration schemes can be used for the discretization of time derivatives, including Leap-Frog (LF) and Runge-Kutta (RK).

The DGTD method for solving the time domain Maxwell equations is increasingly adopted by several physics communities. Concerning nanophotonics, unstructured mesh based DGTD methods have been developed and have demonstrated their potentialities for being considered as viable alternatives to the FDTD method. The most remarkable achievements in the nanophotonics domain since 2009 are due to Busch et al. Busch [NKSB09]-[SKNB09]-[BKN11] has been at the origin of seminal works on the development and application of the DGTD method in this domain. These works not only deal with the extension of the DGTD method with regards to the complex material models and source settings required by applications relevant to nanophotonics and plasmonics [KBN10]-[MNHB11]-[WROB13], but also to core contributions aiming at improving the accuracy and the efficiency of the proposed DGTD solvers [NKP+10]-[NDB12]-[DNBH15].

1.5 Outline

The remaining of this manuscript is structured in the following way:

- Chapter 2 presents the usual concepts of electromagnetics, as well as some standard textbook problems and their analytical solutions. An extensive presentation and analysis of dispersive models for metals follows, along with a comparison of our custom generalized dispersive model with other classical dispersion models.
- The first section of chapter 3 runs, step by step, through the spatial discretization of Maxwell's equations by the discontinuous Galerkin method. Then, two classical time integration methods are proposed and briefly studied to complete the discretization. The algorithm is then validated for classical and dispersive materials. Finally, a few theoretical results are given on the method.

- Chapter 4 regroups practical techniques that are pre-requisites for the resolution of realistic problems, such as perfectly-matched layers, sources, total-field scattered-field technique, as well as physical post-treatments.
- In chapter 5, the DG method is extended to the use of quadratic tetrahedra, which allow both a better geometrical description of the problems, and lifts the numerical accuracy limit from 2^{nd} to 4^{th} order in the case of curved geometries. Several nano-optics relevant test-cases are considered that confort the interest of this development.
- Chapter 6 is dedicated to a locally-adaptive DG formulation, where polynomial interpolation order can be defined independently in each cell of the mesh. An efficient repartition algorithm is supplied, which provides interesting speedups over homogeneous polynomial repartition in several realistic test-cases.
- The sequential and parallel performances of our Fortran discontinuous Galerkin time-domain (DGTD) implementation are assessed in chapter 7. First, a renumbering algorithm is proposed that enhances the sequential performances by reducing adressing time. Then, the speedup and parallel balance of the MPI implementation are tested on a standard cavity case.
- The last chapter is dedicated to realistic nanophotonics computations processed with our DGTD code: (i) the electron energy loss spectrum (EELS) of an aluminium nanosphere, (ii) the gap-plasmon resonances obtained under chemically-produced nanocubes with realistic shapes, and (iii) 1D and 2D dielectric reflectarrays, with study of the lithography defects on their performances.

2

Classical electromagnetics

Before focusing on nanophotonics, it seems necessary to recall the classical principles of electromagnetics. First, Maxwell's equations are presented in vacuum and dielectric media (section 2.1), and a few exact solutions are exhibited. To remain concise, the covered concepts are restricted to the minimum necessary for the present study (however, a very complete presentation of classical electrodynamics can be found in [Jac98] or [RC01]). Then, the modeling of dispersive media (such as metals in the visible spectrum) is introduced (section 2.2). A generalized model is presented, and its accuracy is compared to standard ones. An extension to non-local models is also briefly outlined.

2.1 Maxwell's equations

In a somehow tautological way, the electric charge is usually defined as the fundamental property of matter that causes it to undergo the electromagnetic interaction. More precisely, a particle of charge q and speed $\mathbf{v}$ is subject to the Lorentz force:

$$\mathbf{F} = q\left(\mathbf{E} + \mathbf{v} \times \mathbf{B}\right), \tag{2.1}$$

where $\mathbf{E}$ and $\mathbf{B}$ are respectively the electric field and the magnetic induction vectors in $\mathbb{R}^3$. In most physics textbooks, $\mathbf{E}$ and $\mathbf{B}$ are considered to be the "fundamental fields". However, it is customary to introduce additional fields, namely the electric displacement $\mathbf{D}$ and the magnetic field $\mathbf{H}$. One shall see in the next section how these are related to $\mathbf{E}$ and $\mathbf{B}$. For a given medium, we also introduce the density of free electric charges ρ, and the free electric current density $\mathbf{J}$. All these quantities depend on position $\mathbf{x} = {}^t(x, y, z)$ and time t. One can now write Maxwell's equations in their modern version, in SI units:

Maxwell's equations

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

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

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

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

along with the continuity equation[1]:

Continuity equation

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \mathbf{J} = 0. \tag{2.6}$$

The two curl equations are often called "fundamental" equations, while the two divergence ones are referred to as "auxiliary" equations. Indeed, one can see that (2.4) and (2.5) are not evolutionary, in the sense that they do not contain any time derivative, but only bring constraints on the solutions of (2.2) and (2.3). Taking the divergence of (2.2) and (2.3), and combining with (2.6), one obtains:

$$\begin{aligned} \frac{\partial}{\partial t}\left(\nabla \cdot \mathbf{D} - \rho\right) &= 0, \\ \frac{\partial}{\partial t}\left(\nabla \cdot \mathbf{B}\right) &= 0. \end{aligned} \tag{2.7}$$

Hence, if the divergence conditions are verified for the initial state, they should also be verified for any future state. One shall therefore drop the divergence conditions in the remaining of this thesis by considering that they are verified for all the considered initial states. Additional considerations on this topic can be found in [RC01].

By examining system (2.2 – 2.3), one may notice that it contains 12 scalar unknowns for only 6 scalar equations. Hence, the system is not closed, and therefore not fit for solving. This is the purpose of next section.

2.1.1 Constitutive relations

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

$$\begin{aligned} \mathbf{D} &= \bar{\bar{\varepsilon}}\,\mathbf{E}, \\ \mathbf{B} &= \bar{\bar{\mu}}\,\mathbf{H}, \end{aligned} \tag{2.8}$$

where $\bar{\bar{\varepsilon}}$ and $\bar{\bar{\mu}}$ are tensors depending on $\mathbf{x}$, t, $\mathbf{E}$ and $\mathbf{B}$. To simplify this presentation, a few assumptions are made, at least temporarily:

[1] At this point, it is important to notice that although (2.6) can be derived from (2.3) and (2.4), it can also be derived independently from physical considerations (see [RC01] for more details).

Table 2.1 | Units and numerical values of electromagnetic constants.

	ε_0	μ_0	Z_0	c_0	q
Unit	$\mathrm{F.m^{-1}}$	$\mathrm{H.m^{-1}}$	Ω	$\mathrm{m.s^{-1}}$	C
Type of value	Approx.	Exact	Approx.	Exact	Approx.
Value	8.854×10^{-12}	$4\pi \times 10^{-17}$	$119.9\times\pi$	$2.997\,924\,58 \times 10^{8}$	1.602×10^{-19}

- ◇ The considered materials are linear, thus $\bar{\bar{\varepsilon}}$ and $\bar{\bar{\mu}}$ are independent of $\mathbf{E}$ and $\mathbf{B}$;
- ◇ Materials are isotropic, which means $\bar{\bar{\varepsilon}} \equiv \varepsilon \mathbb{I}_3$ and $\bar{\bar{\mu}} \equiv \mu \mathbb{I}_3$;
- ◇ Materials are homogeneous, *i.e.* ε and μ are constant within a given material;
- ◇ Although dispersive materials will be a central point in this work, it is assumed temporarily that ε and μ are independent of time.

Hence, in such a material with constant permittivity ε and permeability μ, (2.8) becomes:

$$\begin{aligned} \mathbf{D} &= \varepsilon\, \mathbf{E}, \\ \mathbf{B} &= \mu \mathbf{H}. \end{aligned}$$

It is customary to introduce ε_0 and μ_0 the vacuum permittivity and permeability, as well as ε_r and μ_r the relative permittivity and permeability of the considered material. Obviously, in vacuum, $\varepsilon_r = 1$ and $\mu_r = 1$. Hence, the constitutive relations are written as follows:

$$\begin{aligned} \mathbf{D} &= \varepsilon_0\, \varepsilon_r\, \mathbf{E}, \\ \mathbf{B} &= \mu_0 \mu_r \mathbf{H}. \end{aligned} \tag{2.9}$$

It is then straightforward to obtain Maxwell's equations for linear, homogeneous, isotropic, non-dispersive materials:

$$\begin{aligned} \nabla \times \mathbf{E} &= -\mu_0 \mu_r \frac{\partial \mathbf{H}}{\partial t}, \\ \nabla \times \mathbf{H} &= \quad \varepsilon_0\, \varepsilon_r \frac{\partial \mathbf{E}}{\partial t} + \mathbf{J}. \end{aligned} \tag{2.10}$$

System (2.10), completed with adequate boundary and initial conditions, is now fit to solving. However, it is preferable to eliminate ε_0 and μ_0 from the equations for the numerical treatment. This is the purpose of next section.

2.1.2 Adimensionning

New variables are introduced to normalize system (2.10). For a physical variable X, the new variable is noted $\widetilde{X}$. The adequate substitutions are:

$$\widetilde{\mathbf{H}} = Z_0 \mathbf{H}, \quad \widetilde{\mathbf{E}} = \mathbf{E}, \quad \widetilde{t} = c_0 t, \quad \text{and} \quad \widetilde{\mathbf{J}} = Z_0 \mathbf{J},$$

where $Z_0 = \sqrt{\frac{\mu_0}{\varepsilon_0}}$ is the vacuum impedance and $c_0 = \frac{1}{\sqrt{\varepsilon_0 \mu_0}}$ the speed of light in vacuum. It seems useful to remind the values and units of these constants, which is done in table 2.1. Then, the normalized system is:

Table 2.2 | Units of the original and the normalized Maxwell systems.

	H	**E**	**J**	t
Original unit	$\mathrm{A.m^{-1}}$	$\mathrm{V.m^{-1}}$	$\mathrm{A.m^{-2}}$	s
Normalized unit	$\mathrm{V.m^{-1}}$	$\mathrm{V.m^{-1}}$	$\mathrm{V.m^{-2}}$	m

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

The units of the original and normalized systems are given in table 2.2. Given the definitions of c_0 and Z_0, one sees that $\frac{\mu_0 c_0}{Z_0} = \varepsilon_0 c_0 Z_0 = 1$. Hence, dropping the tilde notation, one obtains the normalized Maxwell system, which will be exploited from now on:

Maxwell normalized system

$$\frac{\partial \mathbf{H}}{\partial t} = -\frac{1}{\mu_r}\nabla \times \mathbf{E}, \tag{2.11}$$
$$\frac{\partial \mathbf{E}}{\partial t} = \frac{1}{\varepsilon_r}\left(\nabla \times \mathbf{H} - \mathbf{J}\right). \tag{2.12}$$

2.1.3 Material interfaces

Ampere's, Faraday's and Gauss' laws

Ampere's, Faraday's and Gauss' laws are obtained by applying the Stokes and Ostrogradsky formulae to (2.2 – 2.5), leading to four integral forms that will help derive the interface conditions.

For a closed contour Γ delimitating a surface $\mathcal{S}$, one obtains Ampere's law by applying the Stokes formula to (2.3):

$$\oint_{\Gamma} \mathbf{H} \cdot d\mathbf{l} = \iint_{\mathcal{S}} \left(\mathbf{J} + \frac{\partial \mathbf{D}}{\partial t}\right) \cdot \mathbf{n}_{\mathcal{S}}\, d\mathcal{S}, \tag{2.13}$$

where $\mathbf{n}_{\mathcal{S}}$ is the unit normal to surface $\mathcal{S}$. In a similar fashion, applying Stokes formula to (2.2) yields Faraday's law:

$$\oint_{\Gamma} \mathbf{E} \cdot d\mathbf{l} = -\iint_{\mathcal{S}} \frac{\partial \mathbf{B}}{\partial t} \cdot \mathbf{n}_{\mathcal{S}}\, d\mathcal{S}. \tag{2.14}$$

For a closed surface Σ delimitating a volume $\mathcal{V}$, the Ostrogradsky formula applied to (2.4) and (2.5) respectively gives Gauss' laws for electric and magnetic fields:

$$\oiint_{\Sigma} \mathbf{D} \cdot \mathbf{n}_{\Sigma}\, d\Sigma = \iiint_{\mathcal{V}} \rho\, d\mathcal{V}, \tag{2.15}$$
$$\oiint_{\Sigma} \mathbf{B} \cdot \mathbf{n}_{\Sigma}\, d\Sigma = \mathbf{o}, \tag{2.16}$$

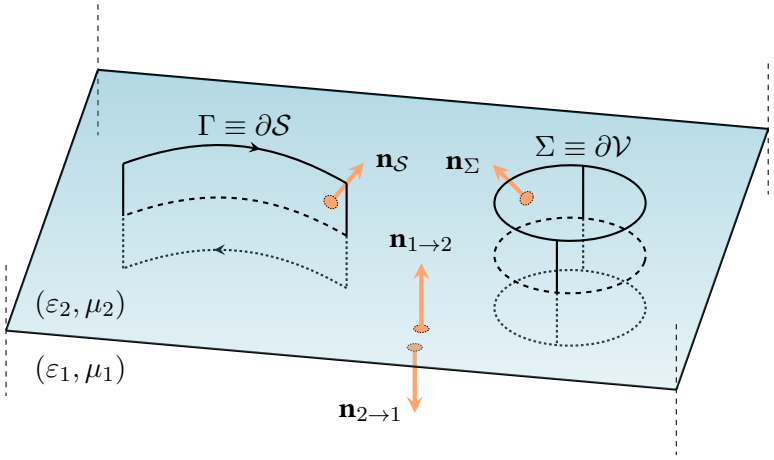


Figure 2.1 | Integration domains for jump relations. The blue plane represents the interface between the two materials. Γ is a closed curve on which the Ampere theorem is applied, while Σ is a closed surface used for the Gauss theorem. The interface hosts free surface currents $\mathbf{J}_s$ and charges ρ_s.

where $\mathbf{n}_\Sigma$ is the unit normal to surface Σ.

Interface conditions

In the presence of a material interface, *i.e.* a jump of ε_r or μ_r across a surface, the smoothness of the electromagnetic field is not preserved. To obtain a solution to Maxwell's equations, one must inspect the behavior of $\mathbf{E}$ and $\mathbf{H}$ across the discontinuity. To do so, consider the situation presented on figure 2.1. Suppose an interface between two materials of parameters (ε_1, μ_1) and (ε_2, μ_2). The adequate integration domains Γ and Σ are defined to apply Ampere's, Faraday's and Gauss' laws across the material interface. The interface is supposed to hold free surface currents $\mathbf{J}_s$ and charges ρ_s. Applying Ampere's law (2.13) to the closed contour Γ, and taking the cross-product with $\mathbf{n}_{1\to2}$ yields, after a few manipulations:

$$\mathbf{n}_{2\to1} \times \mathbf{H}_1 + \mathbf{n}_{1\to2} \times \mathbf{H}_1 = \mathbf{J}_s.$$

Here, the sign $\mathbf{J}_s$ obviously depends on the orientation of the surface. Taking into account that $\mathbf{n}_{1\to2} = -\mathbf{n}_{2\to1}$, these normals are indifferently replaced by $\mathbf{n}$ and $-\mathbf{n}$. Then, one obtains the following condition for the tangential magnetic field at the interface:

$$\mathbf{n} \times (\mathbf{H}_1 - \mathbf{H}_2) = \mathbf{J}_s.$$

On the other hand, Gauss' law for magnetic fields (2.16) yields:

$$\mu_1 \mathbf{n} \cdot \mathbf{H}_1 = \mu_2 \mathbf{n} \cdot \mathbf{H}_2.$$

In the same manner, Gauss's law for electric fields (2.15) gives:

$$\mathbf{n} \cdot (\varepsilon_1 \mathbf{E}_1 - \varepsilon_2 \mathbf{E}_2) = \rho_s.$$

Finally, Faraday's law (2.14) yields the continuity of the tangential electric field:

$$\mathbf{n} \times \mathbf{E}_1 = \mathbf{n} \times \mathbf{E}_2.$$

Hence, for a general material interface between two media, only the tangential component of $\mathbf{E}$ is continuous:

Interface conditions

$$\begin{aligned} \mathbf{n} \times (\mathbf{H}_1 - \mathbf{H}_2) &= \mathbf{J}_s, \\ \mathbf{n} \times (\mathbf{E}_1 - \mathbf{E}_2) &= \mathbf{o}, \\ \mathbf{n} \cdot (\mu_1 \mathbf{H}_1 - \mu_2 \mathbf{H}_2) &= 0, \\ \mathbf{n} \cdot (\varepsilon_1 \mathbf{E}_1 - \varepsilon_2 \mathbf{E}_2) &= \rho_s. \end{aligned} \tag{2.17}$$

Conditions on a perfect electric conductor

Following what was established above, it is easy to deduce the boundary conditions on a perfect electric conductor (PEC). Considering that all fields must be equal to zero inside the conductor, one obtains:

PEC conditions

$$\begin{aligned} \mathbf{n} \times \mathbf{H} &= \mathbf{J}_s, \\ \mathbf{n} \times \mathbf{E} &= \mathbf{o}, \\ \mathbf{n} \cdot \mathbf{H} &= 0, \\ \mathbf{n} \cdot \mathbf{E} &= \frac{\rho_s}{\varepsilon}. \end{aligned} \tag{2.18}$$

2.1.4 Some analytical solutions to Maxwell's equations

The handful of electromagnetic propagation problems that admit an analytical solution are essential in validating numerical implementations of electromagnetic solvers. In this section, the solutions to six elementary propagation problems are presented. They will be used as reference solutions later in this manuscript.

Plane wave in a homogeneous medium

In this section, Maxwell's equations are considered in a homogeneous medium of constant relative material parameters (ε_r, μ_r). Additionally, it is considered to be source-free, *i.e.* $\mathbf{J}$ and ρ are equal to zero. By combining the curl of (2.11) and the time derivative of (2.12), one obtains after some manipulations:

$$\Delta \mathbf{E} = \frac{1}{c_r^2} \frac{\partial^2 \mathbf{E}}{\partial t^2}, \tag{2.19}$$

where $c_r = \frac{1}{\sqrt{\mu_r \varepsilon_r}}$ is the relative speed of light. Taking the Fourier transform (see section 4.3) of (2.19) yields:

$$\Delta \widehat{\mathbf{E}} = \frac{\omega^2}{c_r^2} \widehat{\mathbf{E}}, \tag{2.20}$$

where $\widehat{\mathbf{E}}$ designates the frequency-dependent field associated to the time-dependent field $\mathbf{E}$, and ω the angular frequency. Propagating solutions of (2.20) in $\mathbb{R}^3$ are given by:

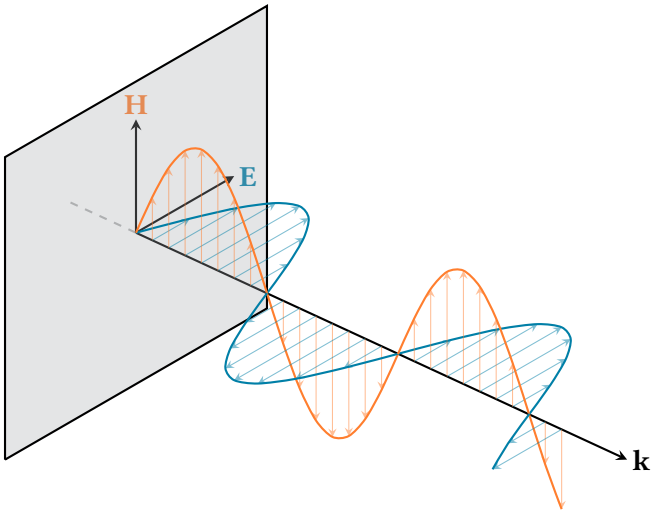


Figure 2.2 | Spatial representation of a linearly polarized plane wave. **E**, **H** and **k** are orthogonal two by two.

$$\widehat{\mathbf{E}} = \widehat{\mathbf{E}}_0 \, \mathrm{e}^{i\mathbf{k}\cdot\mathbf{x}},$$

with $\mathbf{k}$ the wave vector, related to the angular frequency ω by:

$$\omega^2 = |\mathbf{k}|^2 c_r^2.$$

The expression for $\widehat{\mathbf{H}}$ can be obtained by exploiting the Fourier transform of (2.11):

$$\widehat{\mathbf{H}} = \sqrt{\frac{\varepsilon_r}{\mu_r}} \frac{\mathbf{k}}{|\mathbf{k}|} \times \widehat{\mathbf{E}}_0 \, \mathrm{e}^{i\mathbf{k}\cdot\mathbf{x}},$$

It is now straightforward to deduce time-domain solutions for **E** and **H**:

$$\begin{aligned} \mathbf{E}(\mathbf{x}, t) &= \mathbf{E}_0 \left(t - \frac{\mathbf{k}\cdot\mathbf{x}}{|\mathbf{k}| c_r} \right), \\ \mathbf{H}(\mathbf{x}, t) &= \sqrt{\frac{\varepsilon_r}{\mu_r}} \frac{\mathbf{k}}{|\mathbf{k}|} \times \mathbf{E}. \end{aligned} \tag{2.21}$$

Hence, for plane waves, both **E** and **H** are constant at every point in the plane perpendicular to the propagation direction, while **E**, **H** and **k** are orthogonal two by two. Illustrated on figure (2.2) is the case of a rectilinear polarization (*i.e.* there is no initial phase delay between the different components of the electromagnetic fields).

Dielectric film in normal incidence

The considered set-up, shown in figure 2.3, consists in a thin slab made of medium ②, sandwiched between two media ① and ③, and infinite in the $x-$ and $x+$ directions (see figure 2.3). The geometry is periodic in both y and z directions. A plane wave traveling in ① in the $x+$ direction is considered, impinging in normal incidence on the slab. At the interface between ① and ②, the incident field $\mathbf{E}_{\text{inc}}$ is partially reflected to ①, and partially transmitted to ②, in the following fashion:

$$\mathbf{E}_1 = r_{12}\mathbf{E}_{\text{inc}} \quad \text{and} \quad \mathbf{E}_{12} = t_{12}\mathbf{E}_{\text{inc}}.$$

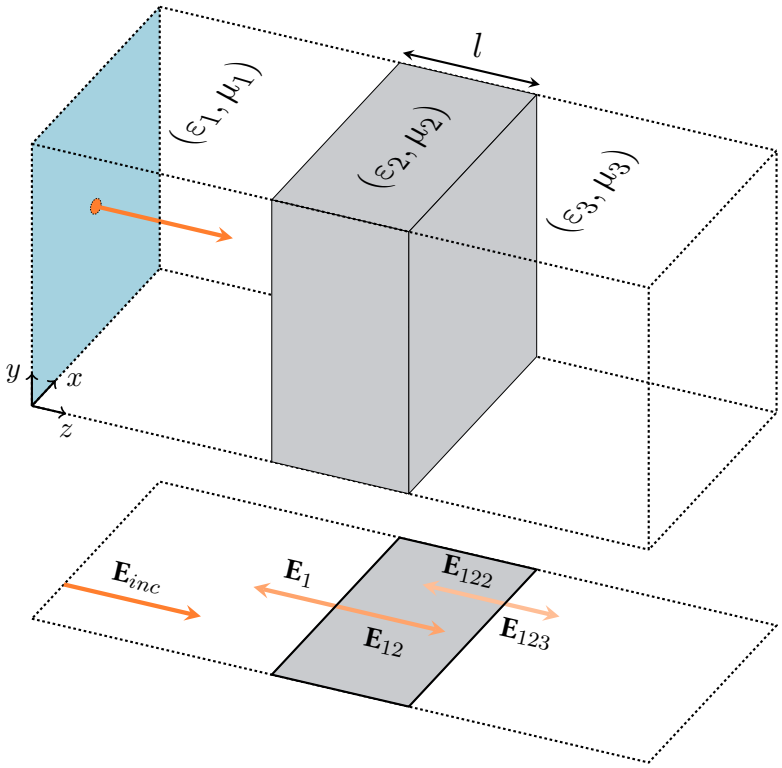


Figure 2.3 | Dielectric slab illuminated with a plane wave. The system is periodic in both x and y directions, while the incident plane wave propagates in the $z+$ direction. At the bottom of the picture, a few reflected and transmitted waves are represented.

Here, $\mathbf{E}_1$ represents the wave that was reflected in ① at the interface with ②, and $\mathbf{E}_{12}$ the wave that was transmitted from ① to ②. r_{12} is the amplitude reflection coefficient of ① on ②, and t_{12} is the amplitude transmission coefficient from ① to ②. From the interface relations (2.17), it is possible to deduce the values of these parameters:

$$r_{12} = \frac{n_1 - n_2}{n_1 + n_2} \quad \text{and} \quad t_{12} = \frac{2n_1}{n_1 + n_2},$$

where $n_i = \sqrt{\varepsilon_i \mu_i}$ is the refractive index of the i^{th} medium. While $\mathbf{E}_1$ propagates indefinitely toward $z-$, the same scenario is repeated when $\mathbf{E}_{12}$ reaches the interface between ② and ③:

$$\mathbf{E}_{122} = r_{21}\mathbf{E}_{12} \quad \text{and} \quad \mathbf{E}_{123} = t_{23}\mathbf{E}_{12}.$$

Eventually, an infinite number of reflections and transmissions occur at the two interfaces, yielding a solution in the form of an infinite summation of waves, all proportional to $\mathbf{E}_i$ *via* a composition of r and t coefficients. In the stead of exploiting a truncated solution in time domain, it is possible to calculate the power reflection and transmission coefficients, given by:

$$R = \frac{r_{12} + r_{23}\,\mathrm{e}^{-2ikl}}{1 + r_{12}r_{23}\,\mathrm{e}^{-2ikl}}, \tag{2.22}$$

$$T = \frac{(1 + r_{12})(1 + r_{23})\,\mathrm{e}^{-ikl}}{1 + r_{12}r_{23}\,\mathrm{e}^{-2ikl}}, \tag{2.23}$$

with k the modulus of the wavevector ($k = |\mathbf{k}|$) and l the thickness of the dielectric slab.

Perfect electric conductor cavities

Vacuum-filled cubic cavity Closed cavities surrounded by perfect electric conductor (PEC) walls in simple geometries also allow the full calculation of time-domain solutions. First, a parallelepipedic cavity of side lengths (a_x, a_y, a_z) filled with vacuum is considered. On all its external faces, PEC conditions are applied (see (2.18)). This cavity supports an infinite number of modes, whose expressions are of the form:

$$\mathbf{E}(x,y,z,t) = \begin{bmatrix} E_{x,0}\cos(k_x x)\sin(k_y y)\sin(k_z z) \\ E_{y,0}\sin(k_x x)\cos(k_y y)\sin(k_z z) \\ E_{z,0}\sin(k_x x)\sin(k_y y)\cos(k_z z) \end{bmatrix} \cos(\omega t),$$

$$\mathbf{H}(x,y,z,t) = \begin{bmatrix} H_{x,0}\sin(k_x x)\cos(k_y y)\cos(k_z z) \\ H_{y,0}\cos(k_x x)\sin(k_y y)\cos(k_z z) \\ H_{z,0}\cos(k_x x)\cos(k_y y)\sin(k_z z) \end{bmatrix} \sin(\omega t).$$

In the previous expressions, $k_i = \frac{n_i \pi}{a_i}$ for any integers $n_i \neq 0$. Following (2.11), the amplitude vectors of $\mathbf{E}$ and $\mathbf{H}$ are related as:

$$\mathbf{H}_0 = \frac{\mathbf{k} \times \mathbf{E}_0}{\omega},$$

where:

$$\mathbf{E}_0 = \begin{bmatrix} E_{x,0} \\ E_{y,0} \\ E_{z,0} \end{bmatrix} \quad \text{and} \quad \mathbf{H}_0 = \begin{bmatrix} H_{x,0} \\ H_{y,0} \\ H_{z,0} \end{bmatrix}$$

Additionally, the following equality must be verified to fulfill the divergence condition:

$$k_x E_{x,0} + k_y E_{y,0} + k_z E_{z,0} = 0.$$

Finally, the frequency of the (n_x, n_y, n_z) mode is given by:

$$\omega = \pi \sqrt{\frac{n_x^2}{a_x^2} + \frac{n_y^2}{a_y^2} + \frac{n_z^2}{a_z^2}}.$$

For the needs of this work, a $(n_x = n_y = n_z = 1)$ mode is considered in a unit cavity ($a_x = a_y = a_z = 1$). Hence, with $(k_x = k_y = k_z = k = \pi)$ and $\omega = \sqrt{3}\pi$:

$$\begin{aligned} \mathbf{E}(x,y,z,t) &= \begin{bmatrix} -\cos(kx)\sin(ky)\sin(kz) \\ 0 \\ \sin(kx)\sin(ky)\cos(kz) \end{bmatrix} \cos(\omega t), \\ \mathbf{H}(x,y,z,t) &= \begin{bmatrix} -\sin(kx)\cos(ky)\cos(kz) \\ 2\cos(kx)\sin(ky)\cos(kz) \\ -\cos(kx)\cos(ky)\sin(kz) \end{bmatrix} \frac{k}{\omega} \sin(\omega t). \end{aligned} \tag{2.24}$$

Vacuum-filled spherical cavity In the case of a spherical cavity of unit radius, a similar, however more tedious derivation is possible. For the needs of this work, the following $(0, 1, 1)$ mode will be considered:

$$\begin{aligned} \mathbf{H}(x,y,z,t) &= \frac{\sin(\omega t)}{kr^2} \left(\frac{\sin(kr)}{kr} - \cos(kr) \right) \begin{bmatrix} -y \\ x \\ 0 \end{bmatrix}, \\ \mathbf{E}(x,y,z,t) &= \frac{z\cos(\omega t)}{k^2 r^4} \left(\sin(kr) \left(kr - \frac{3}{kr} \right) + 3\cos(kr) \right) \begin{bmatrix} x \\ y \\ z \end{bmatrix} \\ &\quad - \frac{\cos(\omega t)}{k^2 r^2} \left(\sin(kr) \left(kr - \frac{1}{kr} \right) + \cos(kr) \right) \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \end{aligned} \tag{2.25}$$

where $r = |\mathbf{x}|$. The mode frequency is solution of a transcendental equation, and the approximate value $\omega \simeq 0.13091174401040966770$ GHz is retained.

Cubic cavity filled with an anisotropic material The solution of the cubic cavity filled with an anisotropic material is also available. In the most general case, the permittivity tensor is real and symmetric, and can therefore be diagonalized. Hence, we here restrain ourselves to the case of diagonal permittivity tensors $\bar{\bar{\varepsilon}}_r = \mathrm{diag}\left[\varepsilon_x, \varepsilon_y, \varepsilon_z\right]$. Unlike the isotropic case, for a given $\bar{\bar{\varepsilon}}_r$ the dispersion relation is a fourth-order equation in ω, which allows two real modes in the cavity. For the sake of brevity, the full derivation is not detailed here, and we settle for providing an explicit solution of the problem. As in the isotropic case, modes are of the form:

$$\mathbf{E}_j(x,y,z,t) = \begin{bmatrix} E_{x,j,0}\cos(k_x x)\sin(k_y y)\sin(k_z z) \\ E_{y,j,0}\sin(k_x x)\cos(k_y y)\sin(k_z z) \\ E_{z,j,0}\sin(k_x x)\sin(k_y y)\cos(k_z z) \end{bmatrix} \cos(\omega_j t),$$
$$\mathbf{H}_j(x,y,z,t) = \begin{bmatrix} H_{x,j,0}\sin(k_x x)\cos(k_y y)\cos(k_z z) \\ H_{y,j,0}\cos(k_x x)\sin(k_y y)\cos(k_z z) \\ H_{z,j,0}\cos(k_x x)\cos(k_y y)\sin(k_z z) \end{bmatrix} \sin(\omega_j t),$$

where $j \in \{1,2\}$. Once again, we choose the $(n_x = n_y = n_z = 1)$ modes in a unit cavity $(a_x = a_y = a_z = 1)$, which gives $(k_x = k_y = k_z = k = \pi)$. The anisotropic material is chosen as $\bar{\bar{\varepsilon}}_r = \mathrm{diag}\,[1,3,5]$, which yields two possibles modes, $\omega_1 \simeq 0.837624$ GHz and $\omega_2 \simeq 1.42078$ GHz, both solutions of the dispersive relation. Finally, the divergence condition is taken into account to choose the amplitude vectors $\mathbf{E}_{j,0}$ and $\mathbf{H}_{j,0}$. A possible choice is:

$$\begin{aligned} E_{x,j,0} &= (\varepsilon_{r,2} - \varepsilon_{r,3})\, k^2, \\ E_{y,j,0} &= \varepsilon_{r,1}\varepsilon_{r,3}\omega_j^2 - (2\varepsilon_{r,3} + \varepsilon_{r,1})\, k^2, \\ E_{z,j,0} &= -\frac{1}{\varepsilon_{r,3}} \left(\varepsilon_{r,1} E_{x,j,0} + \varepsilon_{r,2} E_{y,j,0}\right), \\ \mathbf{H}_{j,0} &= \frac{\mathbf{k} \times \mathbf{E}_{j,0}}{\omega_j}. \end{aligned}$$

Solutions based on the Mie theory

The Mie theory [vdH81] was derived in 1908 by Gustav Mie, and brings an analytical solution to the scattering of spherical particles in the form of infinite series of Hankel functions and Legendre polynomials. Starting from the Helmholtz equation, the fields are split in separate variables. After a few calculations, the radial part is solution of a Bessel equation, the polar part, of a Legendre equation, while the azimuthal part verifies a simple oscillatory problem. The aforementioned reference contains the detailed derivation, which is thus not reported here. Among others, it allows for the computation of the near and far field, as well as the cross-sections (see section 4.4) of spherical scatterers. In this manuscript, it will be exploited as a reference solution in section 5.3 to compare the computed cross-section of a metallic sphere.

2.2 Dispersive models

2.2.1 Underlying physics

Dispersion is a common phenomenon to all kinds of waves traveling through a medium: it results from the way the latter reacts to the presence of the wave, therefore affecting its propagation. For a polychromatic wave, it often happens that all the frequencies do not travel at the same speed through the medium: this phenomenon is called dispersion. Among the numerous phenomena encountered in electromagnetics, many rely on the dispersive properties of materials. Indeed, in specific ranges of wavelengths, biological tissues [GGC96], noble [JC72] and transition metals [JC74], but also glass [Fle78] and certain polymers [CC41] exhibit non-negligible dispersive behaviors. In the mathematical framework, this phenomenon is modeled by a frequency-dependent permittivity[2] function $\varepsilon(\omega)$, often derived from physical considerations. Regarding nanophotonics applications, an accurate modeling of the permittivity function for

[2] We remind the reader that this work is restrained to non-magnetic materials

metals in the visible spectrum is crucial. Indeed, the free electrons of metals are the key ingredient in the propagation of surface plasmons [NH07] (see section 2.2.5 for more details).

In the presence of an exterior electric field, the electrons of a metal are subject to a Coulomb force which brings them, in a given characteristic time τ_c, to an equilibrium position. This leads to a general electric polarization of the metal, which is usually expressed in the frequency domain with the polarization vector $\widehat{\mathbf{P}}$. The latter constitutes an additional term to the electric displacement: $\widehat{\mathbf{D}} = \varepsilon_0\widehat{\mathbf{E}} + \widehat{\mathbf{P}}$. Moreover, $\widehat{\mathbf{P}}$ can be related to $\widehat{\mathbf{E}}$ in homogeneous isotropic media through its susceptibility $\chi(\omega)$ such that $\widehat{\mathbf{P}} = \chi(\omega)\widehat{\mathbf{E}}$. If one is to consider a variable electric field of given angular frequency ω, the frequency dependence of $\widehat{\mathbf{P}}$ can be intuitively understood: for sufficiently low frequencies, the electrons relaxation time τ is negligible compared to $\frac{1}{\omega}$. Therefore, the electrons dispose of a sufficient amount of time to adapt to the variations of the electric field. However, at higher frequencies, the field varies significantly during the time τ required by the electrons to reach a stable state. Then, the higher the frequency, the shorter the distance traveled by the electrons from their steady state equilibrium, and the lower the polarization. This explains the observed transparency of the metals for very high frequencies electromagnetic waves. One should now grasp the importance of taking the dispersion effects into account when $\widehat{\mathbf{P}}$ cannot be neglected, since it has a significant influence on the permittivity $\varepsilon(\omega)$ of the considered medium, and hence on its refractive index.

2.2.2 Drude and Drude-Lorentz models

The Drude model is based on the kinetic theory of gases [Druoo]. In this approximation, the metal is considered as a static lattice of positive ions immersed in a free electrons gas. The interactions of these electrons with the ion lattice are condensed in a collision frequency parameter γ_d, while electron-electron interactions are totally neglected. For the electron gas, this leads to the following classical equation of motion:

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

where m_e represents the electron mass, and e the electronic charge. It is worth noticing that γ_d matches the definition of the inverse of the mean free path τ_f. Then, considering a harmonic time-dependence of the form $\mathrm{e}^{-i\omega t}$, one obtains:

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

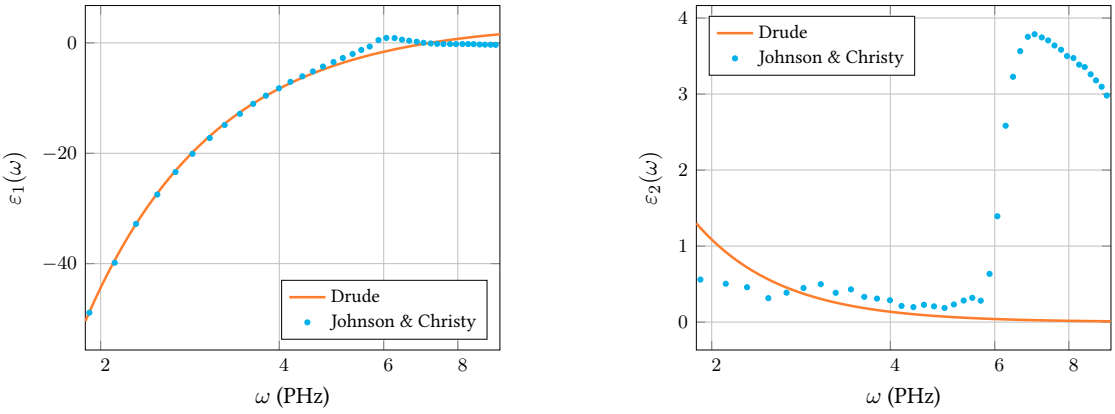
Given the definition of the polarization $\widehat{\mathbf{P}} = -n_e e\widehat{\mathbf{x}}$, with n_e the electronic density, the latter equality can be rewritten as:

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

Then, 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 [Maio7]:



(a) Real part

(b) Imaginary part

Figure 2.4 | Real and imaginary parts of the silver relative permittivity predicted by the Drude model compared to experimental data from Johnson & Christy. The parameter values are $\varepsilon_\infty = 3.7362$, $\omega_d = 1.3871 \times 10^7$ GHz and $\gamma_d = 4.5154 \times 10^4$ GHz.

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

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

$$\varepsilon_{r,d}(\omega) = \varepsilon_\infty - \frac{\omega_d^2}{\omega^2 + i\omega\gamma_d}, \tag{2.26}$$

where $\varepsilon_\infty = 1 + \chi_b$ is the permittivity at infinite frequency. The real and imaginary parts of the Drude permittivity function for silver are plotted in figure 2.4, along with experimental curves from Johnson and Christy [JC72]. One notices that, if the real part fits the Drude prediction, the experimental imaginary part shows features that are not predicted by the model. For certain metals (especially noble ones), electronic transitions between valence and conduction band occur around the visible frequency range. These contributions correspond to electrons that are bound to their ion cores. Hence, in the same classical fashion as before, a spring term is added to the equation of motion:

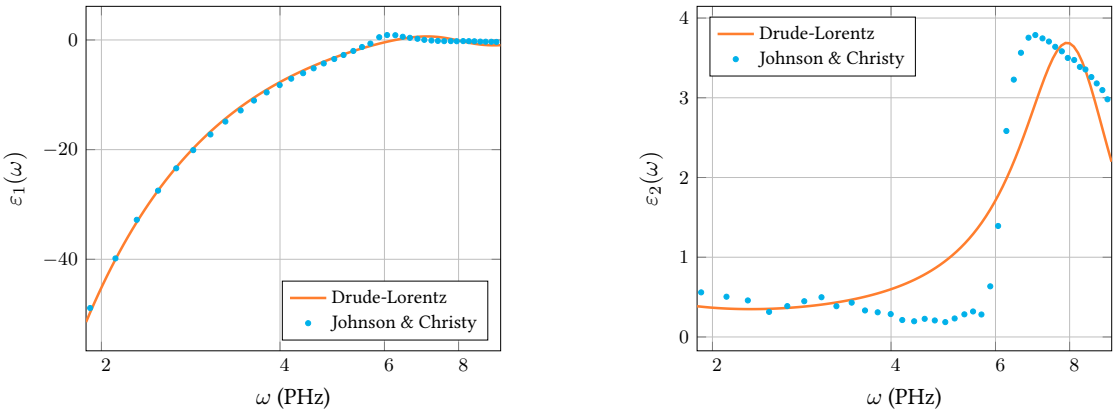
$$\frac{\partial^2 \mathbf{x}}{\partial t^2} + \gamma_l \frac{\partial \mathbf{x}}{\partial t} + \omega_l^2 \mathbf{x} = -\frac{e}{m_e} \mathbf{E}(t).$$

Following the same development as for the Drude model, one easily 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 total permittivity of the Drude-Lorentz model is the simple addition of the Drude and Lorentz terms:

$$\varepsilon_{r,dl}(\omega) = \varepsilon_\infty - \frac{\omega_d^2}{\omega^2 + i\omega\gamma_d} - \frac{\Delta\varepsilon\omega_l^2}{\omega^2 - \omega_l^2 + i\omega\gamma_l}. \tag{2.27}$$



(a) Real part

(b) Imaginary part

Figure 2.5 | Real and imaginary parts of the silver relative permittivity predicted by the Drude-Lorentz model compared to experimental data from Johnson & Christy. The parameter values are $\varepsilon_\infty = 2.7311$, $\omega_d = 1.4084 \times 10^7$ GHz, $\gamma_d = 6.6786 \times 10^3$ GHz, $\Delta\varepsilon = 1.6336$, $\omega_l = 8.1286 \times 10^6$ GHz and $\gamma_l = 3.6448 \times 10^6$ GHz.

Here, the $\Delta\varepsilon$ parameter represents the amplitude of the associated Lorentz pole. As can be seen for silver in figure 2.5, the high-frequency range of the imaginary part is in better adequation with experimental data than it was for the Drude model. However, there is still room for improvement: for some metals such as gold or silver, the addition of multiple Lorentz terms brings a much better fit between experimental and theoretical values, at the cost of an increased complexity of the model. Based on this remark, the L4 model of [HN07] combines four Lorentz poles with a conductivity term.

2.2.3 Generalized model

Given an experimental set of points describing a permittivity function of a material, a Padé type approximation is a convenient analytical coefficient-based function to approach experimental data. 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

$$\varepsilon_{r,g}(\omega) = \varepsilon_\infty - \frac{\sigma}{i\omega} - \sum_{l\in L_1} \frac{a_l}{i\omega - b_l} - \sum_{l\in L_2} \frac{c_l - i\omega d_l}{\omega^2 - e_l + i\omega f_l}, \tag{2.28}$$

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 real constants, and L_1, L_2 are non-overlapping sets of indices. The constant ε_∞ represents the permittivity at infinite frequency, and σ the conductivity.

This general writing allows an important flexibility for several reasons. First, it unifies most of the common dispersion models in a single formulation. Indeed, Debye (biological tissues in the MHz regime), Drude and Drude-Lorentz (noble metals in the THz regime), retarded Drude and Drude-Lorentz (transition metals in the THz regime), but also Sellmeier's law (glass in the THz regime), are naturally included. Second, as will be shown later, it permits to fit a large range of experimental data set in a limited number of poles (thus leading to reasonable memory and CPU overheads). A similar approach was used in the

Table 2.3 | Quality of the fit obtained by various dispersion models for silver and gold in the $[300, 1500]$ THz range.

	Silver		Gold	
	Δ_r	Δ_i	Δ_r	Δ_i
Drude	0.8366	1.622	1.715	3.752
Drude-Lorentz	0.4649	0.4412	0.5482	0.5759
L4	0.2028	0.2199	0.2354	0.3256
1SOGP	0.8366	1.738	1.328	2.960
2SOGP	0.2061	0.2458	0.2034	0.1891
4SOGP	0.08928	0.06690	0.1019	0.1237

case of the Critical Points (CP) model with two (see [VLDC11]) and three (see [LC09]) poles, and in the Complex-Conjugate Pole-Residue Pairs model (CCPRP) (see [HDF06]). In essence, these techniques allow for complex coefficients in their developments, and can therefore write the decomposition of the permittivity function in pairs of single-order poles only, whereas choosing real coefficients leads to a collection of first-order and second-order poles. However, the numerical complexity of their implementations is equivalent to that of the generalized dispersive model.

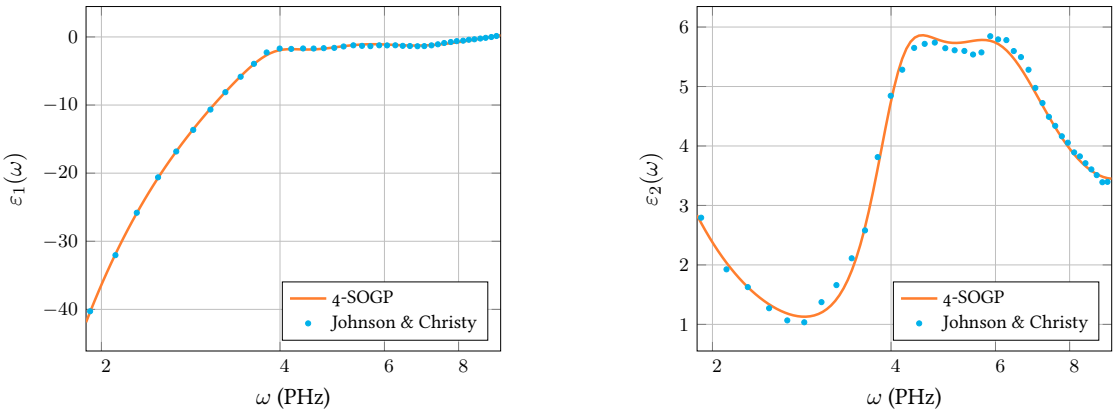
In order to fit the coefficients of (2.28) to experimental data, various techniques can be used, such as the well-known least square method. Vector fitting techniques (see [GS99]) are also well developed for the CCPRP formulation. For an increasing number of poles, one can be left with a large optimization problem presenting many local maxima. Simulated Annealing (SA) methods have proved to be particularly efficient in finding global maxima in these situations, even when the initial guess is far from the optimal point ([KGV83]). Hence, a free existing algorithm from W. L. Goffe[3] was adapted for this study. In practice, for a given model, a set of experimental data is provided to the optimization algorithm: in this study, the well-known Johnson and Christy tables ([JC72], [JC74]) were exploited, although others are also widely-used [Pal98]. This method demonstrated good efficiency while fitting up to 17 parameters simultaneously.

A key point in the quality of the fitting is the wideness of the spectrum of interest. Indeed, for a fixed number of parameters and poles, and depending on the behavior of the experimental permittivity function in the selected frequency range, one can obtain a good or a poor fit. In this section, the frequency interval is set to $[300, 1500]$ THz, which constitutes a wide enough range for most problems in this manuscript.

For gold, silver and copper, the results obtained with Drude, Drude-Lorentz, 2SOGP, 4SOGP, and L4 models are compared. All the parameters were fitted with the SA algorithm. The quality of the fit is evaluated by a point-by-point L^1 error normalized by the number of experimental samples. The quality of the real part fit is noted Δ_r, while Δ_i is the one of the imaginary part. Results are displayed in table 2.3. As can be seen, using SOGP instead of classical Drude and Lorentz poles provides a neat benefit in the description of the permittivity function. The 2SOGP and 4SOGP fits reduce the errors by a factor 2 when compared to the Drude-Lorentz and L4 fits, for both gold and silver. The only case where no improvement is obtained is the 1SOGP fit for silver, compared to the Drude model. As an illustration, the fitting obtained for gold with the 4SOGP model is presented in figure 2.6.

Although gold and silver are widely used, transition metals, such as nickel, cobalt or iron, were recently considered for plasmonic applications [PPM+14]. However, the permittivity functions of such metals cannot be represented by the classical Drude model, since the latter assumes the electrons to be

[3] http://ideas.repec.org/c/wpa/wuwppr/9406001.html



(a) Real part

(b) Imaginary part

Figure 2.6 | Real and imaginary parts of the gold relative permittivity predicted by the 4SOGP model compared to experimental data from Johnson & Christy. The parameter values can be found in appendix A.

Table 2.4 | Quality of the fit obtained by Drude and 1SOGP models for nickel in the $[300, 1500]$ **THz range.**

	Δ_r	Δ_i
Drude	1.079	8.323
1SOGP	1.1272	0.8750

non-correlated. In transition metals, however, this assumption is not true, and the global equilibrium is not reached instantaneously, inducing a retardation effect [WROB13]. In the latter reference, a retarded Drude model is derived from physical considerations, which can be represented by a proper choice of parameters with a SOGP, the d_l parameter in (2.28) being linked to the relaxation time scale. Here, this feature is illustrated by computing coefficients for nickel with Drude and 1SOGP models: the results are displayed in table 2.4, and a graphic representation can be found in figure 2.7. While the real part is close to experimental data, one clearly sees how the Drude model underestimates the losses in the metal. The improvement of the imaginary part with the 1SOGP model is very appreciable, for a constant memory cost.

2.2.4 Maxwell's equations in dispersive materials

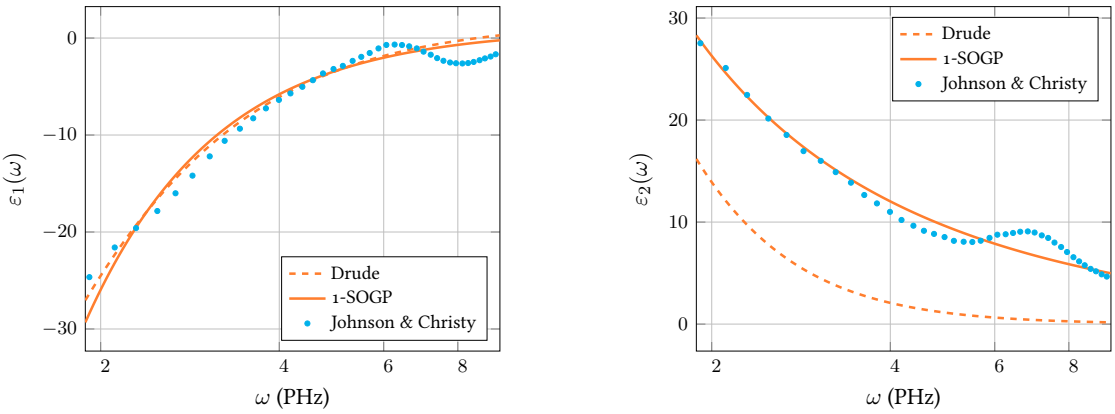
In this section, the modified Maxwell's equations for dispersive media are derived. As a first example, the Drude model is considered. Then, the equations for the generalized dispersive model are given.

Maxwell-Drude equations

Consider the case of a frequency-dependent medium, under the hypothesis of a Drude model:

$$\varepsilon_r(\omega) = \varepsilon_\infty - \frac{\omega_d^2}{\omega^2 + i\omega\gamma},$$

The constitutive relation on $\widehat{\mathbf{E}}$, already given in (2.9), can now be written:



(a) Real part

(b) Imaginary part

Figure 2.7 | Real and imaginary parts of the nickel relative permittivity predicted by the Drude and the 1SOGP model compared to experimental data from Johnson & Christy. The parameter values are, for Drude: $\varepsilon_\infty = 1.78$, $\omega_d = 1.16 \times 10^7$ GHz and $\gamma_d = 1.058 \times 10^6$ GHz; for 1SOGP: $\varepsilon_\infty = 1.0$, $c_1 = 1.1943 \times 10^{14}$ GHz^2, $d_1 = 4.6603 \times 10^7$ GHz, $e_1 = 0.0$ GHz^2 and $f_1 = 2.2176 \times 10^5$ GHz.

$$\widehat{\mathbf{D}}(\omega) = \varepsilon_0 \varepsilon_\infty \widehat{\mathbf{E}}(\omega) - \varepsilon_0 \frac{\omega_d^2}{\omega^2 + i\omega\gamma} \widehat{\mathbf{E}}(\omega).$$

which is traditionally summed up under the following formulation:

$$\widehat{\mathbf{D}}(\omega) = \varepsilon_0 \varepsilon_\infty \widehat{\mathbf{E}}(\omega) + \widehat{\mathbf{P}}, \tag{2.29}$$

where $\widehat{\mathbf{P}}$ is the polarization of the medium. Combining the inverse Fourier transform of (2.29) with Maxwell's equations yields:

$$\begin{aligned} \mu_0 \frac{\partial \mathbf{H}}{\partial t} &= -\nabla \times \mathbf{E}, \\ \varepsilon_0 \varepsilon_\infty \frac{\partial \mathbf{E}}{\partial t} &= \nabla \times \mathbf{H} - \frac{\partial \mathbf{P}}{\partial t} - \mathbf{J}_s. \end{aligned}$$

Here, the notation $\mathbf{J}_s$ refers to the source currents only, to avoid confusion with other types of currents. Above, $\widehat{\mathbf{P}}$ was defined as:

$$\widehat{\mathbf{P}}(\omega) = -\varepsilon_0 \frac{\omega_d^2}{\omega^2 + i\omega\gamma} \widehat{\mathbf{E}}(\omega).$$

Hence, an inverse Fourier transform gives:

$$\frac{\partial^2 \mathbf{P}}{\partial t^2} + \gamma_d \frac{\partial \mathbf{P}}{\partial t} = \varepsilon_0 \omega_d^2 \mathbf{E}. \tag{2.30}$$

By defining the dipolar current vector $\mathbf{J}_p = \dfrac{\partial \mathbf{P}}{\partial t}$, the global system can be rewritten as follows:

$$\begin{aligned}
\mu_0 \frac{\partial \mathbf{H}}{\partial t} &= -\nabla \times \mathbf{E}, \\
\varepsilon_0 \varepsilon_\infty \frac{\partial \mathbf{E}}{\partial t} &= \quad \nabla \times \mathbf{H} - \mathbf{J}_p - \mathbf{J}_s, \\
\frac{\partial \mathbf{J}_p}{\partial t} &= -\gamma_d \mathbf{J}_p + \varepsilon_0 \omega_d^2 \mathbf{E}.
\end{aligned}$$

After a normalization similar to what was done in section 2.1.2, one obtains:

$$\begin{aligned}
\frac{\partial \mathbf{H}}{\partial t} &= -\nabla \times \mathbf{E}, \\
\frac{\partial \mathbf{E}}{\partial t} &= \frac{1}{\varepsilon_\infty} \left(\nabla \times \mathbf{H} - \mathbf{J}_p - \mathbf{J}_s \right), \\
\frac{\partial \mathbf{J}_p}{\partial t} &= -\gamma_d \mathbf{J}_p + \omega_d^2 \mathbf{E}.
\end{aligned} \tag{2.31}$$

In the latter system, γ_d and ω_d are normalized by c_0.

Maxwell-generalized dispersive model equations

Following similar steps as for the Drude model, one derives the system of PDEs, accounting for the generalized dispersive model in time-domain:

Maxwell-Generalized dispersive model

$$\begin{aligned}
\frac{\partial \mathbf{H}}{\partial t} &= -\nabla \times \mathbf{E}, \\
\frac{\partial \mathbf{E}}{\partial t} &= \frac{1}{\varepsilon_\infty} \left(\nabla \times \mathbf{H} - \mathbf{J}_s - \boldsymbol{\mathcal{J}}_0 - \sum_{l \in L_1} \boldsymbol{\mathcal{J}}_l - \sum_{l \in L_2} \boldsymbol{\mathcal{J}}_l \right), \\
\boldsymbol{\mathcal{J}}_0 &= (\sigma + \sum_{l \in L_2} d_l) \mathbf{E}, \\
\boldsymbol{\mathcal{J}}_l &= a_l \mathbf{E} - b_l \mathbf{P}_l && \forall l \in L_1, \\
\frac{\partial \mathbf{P}_l}{\partial t} &= \boldsymbol{\mathcal{J}}_l && \forall l \in L_1, \\
\frac{\partial \boldsymbol{\mathcal{J}}_l}{\partial t} &= (c_l - d_l f_l) \mathbf{E} - f_l \boldsymbol{\mathcal{J}}_l - e_l \mathbf{P}_l && \forall l \in L_2, \\
\frac{\partial \mathbf{P}_l}{\partial t} &= d_l \mathbf{E} + \boldsymbol{\mathcal{J}}_l && \forall l \in L_2.
\end{aligned} \tag{2.32}$$

2.2.5 An illustration: a metallic sphere

In order to illustrate the concept of localized surface plasmon, the illumination of a gold nanosphere of radius $R = 50$ nm by a plane wave of unit amplitude is considered. The gold is described by a simple Drude model of parameters $\varepsilon_\infty = 1.0$, $\gamma_d = 3.23 \times 10^4$ GHz, and $\omega_d = 1.39 \times 10^7$ GHz. For this configuration, the Mie theory predicts a resonance around $f_{\text{res}} = 1053$ THz [Maio7]. At this frequency, the collective oscillation of electrons reaches its maximal amplitude, producing locally enhanced electric

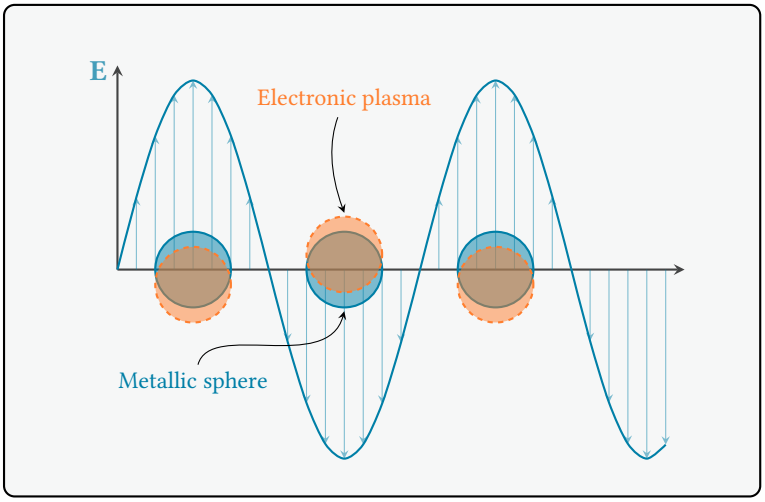
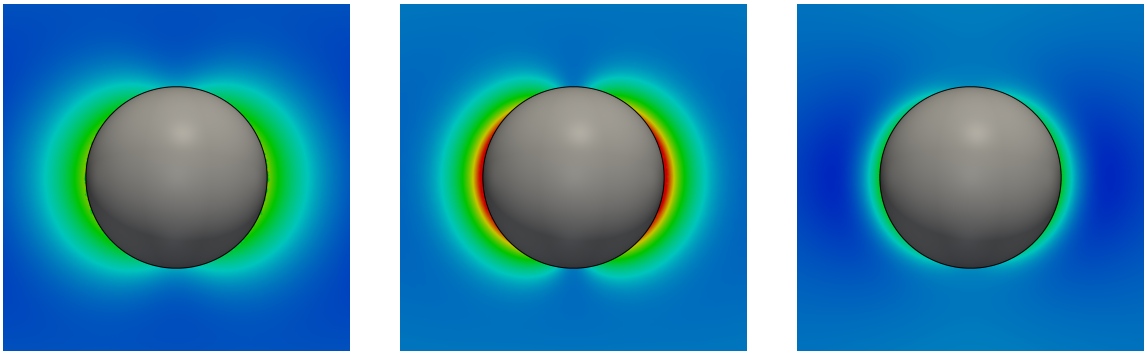


Figure 2.8 | Plasmon oscillation in a metallic sphere due to an exterior electric field. This scheme assumes that the plasmonic oscillation takes place instantly as the electric field is imposed. The electronic displacement causes an induced electric field outside the sphere. Its maximum intensity is reached in the polarization direction, at the dielectric/metal interface.

field in the vicinity of the sphere, along the polarization direction of the incident plane wave (see figure 2.8). To illustrate this resonance, the full Fourier transform of the electric field is computed on the whole domain, at frequencies below, equal and above the resonance frequency. The results are displayed on figure 2.9. A local field enhancement factor of approximately 8 is obtained at $f = f_{\text{res}}$ at the poles of the sphere in the $\mathbf{x}$ direction, which corresponds to the polarization of the incident wave. Below and above f_{res}, the observed field enhancement is weaker.



(a) $f = 0.8 \times f_{\text{res}}$ (b) $f = f_{\text{res}}$ (c) $f = 1.2 \times f_{\text{res}}$

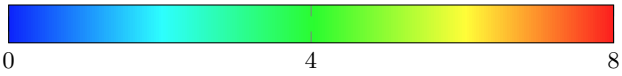


Figure 2.9 | Plasmonic resonance of a gold nanosphere. The plots show the modulus of the electric field Fourier transform. The metal is described by a Drude model. The full resonance is obtained for $f_{\text{res}} = 1053$ THz (figure 2.9(b)). Above and below f_{res} (resp. figures 2.9(a) and 2.9(c)), the local field enhancement is weaker. In all figures, the maximum electric field amplitude is arbitrarily set to 8.

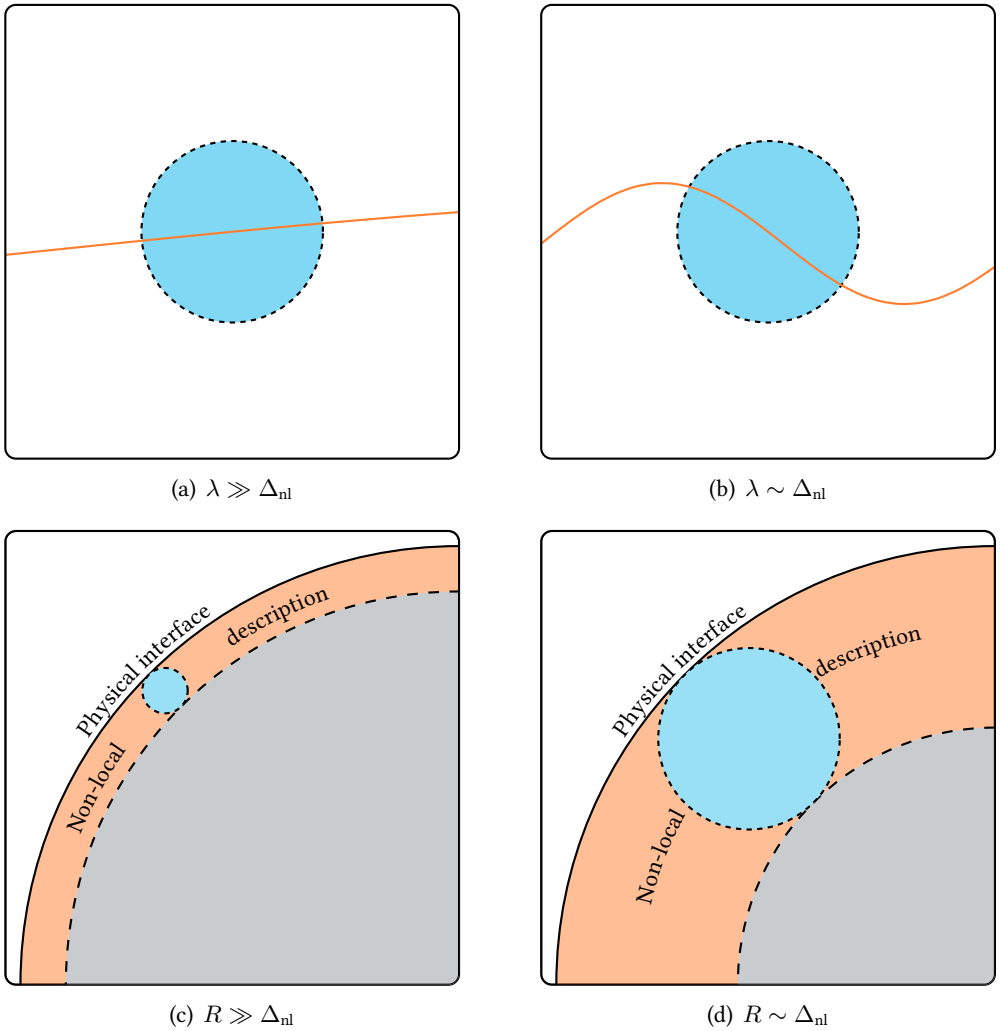
2.2.6 A digression on non-local models

In the classical description of the electron, the latter is considered to be a point particle, *i.e.* its size is negligible. This assumption has two important consequences, which are (i) electrons do not interact with each other, and (ii) the response of an electron to an exterior field only depends on the value of this field at its location. However, quantum mechanics teach us that, under the wave/particle duality description, an electron should be described by a probability density function: from a naive (and sufficient) point of view, that means that to be more rigorous, an electron should be considered to have a non-zero spatial extent, the latter being called "range of non-locality", noted Δ_{nl}, and which typical value would be a few Å [ESVM+06].

One would now be interested in knowing when the impact of this range of non-locality is to be considered, and when it is not. To be as clear as possible, consider the different situations presented on figure 2.10. In case 2.10(a), the wavelength λ of the electric field experienced by the electron is much larger than Δ_{nl}. Therefore, a relatively reasonable error is committed by assuming that (i) the response of the electron will be mainly driven by the value of $\mathbf{E}$ at the center of its non-locality range, and (ii) the electron will have no influence on its neighbors. The resulting error would typically be of order $\frac{\Delta_{\text{nl}}}{\lambda}$ [ESVM+06]. On the contrary, in 2.10(b), one sees that an important error is committed, since the electric field varies significantly on a typical length of Δ_{nl}. In this case, it is required to consider that non-local effects will be brought into play. An easy calculation shows that a 1 % error is committed for a wavelength roughly equal to 10 nm, which approximately corresponds to a few PHz. However, one must not think that below such values, non-locality does not take part into photonic calculations. Indeed, another situation where an electron experiences important variations of the electric field inside its range of non-locality is the edge of the considered nano-structure, where material properties (and therefore fields) change rapidly: as depicted on figure 2.10(c), every electron found in the orange shell would be poorly described by a standard model at any frequency. Again, this impact would be limited in every case where the typical dimension (hereafter noted R) of the object (or the gap) is large compared to Δ_{nl}. Indeed, in the standard model, all the electrons are considered to reside in a shell of infinitesimal thickness at the surface of the object. But, as can be seen on figure 2.10(d), as soon as R is of the order of a few Δ_{nl}, the response of the material is not properly described for a majority of its free electrons, leading to biased results. In order not to confuse causes and consequences, the latter paragraphs are summed up in a few words:

- The non-local model gives a better description of what an electron is and how it reacts to the presence of (i) an external electric field, and (ii) other electrons. This improved description is made by defining the non-locality range, which is a consequence of the quantum theory;
- The sensibility of the electron to very short wavelengths is therefore a consequence of the latter point;
- As well, the fact that the electrons "see the walls" as well as each other is a consequence of their non-zero spatial extent.

The spatial extension of the electrons causes a new kind of interaction that was excluded up to now, namely the electron-electron interaction. The quantum theory tells us that the electrons, as every other fermions, obey the famous Pauli principle which states that two identical fermions cannot occupy the same quantum state at the same time. At our level, the main consequence is that two electrons will experience a repulsive force between them that will be proportional to the overlapping of their respective non-locality ranges. In the hydrodynamic evolution equation, this repulsion will be expressed by a pressure term proportional to the gradient of the electronic density [Boa82]:



(a) $\lambda \gg \Delta_{\text{nl}}$ (b) $\lambda \sim \Delta_{\text{nl}}$

(c) $R \gg \Delta_{\text{nl}}$ (d) $R \sim \Delta_{\text{nl}}$

Figure 2.10 | Non-locality is required when the wavelength and/or the geometrical features are of comparable size with the range of non-locality (right panels). Otherwise, a local description is usually sufficient (left panels).

$$m\frac{\partial n\mathbf{v}}{\partial t} = n_e e\mathbf{E} - m\gamma_d\mathbf{v} - m\beta^2\nabla n. \tag{2.33}$$

Therefore, electronic density profiles will be smoothened by the pressure term, and higher field values will be required to increase the density. Here, β is a phenomenological parameter whose value is proportional to the Fermi velocity[4] v_F. The proportionality constant depends on the dimension of the problem, as well as on the frequency. A short discussion on this matter can be found in [MCS13]. The same development as in the standard Drude case can be followed:

$$\frac{\partial \mathbf{J}}{\partial t} = \omega_d^2\mathbf{E} - \gamma_d\mathbf{J} - \beta^2\nabla ne. \tag{2.34}$$

Integrating the continuity equation with respect to time leads to the following equality, introducing the polarization $\mathbf{P}$ such as $\frac{\partial \mathbf{P}}{\partial t} = \mathbf{J}$:

$$ne + \nabla\cdot\mathbf{P} = 0.$$

Considering that the system is initially at rest, the integration constant is taken equal to 0. Plugging the latter equality into (2.34) yields:

$$\frac{\partial^2\mathbf{P}}{\partial t^2} = \omega_d^2\mathbf{E} - \gamma_d\frac{\partial\mathbf{P}}{\partial t} + \beta^2\nabla\left(\nabla\cdot\mathbf{P}\right). \tag{2.35}$$

This additional differential equation (ADE) to the Maxwell system can be expressed in the (k,ω) space through the following relation:

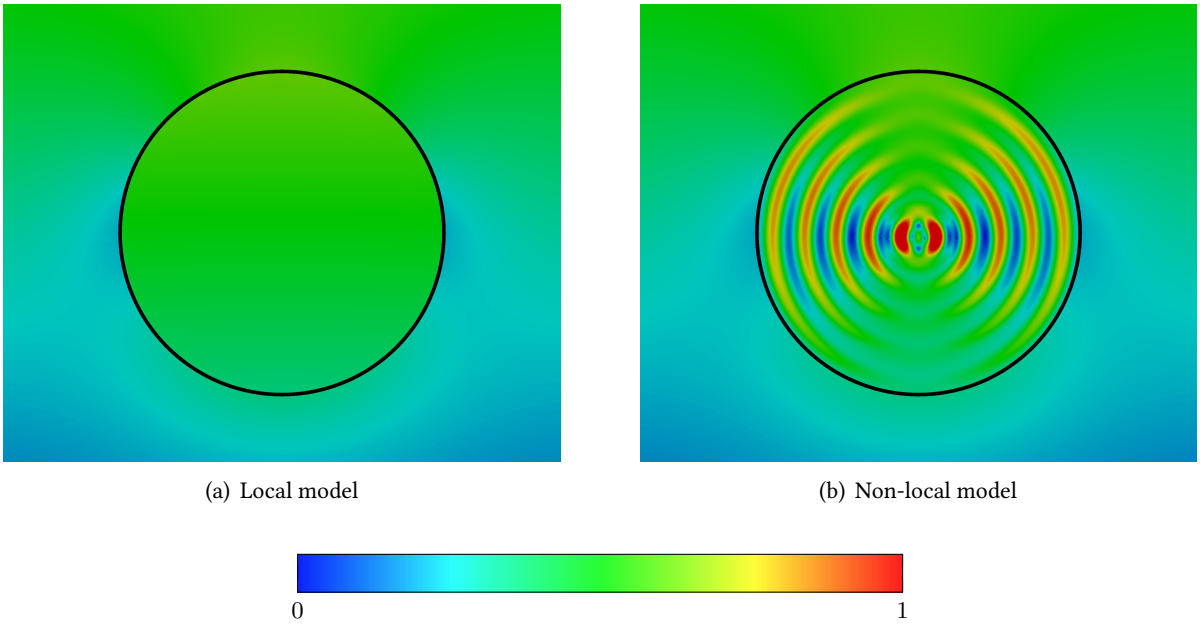
$$\varepsilon_{\text{Drude, non local}}(k,\omega) = -\frac{\omega_d^2}{\omega^2 + i\gamma_d\omega - \beta^2k^2}. \tag{2.36}$$

From a numerical point of view, the main difference with local models is that the additional equation is now a PDE. To illustrate this topic, the illumination of an infinite gold nanowire by a plane wave is considered, both with local and non-local models. The radius of the nanowire is 2 nanometers while the pulse central frequency is $f_c = 1.602\times10^4$ THz. The gold is described by a Drude model with parameters $\varepsilon_\infty = 1$, $\omega_d = 1.339\times10^4$ THz, $\gamma_d = 1.143\times10^2$ THz, and $\beta = 1.1349\times10^6$ m.s^{-1} (β is set to zero for the local model). The computations are performed with the code described in [SSL+], and the full Fourier transform of the electric field are extracted: plots for local and non-local models are presented on figure 2.11. As can be seen, in the non-local case, a bulk plasmon is excited that does not appear with the local model. Besides exciting this volume plasmon, the non-local model also has the property to blue-shift the localized surface plasmon resonance of the nanosphere. One should note that this behavior is only obtained for frequencies above the plasma frequency. Further study and implementation of the non-local model were carried out in 2D by N. Schmitt [SSL+], and the reader is referred to this publication for a more complete presentation of this phenomenon.

2.2.7 Causality principle

Whatever the chosen dispersion model, the latter has to respect the causality principle, which relies on the natural observation that any physical system should not depend on future states of the system. It can be expressed mathematically thanks to the Kramers-Krönig relations as an analyticity condition for the frequency-dependent permittivity function. Even if this characterization is well established among

[4] See [ESVM+06] [MCS13] and references therein.



(a) Local model (b) Non-local model

Figure 2.11 | Non-local resonance of a gold nanosphere at frequency $f_c = 1.602 \times 10^4$ THz. The plots show the modulus of the electric field Fourier transform. The right panel shows the excited bulk plasmon due to non-local model, which does not appear for the local model, on the left panel. Note that the situation is different when the incident frequency is below the plasma frequency (see [SSL+]). The sphere radius is 2 nm, and is described by a hydrodynamic Drude model, with parameters $\varepsilon_\infty = 1$, $\omega_d = 1.339 \times 10^4$ THz, $\gamma_d = 1.143 \times 10^2$ THz, and $\beta = 1.1349 \times 10^6$ m.s^{-1} (β is set to zero for the local model).

physicists, the justification of this condition may sometimes be quite vague. The proof of the causality of the generalized dispersive model is detailed in [LSV] and is not reproduced here. However, it is staightforward to prove that it will impose $d_l > 0$, $e_l > 0$ and $f_l > 0$.

3 The DGTD method

To go further than the few cases for which we have access to an analytical solution of Maxwell's equations, resorting to a numerical method is unavoidable. To this end, we introduce the basics of the discontinuous Galerkin (DG) method, for dielectric (section 3.1) and dispersive media (section 3.4) in three spatial dimensions. Various possible time integration techniques are presented in section 3.2. Basic validations of the method are presented in section 3.3. To remain as clear as possible, the presentation of some technical details useful to DG calculations is postponed to chapter 4. Eventually, theoretical stability and convergence results are presented (section 3.5).

3.1 DG method for Maxwell equations

3.1.1 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 $\mathcal{T}_h$ verifying $\mathcal{T}_h = \bigcup_{i=1}^{N} T_i$, where $N \in \mathbb{N}^*$ is the number of mesh elements, and $(T_i)_{i\in[\![1,N]\!]}$ the set of simplices. The internal faces of the discretization are denoted $a_{ik} = T_i \bigcap T_k$ if T_i and T_k are adjacent cells, and $\mathbf{n}_{ik}$ is defined as the unit normal vector to the face a_{ik}, oriented from T_i toward T_k. For each cell T_i, $\mathcal{V}_i$ is the set of indices $\{k \in [\![1, N]\!] \mid T_i \bigcap T_k \text{ is a trangular face}\}$. Then, the quasi-uniform assumption implies that:

$$\exists \delta, \forall T_i \in \mathcal{T}_h, \forall k \in \mathcal{V}_i, h_k \leq \delta h_i,$$

with h_i the size of element T_i. It is now possible to write the weak formulation of problem (2.11) in the cell T_i. By taking the L^2 scalar product of each term with a vector test function $\boldsymbol{\psi}$, one obtains the following variational problem:

$$\begin{aligned}
&\text{Find } (\mathbf{E}, \mathbf{H}) \in H_0(\mathbf{curl}, \Omega_h) \times H(\mathbf{curl}, \Omega_h) \text{ such that } \forall\, \boldsymbol{\psi} \in H(\mathbf{curl}, \Omega_h),\\
&\int_{T_i} \mu_r \frac{\partial \mathbf{H}}{\partial t} \cdot \boldsymbol{\psi} + \int_{T_i} \nabla \times \mathbf{E} \cdot \boldsymbol{\psi} = \mathbf{o},\\
&\int_{T_i} \varepsilon_r \frac{\partial \mathbf{E}}{\partial t} \cdot \boldsymbol{\psi} - \int_{T_i} \nabla \times \mathbf{H} \cdot \boldsymbol{\psi} = -\int_{T_i} \mathbf{J} \cdot \boldsymbol{\psi}.
\end{aligned}$$

Formally rewriting the latter equalities using classical vectorial calculus and Green formulae gives:

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

One immediatly notices that the previous equality only holds if the boundary terms exist. Considering the properties of the mixed product, the latter becomes:

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

which implies that taking $\mathbf{E}$ in $H_0(\mathbf{curl}, \Omega_h)$ requires the existence of the trace of $\boldsymbol{\psi}$ on ∂T_i. We will thus take $\boldsymbol{\psi}$ in $H^1(\Omega_h)$ instead of $H(\mathbf{curl}, \Omega_h)$. Hence, $\forall T_i, \forall \boldsymbol{\psi} \in H^1(\Omega_h)$,

Weak formulation

$$\int_{T_i} \mu_r \frac{\partial \mathbf{H}}{\partial t} \cdot \boldsymbol{\psi} + \int_{T_i} \mathbf{E} \cdot \nabla \times \boldsymbol{\psi} = \int_{\partial T_i} (\mathbf{E} \times \mathbf{n}_i) \cdot \boldsymbol{\psi},$$
$$\int_{T_i} \varepsilon_r \frac{\partial \mathbf{E}}{\partial t} \cdot \boldsymbol{\psi} - \int_{T_i} \mathbf{H} \cdot \nabla \times \boldsymbol{\psi} = -\int_{T_i} \mathbf{J} \cdot \boldsymbol{\psi} - \int_{\partial T_i} (\mathbf{H} \times \mathbf{n}_i) \cdot \boldsymbol{\psi}. \tag{3.1}$$

3.1.2 Space discretization

First, we define the following approximation space V_h:

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

where $\mathbb{P}_p(T_i)$ is the space of polynomials of maximum degree p on T_i. The semi-discrete fields, seeked 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 is the number of degrees of freedom (d.o.f.) per dimension. Additionally, to each scalar basis function, the three vectors $\boldsymbol{\phi}_{\boldsymbol{ik}}^{\boldsymbol{v}}$ are associated:

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

One now seeks the approximations $\mathbf{E}_h$ and $\mathbf{H}_h$ of $\mathbf{E}$ and $\mathbf{H}$ in space V_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:

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

that will be handy to cast the matrix-vector form of our system. In the following sections, focus is made on the **E** evolution equation. However, the **H** evolution equation is treated in the exact same way to obtain the discrete system. Hence, a discrete variational formulation can be written with the unknowns $\mathbf{E}_h$ and $\mathbf{H}_h$, analogously to (3.1). However, as will be shown in next section, the boundary terms in this formulation require some additional treatment before progressing further on in the discretization process.

3.1.3 Numerical fluxes

Given that the test functions are now allowed to be discontinuous at the interfaces between cells, it is important to notice that the surface integrals, such as:

$$\int_{a_{il}} (\mathbf{E}_h \times \mathbf{n}_{il}) \cdot \boldsymbol{\psi}, \tag{3.3}$$

and

$$\int_{a_{il}} (\mathbf{H}_h \times \mathbf{n}_{il}) \cdot \boldsymbol{\psi}, \tag{3.4}$$

are not unequivocal, since the unknows can relate to either the field value on the T_i or the T_l side of the interface. The introduction of a numerical flux allows to recover a proper definition of the latter surface integrals, and is essential to connect the field values between neighbouring cells. One must notice, however, that there is not a unique valid choice for fluxes, and that in the case of a set of linear equations, different choices can lead to stable and convergent discrete schemes. The expressions of equations (3.3) and (3.4) are therefore replaced with the following ones:

$$\int_{a_{il}} (\mathbf{E}_* \times \mathbf{n}) \cdot \boldsymbol{\psi},$$

and

$$\int_{a_{il}} (\mathbf{H}_* \times \mathbf{n}) \cdot \boldsymbol{\psi},$$

where $\mathbf{E}_*$ and $\mathbf{H}_*$ remain to be defined. As will be shown later, the flux can be seen as the solution of a Riemann problem at cell interfaces. However, in order not to interfere with the development of the DG formulation, this technical calculation is postponed to section 3.1.8. In this thesis, we will exploit two very common flux choices. The first one is the centered flux, which reads:

$$\mathbf{E}_* = \frac{\mathbf{E}_i + \mathbf{E}_l}{2}, \quad \mathbf{H}_* = \frac{\mathbf{H}_i + \mathbf{H}_l}{2}. \tag{3.5}$$

This flux is in essence non-dissipative and leads to an L^2 spatial convergence in h^p if fields are searched for in V_h. Coupled to a non-dissipative time-integration scheme such as Leap-Frog (see section 3.2.2), this choice can yield a totally non-dissipative DGTD scheme [F+05]. A weighted version of the centered flux is also commonly exploited:

$$\mathbf{E}_* = \frac{Y_i\mathbf{E}_i + Y_l\mathbf{E}_l}{Y_i + Y_l}, \quad \mathbf{H}_* = \frac{Z_i\mathbf{H}_i + Z_l\mathbf{H}_l}{Z_i + Z_l}, \tag{3.6}$$

where $Y_i = \sqrt{\frac{\varepsilon_i}{\mu_i}}$ is the admittance for cell T_i, and $Z_i = \frac{1}{Y_i} = \sqrt{\frac{\mu_i}{\varepsilon_i}}$ is its impedance. The effects of weighting on convergence are shortly explored in section 3.3.3. The second possibility is the upwind flux, which expression is given by:

$$\mathbf{E}_* = \frac{1}{Y_i + Y_l}\left(\{Y\mathbf{E}\}_{il} + \alpha\mathbf{n} \times [\![\mathbf{H}]\!]_{il}\right), \quad \mathbf{H}_* = \frac{1}{Z_i + Z_l}\left(\{Z\mathbf{H}\}_{il} - \alpha\mathbf{n} \times [\![\mathbf{E}]\!]_{il}\right), \tag{3.7}$$

where $\{A\}_{il} = A_i + A_l$ is twice the mean value of A across the interface, $[\![A]\!]_{il} = A_l - A_i$ is the jump of A across the interface, and $\alpha \in [0, 1]$ is a tunable parameter that allows to vary between the centered flux (3.6) for $\alpha = 0$, to a fully upwind flux for $\alpha = 1$. Unlike its centered counterpart, the jump term of the upwind flux introduces dissipation in the DG scheme, which can be very helpful in situations where instabilities might occur [HW08], since it helps in damping unphysical modes (see section 3.1.9). Additionaly, it leads to an L^2 spatial convergence as h^{p+1}. The convergence for intermediate values of α is assessed numerically on a simple textbook case in section 3.3.2.

3.1.4 DG matrices

Mass matrix

Test functions $\boldsymbol{\psi}$ are chosen to be the $3\,d_i$ vectors $\boldsymbol{\phi}^{\boldsymbol{v}}_{\boldsymbol{ik}}$, which constitutes the Galerkin choice:

$$\begin{aligned}
&\int_{T_i} \mu_r \frac{\partial \mathbf{H}_i}{\partial t} \cdot \boldsymbol{\phi}^{\boldsymbol{v}}_{\boldsymbol{ik}} + \int_{T_i} \mathbf{E}_i \cdot \nabla \times \boldsymbol{\phi}^{\boldsymbol{v}}_{\boldsymbol{ik}} = \sum_{l \in \mathcal{V}_i} \int_{a_{il}} (\mathbf{E}_* \times \mathbf{n}_{il}) \cdot \boldsymbol{\phi}^{\boldsymbol{v}}_{\boldsymbol{ik}}, \\
&\int_{T_i} \varepsilon_r \frac{\partial \mathbf{E}_i}{\partial t} \cdot \boldsymbol{\phi}^{\boldsymbol{v}}_{\boldsymbol{ik}} - \int_{T_i} \mathbf{H}_i \cdot \nabla \times \boldsymbol{\phi}^{\boldsymbol{v}}_{\boldsymbol{ik}} = -\sum_{l \in \mathcal{V}_i} \int_{a_{il}} (\mathbf{H}_* \times \mathbf{n}_{il}) \cdot \boldsymbol{\phi}^{\boldsymbol{v}}_{\boldsymbol{ik}} - \int_{T_i} \mathbf{J}_i \cdot \boldsymbol{\phi}^{\boldsymbol{v}}_{\boldsymbol{ik}}.
\end{aligned} \tag{3.8}$$

In the remaining of this manuscript, the indices present in formulations such as (3.8) are defined over the following sets: $i \in [\![1, N]\!]$, $k \in [\![1, d_i]\!]$ and $v \in \{x, y, z\}$. One then exploits the local field expansions from (3.2). For the first component of the time-derivative term of equation (3.1), this yields:

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

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 matrix

Focus is now made on the curl integral of the equality. The first component of the **E** evolutionary equation is:

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

Here, the three stiffness matrices were introduced:

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

with $(j,k) \in [\![1, d_i]\!]^2$. From the latter definition, we define the general $3d_i \times d_i$ stiffness matrix that will be used in 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

The last part to consider is the surface integral that contains the flux contribution. The calculation is here made with the centered flux (3.5), but the generalization to other fluxes is straightforward. We also note that in the conforming case, expanding a field defined on a_{il} over the basis functions of T_i or T_l is equivalent if the same basis expansions are used. In the non-conforming case, one should consider the respective expansions of the field over the respective bases of the two cells (see chapter 6). We proceed as we did previously, focusing on the x component of the flux:

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

where the flux matrices are, in the conforming case:

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

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

General matrix-vector formulation

It is necessary to define extended mass and flux matrices in order to write the semi-discrete formulation in a compact manner:

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

These definitions lead to the following compact expression of the semi-discrete DG scheme for Maxwell's equations:

Semi-discrete scheme

$$\begin{aligned}
\overline{\mathbb{M}}_i^{\mu_r} \frac{\partial \overline{\mathbf{H}}_i}{\partial t} &= -\overline{\mathbb{K}}_i \times \overline{\mathbf{E}}_i + \sum_{l \in \mathcal{V}_i} \overline{\mathbb{S}}_{il} \left(\overline{\mathbf{E}}_* \times \mathbf{n}_{il} \right), \\
\overline{\mathbb{M}}_i^{\varepsilon_r} \frac{\partial \overline{\mathbf{E}}_i}{\partial t} &= \quad \overline{\mathbb{K}}_i \times \overline{\mathbf{H}}_i - \sum_{l \in \mathcal{V}_i} \overline{\mathbb{S}}_{il} \left(\overline{\mathbf{H}}_* \times \mathbf{n}_{il} \right) - \overline{\mathbb{M}}_i \bar{\mathbf{J}}_i.
\end{aligned} \tag{3.9}$$

3.1.5 Mapping from a reference element

As a finite element method, a strength of the DG method is that the FE matrices are not stored, but are calculated once on a reference element $\widehat{T}$, 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:

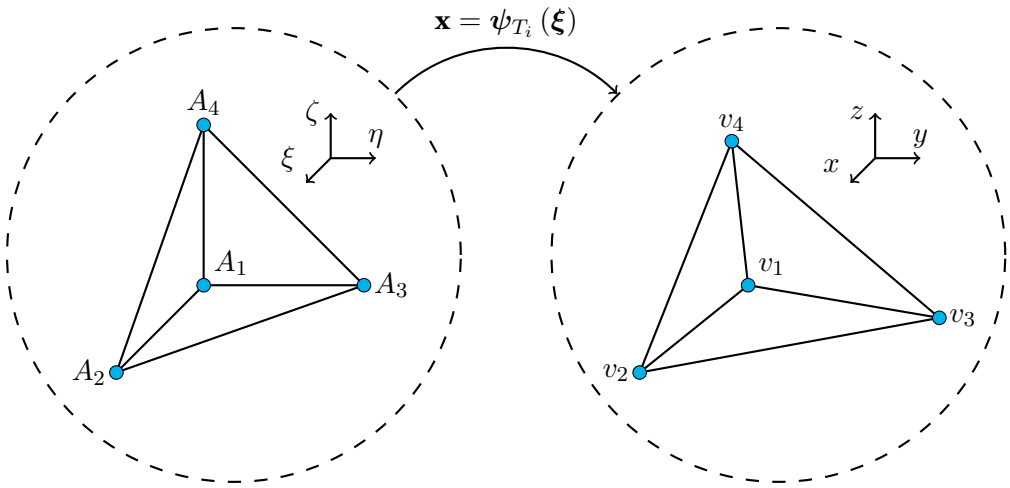


Figure 3.1 | Linear mapping from the reference element $\widehat{T}$ to the physical element T_i.

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

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

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

A visual representation can be found on figure 3.1. The vertices of $\widehat{T}$ are noted (A_1, A_2, A_3, A_4), whereas the vertices of T_i are (v_1, v_2, v_3, v_4). In this case, the mapping is a linear combination of ξ, η and ζ:

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

Let us see how the finite element matrices, calculated on the reference element, are then mapped to the physical tetrahedra. 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, the mass matrix on the element T_i will be defined 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:

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

Its determinant $\left|\mathbf{J}_{\boldsymbol{\psi}_{T_i}}\right|$ happens here to be a constant, depending only of the coordinates of the physical vertices (v_1, v_2, v_3, v_4). Hence, in the case of a linear mapping, the mass matrix for each physical tetrahedron is simply a multiple of the mass matrix calculated on the reference tetrahedron:

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

A similar situation occurs for the stiffness and flux matrices, through the following change of variables (see [Mono3] for additional details):

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

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 precalculated matrices $\left(\widehat{\mathbb{K}}^m\right)_{jk}$ on the reference element. In the same fashion, the surface matrix can be rewritten as:

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

where $\widehat{t}$ denotes the reference triangle. As well, these matrices can be deduced from the precalculated $\left(\widehat{\mathbb{S}}\right)_{jk}$. Finally, the unitary normals 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, a single (3×3) matrix needs to be stored for each cell, in the stead of the whole set of FE matrices.

3.1.6 Polynomial expansion basis

Although the derivation of the semi-discrete scheme is achieved, the basis functions of the reference element $\left(\widehat{\phi}_j\right)_{j=1..d_i}$ remain to be specified. Although many polynomial bases are available, Lagrange polynomials are quite a common choice. They can be defined by a set of interpolation nodes spread across the element, which constitutes another free parameter of the DG method. At first, the most natural choice seems to use equispaced interpolation points (see figure 3.2 for an illustration of their distribution on triangles and tetrahedra). However, for high-order polynomials, this kind of interpolation is known to be

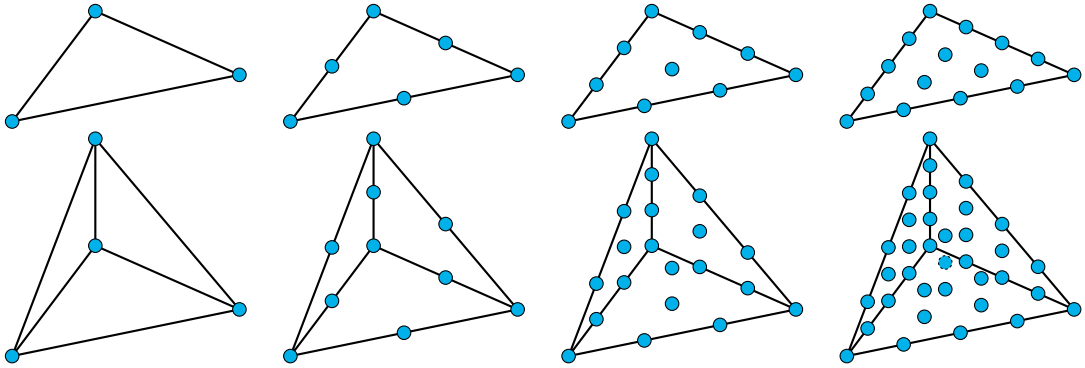


Figure 3.2 | Equispaced Lagrange nodes on triangles and tetrahedra for orders ranging from 1 to 4. For tetrahedra, dashed lines indicate that the node is located inside the volume.

ill-conditioned [HW08]. In this case, more suitable choices exist, such as the Warp & Blend interpolation sets [Waro6]. Nevertheless, as will be shown later, fourth-order approximation is rarely exceeded in practical computations. When referring to [Waro6], one sees that the improvement of the condition numbers with the Warp & Blend interpolation for orders lower than five is not so clear when compared to equispaced interpolation (resp. 2.11 against 2.27 for order 3, and 2.66 against 3.47 for order 4). For this reason, equispaced Lagrange node distributions will be used throughout the remaining of this manuscript. The Lagrange interpolants L_i are defined by the following property:

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

Hence, there must be an equal number of polynomials and nodes to actually define a complete basis. In a tetrahedron, for a polynomial order p, the number of Lagrange nodes in the volume is equal to:

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

while on each tetrahedron face, the number of Lagrange nodes is:

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

Exploiting Lagrange polynomials with equispaced nodes allows a very simple integration of the finite element matrices on the reference element, since the nodes positions are known accurately. On figure 3.3, the six second-order Lagrange polynomials on the reference triangle are presented. Amplitudes are magnified for a better visualization.

3.1.7 Boundary conditions

Every computational setup must be terminated by boundary conditions, which are chosen depending on the physics of the problem. Electromagnetic cavity problems are usually terminated with perfect electric conductor (PEC) conditions, which will perfectly reflect the incident waves. For open problems, it is necessary to artificially truncate the considered physical domain, since computational domains cannot describe unbounded volumes. Another possibility is to use periodic boundary conditions (PBC), which describe an infinite repetition of the same pattern.

In the case of DGTD methods, boundary conditions can be imposed by adding an extra layer of ghost cells outside the computational domain (see figure 3.4). By enforcing specific values of the fields in these

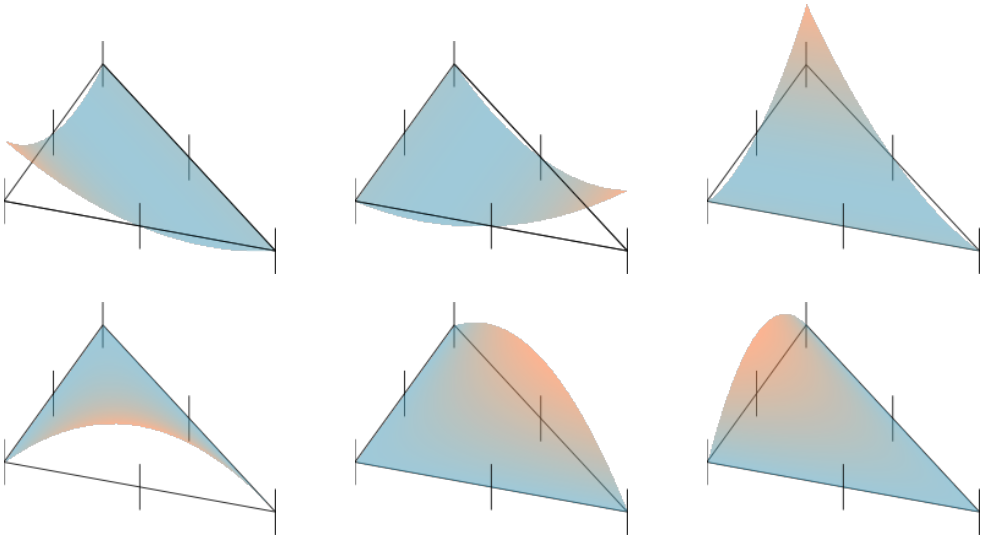


Figure 3.3 | Second order Lagrange polynomials on triangles. Amplitudes are magnified by a factor of 4 for visibility.

cells, the behavior of the solution on the boundary is naturally controlled *via* the numerical flux. Here, field inside the ghost cells are noted $\mathbf{E}_{gc}$ and $\mathbf{H}_{gc}$, while $\mathbf{E}_{bc}$ and $\mathbf{H}_{bc}$ denote the fields in the boundary cells.

Perfect electric conductor condition

PEC condition was previously defined (see system (2.18)). To impose a zero tangential electric field on a PEC boundary, one can simply enforce the field values in the ghost cells as follows:

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

Perfect magnetic conductor condition

PMC condition is the reciprocal of the PEC one, and is often used to impose symmetry planes. A zero tangential magnetic field is enforced by setting:

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

Absorbing boundary condition

Absorbing boundary conditions (ABC) are a family of boundary conditions that allow to partially absorb fields radiating out from the physical domain. There exist many forms, the most common being the first-order Silver-Müller boundary condition [Mono3]:

Silver-Müller boundary condition

$$\begin{aligned} \mathbf{n} \times (\mathbf{E} + Z\,(\mathbf{n} \times \mathbf{H})) &= \mathbf{o} \\ \mathbf{n} \times (\mathbf{H} - Y\,(\mathbf{n} \times \mathbf{E})) &= \mathbf{o} \end{aligned} \tag{3.10}$$

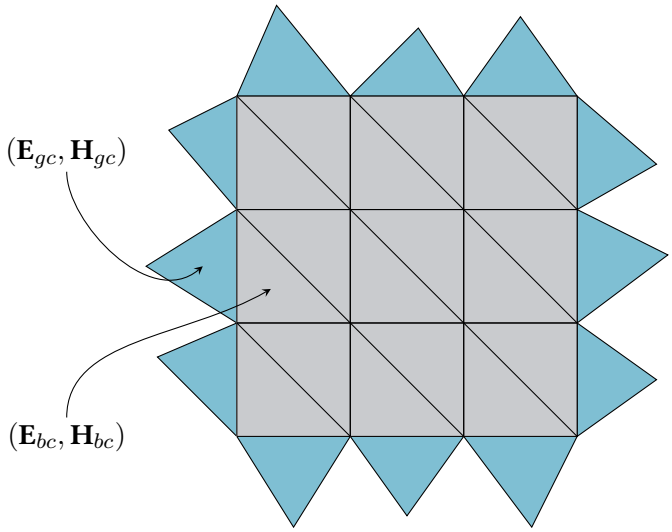


Figure 3.4 | Ghost cells layer on the computational domain boundary.

When imposed on the boundary, this condition perfectly absorbs normally-incident plane waves. However, its performance rapidly decreases when waves are incident at increasing angles (the reader can refer to section 4.1.1 for technical discussions and numerical tests). Imposing this condition is equivalent to setting the incoming flux to zero on the boundary. Its expression depends on the upwinding factor α:

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

Periodic boundary condition

Periodic boundary conditions (PBC) allow to simulate artificially infinite mono-directional or bi-directional arrays while considering only one elementary pattern. To do so, cells from a periodic boundary face are matched with their neighbors on the opposite boundary of the domain. This way, every cell has a well-defined neighbor, and standard fluxes can be applied.

3.1.8 Derivation of the flux formulation

This part is dedicated to the derivation of the flux formulation (3.7). This is done *via* the resolution of a Riemann problem on a cell interface. First, the source-free Maxwell's equations are cast under a conservative form. Then, the Rankine-Hugoniot relations are applied, which leads to expressions (3.7).

Conservative form

Let us recall the Maxwell's normalized equations, for the special case of source-free regions:

$$\begin{aligned} \frac{\partial \mathbf{H}}{\partial t} &= -\frac{1}{\mu_r} \nabla \times \mathbf{E}, \\ \frac{\partial \mathbf{E}}{\partial t} &= \quad \frac{1}{\varepsilon_r} \nabla \times \mathbf{H}. \end{aligned}$$

Both electric and magnetic fields are cast into a unique solution vector $\mathbf{W}$, which therefore holds six components. A 6x6 material matrix is also defined:

$$\mathbf{W} = \begin{bmatrix} H_x \\ H_y \\ H_z \\ E_x \\ E_y \\ E_z \end{bmatrix}, \quad \mathbb{Q} = \left[\text{diag}\left(\mu_r, \mu_r, \mu_r, \varepsilon_r, \varepsilon_r, \varepsilon_r\right)\right].$$

Then, three vector functions $\mathbf{F}_x$, $\mathbf{F}_y$ and $\mathbf{F}_z$ are defined:

$$\mathbf{F}_x\left(\mathbf{W}\right) = \begin{bmatrix} 0 \\ -E_z \\ E_y \\ 0 \\ H_z \\ -H_y \end{bmatrix}, \quad \mathbf{F}_y\left(\mathbf{W}\right) = \begin{bmatrix} E_z \\ 0 \\ -E_x \\ -H_z \\ 0 \\ H_x \end{bmatrix}, \quad \text{and} \quad \mathbf{F}_z\left(\mathbf{W}\right) = \begin{bmatrix} -E_y \\ E_x \\ 0 \\ H_y \\ -H_x \\ 0 \end{bmatrix}.$$

One easily sees that the two curl equations can be summed up in the following formulation:

$$\mathbb{Q}\frac{\partial \mathbf{W}}{\partial t} + \frac{\partial \mathbf{F}_x}{\partial x} + \frac{\partial \mathbf{F}_y}{\partial y} + \frac{\partial \mathbf{F}_z}{\partial z} = \mathbf{0}.$$

A "vector of vectors" is introduced with $\mathbf{F} = {}^{T}\left[\mathbf{F}_x, \mathbf{F}_y, \mathbf{F}_z\right]$. By extending the definition of the divergence $(\nabla\cdot)$ to a vector of vectors, the system can be rewritten in the following conservative form:

Conservative form

$$\mathbb{Q}\frac{\partial \mathbf{W}}{\partial t} + \nabla \cdot \mathbf{F}\left(\mathbf{W}\right) = \mathbf{0}. \tag{3.11}$$

Riemann problem

In its simplest form, a Riemann problem is composed of a conservation equation like (3.11), along with a discontinuous, piecewise initial condition. As will be shown in this section, the solution of such a problem is of particular interest in the DG framework because of the intrinsic discontinuities of the fields at cells interfaces. Expression (3.11) is reformulated in a matrix form:

$$\mathbb{Q}\frac{\partial \mathbf{W}}{\partial t} + \mathbb{D}_x\frac{\partial \mathbf{W}}{\partial x} + \mathbb{D}_y\frac{\partial \mathbf{W}}{\partial y} + \mathbb{D}_z\frac{\partial \mathbf{W}}{\partial z} = \mathbf{0}.$$

Since solutions are sought along the unit normal $\mathbf{n}$, the following operator is introduced:

$$\mathbb{D} = \sum_{i\in\{x,y,z\}} n_i\mathbb{D}_i,$$

where n_i is the i^{th} component of the unit normal $\mathbf{n}$. Hence, whatever the dimension of the original problem, the resulting problem along the normal direction is a 1D problem. The dynamics of this system are given by the eigenvalues of $\mathbb{Q}^{-1}\mathbb{D}$. Three repeated eigenvalues are obtained, that define the possible velocities of waves propagating in each medium:

$$\lambda_1^{\pm} = c^{\pm}, \quad \lambda_2^{\pm} = 0, \quad \lambda_3^{\pm} = -c^{\pm} \quad \text{with} \quad c^{\pm} = \frac{1}{\sqrt{\mu^{\pm}\varepsilon^{\pm}}}$$

The structure of the 1D Riemann problem (RP) is presented on figure 3.5. Figure 3.5(a) depicts the smooth fields in cells on each side of the interface. On figure 3.5(b), the situation is considered along the normal direction. This constitutes a generalized Riemann problem (GRP), *i.e.* fields are not constant on each side of the discontinuity. A method for solving such problems is presented in great details in [Toro09]. However, the full solving of a GRP is more the prerogative of ADER methods [TDMS09], where the flux computation provides a high-order approximation in space and time simultaneously. Here, only the leading term of the GRP is considered. The leading term $\mathbf{W}_l$ is defined as the limit of $\mathbf{W}$ on the interface:

$$\begin{aligned} \mathbf{W}_l^- &= \lim_{\mathbf{x}\to 0^-} \mathbf{W}^-(\mathbf{x}), \\ \mathbf{W}_l^+ &= \lim_{\mathbf{x}\to 0^+} \mathbf{W}^+(\mathbf{x}). \end{aligned}$$

In the DG framework, these definitions clearly find their numerical counterparts, since nodes on a boundary exist separately on both sides. Close to the boundary, it is therefore possible to consider the limit RP of figure 3.5(c). This constitutes a standard Riemann problem, which can be solved with the method of characteristics (figure 3.5(d)). The solution contains four zones of constant values: the first two where the fields are known and equal to $\mathbf{W}_l^-$ and $\mathbf{W}_l^+$; the last two, where the fields are unknown and equal to $\mathbf{W}_*^-$ and $\mathbf{W}_*^+$.

Rankine-Hugoniot relations

To complete the resolution of the RP, it is necessary to apply the Rankine-Hugoniot relations at the jumps between the four zones [Toro09]. Given that there are three distinct eigenvalues, these conditions yield the following equations, where, for the sake of simplicity, the $\mathbf{W}^l$ notation is dropped:

$$c^-\mathbb{Q}^-\left(\mathbf{W}_*^- - \mathbf{W}^-\right) + \mathbf{n}\cdot\left(\mathbf{F}_*^- - \mathbf{F}^-\right) = \mathbf{o} \tag{3.12}$$
$$\mathbf{n}\cdot\left(\mathbf{F}_*^- - \mathbf{F}_*^+\right) = \mathbf{o} \tag{3.13}$$
$$-c^+\mathbb{Q}^+\left(\mathbf{W}_*^+ - \mathbf{W}^+\right) + \mathbf{n}\cdot\left(\mathbf{F}_*^+ - \mathbf{F}^+\right) = \mathbf{o} \tag{3.14}$$

First, (3.13) and (3.14) are combined:

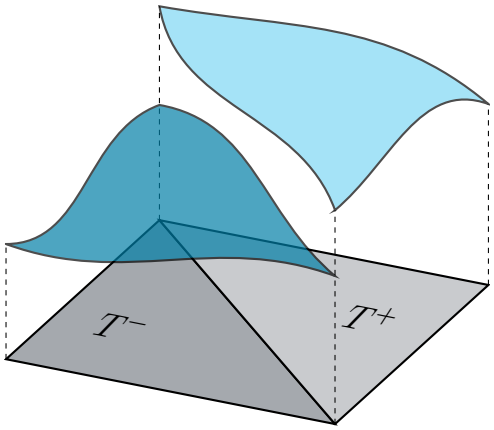
$$c^-\mathbb{Q}^-\left(\mathbf{W}_*^- - \mathbf{W}^-\right) + \mathbf{n}\cdot\left(\mathbf{F}_*^- - \mathbf{F}^-\right) = \mathbf{o} \tag{3.15}$$
$$-c^+\mathbb{Q}^+\left(\mathbf{W}_*^+ - \mathbf{W}^+\right) + \mathbf{n}\cdot\left(\mathbf{F}_*^- - \mathbf{F}^+\right) = \mathbf{o}. \tag{3.16}$$

Then, evaluating $c^+\mathbb{Q}^+(3.15) + c^-\mathbb{Q}^-(3.16)$ yields:

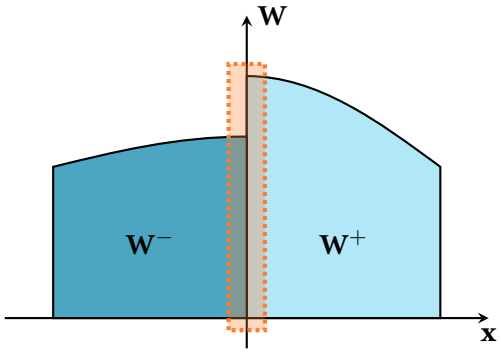
$$c^+c^-\mathbb{Q}^+\mathbb{Q}^-\left(\mathbf{W}_*^- - \mathbf{W}^- - \mathbf{W}_*^+ + \mathbf{W}^+\right) + c^+\mathbb{Q}^+\mathbf{n}\cdot\left(\mathbf{F}_*^- - \mathbf{F}^-\right) + c^-\mathbb{Q}^-\mathbf{n}\cdot\left(\mathbf{F}_*^- - \mathbf{F}^+\right) = \mathbf{o}$$

with:

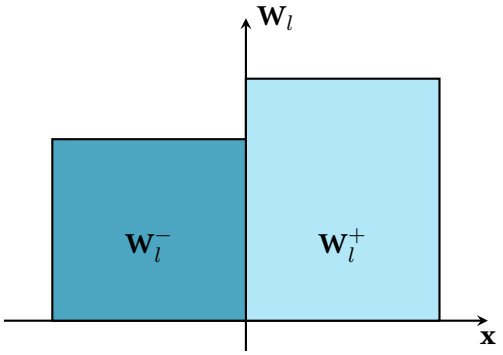
$$\mathbf{n}\cdot\mathbf{F} = \begin{bmatrix} \mathbf{n}\times\mathbf{E} \\ -\mathbf{n}\times\mathbf{H} \end{bmatrix}.$$



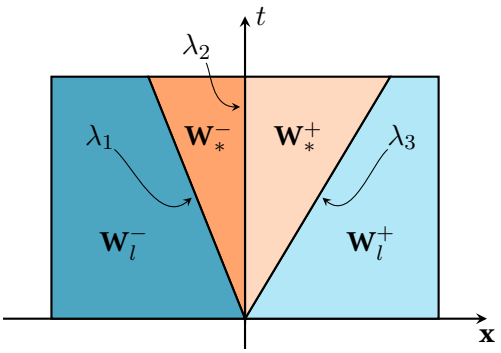
(a) Discontinuous solution at cell interface



(b) Generalized Riemann problem along face normal at initial state



(c) Leading term Riemann problem along face normal at initial state



(d) Solution of the leading term Riemann problem

Figure 3.5 | Structure of the Riemann problem at cells interfaces. In 3.5(a), the smooth solutions are shown in each cell, with the discontinuity occurring at the cells interface. When considered in the direction of the normal to the interface, this constitutes a generalized Riemann problem, as shown in 3.5(b). The infinitesimally small volume close to the discontinuity, represented in orange, corresponds to the leading term Riemann problem, and is reproduced in 3.5(c). In 3.5(d), its solution is shown in the $(\mathbf{x} - t)$ space.

In the following, focus is made on the second component of the flux. However, very similar treatment can be applied to the first one. Rewriting the previous equality one obtains:

$$\begin{aligned} c^+c^-\varepsilon^+\varepsilon^-\left(\mathbf{E}_*^- - \mathbf{E}_*^+ + \mathbf{E}^+ - \mathbf{E}^-\right) - \left(c^+\varepsilon^+ + c^-\varepsilon^-\right)\mathbf{n}\times\mathbf{H}_*^- \\ + c^+\varepsilon^+\mathbf{n}\times\mathbf{H}^- + c^-\varepsilon^-\mathbf{n}\times\mathbf{H}^+ = \mathbf{o}. \end{aligned} \tag{3.17}$$

Dot multiplying (3.17) with $\mathbf{n}$, one sees that:

$$\mathbf{n}\cdot\left(\mathbf{E}_*^- - \mathbf{E}_*^+\right) = \mathbf{n}\cdot\left(\mathbf{E}^- - \mathbf{E}^+\right).$$

Additionally, by (3.13):

$$\mathbf{n}\times\left(\mathbf{E}_*^- - \mathbf{E}_*^+\right) = \mathbf{o}.$$

Now, consider the following equality:

$$\mathbf{U} = (\mathbf{n}\cdot\mathbf{U})\,\mathbf{n} - \mathbf{n}\times(\mathbf{n}\times\mathbf{U}), \tag{3.18}$$

and applying it to $\left(\mathbf{E}_*^- - \mathbf{E}_*^+\right)$ gives, considering the previous equalities:

$$\mathbf{E}_*^- - \mathbf{E}_*^+ = \left(\mathbf{n}\cdot\left(\mathbf{E}_*^- - \mathbf{E}_*^+\right)\right)\mathbf{n} - \mathbf{n}\times\left(\mathbf{n}\times\left(\mathbf{E}_*^- - \mathbf{E}_*^+\right)\right),$$

which simplifies as:

$$\mathbf{E}_*^- - \mathbf{E}_*^+ = \left(\mathbf{n}\cdot\left(\mathbf{E}^- - \mathbf{E}^+\right)\right)\mathbf{n}. \tag{3.19}$$

Equality (3.18) applied to $\left(\mathbf{E}^- - \mathbf{E}^+\right)$ yields:

$$\mathbf{E}^- - \mathbf{E}^+ = \left(\mathbf{n}\cdot\left(\mathbf{E}^- - \mathbf{E}^+\right)\right)\mathbf{n} - \mathbf{n}\times\left(\mathbf{n}\times\left(\mathbf{E}^- - \mathbf{E}^+\right)\right),$$

which, combined with (3.19), eventually gives:

$$\mathbf{E}_*^- - \mathbf{E}_*^+ = \mathbf{E}^- - \mathbf{E}^+ + \mathbf{n}\times\left(\mathbf{n}\times\left(\mathbf{E}^- - \mathbf{E}^+\right)\right). \tag{3.20}$$

Hence, the numerical flux for the evolution equation on $\mathbf{E}$ can be written as:

$$-\mathbf{n}\times\mathbf{H}_*^- = \frac{1}{Z^+ + Z^-}\left(-\mathbf{n}\times\{Z\mathbf{H}\}_{+-} + \mathbf{n}\times(\mathbf{n}\times[\![\mathbf{E}]\!]_{+-})\right), \tag{3.21}$$

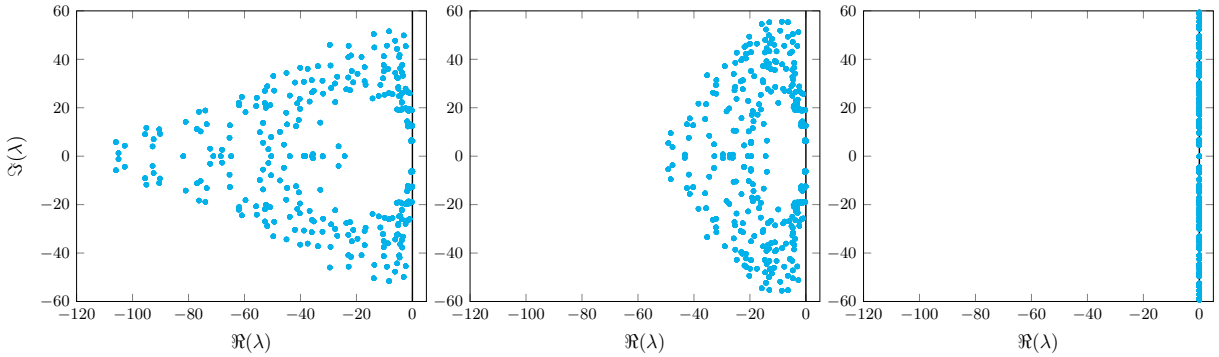
It appears here that the fully upwind flux ($\alpha = 1$) is the exact solution of the lead term RP, while partially penalized and centered fluxes ($\alpha \in [0, 1[$) are only approximate solutions.

3.1.9 Spectrum of the DG operator

In the same way as for continuous FE methods, the global matrix of the spatial DG operator can be assembled. The difference to the FE case is that there will be no overlap of local matrices in the global matrix, since at each cells interface the degrees of freedom exist independently on both sides. Eventually, the semi-discrete DG problem can be rewritten under the following generic form:

$$\frac{\partial\mathbf{W}_h}{\partial t}(t) = \mathbb{A}_h\mathbf{W}_h(t) + \mathbf{B}_h(t), \tag{3.22}$$

which is a global form, *e.g.* it includes every cell in the mesh. In (3.22), $\mathbf{B}_h(t)$ represents the exterior sources, while the matrix $\mathbb{A}_h$ holds all the information relative to the spatial discretization. As will be



(a) $\alpha = 1$ (b) $\alpha = 0.5$ (c) $\alpha = 0$

Figure 3.6 | Eigenvalues of the discrete DG operator with PBCs for various upwinding factors. The considered case is a cubic cavity made of 162 tetrahedra.

seen in next section, there is no particular reason for $\mathbb{A}_h$ to be diagonalizable in the general case. Still, it remains instructive to give a look at its eigenspectrum: as an example, we compute the eigenvalues of $\mathbb{A}_h$ on a $[0, 1]^3$ cubic domain meshed with 162 tetrahedra with PBC on all edges. $\mathbb{P}_3$ polynomial expansion is used, for various upwinding factors (see figure 3.6). As can be seen, for $\alpha = 0$ the DG spectrum is entirely located on the imaginary axis of the complex plane, yielding a non-dissipative and stable formulation. However, as noted in [HW08], the spectrum of the discretized DG operator includes both physical and spurious eigenvalues, which implies here that although the spurious modes will not grow in time, they will not be damped. For increasing values of α, the situation is different, since a large part of the spurious eigenvalues corresponding to the spatial discretization have a negative real part, meaning that these modes will be rapidly damped. As will be seen in next section, the shape of the DG eigenspectrum has a crucial importance in the choice and efficiency of the timestepping algorithm.

3.2 Time discretization

In the previous section, the semi-discrete scheme (3.9) was obtained by discretizing the spatial derivatives contained in Maxwell's equations. We now turn to the design of a time discretization. For the study of the different time schemes, we consider the reduced problem of the 1D Maxwell's equations:

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

The semi-discrete formulation associated to this system is (the spatial subscripts of the unknowns are dropped):

$$\begin{aligned}\mathbb{M}_i^{\mu_r} \frac{\partial H_i}{\partial t} &= \mathbb{K}_i E_i + [E_*]_{x_{i-1}}^{x_i},\\ \mathbb{M}_i^{\varepsilon_r} \frac{\partial E_i}{\partial t} &= \mathbb{K}_i H_i + [H_*]_{x_{i-1}}^{x_i} - \mathbb{M}_i J_i,\end{aligned} \tag{3.23}$$

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 indifferently designates $i-1$ or $i+1$. When trying to put system (3.23) under the form (3.22), one obtains, in the general case, a non-symmetric $\mathbb{A}$ matrix:

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

The off-diagonal blocks represent the stiffness part, as well as the centered part of the flux, while the diagonal blocks account for the upwind contribution. Hence, in the case of centered fluxes, $\mathbb{A}$ is purely anti-diagonal. One should note that, in the case of an homogeneous medium ($Y_i = Y_l = Y$), we obtain the following properties:

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

For the sake of simplicity, we consider the simple case of vacuum ($\varepsilon_r = \mu_r = 1$), which leads to a symmetric $\mathbb{A}$ matrix, and therefore to a diagonalizable system. The main consequence is that, as a first step for the study of the timestepping schemes, it is sufficient to retain the corresponding formulation in the diagonalized basis, which reduces to a system of ODEs of the form:

$$\frac{\partial \phi}{\partial t}(t) = \lambda\phi(t) + b(t) \equiv f(t, \phi(t)), \tag{3.24}$$

for each λ eigenvalue of $\mathbb{A}$. A large panel of time-integration techniques coming from the ODE community are suited to solve (3.24). In the following, a short overview of possible methods is presented. Two particular techniques, the Leap-Frog (LF) method and the Low-Storage Runge-Kutta (LSRK) method, are then explored in more details. For both of them, fully-discrete schemes are derived, providing two Discontinuous Galerkin Time-Domain (DGTD) algorithms. Eventually, the last section deals with appropriate timestep choices.

3.2.1 A quick overview of time integration schemes

Timestepping methods are generally classified in two main categories. The first one gathers the explicit time integration techniques, for which the time state $\phi(t+\Delta t)$ is computed explicitely from $\phi(t)$, and possibly other previous time stations. The second category regroups the implicit methods: in this case, the time-updated solution is obtained by solving an implicit expression of the form $g(\phi(t), \phi(t+\Delta t)) = 0$. Numerically, this implies the resolution of a linear system of equations at each timestep, which appears to be much more expensive than explicit techniques. However, most implicit methods are unconditionnaly stable, which means that any choice of Δt will lead to a stable algorithm. In this case, the only limit to impose on Δt is deduced from the characteristics of the physical phenomenon being simulated. For explicit methods, a numerical criterion, the CFL condition (see section 3.2.5), must be fullfiled, without which the resulting algorithm will be unstable and blow up. Hence, explicit timestepping usually requires much more timesteps than implicit methods, but each timestep requires a considerably lower computational effort. In the remaining of this manuscript, solutions are seeked on intervals of the form $[0, T]$ with $T > 0$, and this interval is discretized in timesteps of length Δt. The notation ϕ^n is used to designate the discrete approximation of $\phi(t_n)$, with $t_n = n\Delta t$.

Regarding DG methods, a word can be said about space-time DG methods [PFT00] [AAPG14], in which time is treated in the same way as space, thus allowing discontinuities between temporal slabs. The major drawback of these methods is that they usually result in an implicit scheme.

3.2.2 Leap-Frog schemes

The LF2 scheme

Leap-Frog algorithms are a class of multi-level algorithms, which means they make use of more than one known value of the solution to evaluate the next step. For ϕ smooth enough, consider the following Taylor expansions:

$$\phi(t+\Delta t) = \phi(t) + \Delta t \frac{\partial \phi}{\partial t}(t) + \frac{\Delta t^2}{2}\frac{\partial^2 \phi}{\partial t^2}(t) + \frac{\Delta t^3}{6}\frac{\partial^3 \phi}{\partial t^3}(t) + \frac{\Delta t^4}{24}\frac{\partial^4 \phi}{\partial t^4}(t) + O\left(\Delta t^5\right), \tag{3.25}$$

$$\phi(t-\Delta t) = \phi(t) - \Delta t \frac{\partial \phi}{\partial t}(t) + \frac{\Delta t^2}{2}\frac{\partial^2 \phi}{\partial t^2}(t) - \frac{\Delta t^3}{6}\frac{\partial^3 \phi}{\partial t^3}(t) + \frac{\Delta t^4}{24}\frac{\partial^4 \phi}{\partial t^4}(t) + O\left(\Delta t^5\right), \tag{3.26}$$

Substracting (3.26) to (3.25) yields:

$$\phi(t+\Delta t) = \phi(t-\Delta t) + 2\Delta t \frac{\partial \phi}{\partial t}(t) + O\left(\Delta t^3\right). \tag{3.27}$$

Considering (3.24), this is equivalent to:

$$\phi(t+\Delta t) = \phi(t-\Delta t) + 2\Delta t f(t, \phi(t)) + O\left(\Delta t^3\right). \tag{3.28}$$

Hence, the second-order accurate LF scheme simply writes as:

LF2 scheme

$$\phi^{n+1} = \phi^{n-1} + 2\Delta t f\left(t_n, \phi^n\right). \tag{3.29}$$

Equation (3.29) expresses that the LF2 scheme approximates the first-order time-derivative of ϕ with a centered finite difference:

$$\frac{\partial \phi}{\partial t}(t)\Big)_{2^{nd}\text{ order}} = \frac{\phi(t+\Delta t)-\phi(t-\Delta t)}{2\Delta t} = \frac{\partial \phi}{\partial t}(t) + \frac{\Delta t^2}{3}\frac{\partial^3 \phi}{\partial t^3}(t) + O\left(\Delta t^4\right) \tag{3.30}$$

Building high-order LF schemes

There are different ways to build higher-order schemes. A first way consists in evaluating the first-order derivative with richer combinations of $\phi(t \pm k\Delta t)$, in order to successively eliminate the high-order residual derivatives in (3.29). An example of fourth-order scheme reads:

$$\begin{aligned}\frac{\partial \phi}{\partial t}(t)\Big)_{4^{th}\text{ order}} &= \frac{-\phi(t+2\Delta t)+8\phi(t+\Delta t)-8\phi(t-\Delta t)+\phi(t-2\Delta t)}{12\Delta t} \\ &= \frac{\partial \phi}{\partial t}(t) - \frac{\Delta t^4}{30}\frac{\partial^5 \phi}{\partial t^5}(t) + O\left(\Delta t^6\right)\end{aligned} \tag{3.31}$$

Although scheme (3.31) has the required accuracy, the full storage of multiple previous timestep solutions is not a desirable feature. Additionally, initiating the timestepping requires three initial conditions, two of which must be computed with self-starting schemes. A friendlier solution is obtained by finding an approximation of the first residual term of (3.29) (*i.e.* the third-order derivative of ϕ) *via* successive convolutions of operator f. This procedure is valid in the case where $b \equiv 0$ and ϕ is sufficiently regular:

$$f^3\left(\phi(t)\right) = f\left(f\left(f\left(\phi(t)\right)\right)\right) = \frac{\partial^3 \phi}{\partial t^3}(t) \quad \text{for} \quad b \equiv 0.$$

However, this is obviously not the case in the presence of a source term, because of the chain derivation rule. One can prove that the following LF$_4$ scheme is only fourth-order accurate in the absence of external sources:

LF$_4$ scheme

$$\phi^{n+1} = \phi^{n-1} + 2\Delta t f\left(\phi^n\right) + \frac{\Delta t^3}{3} f^3\left(\phi^n\right). \tag{3.32}$$

The scheme (3.32) does not require additional storage when compared to LF2 (3.29). The higher accuracy, in the linear source-free case, is obtained at the cost of two additional matrix-vector multiplications. In a more general way, LF schemes of arbitrarily high orders (hereafter denoted LFp schemes) can be built in the following fashion for even values of $p \in \mathbb{N}$:

$$\phi^{n+1} = \phi^{n-1} + \sum_{\substack{0<k<p \\ k \text{ odd}}} 2\frac{\Delta t^k}{k!} f^k\left(\phi^n\right). \tag{3.33}$$

Properties of the LFp schemes

The properties of time schemes can be studied by using monochromatic solutions $\phi^n = \mathrm{e}^{i\omega n\Delta t}$. A general procedure consists in considering the amplification factor A that links the values of ϕ^n at two successive timesteps:

$$\phi^{n+1} = A\phi^n. \tag{3.34}$$

Considering the monochromatic solution, one can plug (3.34) in (3.29) and then use the definition of f to obtain, for the LF2 scheme:

$$A^2 - 2i\Delta t\omega A - 1 = 0,$$

which solutions are:

$$A_\pm = i\tilde{\omega} \pm \sqrt{1-\tilde{\omega}^2}, \quad \text{with} \quad \tilde{\omega} \equiv \Delta t\omega.$$

A_+ is called physical mode, *i.e.* the approximation of the solution of the original ODE, while A_- is the computational mode, *i.e.* a parasitic mode that arises from the numerical procedure. The computational mode originates from the need of two initial conditions for the LF scheme to start, while the original differential equation only requires one [Str04]. For $\{|\tilde{\omega}| < 1, \tilde{\omega} \in \mathbb{R}\}$, it is easy to see that $|A_\pm| = 1$, which corresponds to the well-known non-dissipative property of LF schemes: the eigenvalues located on the imaginary axis are neither damped nor amplified. Although this is a good property for the computational mode, it also means that the parasitic mode is undamped. If $|\tilde{\omega}| > 1$, then $|A_+| > 1$, meaning the scheme is unstable. When looking at the asymptotic behavior of $A_\pm$, one sees that $A_+ \xrightarrow[|\tilde{\omega}|\to 0]{} 1$, which translates the increasing closeness between the physical mode and the exact solution of the ODE. However, $A_- \xrightarrow[|\tilde{\omega}|\to 0]{} -1$: the numerical mode is oscillatory, switching sign at every iteration. Additionally, the parasitic mode travels in the direction opposite to the physical one.

The LF2 scheme was originally designed specifically for its non-dissipative property, which is effective for eigenvalues located on a portion of the imaginary axis. Since there is no amplitude error in this region, all the error is committed on the phase. The phase error of the scheme is given by:

$$\tilde{\theta}\left(\tilde{\omega}\right) \equiv \arg\left(A_+\right) - \tilde{\omega} = \arctan\left(\frac{\Im\left(A_+\right)}{\Re\left(A_+\right)}\right) - \tilde{\omega}.$$

This yields, for the LF2 scheme:

$$\tilde{\theta}_2(\tilde{\omega}) = \arctan\left(\frac{\tilde{\omega}}{\sqrt{1-\tilde{\omega}^2}}\right).$$

The properties of higher-order schemes of the form (3.33) can be studied using the same procedure. For the LFp scheme, one defines the following function, for even values of p:

$$f_p(\tilde{\omega}) = \sum_{\substack{0<k<p \\ k \text{ odd}}} 2i^{k-1}\frac{\tilde{\omega}^k}{k!}.$$

It should be noted that f_p is a real function. Then, the stability region of the LFp scheme is given by:

$$4 - f_p(\tilde{\omega}) > 0, \tag{3.35}$$

and the relative phase error is, for $\tilde{\omega}$ verifying (3.35):

$$\tilde{\theta}_p = \arctan\left(\frac{f_p(\tilde{\omega})}{\sqrt{4-f_p(\tilde{\omega})^2}}\right). \tag{3.36}$$

As said previously, the amplitude error of the LF schemes is 0 in their stability ranges. Outside these regions, an amplitude error is defined for the physical mode as:

$$\widetilde{A}_+ \equiv |A_+| - 1. \tag{3.37}$$

A plot of the stability regions for different LF schemes is presented on figure 3.7, along with amplitude and phase error plots along the upper imaginary axis. The LF2 stability region includes the $\mathbb{C}^-$ half-plane, but the non-dissipative property is only observed for the $\{|\tilde{\omega}| < 1, \tilde{\omega} \in i\mathbb{R}\}$ set. For the LF4 scheme, this property is obtained on the larger set $\{|\tilde{\omega}| < 2.845, \tilde{\omega} \in i\mathbb{R}\}$, at a higher computational cost. In return, the stability region in the left complex half-plane is reduced comparatively to LF2. Additional stability areas appear in the right complex half-plane, but are of no use for the hyperbolic problems considered in this manuscript. For LF6, the non-dissipative region is once again larger than that of LF4. However, it is not convex anymore, with two unstable "holes" appearing for (roughly) $\{1.5 < |\tilde{\omega}| < 1.7, \tilde{\omega} \in i\mathbb{R}\}$. This region is clearly visible on the amplitude error plot, and makes the LF6 scheme of questionable interest compared to LF4. Regarding dispersion, the LF2 error grows rapidly with $\tilde{\omega}$, indicating that a fine time discretization is necessary to achieve an accurate approximation of the exact phase. Moving to higher-order schemes significantly increases the $\tilde{\omega}$ range in which numerical dispersion is negligible.

LF-DG formulation for Maxwell's equations

Semi-discrete DG formulation for Maxwell's equations is here recalled for clarity of the following discussion:

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

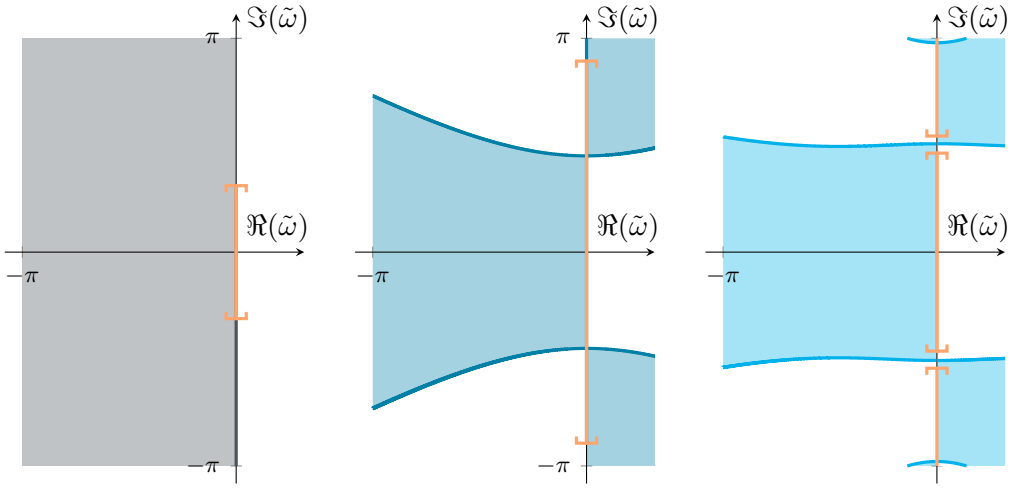
For the discretization of Maxwell's equations, is it more suitable to rewrite (3.29) as:

$$\phi^{n+1} = \phi^n + \Delta t f \left(t_{n+\frac{1}{2}}, \phi^{n+\frac{1}{2}} \right).$$

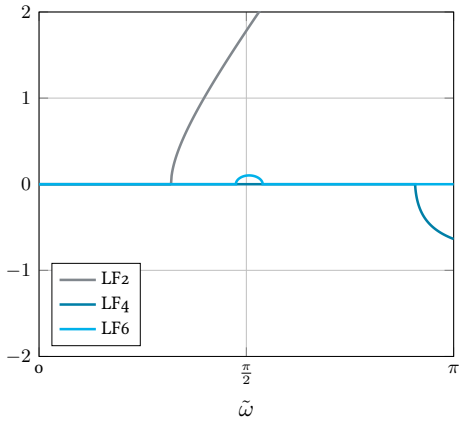
In the case of multiple variables problems (such as Maxwell's equations), LF schemes need to sample the unknowns on a staggered grid to remain explicit. A common choice is to split every timestep Δt in two: $\mathbf{E}$ is approximated at even time stations $t_n = n\Delta t$, while $\mathbf{H}$ and $\mathbf{J}$ are approximated at odd time stations $t_{n+\frac{1}{2}} = \left(n + \frac{1}{2}\right) \Delta t$. Given what was said earlier, the LF2 scheme consists in seeking the values of $\overline{\mathbf{E}}_i^{n+1}$ and $\overline{\mathbf{H}}_i^{n+\frac{3}{2}}$ when knowing those of $\overline{\mathbf{E}}_i^{n}$, $\overline{\mathbf{H}}_i^{n+\frac{1}{2}}$ and $\bar{\mathbf{J}}_i \left(t_{n+\frac{1}{2}} \right)$ with the following discretization:

$$\begin{aligned}
\frac{\overline{\mathbb{M}}_i^{\mu_r}}{\Delta t} \left(\overline{\mathbf{H}}_i^{n+\frac{3}{2}} - \overline{\mathbf{H}}_i^{n+\frac{1}{2}} \right) &= -\overline{\mathbb{K}}_i \overline{\mathbf{E}}_i^{n+1} + \sum_{l \in \mathcal{V}_i} \overline{\mathbb{S}}_{il} \left(\overline{\mathbf{E}}_*^{n+1} \times \mathbf{n}_{il} \right), \\
\frac{\overline{\mathbb{M}}_i^{\varepsilon_r}}{\Delta t} \left(\overline{\mathbf{E}}_i^{n+1} - \overline{\mathbf{E}}_i^{n} \right) &= \quad \overline{\mathbb{K}}_i \overline{\mathbf{H}}_i^{n+\frac{1}{2}} - \sum_{l \in \mathcal{V}_i} \overline{\mathbb{S}}_{il} \left(\overline{\mathbf{H}}_*^{n+\frac{1}{2}} \times \mathbf{n}_{il} \right) - \bar{\mathbf{J}}_i \left(t_{n+\frac{1}{2}} \right).
\end{aligned}$$

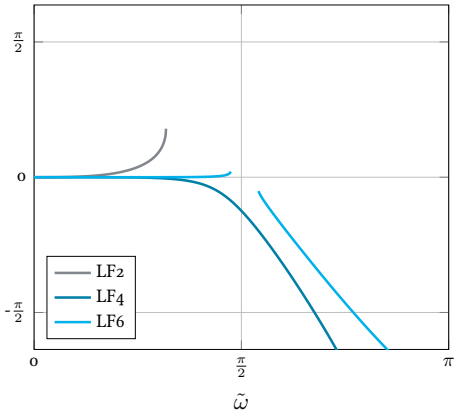
The previous formulation can be rewritten in a more general form:



(a) Stability region for LF2 (b) Stability region for LF4 (c) Stability region for LF6



(d) Amplitude error $\widetilde{A}_+ \equiv |A_+| - 1$ along the upper imaginary axis



(e) Phase error $\tilde{\theta}(\tilde{\omega}) \equiv \arg(A_+) - \tilde{\omega}$ along upper the imaginary axis

Figure 3.7 | Stability regions, amplitude error $\widetilde{A}_+$ and phase error $\tilde{\theta}$ induced by LF schemes of order 2 to 6. The first three plots present the stability regions in the complex plane. The orange section along the imaginary axis represents the set where the non-dissipative property is verified. The lower two plots only show the errors corresponding to the physical mode, on the upper imaginary axis.

LF2-DG scheme

$$\begin{aligned}\overline{\mathbf{H}}_i^{n+\frac{3}{2}} &= \overline{\mathbf{H}}_i^{n+\frac{1}{2}} + \Delta t\, G_{el}\left(\overline{\mathbf{E}}_h^{n+1}\right), \\ \overline{\mathbf{E}}_i^{n+1} &= \overline{\mathbf{E}}_i^{n} \quad + \Delta t\, G_{mag}\left(t_{n+\frac{1}{2}}, \overline{\mathbf{H}}_h^{n+\frac{1}{2}}\right).\end{aligned} \tag{3.38}$$

In the last equalities, the operators G_{el} and G_{mag} were introduced:

$$G_{el}\left(\overline{\mathbf{E}}_h^{n}\right) = \left(\overline{\mathbb{M}}_i^{\mu_r}\right)^{-1}\left(-\overline{\mathbb{K}}_i\,\overline{\mathbf{E}}_i^{n} + \sum_{l\in\mathcal{V}_i}\overline{\mathbb{S}}_{il}\left(\overline{\mathbf{E}}_*^{n}\times\mathbf{n}_{il}\right)\right),$$

$$G_{mag}\left(t_{n+\frac{1}{2}}, \overline{\mathbf{H}}_h^{n+\frac{1}{2}}\right) = \left(\overline{\mathbb{M}}_i^{\varepsilon_r}\right)^{-1}\left(\overline{\mathbb{K}}_i\,\overline{\mathbf{H}}_i^{n+\frac{1}{2}} - \sum_{l\in\mathcal{V}_i}\overline{\mathbb{S}}_{il}\left(\overline{\mathbf{H}}_*^{n+\frac{1}{2}}\times\mathbf{n}_{il}\right) - \bar{\mathbf{J}}_i\left(t_{n+\frac{1}{2}}\right)\right).$$

To achieve higher-order accuracy, one can simply adapt the formulation given in (3.33). In the present thesis, schemes are limited to fourth order accuracy, which leads to the following LF4 formulation, which is of fourth order in the source-free case ($\bar{\mathbf{J}}_h \equiv 0$):

LF4-DG scheme

$$\begin{aligned}\overline{\mathbf{H}}_i^{n+\frac{3}{2}} &= \overline{\mathbf{H}}_i^{n+\frac{1}{2}} + \Delta t\, G_{el}\left(\overline{\mathbf{E}}_h^{n+1}\right) + \frac{\Delta t^3}{24}\, G_{el}\circ G_{mag}\circ G_{el}\left(t_{n+1}, \overline{\mathbf{E}}_h^{n+1}\right), \\ \overline{\mathbf{E}}_i^{n+1} &= \overline{\mathbf{E}}_i^{n} \quad + \Delta t\, G_{mag}\left(t_{n+\frac{1}{2}}, \overline{\mathbf{H}}_i^{n+\frac{1}{2}}\right) \\ &\qquad + \frac{\Delta t^3}{24}\, G_{mag}\circ G_{el}\circ G_{mag}\left(t_{n+\frac{1}{2}}, \overline{\mathbf{H}}_i^{n+\frac{1}{2}}\right).\end{aligned} \tag{3.39}$$

Because of the second-order accuracy limitation in the presence of a source, other time schemes must also be considered. In the next section, we focus on the most used explicit time scheme, *i.e.* the Runge-Kutta scheme.

3.2.3 Runge-Kutta schemes

Classical RK schemes

Runge-Kutta schemes are a class of multi-stage algorithms that rely on multiple evaluations of the RHS of (3.24) to evolve the system in time. Unlike LF schemes, they do not combine different time levels to cancel terms in the Taylor expansions, which leads to dispersive and dissipative schemes. Suppose that one formally integrates problem (3.24) between t and $t+\Delta t$:

$$\phi(t+\Delta t) = \phi(t) + \int_t^{t+\Delta t} f(u,\phi(u))du. \tag{3.40}$$

Equation (3.40) could be approximated by a quadrature formula with s terms:

$$\phi(t+\Delta t) \simeq \phi(t) + \Delta t \sum_{j=1}^{s} \beta_j f(t+\delta_j \Delta t, \phi(t+\delta_j \Delta t))du, \tag{3.41}$$

where $(\beta_j)_{j\in[\![1,s]\!]}$ and $(\delta_j)_{j\in[\![1,s]\!]}$ are constants given by the choice of the quadrature formula. To evaluate the different $\phi(t+\delta_j\Delta t)$ values, RK methods exploit a prediction/correction technique, making use of previous guesses to calculate the next one. A standard way of writing the n^{th} timestep with an s-stage RK algorithm is:

$$\begin{aligned}
\phi_1 &= f\left(t_n, \phi^n\right) \\
\phi_k &= f\left(t_n + \delta_k \Delta t, \phi^n + \Delta t \sum_{j=1}^{s} \alpha_{j,k}\phi_j\right) \quad \text{for} \quad k = 2, ..., s, \\
\phi^{n+1} &= \phi^n + \Delta t \sum_{j=1}^{s} \beta_j \phi_j,
\end{aligned} \tag{3.42}$$

where it is supposed that $\phi_0 = \phi^n$. In the general case, the system (3.42) is implicit, since the summation in each intermediate stage extends to the maximum number of stages. In the following, we choose to work with explicit RK schemes (*i.e.* $\alpha_{j,k} = 0 \,\forall k \geq j$), which can be written as:

Explicit RK schemes

$$\begin{aligned}
\phi_1 &= f\left(t_n, \phi^n\right) \\
\phi_k &= f\left(t_n + \delta_k \Delta t, \phi^n + \Delta t \sum_{j=1}^{k-1} \alpha_{j,k}\phi_j\right) \quad \text{for} \quad k = 2, ..., s, \\
\phi^{n+1} &= \phi^n + \Delta t \sum_{j=1}^{s} \beta_j \phi_j.
\end{aligned} \tag{3.43}$$

Let us present an RK algorithm in details. To simplify, only 2 stages are considered. However, the principle for a larger number of stages remains identical. At time step n, the first stage of the method is:

$$\phi_1 = f\left(t_n, \phi^n\right) \equiv f_{t_n}. \tag{3.44}$$

Given (3.24), ϕ_1 is obviously an estimate of the slope of the solution at $t = t_n$. Then, the second stage is:

$$\phi_2 = f\left(t_n + \delta_2 \Delta t, \phi^n + \Delta t \alpha_{1,2}\phi_1\right). \tag{3.45}$$

In (3.45), one recognizes $\phi^n + \Delta t\alpha_{1,2}\phi_1$ as a forward Euler method, providing 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 stage 1. Hence, by setting $\delta_2 = \alpha_{1,2}$, one obtains with ϕ_2 an approximate value of the slope of the solution at $t_n + \alpha_{1,2}\Delta t$. Since the method is supposed to be done in two stages, the solution at the next timestep is obtained as:

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

Taylor-expanding the last term to the first order in Δt gives:

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

The latter expression can be matched with the Taylor expansion of ϕ:

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

where $f_t = f(t,\phi(t))$. In practice, the residual $O\left(\Delta t^3\right)$ is dropped, and the resulting system is:

$$\begin{aligned}\beta_1 + \beta_2 &= 1,\\ \beta_2\alpha_{1,2} &= \frac{1}{2}.\end{aligned} \tag{3.46}$$

The first equality reminds that RK schemes are equivalent to a quadrature rule, for which the sum of the weights must be unity. One immediately notices that (3.46) is underdetermined, meaning that there is not a unique set of coefficients verifying (3.46). A common choice satisfying the latter system is:

$$\begin{aligned}\alpha_{1,2} &= 1,\\ \beta_1 = \beta_2 &= \frac{1}{2}.\end{aligned} \tag{3.47}$$

This choice corresponds to a second-order RK method:

$$\begin{array}{rcl}\phi_1 & = & f\left(t_n, \phi^n\right)\\ \phi_2 & = & f\left(t_n + \Delta t, \phi^n + \Delta t\phi_1\right)\\ \phi^{n+1} & = & \phi^n + \dfrac{\Delta t}{2}\left(\phi_1 + \phi_2\right).\end{array} \tag{3.48}$$

In the first step, f_{t_n} gives an estimate of $\frac{\partial\phi}{\partial t}(t_n)$. Then, a first-order approximation $\phi_*^{n+1} = \phi^n + \Delta t\phi_1$ of $\phi(t+\Delta t)$ is obtained *via* a forward Euler step. In the second step, ϕ_*^{n+1} is used to estimate $\frac{\partial\phi}{\partial t}(t_{n+1})$. Finally, ϕ^{n+1} is obtained from a second forward Euler step, for which the slope is average between ϕ_1 and ϕ_2. The algorithm is summed up on figure 3.8.

In the same fashion, the four steps fourth-order RK algorithm extends this method by exploiting four different evaluations of the slope in the average. The most classical version of this scheme reads:

$$\begin{array}{rcl}\phi_1 & = & f\left(t_n, \phi^n\right)\\ \phi_2 & = & f\left(t_n + \dfrac{\Delta t}{2}, \phi^n + \dfrac{\Delta t}{2}\phi_1\right)\\ \phi_3 & = & f\left(t_n + \dfrac{\Delta t}{2}, \phi^n + \dfrac{\Delta t}{2}\phi_2\right)\\ \phi_4 & = & f\left(t_n + \Delta t, \phi^n + \Delta t\phi_3\right)\\ \phi^{n+1} & = & \phi^n + \dfrac{\Delta t}{6}\left(\phi_1 + 2\phi_2 + 2\phi_3 + \phi_4\right).\end{array} \tag{3.49}$$

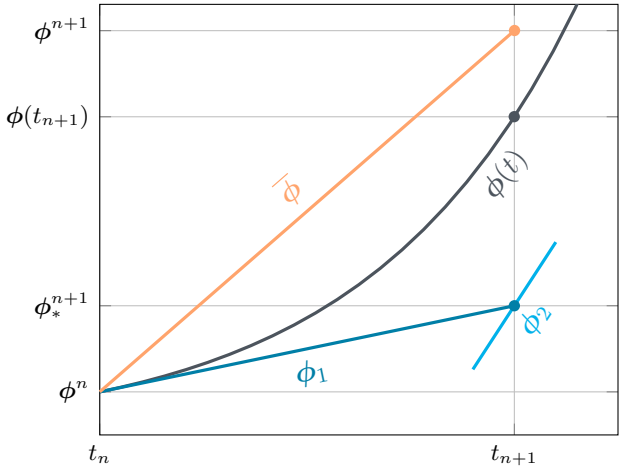


Figure 3.8 | Steps of the RK2 algorithm. ϕ_*^{n+1} is the estimate obtained by the forward Euler method, for which the slope of the solution between t_n and $t_n + \Delta t$ is approached by ϕ_1. The RK2 estimate is obtained by averaging ϕ_1 with a second approximation of the slope, obtained from ϕ_*^{n+1} in $t_n + \Delta t$, thus yielding a more precise approximation of $\phi(t_{n+1})$. $\overline{\phi}$ represents the average of the different slopes.

Table 3.1 | Minimal number of stages to obtain p^{th}-order convergence for standard explicit RK schemes (see [But87] and [Lam91]).

p	1	2	3	4	5	6	7	8
s_{min}	1	2	3	4	6	7	9	11

This algorithm is summed up on figure 3.9. Although it could seem that a standard s-stages RK algorithm is accurate to s^{th} order, this is only true up to order four. Indeed, above this value, obtaining an RK scheme of order p requires more than p stages ([But87], [Lam91]). The minimal number of stages to obtain a given order of accuracy are displayed on table 3.1.

Properties of the RKp schemes

As in the case of LF schemes (see section 3.2.2), the properties of RK schemes can be studied *via* their amplification factor in the case of a monochromatic solution. Here, we consider a p^{th}-order RK algorithm based on a Taylor development to order s:

$$\phi(t_{n+1}) \simeq \phi(t_n) + \sum_{k=1}^{s} \frac{\Delta t^k}{k!} \frac{\partial^k \phi}{\partial t^k}. \tag{3.50}$$

The amplification factor of (3.50) is:

$$A = \sum_{k=0}^{s} \frac{(i\tilde{\omega})^k}{k!}. \tag{3.51}$$

The difference to LF schemes is clearly visible, since no compensation can occur during the calculation of $|A|$. The stability regions and phase errors computed with expression (3.51) are presented on figure 3.10 for classical RK schemes from first to fourth order. The size of the stability region increases with the order of the scheme, thus allowing larger timesteps to be used for the computation. However, this is at

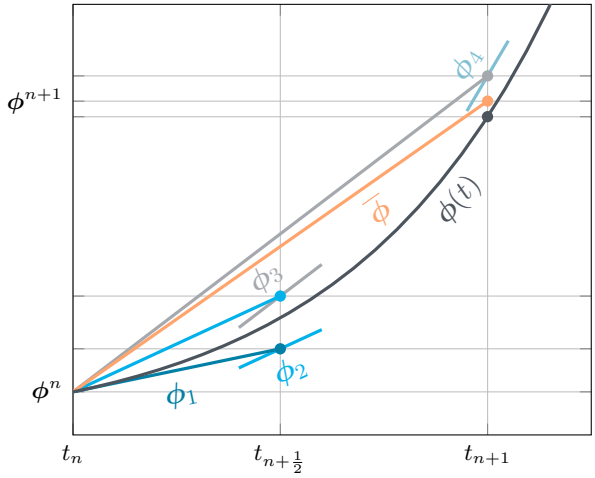


Figure 3.9 | Steps of the RK4 algorithm. $\overline{\phi}$ represents the average of the different slopes.

the cost of a higher computational effort and memory occupation. For eigenvalues along the imaginary axis, the two classical RK3 and RK4 algorithms are accelerating, the fourth-order scheme presenting a very low phase error for $\widetilde{\omega}$ up to $\frac{\pi}{2}$.

3.2.4 Low-storage schemes

Although standard RK schemes can achieve very high accuracy, an immediate remark is that, in the general case, an algorithm of the form (3.43) with s substeps requires s memory registers, in addition to the solution register. This situation can seem bearable for small systems and/or low values of s, but memory consumption can quickly become a constraining factor for large problems. A possible solution is to use a specific class of RK algorithms, the low-storage RK schemes, for which the required memory is limited. Such schemes are usually presented under the Williamson formulation [Wil80]:

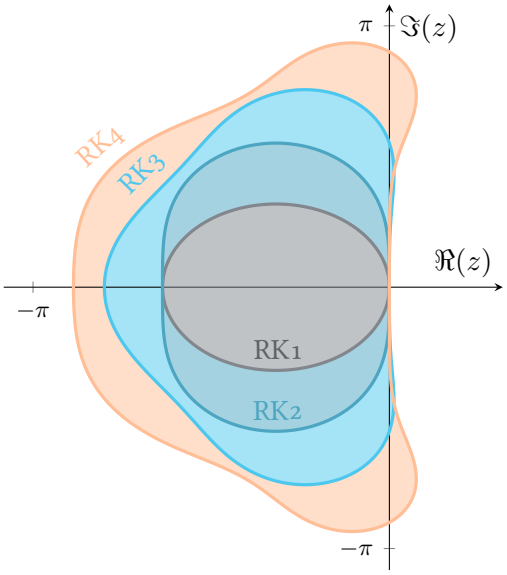
Low-storage RK schemes

$$\left.\begin{array}{rcl} \phi_1 & = & \phi^n \\ \phi_2 & = & a_k\phi_2 + \Delta t f\left(t_n + d_k\Delta t, \phi_1\right), \\ \phi_1 & = & \phi_1 + b_k\phi_2, \\ \phi^{n+1} & = & \phi_1. \end{array}\right\} \text{ for } k = 1, \ldots, s \tag{3.52}$$

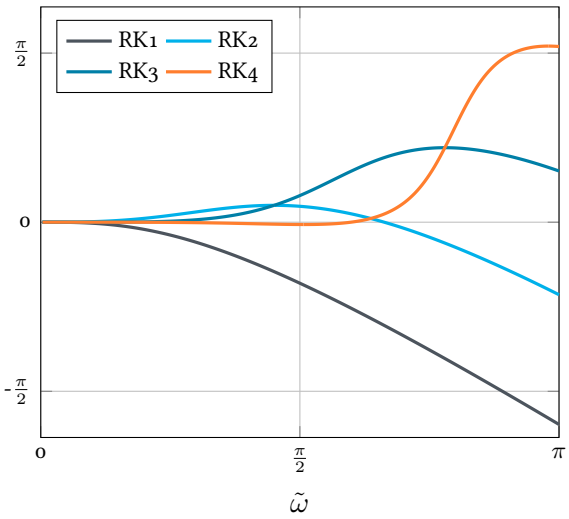
It is clear that such algorithms only require two memory registers at every moment, whatever the number of stages. However, convergence to order p is not ensured in s substeps anymore. At least 5 stages are necessary to achieve a fourth-order convergence [CK94], leading to an amplification factor of the form:

$$A = \sum_{k=0}^{s} \gamma_k (i\widetilde{\omega})^k. \tag{3.53}$$

Since the condition for fourth-order accuracy only imposes four equations on the γ_k coefficients, LSRK schemes properties such as dissipation, dispersion or size of the stability region can be tuned by exploiting



(a) Stability contour



(b) Phase error along upper the imaginary axis

Figure 3.10 | Stability contour and phase error induced by classical Runge-Kutta schemes of order 1 to 4. Left panel: The size of the stability region increases with the order of the scheme, thus allowing larger timesteps to be used for the computation. However, this is at the cost of a higher computational effort and memory occupation. Right panel: The two classical RK3 and RK4 algorithms are accelerating, the fourth-order scheme presenting a very low phase error for $\tilde{\omega}$ up to $\frac{\pi}{2}$.

the remaining parameters (see section 3.2.4). The functioning of LSRK schemes is slightly different from what was exposed previously. For classical RK algorithms, the slope is estimated several times between t_n and t_{n+1} using better and better averages at each stage. Finally, the solution is evolved from ϕ^n to ϕ^{n+1} in a single Euler step using an average of all the previously calculated slopes. For LSRK methods, 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 $t_n + b_1$. During the next stage, the slope stored in ϕ_2 will be a weighted average between the previous estimate and the new one, calculated from ϕ_1 at $t_n + b_1$. Similarly, at each stage, the slope stored in ϕ_2 is an average between all the preceding slope values and the new one. An example of LSRK algorithm is presented on figure 3.11.

LSRK-DG formulation for Maxwell's equations

One of the most spread LSRK scheme in computational electromagnetics is the 5-stage fourth-order algorithm proposed by Kennedy and Carpenter in 1994 [CK94], hereafter referred to as LSRK4-5. In this thesis, optimized low-storage RK4 schemes from [NDB12] will be used. Their stability regions are tailored to fit the DG spectrum as closely as possible, depending on the upwinding factor α. Hence, a 12-stage scheme (for centered flux) and a 14-stage scheme (for upwind flux) are considered, both being fourth-order accurate. On figure 3.12(a), the stability contour of the LSRK4-14 algorithm is presented, and compared to that of the LSRK5 scheme. Although it appears that the stability contour of the LSRK4-14 method is larger, for an accurate comparison the latter must be scaled by the number of steps required, so the expected gain in efficiency can be seen directly on the plot. This is presented on figure 3.12(b).

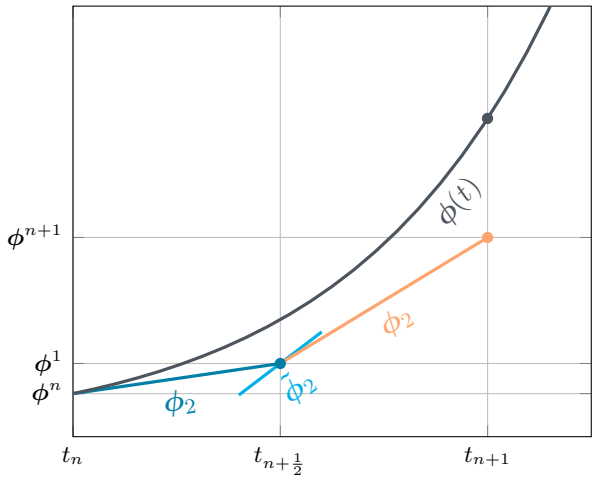
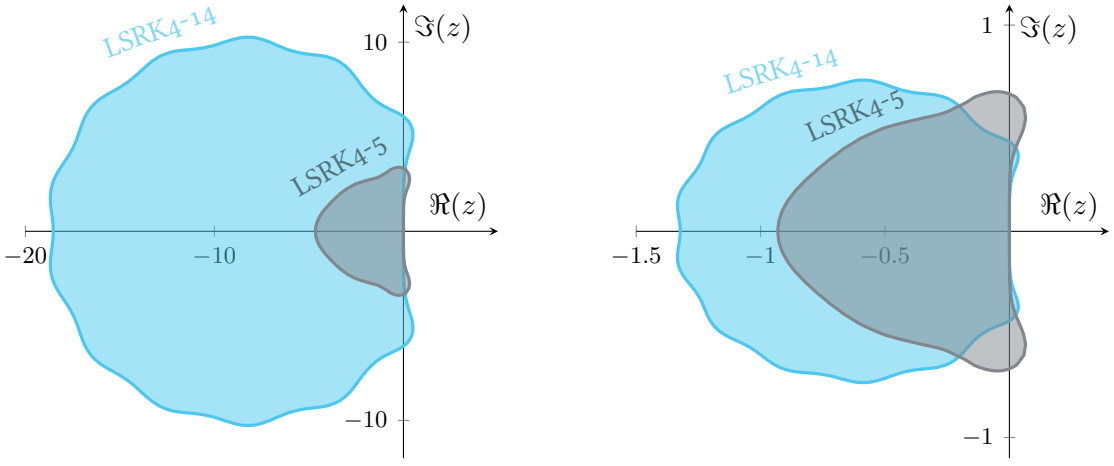


Figure 3.11 | Steps of the LSRK2 algorithm. Here, $\tilde{\phi}_2$ represents $\Delta t f\left(t_n + \frac{\Delta t}{2}, \phi^1\right)$. The number of stages is voluntarily reduced to 2 for clarity. However, for a larger number of stages, the principle remains identical.



(a) Non-scaled stability contours

(b) Scaled stability contours

Figure 3.12 | Stability contours of the LSRK4-14 and LSRK4-5 schemes. On the left panel, the contours are scaled by the number of stages required to complete one iteration of the algorithm. Despite the large number of required stages, the LSRK4-14 scheme provides a much larger maximal stable timestep.

3.2.5 Timestep choice and the CFL condition

A theoretically possible method to select the timestep in practice would be to compute the set of the largest DG operator eigenvalues, and to scale them by a small enough timestep so they all fit inside the stability contour of the time integrator. This technique, however, suffers from two major drawbacks: (i) it only constitutes a necessary condition, and (ii) it can become very expensive, even for moderate size systems. A more thorough method consists in deducing a timestep restriction from an energy-based stability study (see section 3.5). Such conditions are usually known as Courant-Friedrich-Levy (CFL) conditions, and are inherent to every explicit timestepping techniques. In this work, we exploit the theoretical results from [F+05]. Therefore, for a space discretization with polynomial order p, the timestep is chosen as follows:

$$\Delta t_p = c_p \min_{T_i \in \mathcal{T}_h} \frac{V_{T_i}}{A_{T_i}}, \tag{3.54}$$

where V_{T_i} and A_{T_i} are respectively the volume and the area of cell T_i, and c_p is an order-dependent constant. In practice, the maximal acceptable value for c_p is determined on a basic test case such as that described in section 2.1.4.

3.3 Validation and numerical experiments

3.3.1 PEC cubic cavity mode

To validate the implementation of the DGTD algorithm, it is necessary to check that the convergence rate of the numerical method matches the theoretical rate (see section 3.5). To do so, the cubic cavity mode described in section 2.1.4 is considered. Increasingly fine meshes are generated, for which the minimal edge size is noted h_m, where m is the index of the mesh (mesh characteristics are given in table 3.2). The mode is evolved during a time t_{max} corresponding to 30 periods in the cavity. For each simulation, the global $l^\infty\left([0, t_{\text{max}}], L^2\right)$ error is computed. For two successive meshes, the maximum error levels are measured. Then, the numerical rate of convergence is deduced as:

$$r_{\frac{m+1}{m}} = \frac{\log\left(\dfrac{\max\limits_{t\in[0,t_{\text{max}}]} \|\mathbf{E} - \mathbf{E}_{h_m}\|}{\max\limits_{t\in[0,t_{\text{max}}]} \|\mathbf{E} - \mathbf{E}_{h_{m+1}}\|}\right)}{\log\left(\dfrac{h_m}{h_{m+1}}\right)}. \tag{3.55}$$

The numerical convergence rates are calculated for various spatial and temporal approximations, and with centered and upwind fluxes. The results are summed up in table 3.3. The asymptotical theoretical error is in $O\left(\Delta t^n + h^p\right)$ for centered fluxes, and in $O\left(\Delta t^n + h^{p+1}\right)$ for the upwind case. The numerical rates match for all combinations of space and time discretization, which validates the implementation of the DGTD method.

3.3.2 Convergence with centered and upwind fluxes

Although the theoretical rates of convergence are known for centered ($\alpha = 0$) and upwind ($\alpha = 1$) fluxes, the situation for intermediate values of α is unclear. To investigate this matter, convergence is tested with increasing values of α for the cavity case considered in last section. To obtain significant results, the retained rate is computed between meshes M3 and M4 only. Results for third order are displayed on figure 3.13: as can be seen, a very small upwinding (slightly superior to 0.1) is sufficient to obtain fourth

Table 3.2 | Meshes characteristics for the cubic PEC cavity case. n_s is the number of vertices, n_t the number of tetrahedrons and h_m the typical size of the largest tetrahedron.

	M1	**M2**	**M3**	**M4**
n_s	125	729	4913	35937
n_t	384	3072	24576	196608
h_m	0.433	0.216	0.108	0.0541

Table 3.3 | Error levels and convergence rates of the cubic cavity case for different approximation orders, fluxes and time schemes with meshes of increasing refinement. 0 refers to centered fluxes with LF2, 1 to centered fluxes with LF4, 2 to centered fluxes with LSRK4, and 3 to upwind fluxes with LSRK4.

		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
$\mathbb{P}_1$	0	5.42×10^{-1}	–	5.48×10^{-2}	3.31	2.47×10^{-2}	1.72	9.37×10^{-3}	1.04
	1	–	–	–	–	–	–	–	–
	2	2.85×10^{-2}	–	4.42×10^{-2}	2.69	1.71×10^{-2}	1.37	7.69×10^{-3}	1.15
	3	2.87×10^{-1}	–	6.05×10^{-2}	2.25	8.66×10^{-3}	2.80	1.46×10^{-3}	2.57
$\mathbb{P}_2$	0	5.55×10^{-2}	–	9.99×10^{-3}	2.47	2.27×10^{-3}	2.13	5.49×10^{-4}	2.03
	1	–	–	–	–	–	–	–	–
	2	4.32×10^{-2}	–	5.15×10^{-3}	3.07	9.83×10^{-4}	2.39	2.17×10^{-4}	2.18
	3	1.47×10^{-2}	–	1.36×10^{-3}	3.43	1.75×10^{-4}	2.96	2.19×10^{-5}	3.00
$\mathbb{P}_3$	0	1.62×10^{-2}	–	3.83×10^{-3}	2.07	9.48×10^{-4}	2.01	2.36×10^{-4}	2.00
	1	5.25×10^{-3}	–	4.88×10^{-4}	3.42	4.97×10^{-5}	3.29	5.91×10^{-6}	3.03
	2	5.12×10^{-3}	–	4.82×10^{-4}	3.41	4.96×10^{-5}	3.28	5.91×10^{-6}	3.07
	3	9.24×10^{-4}	–	5.87×10^{-5}	3.98	3.72×10^{-6}	3.98	2.33×10^{-7}	4.00
$\mathbb{P}_4$	0	–	–	–	–	–	–	–	–
	1	5.45×10^{-4}	–	2.84×10^{-5}	4.26	1.48×10^{-6}	4.15	6.36×10^{-8}	4.02
	2	4.10×10^{-4}	–	2.23×10^{-5}	4.20	1.25×10^{-6}	4.16	7.14×10^{-8}	4.13
	3	9.45×10^{-5}	–	3.11×10^{-6}	4.92	1.98×10^{-7}	3.97	1.15×10^{-8}	4.11

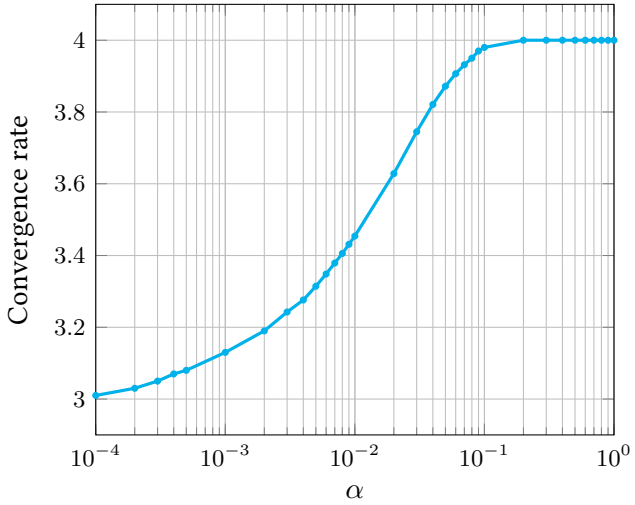


Figure 3.13 | Convergence rate for DG $\mathbb{P}_3$ method with LSRK4 algorithm, for various values of the unwinding factor α.

order. This remark is coherent with what is exposed in [Gon13]. A possible followup to this point would be to evaluate the difference induced on realistic cases in terms of dissipation of the discrete scheme.

3.3.3 Flux weightings

As shown in section 3.3.2, there is not a unique choice of flux leading to a stable and convergent DGTD formulation. The upwind and centered fluxes, as well as all the partially penalized fluxes obtained for intermediate values of α between 0 and 1 are acceptable, along with a larger choice available from the finite volume community. In certain references from the DGTD literature for Maxwell's equations, centered and upwind fluxes are sometimes used in slightly different versions than those presented in section 3.1.3, in the sense that the contributions from each side of the face are not weighted with their respective impedance/inductance. Although they are valid flux choices, it seems intuitive that they will be less efficient than their weighted counterparts in the case of heterogeneous systems with large ε and μ jumps. To verify this hypothesis, a doubly periodic dielectric slab surrounded with vacuum is considered (see section 3.4.2). The latter is illuminated by a wideband plane wave in normal incidence, for which the frequency range of interest is $[300, 1500]$ THz. For values of ε_r from 2 to 100, the numerical reflection coefficient (see section 4.4.3) is computed and compared with the analytical solution of the problem. A $\mathbb{P}_3$ spatial discretization is used, along with a fourth-order LSRK scheme, with both centered and upwind fluxes. In both cases, weighted and non-weighted versions are used: results are summed up on figure 3.14. Although the difference is not clear for the centered case, the weighting of the upwind flux seems to asymptotically induce a factor $\frac{1}{2}$ with the non-weighted flux on the total error, thus confirming the interest of the weighted flux over the non-weighted version. For small jumps of the permittivity, however, the interest is not so obvious. Based on this remark, we propose to exploit and validate the use of the non-weighted flux for anisotropic materials in next section.

3.3.4 Fluxes for anisotropic materials

The Riemann problem presented in section 3.1.8 has to be rewritten and solved to account for anisotropic materials, which is done in [AABG12]. However, in the same fashion as last section, non-weighted fluxes

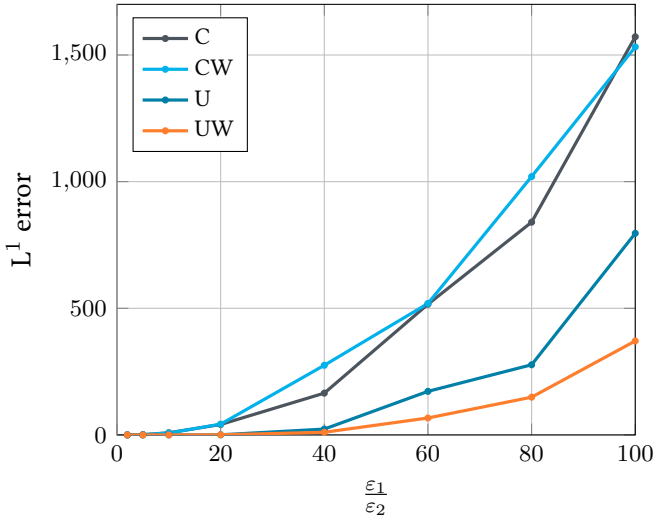


Figure 3.14 | L^1 error obtained for increasing jumps of ε, with Centered (C), Centered Weighted (CW), Upwind (U) and Upwind Weighted (UW) fluxes. Although the difference is not clear for the centered case, the weighting of the upwind flux seems to asymptotically induce a factor $\frac{1}{2}$ with the non-weighted flux.

Table 3.4 | Convergence rates for the anisotropic PEC cavity with meshes of increasing refinement and $\mathbb{P}_3$ polynomials. Upwind flux and LSRK$_4$ time-scheme were used. ω_1 and ω_2 refer to the two admissible modes of the cavity.

	M1	**M2**	**M3**	**M4**
ω_1	–	3.96	3.98	4.00
ω_2	–	4.03	4.00	4.00

naturally account for anisotropic materials with reasonable accuracy. To support this statement, we consider the case of a PEC cavity filled with an anisotropic material. The solution of this problem is known, and presented at the end of section 2.1.4. A short convergence study is conducted for both frequencies ω_1 and ω_2 on increasingly refined meshes, similarly to section 3.3.1. $\mathbb{P}_3$ polynomial approximation is used in conjunction with LSRK$_4$ scheme in time and fully upwind flux: results are summed up on table 3.4. As expected, the convergence rate in $O\left(\Delta t^n + h^{p+1}\right)$ is obtained, which validates the use of a non-weighted upwind flux for anisotropic materials.

3.4 The ADE method for dispersive materials

3.4.1 ADE formulation in the DGTD framework

The discretization of the ADE formulations presented previously (see 2.31 and 2.32) is straightforward, since the additional equations do not contain any spatial derivatives. Here, the second-order LFDG scheme for Maxwell's equations in a Drude material is given as:

$$\begin{aligned}
\frac{\overline{\mathbb{M}}_i}{\Delta t}\left(\overline{\mathbf{H}}_i^{n+\frac{3}{2}} - \overline{\mathbf{H}}_i^{n+\frac{1}{2}}\right) &= -\overline{\mathbb{K}}_i\,\overline{\mathbf{E}}_i^{n+1} + \sum_{l\in\nu_i}\overline{\mathbb{S}}_{il}\left(\overline{\mathbf{E}}_l^{n+1}\times\mathbf{n}_{il}\right), \\
\frac{\overline{\mathbb{M}}_i^{\varepsilon_\infty}}{\Delta t}\left(\overline{\mathbf{E}}_i^{n+1} - \overline{\mathbf{E}}_i^{n}\right) &= \quad \overline{\mathbb{K}}_i\,\overline{\mathbf{H}}_i^{n+\frac{1}{2}} - \sum_{l\in\nu_i}\overline{\mathbb{S}}_{il}\left(\overline{\mathbf{H}}_l^{n+\frac{1}{2}}\times\mathbf{n}_{il}\right) - \overline{\mathbb{M}}_i\,\bar{\mathbf{J}}_i^{n+\frac{1}{2}}, \\
\frac{1}{\Delta t}\left(\bar{\mathbf{J}}_i^{n+\frac{3}{2}} - \bar{\mathbf{J}}_i^{n+\frac{1}{2}}\right) &= \quad \omega_d^2\,\overline{\mathbf{E}}_i^{n+1} - \frac{\gamma_d}{2}\left(\bar{\mathbf{J}}_i^{n+\frac{3}{2}} + \bar{\mathbf{J}}_i^{n+\frac{1}{2}}\right).
\end{aligned}$$

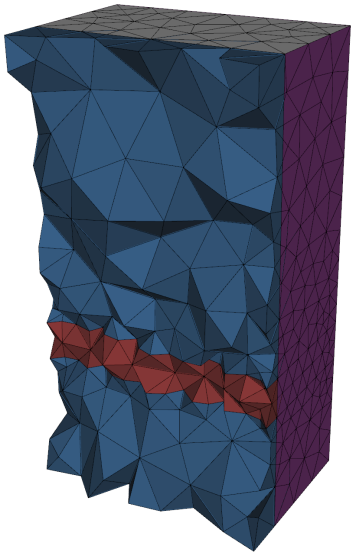
As can be noticed, the currents are evaluated at odd time-stations. Given the relation between $\mathbf{J}$ and $\mathbf{P}$, with this choice polarizations must be evaluated at even time-stations. This remark remains valid in the case of the generalized dispersive model. In the case of a Runge-Kutta time scheme, the discretization is straightforward, since all fields are evaluated at the same time-stations.

3.4.2 Validation

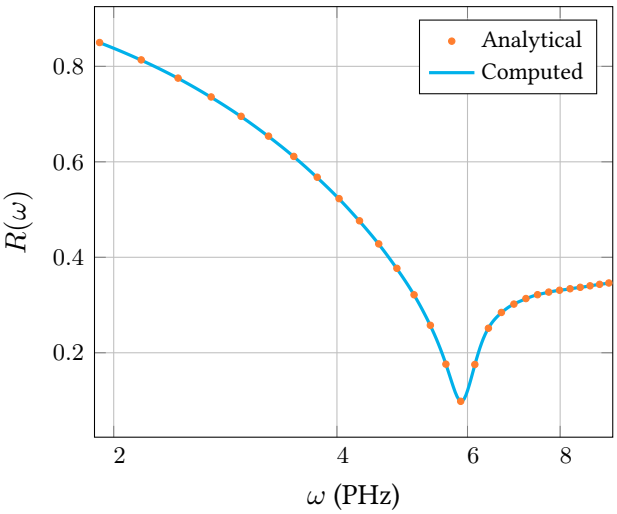
To validate the implementation of the generalized dispersive model in the DGTD framework, a simple setup composed of a doubly periodic silver slab of thickness 10 nm is considered. The latter is illuminated by a wideband plane wave in normal incidence, for which the frequency range of interest is $[300, 1500]$ THz. Silver is described by a 4SOGP model (see section 2.2.3), which parameters are available in appendix A. The numerical reflection coefficient (see section 4.4.3) is computed and compared with the analytical solution of the problem. The set-up is presented on figure 3.15(a), while the results are displayed in figure 3.15(b). For this model as well as for many others tested, a perfect agreement between analytical and computed solution is obtained, which validates the implementation.

3.4.3 On the necessity of a good description of dispersive materials

In this section, the computation of the scattering cross-section of a core-silica-gold-shell device is presented. Its geometrical parameters are $R_{in} = 150$ nm, and $R_{out} = 172$ nm. The latter is enclosed in a Total-Field/Scattered-Field (TF/SF) interface (see section 4.2.2), on which a wideband plane wave is imposed. Details about the computation of the scattering cross-section can be found in section 4.4.1. Several computations are done, for increasingly-complex gold dispersion models, while the silica core is described by a constant $\varepsilon = 1.5$ permittivity. Results for Drude and 4SOGP models are presented in figure 3.16. As could be expected from what was presented in section 2.2, the high-frequency behavior of the scatterer is strongly modified when an enhanced dispersion model is used. Hence, depending on the considered frequency range, a careful selection of the number of poles must be done, 4 second-order generalized poles being a good compromise for wideband computations. To support this statement, the induced overhead was calculated for different dispersion models. In average, a dispersive tetrahedron only requires 4.6% additional memory space and 0.6% additional CPU time per pole compared to a non-dispersive one. Given that the amount of dispersive tetrahedra in classical nano-optics devices is usually less than 20% of

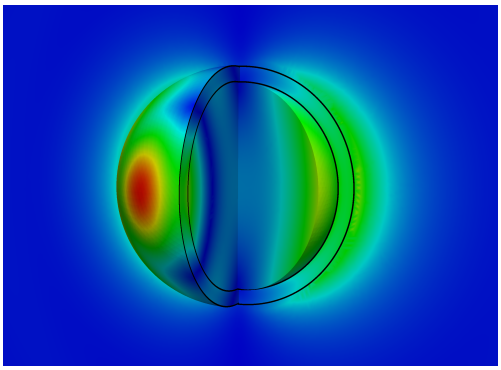


(a) Mesh of the doubly periodic silver slab. The slab is in red, while the blue corresponds to vacuum. Purple boundary triangles correspond to periodic conditions, while on gray ones an absorbing boundary condition is imposed.

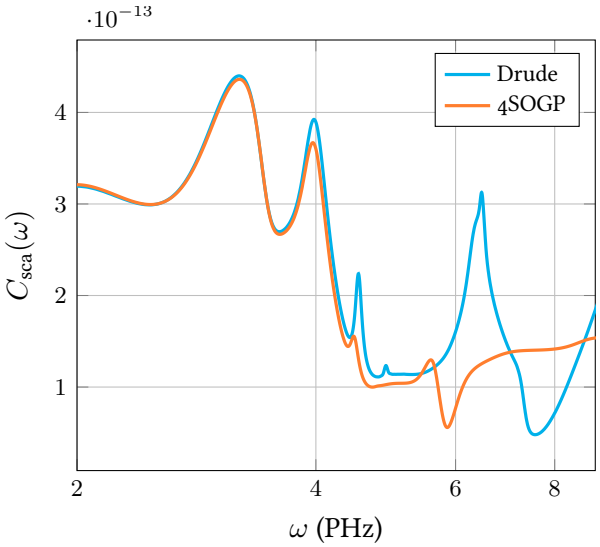


(b) Analytical and computed reflection spectra for a 4SOGP fit. As can be seen the DGTD results are in good agreement with the analytical reflection coefficient. The results are similar for other dispersive models.

Figure 3.15 | Set-up (3.15(a)) and results (3.15(b)) of the doubly periodic silver slab. $\mathbb{P}_4$ polynomial approximation is used for the spatial DG approximation.



(a) Modulus of the **E** field in the vicinity of the nanoshell at $t = 2 \times 10^{-14}$ s. A 4SOGP dispersion model is used to describe the gold shell.



(b) Computed scattering cross-sections of the nanoshell for various gold dispersion models.

Figure 3.16 | E near-field solution (3.16(a)) and scattering cross-section (3.16(b)) of the silica/gold nanoshell device. $\mathbb{P}_4$ polynomial approximation is used for the spatial DG discretization, along with curvilinear elements for an enhanced geometrical description of the shell (see chapter 5).

the total, this makes the generalized dispersive model a cheap way to achieve a good description of the material properties.

3.5 Theoretical results

In this section, stability and convergence results related to LF- and RK-DGTD are presented for both standard and dispersive materials. In each case, the sketches of the proofs are briefly reminded, the full demonstrations being conducted in the references given below. Throughout this section, the norm $\|\cdot\|$ is understood as $\|\cdot\|_{L^2(\Omega)}$, unless stated otherwise. Moreover, any constant C is supposed to be independant of the time and space discretizations, namely Δt and h.

3.5.1 Stability

The stability of the fully-discrete DGTD algorithm can be proved by energetic considerations. First, an energy, *i.e.* a quadratic definite positive form of the variables, is associated to the considered differential system. For the non-dispersive, non-magnetic Maxwell equations (2.11) - (2.12) on domain Ω with metallic boundary conditions, a possible form is:

$$\xi(t) = \frac{1}{2}\left(\|\mathbf{H}(t)\|^2 + \varepsilon_r \|\mathbf{E}(t)\|^2\right). \tag{3.56}$$

The stability of the system is associated to a decreasing, or at least bounded energy in time. Assuming that $(\mathbf{E}, \mathbf{H})$ have a sufficient regularity, it can be proved that:

$$\frac{\partial \xi}{\partial t} = 0 \quad \text{on } [0, T].$$

A similar result can be obtained for the semi-discrete as well as for the fully-discrete Maxwell system. For the Leap-Frog time scheme, a possible discrete energy form at time-station t_n is:

$$\xi^n = \sum_{i=1}^{N} \xi_i^n = \sum_{i=1}^{N} \frac{1}{2}\left(\int_{T_i} \mathbf{H}_i^{n+\frac{1}{2}} \cdot \mathbf{H}_i^{n-\frac{1}{2}} + \varepsilon_r \int_{T_i} \mathbf{E}_i^n \cdot \mathbf{E}_i^n\right), \tag{3.57}$$

where ξ_i^n denotes the local contribution of cell T_i to the total energy ξ^n. Starting from (3.57) and using centered fluxes, several lines of work [F+05] yield the following equality:

$$\frac{\xi^{n+1} - \xi^n}{\Delta t} = -\sum_{i=1}^{N} \sum_{k \in \mathcal{V}_i} \frac{1}{2} \int_{a_{ik}} \left(\mathbf{E}_i^{\left[n+\frac{1}{2}\right]} \times \mathbf{H}_k^{n+\frac{1}{2}} + \mathbf{E}_k^{\left[n+\frac{1}{2}\right]} \times \mathbf{H}_i^{n+\frac{1}{2}}\right) \cdot \mathbf{n}_{ik}.$$

Then, for metallic boundary conditions:

$$\frac{\xi^{n+1} - \xi^n}{\Delta t} = 0.$$

Then, under a CFL condition of the type $\Delta t \leq Ch$, one can prove that (3.57) is a a quadratic definite positive form, and therefore that $\left\|\mathbf{H}^{n+\frac{1}{2}}\right\|$ and $\|\mathbf{E}^n\|$ are bounded independantly of $n \in \mathbb{N}$. We conducted a similar work for materials described by the generalized dispersive model, both in centered Leap-Frog [VKLS13] and upwind Runge-Kutta [LSV]. The retained energy for the continuous fields is:

$$\begin{aligned}\xi(t) = \frac{1}{2}\Bigg(&\|\mathbf{H}(t)\|^2 + \varepsilon_\infty \|\mathbf{E}(t)\|^2 + \sum_{l \in L_1} \frac{b_l}{a_l} \|\mathbf{P}_l(t)\|^2 \\ &+ \sum_{l \in L_2} \frac{e_l}{c_l + d_l f_l} \|\mathbf{P}_l(t)\|^2 + \sum_{l \in L_2} \frac{1}{c_l + d_l f_l} \|\mathcal{J}\|^2\Bigg),\end{aligned} \tag{3.58}$$

In both cases, the associated fully discrete fields were proved to be bounded under the same type of CFL condition.

3.5.2 Convergence

The goal of the convergence study is to prove that the norm of the difference between the exact and the numerical solution is controlled by a norm of the exact solution, and that it goes to 0 as $(\Delta t, h) \to 0$. In a first step, the convergence proof of the semi-discrete problem is sketched. Then, using the previous result, the main steps of the convergence proof of the fully-discrete scheme are given.

Semi-discrete scheme

Let us define π_h as the orthogonal L^2 projector of the continuous solution on space V_h. Then, let $(\mathbf{H}, \mathbf{E})$ be the solution of the continuous Maxwell problem, and $(\mathbf{H}_h, \mathbf{E}_h) \in \mathcal{C}^1\left([0,T], V_h^2\right)$ that of the semi-discrete Maxwell problem. Finally, we define $\gamma(t)$ as:

$$\gamma(t) = \|\pi_h(\mathbf{H}) - \mathbf{H}_h\|^2 + \|\pi_h(\mathbf{E}) - \mathbf{E}_h\|^2, \forall t \in [0,T].$$

Following the classical bounding techniques such as in [SL11], one can prove that under CFL condition, if $(\mathbf{H}, \mathbf{E}) \in \mathcal{C}^0\left([0,T], H^{s+1}(\Omega)^6\right)$ for $s \geq 0$, then there exists $C \geq 0$ independent of h such that:

$$\max_{t\in[0,T]} \gamma(t)^{\frac{1}{2}} \leq C h^{\min(s,p)} \|(\mathbf{H}, \mathbf{E})\|_{\mathcal{C}^0\left([0,T], H^{s+1}(\Omega)^6\right)}.$$

This result is now used to prove the convergence of the fully-discrete scheme. Similar developments can be done in the context of Drude and generalized dispersive models for centered [VKLS13] and upwind fluxes [LSV].

Fully-discrete scheme

To prove convergence of the fully-discrete scheme, one needs to bound the following term:

$$\max_{n\in[0,N_t]} \left(\left\| \mathbf{H}\left(t_{n+\frac{1}{2}}\right) - \mathbf{H}_h^{n+\frac{1}{2}} \right\|^2 + \left\| \mathbf{E}(t_n) - \mathbf{E}_h^n \right\|^2 \right)^{\frac{1}{2}},$$

assuming that $(\mathbf{H}, \mathbf{E}) \in \mathcal{C}^3\left([0,T], L^2(\Omega)^6\right) \bigcap \mathcal{C}^0\left([0,T], H^{s+1}(\Omega)^6\right)$. Here, N_t denotes the maximal number of timesteps. This can be done in three steps, by splitting the latter quantity with a triangular inequality as follows (the development is written for the $\mathbf{E}$ part, but is identical for the $\mathbf{H}$ part):

$$\begin{aligned} \|\mathbf{E}(t_n) - \mathbf{E}_h^n\| &= \|\mathbf{E}(t_n) - \pi_h(\mathbf{E})(t_n) + \pi_h(\mathbf{E})(t_n) - \mathbf{E}_h(t_n) + \mathbf{E}_h(t_n) - \mathbf{E}_h^n\| \\ &\leq \underbrace{\|\mathbf{E}(t_n) - \pi_h(\mathbf{E})(t_n)\|}_{\beta_1} + \underbrace{\|\pi_h(\mathbf{E})(t_n) - \mathbf{E}_h(t_n)\|}_{\beta_2} + \underbrace{\|\mathbf{E}_h(t_n) - \mathbf{E}_h^n\|}_{\beta_3}. \end{aligned}$$

Bounding of β_1 is easily obtained as a property of the projector π_h, while that of β_2 is a direct consequence of what was shown for the semi-discrete scheme. The bounding of β_3 relies on the following steps:

- The quantity $\tilde{\mathbf{E}}_h^{n+1}$ is defined as the solution of the fully-discrete scheme, with the semi-discrete solution at timestep t_n as the input data;

- ◇ The consistency error $\left\|\mathbf{E}_h\left(t_{n+1}\right)-\tilde{\mathbf{E}}_h^{n+1}\right\|$ is bounded by $C\Delta t^3\left\|\mathbf{E}\right\|_{\mathcal{C}^3([0,T],L^2\Omega^3)}$ using Taylor expansions;
- ◇ This result is used to bound terms of the form $\left\|\mathbf{E}_h\left(t_{n+1}\right)-\mathbf{E}_h^{n+1}\right\|$ via the definition of a discrete error energy and using similar arguments as for the fully-discrete stability proof;
- ◇ Finally, the error energy contributions, similar to β_3, are bounded using the latter results.

Combining the estimations of β_1, β_2 and β_3, one obtains the following convergence result under CFL condition for centered fluxes:

$$\begin{aligned}&\max_{n\in[0,N]}\left(\left\|\mathbf{H}\left(t_{n+\frac{1}{2}}\right)-\mathbf{H}_h^{n+\frac{1}{2}}\right\|^2+\left\|\mathbf{E}\left(t_n\right)-\mathbf{E}_h^n\right\|^2\right)^{\frac{1}{2}}\\&\qquad\leq C\left(\Delta t^2+h^{\min(s,p)}\right)\left(\left\|(\mathbf{H},\mathbf{E})\right\|_{\mathcal{C}^3\left([0,T],L^2(\Omega)^6\right)}+\left\|(\mathbf{H},\mathbf{E})\right\|_{\mathcal{C}^0\left([0,T],H^{s+1}(\Omega)^6\right)}\right).\end{aligned}$$

As before, the same results can be obtained for Drude and generalized dispersive model, both for Leap-Frog [VKLS13] and Runge-Kutta [LSV].

4 Technicalities

4.1 Domain truncation

In a vast majority of situations, the considered physical devices have a very limited extension in the 3D space. From a numerical point of view, only a restrained region around the device is usually relevant, and it is necessary to truncate the computational domain without affecting the accuracy of the results. Limiting the size of the considered system also has an obvious impact on the memory and time required to compute the problem. The method used for the truncation must totally absorb all radiations crossing the frontier of the domain, for all wavelengths, polarization, and incidence angle on the boundary.

Two main types of methods are available to achieve this result. The first type consists in implementing a special boundary condition, called Absorbing Boundary Conditions (ABC), on the exterior surface of the domain. These conditions authorize waves to leave the domain, but no incoming wave is permitted. The other possibility is to exploit a special volume around the physical system, in which Maxwell's equations are modified to strongly damp all the waves that travel through it. This class of methods ensures that no spurious reflections occurs at the interface between the physical space and the damping volume, which is why they are called Perfectly Matched Layers (PML). In the following, a short presentation of ABC is made, followed by the assessment of a particular class of PMLs.

4.1.1 Absorbing boundary conditions

ABC were previously introduced in section 3.1.7, where expression of the first-order Silver-Müller condition was stated. Their proper derivation rely on the expansion theorem for electromagnetic fields, due to Wilcox [Wil56]. This theorem states that, outside a sphere enclosing all the scatterers, any radiating solution to Maxwell's equations can be expanded in the following form:

$$\mathbf{E}(\mathbf{r}) = \frac{\mathrm{e}^{-ikr}}{r} \sum_{n=0}^{\infty} \frac{\mathbf{A}_n(\theta, \phi)}{r^n}. \tag{4.1}$$

Then, an n^{th} order ABC is obtained by finding an operator that cancels the first n terms of (4.1). This work was done by Webb and Kanellopoulos in [WK89]. The first-order operator they obtain is nothing else than the Silver-Müller condition (SMC) on the domain boundary:

$$\mathbf{n} \times (\mathbf{E} + Z\,(\mathbf{n} \times \mathbf{H})) = \mathbf{o},$$

where $Z = \frac{1}{Y} = \sqrt{\mu\varepsilon}$. Consider the upwind flux formulation, presented earlier in section 3.1.3, on a face a_{il} of the mesh:

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

Now, suppose the medium is homogeneous, *i.e.* $Y_i = Y_l = Y$. Then:

$$\mathbf{E}_* = \frac{1}{2} \left(\{\mathbf{E}\}_{il} + \frac{\alpha}{Y} \mathbf{n} \times [\![\mathbf{H}]\!]_{il} \right).$$

Given that $Y = \frac{1}{Z}$, one obtains, for a fully upwind scheme ($\alpha = 1$):

$$\mathbf{E}_* = \frac{1}{2} \left(\{\mathbf{E}\}_{il} + Z\mathbf{n} \times [\![H]\!]_{il} \right) = \frac{1}{2} \underbrace{(\mathbf{E}_i - Z\mathbf{n} \times \mathbf{H}_i)}_{\text{Outgoing wave}} + \frac{1}{2} \underbrace{(\mathbf{E}_l + Z\mathbf{n} \times \mathbf{H}_l)}_{\text{Incoming wave}}.$$

One immediately sees that the SMC corresponds to the canceling of the incoming wave term in the flux expression, which is consistent with what was presented in section 3.1.7. A similar reasoning can be done for the **H** condition. The SMC being a first-order condition, it perfectly absorbs normally-incident plane waves, but for increasing angles of incidence, its performance rapidly decreases. The reflection coefficient of the SMC is given by [CFS06]:

$$R_{\text{SMC}}(\theta) = \frac{1 - \cos(\theta)}{1 + \cos(\theta)},$$

where θ is the incidence angle calculated from normal incidence. Moving to higher-order ABC yields better-performing reflection coefficients [Bó7]:

$$R_{\text{ABCn}}(\theta) = \left(\frac{1 - \cos(\theta)}{1 + \cos(\theta)} \right)^n.$$

A plot of the reflection coefficients is given on figure 4.1. As expected, SMC requires a near-normal incidence to correctly absorb outgoing waves. Situation improves for ABC2, however at $\theta = \frac{\pi}{4}$ reflection are still non-negligible. Higher-orders progressively extend the θ range for which R is acceptable. Although it is theoretically possible to keep raising the truncation order, high-order ABC suffer from two major drawbacks: (i) the complexity of their implementation rises dramatically [WK89], the SMC being the only condition which implementation is straightforward, and (ii) near-grazing incidences cannot be well-absorbed, whatever the order of the condition.

In the general case, the ABC must be located relatively far from the scatterer, so the local curvature of the radiated wavefronts is approximately normal to the boundary of the domain. Obviously, better performances are obtained for spherical boundaries than cubic ones. For the SMC, a rule of thumb is to put the boundary approximately one wavelength away from the scatterer[1]. This usually leads to a severe overhead in computational time and memory consumption, since the extra volume of free space can become very large. For this reason, numerical tools often make use of PMLs to truncate computational domains.

[1] The considered wavelength must be the largest wavelength considered in the problem.

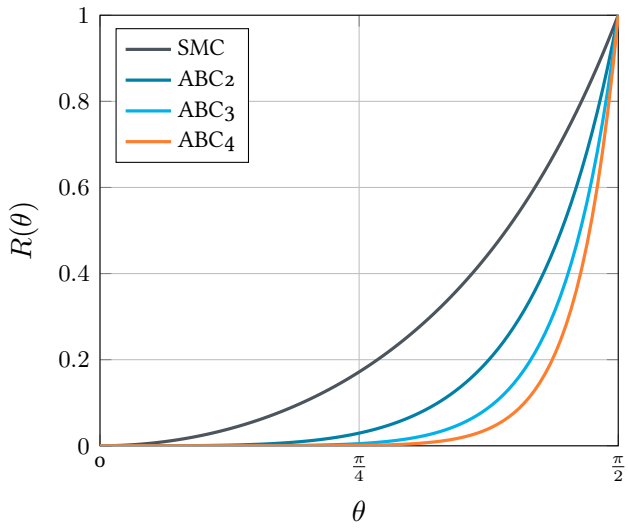


Figure 4.1 | Reflection coefficients for first to fourth-order absorbing boundary conditions. θ is the incidence angle calculated from normal incidence.

4.1.2 Perfectly matched layers

To overcome the limitations of ABC, Bérenger developed in 1994 a novel numerical concept to absorb the waves radiated from a system. Instead of using an actual boundary, he defined a volume surrounding the physical space in which the damping should occur progressively. This damping is introduced artificially by modifying Maxwell's equations in this specific zone of the domain. By correctly tailoring the artificial medium, it is possible to obtain no reflection at the interface between physical space and the absorbing layer. For this reason, Bérenger called them perfectly matched layers. The functioning of PMLs is depicted in figure 4.2: the outgoing wave propagates in the physical domain toward the PML, and crosses the interface. No reflexion occurs, and the wave continue to propagate in the PML, while being damped by the artificial medium. It eventually encounters the edge of the computational domain, which can be either PEC or ABC. In either case, the remaining of the wave is totally or partially reflected toward the domain, and is therefore damped a second time over the PML length. When it re-enters the physical zone, its amplitude is attenuated by several orders of magnitude. Hence, the error induced by the PMLs is supposed to be small enough not to lose the benefits of the high-order method. While standard ABC is only a "geometric" condition (*i.e.* it only becomes more efficient with a larger distance from the source), PMLs take advantage of the high-order spatial discretization, allowing higher levels of damping with increasing polynomial orders.

PMLs have evolved since Bérenger's implementation, and several varieties are now available [Bó7]. Two important versions in use for Maxwell's equations are the uniaxial PML (UPML), and the complex frequency-shifted PML (CFS-PML).

UPML

UPMLs are widely used in computational electromagnetics, for their implementation is quite straightforward, and their memory requirements remain low when compared to other types of PMLs. In order to achieve the perfectly matched feature for any incidence angle, it is not possible to suppose an isotropic medium in the absorbing layer. However, considering an anisotropic medium makes it feasible [Nieo9]:

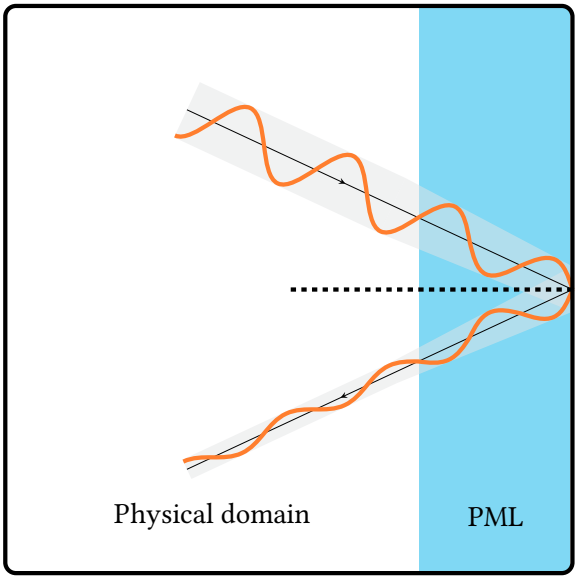


Figure 4.2 | General functioning of the PML. The outgoing wave propagates in the physical domain toward the PML, and crosses the interface. No reflexion occurs, and the wave continue to propagate in the PML, while being damped by the artificial medium. It eventually encounters the edge of the computational domain, (here PEC). In either case, the remaining of the wave is totally or partially reflected toward the domain, and is therefore damped a second time over the PML length. When it re-enters the physical zone, its amplitude is attenuated by several orders of magnitude. Hence, the error induced by the PMLs is supposed to be small enough not to lose the benefits of the high-order method.

$$\bar{\bar{\varepsilon}} \equiv \bar{\bar{\Lambda}}\,\varepsilon \;\text{ and }\; \bar{\bar{\mu}} \equiv \bar{\bar{\Lambda}}\,\mu \;\text{ with }\; \bar{\bar{\Lambda}} = \begin{bmatrix} \frac{s_y s_z}{s_x} & 0 & 0 \\ 0 & \frac{s_x s_z}{s_y} & 0 \\ 0 & 0 & \frac{s_x s_y}{s_z} \end{bmatrix},$$

where:

$$s_k(\omega) = 1 - \frac{\sigma_k}{i\omega}, \;\text{ with }\; k \in \{x, y, z\}\,.$$

Here, σ_k represents the loss rate of the PML in each direction. Modified equations for the PML region are easily obtained with the additional differential equation technique. Since they will not be exploited in this thesis, the complete derivation is not given here, and the reader is referred to [Nie09] for additional details. In counterpart to their low computational cost and easy implementation, the UPMLs present several limitations: (i) their theoretical damping can be very high, which can cause spurious reflections inside the physical domain (see section 4.1.3), (ii) their formulation must be modified if a dispersive or lossy material enters in contact with the PML, and in this case its memory consumption rises, and (iii) they only provide one parameter for optimization. Additionally, their performance is not as good as that of the CFS-PML [Kön11].

CFS-PML

Another approach to design PMLs is to exploit a complex stretch of coordinates in the spatial operator. This approach was successfully designed by Kuzuoglu and Mittra [KM96], although further developments showed that it was originally for an erroneous reason [BPG02]. It has been widely used in the FDTD

community since [RGoo]. Recently, König [Kön11] proposed a DGTD implementation of this PML, on which section 4.1.4 is based. The complex stretching is imposed by the following change of variables:

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

with:

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

As for the UPMLs, σ_k represents the loss rate of the PML. However, two new parameters are introduced. κ_k represents a real stretching factor, which effect is only to artificially lengthen the PML. This parameter is often exploited in FDTD to move away an ABC condition without modifying the computation grid. However, its effect is limited, since it can rapidly degrade the sampling of the fields inside the PML (see section 4.1.3). The α_k parameter is the actual frequency shift, since it moves the pole of $s_k(\omega)$ from $\omega_p = 0$ to $\omega_p = i\omega$. In the case $\alpha_k = 0$, linearly-growing instabilities in long-time computations can arise [BPGo2]. These instabilities usually take place when fields tend to be constant inside the PMLs (after the incident field has left the system, and the resonating structures are almost at rest).

4.1.3 Properties of PMLs

Since the study of PMLs is not at the heart of this manuscript, only the main ideas are exposed here, without proofs. However, all the derivations and detailed properties can be found in [Bó7]. Consider the very simple case of a one-dimensional plane wave propagating in the $x+$ direction toward a PML zone. In vacuum, the wave is described by:

$$\phi(x,t) = \phi_0(x,t) \equiv \mathrm{e}^{i(k_0 x - \omega t)}, \quad \text{with} \quad k_0 \equiv \frac{\omega}{c}, \tag{4.2}$$

where c is the wave speed. In the PML, the stretched coordinate transformation is applied, yielding a modified dispersion relation:

$$k = \frac{\omega s(\omega)}{c}.$$

For the CFS-PML, one obtains:

$$k = k_0 + k_1 + ik_2, \quad \text{with} \quad k_1 = \frac{\omega}{c}\left(\kappa - 1 + \frac{\alpha\sigma}{\alpha^2 + \omega^2}\right) \quad \text{and} \quad k_2 = \frac{\sigma\omega^2}{c\,(\alpha^2 + \omega^2)}. \tag{4.3}$$

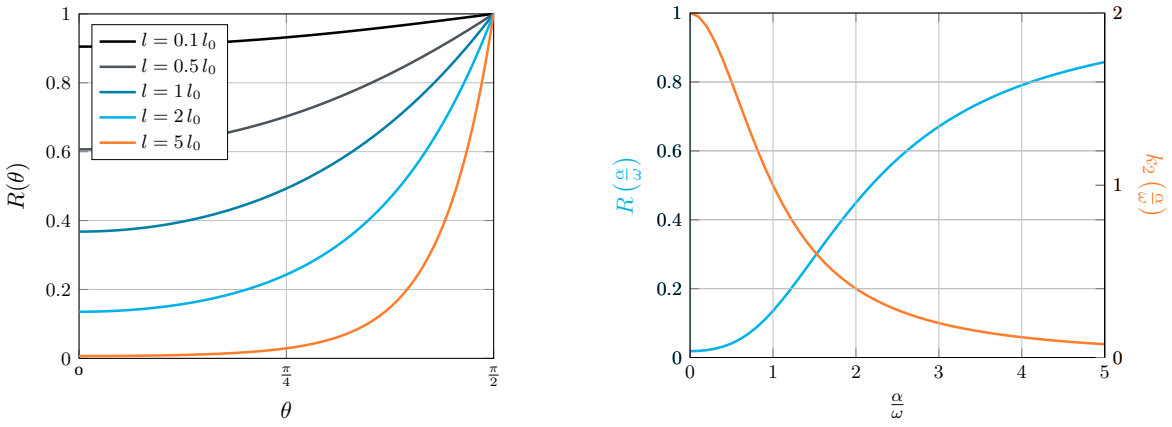
k_1 is a phase term induced by the complex shift. k_2, on the other hand, is the actual damping term of the PML. Indeed, plugging (4.3) in (4.2) yields:

$$\phi(x,t) = \phi_0(x,t)\,\mathrm{e}^{ik_1 x}\,\mathrm{e}^{-k_2 x}. \tag{4.4}$$

Given (4.4), it is easy to deduce that, for a PEC-backed PML of thickness l, the reflection coefficient in normal incidence is:

$$R_{\mathrm{PML}}(0) = \mathrm{e}^{-2k_2 l},$$

the 2 factor coming from the reflection of the damped wave on the back of the PML. In the more general case of an oblique incidence, one obtains:



(a) Increasing l

(b) Increasing $\frac{\alpha}{\omega}$

Figure 4.3 | Reflection coefficient of the CFS-PML for traveling waves in normal incidence. θ is the incidence angle calculated from normal incidence. In plot 4.3(a) the effect of an increasing thickness is showed. Parameters are $\alpha = 1$, $\omega = 1$, $\sigma = 1$ and $c = 1$. Plot 4.3(b) presents the influence of the quotient $\frac{\alpha}{\omega}$ on R and k_2. In this case, parameters are $l = 1$, $\sigma = 1$ and $c = 1$.

$$R_{\text{PML}}(\theta) = \mathrm{e}^{-2k_2 l\cos(\theta)}, \tag{4.5}$$

where θ is the incidence angle calculated from normal incidence. By observing (4.5) one sees that reflection from the PML layer can be lowered exponentially by increasing its width (see plot 4.3(a)), at a certain computational cost. It can also be lowered by choosing higher values of σ. From a numerical point of view, increasing σ implies that the numerical method must resolve a steeper exponential decay inside the layer without additional degrees of freedom. Hence, the fields inside the PML could become under-resolved by the spatial discretization for too high values of σ, producing spurious reflections inside the physical domain. Concerning this point, a good tradeoff must be found (see section 4.1.5). Regarding the values of α, one sees that $\alpha \gg \omega$ yields $k_2 \to 0$: in this case, frequencies that are low compared to α will not be absorbed correctly. On the other hand, $\alpha \ll \omega$ gives $k_2 \simeq \frac{\sigma}{c}$ (equal to 1 on the plot), which is the result for the standard Berenger PML. This case yields a low theoretical reflection, but in practice, the exponential decay may become too steep to be correctly resolved (see plot 4.3(b)). Intermediate values of α, on top of preventing long-time instabilities, allow a moderate absorption, neither too strong nor too weak, that can be resolved by the numerical method.

The case of evanescent waves remains to be considered. Since they are already exponentially decreasing, adding extra absorption to an evanescent wave entering a PML would most certainly end up in spurious reflections from under-resolved fields. In [Bó7], the author shows that the CFS-PML, contrarily to the standard PML, only provides a real stretch of coordinates for evanescent waves, proportional to k_1. Hence, an artificially extended domain is provided to evanescent waves to naturally decay in the PML.

4.1.4 CFS-PML for Maxwell's equations

The full derivation of the CFS-PML formulation is not reproduced here, and the reader is referred to [Kön11] for the full details. Starting from the coordinate stretch, the frequency-domain equation on the E_x component becomes:

$$-i\omega\varepsilon_r E_x = \frac{1}{s_y}\frac{\partial H_z}{\partial y} - \frac{1}{s_z}\frac{\partial H_y}{\partial z}.$$

The contributions coming from the ∂_y and the ∂_z will be kept in separate auxiliary variables, respectively noted G^E_{xy} and G^E_{xz}. After some algebraic manipulations, one obtains the following time-domain equations:

$$\begin{aligned}
\varepsilon_r\frac{\partial E_x}{\partial t} &= \frac{1}{\kappa_y}\frac{\partial H_z}{\partial y} - \frac{1}{\kappa_z}\frac{\partial H_y}{\partial z} - G^E_{xy} - G^E_{xz},\\
\frac{\partial G^E_{xy}}{\partial t} &= \frac{\sigma_y}{\kappa_y^2}\frac{\partial H_z}{\partial y} - \left(\alpha_y + \frac{\sigma_y}{\kappa_y}\right) G^E_{xy},\\
\frac{\partial G^E_{xz}}{\partial t} &= -\frac{\sigma_z}{\kappa_z^2}\frac{\partial H_y}{\partial z} - \left(\alpha_z + \frac{\sigma_z}{\kappa_z}\right) G^E_{xz}.
\end{aligned}$$

Equations for the other components are obtained by proper substitutions. As stated previously, two additional fields per physical component are necessary to account for the CFS-PML. However, as will be seen in the next section, in most cases a single layer of PML cells suffice to obtain a proper absorption of outgoing waves, thus limiting the computational overhead. Following the usual steps, a DG formulation is obtained:

$$\mathbb{M}_i^{\varepsilon_r}\frac{\partial E_{x,i}}{\partial t} = \frac{1}{\kappa_y}\mathbb{K}_i^y H_{z,i} - \frac{1}{\kappa_z}\mathbb{K}_i^z H_{y,i} - \mathbb{M}_i\left(G^E_{xy,i} + G^E_{xz,i}\right) - \sum_{l\in\mathcal{V}_i}\mathbb{S}_{il}\left[\mathbf{H}_*\times\mathbf{n}_{il}\right]_x, \tag{4.6}$$

$$\mathbb{M}_i\frac{\partial G^E_{xy,i}}{\partial t} = \frac{\sigma_y}{\kappa_y^2}\mathbb{K}_i^y H_{z,i} - \left(\alpha_y + \frac{\sigma_y}{\kappa_y}\right)\mathbb{M}_i G^E_{xy,i} - \frac{\sigma_y}{\kappa_y^2}\sum_{l\in\mathcal{V}_i}\mathbb{S}_{il}\left[\mathbf{H}_*\times\mathbf{n}_{il}\right]_{xy}, \tag{4.7}$$

$$\mathbb{M}_i\frac{\partial G^E_{xz,i}}{\partial t} = -\frac{\sigma_z}{\kappa_z^2}\mathbb{K}_i^z H_{y,i} - \left(\alpha_z + \frac{\sigma_z}{\kappa_z}\right)\mathbb{M}_i G^E_{xz,i} - \frac{\sigma_z}{\kappa_z^2}\sum_{l\in\mathcal{V}_i}\mathbb{S}_{il}\left[\mathbf{H}_*\times\mathbf{n}_{il}\right]_{xz}, \tag{4.8}$$

where $\left[\mathbf{H}_*\times\mathbf{n}_{il}\right]_{xy}$ corresponds to the y-derivative part of the x component of the flux (following the notation G^E_{xy}), *i.e.* the part involving the H_z term. Hence, one obtains:

$$\left[\mathbf{H}_*\times\mathbf{n}_{il}\right]_x = \left[\mathbf{H}_*\times\mathbf{n}_{il}\right]_{xy} + \left[\mathbf{H}_*\times\mathbf{n}_{il}\right]_{xz}. \tag{4.9}$$

Although both (4.7) and (4.8) contain additional flux terms, it is not necessary to calculate them during the update of G^E_{xy} and G^E_{xz}. By separately calculating and storing (only for the PML cells) the two stiffness and flux contributions during the electric field update, the computational cost of the CFS-PML remains limited.

Since the parameters of the PML are different in each direction of space, it is necessary in practice to differentiate three types of volumes. In the first kind, only one of the three damping parameters will be non-zero (for example, σ_x if it corresponds to a face perpendicular to the x axis): this results in six faces, two for each direction. In the second kind, two out of three parameters are non-zero, leading to twelves ridges in the cubic domain. The last kind corresponds to areas where the three parameters are non-zero, yielding eight corners in the final computational domain. A sketch of the 3D configuration is shown on figure 4.4

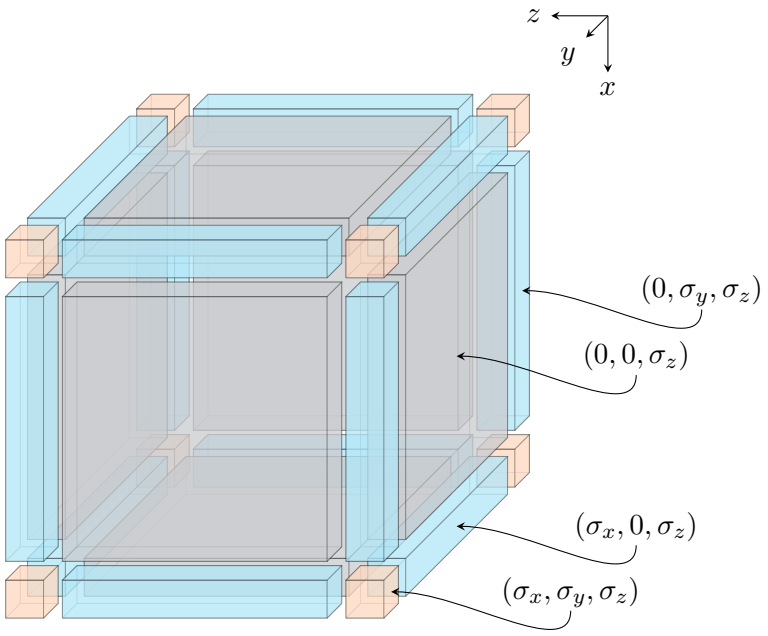


Figure 4.4 | PML configuration for a cubic domain. Grey faces correspond to one non-zero parameter, blue ridges to two non-zero parameters, and orange corners to three non-zero parameters. This results in 26 specific areas.

4.1.5 Performance assessment in the DGTD framework

In this section, the performances of the CFS-PML are assessed on a textbook case, similar to the one described in [Kön11]. At the center of a cubic domain of lateral size 2, a gaussian pulse in time and space is imposed, propagating in time toward outer space. The pulse is imposed *via* a source current, as described in 4.2.1. The computational domain is terminated with a single-cell layer of PML, backed with a SMC. In this case, the PML parameters are identical in all directions (*i.e.* $\sigma_x = \sigma_y = \sigma_z = \sigma$, and the same for α). The domain is divided in small cubes, which are in turn divided in six identical tetrahedra of side length 0.5. The resulting mesh contains 1331 vertices and 6000 tetrahedra. During the simulation, fields are recorded at a probe point located in $(1.5, 1.5, 1.5)$. To obtain a reference solution, a first possibility would be to exploit an exact solution of the problem. However, the error calculated in this procedure would incorporate both the error due to the spatial and temporal discretization, and the error due to the PML. Hence, a numerical reference solution is preferred. It is obtained by computing the pulse propagation on a very large PEC-backed domain, large enough so that the waves reflected on the boundary do not have the time to travel back to the center of the domain. The large domain has the same spatial discretization as the small one, and therefore the same timestep is used in both cases. After the computation, the error due to reflections coming from the PML is evaluated at the chosen probe point as:

$$\Delta = \frac{\max\limits_t |E_z^{\text{num}} - E_z^{\text{ex}}|}{\max\limits_t |E_z^{\text{ex}}|}.$$

This error is plotted on figure 4.5 for different combinations of (σ, α), with polynomial orders ranging from 1 to 4. For this configuration, $\kappa = 1$. It is visible that higher polynomial orders allow a better resolution of the exponential decay that takes place in the PML, hence leading to less spurious reflections

inside the physical domain. For a fixed value of α, one sees the existing balance between absorption and resolution of the exponential decay. When σ is too low, the decay is very well resolved, but the absorption is not important enough. Rising σ eventually leads to an optimum value for which the reflections in the physical domain are minimal. Moving to higher values of σ re-introduces spurious reflections caused by the under-resolved field variations in the PML.

The overhead in terms of memory and CPU induced by one layer of PMLs around the domain is calculated and averaged over the four polynomial orders. One layer of PMLs introduced approximately 18.9 % of additional CPU time, and required 7.2 % more memory than the same configuration with the PMLs replaced by vacuum. It must be kept in mind that here, the single PML layer holds more than 50 % of the total number of tetrahedra: this is a very high value due to the homogeneous meshing, and tends to be smaller in realistic configurations.

For comparison, the same test-case is considered, with a SMC condition instead of the PML. The distance d from the source to the boundary is progressively enlarged, and at each step the memory consumption, the CPU time and the reflection error are computed. Results are presented on figure 4.6. Since the SMC is a geometric condition, the errors obtained are almost independent of the polynomial order. The CPU time and memory consumption, however, are not. In both cases, it makes no doubt that the PML is profitable to the performance of the computations.

4.2 Sources and TF/SF formulation

4.2.1 Sources

A good control of the properties of incident fields is of major concern, since the physical response of a nanophotonic system depends, in the first place, on how it is excited. In this thesis, only plane waves, dipoles and waveguide modes will be considered. However, a wide range of sources is available from realistic physical applications, including for example laser beams [NH07].

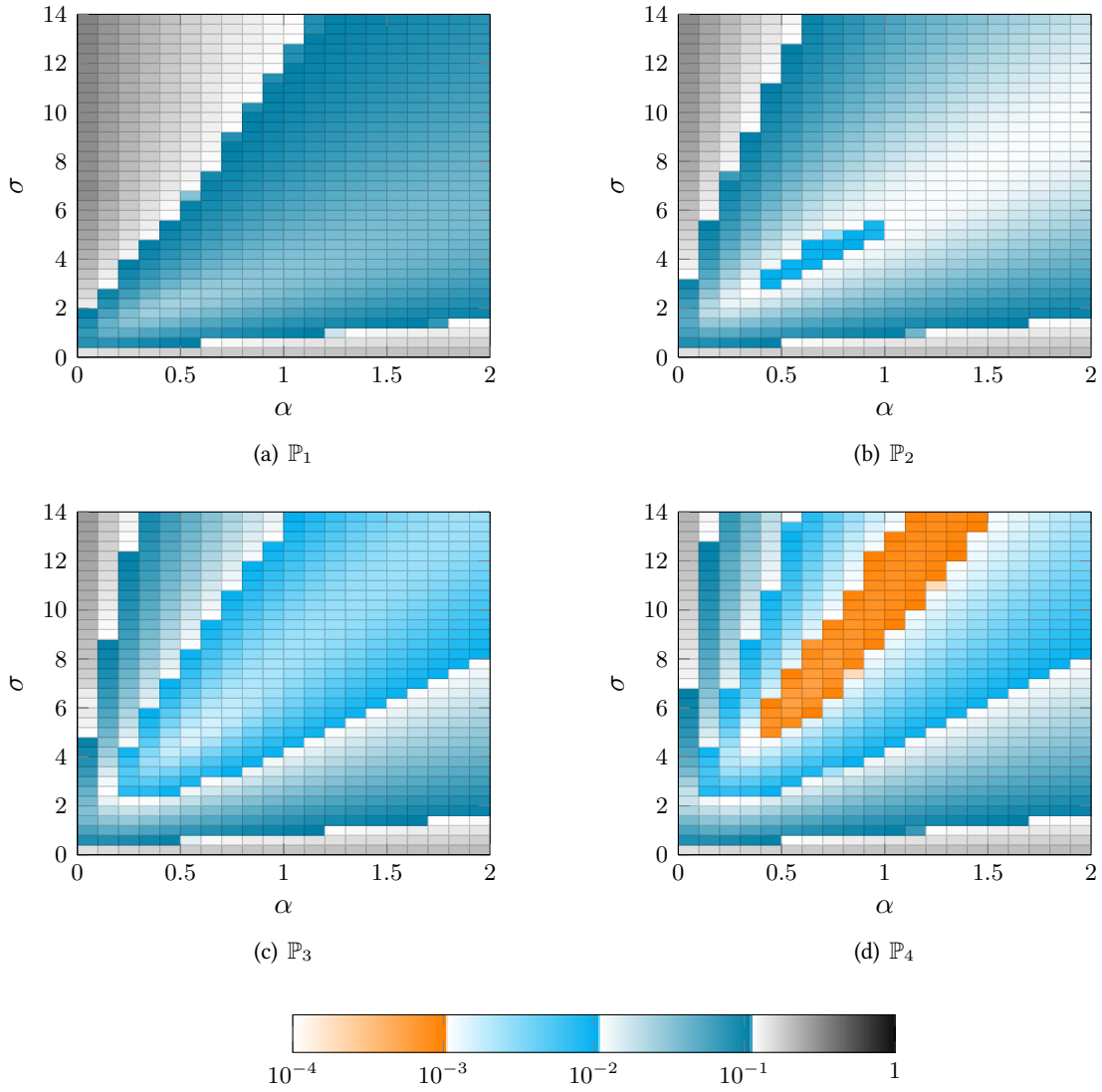
Plane waves

Plane waves are the most simple kind of source. Although they only correspond to an asymptotic physical configuration (*i.e.* any radiating source propagating on a sufficiently large distance should resemble a plane wave), they are often used in numerical electromagnetics to determine the fundamental properties of a physical system. The spatial profile of plane waves was already described in section 2.1.4. However, the choice of the time dependence is crucial. Indeed, on the spectral profile of the source depends the modes and/or resonances that will be excited or not in the physical system. The most basic kind of time dependence is the monochromatic plane wave, which expression is, in the temporal domain:

$$\mathbf{E}(t) = \mathbf{E}_0 \sin(\omega_0 t). \tag{4.10}$$

It is obvious that the Fourier transform of (4.10) is proportional to the Dirac function δ_{ω_0}. Although this type of source can be useful in several cases (see for example section 8.3), running a whole time-domain simulation only to obtain the response of the system at one frequency may not be worth it, and frequency domain methods may be more suited. To obtain a wider frequency spectrum, a usual time dependency is:

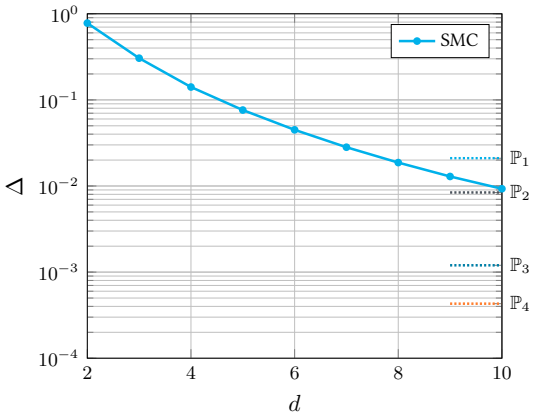
$$\mathbf{E}(t) = \mathbf{E}_0 \sin(\omega_0 (t - t_0)) \, \mathrm{e}^{-\frac{(t-t_0)^2}{2\sigma^2}}, \tag{4.11}$$



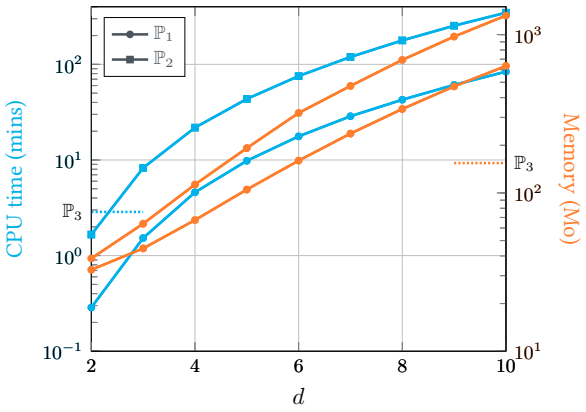
(a) $\mathbb{P}_1$ (b) $\mathbb{P}_2$

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

Figure 4.5 | Error due to the CFS-PML for polynomial orders ranging from 1 to 4, with a single cell layer of PML. In this case, $\kappa = 1$.



(a) Error induced by the Silver-Muller condition for various domain sizes compared to the optimal errors obtained with one layer of CFS-PML on a fixed-size domain



(b) CPU time and memory consumption with the Silver-Muller condition for various domain sizes compared to those of the CFS-PML on a fixed-size domain

Figure 4.6 | Performance of the Silver-Muller condition compared to the CFS-PML. The distance d from the source to the boundary is progressively enlarged, and at each step the memory consumption, the CPU time and the reflection error are computed. Since the SMC is a geometric condition, the errors obtained on 4.6(a) are almost independent of the polynomial order. The error levels obtained with the PML on the small domain described previously are noted with their approximation order on the right of the plot. The CPU time and memory consumption for the SMC are plotted (for $\mathbb{P}_1$ and $\mathbb{P}_2$) as functions of d on 4.6(b), and compared to those required with the PML in the $\mathbb{P}_3$ case.

which is a Gaussian function of width σ centered around t_0, modulated by a sine function. This function has several properties:

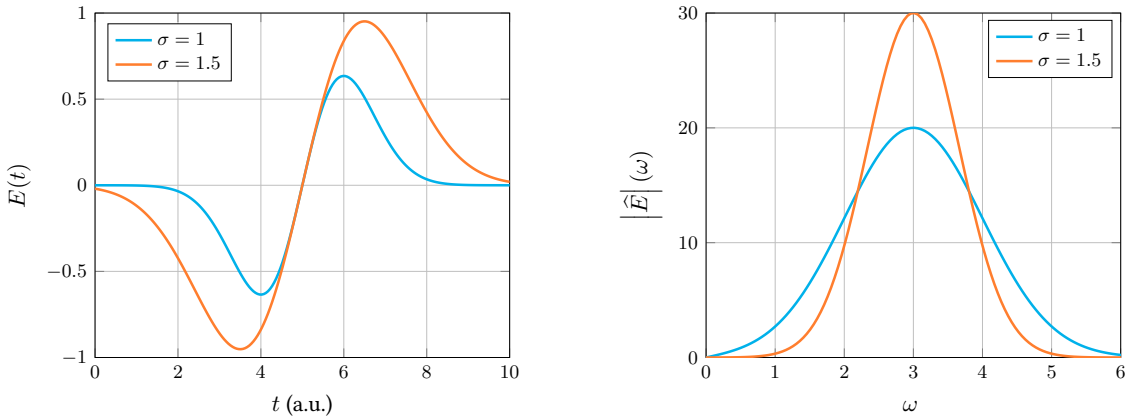
- For an appropriate choice of t_0, $\mathbf{E}(0) \simeq \mathbf{o}$;
- In any case, for t sufficiently large, $\mathbf{E}(t)$ decays to $\mathbf{o}$;
- Its Fourier transform is known:

$$\widehat{\mathbf{E}}(\omega) = \mathbf{E}_0 \frac{i\sigma}{2} \sqrt{\frac{\pi}{2}}\, \mathrm{e}^{i\omega t_0} \left(\mathrm{e}^{-\frac{\sigma^2(\omega-\omega_0)^2}{2}} - \mathrm{e}^{-\frac{\sigma^2(\omega+\omega_0)^2}{2}} \right), \tag{4.12}$$

  which means that such a pulse traveling through a structure will excite it on a wideband of frequencies, and not just on a single one as before (see figure 4.7);
- Direct control of the spectral profile of the pulse is available by tuning the parameter σ. The shorter the time-domain pulse, the wider the frequency-domain spectrum. From a practical point of view, there is a tradeoff between the length of the pulse that will travel through the domain (*i.e.* the longer the pulse, the more expensive the computation), and the frequencies that will be excited in the physical system. Choosing a very short pulse may not be a cheap solution, since potentially uninteresting resonances may be excited and resonate for a very long time, thus leading to an extended computational time.

Dipoles

Dipoles are commonly used to model sources of finite extension in space (which is not the case for plane waves). Two different techniques are available in the DGTD framework to impose such a source. The first



(a) Time-domain

(b) Frequency-domain

Figure 4.7 | 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.

one consists in approximating the source point by a steep gaussian. This can be done by adding a source current to one of the components of the electric field, in the FDTD soft source fashion [CBB09][2]. This method can be fairly easily implemented, but it is approximative, and the resolution of the very intense fields in the vicinity of the source origin requires a very fine local meshing. Recall the normalized **E** field evolution Maxwell's equation with a current source:

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

Integrating between $t = 0$ and t gives:

$$\mathbf{E}(\mathbf{x}, t) - \mathbf{E}(\mathbf{x}, 0) = \frac{1}{\varepsilon_r}\left(\int_0^t \nabla \times \mathbf{H}(\mathbf{x}, u)du - \int_0^t \mathbf{J}_s(\mathbf{x}, u)du\right).$$

Hence, to obtain a virtual electric field source of a chosen form $\mathbf{E}_s(\mathbf{x}, t)$, one should apply a current source which time dependence is proportional to the time derivative of $\mathbf{E}_s(\mathbf{x}, t)$. For example, if one wants a source of the following form (the origin of the source is taken at (x_s, y_s, z_s)):

$$\mathbf{E}_s(\mathbf{x}, t) = \mathrm{e}^{-\frac{(t-t_0)^2}{2\sigma^2}}\,\mathrm{e}^{-\left((x-x_s)^2+(y-y_s)^2+(z-z_s)^2\right)},$$

then the proper current source is:

$$\mathbf{J}_s(\mathbf{x}, t) = \varepsilon_r \frac{(t-t_0)}{\sigma^2}\,\mathrm{e}^{-\frac{(t-t_0)^2}{2\sigma^2}}\,\mathrm{e}^{-\left((x-x_s)^2+(y-y_s)^2+(z-z_s)^2\right)}.$$

The second method consists in imposing the exact electric and magnetic fields *via* the numerical flux on a TF/SF interface (see section 4.2.2). With this method, it is not necessary to discretize the high intensity fields close to the origin of the dipole, since the resulting electric and magnetic fields are imposed further away from it, where the interface is defined. Therefore, it leads to a more accurate approximation of the dipolar source. The calculations of the electric and magnetic fields to be imposed on the TF/SF interfaces

[2] Note that, if this method is here used to approximate a dipole, it is generic and can be used for a lot of other time and space dependences.

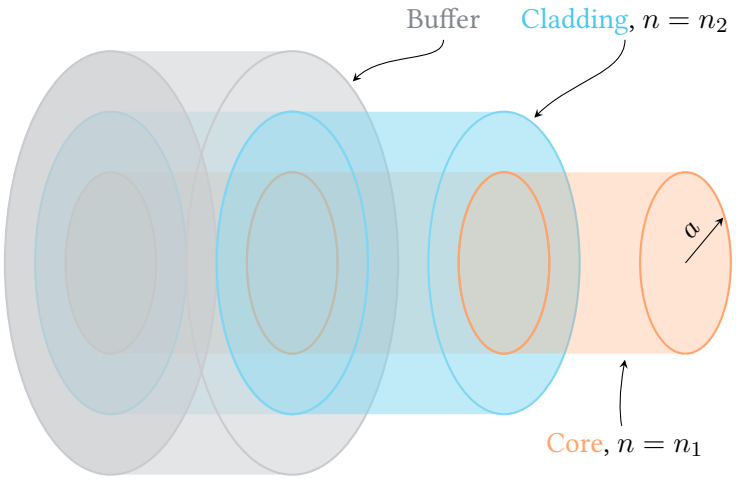


Figure 4.8 | Composition of a step-index optical fiber. Typical values for the radii are 3 to 10 μm for the core, around 130 for the cladding, and around 250 for the buffer. A set of typical values can be $a = 5$ μm, $n_1 = 1.455$ and $n_2 = 1.45$.

in the general case are detailed in [Nie09], and are not reproduced here. Here, the solution for a z-oriented dipole with a general time dependence $p(t)$ is given in spherical coordinates:

$$\mathbf{E}_r(r,\theta,\phi,t) = \frac{2Zc}{4\pi\varepsilon_0}\frac{\cos\theta}{r}\left(\frac{\dot{p}(\bar{t}\,)}{cr} + \frac{p(\bar{t}\,)}{r^2}\right),$$
$$\mathbf{E}_\theta(r,\theta,\phi,t) = \frac{Zc}{4\pi\varepsilon_0}\frac{\sin\theta}{r}\left(\frac{\ddot{p}(\bar{t}\,)}{c^2} + \frac{\dot{p}(\bar{t}\,)}{cr} + \frac{p(\bar{t}\,)}{r^2}\right),$$
$$\mathbf{H}_\phi(r,\theta,\phi,t) = \frac{1}{4\pi}\frac{\sin\theta}{r}\left(\frac{\ddot{p}(\bar{t}\,)}{c} + \frac{\dot{p}(\bar{t}\,)}{r}\right),$$

with $\mathbf{E}_\phi = \mathbf{H}_r = \mathbf{H}_\theta = 0$ and $\bar{t} = t - \frac{r}{c}$.

Step-index optical fiber modes

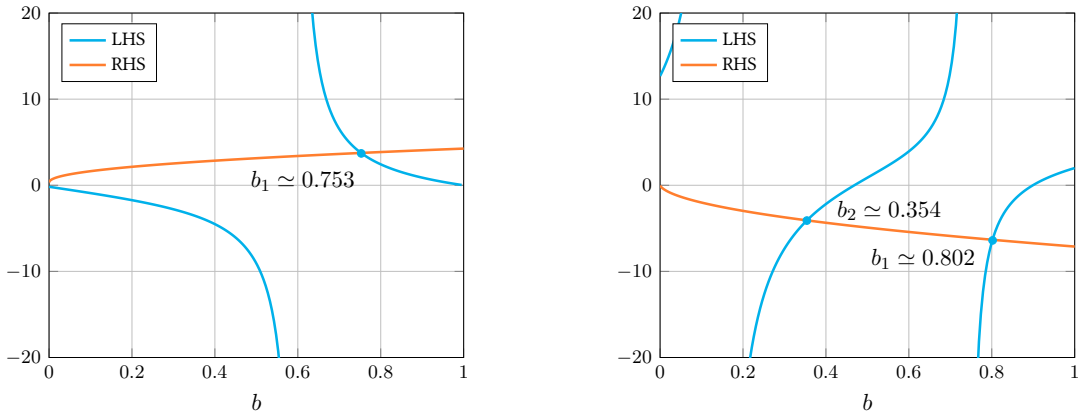
Among the large variety of waveguides, we consider a step-index optical fiber (see sketch in figure 4.8), for which the core and the cladding have slightly different optical indices ($n_1 \gtrsim n_2$, where n_1 is the core index, and n_2 the cladding index). The modes allowed to travel in such fibers are, in polar coordinates, of the form:

$$\mathbf{E}(r,\phi,z) = \mathbf{e}(r)\,\mathrm{e}^{-il\phi}\,\mathrm{e}^{-i\beta z}, \tag{4.13}$$

where l is an integer, and β remains to be determined. Skipping the derivations (see for example [Buc04]), solutions for the radial part are standard Bessel functions (J_l) in the core, and modified Bessel functions (K_l) in the cladding:

$$\mathbf{e}(r) = \mathbf{E}_0\frac{J_l\left(U\frac{r}{a}\right)}{J_l(U)} \qquad \text{for } \; 0 \le r < a,$$
$$\mathbf{e}(r) = \mathbf{E}_0\frac{K_l\left(W\frac{r}{a}\right)}{K_l(W)} \qquad \text{for } \; a < r,$$

with



(a) $a = 5\ \mu$m, $l = 0$ (b) $a = 10\ \mu$m, $l = 1$

Figure 4.9 | Solutions of the mode equation for step-index optical fibers. The plots represent the left hand side (LHS) and right hand side (RHS) of the transcendental equation (4.14). In both cases, $\lambda_0 = 1\ \mu$m, $n_1 = 1.455$ and $n_2 = 1.45$. On the left panel, for $a = 5\ \mu$m and $l = 0$, a unique solution is found, yielding $\frac{\beta_1}{k_0} \simeq 1.453764$. Hence, only one mode is able to propagate in the fiber. On the right panel, for $a = 10\ \mu$m and $l = 1$, two solutions are obtained: $\frac{\beta_1}{k_0} \simeq 1.451772$ and $\frac{\beta_2}{k_0} \simeq 1.454009$.

$$U = a\sqrt{k_0^2 n_1^2 - \beta^2} \quad \text{and} \quad W = a\sqrt{\beta^2 - k_0^2 n_2^2}.$$

In the latter expressions, $k_0 = \frac{2\pi}{\lambda_0}$ is the wavenumber of the source in vacuum[3], λ_0 is the wavelength in vacuum, and n is the optical index, equal to n_1 or n_2 depending on the position. One can feel that, because of the index jump, the mode propagating inside the fiber will have an intermediate wavenumber between those of the core ($k_1 = n_1 k_0$) and the cladding ($k_2 = n_2 k_0$). β can be obtained by considering the continuity of the fields at the core/cladding interface. The calculation yields the following equalities:

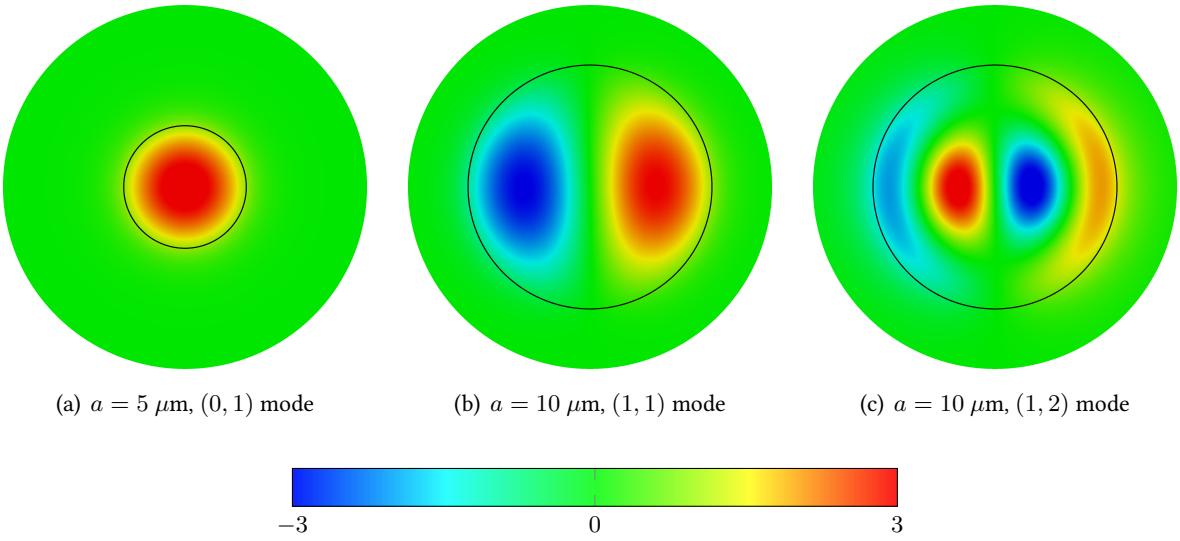
$$\begin{aligned} \theta \frac{J_1(\theta)}{J_0(\theta)} &= \eta \frac{K_1(\eta)}{K_0(\eta)} \qquad \text{for } l = 0, \\ \theta \frac{J_{l-1}(\theta)}{J_l(\theta)} &= -\eta \frac{K_{l-1}(\eta)}{K_l(\eta)} \qquad \text{for } l \geq 1, \end{aligned} \tag{4.14}$$

with

$$\theta = V\sqrt{1-b}, \quad \eta = V\sqrt{b}, \quad V = \sqrt{U^2 + W^2}, \quad \text{and} \quad b = \frac{W^2}{V^2}.$$

The resolution of the transcendental equation (4.14) finally provides all the information necessary to compute an incident fiber mode. Depending on the parameters, (4.14) may have zero, one or more solutions for β. These solutions are numbered with the integer $m = 1, 2, 3, ..$, and the m^{th} solution then corresponds to the (l, m) mode. A few examples of the solution of the transcendental equation (4.14) are presented on figure 4.9, while the resulting fiber modes are plotted on figure 4.10. As can be seen, the field slightly extends into the cladding, although it is rapidly damped.

[3]The field inside the fiber is generated by an emitting diode, for which the parameters are imposed relatively to vacuum.



(a) $a = 5$ μm, $(0, 1)$ mode (b) $a = 10$ μm, $(1, 1)$ mode (c) $a = 10$ μm, $(1, 2)$ mode

Figure 4.10 | Examples of electric field map in fiber modes. The radius and mode numbers (l, m) are given in each subcaption. The black ring indicates the limit between the core and the cladding, and the geometrical scaling is identical for all three figures (the total radius is 15 μm). As can be seen, the field slightly extends into the cladding, although it is rapidly damped.

4.2.2 TF/SF formulation

The way of imposing the sources inside a physical domain has not been discussed yet. A first, simple possibility is to use directly the ghost cells of the SMC (see section 3.1.7), allowing an imposed incident field to enter the domain. This is a viable solution, which however, does not apply in the presence of PMLs. Another possibility is to define an additional artificial contour inside the physical domain, on which the field could be imposed directly (see figure 4.11).

Consider the following splitting of the electric field:

$$\mathbf{E}_{\text{tot}}(\mathbf{x}, t) = \mathbf{E}_{\text{inc}}(\mathbf{x}, t) + \mathbf{E}_{\text{sca}}(\mathbf{x}, t), \tag{4.15}$$

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. The incident field is known, since it is imposed by the user. Consider now a splitting of the computational domain in two parts, as presented on figure 4.11: a convex region, in which the total field is computed, encloses the physical device, while in the remaining of the domain, the scattered field is computed. The interface between these two regions is called the total field/scattered field (TF/SF) interface. In each region, the DGTD formulation derived earlier (see equation 3.9) is valid, and no modification is required. At the interface, however, the computation of the flux between the two regions is modified. Consider an interface between two cells, such that the local cell T_i is a total field cell, while the neighbor cell T_l is a scattered field cell (situation is shown on figure 4.12). 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 (4.15), one easily shows that the right flux can be calculated as follows:

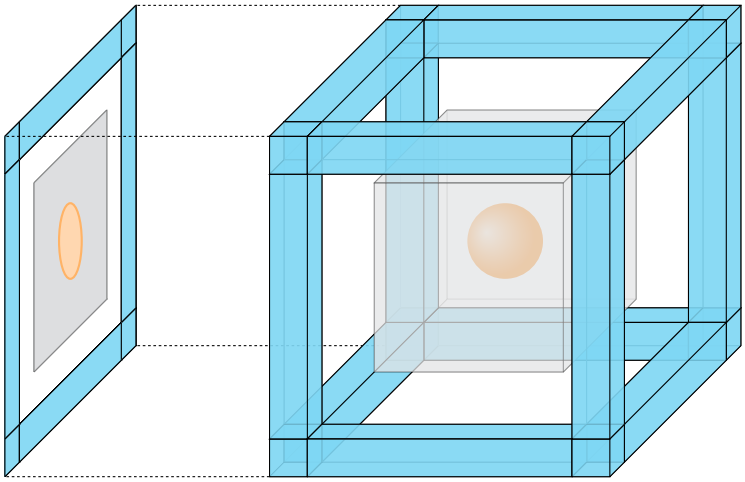


Figure 4.11 | Mesh configuration including a scatterer, a TF/SF interface, and a PML boundary layer. The scatterer, in orange, is enclosed by the TF/SF interface, in light gray. The faces of the PML (in light blue) are removed for clarity.

$$\mathbf{E}_{*,\text{tot}} = \underbrace{\frac{1}{Y_i + Y_l}\left(\{Y\mathbf{E}\}_{il} + \alpha\mathbf{n} \times [\![\mathbf{H}]\!]_{il}\right)}_{\mathbf{E}_*} + \underbrace{\frac{1}{Y_i + Y_l}\left(Y_l\mathbf{E}_{\text{inc}} + \alpha\mathbf{n} \times \mathbf{H}_{\text{inc}}\right)}_{\mathbf{E}_{*,\text{inc}}}, \tag{4.16}$$

the minus sign coming from the definition of the jump. Symmetrically, the upwind flux for cell T_l in the scattered field region is:

$$\mathbf{E}_{*,\text{sca}} = \mathbf{E}_* - \mathbf{E}_{*,\text{inc}}. \tag{4.17}$$

Similar derivations can be obtained for the other curl equation. From (4.16) and (4.17), it is obvious that 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 has several uses: (i) it can be used as a simple interior boundary to impose incident fields in cases where PMLs are used, (ii) it allows to observe both the total field in the vicinity of the scatterer and the scattered field far from it in a single run, and (iii) it is useful to compute relevant scattering quantities, such as cross-sections, without using additional interior countours (see section 4.4).

4.3 Fourier transform

Although the numerical method considered here is in the time-domain, many physical quantities, such as cross-sections (see section 4.4.1) or reflection and transmission coefficients (see section 4.4.3), are better defined in the frequency domain. Hence, an efficient discrete Fourier transform must accompany the time-domain solver, in order to extract the Fourier fields on relevant surfaces. As before, for a time-dependent variable $\phi(t)$ being at least in $L^1(\mathbb{R})$, its Fourier counterpart is noted $\widehat{\phi}(\omega)$, and the two are connected *via* the Fourier transform:

$$\widehat{\phi}(\omega) = \int_{-\infty}^{+\infty} \phi(t)\, \mathrm{e}^{-i\omega t} dt. \tag{4.18}$$

A discrete version of (4.18) is:

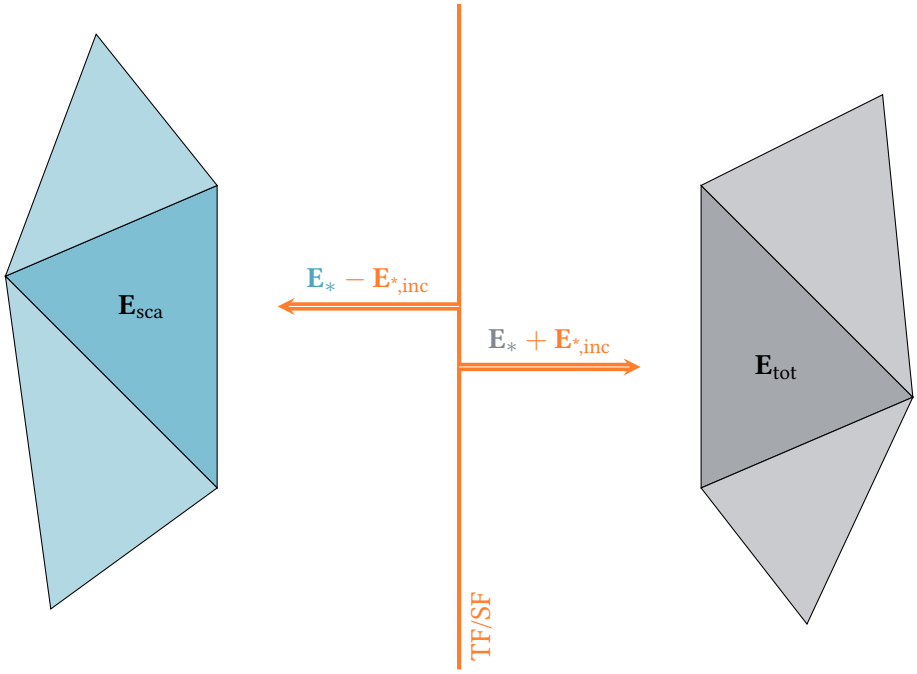


Figure 4.12 | TF/SF interface with modified fluxes.

$$\widehat{\phi}(\omega) \simeq \sum_{t_j = t_i + j\Delta t} \phi(t_j)\, \mathrm{e}^{-i\omega t_j}\, \Delta t. \tag{4.19}$$

Having selected a frequency range of interest $[\omega_i, \omega_f]$ and a frequency step $\Delta\omega$, one must loop over the $\omega_k = \omega_i + k\Delta\omega$ at each time step, and update (4.19) at each degree of freedom of the selected faces and/or cells:

$$\widehat{\phi}(\omega_k) \leftarrow \widehat{\phi}(\omega_k) + \phi(t_j)\, \mathrm{e}^{-i\omega_k t_j}\, \Delta t.$$

A few remarks arise about this method:

- The accuracy of the algorithm depends only on Δt, which is usually small enough given the spatial resolutions required by nanophotonics problems;
- The spectral resolution depends on $\Delta\omega$ only;
- The memory requirements depend on ω_i, ω_f and $\Delta\omega$, but are independent of the total simulation time;
- The discrete Fourier transform (DFT) of a non-periodic signal (such as (4.11) for example) usually requires that this signal starts with a zero amplitude, and that this amplitudes decays to zero again before the end of the DFT. Otherwise, the obtained spectrum is polluted by what is called spectral leakage, which is a consequence of the time window used to evaluate the DFT. Hence, for actual nanophotonics computations, it is necessary to use a time-domain pulse with a zero starting amplitude, and to wait for a sufficient decay of the field amplitudes before ending the computation. For certain configurations, this can lead to highly increased computational times, and therefore the relevant frequency range of the incident pulse must be chosen with care.

4.4 Relevant quantities in electromagnetic scattering

4.4.1 Cross-sections

To describe the power density vehicled by a propagating electromagnetic wave, it is common to use the Poynting vector, whose definition in time-domain is:

$$\boldsymbol{\pi}(\mathbf{x}, t) = \mathbf{E}(\mathbf{x}, t) \times \mathbf{H}(\mathbf{x}, t). \tag{4.20}$$

When dealing with the range of frequencies encountered in the field of nanophotonics, it seems quite obvious that any regular experimentation device cannot capture the time-domain dynamics of light. The right quantity to take into account is here the time-averaged Poynting vector:

$$\boldsymbol{\pi}(\mathbf{x}, \omega) = \frac{1}{2}\Re\left(\widehat{\mathbf{E}}(\mathbf{x}, \omega) \times \widehat{\mathbf{H}}^{*}(\mathbf{x}, \omega)\right), \tag{4.21}$$

which is not the Fourier transform of (4.20). Given that definition, consider the situation of figure 4.11, where the scatterer is totally enclosed by the TF/SF contour. Inside the total field region, the computed fields are $\mathbf{E}_{\text{tot}}$ and $\mathbf{H}_{\text{tot}}$ such 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 is imposed on the TF/SF interface, and therefore, in the scattered region, the computed fields are $\mathbf{E}_{\text{sca}}$ and $\mathbf{H}_{\text{sca}}$. From now on, S denotes the closed TF/SF surface. The absorbed energy is defined as:

$$W_{\text{abs}}(\omega) = -\int_S \boldsymbol{\pi}_{\text{tot}} \cdot \mathbf{n}, \tag{4.22}$$

with $\mathbf{n}$ the outward normal to surface S. In the absence of scatterer, all the energy that enters the total field region leaves it, and therefore $W_{\text{abs}} = 0$, since the different contributions of the integral will compensate. If a scatterer is added, then a part of the incoming energy may be absorbed, and hence not all the energy that enters S leaves it. The absorbed part therefore corresponds to the quantity (4.22). Similarly, the scattered energy is defined as:

$$W_{\text{sca}}(\omega) = \int_S \boldsymbol{\pi}_{\text{sca}} \cdot \mathbf{n}, \tag{4.23}$$

which would trivially be zero in the absence of a scatterer inside S. To eliminate the dependance of (4.22) and (4.23) on the amplitude of the incident field, one defines the absorption cross section and the scattering cross section, respectively:

$$C_{\text{abs}}(\omega) = -\frac{\int_S \boldsymbol{\pi}_{\text{tot}} \cdot \mathbf{n}}{S_{\text{inc}}}, \tag{4.24}$$

and:

$$C_{\text{sca}}(\omega) = \frac{\int_S \boldsymbol{\pi}_{\text{scat}} \cdot \mathbf{n}}{S_{\text{inc}}}, \tag{4.25}$$

where $S_{\text{inc}} = |\boldsymbol{\pi}_{\text{inc}}|$ is the incident power density in W.m^{-2}. Additionally, one defines the extinction cross section as:

$$C_{\text{ext}} = C_{\text{abs}} + C_{\text{sca}}. \tag{4.26}$$

Sometimes, physical cross-sections are normalized by the geometric cross-section of the scatterer. These quantities, noted Q_{ext}, Q_{abs} and Q_{sca}, are often called extinction, absorption and scattering efficiencies.

4.4.2 Volumetric absorption

The absorption due to lossy materials can also be computed with a volumetric method. Indeed, it is possible to evaluate the Ohm losses directly inside the material instead of computing the flux of the Poynting vector through a surface enclosing the scatterer. It can be shown that the power absorbed by the scatterer as ohmic losses is [LL60]:

$$P_{\text{Ohm}}(\omega) = \frac{\varepsilon_0 \omega}{2} \int_{\Omega_s} \Im(\varepsilon_r(\omega)) \left|\widehat{\mathbf{E}}(\mathbf{r}, \omega)\right|^2, \tag{4.27}$$

where Ω_s is the volume delimited by the scatterer. In the case of a single scatterer, the losses can either be computed *via* the surfacic method (with C_{abs}) or the volumetric method (with P_{Ohm}). Depending on the sizes and discretization of the scatterer and the TF/SF interface, the costs and accuracy of both methods can vary. The scattering regime also plays a role in the accuracy of both methods, as shown in [KM10].

4.4.3 Reflection and transmission

In the case of periodic structures, the definition of cross-sections does not make sense, since the TF/SF interface is composed of two infinite planes, numerically delimited by PBCs (see figure 4.13). However, two natural quantities can be computed in such cases: the reflection and the transmission coefficients, which are both frequency-dependent. When a scatterer is illuminated by an incident field, the first one represents the proportion of energy coming back due to the scatterer, while the second one represents the amount of energy that travelled through the system. Hence, one defines the reflection and tramission coefficients as, respectively:

$$R(\omega) = \frac{\displaystyle\int_{S_i} \boldsymbol{\pi}_{\text{sca}} \cdot \mathbf{n}}{\displaystyle\int_{S_i} \boldsymbol{\pi}_{\text{inc}} \cdot \mathbf{n}}, \tag{4.28}$$

and

$$T(\omega) = \frac{\displaystyle\int_{S_o} \boldsymbol{\pi}_{\text{tot}} \cdot \mathbf{n}}{\displaystyle\int_{S_i} \boldsymbol{\pi}_{\text{inc}} \cdot \mathbf{n}}, \tag{4.29}$$

where S_i and S_o are the planes defined on figure 4.13. Additionally, the absorption can be defined from R and T as:

$$A(\omega) = 1 - T(\omega) - R(\omega). \tag{4.30}$$

Obviously, the absorption is zero if only dielectric scatterers are present in the TF zone.

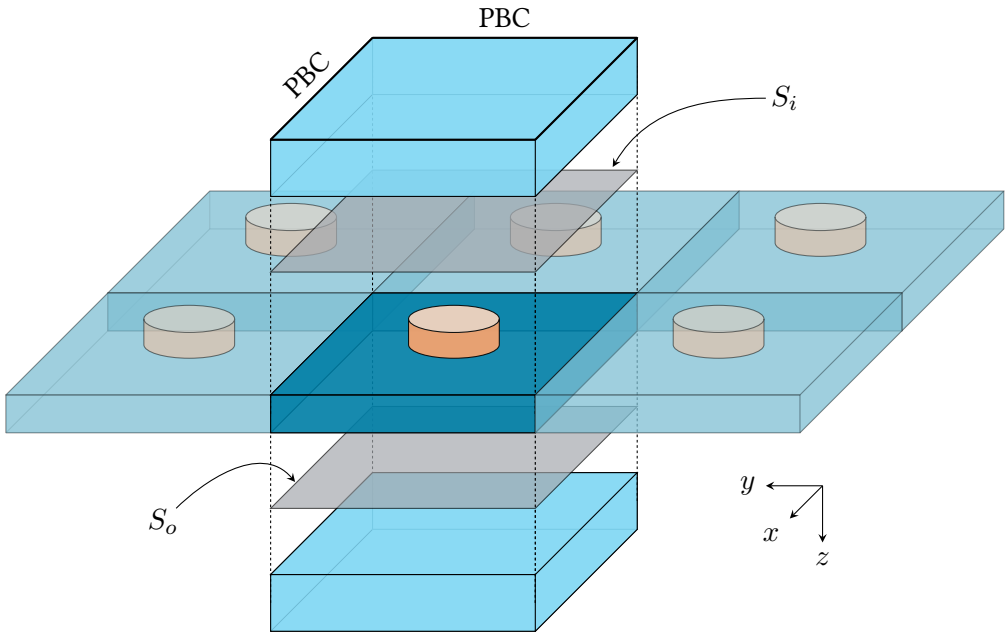


Figure 4.13 | Periodic array of scatterers on a metallic slab. The TF/SF planes, S_i and S_o, are respectively above and below the photonic device (see the axes). In the z direction, the domain is terminated by a PML layer, while it is periodic in the x and y directions.

4.4.4 Far field and radar cross-section

Although the quantities presented above can provide crucial informations on a given nanostructure, their knowledge seems, in most cases, insufficient. Indeed, any notion of directivity is lost in the computation of cross-sections because positive and negative contributions to the Poynting flux are averaged on a closed surface. A large chunk of information is lost as well in the case of reflection and transmission coefficients. However, for some applications, the directivity pattern of the considered nanostructure will play a crucial role in its effectiveness. A possible quantity to measure the directivity of a device (a nanoantenna, for example) is its radar cross section (RCS), which measures how well it can be detected from afar, in a given direction. In spherical coordinates, its definition is [TH05]:

$$\sigma_{\text{RCS}}(\theta,\phi) = \lim_{r\to\infty} 4\pi r^2 \frac{S_{\text{sca}}(r,\theta,\phi)}{S_{\text{inc}}(\theta_{\text{inc}},\phi_{\text{inc}})}, \tag{4.31}$$

where S_{inc} and S_{sca} are respectively the incident power density and the scattered power density seen at a distance r from the source. It is noticeable that σ_{RCS} does not depend on r, and is therefore a far field quantity. The θ_{inc} and ϕ_{inc} angles correspond to the angles of the incident field, which is then scattered unequally by the nanostructure in all directions (described by θ and ϕ), hence providing a directivity pattern. In the following, the spherical angles are those defined on figure 4.14.

Here, we follow the procedure described in [TH05] for the computation of the RCS. The main steps are reproduced, and the reader is referred to the aforementionned reference for further details. As for scattering cross-sections, the Fourier transforms of the scattered field must be computed on the TF/SF interface. Then, the equivalents currents (see the equivalence theorem in [TH05]) are computed on the TF/SF surface:

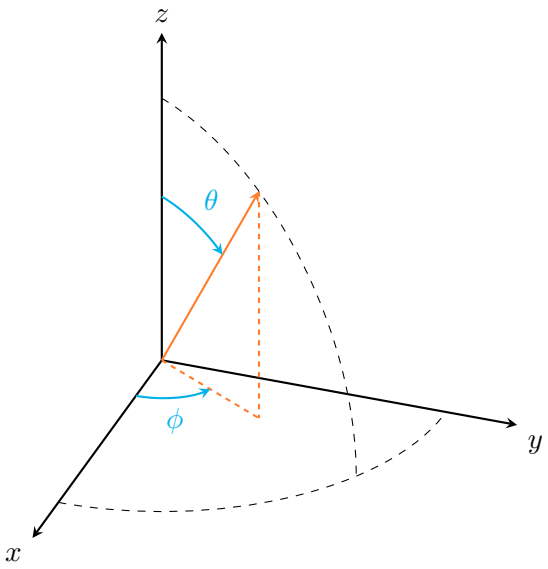


Figure 4.14 | Spherical coordinates system for the RCS computation.

$$\begin{aligned} \mathbf{J}_s(\omega) &= \mathbf{n} \times \widehat{\mathbf{H}}_{\text{sca}}(\omega) \\ \mathbf{M}_s(\omega) &= -\mathbf{n} \times \widehat{\mathbf{E}}_{\text{sca}}(\omega). \end{aligned}$$

In essence, the equivalence theorem implies that the actual source (for example the nanoantenna illuminated by the incident field) can be replaced by fictitious surfacic current sources, here denoted by $\mathbf{J}_s$ and $\mathbf{M}_s$. From these currents, the scattered field at any point can be computed by integrating the elemental sources over the closed surface S. This is done by computing the components of the field potentials[4] in the (θ, ϕ) direction:

$$\begin{aligned} N_\theta(\omega) &= \int_S \left(J_x \cos\theta \cos\phi + J_y \cos\theta \sin\phi - J_z \sin\theta\right) \dot{e}\, dS, \\ N_\phi(\omega) &= \int_S \left(-J_x \sin\phi + J_y \cos\phi\right) \dot{e}\, dS, \\ L_\theta(\omega) &= \int_S \left(M_x \cos\theta \cos\phi + M_y \cos\theta \sin\phi - M_z \sin\theta\right) \dot{e}\, dS, \\ L_\phi(\omega) &= \int_S \left(-M_x \sin\phi + M_y \cos\phi\right) \dot{e}\, dS, \end{aligned}$$

with:

$$\dot{e} = \mathrm{e}^{ik(x \sin\theta \cos\phi + y \sin\theta \sin\phi + z \cos\theta)}.$$

From the latter expressions, the RCS is deduced [TH05]:

$$\sigma_{\text{RCS}}(\theta, \phi) = \frac{k^2}{8\pi Z_0 S_{\text{inc}}} \left(|L_\phi + Z_0 N_\theta|^2 + |L_\theta - Z_0 N_\phi|^2\right). \tag{4.32}$$

[4] These potentials are simplified thanks to the far field assumption

5

Curvilinear elements

As shown previously, the standard DG method relies on a tessellation composed of straight-edged elements mapped linearly from a reference element (see section 3.1.5). However, for problems with curved interfaces or boundaries, such meshes represent a serious hindrance for the high-order convergence, since they limit the accuracy to second order in the spatial discretization. Thus, exploiting an enhanced representation of physical geometries is in agreement with the natural procedure of the DG method. There are several ways to account for curved geometries. One could choose to incorporate the knowledge coming from CAD in the method to design the geometry and the approximation : these methods are called isogeometric [HCB05], and have received a lot of attention recently. This naturally implies to have access to CAD models of the geometry. On the other hand, isoparametric usually rely on a polynomial approximation of both the boundary and the solution. This can be added fairly easily on top of existing implementations. Hereafter, we will focus on the latter type of method, since our goal is first to envisage the benefit of curvilinear meshes in nano-optics.

Early implementations of isoparametric elements have been made in the field of computational fluid dynamics by Bassi and Rebay for the 2D Euler equations : in [BR97], the authors exhibit cases where the physical meaning of the numerical solution is not consistant with reality unless a proper description of the boundaries by curved elements is used. In [SSS13], realistic situations of 3D Euler flows around airfoil profiles have been numerically studied; in these cases, the non-analytic nature of the geometry implies improved results for arbitrary orders of boundary approximation. In computational electromagnetics, curvilinear elements have also been put at use. In [Nie09], the authors present a 3D discontinuous Galerkin time-domain (DGTD) method exploiting a second-order mapping. A low storage version of the method has been proposed in [War10], where the additional memory cost due to curved elements is reduced by a modification of the basis functions. In [Fah11], the author points out the causes of suboptimal rates of convergence for 3D geometries when high-order mappings are considered (cubic and higher). Realistic 2D nanostructures have already proved to benefit from the use of curved elements, especially in the DGTD framework [HKGE10].

In this chapter, the benefits of a curvilinear DGTD approach for nano-optics are assessed. First, the necessary ingredients to the implementation and use of curvilinear tetrahedra in the DGTD framework are presented. Then, these procedures are validated with a textbook case. Finally, the practical interest of curvilinear elements in nano-optics is demonstrated with realistic geometries of nanocube absorbers.

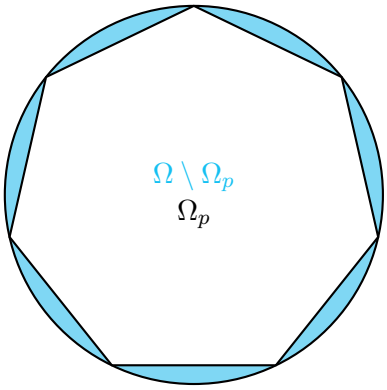


Figure 5.1 | Polygonal approximation of a domain with curved boundaries.

5.1 Formulation for curved elements

5.1.1 Intrinsic limitation of linear elements

Geometrical error

In most cases, FE formulations are solved on a polygonal approximation Ω_p of a convex physical domain Ω. Hence, $\partial\Omega \cap \partial\Omega_p$ is a set composed of a finite number of points (see figure 5.1). Consider the following generic continuous problem with homogeneous Dirichlet conditions and $f \neq 0$:

$$\begin{aligned} \mathcal{L}_\Omega(u) &= f \text{ on } \Omega, \\ u &= 0 \text{ on } \partial\Omega. \end{aligned} \tag{5.1}$$

Because of the tessellation required by FE methods, the resolution of the FE formulation associated to (5.1) will provide an approximation of the exact solution of the following problem:

$$\begin{aligned} \mathcal{L}_{\Omega_p}(u_p) &= f \text{ on } \Omega_p, \\ u_p &= 0 \text{ on } \partial\Omega_p. \end{aligned} \tag{5.2}$$

An interesting point would be to obtain a bound of the difference $\|u - u_p\|_{\Omega_p}$ in terms of appropriate geometrical parameters. This question was addressed by Thomée [Tho73] in the case of a stationary heat equation with homogeneous Dirichlet boundary conditions. Consider the situation described on figure 5.2. The geometrical distance between x and y_x is denoted $d(x, y_x)$. Here, x is a point of $\partial\Omega_p$, and y_x is the point of shortest distance to x located on $\partial\Omega$. By means of an appropriate set of rotation/translation, A can be placed at the origin of an orthonormal frame of vector space, with AB aligned on the abscissa axis. If $\partial\Omega$ is not polynomial, it is assumed that h is small enough so that $\partial\Omega$ can be expanded in a polynomial series in the vicinity of A or B. Then it is straightforward to prove that the maximal distance between the polygonal and the smooth boundary can be expressed as a polynomial in h, which lowest order is 2. Therefore, one obtains:

$$d(x, y_x) \leq Ch^2.$$

Let $e_p = u - u_p$, and suppose that $e_p \in L^\infty(\Omega_p)$. Then, one can write:

$$\sup_{\partial\Omega_p} \|e_p\| = \sup_{\partial\Omega_p} \|u - u_p\|.$$

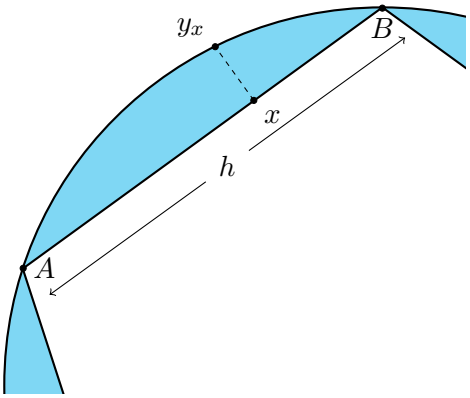


Figure 5.2 | Geometrical distance between a polygonal and a polynomial boundary.

Given that $u_p = 0$ on $\partial\Omega_p$, then one obtains:

$$\sup_{\partial\Omega_p} \|e_p\| = \sup_{\partial\Omega_p} \|u\|$$

Exploiting the fact that $u = 0$ on $\partial\Omega$, one can write:

$$\sup_{x\in\partial\Omega_p} \|e_p(x)\| = \sup_{x\in\partial\Omega_p} \|u(y_x) - u(x)\|.$$

The right hand side can then be bounded as follows thanks to a mean value theorem:

$$\sup_{x\in\partial\Omega_p} \|u(y_x) - u(x)\| \leq Ch^2 \left|u\right|_{1,\Omega}.$$

Up to now, no assumption has been made on the considered problem. However, the conclusion of this development requires a major assumption that is known to hold in the case of an elliptic problem, namely the maximum principle:

$$\sup_{x\in\partial\Omega_p} \|e_p(x)\| \geq \sup_{x\in\Omega_p} \|e_p(x)\|.$$

If it is verified, then it ensures the following result for a convex Ω:

$$\sup_{\partial\Omega_p} \|u - u_p\| \leq Ch^2 \left|u\right|_{1,\Omega}.$$

The latter results implies that any finite element solution of problem (5.2) will approximate the exact solution of problem (5.1) at most with second order accuracy, irrespectively of the order of approximation of the FE method. However, there is no such theorem for Maxwell's equations with PEC boundary conditions, and although the latter result can be numerically verified in this framework (see section 5.2), to the best knowledge of the author, no theoretical proof is available.

Numerical error

The error induced by the numerical scheme exploited to solve the problem on Ω_p is also a concern. Standard stability and convergence studies for the DG discretization on rectilinear meshes [F+05] do not hold, since the involved classical inequalities are usually mapped from the reference element. Given

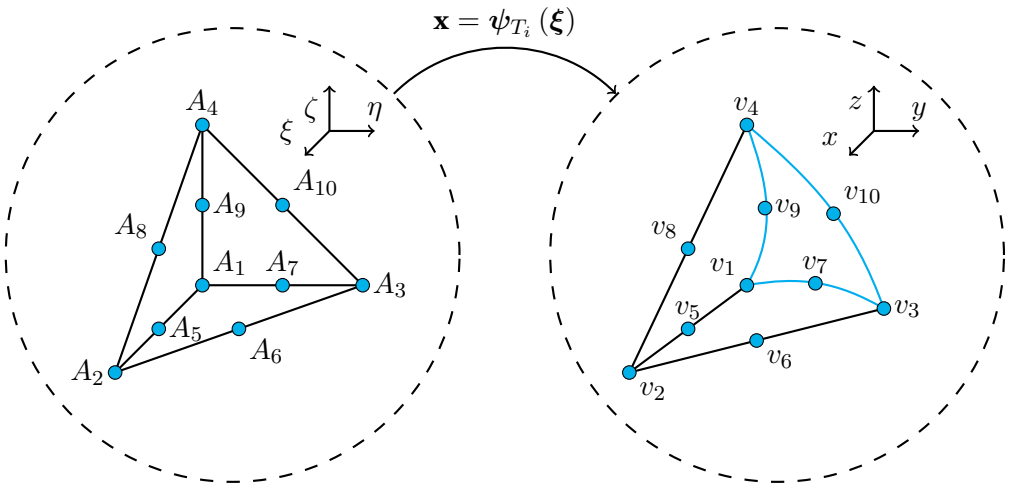


Figure 5.3 | Second order mapping from the reference element $\hat{T}$ to the physical element T_i.

that we are mainly concerned by the feasibility and the pertinence of the use of curvilinear meshes for nanophotonics and nanoplasmonics problems, we do not go back over these proofs. However, the reader can refer to [War10] for the main ideas.

5.1.2 High-order mapping

Expression and jacobian matrix

When curvilinear tetrahedra are present in the mesh, a mapping from the reference element $\hat{T}$ is still used. The difference with the rectilinear case lies in the mapping itself, which will now be non-linear in (ξ, η, ζ). To define higher order mappings from $\hat{T}$ to T_i, one needs to define more control points, and the reader might feel (rightly) that this will be connected in some way with the use of higher interpolation orders on the tetrahedrons : when using a DG method based on Lagrange polynomials, this connection is straightforward, and a higher order mapping will be a weighted sum of Lagrange polynomials defined on $\hat{T}$. Therefore, for a n^{th} order mapping, the usual M_n degrees of freedom (d.o.f.) on the reference tetrahedron $(A_j)_{j=1,..,M_n}$ are defined, where $M_n = \frac{1}{6}(n+1)(n+2)(n+3)$. Hence:

$$\boldsymbol{\psi}_{T_i}^{(n)}(\boldsymbol{\xi}) = \sum_{j=1}^{M_n} L_j^{(n)}(\boldsymbol{\xi})\, v_j = \sum_{0 \leq j+k+l \leq n} a_{jkl}^{(n)} \xi^j \eta^k \zeta^l.$$

For example, a second-order mapping would be written as follows:

$$\begin{aligned} \boldsymbol{\psi}_{T_i}^{(2)}(\boldsymbol{\xi}) = a_1^{(2)} + a_2^{(2)}\xi + a_3^{(2)}\eta + a_4^{(2)}\zeta \\ + a_5^{(2)}\xi\eta + a_6^{(2)}\xi\zeta + a_7^{(2)}\eta\zeta + a_8^{(2)}\xi^2 + a_9^{(2)}\eta^2 + a_{10}^{(2)}\zeta^2. \end{aligned} \tag{5.3}$$

A visual representation of a second-order mapping is given on figure 5.3. In this case, the terms of the jacobian matrix are first-order polynomials in ξ, η, ζ, and its determinant is therefore a third order polynomial in the same variables.

Coefficients

This section details the calculation of the coefficients for the quadratic mapping. From now on, the ridge that connects the vertices v_1 and v_2 through v_5 is denoted $v_1 - v_2$. To determine the coefficients $a_k^{(2)}$ from the coordinates of the vertices, each ridge is parameterized by choosing an appropriate set of coordinates for (ξ, η, ζ) in the expression of the quadratic mapping. If one considers the ridge $v_1 - v_2$, for example, an appropriate parametrization is:

$$\boldsymbol{\psi}_{T_i}^{(2)}(t, 0, 0) = a_1^{(2)} + a_2^{(2)} t + a_8^{(2)} t^2.$$

Then, one evaluates the latter parametrization for $t = 0$, $t = \frac{1}{2}$ and $t = 1$, which respectively correspond to v_1, v_5 and v_2. This leads to:

$$\begin{aligned} a_1^{(2)} &= v_1, \\ a_1^{(2)} + \frac{a_2^{(2)}}{2} + \frac{a_8^{(2)}}{4} &= v_5, \\ a_1^{(2)} + a_2^{(2)} + a_8^{(2)} &= v_2, \end{aligned}$$

whose resolution is straightforward. Repeating this procedure on the other ridges with adapted parameterizations then yields the following quadratic mapping coefficients :

$$\begin{aligned} a_1^{(2)} &= v_1, \\ a_2^{(2)} &= 4v_5 - v_2 - 3v_1, \\ a_3^{(2)} &= 4v_7 - v_3 - 3v_1, \\ a_4^{(2)} &= 4v_9 - v_4 - 3v_1, \\ a_5^{(2)} &= 4\left(v_1 + v_6 - v_5 - v_7\right), \\ a_6^{(2)} &= 4\left(v_1 + v_8 - v_5 - v_9\right), \\ a_7^{(2)} &= 4\left(v_1 + v_{10} - v_7 - v_9\right), \\ a_8^{(2)} &= 2\left(v_1 + v_2 - 2v_5\right), \\ a_9^{(2)} &= 2\left(v_1 + v_3 - 2v_7\right), \\ a_{10}^{(2)} &= 2\left(v_1 + v_4 - 2v_9\right). \end{aligned} \tag{5.4}$$

Interpolation and mapping compatibility

There are a few remarks to make regarding boundary approximation and order of interpolation. To make them clearer, consider the limit case presented on figure 5.4, where a disk is approached by four triangles (only the upper right part of this configuration is presented). To reuse the notations introduced in section 5.1.1, the hypotenuse of the triangle is of length h, and therefore the disk radius is $\frac{h}{\sqrt{2}}$. The coefficients are computed as explained in the previous section, by considering the following parametrization of the edge:

$$\begin{aligned} x &= t \\ y &= \alpha + \beta t + \gamma t^2, \end{aligned}$$

and the following conditions are imposed:

$$y(0) = \frac{h}{\sqrt{2}},\ y\left(\frac{h}{2}\right) = \frac{h}{2},\ y(\frac{h}{\sqrt{2}}) = 0.$$

Then, one deduces the parametrization:

$$\begin{aligned} x &= t \\ y &= \frac{h}{\sqrt{2}} + t - \frac{2\sqrt{2}}{h}t^2, \end{aligned}$$

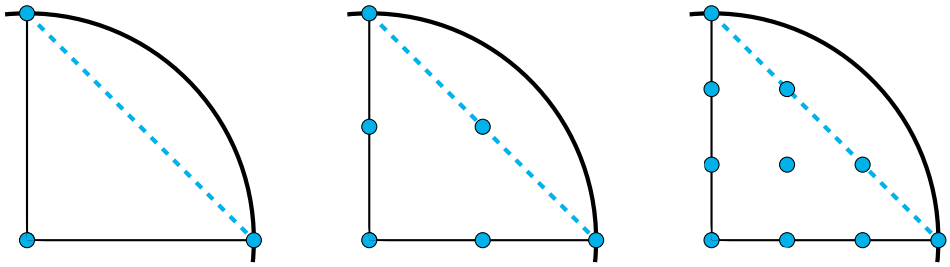
which is represented on figure 5.4 (the reader is reminded that n is the mapping order whereas p is the approximation order). It is quite obvious that the physical surface is not exactly matched by a second-order mapping, since the process described previously attempts to approach a circle with a parabola in 2D, and a sphere with a paraboloid in 3D. As presented on figure 5.4 for 2D settings, not every (n, p) combination would lead to a proper and efficient use of high orders. Indeed, one could wonder about the interest of the $(1, 2)$ and $(1, 3)$ cases, since more points are used to interpolate fields on an improper shape. The $(2, 1)$ case is obviously uninteresting, since the d.o.f. are the same as in the $(1, 1)$ case, the second-order mapping has no effect. Higher order is put at good use for the $(2, 2)$ case : indeed, the geometry is better fitted than in the linear case, and the additional d.o.f. resulting from increased p lie on the physical boundary. Increasing p again, the $(2, 3)$ case raises the same kind of wonders as the $(1, 2)$. However, as will be shown in section 5.2, exploiting a quadratic mapping lifts the spatial accuracy limit of the method from order 2 to order 4: going beyond that limit would require higher mapping order. Although this reasoning was done for a 2D situation, the discussion is quite similar in 3D, and the conclusions hold.

Higher order mappings could be achieved, but several reasons restrain us from exploiting them : first, curvilinear mesh untangling can become quite problematic, as will be explained in next section. Then, although they would lift the accuracy level from fourth to sixth and higher orders, computations of realistic cases rarely allow the use of very high polynomial orders for practical reasons. Hence, for the sake of robustness and simplicity, only quadratic mappings will be considered in the remaining of this thesis.

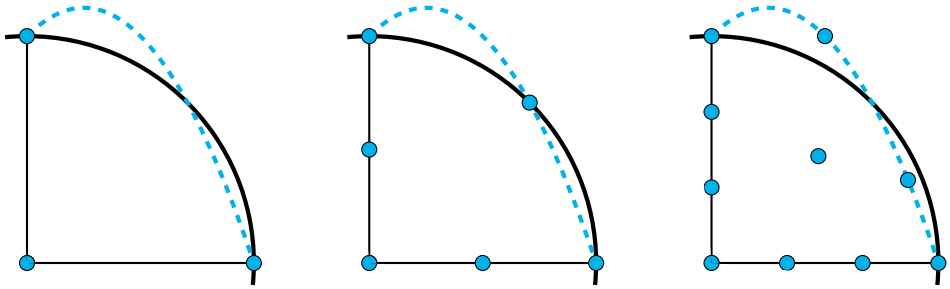
Curvilinear elements validity and quality

The matter of validity and quality of straight-edged elements have been widely studied in the previous decade, leading to a variety of highly-efficient mesh generators, such as `gmsh` [GR09] or `Netgen` [Sch97], for example. Each of these mesh builders make use of their own mesh quality optimization algorithm, which criteria are based on various geometrical factors such as lengths, angles and volumes. Among these, the most common are the aspect ratio, defined as the ratio of the longest edge of an element to its shortest, and the jacobian determinant, which gives an indication of the deformation of the physical element compared to the reference one.

The usual curvilinear meshing procedure consists in (i) creating a rectilinear mesh from a given geometry with usual tools, then (ii) orthogonally project the high-order nodes of the rectilinear mesh on the curved geometry [DOS99]. However simple, this method usually leads to a small amount of tangled elements, which detection can be achieved from a study of the jacobian determinant of the transformation [TGRL12]. Indeed, unlike the rectilinear case, it is not trivial to ensure that the jacobian determinant remains positive everywhere inside the high-order mapped element. Although a direct computation of the minimum of the jacobian determinant over the cell is possible, the cost of this method happens to rise dramatically in the case of three space dimensions and high mapping order: indeed, it requires to solve

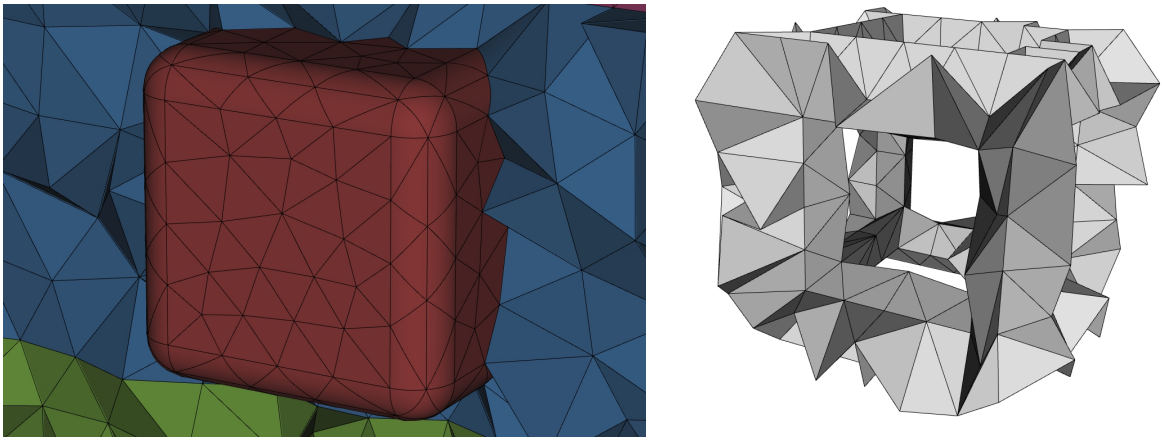


(a) $(1, 1)$ (b) $(1, 2)$ (c) $(1, 3)$



(d) $(2, 1)$ (e) $(2, 2)$ (f) $(2, 3)$

Figure 5.4 | Proper and improper (n, p) combinations for a quadratic physical surface. The latter is in thick continuous black, whereas the mapped surface is in dashed blue. The d.o.f. are represented by blue dots.



(a) Nanocube mesh

(b) Selected curvilinear elements

Figure 5.5 | Curvilinear elements selection on a nanocube mesh with rounded corners.

a cubic trivariate equation on each cell for a second order mapping, and a sextic one for a third order mapping. In [TGRL12], the authors exploit a property of the Bezier polynomials to determine bounds of the scaled jacobian determinant. These bounds are then exploited in a mesh optimization procedure that balances the displacement of the nodes with the values of the scaled jacobian. In [JRG12], the latter bounds are improved by the use of an adaptative procedure, which shows better performances and robustness than brute-force based algorithms.

Selecting curved elements

Usually, curvilinear mesh generators provide fully-quadratic meshes. Hence, during a pre-processing phase, their elements need to be sorted, in order to separate straight sided elements from truly curved ones. This procedure can be executed by examining the scaled jacobian in each element:

$$|\mathbf{J}_s|(\mathbf{x}) = \frac{|\mathbf{J}_\psi|(\mathbf{x})}{|\mathbf{J}_0|},$$

where $|\mathbf{J}_0|$ is the jacobian value for the straight sided element (*e.g.* a constant). Therefore, sorting the elements can be done by exploiting the adaptive procedure described previously. However, for the sake of simplicity, we make use of a more basic (and more expensive) method, which consists in evaluating $|1 - |\mathbf{J}_s||$ for a given number of random points n_r in the physical element. Although far from optimal, this procedure has proven to be effective on all the considered situations. As an illustration, figure 5.5 shows the elements selected by this method on a nanocube mesh with rounded corners (see section 5.4). The selection procedure is detailed in algorithm 1. In all the computations of the present study, the error parameter δ is taken equal to 1×10^{-5}, while the number of random points per element is 100.

Example of hand-made quadratic tetrahedral mesh : the sphere case

Although it exists a wide variety of free meshers, the number of those able to handle the generation of quadratic meshes is somehow limited. In this work, the well-known `gmsh` tool is exploited for every geometry more complex than a simple sphere. However, the quadratic meshing of a sphere is somewhat

Algorithm 1 Curvilinear elements selection

for $i \leftarrow 1, n_t$ **do**
    Compute $\alpha_j^{(2)}$ ▷ Compute mapping coefficients
    Compute $|\mathbf{J}_0|$ ▷ Compute linear jacobian
end for

for $i \leftarrow 1, n_t$ **do**
    for $k \leftarrow 1, n_r$ **do** ▷ Loop over random points inside T_i
        Compute $|\mathbf{J}_s|(\mathbf{x})$ ▷ Compute values of scaled jacobian
    end for

    Find $|\mathbf{J}_s|_{\text{min}}$ and $|\mathbf{J}_s|_{\text{max}}$ on T_i
    if $|1 - |\mathbf{J}_s|_{\text{min}}| > \delta$ **or** $|1 - |\mathbf{J}_s|_{\text{max}}| > \delta$ **then** ▷ Test if scaled jacobian is close to 1
        T_i is a curved element
    else
        T_i is a linear element
    end if
end for

simple, and can be implemented fairly easily as a pre-processing step of any numerical solver. This is the purpose of this section.

The coordinates of the center of the sphere are denoted $P_c = (x_c, y_c, z_c)$, r is the radius, and it is assumed that one disposes of a rectilinear mesh of the latter sphere. The first step consists of determining the tetrahedra that belong to the boundary of the sphere, which can be done considering simple geometrical tests on the coordinates of the vertices. Once this is done, it is necessary to determine if the boundary tetrahedron has three vertices on the boundary, or only two, in order to select the degrees of freedom whose coordinates will have to be modified. $P_1 = (x_1, y_1, z_1)$ now generically denotes the coordinates of a $\mathbb{P}_2$ degree of freedom that actually has to be moved on the boundary. The next step consists in seeking the intersection of the line defined by (P_c, P_1) and the sphere, which will be the final location of the bent degree of freedom, $P_s = (x_s, y_s, z_s)$. A parameterization of the line is:

$$\begin{aligned} x &= x_c + (x_1 - x_c)t, \\ y &= y_c + (y_1 - y_c)t, \\ z &= z_c + (z_1 - z_c)t, \end{aligned}$$

whereas the sphere is defined by:

$$(x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2 = r^2.$$

Since P_s belongs to both, it is trivial to determine that:

$$\begin{aligned} x_s &= x_c + (x_1 - x_c)t_s, \\ y_s &= y_c + (y_1 - y_c)t_s, \\ z_s &= z_c + (z_1 - z_c)t_s, \end{aligned}$$

with:

$$t_s = \frac{r}{\sqrt{(x_1 - x_c)^2 + (y_1 - y_c)^2 + (z_1 - z_c)^2}}.$$

5.1.3 Numerical integration

One of the major drawbacks of using curved elements is that the FE matrices on a curved tetrahedron T_i cannot be calculated from the FE matrices on the reference tetrahedron $\hat{T}$ as easily as in the linear case. Indeed, the Jacobian determinant is not a constant over T_i, and therefore the integrals must be calculated by means of numerical integration on each curved tetrahedron, which is what we will describe now. The general expressions of the substitutions for the matrices have already been given in section 3.1.4, and therefore the reader is referred to this part of the manuscript. In the following, we remind the concept of numerical integration, and detail a few facts that will affect our upcoming integration strategy.

The basic principle of quadrature and cubature rules is to evaluate the integral of a continuous function by a weighted sum of its values at well-chosen points. Hence, a quadrature or cubature rule can be defined by a set of weight $(w_i)_{i=1,..,N}$ and a set of points $(\boldsymbol{\lambda}_i)_{i=1,..,N}$:

$$\int_{\hat{T}} f(\mathbf{x}) d\mathbf{x} \simeq \sum_{i=1}^{N} w_i f(\boldsymbol{\lambda}_i)$$

Several matters arise from these ideas, among which are (i) the assumptions made on f and (ii) the number of points N necessary to obtain a good approximation of the integral. The answer to (i) is simply that f should be smooth enough to be well approximated by polynomials. In the case where f is polynomial, the integral can be calculated exactly, provided that the cubature rule used to evaluate it is of sufficient order[1]. The matter raised in (ii) is indeed an important one, since it will affect the time required for the calculation of the integral, and, in our case, the time necessary for the assembly of the FE matrices. A very classical set of quadrature and cubature rules are the Gauss-Legendre ones [KS05], which can be exact up to any order when there is no restriction on the number of cubature points. However, the Gauss-Legendre exact quadrature rule of a polynomial of order $2r$ over a d-simplex will require $(r+1)^d$ points, which can lead to heavy computational loads in three dimensions of space. In [Coo03], the authors have compiled optimal existing cubature rules from various references up to the 11^{th} order for tetrahedrons. They require considerably less points for the same accuracy when compared to Gauss-Legendre. It is quite straightforward to determine the polynomial order of the integrand for each matrix:

- For the mass matrix, the integrand is a polynomial of order $2p + 3(n-1)$;
- For the stiffness matrix, the integrand is a polynomial of order $2p - 1 + 3(n-1)$;
- For the surface mass matrix, the integrand is not a polynomial.

Therefore, for a second order mapping ($n = 2$), exact integration can be achieved for the mass and stiffness matrices for $p \in \{2, 3, 4\}$, but no exact integration can be obtained for a third order mapping with these optimal quadrature rules. Therefore, another set of quadrature and cubature rules is chosen here [ZCL09]. The rules produced by this algorithm require a bit more evaluation points than the aforementioned ones, but they are all generated from the same source, and are available up to the 13^{th} order for tetrahedrons, and 20^{th} order for triangles. The number of required integration points for the triangle and the tetrahedron are respectively given in table 5.1 and 5.2. A graphic comparison of the sets from [ZCL09] with classical Gauss-Legendre rules is given in figure 5.6.

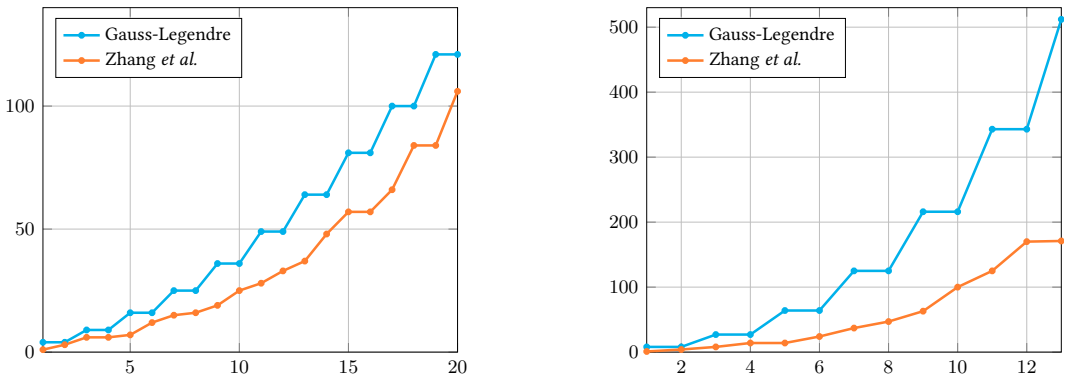
[1] A cubature rule is said to be of order m when it evaluates exactly the integral of a polynomial of same order

Table 5.1 | Number of integration points required for exact integration on the triangle for order 1 to 20.

Order	1	2	3	4	5	6	7	8	9	10
[ZCL09]	1	3	6	6	7	12	15	16	19	25
Gauss-Legendre	4	4	9	9	16	16	25	25	36	36
Order	11	12	13	14	15	16	17	18	19	20
[ZCL09]	28	33	37	48	57	57	66	84	84	106
Gauss-Legendre	49	49	64	64	81	81	100	100	121	121

Table 5.2 | Number of integration points required for exact integration on the tetrahedron for order 1 to 13.

Order	1	2	3	4	5	6	7	8	9	10	11	12	13
[ZCL09]	1	4	8	14	14	24	37	47	63	100	125	170	171
Gauss-Legendre	8	8	27	27	64	64	125	125	216	216	343	343	512



(a) Triangle (b) Tetrahedron

Figure 5.6 | Number of integrations points required by Gauss-Legendre and [ZCL09] rules on triangle and tetrahedron.

Table 5.3 | Meshes characteristics for the spherical PEC cavity case. n_s is the number of vertices, n_t the number of tetrahedrons and $h_{\max}$ the typical size of the largest tetrahedron. For the curvilinear versions, n_c represents the number of curved tetrahedrons, whereas n_r is the number of rectilinear tetrahedrons.

	M1	**M2**	**M3**	**M4**
n_s	309	2057	14993	114465
n_t	1280	10240	81920	655360
$h_{\max}$	0.461	0.245	0.128	0.0648
n_c	560	2400	9920	40320
n_r	720	7840	72000	615040

***Remark** : It must be noted that the surface matrices are here never exactly integrated, since their integrand is not polynomial.*

5.1.4 CFL condition

Since the usual CFL condition relies on the minimal mesh length, it is necessary to take into account the curvature of the tetrahedra to compute the timestep. In [HW01], the authors provide the following expression :

$$\Delta t \leqslant \mathrm{CFL} \min_{\Omega} \frac{\sqrt{\varepsilon_r \mu_r}}{|\chi|},$$

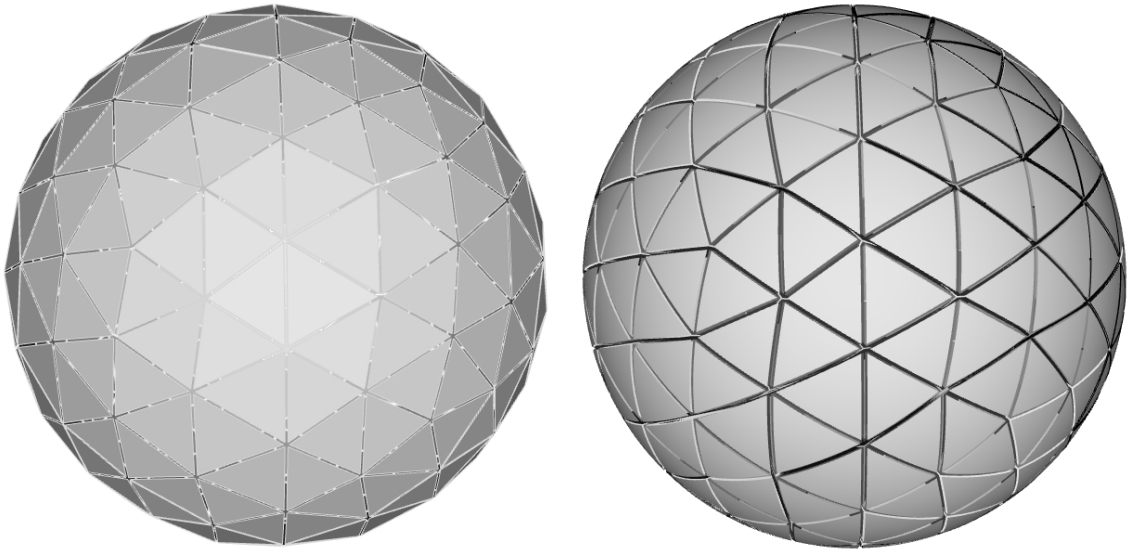
where, for the reference element given above, and for a polynomial approximation of the fields of order p :

$$\chi = \frac{1}{p} \left[\left| \begin{matrix} \frac{\partial \boldsymbol{\xi}}{\partial x} \\ \frac{\partial \boldsymbol{\xi}}{\partial y} \\ \frac{\partial \boldsymbol{\xi}}{\partial z} \end{matrix} \right| + \left| \begin{matrix} \frac{\partial \boldsymbol{\eta}}{\partial x} \\ \frac{\partial \boldsymbol{\eta}}{\partial y} \\ \frac{\partial \boldsymbol{\eta}}{\partial z} \end{matrix} \right| + \left| \begin{matrix} \frac{\partial \boldsymbol{\zeta}}{\partial x} \\ \frac{\partial \boldsymbol{\zeta}}{\partial y} \\ \frac{\partial \boldsymbol{\zeta}}{\partial z} \end{matrix} \right| \right].$$

5.2 Validation on a spherical PEC cavity

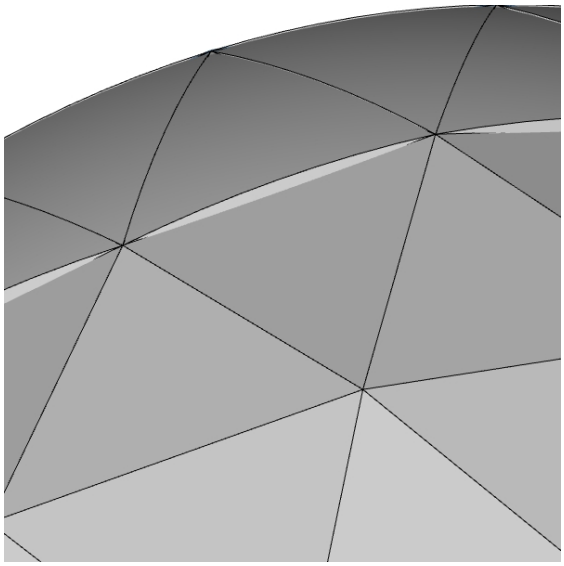
In this section, the implementation of the curvilinear DGTD method is validated using the spherical PEC cavity test-case presented in section 2.1.4 (see equation (2.25)). The cavity mode is simulated for a physical time $t_{\max} = 3.817\,10^{-8}$ s, which corresponds to 5 periods. Four different rectilinear meshes of increasing refinement were generated: their characteristics are summed up on table 5.3, and visual representations of the M1 mesh can be found on figure 5.7. These meshes were generated with `gestikulator` [Vel], that produces very high quality sphere meshes from successive refinements of the icosahedron, and the surface tetrahedra were then bent following the procedure described in section 5.1.2. For each mesh, $\mathbb{P}_1$ to $\mathbb{P}_4$ approximations are used, both on rectilinear and curvilinear versions, in combination with centered or upwind fluxes and RK4 time integration.

In the first subsection, the h-convergence rates of the method are calculated numerically, confirming the statements of section 5.1.1. A focus is also made on the CPU time required to numerically integrate the FE matrices. To conclude, a few remarks are made on the importance of visualization choices.



(a) Rectilinear mesh

(b) Curvilinear mesh



(c) Boundary representation

Figure 5.7 | M1 mesh for the PEC cavity case and zoom on the boundary.

Table 5.4 | Convergence rates of the spherical cavity case for different approximation orders and fluxes, with rectilinear and curvilinear meshes of increasing refinement. α is the upwinding factor.

	α	M1		M2		M3		M4	
		Rect.	Curv.	Rect.	Curv.	Rect.	Curv.	Rect.	Curv.
$\mathbb{P}_1$	0	–	–	2.01	–	1.85	–	1.51	–
	1	–	–	2.14	–	2.03	–	2.01	–
$\mathbb{P}_2$	0	–	–	2.19	2.85	2.14	2.29	2.03	2.30
	1	–	–	2.20	3.25	2.03	3.08	2.01	3.06
$\mathbb{P}_3$	0	–	–	2.19	4.17	2.14	3.82	2.03	3.55
	1	–	–	2.20	4.36	2.03	4.03	2.03	4.03
$\mathbb{P}_4$	0	–	–	2.18	4.37	2.14	4.25	2.03	4.04
	1	–	–	2.18	4.36	2.03	4.26	2.03	4.05

5.2.1 Results

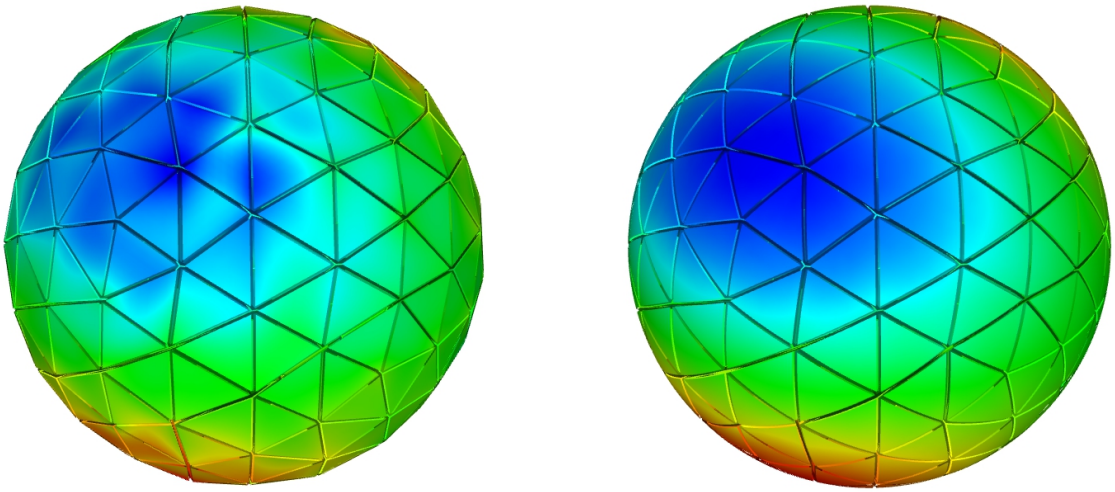
h-convergence

For each simulation, the global L^2 error is calculated and stored over time for the whole mesh. For two successive meshes, the maximum error levels are measured, and the h-convergence rates are then deduced. The values are displayed on table 5.4 for the three proposed set-ups. These results underline the statements made in figure 5.3 about the relation between the geometry and the field approximations: the use of curvilinear tetrahedra restores the optimal spatial convergence rates of the DGTD method, proportional to h^k for centered fluxes, and to h^{k+1} for upwind fluxes.

For a constant error level, curvilinear elements allow to save a lot in terms of degrees of freedom, and therefore in CPU time : for the upwind flux, the best solution obtained with linear elements is compared with a solution of similar error level obtained from curved elements. Hence, we take on the $\mathbb{P}_1$ solution on the rectilinear M4 mesh ($\varepsilon = 3.93\,10^{-3}$, memory consumption of 1500 MB, CPU time 9664 seconds) and the $\mathbb{P}_3$ solution of the curvilinear M1 mesh ($\varepsilon = 1.73\,10^{-3}$, memory consumption of 39.6 MB, CPU time 50 seconds). To reach a similar level of accuracy, the curvilinear solution requires 35 times less memory, and is almost 200 times faster. For the reader, it is worth noticing that in the case of the linear mesh, increasing the order of approximation yielded no improvement in the solution. As a visual representation of the benefits of curvilinear elements, the $\mathbb{P}_2$ numerical solution obtained on the M1 mesh is plotted on figure 5.8 for both linear and curved meshes. As one can notice, the curvilinear solution is already almost converged, which is not the case of the linear solution.

Numerical integration of FE matrices

One might wonder about the time required to numerically integrate the FE matrices in the pre-processing steps of a curvilinear simulation. On figure 5.9, the time required per curved tetrahedron to perform this integration (including the mass matrix inversion) on the M3 mesh is given. The cubature is executed following the rules given in section 5.1.3, whereas the surface quadrature is performed using a rule which is always one order higher than the cubature one. The fact that the stiffness and the surface matrices require approximately the same time relies on the fact that for both, expensive operations have to be done : to build $\mathbb{K}$, three different matrices have to be generated (one for each direction of space), each one requiring to calculate the value of the derivates of the Lagrange functions in a different direction; on the other hand, building $\mathbb{S}$ requires to run a similar process for the four different faces of each tetrahedron,



(a) Rectilinear M1 mesh

(b) Curvilinear M1 mesh

Figure 5.8 | $\mathbb{P}_2$ numerical solution for the E_x field on the boundary of the M1 mesh, both in rectilinear and curvilinear cases. The solutions have been scaled to the range $[-0.14, 0.14]$.

thus leading to an important percentage of the total construction. The mass matrix is inverted with the Lapack algebra library.

5.2.2 Visualization issues

Although DG methods allow to compute high order accurate numerical solutions, the 3D visualization of post-processed solution often lacks the same accuracy, and it is not uncommon to project a high-order $\mathbb{P}_k$ solution as a $\mathbb{P}_1$ solution on an adequatly refined mesh. Although there are sometimes no other alternatives, we would like to show that this procedure induces even more inaccuracies in the case of curvilinear meshes. To do so, the $\mathbb{P}_2$ curvilinear solution from the M1 mesh is projected as a $\mathbb{P}_1$ solution on the M2 mesh (see figure 5.10) : although it is not clearly visible, the palette indicates different minimum and maximum values for the two visualizations, the relative difference being of a few percent.

5.3 Plasmonic resonances of isolated and coupled gold nanospheres

5.3.1 Isolated nanosphere

Setup

Many nano-optics devices rely on the coupled plasmon resonances of metallic nanospheres, such as nano-arrays for Raman scattering [LBU+08], Fano resonators [LZM+10], or nanosphere-based biosensors [CLS+11]. For this reason, we analyse the improvements obtained with curvilinear meshes on isolated and coupled nanospheres. First, high-order elements are exploited to realize efficient cross-section computations of a single gold nanosphere described by a Drude model. The analytical solution of this problem can be computed *via* the Mie scattering theory [vdH81]. Here, we consider a sphere of radius $r = 50$ nm with Drude parameters $\varepsilon_\infty = 3.7362$, $\omega_d = 1.387 \times 10^7$ GHz, $\gamma_d = 4.515 \times 10^4$ GHz, and we are interested in its behavior in the $[600, 1200]$ THz range. The incident field is a plane wave, with a sine-module

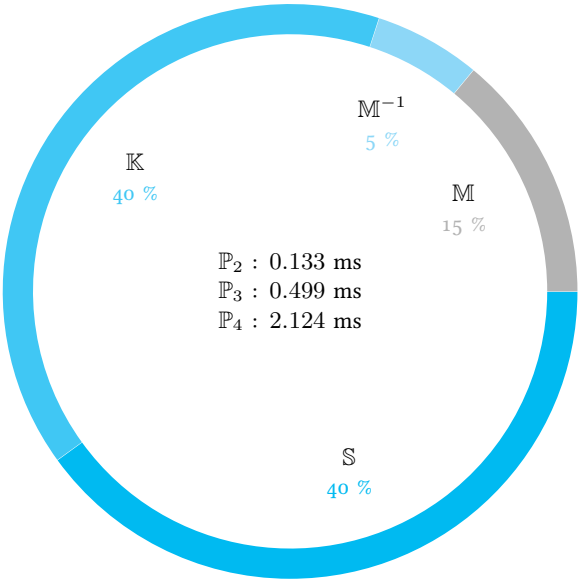
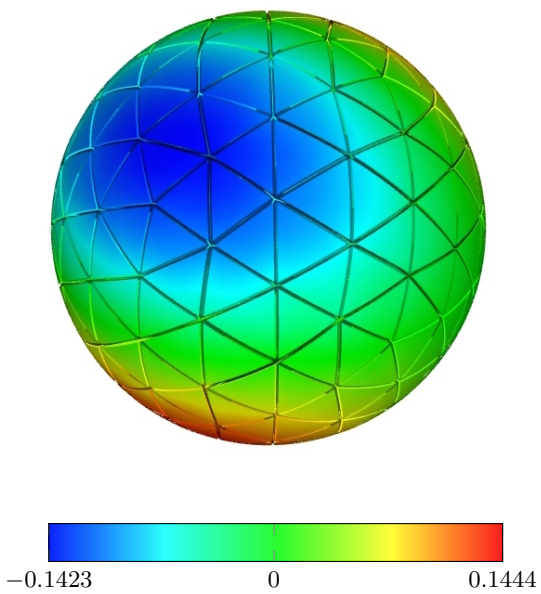
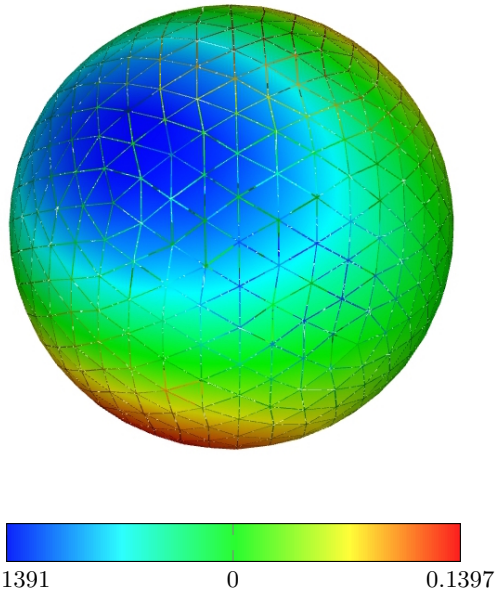


Figure 5.9 | Total time per curved tetrahedron required for the numerical integration of the FE matrices for $\mathbb{P}_2$, $\mathbb{P}_3$ and $\mathbb{P}_4$ approximations. The displayed percentages correspond to mean values of time required by each step of the numerical integration.



(c) $\mathbb{P}_2$ curvilinear solution on M1



(d) $\mathbb{P}_2$ curvilinear solution from M1 projected as a $\mathbb{P}_1$ solution on M2

Figure 5.10 | $\mathbb{P}_2$ curvilinear numerical solution of the spherical cavity mode on the M1 mesh projected as a $\mathbb{P}_1$ solution on the M2 mesh.

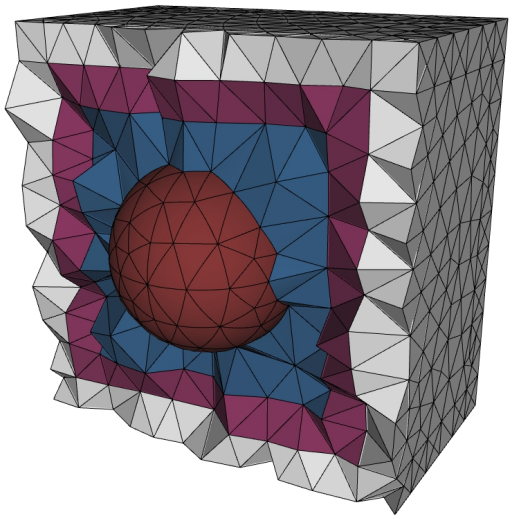


Figure 5.11 | M1 mesh for the cross-section calculation. The scatterer (in red) is enclosed by the total field region (in blue), delimited by the TF/SF interface on which the incident field is imposed. Then we find the scattered field region (in purple), surrounded by PMLs (in gray).

Gaussian time profile, in order to provide a wide enough spectrum for the calculation. The scatterer is enclosed by a total-field/scattered-field interface, on which the incident field is imposed. A CFS-PML layer (see section 4.1.2) surrounds the scattered-field region, and is terminated by an ABC condition.

Results

We compare the results from DGTD simulations with the Mie solution of the problem. The latter is calculated with a Matlab script written by Dirk Baumann [BFHL09]. To conduct this study, we build three meshes M1, M2 and M3 with `gmsh`, for which the geometry of the sphere is meshed with an increasing accuracy (the mesh characteristics and a visual representation are respectively given in table 5.5 and figure 5.11). Curved elements are exploited only with the M1 mesh, whereas linear elements are used for the three meshes : all results are presented in figure 5.12. One immediately notices the convergence of the results obtained on the linear meshes toward the reference solution. The linear solution on M3 mesh almost perfectly fits the Mie prediction, at the cost of a high refinement level of the sphere surface. On the other hand, the solutions obtained with the curvilinear M1 mesh are already in very good agreement with the reference solution: the $\mathbb{P}_2$ result is close, but the amplitude of the second resonance peak is still a bit undervalued. The numerical solution is improved by exploiting $\mathbb{P}_3$ approximation, yielding a relative error to the exact solution of less than 1%.

Although this case corresponds to a basic but realistic nanophotonics configuration, the gains obtained in terms of CPU time and memory consumption are very encouraging. The best curvilinear solution ($\mathbb{P}_3$ M1) required 92 MB of memory and 884 sec of CPU time[2]. In comparison, the best linear solution ($\mathbb{P}_2$ M3), which is of similar accuracy, required 312 MB and 6800 sec. Hence, it makes the curvilinear simulation more than 3 times cheaper in terms of memory, and more than 7 times faster. The difference between these values and those obtained for the spherical PEC cavity can be explained by (i) the more

[2] All the simulations are run in parallel on 16 CPUs.

Table 5.5 | Meshes characteristics for the metallic sphere cross section computation case. n_s is the number of vertices, n_t the number of tetrahedrons and h_{sphere} the typical size of the largest tetrahedron used to discretize the scatterer. For the curvilinear versions, n_c represents the number of curved tetrahedrons, whereas n_r is the number of rectilinear tetrahedrons.

	M1	**M2**	**M3**
n_s	962	1677	10736
n_t	4706	8767	61718
h_{sphere}	25×10^{-9}	10×10^{-9}	3.5×10^{-9}
n_c	764	0	0
n_r	3942	8767	61718

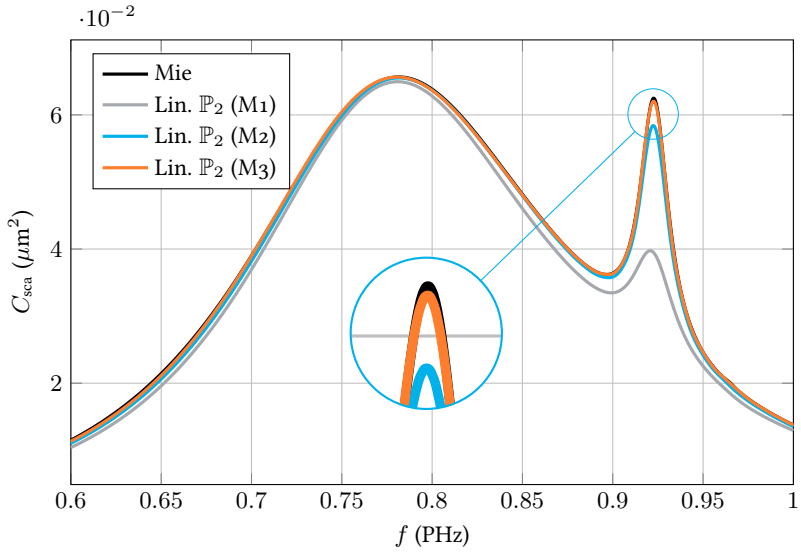
realistic nature of this test-case, that includes more features (TF/SF interface, PMLs, dispersive materials, on-the-fly Fourier transform), and by (ii) the fact that the successive refinements are not global, but only affect the surface of the sphere.

It is worth to note that for the M1 mesh, curved tetrahedra roughly represent 15 % of the total number of cells. As a consequence, the CPU overhead they induce remains limited, which makes them a good default choice for any problem involving non-trivial geometries. Indeed, on the ($\mathbb{P}_2$ M1) case, the CPU time required for the rectilinear mesh is 234 sec, whereas it is 253 sec for the curvilinear mesh, which represents a 8.1 % increase.

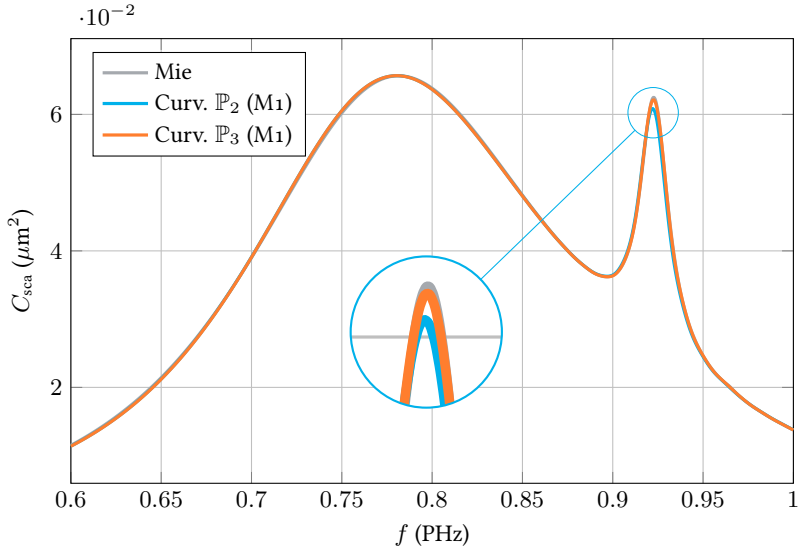
5.3.2 Coupled nanospheres

As stated earlier, plasmonic coupling between nanoparticles is at the heart of many applications in nano-optics [FVV07]. Hence, we now consider the coupling of two identical gold nanospheres, with the same parameters as earlier. The two nanospheres are aligned along the polarization direction of the incident field, and the surface-to-surface distance is set to 4 nm. In this configuration, the coupled plasmon resonance induces very intense fields in the gap between the particles. Then, a proper near-field resolution is essential to a good understanding of the properties of such coupled structures. To properly account for the high intensity of the fields, we use $\mathbb{P}_4$ polynomial approximation with upwind fluxes and a low storage Runge Kutta time-scheme of order 4 (LSRK4). For both rectilinear and curvilinear meshes, the total Fourier transform of the electric field is computed at the resonance frequency $f = 953$ THz during the whole simulation, and its modulus is then extracted: a field map is shown on figure 5.13. For the rectilinear mesh, the field intensity in the gap is underestimated, while the field behavior at the surface of the spheres is unclear. The secondary resonances that appear on the curvilinear solution are not resolved at all on the linear solution. We would like to make the most of this example to illustrate the importance of near-field resolution in nano-optic devices: in the present case, the two field maps shown on figure 5.13 present a striking contrast, which is also clearly visible at the same frequency on the cross-section computation of figure 5.14.

In this configuration, the ratio of curvilinear tetrahedra over the total number of cells is higher, reaching almost 19 %. Roughly half of these tetrahedra lie in the dispersive region. It yields a CPU overhead of 27.5 %, and an additional memory consumption of 78.3 %. This very high value is explained by the large size of the curvilinear finite element matrices for a $\mathbb{P}_4$ tetrahedron. Indeed, each tetrahedron here requires the storage of four 35×35 and four 15×15 matrices, which also considerably increases the memory adressing time.

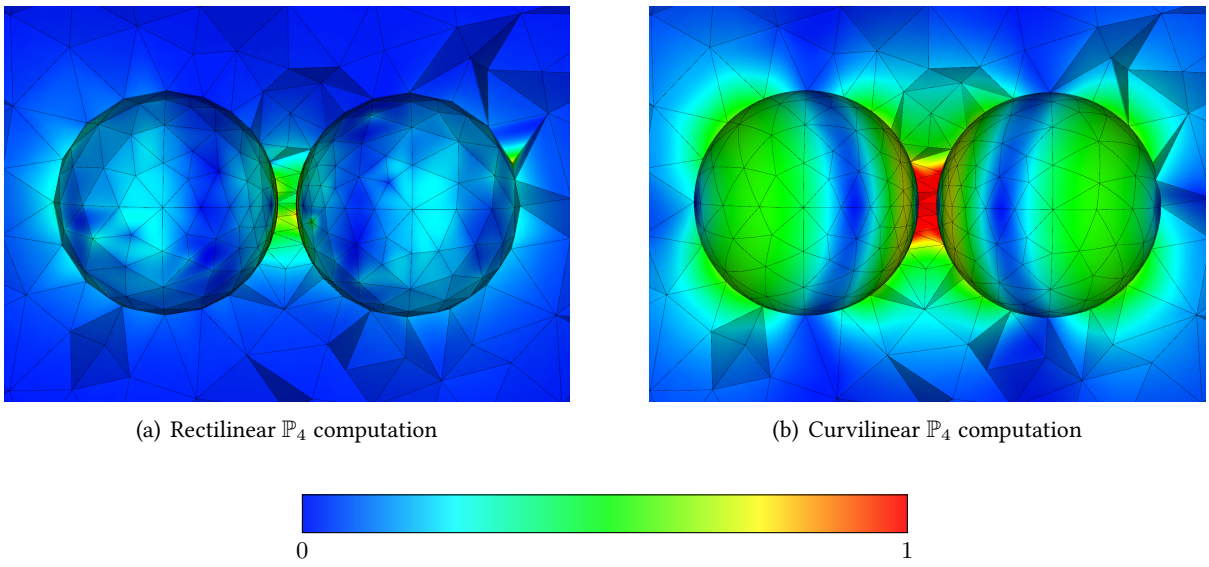


(a) C_{sca} calculations with linear elements



(b) C_{sca} calculations with curvilinear elements

Figure 5.12 | Scattering cross-section of a metallic sphere obtained with $\mathbb{P}_2$ and $\mathbb{P}_3$ approximations for linear and curvilinear meshes on various refinement levels.



(a) Rectilinear $\mathbb{P}_4$ computation

(b) Curvilinear $\mathbb{P}_4$ computation

Figure 5.13 | Near-field visualization of the electric field Fourier transform for a gold nanosphere dimer. The computation is conducted with $\mathbb{P}_4$ approximation, for both rectilinear and curvilinear meshes. The field values are normalized to 1 in both cases.

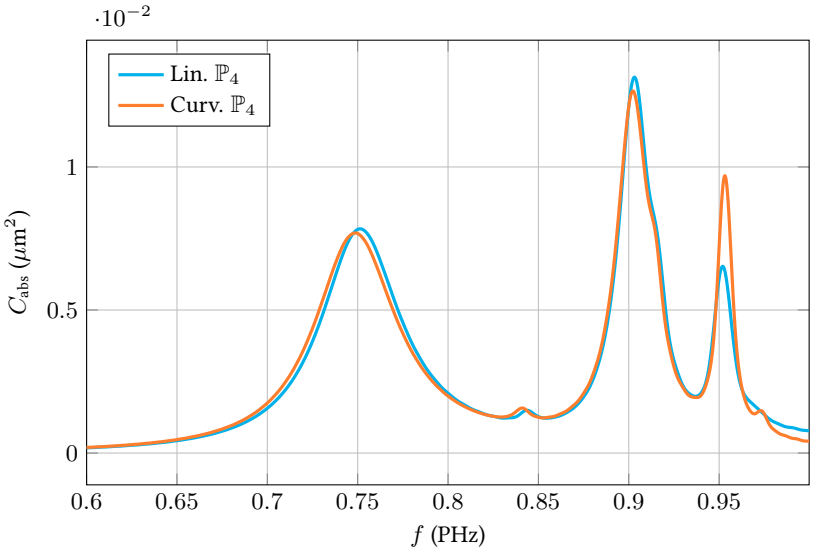


Figure 5.14 | Absorption cross-section of a gold nanosphere dimer obtained with $\mathbb{P}_4$ approximation for linear and curvilinear meshes.

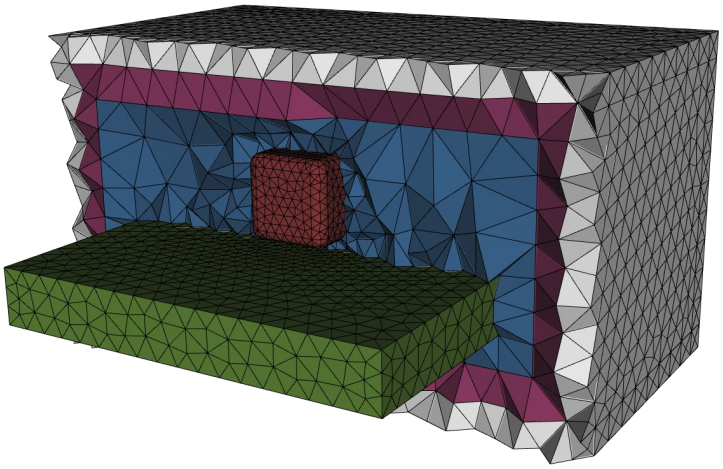


Figure 5.15 | Nanocube mesh with rounded corners. The silver cube (in red) is put on a gold layer (in green), both enclosed by the total field region (in blue). The latter is surrounded by a TF/SF interface on which the incident field is imposed. Then, we find the scattered field region (in purple), surrounded by PMLs (in gray).

5.4 Realistically-rounded metallic nanocubes

Randomly arranged silver nanocubes placed on a gold film have recently attracted attention for their capacity to support gap plasmon resonances when placed above metallic films [MCM+12]. The cubes are usually chemically produced, which causes small roundings to appear at their angles. A good geometrical resolution of these roundings is critical in the design process of nanocube-based devices, since they greatly affect the geometry of the gap. In the DGTD framework, this geometrical feature can be quite expensive to resolve with straight sided elements, given that the typical rounding size is very small compared to the size of the global device (including the gold film). In this section, the impact of this rounding is presented when it is geometrically resolved by linear and quadratic elements. An extended study of metallic nanocubes is presented in section 8.2.

A gold slab of thickness 50 nm topped with a 7 nm dielectric spacer of permittivity 1 is considered. A silver nanocube of edge length 75 nm is placed on top, with circular edge roundings of radius 10 nm. Gold and silver are described by Drude models, which take into account the dispersive properties of metals. The general setup is illuminated by a wideband sine-modulated gaussian pulse of central frequency 550 THz imposed on a TF/SF interface, and the chosen physical time is 0.233 ps. The domain is terminated by a CFS-PML layer: the global setup is shown on figure 5.15. Absorption cross-sections of the whole device are computed with $\mathbb{P}_3$ approximation and compared for linear and curvilinear elements : results are presented on figure 5.16. As one can notice, curvilinear elements make a significant difference in the results by properly resolving the cavity geometry, resulting in a blueshift of 10 THz and a difference in amplitude of almost 10 % for the absorption peak. This enhanced resolution is obtained for a very reasonable cost, since the required CPU time for the curvilinear solution (7941 s) is only 6.3 % higher than the rectilinear one (7469 s)[3]. The memory consumption follows the same trend, with an overhead of 9.7 % (555.6 Mo for the rectilinear solution against 609.4 Mo for the curvilinear solution).

[3]All the simulations are run in parallel on 32 CPUs.

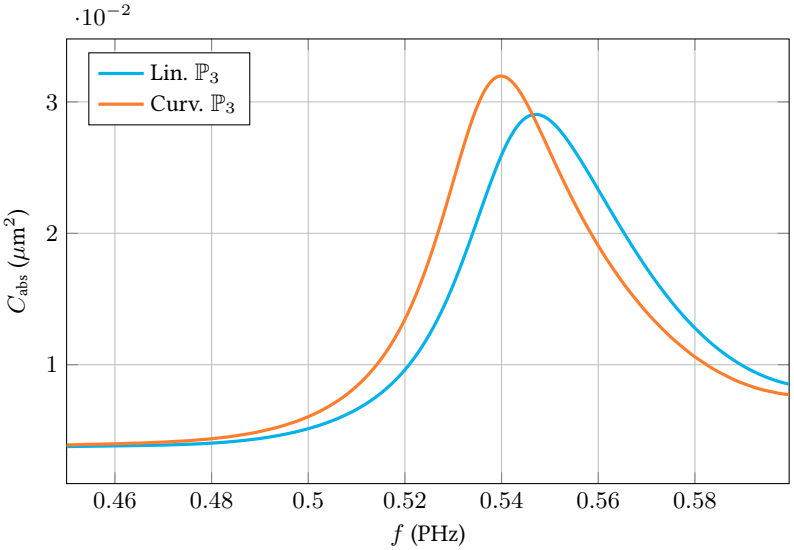


Figure 5.16 | Absorption cross-section of a rounded silver nanocube on a gold film obtained with $\mathbb{P}_3$ approximation for linear and curvilinear meshes.

6

Locally adaptive DGTD method

In section 3.1, the DG formulation was derived assuming a uniform polynomial order across the whole computational domain. However, in the case of a mesh showing large variations in cell size, the timestep imposed by the smallest cells can be a serious hindrance when trying to exploit high approximation orders. Indeed, part of the CPU overhead is devoted to the computation of the fields inside small cells where high polynomial orders might not be essential, while they could be necessary in the larger cells.

To overcome this limitation, several strategies can be considered. The first one consists in replacing the explicit method by an implicit timestepping algorithm. This, however, is at the expense of the resolution of a large linear system at each timestep [VB09]. In [Ver10] and [Moy13], the authors consider a hybrid formulation, where only the smallest cells are treated *via* an implicit scheme, while keeping an explicit time integration for the rest of the tesselation, thus limiting the timestep constraint. Although it provides very interesting results in terms of CPU speedup, maintaining high-order time integration has proven to be difficult [Moy13]. Another possiblity is to exploit local timestepping: based on the size of the elements, the mesh is divided in regions, each of which being assigned an appropriate timestep for an explicit time integration (see for example [Pip06] and [DG09]). As shown in [DG09], high-order convergence is preserved by this method. However, it seems difficult to ensure a good load balance in the case of a parallel implementation, given that the natural repartition is based on a timestep criterion, instead of a parallel-related one.

A complementary strategy relies on the use of variable polynomial orders: by imposing low orders in small cells, and high orders in large cells, it is possible to significantly alleviate both the global number of degress of freedom and the timestep restriction with a minimal impact on the method accuracy. 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. Here, the point of view is quite different: starting from a given mesh and an homogeneous polynomial order $\mathbb{P}_k$, the LA strategy exploits all the polynomial orders $\mathbb{P}_p$ with $p \leq k$ to obtain a solution of similar accuracy for a reduced computational cost.

In this chapter, the DG formulation of section 3.1 is modified to account for variable polynomial orders. The resulting algorithm is tested for convergence, and an extended performance study is provided, both for sequential and parallel executions. Finally, the CPU gains are evaluated on two configurations relevant to nanophotonics.

6.1 DG formulation

In this section, we start again from the variational formulation (3.8), which is reproduced here. For simplicity, the current source term is dropped:

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

with $v \in \{x, y, z\}$. We recall that the i subscript refers to the index of the cell T_i, with $i \in [\![1, N]\!]$. A $\mathbb{P}_p$ polynomial approximation of the field components is used. For the volumic integrals of the above system (*i.e.* mass and stiffness matrices), the development is exactly the same as in section 3.1, and it is therefore not reproduced here. The flux integral, however, must undergo a different treatment. For the sake of simplicity, the centered flux (3.5) is considered. However, the generalization to other fluxes is straightforward. For a given $i \in [\![1, N]\!]$, consider a neighbor cell T_l of T_i with a $\mathbb{P}_m$ polynomial approximation. The flux integral on their common face from the T_i side for the x component is:

$$\begin{aligned}
\int_{a_{il}} (\mathbf{H}_* \times \mathbf{n}_{il}) \cdot \boldsymbol{\phi}_{\boldsymbol{ik}}^{\boldsymbol{x}} &= \int_{a_{il}} \left(H_*^y n_{il}^z - H_*^z n_{il}^y \right) \phi_{ik} \\
&= \int_{a_{il}} \left(\frac{H_i^y + H_l^y}{2} n_{il}^z - \frac{H_i^z + H_l^z}{2} n_{il}^y \right) \phi_{ik}.
\end{aligned} \tag{6.1}$$

Consider the following expansions for H_i^y and H_l^y, and the analogous relations for H_i^z and H_l^z:

$$H_i^y = \sum_{q=1}^{n(p)} H_{iq}^y \phi_{iq} \quad \text{and} \quad H_l^y = \sum_{r=1}^{n(m)} H_{lr}^y \phi_{lr}. \tag{6.2}$$

Plugging (6.2) in (6.1) leads to:

$$
\begin{aligned}
\int_{a_{il}} (\mathbf{H}_* \times \mathbf{n}_{il}) \cdot \boldsymbol{\phi}_{\boldsymbol{ik}}^{\boldsymbol{x}} &= \int_{a_{il}} \frac{n_{il}^z}{2} \left(\sum_{q=1}^{n(p)} H_{iq}^y \phi_{iq} \phi_{ik} + \sum_{r=1}^{n(m)} H_{lr}^y \phi_{lr} \phi_{ik} \right) \\
&\quad - \int_{a_{il}} \frac{n_{il}^y}{2} \left(\sum_{q=1}^{n(p)} H_{iq}^z \phi_{iq} \phi_{ik} + \sum_{r=1}^{n(m)} H_{lr}^z \phi_{lr} \phi_{ik} \right), \\
&= \frac{1}{2} \sum_{q=1}^{n(p)} \left(H_{iq}^y n_{il}^z - H_{iq}^z n_{il}^y \right) \int_{a_{il}} \phi_{iq} \phi_{ik} \\
&\quad + \frac{1}{2} \sum_{r=1}^{n(m)} \left(H_{lr}^y n_{il}^z - H_{lr}^z n_{il}^y \right) \int_{a_{il}} \phi_{lr} \phi_{ik}, \\
&= \frac{1}{2} \sum_{q=1}^{n(p)} \mathbf{H}_{iq} \times \mathbf{n}_{il} \int_{a_{il}} \phi_{iq} \phi_{ik} \\
&\quad + \frac{1}{2} \sum_{r=1}^{n(m)} \mathbf{H}_{lr} \times \mathbf{n}_{il} \int_{a_{il}} \phi_{lr} \phi_{ik}, \\
&= \left(\mathbb{S}_{il} \left(\overline{\mathbf{H}}_{*,i} \times \mathbf{n}_{il} \right) \right)_k^x + \left(\mathbb{S}_{il}^* . \left(\overline{\mathbf{H}}_{*,l} \times \mathbf{n}_{il} \right) \right)_k^x
\end{aligned}
\tag{6.3}
$$

In accordance with the definition of section 3.1.4, the flux matrices are:

$$
\left(\mathbb{S}_{il} \right)_{jk} = \int_{a_{il}} \phi_{ij} \phi_{ik} \quad \text{and} \quad \left(\mathbb{S}_{il}^* \right)_{rk} = \int_{a_{il}} \phi_{lr} \phi_{ik}. \tag{6.4}
$$

The derivation (6.1) easily extends to the other components, as well as other flux choices. To summarize, the flux part is cut in two parts: (i) the part corresponding to local information, which is integrated *via* the "regular" flux matrix, and (ii) the part corresponding to the neighbor information, which is integrated *via* non-conforming matrices. For a $\mathbb{P}_p - \mathbb{P}_m$ interface, the corresponding matrix is rectangular, of size $s(p) \times s(m)$ (we recall that $s(p) = \frac{(p+1)(p+2)}{2}$). Integrated reference matrices for Lagrange $\mathbb{P}_1 - \mathbb{P}_2$, $\mathbb{P}_2 - \mathbb{P}_3$ and $\mathbb{P}_3 - \mathbb{P}_4$ cases are shown in appendix B. From this point, the remaining of the derivation is similar to the standard case, and the reader is referred to section 3.1 for details. The final semi-discrete scheme is:

Locally adaptive semi-discrete scheme

$$
\begin{aligned}
\overline{\mathbb{M}}_i^{\mu_r} \frac{\partial \overline{\mathbf{H}}_i}{\partial t} &= -\overline{\mathbb{K}}_i \times \overline{\mathbf{E}}_i + \sum_{l \in \mathcal{V}_i} \overline{\mathbb{S}}_{il} \left(\overline{\mathbf{E}}_{*,i} \times \mathbf{n}_{il} \right) + \sum_{l \in \mathcal{V}_i} \overline{\mathbb{S}}_{il}^* \left(\overline{\mathbf{E}}_{*,l} \times \mathbf{n}_{il} \right), \\
\overline{\mathbb{M}}_i^{\varepsilon_r} \frac{\partial \overline{\mathbf{E}}_i}{\partial t} &= \quad \overline{\mathbb{K}}_i \times \overline{\mathbf{H}}_i - \sum_{l \in \mathcal{V}_i} \overline{\mathbb{S}}_{il} \left(\overline{\mathbf{H}}_{*,i} \times \mathbf{n}_{il} \right) - \sum_{l \in \mathcal{V}_i} \overline{\mathbb{S}}_{il}^* \left(\overline{\mathbf{H}}_{*,l} \times \mathbf{n}_{il} \right),
\end{aligned}
\tag{6.5}
$$

where the definition of $\overline{\mathbb{S}}_{il}^*$ is easily deduced from (6.4) and the analogous definition of $\overline{\mathbb{S}}_{il}$ given in section 3.1. The time integration can be achieved as in section 3.2.

6.2 h-convergence

We consider the cubic cavity mode of section 2.1.4, and follow the procedure described in section 3.3.1 to compute the numerical convergence rate of the LA-DGTD method. Four meshes of increased resolution are used, whose characteristics are summed up in table 3.2. Unless stated otherwise, a fully upwind flux is used, coupled to an appropriate LSRK4 scheme. In the present case, the used meshes are uniform, and the mesh cells all have the same size. To begin, the polynomial order repatition arbitrarily separates the domain in two halves (see figure 6.1(a)). The results are given in table 6.1. As expected for a $\mathbb{P}_k - \mathbb{P}_l$ configuration, the asymptotic h-convergence order is $\min(k, l) + 1$.

In order to evaluate the impact of non-conforming interfaces, the interpolation order is distributed in four equal stripes instead of two halves (see figure 6.1(b)). Hence, the amount of tetrahedra of each order remains the same as in figure 6.1(a), but the number of interfaces is twice as high. As can be seen on table 6.1, for coarse meshes the higher amount of non-conforming faces yields a slightly higher (but still acceptable) L^2 error. However, this error overhead vanishes for refined meshes. In a last numerical test, another division of the numerical domain in four stripes is used, where each one receives a different order (see figure 6.1(c)). As in the previous cases, and since all the cells are of same size, the numerical error is driven by the lowest order of approximation present in the mesh.

6.3 Order distribution strategy

Starting with a given mesh, it seems obvious that the final repartition of interpolation orders across the different cells will have a major impact on the obtained accuracy, as well as on the CPU time required to obtain the numerical solution. Suppose 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 good distribution of polynomial orders $\mathbb{P}_l$ with $l \leq k$, a solution of similar accuracy can be obtained for a lower computational effort. At first glance, it seems that configurations including small geometrical details, or small gaps between two structures, could benefit from such a strategy. For this reason, for any given mesh, we define the following quantity:

$$\overline{h} = \frac{h_{\text{max}}}{h_{\text{min}}},$$

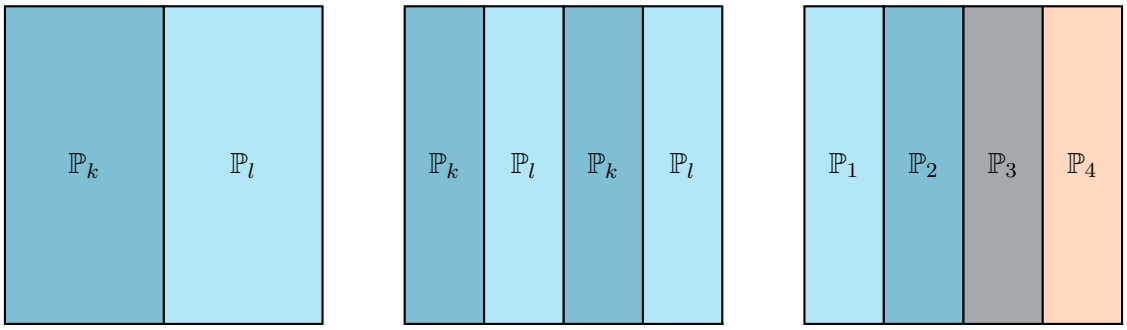
which represents the heterogeneity in terms of cell size in the mesh. In the remaining of this chapter, Δt_i represents the timestep corresponding to the cell T_i, computed following the formula given in section 3.2.5, while Δt_i^p represents the effective timestep obtained if cell T_i is discretized with a $\mathbb{P}_p$ polynomial expansion:

$$\Delta t_i^p = \text{CFL}(p)\Delta t_i.$$

The normalized timestep includes a rough estimate of the computational charge induced by the polynomial order, and is defined as:

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

where $n(p)$ is the number of degrees of freedom inside a $\mathbb{P}_p$ cell. Finally, we define p_{min} and p_{max} the minimal and maximal user-authorized orders in the mesh. We also add the following constraint: non-conforming faces cannot connect cells with an order jump higher than one (the allowed configurations are presented on figure 6.2). Indeed, it is preferable to restrain the number and size of matrices in memory



(a) Test configuration for h-convergence

(b) Influence of the number of non-conforming interfaces

(c) Influence of the type of non-conforming interfaces

Figure 6.1 | Order distribution for the h-convergence validation. To test convergence, the domain is arbitrarily cut in two halves, each part receiving a different order (figure 6.1(a)). In order to assess the impact of non-conforming interfaces, the domain is cut in four equal stripes, thus doubling the number of $\mathbb{P}_k - \mathbb{P}_l$ faces while keeping the same number of tetrahedron per order (figure 6.1(a)). Another test is conducted by setting a different order in each quarter (figure 6.1(c)).

Table 6.1 | Error levels and convergence rates of the cubic cavity case for mixed orders of approximation on meshes of increasing refinement. In the case of mixed orders $\mathbb{P}_k - \mathbb{P}_l$, 1 refers to a domain cut in two halves (see figure 6.1(a)), and 2 to a domain cut in four stripes (see figure 6.1(b)). The case $\mathbb{P}_1 - \mathbb{P}_4$ corresponds to a domain cut in four quarters, as depicted on figure 6.1(c). All simulations were run with upwind fluxes and LSRK4 time integration.

		M1		**M2**		**M3**		**M4**	
		$\|\mathbf{E} - \mathbf{E}_h\|$	$\overline{h}$	$\|\mathbf{E} - \mathbf{E}_h\|$	$\overline{h}$	$\|\mathbf{E} - \mathbf{E}_h\|$	$\overline{h}$	$\|\mathbf{E} - \mathbf{E}_h\|$	$\overline{h}$
$\mathbb{P}_1$	–	2.87×10^{-1}	–	6.05×10^{-2}	2.25	8.66×10^{-3}	2.80	1.46×10^{-3}	2.57
$\mathbb{P}_2$	–	1.47×10^{-2}	–	1.36×10^{-3}	3.43	1.75×10^{-4}	2.96	2.19×10^{-5}	3.00
$\mathbb{P}_3$	–	9.24×10^{-4}	–	5.87×10^{-5}	3.98	3.72×10^{-6}	3.98	2.33×10^{-7}	4.00
$\mathbb{P}_4$	–	9.45×10^{-5}	–	3.11×10^{-6}	4.92	1.98×10^{-7}	3.97	1.15×10^{-8}	4.11
$\mathbb{P}_1 - \mathbb{P}_2$	1	2.65×10^{-1}	–	3.38×10^{-2}	2.97	6.15×10^{-3}	2.46	1.42×10^{-3}	2.12
	2	2.07×10^{-1}	–	3.55×10^{-2}	2.54	6.30×10^{-3}	2.49	1.43×10^{-3}	2.14
$\mathbb{P}_2 - \mathbb{P}_3$	1	8.50×10^{-3}	–	9.21×10^{-4}	3.21	1.18×10^{-4}	2.96	1.48×10^{-5}	3.00
	2	8.70×10^{-3}	–	9.16×10^{-4}	3.25	1.17×10^{-4}	2.97	1.48×10^{-5}	2.98
$\mathbb{P}_3 - \mathbb{P}_4$	1	6.73×10^{-4}	–	4.41×10^{-5}	3.93	2.80×10^{-6}	3.98	1.76×10^{-7}	4.00
	2	6.81×10^{-4}	–	4.47×10^{-5}	3.93	2.85×10^{-6}	3.97	1.79×10^{-7}	3.99
$\mathbb{P}_1 - \mathbb{P}_4$	–	2.65×10^{-1}	–	3.38×10^{-2}	2.97	6.15×10^{-3}	2.46	1.42×10^{-3}	2.12

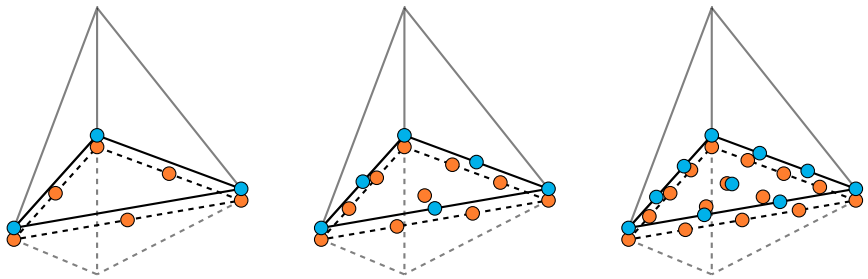


Figure 6.2 | Authorized interfaces in the local order of approximation implementation. Order jumps are limited to one, yielding three types of interfaces for $(p_{min}, p_{max}) = (1, 4)$.

in order to improve data locality. Additionaly, it leads to a robust distribution strategy, as will be shown hereafter.

The first step of the algorithm consists in computing the local Δt_i, and sorting them by ascending order. The cell of lower timestep receives order p_{min}, and we compute its normalized timestep $\overline{\Delta t}_1^{\,p_{\text{min}}}$. Two temporary variables, p_{loc} and $\overline{\Delta t}_{\text{loc}}$, respectively store the current order assigned to elements and the current restrictive normalized timestep. For a given cell, the normalized timestep for increased order $p_{\text{loc}}+1$ is compared to the current limiting normalized timestep $\overline{\Delta t}_{\text{loc}}$. In the case where the first is higher than the second, switching to the higher order is assumed to have a limited impact on the final timestep. Hence, p_{loc} is increased by one, and $\overline{\Delta t}_{\text{loc}}$ is updated. The procedure is summarized in algorithm 2, whose performances will be assessed in next section.

Algorithm 2 Polynomial order distribution

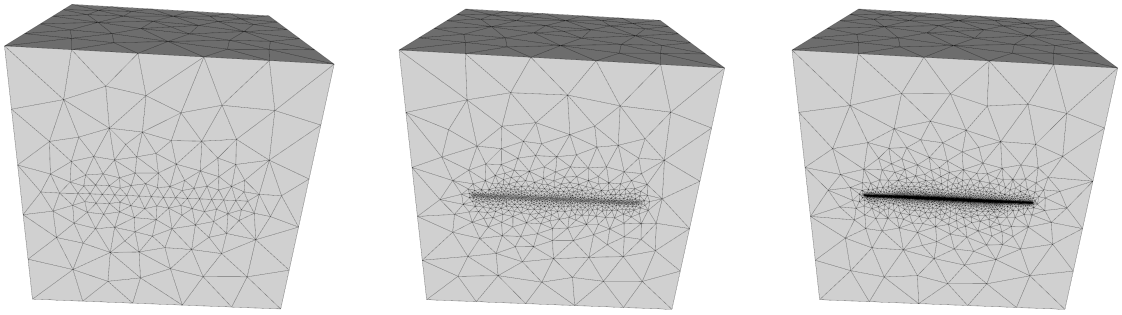
for $i \leftarrow 1, n_t$ **do** ▷ Compute timestep for each cell
  Compute Δt_i
end for

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

for $i \leftarrow 2, n_t$ **do** ▷ Go over cells by ascending order of Δt_i
  if $p_{\text{loc}} + 1 > p_{\text{max}}$ **then** ▷ Check that we do not exceed p_{max}
    $p(i) \leftarrow p_{\text{max}}$
  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 timestep
      $p(i) \leftarrow p_{\text{loc}} + 1$
      $p_{\text{loc}} \leftarrow p_{\text{loc}} + 1$ ▷ Update the current order
    else
      $p(i) \leftarrow p_{\text{loc}}$
    end if
  end if
end for

Table 6.2 | Characteristics of the locally refined cubic cavity meshes. n_s is the number of vertices, n_t the number of tetrahedrons and $\overline{h}$ is the ratio between the largest and the smallest cells in the mesh.

	M'1	M'2	M'3
n_s	427	1429	11975
n_t	1513	5042	42652
$\overline{h}$	10.2	100.9	1000.8



(a) M'1, $\overline{h} = 10$ (b) M'2, $\overline{h} = 100$ (c) M'3, $\overline{h} = 1000$

Figure 6.3 | Meshes with local refinements for the cubic cavity mode.

6.4 Performance assessment

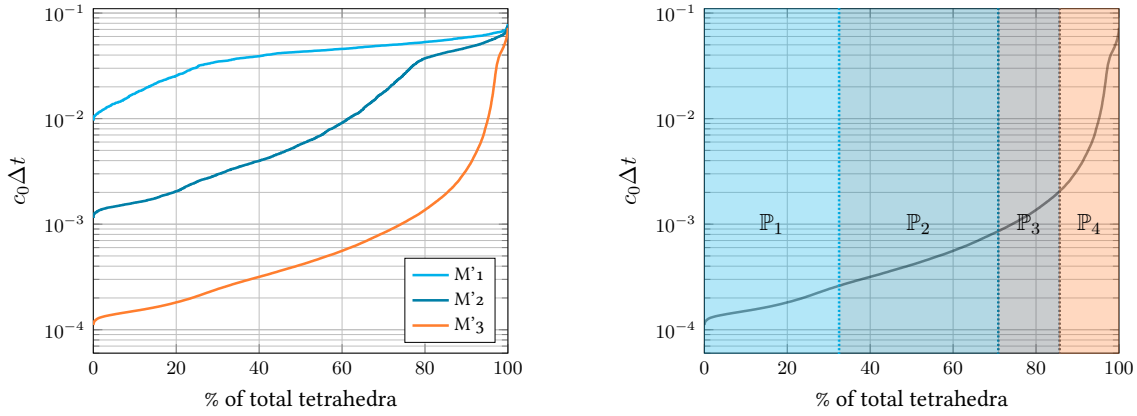
6.4.1 Sequential speedup

To evaluate the gains provided by the LA implementation, we consider the three meshes M'1, M'2 and M'3 shown on figure 6.3. These meshes are obtained by the tesselation of a cubic cavity corresponding to test-case of section 2.1.4, a local refinement being imposed on one side of the box. The characteristics of these meshes are summed up in table 6.2. For each mesh, the cavity mode is computed sequentially for 5 periods. As before, CPU time and maximum L^2 error are stored. The results obtained for homogeneous and mixed orders are presented in table 6.3.

First, it must be noted that the memory occupation values result from a non-optimal implementation. Hence, it is expected that mixed orders computations require the same memory size than homogeneous order ones. For mixed order solutions, the speedup of a $\mathbb{P}_k - \mathbb{P}_l$ computation is obtained by comparing its CPU time with that of a full $\mathbb{P}_{\max(k,l)}$ computation. On the three considered meshes, mixing two polynomial orders leads to speedups ranging from 1.5 to 2.2. An interesting point is that the obtained L^2-errors are identical or less than 1% higher than those of the homogeneous polynomial approximations. Mixing three orders does not provide any improvement for the M'1 mesh, *i.e.* the distribution algorithm attributed no cells to the highest order. This can be easily understood by looking at the compared timestep distribution of the three meshes (see figure 6.4): since the algorithm imposes the lowest order for the cell of smallest timestep, a minimal amplitude in the timestep distribution is required to reach higher orders. As an example, the $\mathbb{P}_1 - \mathbb{P}_4$ distribution is shown on the same figure for mesh M'3. For M'2 and M'3, however, very interesting gains are obtained, with speedups ranging from 3 to 4.5. In this case, it seems that a higher $\overline{h}$ implies a higher benefit from the LA algorithm. Finally, mixing orders from 1 to 4 roughly

Table 6.3 | CPU times, memory consumption and error levels for mixed orders of approximation on locally refined meshes. The order repartition was obtained *via* the procedure described in section 6.3. All simulations were run with upwind fluxes and LSRK$_4$ time integration.

		M'1	**M'2**	**M'3**
$\mathbb{P}_1$	CPU (s.)	1.77	48.7	6030
	Mem. (MB)	14.1	22.1	107.3
	$\|\|\mathbf{E}-\mathbf{E}_h\|\|$	3.38×10^{-2}	3.59×10^{-2}	3.87×10^{-2}
$\mathbb{P}_2$	–	6.50	180	22460
		17.8	34.4	211
		3.68×10^{-3}	3.71×10^{-3}	3.82×10^{-3}
$\mathbb{P}_3$	–	21.8	611	90020
		23.9	54.6	382
		2.14×10^{-4}	2.39×10^{-4}	2.53×10^{-4}
$\mathbb{P}_4$	–	73.8	2106	228220
		32.9	84.7	635
		1.42×10^{-5}	1.76×10^{-5}	1.97×10^{-5}
$\mathbb{P}_1-\mathbb{P}_2$	CPU (s.)	3.84	101	15000
	Speedup	1.69	1.78	1.50
	Mem. (MB)	17.8	34.4	211
	Tet. ratios	0.18, 0.82	0.29, 0.71	0.33, 0.67
	$\|\|\mathbf{E}-\mathbf{E}_h\|\|$	3.84×10^{-3}	3.71×10^{-3}	3.82×10^{-3}
$\mathbb{P}_2-\mathbb{P}_3$	–	14.3	372	40470
		1.52	1.64	2.22
		23.9	54.6	382
		0.25, 0.75	0.38, 0.62	0.43, 0.57
		2.38×10^{-4}	2.39×10^{-4}	2.53×10^{-4}
$\mathbb{P}_3-\mathbb{P}_4$	–	49.2	1390	130730
		1.5	1.51	1.74
		33.0	84.8	635
		0.17, 0.83	0.28, 0.72	0.31, 0.69
		1.42×10^{-5}	1.76×10^{-5}	1.97×10^{-5}
$\mathbb{P}_1-\mathbb{P}_2-\mathbb{P}_3$	–	–	180	19890
			3.39	4.53
			54.7	392
			0.29, 0.31, 0.40	0.33, 0.38, 0.29
			2.39×10^{-4}	2.53×10^{-4}
$\mathbb{P}_2-\mathbb{P}_3-\mathbb{P}_4$	–	–	695	64820
			3.03	3.52
			84.8	635
			0.38, 0.22, 0.40	0.43, 0.27, 0.30
			1.76×10^{-5}	1.97×10^{-5}
$\mathbb{P}_1-\mathbb{P}_2-\mathbb{P}_3-\mathbb{P}_4$	–	–	347	37130
			6.07	6.15
			84.9	636
			0.29, 0.32, 0.13, 0.26	0.33, 0.38, 0.15, 0.14
			1.80×10^{-5}	1.99×10^{-5}



(a) Timestep distribution for meshes M'1, M'2 and M'3

(b) Order distribution for mesh M'3

Figure 6.4 | Compared timestep distribution in M'1, M'2 and M'3 (left), and order distribution for the $\mathbb{P}_1 - \mathbb{P}_4$ case on M'3 (right).

provides a speedup of 6, while barely increasing the global L^2 error (less than 1%).

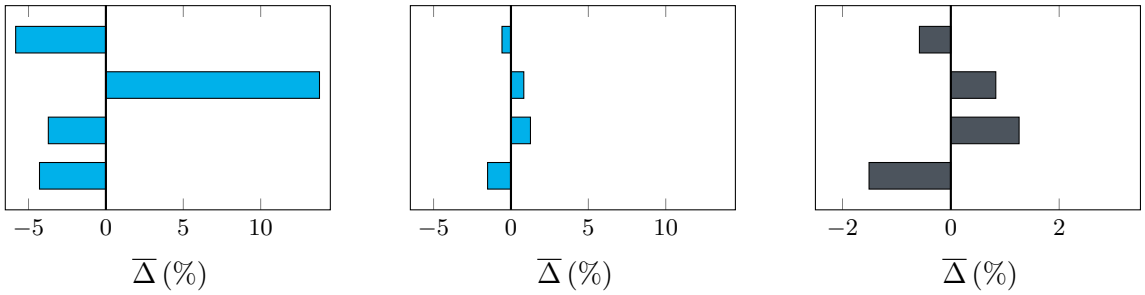
6.4.2 Parallel load balance

In this section, we present the results obtained when trying to balance the computational load for a parallel implementation of the LA-DGTD method. The Metis [KK99] graph partitioning tool is used to split the computational domain in subdomains, each of which is then mapped on a core. The communications between the cores are handled *via* the MPI standard.

At the end of each computation, each CPU core returns its own CPU time, excluding the time spent in the MPI communication routines. Hence, a good load balance between the cores will manifest as nearly-identical CPU times for all cores. To reach this result, the Metis package can be provided a weight w_i for each cell T_i. During the partitioning, this weight is taken into account so that the total weight of the various subdomains are as close as possible. Here, the following weight is used:

$$w_i = n(p_i) + \sum_{l \in \mathcal{V}_i} \max(s(p_i), s(p_l)) \tag{6.6}$$

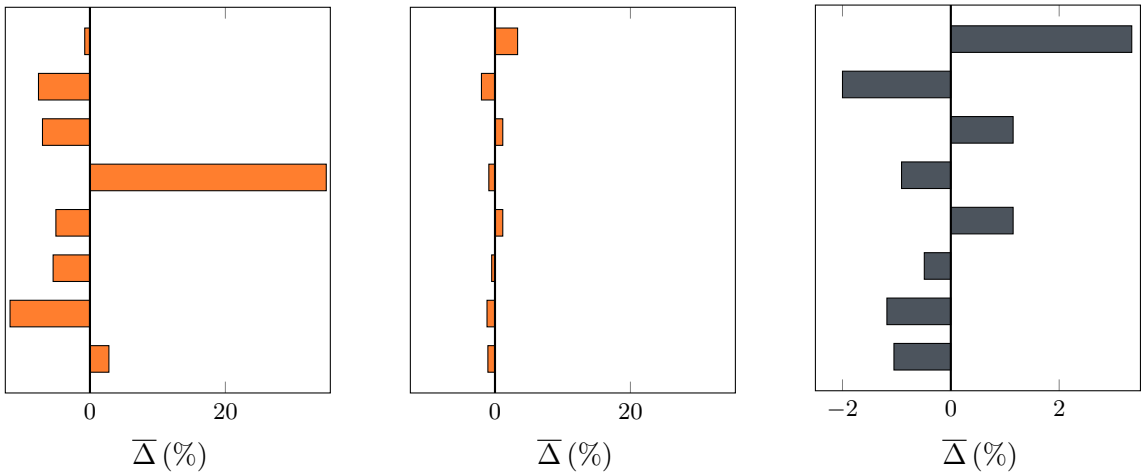
The weighting (6.6) is tested on the cavity mode case using the M'3 mesh and running 100 time iterations. Given the small number of tetrahedra in the mesh, the study is limited to 4 and 8 subdomains partitions. For a simpler comparison between the two partitionnings, the relative deviation $\overline{\Delta}$ (in %) to the mean CPU time is computed for each core. First, the effect of the weighting is assessed by comparing relative deviations obtained from weighted and non-weighted partitions. The results for $\mathbb{P}_1 - \mathbb{P}_4$ approximation with 4 and 8 CPUs are shown on figure 6.5. As can be seen, the use of a weighting pattern is mandatory to preserve good performances in parallel. For both 4 and 8 cores, applying the weighted partitionning results in a maximal relative imbalance lower than 5%. To further explore the behavior of the algorithm in parallel, the CPU load balances with 4 and 8 cores are re-plotted with matching scales. As expected, increasing the number of cores also increases the maximal imbalance between cores, though in reasonable bounds.



(a) $\mathbb{P}_1 - \mathbb{P}_4$ with 4 cores, non-weighted

(b) $\mathbb{P}_1 - \mathbb{P}_4$ with 4 cores, weighted

(c) $\mathbb{P}_1 - \mathbb{P}_4$ with 4 cores, weighted



(d) $\mathbb{P}_1 - \mathbb{P}_4$ with 8 cores, non-weighted

(e) $\mathbb{P}_1 - \mathbb{P}_4$ with 8 cores, weighted

(f) $\mathbb{P}_1 - \mathbb{P}_4$ with 8 cores, weighted

Figure 6.5 | Weighted vs non-weighted parallel load balance on 4 and 8 cores with M'3 mesh. Each bar corresponds to the relative imbalance of a single CPU to the average value computed over all processors. As can be seen, the weighting restores a good balance of the CPU loads, with no relative imbalance exceeding 5%.

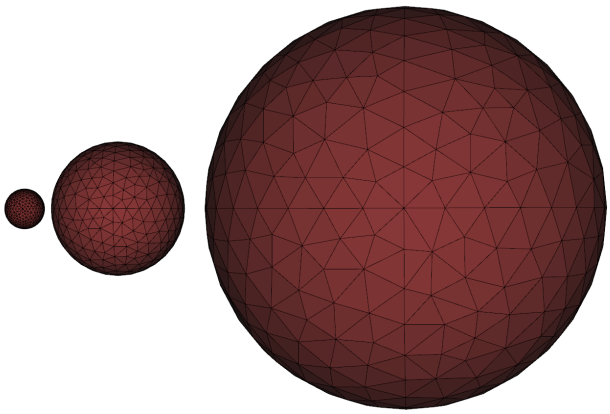


Figure 6.6 | Nanolens composed of three metallic spheres.

6.5 Plasmonic nanolens

This section presents the computation of the field enhancement obtained in a plasmonic nanolens device. To overcome the limitation of the diffraction limit, it is possible to exploit the focusing effect provided by coupled surface plasmons [DCTS08]. The typical nanolens is composed of a chain of metallic nanoparticles (nanospheres being the most common) of decreasing size, aligned with the polarization direction of the incident field (see figure 6.6). When the nanospheres are of significantly different sizes, the local field enhancement of the first particle is not perturbed by the second one because of its small relative size. As a result, the locally enhanced field of the first particle acts as an incident field for the second particle, resulting in a second enhancement, and so on. Eventually, the strongest enhancement is obtained in the gap between the two smaller particles [LSB03].

Here, we consider a nanolens made of three gold spheres. The geometry is taken from [LSB03]: the respective radii are 45, 15 and 4.5 nanometers, while the spacings between the sphere surfaces are respectively 4.5 and 1.5 nanometers. The gold is described by a Drude model of parameters $\varepsilon_\infty = 3.7362$, $\omega_d = 1.387 \times 10^7$ GHz and $\gamma_d = 4.515 \times 10^4$ GHz. The nanolens is illuminated from above with a wideband plane wave of central frequency 700 THz, which polarization is aligned with the natural axis of the lens (here, the $x+$ direction). Finally, a probe point is set at half-distance between each pair of spheres. At these positions, the discrete Fourier transform of the field is computed, and the field enhancement $g = \frac{|\widehat{E}_x|}{|\widehat{E}_i|}$ is deduced. To obtain a proper resolution of the geometry, very small elements must be used on the surface of the smallest sphere, as well as in the smallest gap, while the rest of the geometry (largest sphere, vacuum and PMLs) are meshed with much coarser elements (see mesh on figure 6.7). As a result, the $\overline{h}$ factor is here superior to 800. To obtain convergence with an homogeneous order over the whole mesh, $\mathbb{P}_3$ approximation is required: the solution is obtained in 49 hours 48 minutes on 16 cores, and is taken as a reference. To exploit the LA-DGTD method, $\mathbb{P}_1$ to $\mathbb{P}_3$ approximations are used: the order distribution with respect to timestep is shown on figure 6.8(a). To further understand the behavior of the repartition algorithm, a visual representation is added on figure 6.8(b). As expected, first-order polynomials are attributed to the surface of the smallest sphere, while second-order approximation is used in its vicinity and for the surface of the second sphere. The rest of the mesh is discretized with third-order polynomials.

The computed field enhancements are presented on figure 6.9. As stated in the litterature, particularly intense fields are obtained between the two smallest sphere, where enhancements up to 700 are observed. The $\mathbb{P}_1 - \mathbb{P}_3$ solution is obtained in 19 hours and 17 minutes, hence providing a speedup of 2.6 over the full

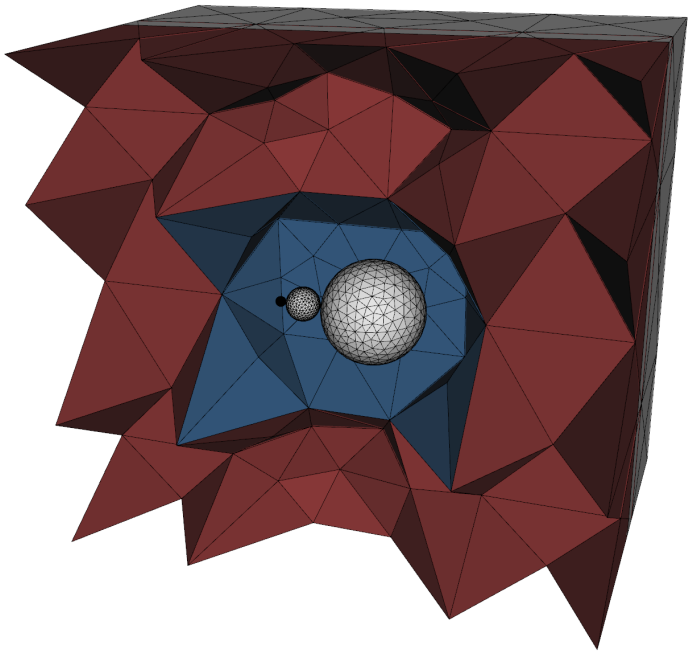
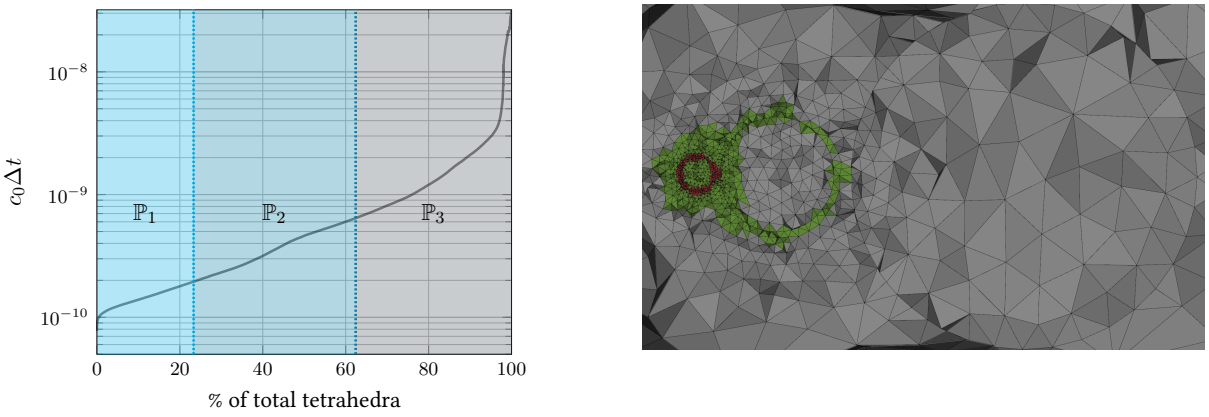


Figure 6.7 | Mesh setup for a metallic nanolens. The gray cells correspond to the metallic spheres, the blue cells to vacuum, while the red cells constitute the PML region. For this mesh, the ratio $\overline{h}$ is above 800.



(a) Order distribution for the nanolens mesh

(b) Order selection in the vicinity of the lens

Figure 6.8 | Polynomial order repartition for the nanolens mesh with respect to timestep (left), and geometrical repartition (right) for the $\mathbb{P}_1 - \mathbb{P}_3$ case. The red elements correspond to $\mathbb{P}_1$ approximation, the green ones to $\mathbb{P}_2$, and the gray ones to $\mathbb{P}_3$.

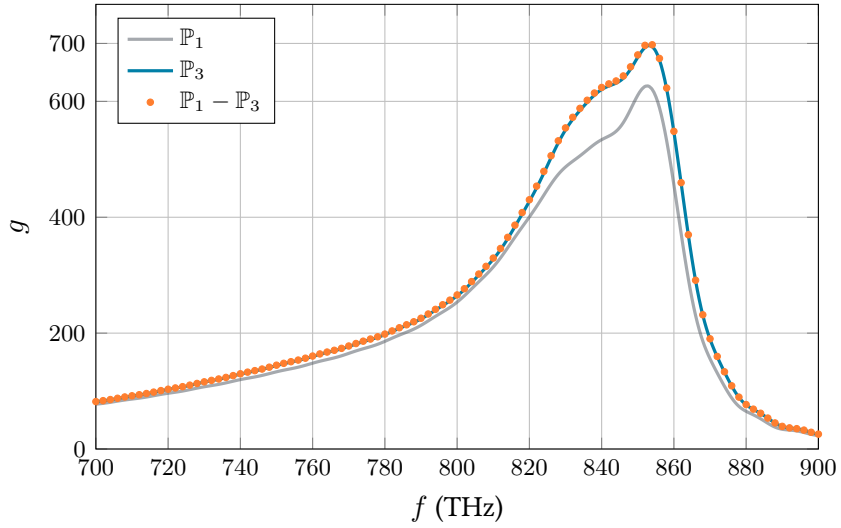


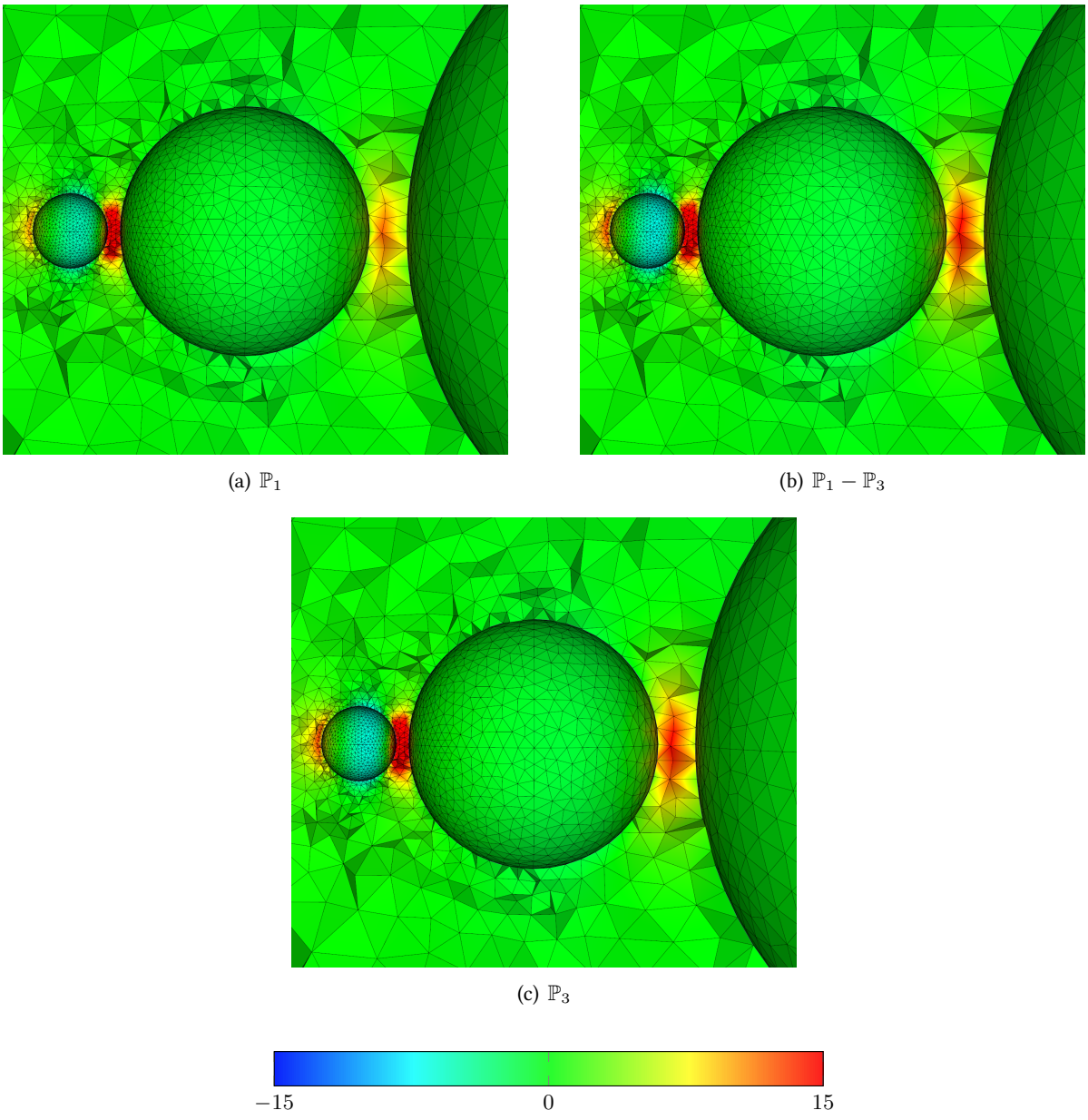
Figure 6.9 | Field enhancement in the vicinity of the smallest sphere of a self-similar nanolens obtained with $\mathbb{P}_1$, $\mathbb{P}_3$ and $\mathbb{P}_1 - \mathbb{P}_3$ approximations. Less than 1 % of relative error is observed between full $\mathbb{P}_3$ and $\mathbb{P}_1 - \mathbb{P}_3$ computations, for a speedup factor of 2.6.

$\mathbb{P}_3$ solution. The observed error over the frequency range of interest is less than 1 %. To further illustrate the alikeness between those results, time-domain field maps are plotted on figure 6.10. As can be seen, there is almost no difference between $\mathbb{P}_1 - \mathbb{P}_3$ and full $\mathbb{P}_3$ approximations. However, the full $\mathbb{P}_1$ solution clearly underestimates the amplitude of the field between the spheres.

6.6 Bowtie nanoantenna

In this section, we focus on the computation of the extinction cross-section of a metallic bowtie nanoantenna. These structures are well-known for the very strong field enhancement they provide between the tips of the two triangular nanoparticles (see figure 6.11), which is known to be inversely proportionnal to the size of the gap. Hence, bowtie nanoantennas are good candidates for surface-enhanced Raman spectroscopy (SERS) applications [HHG+10]. Recent advances in lithography techniques allowed the creation of structures with gaps as small as 3 nm [ZIC14], while the typical size of the full structure can get close to 200 nm. Additionally, realistic geometries of such antennas include small roundings at the edges and tips, which typical size is between a few to a few tens of nanometers [GPK14].

In this case, we consider a pair of 10 nm-thick, equilateral prisms of edge length 100 nm, with a spacing gap of 3 nm. The rounding radius is 2 nm, and is uniform for all edges and tips. The considered material is gold, described by a Drude model of parameters $\varepsilon_\infty = 3.7362$, $\omega_d = 1.387 \times 10^7$ GHz and $\gamma_d = 4.515 \times 10^4$ GHz. The nanoantenna is enclosed in a TF/SF interface (see section 4.2.2), and the domain is terminated by a layer of CFS-PML tetrahedra (see section 4.1.2). As can be seen on figure 6.12, the typical setup for such computations requires very small elements (geometrical details of the nanoantenna) as well as very large ones (vacuum and PML cells), and could therefore make good use of the LA-DGTD formulation presented before. To compute the exctinction cross-section, the bowtie is illuminated from above with a wide-band plane wave of central frequency 500 THz, polarized along the major axis of the antenna. With an homogeneous polynomial order on the whole mesh, convergence is obtained for a $\mathbb{P}_3$ approximation, requiring 30 hours and 37 minutes on 16 cores. This is taken as



(a) $\mathbb{P}_1$

(b) $\mathbb{P}_1 - \mathbb{P}_3$

(c) $\mathbb{P}_3$

Figure 6.10 | E_y field map in the nanolens device at $t = 10$ fs. For the three views, the field values are scaled to $[-15, 15]$.

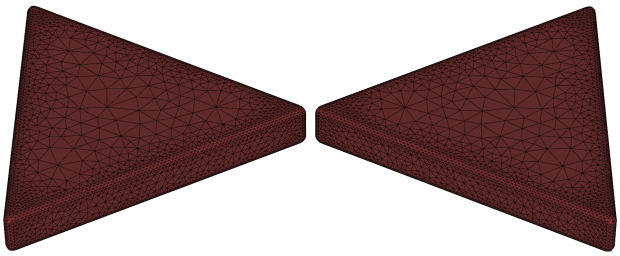


Figure 6.11 | Bowtie nanoantenna with rounded edges.

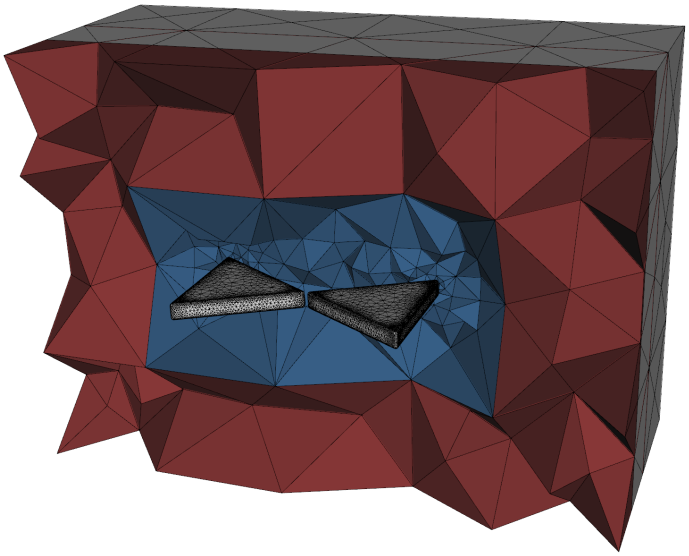
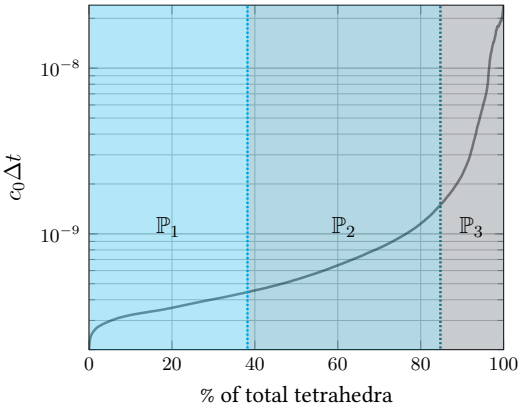


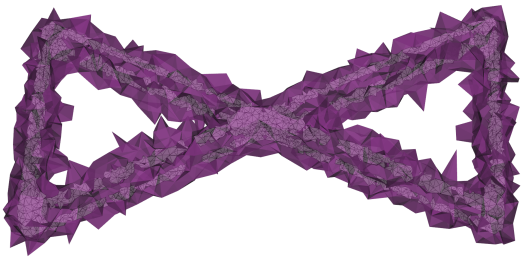
Figure 6.12 | Mesh setup for a bowtie nanoantenna. The gray cells correspond to the nanoantenna, the blue cells to vacuum, while the red cells constitute the PML region. For this mesh, the ratio $\frac{}{h}$ is close to 275.

a reference for LA-DGTD, for which $\mathbb{P}_1$ to $\mathbb{P}_3$ approximations are used. The repartition of orders with regards to the timestep is presented on figure 6.13, along with a visual representation of the order selection in the mesh. As expected, the first order is attributed to the small cells of the edges and tips, which are then enclosed into a second layer of $\mathbb{P}_2$ elements. All the remaining cells (not represented) receive a third order interpolation.

The computed exctinction cross-sections are presented on figure 6.14. A single very large resonance is observed around 418 THz. As can be seen, the $\mathbb{P}_1 - \mathbb{P}_3$ solution properly fits the full $\mathbb{P}_3$ solution, with a deviation of less than 2 %. For further comparison, the full $\mathbb{P}_1$ solution is also plotted. In terms of performance, the $\mathbb{P}_1 - \mathbb{P}_3$ solution is obtained in 13 hours and 59 minutes, hence yielding a 2.19 speedup factor, which is lower than what was observed in section 6.5: the difference can be attributed to the lower proportion of high-order ($\mathbb{P}_3$) elements compared to low-order ones (less than 20 % of $\mathbb{P}_3$ elements here, against 40 % for the nanolens case). However, this remains an appreciable gain for a solution of similar accuracy. As an illustration, a field map of $|\mathbf{E}|$ is plotted on figure 6.15 for $\mathbb{P}_1 - \mathbb{P}_3$ approximation.



(a) Order distribution for the bowtie mesh



(b) Order selection in the vicinity of the antenna

Figure 6.13 | Polynomial order repartition for the bowtie mesh with respect to timestep (left), and geometrical repartition (right) for the $\mathbb{P}_1 - \mathbb{P}_3$ case. The white elements correspond to $\mathbb{P}_1$ approximation, while the purple cells are second-order. The remaining cells (not represented) receive third order approximation.

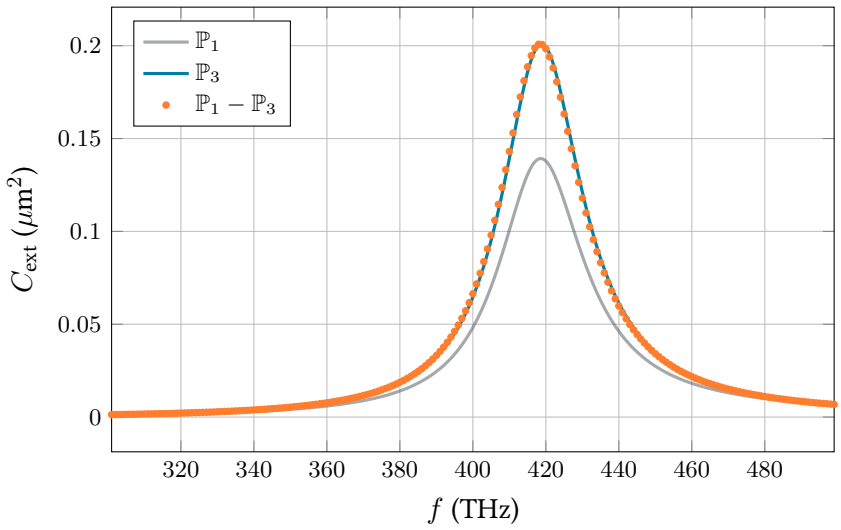
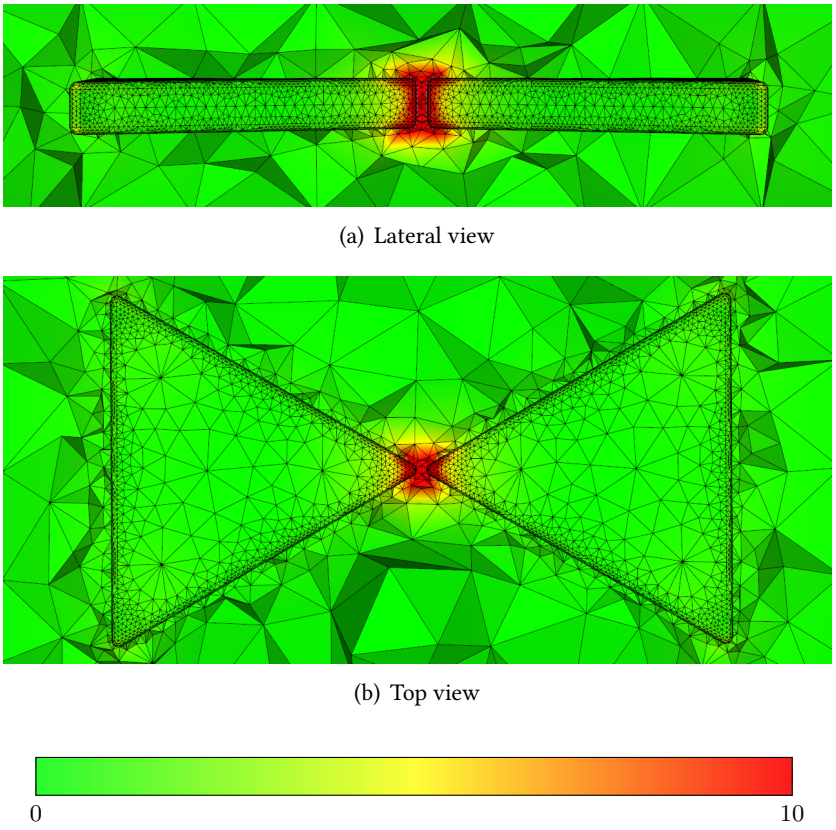


Figure 6.14 | Extinction cross-section of the bowtie nanoantenna obtained with $\mathbb{P}_1$, $\mathbb{P}_3$ and $\mathbb{P}_1 - \mathbb{P}_3$ approximations. Less than 2 % of relative error is observed between full $\mathbb{P}_3$ and $\mathbb{P}_1 - \mathbb{P}_3$ computations, for a speedup factor superior to 2.



(a) Lateral view

(b) Top view

Figure 6.15 | |E| field map in the bowtie device at $t = 12.3$ fs, obtained with a $\mathbb{P}_1 - \mathbb{P}_3$ approximation. The field values are scaled to $[0, 10]$.

6.7 Conclusion

In this chapter, the use of local polynomial approximation in the DGTD method was presented. We have demonstrated the convergence of the algorithm on a standard PEC cavity case. Then, an order-repartition algorithm was proposed that proved to be efficient, both for textbook and realistic nanophotonics-related cases. Although the obtained speedups were lower for realistic cases (between 2 and 2.6 for $\mathbb{P}_1 - \mathbb{P}_3$) than for academic cases (up to 4.5 for $\mathbb{P}_1 - \mathbb{P}_3$ and 6.15 for $\mathbb{P}_1 - \mathbb{P}_4$), the LA-DGTD algorithm represents an interesting gain in speed for day-long nano-optics computations. However, the repartition algorithm being based on a timestep approach, it also implicitely relies on a basic knowledge of the physical behavior of the computed system (the preliminary grasp of the positions of intense fields, basically). A good remedy would consist in a coupling of the algorithm with an *a posteriori* error estimate, in order to dynamically adapt the polynomial order, and possibly the mesh discretization.

7 Improving performances

In a vast majority of nano-optics problems, the computation of the direct problem cannot be performed in a reasonable time on a single processor. After a phase of optimization of the sequential performances, elaborating an efficient parallel implementation represents a crucial step toward realistic and challenging nanophotonics computations. In this chapter, the parallel performances of our DGTD implementation are assessed. First, a renumbering method to improve data locality is proposed, following [LMDL15]. Then, the performances of a SPMD implementation are assessed.

7.1 Reverse Cuthill-McKee renumbering

The computation of the flux on a triangular face (see section 3.1.3), requires that the fields on both sides of the face are known simultaneously. For large enough problems, it is most probable that these informations must be retrieved from the global memory (usually RAM) to the local cache memory. Then, in a very short set of instructions, two memory accesses are required to two memory areas whose proximity is not ensured *a priori*. Therefore, the amount of time spent waiting for the data to flow back to local cache can be severely highered if the locality is not ensured in memory. A natural solution consists in renumbering the initial connectivity matrix to lower its bandwidth *via* the reverse Cuthill-McKee (RCM) algorithm [LS76]. When done before array allocations, this procedure ensures a better locality of the data in memory for neighbouring cells, thus leading to a reduced addressing time.

As an example, we consider the mesh presented on figure 7.1. This mesh is used in section 5.3 for the computation of a gold nanosphere plasmonic resonance. Its size (see table 5.5) and complexity make it a basic yet representative example of the gains obtained with RCM. On figure 7.2, the non-zero elements of the connectivity matrix are represented before and after the RCM processing: as can be seen, the bandwidth is largely reduced, dropping from a maximal value of 116647 cells to only 6387 cells. To assess the gains obtained with the RCM renumbering, we run 100 time iterations of the DGTD method on a single core, for polynomial orders ranging from 1 to 4. The results are summed up in table 7.1: as can be seen, good speedups are obtained for low orders of approximation, while for higher orders the gain is

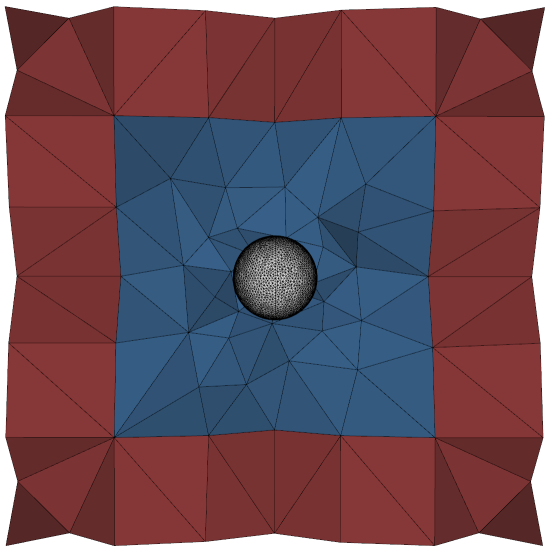
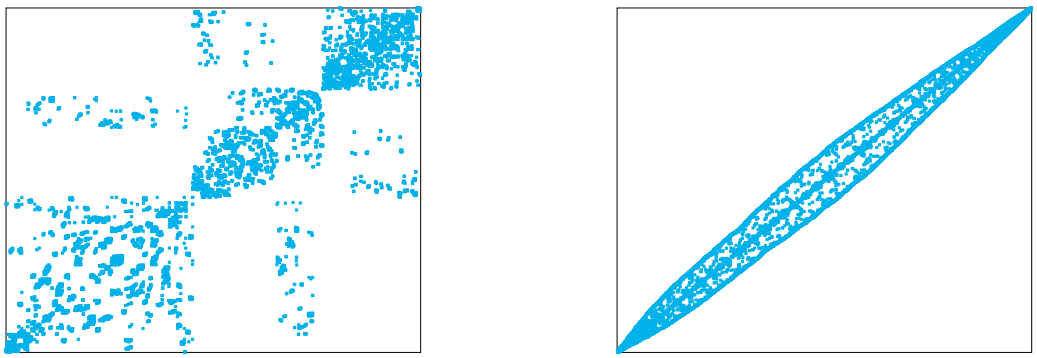


Figure 7.1 | Nanosphere mesh for RCM study.

lower. Indeed, for low orders, the number of tetrahedra for which the full field can be held in the cache is higher. Hence, for a given tetrahedron, the cache reuse ratio is higher with the RCM renumbering, which eventually leads to a reduced number of memory accesses.

7.2 Performances of a non-blocking MPI implementation

In this section, the parallel performances of our MPI DGTD code are assessed for the classic PEC cavity case, detailed in section 2.1.4 with a mesh made of roughly 1,000,000 cells. This very simple configuration induces a minimal imbalance due to the boundary conditions, thus allowing a fair evaluation of the partitioning quality and the solver performance.



(a) Before RCM (b) After RCM

Figure 7.2 | Impact of the RCM renumbering on the connectivity matrix. Here, the connectivity matrix bandwidth is reduced from 116647 to 6387.

Table 7.1 | Sequential speedup with the RCM algorithm for orders from 1 to 4, for 100 iterations of the Drude nanosphere test-case.

	$\mathbb{P}_1$	$\mathbb{P}_2$	$\mathbb{P}_3$	$\mathbb{P}_4$
Computational time without RCM (sec)	115	267	530	1201
Computational time with RCM (sec)	89.8	221	466	1155
Speedup (%)	22.5	17.4	12.1	4.0

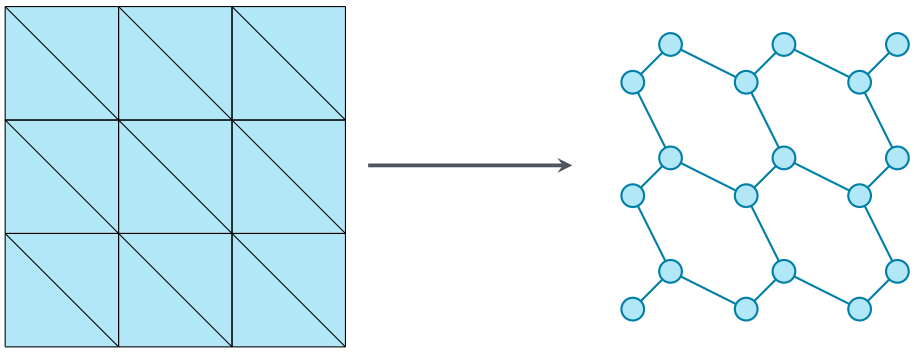


Figure 7.3 | Mesh to graph conversion for partitioning.

7.2.1 Mesh partitioning

The mesh partitioning is performed with the Metis graph partitionner [KK99]. A pre-processing step converts the tetrahedral mesh in a graph structure, where cells are associated to nodes, while faces are associated to vertices (see figure 7.3). Then, the partitionning of the graph produces a set of subgraphs that correspond to the final MPI subdomains. An example of a partitionned mesh is given on figure 7.4, where each colour represents a subdomain. As can be expected, the partitions quality will have a direct impact on the parallel performances of the solver. However, it can prove difficult to properly quantify it: in the following study, it is considered as an inherent part of the final performance of the solver. As an example, we present on figure 7.5 the statistics (deviation from the mean for vertices and tetrahedra repartition, as well as the number of neighbours per subdomain) for the M1 cavity mesh, for a number of subdomains ranging from 1 to 256.

7.2.2 Strong scaling

To measure the actual gain obtained with a parallel implementation, a fixed size problem is considered, while the number of processing units dedicated to its solving is progressively increased. In the following, the time required to compute 100 time iterations of the problem on n CPUs is noted t_n. The performance of the parallel implementation is then given either by the measured speedup $s = \frac{t_1}{t_n}$ or efficiency $e = \frac{t_1}{n\,t_n}$.

Scalability and point-to-point data exchange

The cluster used for the following study is composed of blades, each holding 2 8-cores Intel(R) E5-2670 2.6GHz processors. Here, we want to point out the different types of data exchange used between the cores, depending on whether they are on the same blade, or on a different blade. To do so, we plot in figure 7.6 the total simulation time required to compute 100 time iterations with a $\mathbb{P}_1$ approximation, from

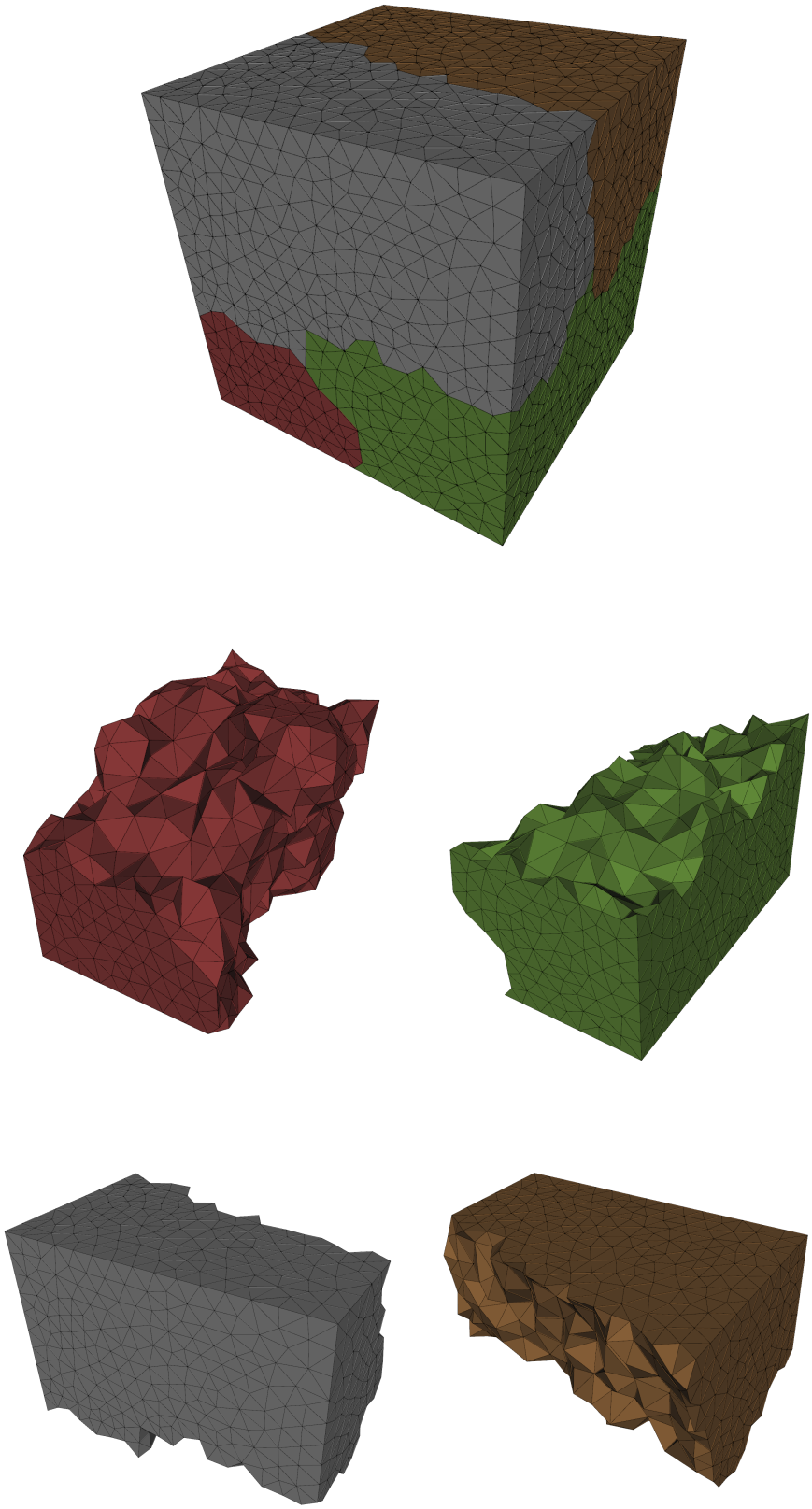
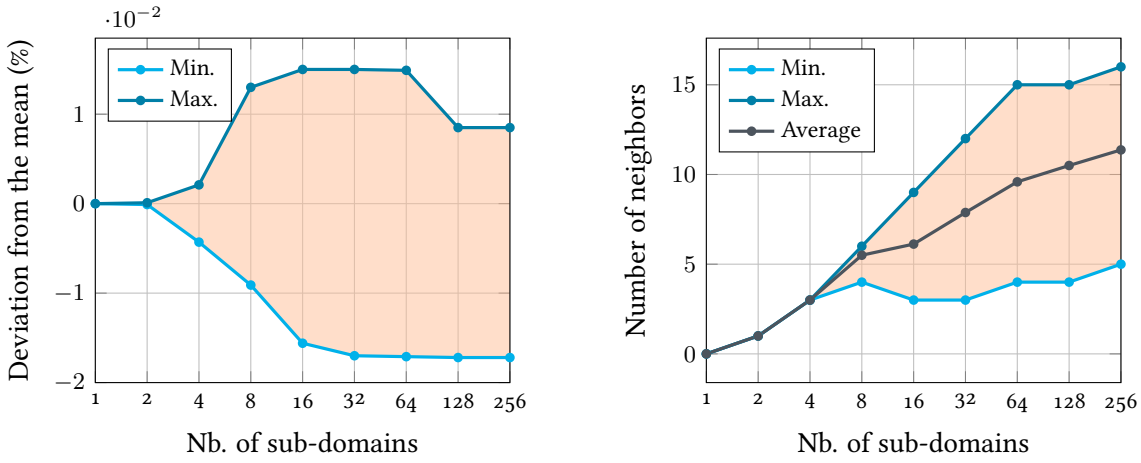


Figure 7.4 | Sub-domains of a Metis-partitioned mesh.



(a) Deviation from the mean number of cells per sub-domain

(b) Minimal, maximal and average number of neighbors per sub-domain

Figure 7.5 | Statistics of Metis-partitioned meshes, for 1 to 256 sub-domains.

1 to 256 cores. The sudden slope variation at 16 cores indicates that data is not exchanged the same way whether the cores are on the same blade or not. In fact, within a blade, point-to-point data exchange is simulated on a shared memory architecture, inducing an inevitable overhead, and dramatically impacting the scalability. Between blades, a true point-to-point communication pattern is used. Hence, in the following, the 16 cores time is taken as a reference for scalability tests.

Scaling measurements

The measured speedup and efficiency for 16 to 256 subdomains are plotted in figure 7.7, for polynomial orders from 1 to 4. Results are acceptable (efficiency > 0.9) up to 128 cores. The drop in efficiency for 256 cores cannot be attributed to the partitionning, since the partitions are very well balanced, and we observe no sudden rise in the number of neighbors per subdomain (see figure 7.5). However, at 256 cores, the number of cells per sub-domain becomes particularly low (around 15560 for 64 cores, 7780 for 128 cores, and 3890 for 256 cores). The time spent in communications becomes too large compared to that spent actually computing the DG algorithm, leading to the observed drop in efficiency. This is easily illustrated by plotting the ratio of average time spent in the MPI patterns against the average time spent in actual computations, which is done in figure 7.8. Apart from $\mathbb{P}_2$, which displays a slightly different behavior, one can observe a dramatic rise of the ratio MPI/CPU time. For 256 cores, it ranges from 0.56 for $\mathbb{P}_4$ to almost 1 for $\mathbb{P}_1$.

7.2.3 Parallel balance

In the case of a parallel computation, each CPU has to share its time between actual computations (hereafter noted CPU time), and communications *via* the MPI communication pattern (MPI time). Ideally, MPI times should be equal for all the CPUs used in the computation. However, because of the unbalance of the partition and the choices made in the parallel implementation, increasing discrepancies appear in the computational loads with the number of sub-domains. In the case of the cavity mesh, the total number of tetrahedra per core ranges from almost 1,000,000 (for 1 core) to less than 4,000 (for 256 cores). Hence, important load discrepancies are to be expected for high core numbers. To illustrate this point,

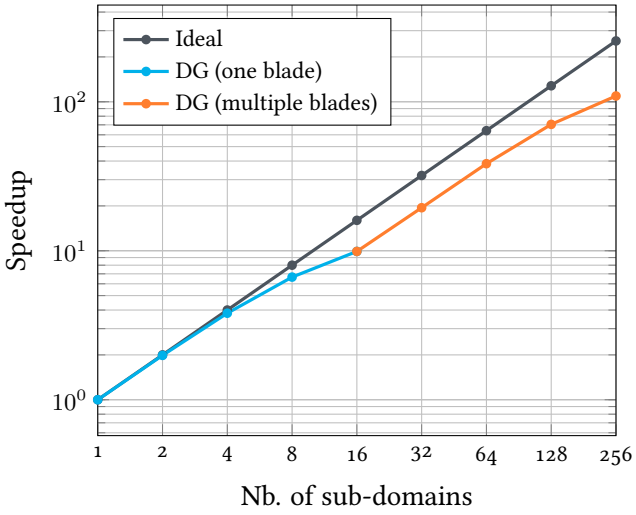
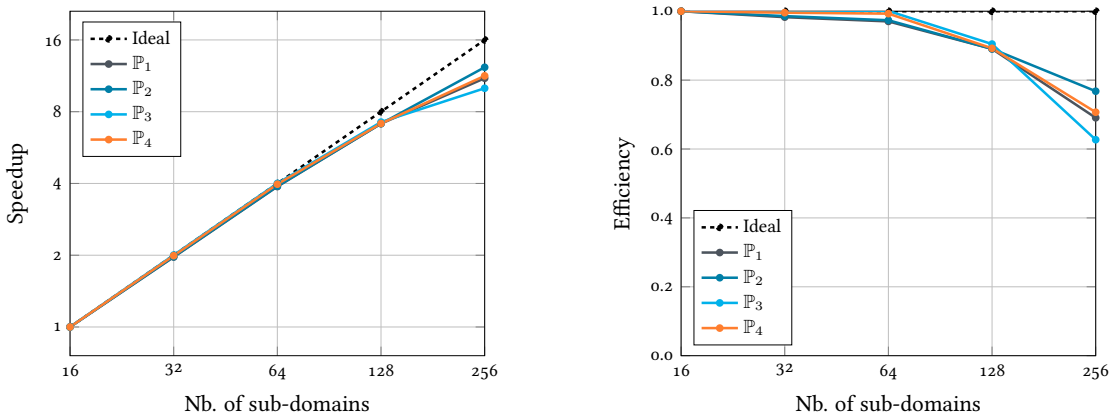


Figure 7.6 | Variable data exchange procedures between cores depending on wether they are on the same blade, or on a different blade.



(a) Speedup

(b) Efficiency

Figure 7.7 | MPI speedup and efficiency for the PEC cavity case on mesh M1, for 16 to 256 subdomains.

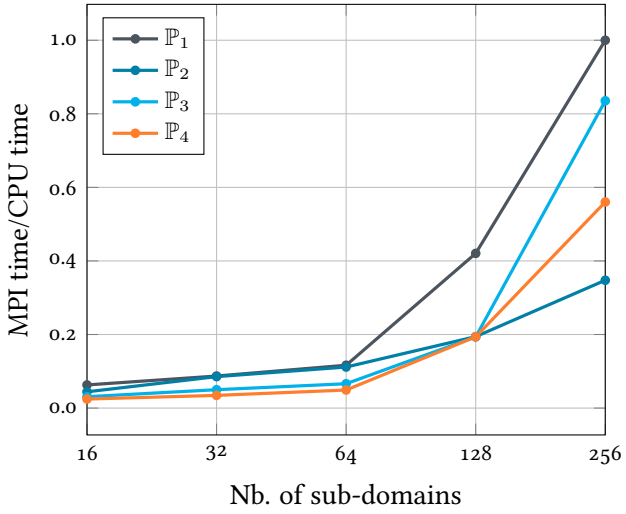


Figure 7.8 | MPI time over CPU time ratio, for 16 to 256 subdomains.

we plot on figure 7.9 the evolution of the maximal CPU time deviation to the mean (noted $\overline{\Delta}_{\max}$) over the subdomains, for increasing number of cores and with polynomial orders ranging from 1 to 4.

7.2.4 Conclusion

As a conclusion, we can state that the proposed MPI implementation presents interesting performances on partitions with more than 10,000 cells per subdomain, and is suitable for small computation workstations (a few tens of cores) to medium-sized clusters (a few hundreds of cores). One of the possible reasons for the collapse of the speedup curve above 128 cores is the rising discrepancy between the minimal and maximal number of neighbours per subdomain, with a ratio superior to 3 for 256 cores. A first step to enhanced MPI performances could be to overlap communications with the DG computations of the inner cells of each subdomain.

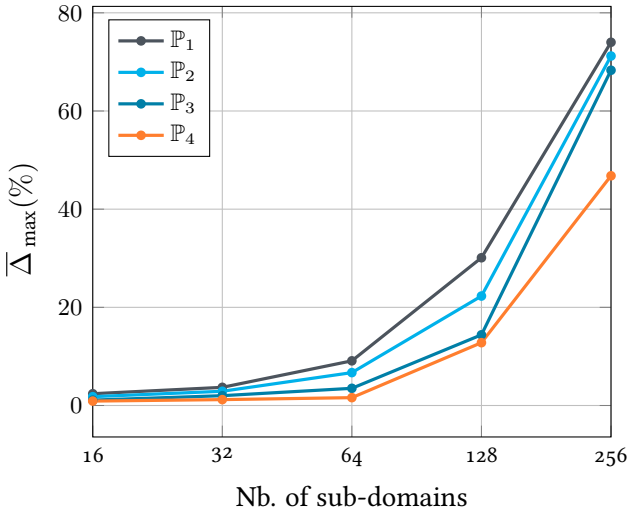


Figure 7.9 | CPU time deviation from the mean for 1 to 256 subdomains, with polynomial orders ranging from 1 to 4.

8

Realistic nano-optics computations

In this chapter, three practical nano-optics configurations are simulated with our DGTD solver. First, the EELS spectrum of a metallic nanosphere is computed, based on a procedure described in [MNHB11]. Then, the behavior of metallic nanocubes on a gold slab is investigated. This work is part of a collaboration with A. Moreau [MCS13]. Finally, several dielectric reflectarray configurations are considered, as part of a collaboration with M. Klemm [ZLGW+14].

These cases required various numerical features and post-treatments that were introduced previously in this manuscript. The meshes were produced with `gmsh` [GR09], which proved very efficient at producing complex curvilinear meshes. 3D visualizations were made with Vizir[1].

8.1 Electron energy loss spectroscopy

8.1.1 Introduction

Popularized in the 1990's, electron energy loss spectroscopy (EELS) consists in using a beam of fast-moving electrons which energy is known, to scan a device and/or a material. The non-zero probability of each electron to interact with the structure produces a measurable energy loss in the beam, thus providing informations on the structure. In particular, various plasmonic resonances can be investigated when the electron beam passes close to the sample (this method is usually known as low-loss EELS).

Numerical treatments of such problems have already been proposed with standard methods such as FDTD [CMLN15] or BEM [HT12]. Here, we follow the procedure proposed for DGTD in [MNHB11]: an electron travels at speed v along a trajectory $\mathbf{r}_e(t)$ colinear to the z axis (*i.e.* $\mathbf{r}_e(t) = \mathbf{r}_0 + vt\,\mathbf{e}_z$). The field generated in vacuum by the moving electron is given by [Fey10]:

[1] https://www.rocq.inria.fr/gamma/gamma/vizir/

$$\begin{aligned}\mathbf{E}(\mathbf{r},t) &= \frac{q}{4\pi\varepsilon_0}\frac{\gamma\mathbf{d}(\mathbf{r},t)}{\left(|\mathbf{d}(\mathbf{r},t)|^2+\frac{\gamma^2-1}{v^2}\left(\mathbf{v}\cdot\mathbf{d}(\mathbf{r},t)\right)^2\right)^{\frac{3}{2}}},\\ \mathbf{H}(\mathbf{r},t) &= \frac{\mathbf{v}}{c}\times\mathbf{E}(\mathbf{r},t),\end{aligned} \tag{8.1}$$

with $\gamma = \sqrt{\frac{1}{1-\left(\frac{v}{c}\right)^2}}$ and $\mathbf{d}(\mathbf{r},t) = \mathbf{r}_e(t) - \mathbf{r}$. The electron's trajectory brushes past an aluminum nanosphere with a minimal distance[2] b, which is typically of a few nanometers. In return, the scattered field radiated by the excited plasmons acts back on the electron, slightly lowering its kinetic energy. As described in [MNHB11], these losses are extremely low compared to the total energy of the electron (at least a thousand times lower). Hence, a rather good approximation, known as no-recoil approximation, consists in neglecting the induced slow-down in the loss computation. The classical way of expressing the energy lost by the electron is to express it as a frequency-dependent loss probability $P(\omega)$, which represents the probability of an electron to lose an energy equal to $\hbar\omega$:

$$P(\omega) = \frac{1}{\pi\hbar\omega}\int_{-\infty}^{\infty}\Re\left(\mathbf{v}\cdot\mathbf{E}_{\text{sca}}(\mathbf{r}_e,\omega)\,\mathrm{e}^{-i\omega t}\right)\,dt. \tag{8.2}$$

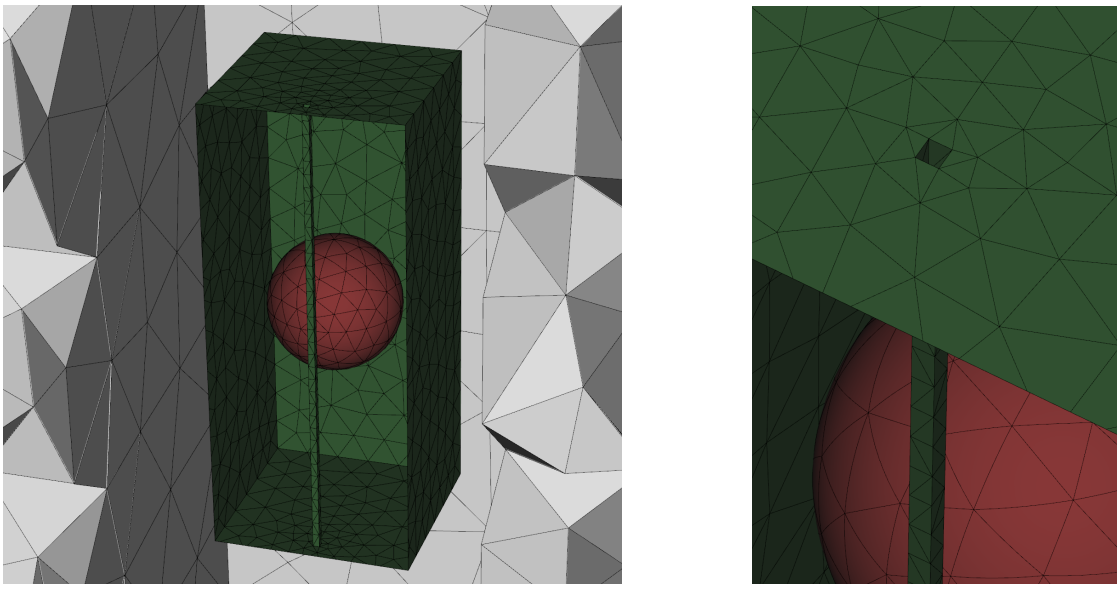
One might notice that the incident field (8.1) is singular at the electron location (*i.e.* for $\mathbf{r} = \mathbf{r}_e$). To avoid this particular problem in practice, this incident field is imposed at a certain distance from the electron's trajectory, on a TF/SF interface, in the same fashion as for dipolar sources (see section 4.2.1). To do so, a cylindrical surface of sufficient length enclosing electron's trajectory is defined in the computational domain. To avoid the singular field at the top and bottom edges of the cylinder, the TF/SF surface is closed inside-out, so the TF region is not simply connex (see mesh on figure 8.1). This technique is only valid if the electron beam does not travel throught the material. In this latter case, using the TF/SF interface method can only lead to approximate results, since a portion of the scatterer is excluded from the total field region. To overcome this limitation, one can use a fully-scattered formulation of Maxwell's equations, which gives access to the scattered field inside the scatterer [Die12].

8.1.2 EELS spectrum of an aluminium nanosphere

As a first computational test, the EELS spectrum of an aluminium nanosphere of radius 10 nm is computed using the method described above (this configuration is directly derived from [MNHB11]). Aluminium is described by a Drude model of parameters $\varepsilon_\infty = 1$, $\omega_d = 2.278\times10^7$ GHz and $\gamma_d = 1.5952\times10^6$ GHz. The impact parameter is 1 nm, and the cylinder enclosing the electron's path ranges from 35 nm above the sphere center to 35 nm below. To compute a discrete version of (8.2), the electron's trajectory is discretized with N probe points evenly spaced every $\Delta z = 0.5$ nm, between a starting and a finishing altitudes z_s and z_f (here, $z_s = -35$ nm, and $z_f = 35$ nm). At these probe points, the discrete Fourier transform (DFT) of E_z is computed along the simulation. To avoid spectral leakage in the DFT, it is necessary that it starts and ends with near-zero fields. Hence, a retardation factor t_0 is added to the incident field (8.1), and the trajectory is described by $\mathbf{r}_e(t) = \mathbf{r}_0 + v(t-t_0)\,\mathbf{e}_z$. At the end of the computation, the loss probability is evaluated following a simple rectangles rule for each frequency of interest:

$$P(\omega) \simeq \frac{1}{\pi\hbar\omega}\sum_{k=1,N} v\,\Re\left(E_{z,sca}(\mathbf{r}_e(t_k),\omega)\,\mathrm{e}^{-i\omega t_k}\right)\,\Delta t,$$

[2] This minimal distance is hereafter called impact parameter.

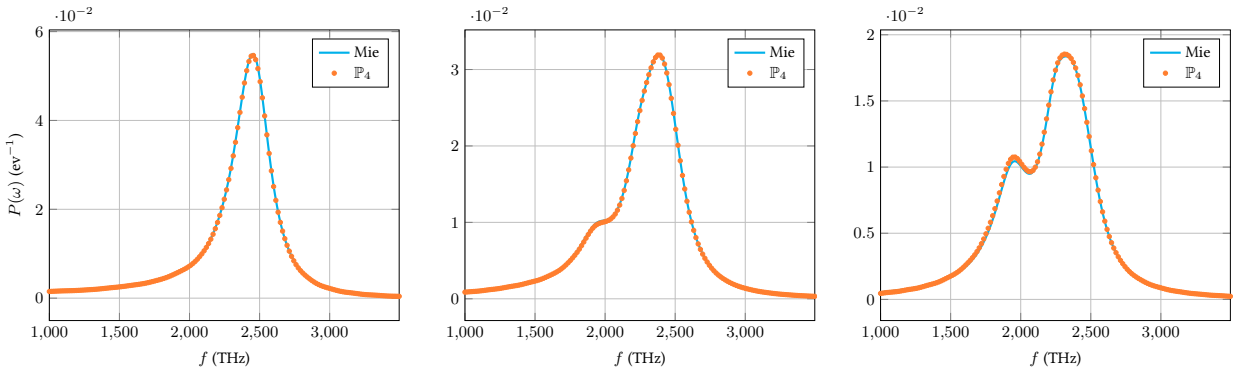


(a) Global view of the mesh

(b) Top TF/SF interface

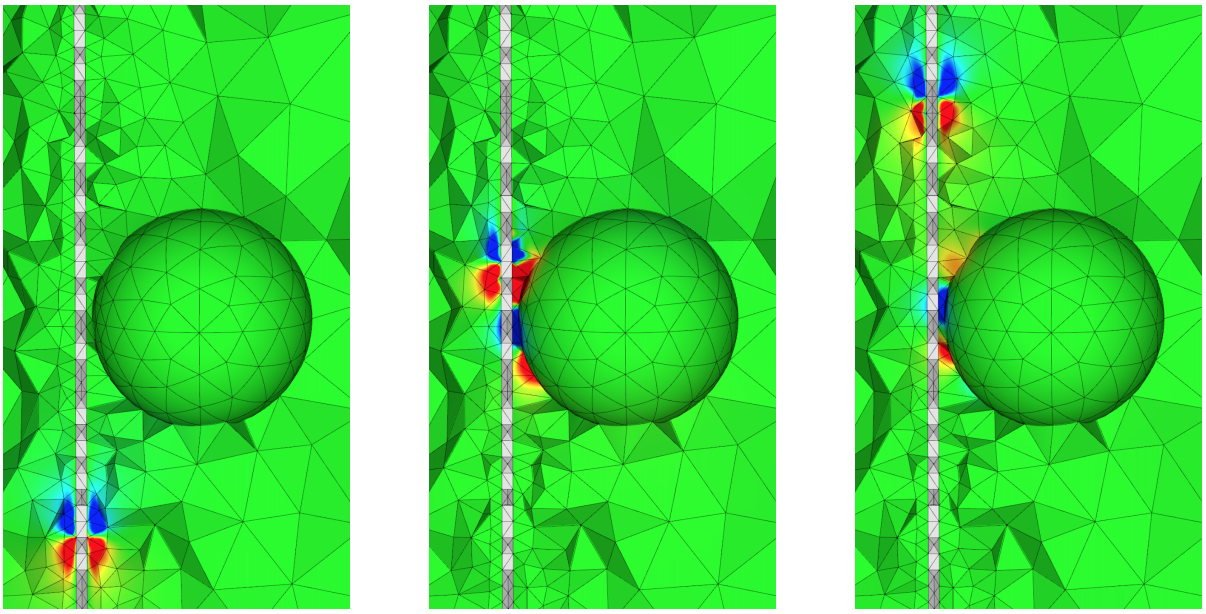
Figure 8.1 | Mesh setup for a metallic sphere EELS spectrum computation. The gray cells correspond to the PML, and the red ones to the metallic sphere. The green triangles define the TF/SF interface, which is closed inside-out thanks to a cylinder connecting the upper and lower faces. The z extension of the TF/SF box is voluntarily reduced for clarity.

where $t_k = \frac{(k-1)\Delta z}{v} + t_0$ and $\Delta t = \frac{\Delta z}{v}$. The mesh used for the computation is presented on figure 8.1: for better accuracy, quadratic tetrahedra are used to discretize the sphere. We observed no significant difference between the use of square-section and circular-section cylinders for the surface enclosing the electron's trajectory. In all the following computations, $\mathbb{P}_4$ polynomials were retained to obtain a fully satisfying resolution of the EELS spectrum. On figure 8.2, the spectra computed with the DGTD implementation are compared to the Mie solution of the problem [Aba10] for $v = 0.1$, 0.2 and 0.3 times the speed of light. As an illustration, field maps of the E_z component are presented on figure 8.3. Computational times on 16 cores range from 2 hours 45 minutes (for $v = 0.3$) to 3 hours (for $v = 0.1$).



(a) $v = 0.1\,c_0$ (b) $v = 0.2\,c_0$ (c) $v = 0.3\,c_0$

Figure 8.2 | EELS spectrum of a single aluminium nanosphere for various electron velocities. $\mathbb{P}_4$ approximation is used in conjunction with curvilinear elements for the sphere.



(a) The electron's trajectory is oriented from bottom to top

(b) The field induced by the electron in the vicinity of the sphere excites a surface plasmon

(c) While the electron moves away, the plasmon continues to resonate

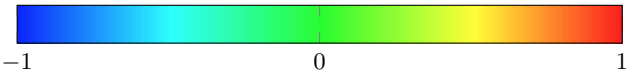


Figure 8.3 | E_z field map during an EELS experiment. The gray cells correspond to the SF cells, in which the field is not represented. In this case the electron velocity is $v = 0.2\,c_0$. For the three views, the field values are arbitrarily scaled to $[-1, 1]$.

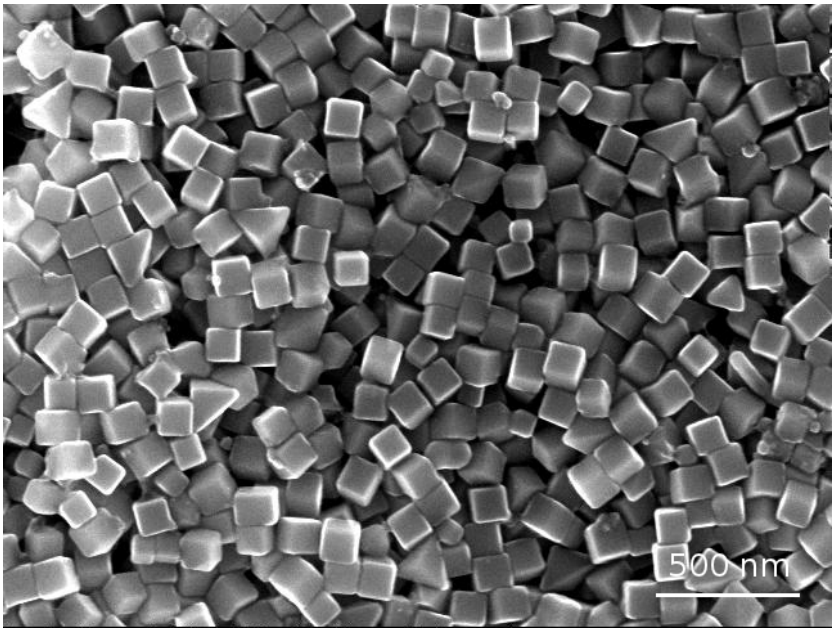


Figure 8.4 | Random arrangement of chemically-produced nanocubes. Courtesy of A. Moreau.

8.2 Gap-plasmon confinement with gold nanocubes

The propagation of light in a slit between metals is known to give rise to guided modes. When the slit is of nanometric size, plasmonic effects must be taken into account, since most of the mode propagates inside the metal. Indeed, light experiences an important slowing-down in the slit, the resulting mode being called gap-plasmon. Hence, a metallic structure presenting a nanometric slit can act as a light trap, *i.e.* light will accumulate in a reduced space and lead to very intense, localized fields. Recently, the chemical production of random arrangements of nanocubes on gold films at low cost was proved possible by Moreau *et al* . [MCM+12] (see figure 8.4). As shown on figure 8.5, nanocubes are separated from the gold substrate by a dielectric spacer of variable thickness, thus forming a narrow slit under the cube. When excited from above, this configuration is able to support gap-plasmon modes which, once trapped, will keep bouncing back and forth inside the cavity. At visible frequencies, the lossy behavior of metals will cause the progressive absorption of the trapped electromagnetic field, turning the metallic nanocubes into efficient absorbers. The frequencies at which this absorption occurs can be tuned by adjusting the dimensions of the nanocube and the spacer. Here, we propose to study the impact of the geometric parameters of the problem on the behaviour of a single nanocube placed over a metallic slab. This work constitutes the base of a wider study in collaboration with A. Moreau.

8.2.1 Physical parameters and quantities

In the following study, both the slab and the cube are made of gold (hence $\varepsilon_c = \varepsilon_g$). A 3SOGP model (see section 2.2.3) is used to fit data from [RDEM98] in the $[200, 750]$ THz frequency range. The dispersion parameters, as well as plots of the real and imaginary parts of the permittivity function can be found in appendix C. In all the computations, the thickness of the metallic slab h is taken equal to 75 nm. The spacer is made of a dielectric of constant permittivity $\varepsilon_s = 2.1316$ corresponding to silica. The rounding of the cube edges and corners is denoted by r, and its effect is studied in the next section. The last section

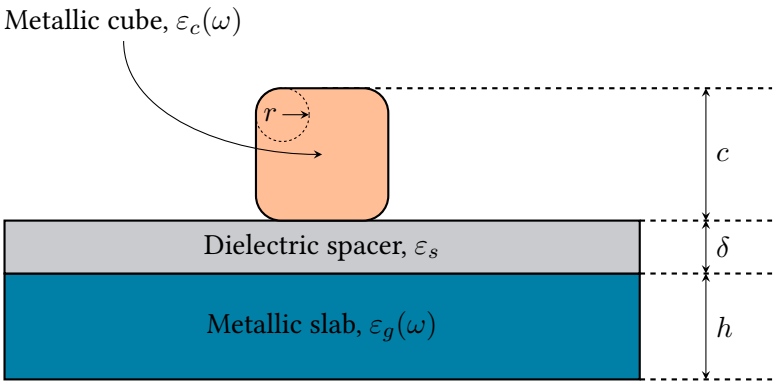


Figure 8.5 | Realistic metallic nanocubes on a dielectric-coated gold slab. The rounding parameter r is the same for all corners and edges of the cube. The view is a lateral cut of the device.

is dedicated to the impact of the two other geometrical parameters, *i.e.* the spacer thickness δ and the nanocube side length c.

Various physical quantities will be considered: (i) the absorption cross-section and efficiency for the whole device (nanocube, spacer and full metallic slab), respectively noted C_{abs} and Q_{abs}, (ii) the absorption efficiency in the cube only, noted Q_{cube}, and (iii) the difference between the last two. The computation of the absorption efficiency is described in section 4.4.1. To compute Q_{cube}, a volumic method is used (see section 4.4.2).

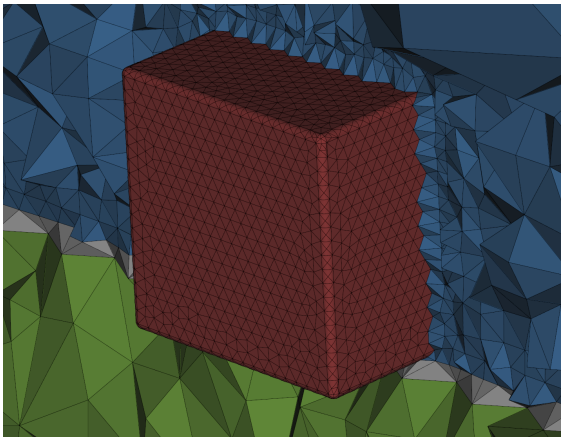
8.2.2 Influence of the rounding

Experimentally, chemically-produced nanocubes present a rounding at the edges and corners, which size ranges from 3 to 10 nm. In this section, the size parameters are set to $\delta = 5$ nm and $c = 75$ nm, while the rounding r is progressively increased from 0 to 10 nm. For each value of r, the absorption cross-section C_{abs} is computed. Mesh examples are presented on figure 8.6, while results are shown on figure 8.7. For frequencies above 450 THz, results are very similar, since the absorption is due to bound electrons of the gold plate (*i.e.* the absorption is not related to the gap plasmon phenomenon). Below 450 THz, one can observe a transition of the gap plasmon absorption peak for increasing values of r, from 325 THz for $r = 0$, to 375 THz for $r = 10$ nm. To obtain a converged solution on such small geometrical details, $\mathbb{P}_3$ polynomial approximation is used in conjunction with curved elements. In the following of this study, we keep a $r = 3$ nm rounding in all computations, which corresponds to the best reproducible chemical production to date.

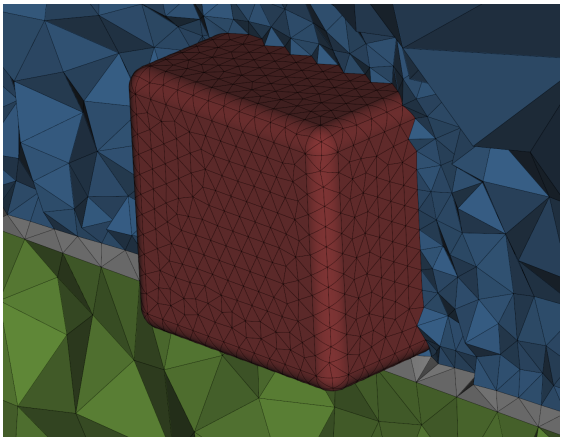
8.2.3 Absorption and scattering regimes

In this section, the behavior of single nanocubes on metallic plates is computed, for lateral sizes ranging from 50 to 80 nm, and spacer thicknesses from 3 to 22 nm. In each case, the resonance frequency is obtained by seeking the maximal value of Q_{cube} over the frequency range. Then, the absorption efficiencies Q_{abs} and Q_{cube} at the resonance frequency are retrieved from the results of each computation. All these results are summed up on figure 8.8. Several remarks can be made from the analysis of these curves:

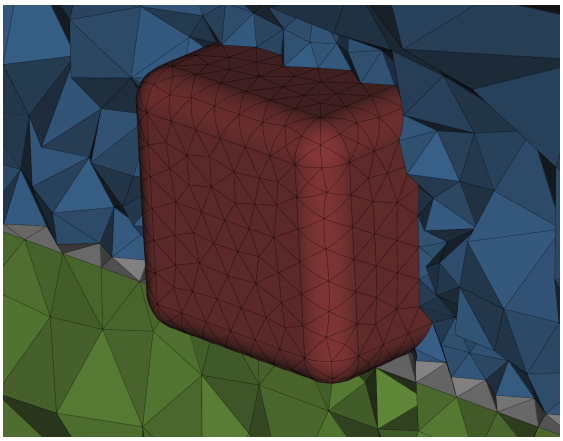
- The absorption in the cube due to the gap-plasmon varies with the size of the cube, and absorption efficiencies as high as 18 are observed for $c = 70$ nm and $\delta = 12$ nm (resonance frequency $f_{\text{res}} = 432$



(a) $r = 2$ nm



(b) $r = 6$ nm



(c) $r = 10$ nm

Figure 8.6 | Meshes of rounded nanocubes with rounding radii ranging from 2 to 10 nm. Red cells correspond to the cube. The latter lies on the dielectric spacer (gray cells) and the metallic plate (green). Blue cells represent the air surrounding the device.

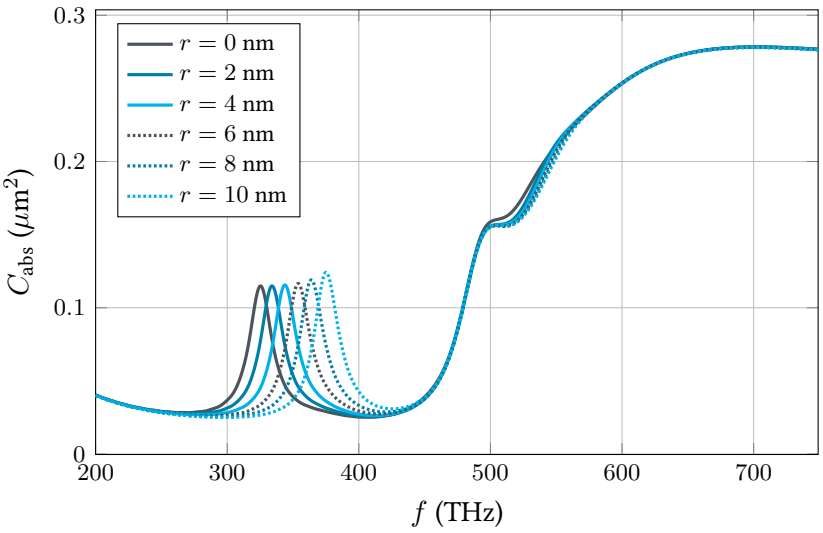


Figure 8.7 | Absorption cross-section of the nanocube device for various edge roundings. Larger roundings occasionate a blueshift of the absorption peak. For these results, $\delta = 5$ nm and $c = 75$ nm.

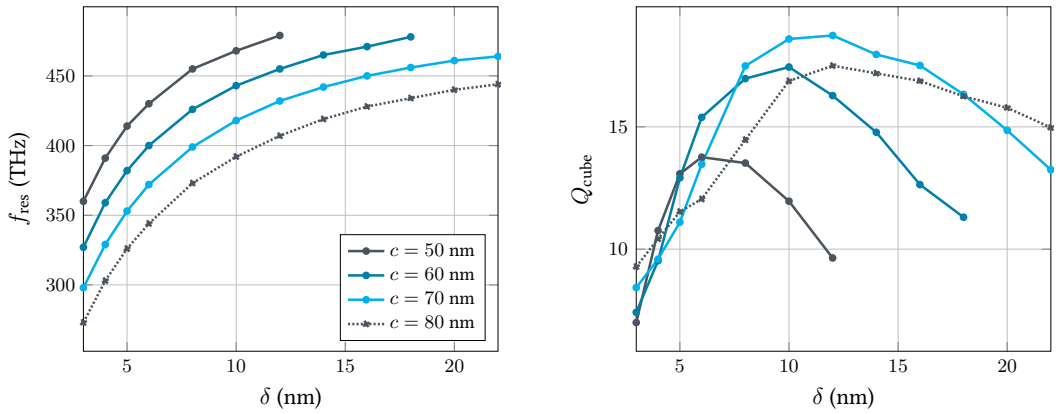
THz). Because the gap plasmon mode is roughly symmetric (see figure 8.9), gap-plasmon absorption efficiencies superior to 30 can be expected, which is in accordance with [MCM+12]. Additionally, there seems to be an optimal set of parameters for $Q_{\text{cube}} = f\,(c, \delta)$;

- By substracting Q_{cube} to Q_{abs}, we obtain the absorption due to (i) the skin effect of the direct illumination of gold, (ii) the absorption in the slab due to the gap plasmon, and (iii) the surface plasmons generated in the slit between the cube and the gold plate. As can be seen on figure 8.8(d), the absorption continues to grow when δ is increased above the Q_{cube} maximum. From that observation, we can state that, when progressively increasing δ from 0, isolated nanocubes exhibit two different behaviors: (i) an absorption regime for low δ values (typically between 5 and 10 nm), where a large part of the absorption occurs in the gap, and (ii) a scattering regime for higher δ values, where most of the energy is transfered to plasmons propagating on the metallic slab, and upward scattering;
- Putting aside figures 8.8(b) and 8.8(d), it seems that small nanocubes are good at generating surface plasmons, but experience less intense absorption due to the gap resonance than their larger counterparts. Large nanocubes, on the contrary, present better absorption rates due to the gap resonance, but are poor surface plasmon producers.

To further illustrate these two regimes, we plot on figure 8.9 the modulus of the $\widehat{\mathbf{H}}$ field, at resonance frequency for the ($c = 70$ nm, $\delta = 12$ nm) and the ($c = 60$ nm, $\delta = 18$ nm) configurations. The obtained $\widehat{\mathbf{H}}$ field is more intense for configurations that yield high Q_{cube} values, which is coherent with [MCM+12]. For the ($c = 70$ nm, $\delta = 12$ nm) case, the gap resonance seems symmetric. However, it is not the case for the ($c = 60$ nm, $\delta = 18$ nm) configuration, where more field is concentrated in the cube than in the metallic slab.

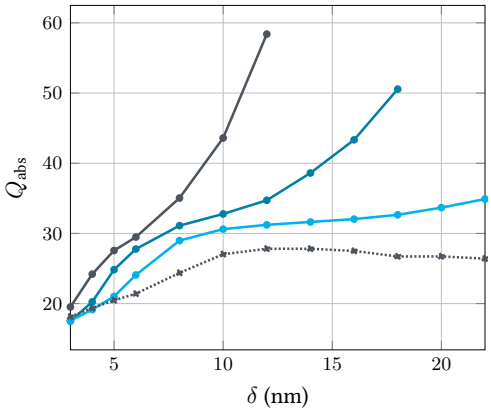
8.2.4 Numerical discussion

All the computations in this study were performed on 8-cores Intel E5-2670 2.6 GHz, with $\mathbb{P}_3$ polynomials. The very small rounding considered for the cube edges induced a dramatic reduction of the minimal edge

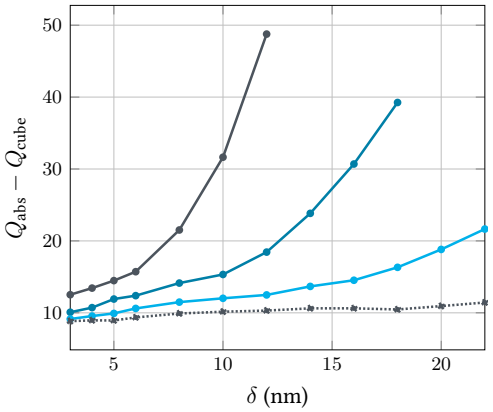


(a) Resonance frequency f_{res}

(b) Cube absorption efficiency Q_{cube}

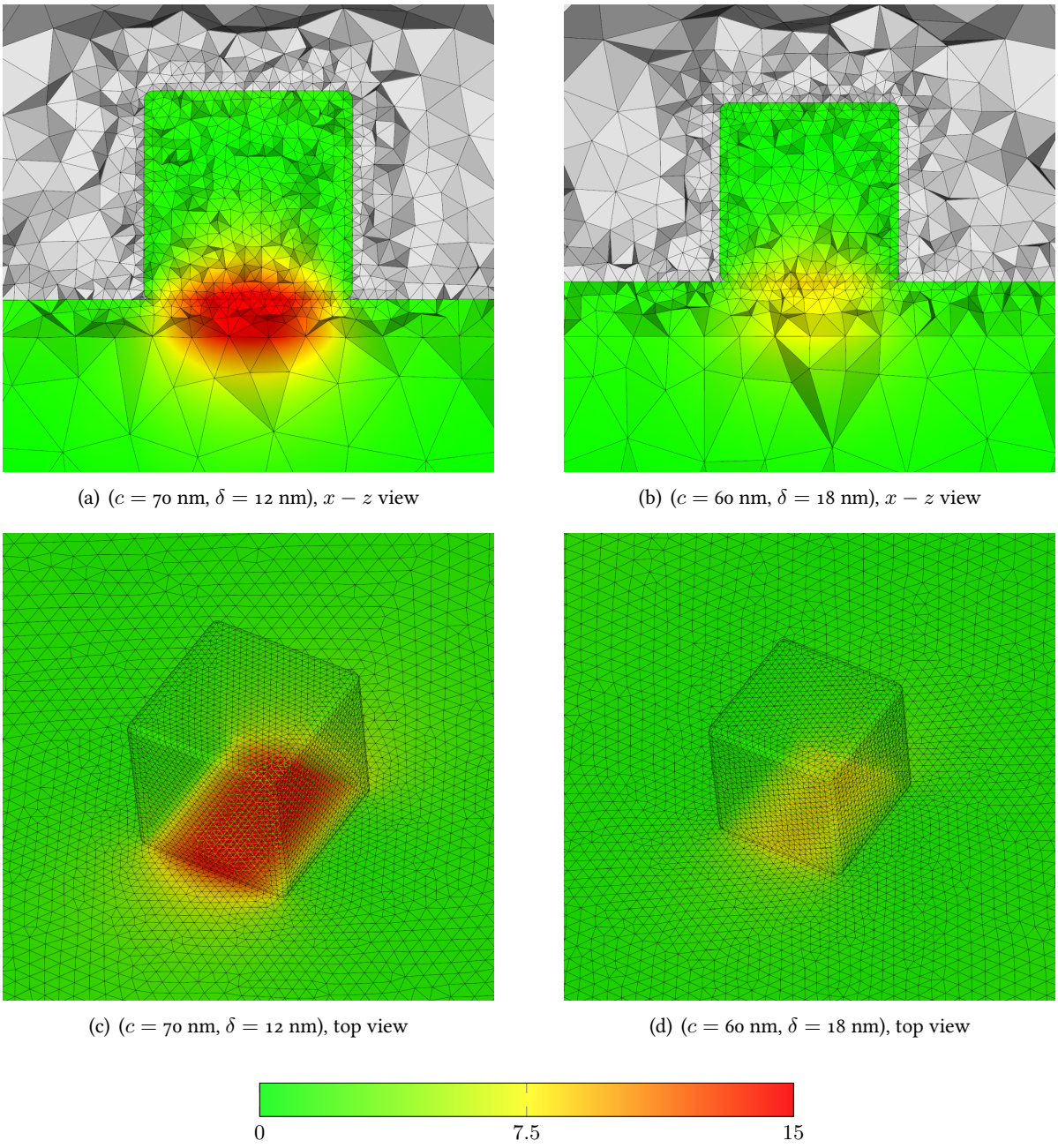


(c) Total absorption efficiency Q_{abs}



(d) Difference $Q_{\text{abs}} - Q_{\text{cube}}$

Figure 8.8 | Resonance frequencies and absorption efficiencies of gold nanocubes for various nanocube sizes and spacer thicknesses.



(a) ($c = 70$ nm, $\delta = 12$ nm), $x - z$ view

(b) ($c = 60$ nm, $\delta = 18$ nm), $x - z$ view

(c) ($c = 70$ nm, $\delta = 12$ nm), top view

(d) ($c = 60$ nm, $\delta = 18$ nm), top view

Figure 8.9 | $\widehat{\mathbf{H}}$ field modulus for different nanocube configurations. All field maps are scaled identically for better comparison. The obtained $\widehat{\mathbf{H}}$ field is more intense for configurations that yield high Q_{cube} values. Although the gap resonance seems almost symmetric on the top left panel, it is not the case on the top right one.

length, as well as an important increase of the number of curved cells. Computational times on 32 cores range from 4 hours 10 minutes for ($r = 0$nm, $c = 75$ nm, $\delta = 5$ nm) to 17 hours 15 minutes for ($r = 3$nm, $c = 90$ nm, $\delta = 3$ nm). Additionally, due to the multiple on-the-fly Fourier transforms on the TF/SF surface and in the cube volume, as well as the presence of PMLs and curved elements, parallel imbalance range from 10 % to more than 60 %, which highlights the need to take these features into account when partitioning the mesh. It should be noted that the computational domain could be made smaller by sticking the extremities of the metallic slab directly into the PML layer. Indeed, the CFS-PML naturally allows dispersive materials to extend into it just by modifying the permittivity accordingly.

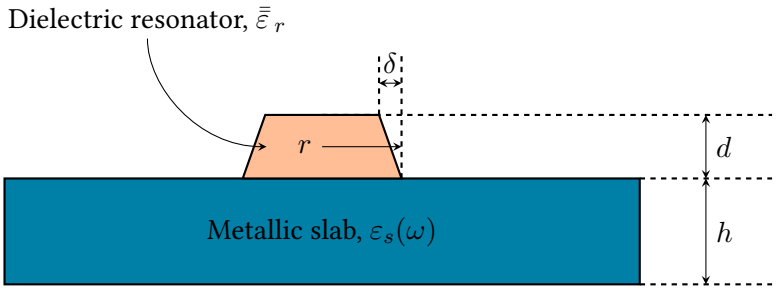


Figure 8.10 | Unit cell of a realistic monodimensional dielectric reflectarray composed of dielectric cylinders on a silver plate. The defect parameter δ is equal to zero for an ideal resonator. The view is a lateral cut of the cell.

8.3 Dielectric reflectarrays

In the past few years, important efforts have been deployed to find alternatives to on-chip, low-performance metal interconnects between devices. Because of the ever-increasing density of integrated components, intra- and inter-chip data communications have become a major bottleneck in the improvement of information processing. Given the compactness and the simple implantation of the devices, communications *via* free-space optics between nanoantenna-based arrays have recently drawn more attention [HE08]. Here, we focus on a specific low-loss design of dielectric reflectarray (DRA), whose geometry is based on a periodic repartition of dielectric cylinders on a metallic plate [ZWS+13]. A sketch of the unit cell is presented on figure 8.10. When illuminated in normal incidence, specific patterns of such resonators provide a constant phase gradient along the dielectric/metal interface, thus altering the phase of the incident wavefront. The gradient of phase shift generates an effective wavevector along the interface, which is able to deflect light from specular reflection. However, as can be seen on figure 8.11, the flaws of the lithographic production process can lead to discrepancies between the ideal device and the actual resonator array.

Here, we propose to exploit our DGTD solver to study the impact of the lithographic flaws on the performance of a 1D reflectarray. Efficient computations are obtained by combining high-order polynomial approximation with curvilinear meshing of the resonators, yielding accurate results on very coarse meshes. The study is continued with the computation of the reflection of a 2D reflectarray. This work constitutes the base of a wider study in collaboration with M. Klemm [ZLGW+14].

8.3.1 Physical parameters and quantities

In the following sections, the silver slab is described by a simple Drude model of parameters $\varepsilon_\infty = 4.0$, $\gamma_d = 2.73 \times 10^4$ GHz and $\omega_d = 1.38 \times 10^7$ GHz. The resonators are made of a diagonally anisotropic material of parameters $\bar{\bar{\varepsilon}}_r = \text{diag}\,[8.29, 8.29, 6.71]$. The slab thickness h, as well as the height d are respectively fixed to 200 and 50 nm. The defect parameter is denoted δ, and describes the impact of the lithography flaws on the cylindrical shape of the resonator. The last geometric parameter is the basis radius of the resonator, denoted by r. In all computations, the devices are terminated with periodic boundary conditions in both planar directions. The incident field is a monochromatic plane wave, impinging from above in normal incidence.

The physical quantities of interest in this work are: (i) the reflection coefficient R, (ii) the reflected phase θ for single resonators, and (iii) the radar cross-section σ_{RCS} for the resonator arrays. The first

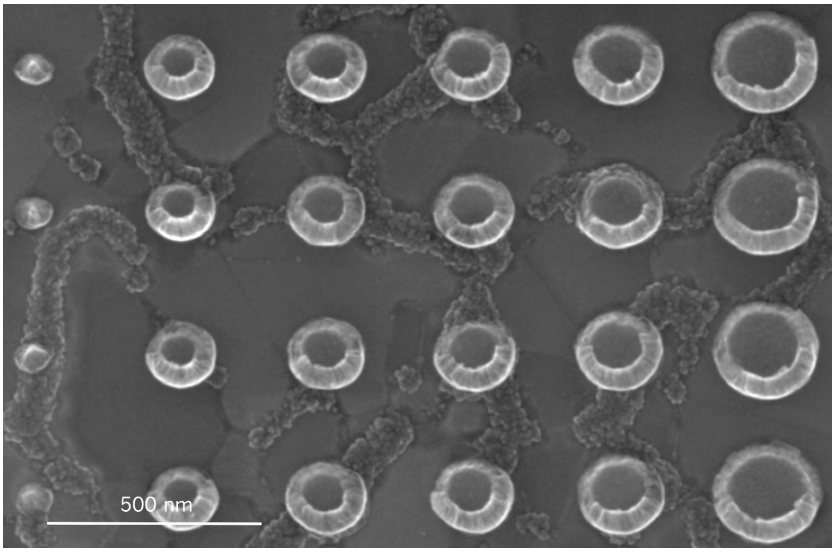


Figure 8.11 | 6-element dielectric reflectarray produced by lithography. Courtesy of M. Klemm.

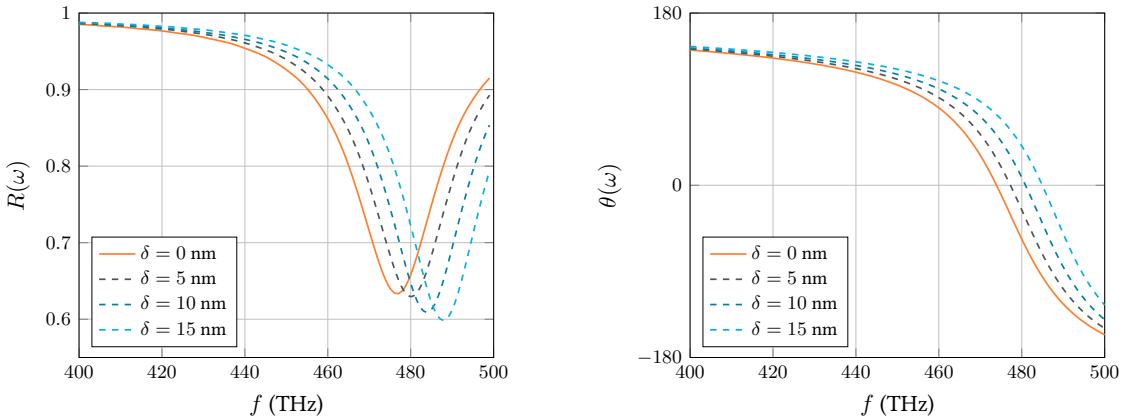
one is computed following the method presented in section 4.4.3, while the second one is obtained by computing the phase of the scattered field above the center of a single resonator. In the last section of the study, the radar cross-section (RCS) of a 1D dielectric reflectarray is computed as described in section 4.4.4.

8.3.2 Influence of lithography defects

We propose here to study the effects of the flaws induced by the lithography production of the dielectric resonators on its scattering regime. A single resonator with doubly periodic boundary conditions is considered. The lateral size of the periodic cell is 350 nm, the radius is fixed to $r = 85$ nm, and δ varies from 0 to 15 nm. The frequency of the incident plane wave is fixed to $f = 473.6$ THz ($\lambda = 633$ nm). The reflection coefficient and the reflected phase are computed, and plotted on figure 8.12. As can be seen, the reflected amplitude and phases are significantly blueshifted when δ is increased, which will have a major impact on the 1D dielectric array, as will be shown in next section.

8.3.3 1D dielectric reflectarray

Here, we consider the 1D dielectric reflectarray presented in [ZLGW+14]. This array is designed to deflect normally-incident light with an angle of 19.9°, according to reflectarray theory. As before, the frequency of the incident plane wave is $f = 473.6$ THz ($\lambda = 633$ nm). The array is declined in two versions: the first one is made of ideal resonators, while the second one is composed of realistic resonators, with representative lithography flaws (see figure 8.13 for a close-up view of the array). The RCS of both arrays is computed with $\mathbb{P}_4$ polynomial approximation and quadratic tetrahedra, and plotted on figure 8.14. The ideal array provides a very good directivity toward 18.0°, with a very small parasitic lobe around 50.0°. This is confirmed by the field map of figure 8.15, where one can clearly see nearly-plane waves propagating away from the array. In this case, nearly 60% of the incident power is deflected with a non-cartesian angle. On the other hand, the realistic array presents more imperfections in its directivity patterns, with numerous parasitic lobes, and a lower efficiency (around 50%). Additionally, the deflection



(a) Reflection coefficient

(b) Reflected phase

Figure 8.12 | Reflection coefficient and reflected phase of a single dielectric resonator with lithography defect.

angle is very different from what was predicted by the reflectarray theory. This results in a much less satisfying field map, where the plane wave is severely distorted. This may enlight the need to compensate these flaws at the conception level by adjusting the physical parameters of the reflectors.

8.3.4 2D dielectric reflectarray

We now consider the 2D reflectarray design presented on figure 8.16. This pattern is obtained by periodically shifting the 1D array of last section while repeating it along the y axis. As before, the reflectarray is illuminated with a plane wave of frequency $f = 473.6$ THz in normal incidence. Its computed RCS is presented on figure 8.17: a clear directivity peak is observed around $(\theta, \phi) = (28.5°, 45°)$, with an efficiency close to 60%. Several parasitic lobes are also present, particularly around the normal direction $(\theta, \phi) = (0°, 0°)$. A 3D time-domain field map is presented on figure 8.18, with the cutting plane oriented in the direction of maximal radiation, *i.e.* $\phi = 45°$.

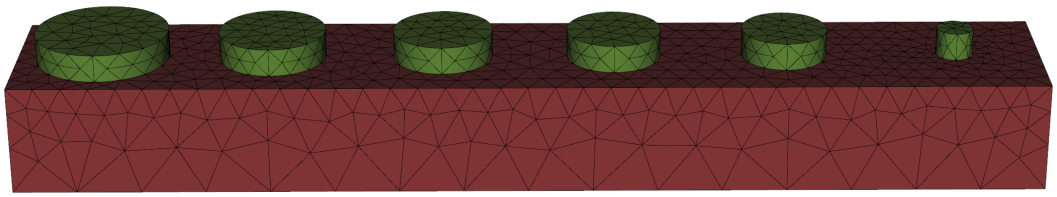
8.3.5 Numerical considerations

In this section, all computations are performed on 8-cores Intel E5-2670 2.6 GHz.

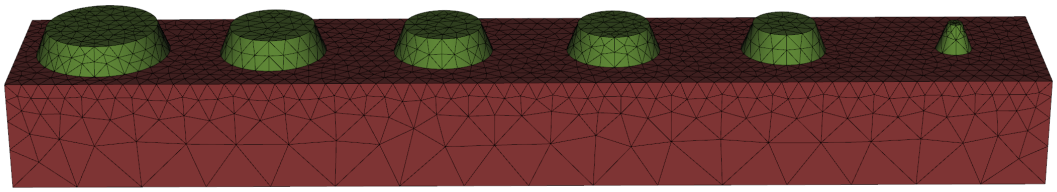
1D reflectarray

The time required to evolve the system from the $t = 0$ to $t = 0.1$ ps on 4 cores is 4 hours 4 minutes, with a maximal CPU imbalance of 8.4 %. On 16 cores, the computational time is 1 hour 24 minutes, with a maximal CPU imbalance of 26.3 %. Here, the low parallel efficiency (0.72 between 4 and 16 cores) has several causes:

- The mesh of the ideal 1D reflectarray is made of 19427 tetrahedra. Hence, on 16 cores, this represents barely more than 1200 tetrahedra per subdomain, which is partcularly low.
- 4233 cells (22 % of total) are curvilinear, while the PML consists of 2558 tetrahedra (13 % of total). These elements induce imbalance in the load allocated to the subdomains, since they are not taken into account during the partitioning. Indeed, when run with linear elements only, CPU imbalances on 4 and 16 cores respectively fall to 6.3 % and 20.4 %.



(a) Ideal reflectarray



(b) Realistic reflectarray, $\delta = 20$ nm

Figure 8.13 | Ideal and realistic 1D dielectric reflectarray meshes. The red tetrahedra correspond to silver, while the green ones are made of an anisotropic dielectric material. The device is surrounded by air and terminated by a PML above and below, and by periodic boundary conditions on the lateral sides.

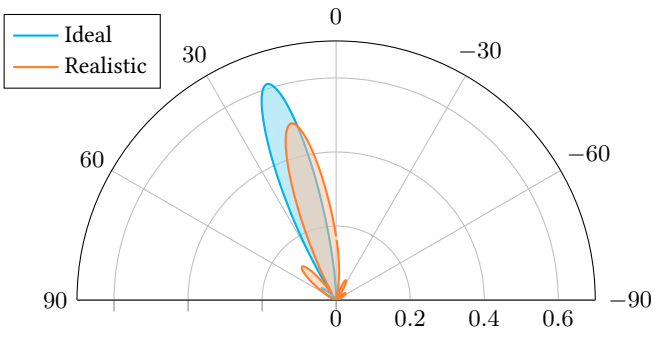
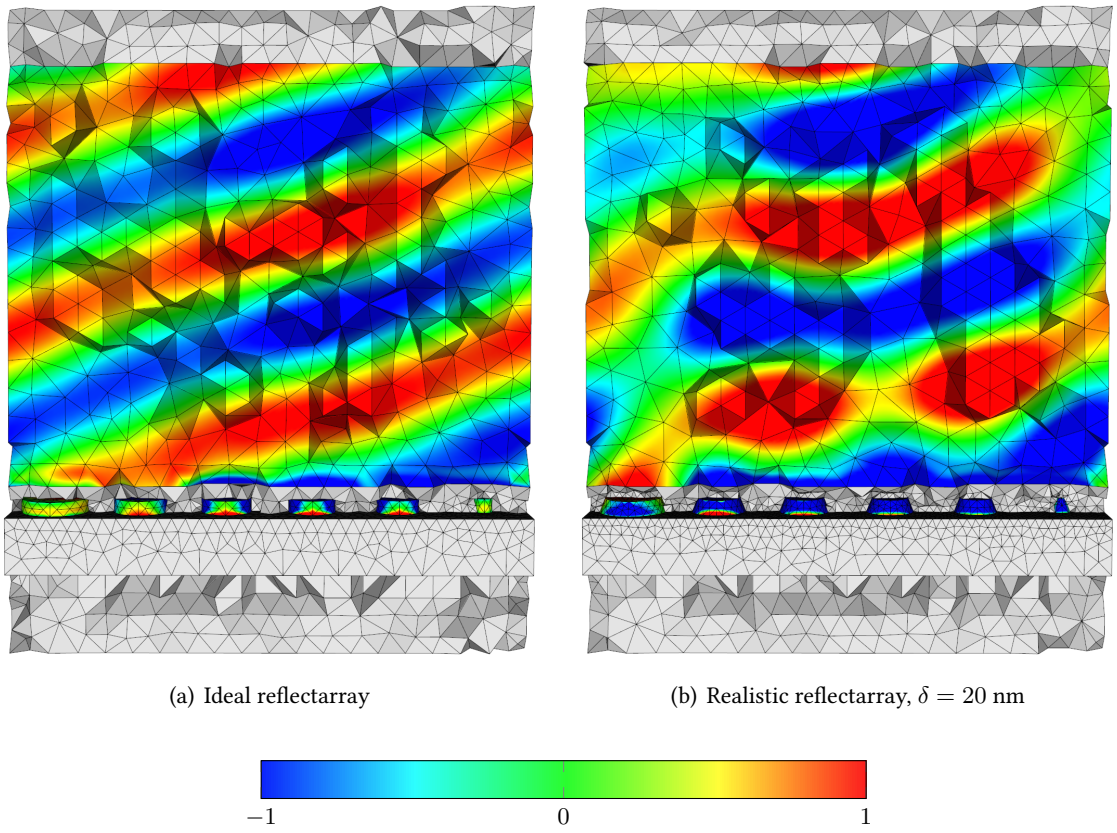


Figure 8.14 | Radar cross-section of ideal and realistic 1D dielectric reflectarrays at frequency f. The directivity peak in the ideal case is observed around 18.0°, while it is obtained at 14.5° for the realistic array.



(a) Ideal reflectarray

(b) Realistic reflectarray, $\delta = 20$ nm

Figure 8.15 | Time-domain snapshot of E_y component for ideal and realistic 1D dielectric reflectarrays. Solution is obtained in established regime at $t = 0.1$ ps. Fields are scaled to $[-1, 1]$.

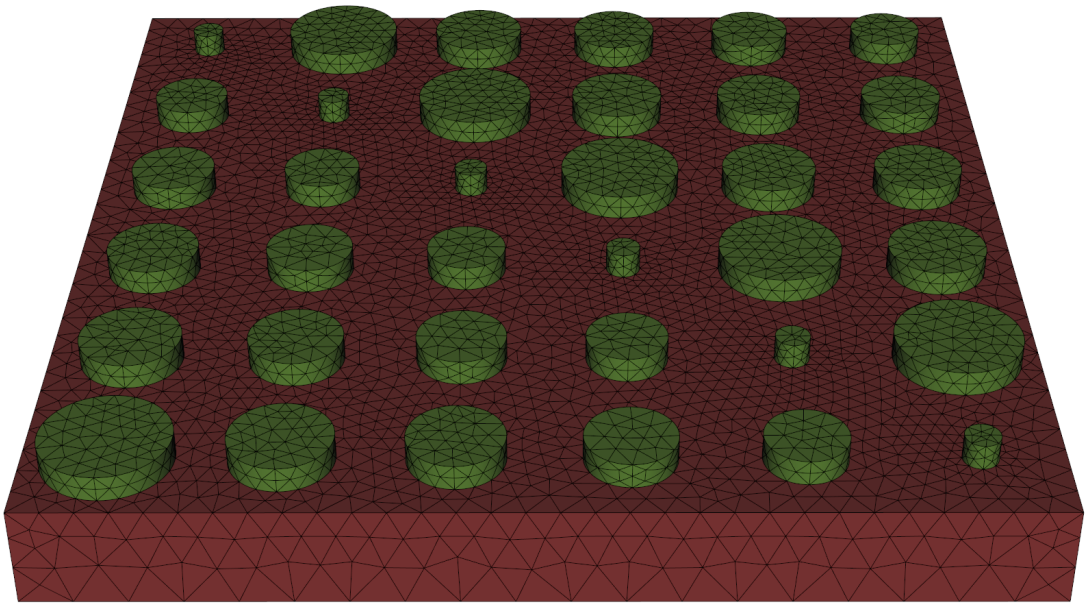


Figure 8.16 | Ideal 2D dielectric reflectarray mesh. The red tetrahedra correspond to silver, while the green ones are made of an anisotropic dielectric material. The device is surrounded by air and terminated by a PML above and below, and by periodic boundary conditions on the lateral sides.

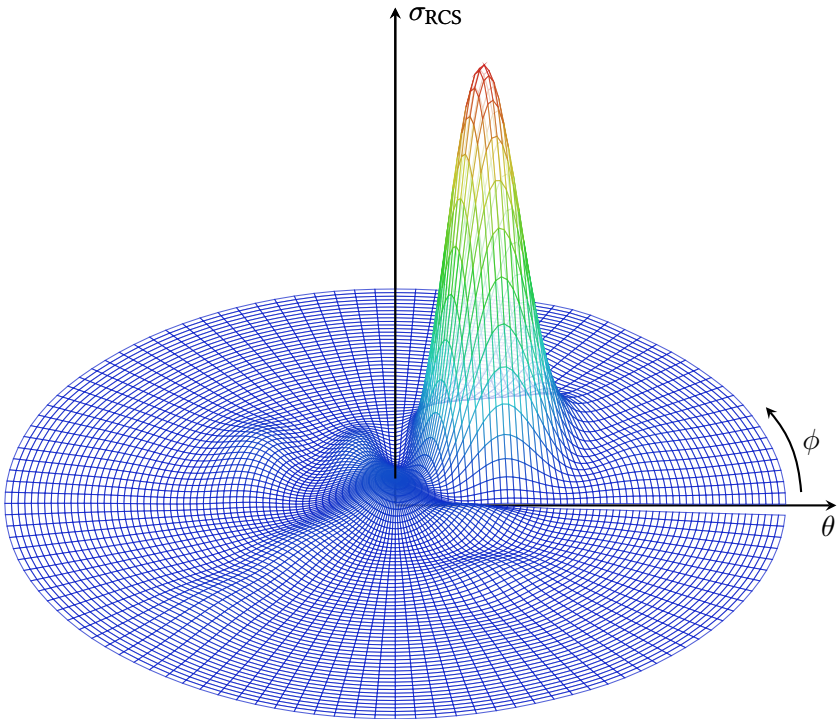


Figure 8.17 | Radar cross-section of an ideal 2D dielectric reflectarray at frequency f. θ varies from 0 to 90°, while ϕ varies from 0 to 360°. The directivity peak is observed around $(\theta, \phi) = (28.5°, 45°)$, with an efficiency close to 60%. Several parasitic lobes are observed, particularly around the normal direction $(\theta, \phi) = (0°, 0°)$.

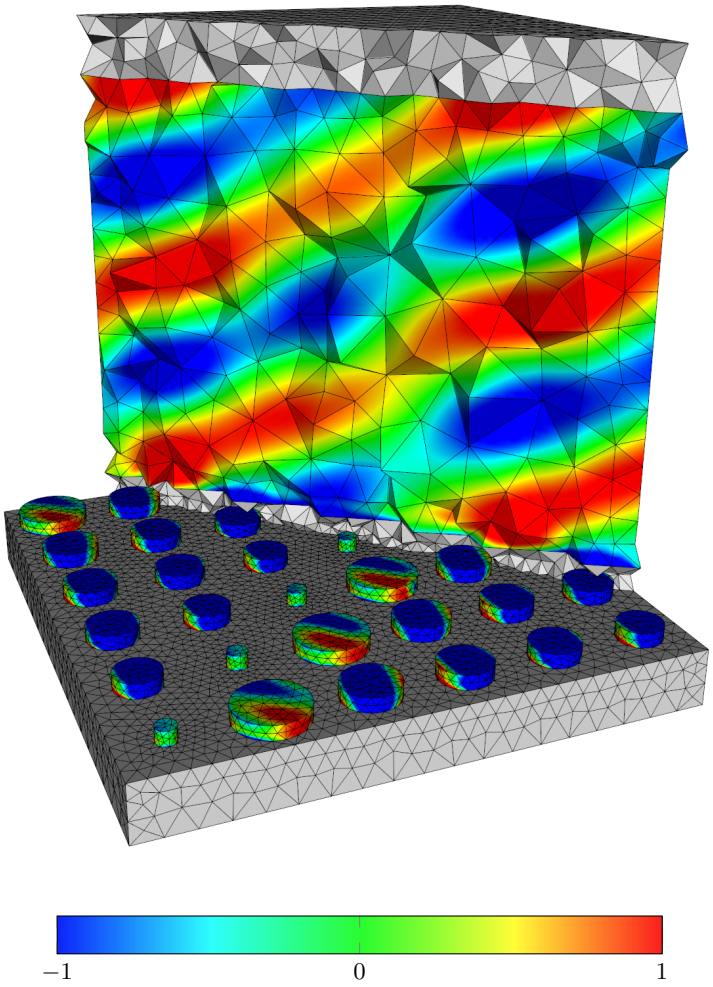


Figure 8.18 | Time-domain snapshot of E_y component for an ideal 2D dielectric reflectarray. Solution is obtained in established regime at $t = 0.1$ ps. Fields are scaled to $[-1, 1]$. The cutting plane is chosen following the direction of maximal radiation, *i.e.* $\phi = 45°$.

- The RCS treatment represents an additional imbalance cause between the processes. When computed sequentially, the total overhead generated by the computation of the RCS (*i.e.* on-the-fly Fourier transform during the computation and final post-treatment) represents 2% of the total computational time.

2D reflectarray

For the 2D array, the solution is obtained in 19 hours 8 minutes on 4 cores, and 7 hours 42 minutes on 16 cores, hence yielding a parallel efficiency of 0.73, similar to the 1D array. For respectively 4 and 16 cores, the parallel imbalances on the curved mesh are 8.9 % and 34.7 %, for a mesh containing 102.984 tetrahedra and 12 % of curved cells. As for the 1D array, these values drop to 6.6 % and 20.6 % respectively for linear mesh. These results highlight the need to take features such as curved elements, PMLs or on-the-fly computations for post-treatments into account at the partitionning level. It should also be noted that, in the case where one does not need field maps, these reflectarray computations can be dramatically shortened by reducing the amount of vacuum below and above the device.

9 Outlook

In this chapter, we go over the content of the present manuscript, and point out the future possible works that are or could be carried to progress toward more complex physics and performant computations.

9.1 Summary

The goal of this thesis was to elaborate a 3D discontinuous Galerkin time-domain code able to handle complex nano-optics configurations. In the following paragraphs, we review the content of this thesis, and point out our efforts and associated contributions toward this objective.

First, a customized generalized dispersive model was developed. This model covers a wide range of dispersive materials, and proved to be roughly twice more accurate to fit experimental data than the widespread Drude and Drude-Lorentz models, for standard metals such as gold and silver in the THz regime. A significant improvement was obtained for nickel (a transition metal) when comparing the performance of the Drude model with a single generalized pole. Finally, a short digression was made on non-local dispersive models, with preliminary results in 2D.

Then, the discontinuous Galerkin time-domain method was thoroughly developed and validated for non-dispersive and dispersive materials, and two time-stepping techniques taken from the literature were proposed. Several numerical experiments related to fluxes were conducted to complete this overview. To conclude, several theoretical proofs were given, some being the result of associated works conducted during this thesis.

The next chapter contains all the numerical developments necessary to handle the computation of realistic cases, such as perfectly-matched layers, total field/scattered field formulation, complex sources, or physical post-treatments. Although additional numerical experiments were conducted about the performances of absorbing boundary conditions and perfectly matched layers, these techniques were all adapted from the literature.

Two methodological developments were then investigated in order to improve the efficiency and the accuracy of the DGTD algorithm. First, the possibility to handle curvilinear elements was considered. This possibility is not new to the DG community, and after a presentation of the mathematical and numerical framework, our approach was resolutely oriented toward the improvements this technique could bring to nano-optics computations. Through increasingly-complex configurations, the use of curvilinear

elements proved to be a serious asset in terms of performance and accuracy. In the following chapter, the possibility to exploit variable polynomial orders through the computational domain was explored. After the necessary developments and a standard validation of the implementation, an order repartition algorithm was proposed that provided interesting speedups on meshes with heterogeneous mesh sizes (with a ratio up to 1000), while the accuracy of the results is barely altered (less than 1% of relative error). However, this implementation relies on a good preliminary knowledge of the physics involved in the considered configuration. A coupling with an *a posteriori* error estimate could lift this limitation by adapting the polynomial order on-the-fly, which could also alleviate the computational cost. The coupling of the order repartition algorithm with curvilinear elements can also constitute an interesting exploration path.

In the following chapter, the sequential and parallel performances of our code were assessed. A cell-renumbering algorithm was shown to provide interesting speedups, especially for low approximation orders. After a few numerical experiments with the Metis mesh partitioning tool, the speedup and efficiency of our parallel MPI implementation was assessed on a standard test-case. Results showed that this implementation provided an acceptable scaling up to a few hundred of cores, as long as the number of cells per core remained sufficiently high (around 10,000). Computation results from other chapters also proved that there was a serious need for a better load balance between cores when complex features (PMLs, curved elements, on-the-fly Fourier transforms, ...) were used.

The final chapter aims at demonstrating the capabilities of our current DGTD implementation on realistic cases. The first case consists in the computation of the EELS spectrum of a metallic nanosphere, and was adapted from existing literature. It constitutes a preliminary step toward more advanced works dealing with the proper treatment of electron-based electromagnetic sources. The second configuration involves the gap-plasmon resonances observed under chemically-produced nanocubes on metallic plates. First, the influence of the rounding at the cube edges was demonstrated. Then, different behaviors were identified, depending on the cube side length and the thickness of the dielectric spacer. These results will constitute the base of a wider study in collaboration with A. Moreau [MCM+12]. The last case deals with 1D and 2D dielectric reflectarrays, which goal is to reflect incident light with a tunable deflection angle. First, the impact of realistic lithography flaws on the performance of a 1D array was assessed. Then, the computation of a larger 2D reflectarray is considered. These results are the first step toward a wider study on this topic in collaboration with M. Klemm [ZLGW+14].

9.2 Future works

The topics presented in this manuscript give rise to a number of possible further developments, both from the numerical and the physical point of view. We close this manuscript with a short discussion of these topics.

9.2.1 Physics and material models

The numerical treatment of the non-local model, briefly presented in section 2.2.6, remains to be thoroughly studied in the DG framework. Because of the very small physical scale involved, 3D computations with non-local models promise to be computationally expensive, and an efficient parallel implementation would constitute a good asset to compute the response of large-scale systems.

The discretization of non-linear materials in the DG framework also remain to be explored in details. Literature on this topic is very shallow, with only a handful of references limited to 1D formulations for Kerr effects [Bla13] [FL15].

9.2.2 Numerical improvements

As was stated in the introduction, the DG method allows a large panel of methodological improvements, each presenting advantages and drawbacks. In most cases, the goal of these enrichments is either to (i) alleviate the number of degrees of freedom (hybrid [LVD+14] and non-conforming meshes [FL10]), to (ii) handle the ill time discretization induced by very small elements (local time-stepping [DG09], implicit/-explicit formulations [Moy12], space-time DG method [PFT00]), or to (iii) obtain a combination of both (*hp*-adaptivity [SW12]).

New numerical methods derived from the classical DG algorithm are also appearing. In the Hybridizable Discontinuous Galerkin (HDG) method, a Lagrange multiplier representing the trace of the numerical solution on the element faces is introduced. A global problem on the mesh skeletton (*i.e.* the faces of the mesh) is obtained and solved, before the volumic solution can be recovered with local, independent computations. Originally designed for the time-harmonic Maxwell's equations [NPC11], implicit time-domain HDG formulation for Maxwell's equations have been developped [LP11]. A more general technique, the Multiscale Hybrid Mixed (MHM) method [HPV13], includes the DG algorithm as an inner solver to handle large, multiscale problems. In this method, the final solution is obtained as the sum of (i) the global solution of the problem of a coarse mesh and (ii) local, independent solutions computed on finer meshes in each cell.

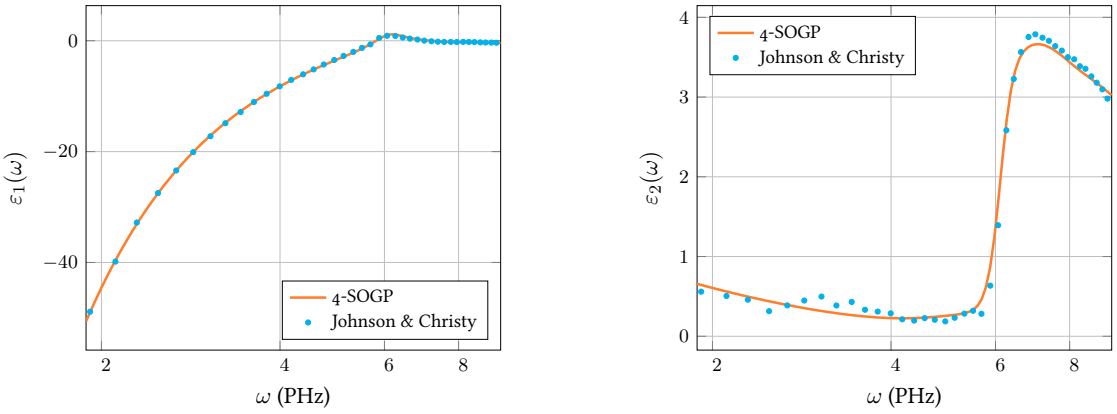
9.2.3 High-performance computing

To compute larger and larger nano-optics systems, one cannot solely rely on Moore's law, and needs to call upon adequate parallel implementations. As can be guessed from the results of the present manuscript, a specific effort must be made to obtain decent scalings out of very large clusters (*i.e.* from several thousands to tens of thousands). OMP- or MPI-only parallel implementations on standard CPU clusters are not likely to achieve such performances. Hybrid parallelism (MPI/OMP [LLS+14]) or specific implementations for advanced HPC architectures (cluster/booster division [LMDL15]) represent potential candidates to efficient massively parallel DG algorithms.

A

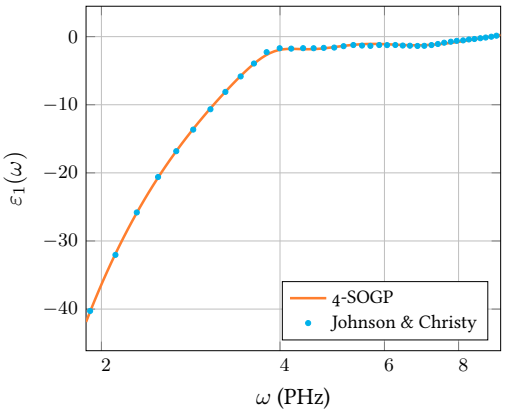
Dispersion parameters

This appendix provides dispersion coefficients for the generalized dispersive formulation for silver and gold over the $[300, 1500]$ THz frequency range. The experimental data, from [JC72], is fitted to different models with a simulated annealing algorithm [KGV83]. For each of the two metals, Drude, Drude-Lorentz, 2-SOGP and 4-SOGP coefficients are given. To ease the reading, quantities are given in PHz. Plots of the 4-SOGP permittivities are displayed in figure A.1.

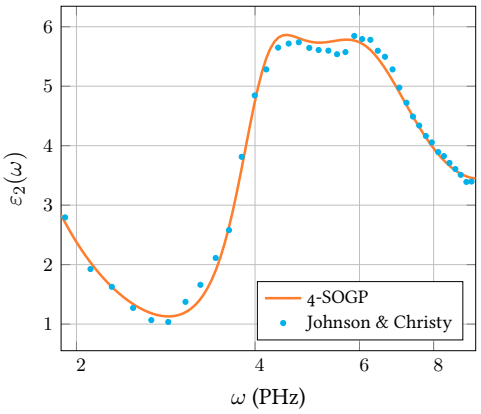


(a) Silver, real part

(b) Silver, imaginary part



(c) Gold, real part



(d) Gold, imaginary part

Figure A.1 | Real and imaginary parts of the silver and gold relative permittivity predicted by the 4-SOGP model compared to experimental data from Johnson & Christy.

Table A.1 | Coefficients of various dispersive models for gold.

Parameters	Drude	Drude-Lorentz	2-SOGP	4-SOGP
ε_∞	3.2629	3.6793	1.0	1.0
ω_d	12.147	13.456	–	–
γ_d	0.24304	0.0	–	–
$\Delta\varepsilon$	–	5.1899	–	–
ω_l	–	6.3681	–	–
γ_l	–	5.7923	–	–
c_1	–	–	161.08	171.61
d_1	–	–	9.0631	0.0
e_1	–	–	0.0	0.0
f_1	–	–	0.0	0.10449
c_2	–	–	20.437	210.17
d_2	–	–	19.956	0.0
e_2	–	–	15.247	149.61
f_2	–	–	2.8022	4.7838
c_3	–	–	–	95.959
d_3	–	–	–	2.3069
e_3	–	–	–	39.550
f_3	–	–	–	4.1445
c_4	–	–	–	15.321
d_4	–	–	–	4.7361
e_4	–	–	–	16.083
f_4	–	–	–	1.4217

Table A.2 | Coefficients of various dispersive models for silver.

Parameters	Drude	Drude-Lorentz	2-SOGP	4-SOGP
ε_∞	3.7362	2.7311	1.2944	1.0
ω_d	13.871	14.084	–	–
γ_d	0.045154	0.0066786	–	–
$\Delta\varepsilon$	–	1.6336	–	–
ω_l	–	8.1286	–	–
γ_l	–	3.6448	–	–
c_1	–	–	189.09	191.92
d_1	–	–	2.6584	0.73725
e_1	–	–	0.0	0.0
f_1	–	–	0.0	0.0
c_2	–	–	56.165	164.28
d_2	–	–	12.005	0.0
e_2	–	–	43.932	75.648
f_2	–	–	3.1709	14.161
c_3	–	–	–	10.581
d_3	–	–	–	10.654
e_3	–	–	–	38.369
f_3	–	–	–	4.3307
c_4	–	–	–	0.0
d_4	–	–	–	1.9950
e_4	–	–	–	37.575
f_4	–	–	–	0.94994

B

Non-conforming $\mathbb{P}_p - \mathbb{P}_m$ matrices

$$\mathbb{S}_{12} = \frac{1}{60}\begin{bmatrix} 2 & -1 & -1 & 8 & 4 & 8 \\ -1 & 2 & -1 & 8 & 8 & 4 \\ -1 & -1 & 2 & 4 & 8 & 8 \end{bmatrix}$$

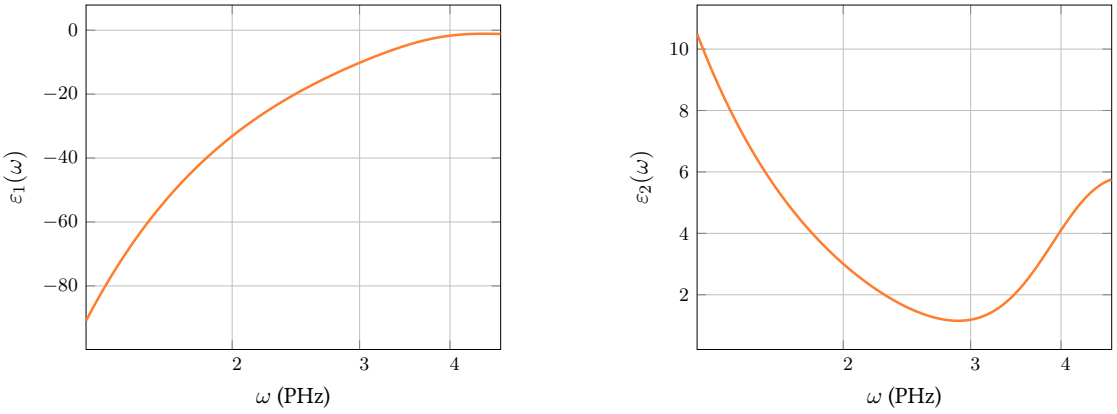
$$\mathbb{S}_{23} = \frac{1}{840}\begin{bmatrix} 10 & 1 & 1 & 27 & -18 & -6 & -6 & -18 & 27 & -18 \\ 1 & 10 & 1 & -18 & 27 & 27 & -18 & -6 & -6 & -18 \\ 1 & 1 & 10 & -6 & -6 & -18 & 27 & 27 & -18 & -18 \\ 4 & 4 & 8 & 48 & 48 & 24 & -12 & -12 & 24 & 144 \\ 8 & 4 & 4 & -12 & 24 & 48 & 48 & 24 & -12 & 144 \\ 4 & 8 & 4 & 24 & -12 & -12 & 24 & 48 & 48 & 144 \end{bmatrix}$$

$$
\mathbb{S}_{34} = \frac{1}{2520}
\left[\begin{array}{cccccccccccccccc}
14 & -1 & -1 & 32 & -28 & 16 & 8 & 0 & 8 & 16 & -28 & 32 & -16 & 16 & 16 \\
-1 & 14 & -1 & 16 & -28 & 32 & 32 & -28 & 16 & 8 & 0 & 8 & 16 & -16 & 16 \\
-1 & -1 & 14 & 8 & 0 & 8 & 16 & -28 & 32 & 32 & -28 & 16 & 16 & 16 & -16 \\
9 & -6 & -6 & 96 & 36 & -24 & -24 & 0 & 0 & 24 & -36 & 48 & 216 & -72 & -72 \\
-6 & 9 & -6 & -24 & 36 & 96 & 48 & -36 & 24 & 0 & 0 & -24 & -72 & 216 & -72 \\
-6 & 9 & -6 & 24 & -36 & 48 & 96 & 36 & -24 & -24 & 0 & 0 & -72 & 216 & -72 \\
-6 & -6 & 9 & 0 & 0 & -24 & -24 & 36 & 96 & 48 & -36 & 24 & -72 & -72 & 216 \\
-6 & -6 & 9 & -24 & 0 & 0 & 24 & -36 & 48 & 96 & 36 & -24 & -72 & -72 & 216 \\
9 & -6 & -6 & 48 & -36 & 24 & 0 & 0 & -24 & -24 & 36 & 96 & 216 & -72 & -72 \\
-6 & -6 & -6 & 48 & 0 & 48 & 48 & 0 & 48 & 48 & 0 & 48 & 288 & 288 & 288
\end{array}\right]
$$

C

Another gold permittivity function

This appendix provides the coefficients for the 3-SOGP model used to approximate the gold behaviour in the nanocubes computations of section 8.2. Units are the same as in appendix A. The frequency range of interest is here $[200, 750]$ THz.



(a) Real part

(b) Imaginary part

Figure C.1 | Real and imaginary parts of the gold relative permittivity predicted by a 3-SOGP model for gold nanocubes.

Table C.1 | Coefficients of 3-SOGP model for gold nanocubes. Quantities are given in PHz.

Parameters	3-SOGP
ε_∞	1.0
c_1	496.49
d_1	0.0
e_1	12.709
f_1	127.52
c_2	150.19
d_2	1.2356
e_2	0.0
f_2	0.097506
c_3	34.069
d_3	17.762
e_3	16.842
f_3	2.7438

Publications

Research articles

- *A DGTD method for the numerical modeling of the interaction of light with nanometer scale metallic structures taking into account non-local dispersion effects*; N. Schmitt, C. Scheid, S. Lanteri, A. Moreau and J. Viquerat, submitted
- *Analysis of a generalized dispersive model coupled to a DGTD method with application to nanophotonics*; S. Lanteri, C. Scheid and J. Viquerat, submitted
- *Simulation of near-field plasmonic interactions with a local approximation order discontinuous Galerkin time-domain method*; J. Viquerat and S. Lanteri, Photonics and Nanostructures - Fundamentals and Applications, **18**, 43 – 58 (2016)
- *A 3D curvilinear discontinuous Galerkin time-domain solver for nanoscale light-matter interactions*; J. Viquerat and C. Scheid, Journal of Computational and Applied Mathematics, **289**, 37 – 50 (2015)
- *A parallel non-conforming multi-element DGTD method for the simulation of electromagnetic wave interaction with metallic nanoparticles*; R. Léger, J. Viquerat, C. Durochat, C. Scheid and S. Lanteri, Journal of Computational and Applied Mathematics, **270**, 330 – 342 (2014)
- *Recent advances on a DGTD method for time-domain electromagnetics*; S. Descombes, C. Durochat, S. Lanteri, L. Moya, C. Scheid and J. Viquerat, Photonics and Nanostructures – Fundamentals and Applications, **11**, 291 – 302 (2013)

Oral presentations

- *Curvilinear discontinuous Galerkin time-domain method for nanophotonics*; ACOMEN, Ghent (2014)
- *Discontinuous Galerkin time-domain method for nanophotonics*; META conference, Singapore (2014)
- *Discontinuous Galerkin time-domain method for nanophotonics*; WAVES conference, Tunis (2013)
- *Simulation de la propagation d'ondes électromagnétiques en nano-optique par une méthode Galerkin discontinue d'ordre élevé*; GDR Ondes GT2, Troyes (2012)

Bibliography

[AABG12] J. Alvarez, L. D. Angulo, A. R. Bretones, and S. G. Garcia. 3D discontinuous Galerkin time-domain method for anisotropic materials. IEEE Microwave and Wireless Components Letters, 11: 1182 – 1185, 2012.

[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.

[Aba10] F. J. G. Abajo. Optical excitations in electron microscopy. Review of modern physics, 82: 209 – 275, 2010.

[Bó7] J.-P. Bérenger. Perfectly matched layer for computational electromagnetics. Morgan & Claypool, first edition, 2007.

[BFHL09] D. Baumann, C. Fumeaux, C. Hafner, and E. P. Li. A modular implementation of dispersive materials for time-domain simulations with application to gold nanospheres at optical frequencies. Optics Express, 17, 2009.

[BFLP06] M. Bernacki, L. Fezoui, S. Lanteri, and S. Piperno. Parallel discontinuous Galerkin unstructured mesh solvers for the calculation of three-dimensional wave propagation problems. Applied Mathematical Modelling, 30: 744 – 763, 2006.

[BKN11] K. Busch, M. König, and J. Niegemann. Discontinuous Galerkin methods in nanophotonics. Laser Photonics Review, 5: 1 – 37, 2011.

[Bla13] E. Blank. The Discontinuous Galerkin method for Maxwell's equations, application to bodies of revolution and Kerr-nonlinearities. PhD thesis, Karlsruher Instituts fur Technologie, 2013.

[Boa82] A. D. Boardman. Electromagnetic Surface Modes. Wiley, New-York, 1982.

[BPG02] E. Bécache, P. G. Petropoulous, and S. D. Gedney. On the long-time behavior of unsplit perfectly matched layers. Technical report, INRIA, 2002.

[BR97] F. Bassi and S. Rebay. High-order accurate discontinuous finite element solution of the 2D Euler equations. Journal of Computational Physics, 138: 251 – 285, 1997.

[Buc04] J. A. Buck. Fundamentals of optical fibers. Wiley-Blackwell, second edition, 2004.

[But87] J. C. Butcher. The Numerical Analysis of Ordinary Differential Equations: Runge-Kutta and General Linear Methods. Wiley, 1987.

[CBB09] F. Costen, J.-P. Bérenger, and A. K. Brown. Comparison of FDTD hard source with FDTD soft source and accuracy assessment in Debye media. IEEE Transactions on Antennas and Propagation, 57: 2014 – 2022, 2009.

[CC41] K. S. Cole and R. H. Cole. Dispersion and absorption in dielectrics - I Alternating current characteristics. Journal of Chemical Physics, 9: 341 – 352, 1941.

[CCKS07] W. Cai, U. Chettiar, A. Kildishev, and V. Shalaev. Optical cloaking with metamaterials. Nature Photonics, 1: 224 – 227, 2007.

[CFS06] J.-P. Cioni, L. Fezoui, and H. Steve. Approximation des équations de maxwell par des schemas décentrés en éléments finis. Technical report, INRIA, 2006.

[CK94] M. H. Carpenter and C. A. Kennedy. Fourth-order 2n-storage Runge-Kutta schemes. Technical report, National Aeronautics and Space Administration, 1994.

[CLS+11] T. Chung, S. Y. Lee, E. Y. Song, H. Chun, and B. Lee. Plasmonic nanostructures for nanoscale biosensing. Sensors, 11: 10907 – 10929, 2011.

[CMLN15] Y. Cao, A. Manjavacas, N. Large, and P. Nordlander. Electron energy-loss spectroscopy calculation in finite-difference time-domain package. ACS Photonics, 2: 369 – 375, 2015.

[Coo03] R. Cools. An Encyclopaedia of cubature formulas. Journal of Complexity, 19: 445 – 453, 2003.

[DCTS08] J. Dai, F. Cajko, I. Tsukerman, and M. I. Stockman. Electrodynamics effects in plasmonic nanolenses. Physical Review B, 77: 115419, 2008.

[DDH01] A. Ditkowski, K. Dridi, and J. S. Hesthaven. Convergent cartesian grid methods for Maxwell's equations in complex geometries. Journal of Computational Physics, 170: 39 – 80, 2001.

[DF94] B. T. Draine and P. J. Flatau. Discrete-dipole approximation for scattering calculations. Journal of the Optical Society of America A, 11: 1491 – 1499, 1994.

[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.

[Die12] R. T. H. Diehl. Analysis of metallic nanostructures by a discontinuous Galerkin time-domain Maxwell solver on graphics processing units. PhD thesis, Karlsruher Instituts für Technologie, 2012.

[DNBH15] A. Demirel, J. Niegemann, K. Busch, and M. Hochbruck. Efficient multiple time-stepping algorithms of higher order. Journal of Computational Physics, 285: 133 – 148, 2015.

[DOS99] S. Dey, R. M. O'Bara, and M. S. Shephard. Curvilinear mesh generation in 3D. Proceedings of the 8th International Meshing Roundtable, pages 407 – 417, 1999.

[Dru00] P. Drude. Zur elektronentheorie der metalle. Annalen der Physik, 306: 566 – 613, 1900.

[DWSL07] G. Dolling, M. Wegener, C. M. Soukoulis, and S. Linden. Negative-index metamaterials at 780 nm wavelength. Optical Letters, 32: 53 – 55, 2007.

[ESVM+06] R. Esquivel-Sirvent, C. Villarreal, W. L. Mochan, A. M. Contreras-Reyes, and V. B. Svetovo. Spatial dispersion in casimir forces : a brief review. Journal of Physics A : Mathematical and General, 39: 6323 – 6331, 2006.

[F+05] L. Fezoui et al. 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.

[Fah11] H. Fahs. Improving accuracy of high-order discontinuous Galerkin method for time-domain electromagnetics on curvilinear domains. International Journal of Computer Mathematics, 88: 1 – 30, 2011.

[Fey10] R. Feynman. The Feynman lectures on physics, Volume II. Basic Books, millenium edition edition, 2010.

[FL10] H. Fahs and S. Lanteri. A high-order non-conforming discontinuous Galerkin method for time-domain electromagnetics. Journal of Computational and Applied Mathematics, 234: 1088 – 1096, 2010.

[FL15] L. Fezoui and S. Lanteri. Discontinuous Galerkin methods for the numerical solution of the nonlinear Maxwell equations in 1D. Technical report, INRIA, 2015.

[Fle78] J. W. Fleming. Material dispersion in lightguide glasses. Electronics Letters, 14: 326 – 328, 1978.

[FVV07] S. Foteinopoulo, J. P. Vigneron, and C. Vandenbem. Optical near-field excitations on plasmonic nanoparticle-based structures. Optics Express, 15: 4253 – 4267, 2007.

[GGC96] C. Gabriel, S. Gabriel, and E. Corthout. The dielectric properties of biological tissues: I. Literature survey. Physics in Medicine and Biology, 41: 2231 – 2249, 1996.

[Gon13] J. A. Gonzalez. A discontinuous Galerkin finite element method for the time-domain solution of Maxwell's equations. PhD thesis, Universidad de Granada, 2013.

[GPK14] B. Grzeskiewicz, K. Ptaszynski, and M. Kotkowiak. Near and far-field properties of nanoprisms with rounded edges. Plasmonics, 9: 607 – 614, 2014.

[GR09] C. Geuzaine and J.-F. Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. International Journal for Numerical Methods in Engineering, 0: 1 – 24, 2009.

[GS99] B. Gustavsen and S. Semlyen. Rational approximation of frequency domain responses by vector fitting. Transactions on Power Delivery, 14: 1052 – 1061, 1999.

[HCB05] T. JR. Hughes, J. A. Cottrell, and Y. Bazilevs. Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement. Computer methods in applied mechanics and engineering, 194: 4135 – 4195, 2005.

[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.

[HE08] J. Huang and J. A. Encinar. Reflectarray antennas. IEEE Press, 2008.

[HHG+10] N. A. Hatab, C.-H. Hsueh, A. L. Gaddis, S. T. Retterer, J.-H. Li, G. Eres, Z. Zhang, and B. Gu. Free-standing optical gold bowtie nanoantenna with variable gap size for enhanced Raman spectroscopy. Nano Letters, 10: 4952 – 4955, 2010.

[HKGE10] A. Hille, R. Kullock, S. Grafström, and L. M. Eng. Improving nano-optical simulations through curved elements implemented within the discontinuous Galerkin method. Journal of Computational and Theoretical Nanoscience, 7: 1581 – 1586, 2010.

[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.

[HN07] F. Hao and P. Nordlander. Efficient dielectric function for FDTD simulation of the optical properties of silver and gold nanoparticles. Chemical Physics Letters, 446: 115 – 118, 2007.

[HPV13] C. Harder, D. Paredes, and F. Valentin. A family of multiscale hybrid-mixed finite element methods for the Darcy equation with rough coefficients. Journal of Computational Physics, 245: 107 – 130, 2013.

[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.

[HT12] U. Hohenester and A. Trugler. MNPBEM - a Matlab toolbox for the simulation of plasmonic particles. Computer Physics Communications, 183: 370 – 381, 2012.

[HW01] J. S. Hesthaven and T. Warburton. High-order/spectral methods on unstructured grids. Technical report, National Aeronautics and Space Administration, 2001.

[HW08] J. S. Hesthaven and T. Warburton. Nodal discontinuous Galerkin methods. Springer, 2008.

[Jac98] J. D. Jackson. Classical Electrodynamics. J. Wiley and Sons, third edition, 1998.

[JC72] P. B. Johnson and R. W. Christy. Optical constants of the noble metals. Physical Review B, 6: 4370 – 4379, 1972.

[JC74] P. B. Johnson and R. W. Christy. Optical constants of transition metals: Ti, V, Cr, Mn, Fe, Co, Ni, and Pd. Physical Review B, 9: 5056 – 5070, 1974.

[JJ07] J. D. Joannopoulos and S. G. Johnson. Photonic Crystals, Molding the Flow of Light. Princeton University Press, second edition, 2007.

[JRG12] A. Johen, J.-F. Remacle, and C. Geuzaine. Geometrical validity of curvilinear finite elements. Proceedings of the 20th International Meshing Roundtable, pages 255 – 271, 2012.

[KBN10] M. König, K. Busch, and J. Niegemann. The discontinuous Galerkin time-domain method for Maxwell's equations with anisotropic materials. Photonics and Nanostructures : Fundamentals and Applications, 8: 303 – 309, 2010.

[Kei98] J. F. Keithley. The Story of Electrical and Magnetic Measurements: From 500 BC to the 1940s. Wiley-IEEE Press, 1998.

[KGV83] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. Science, 220: 671 – 680, 1983.

[KK99] G. Karypis and V. Kumar. A fast and highly quality multilevel scheme for partitioning irregular graphs. SIAM Journal of Scientific Computing, 20: 359 – 392, 1999.

[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] A. M. Kern and O. J. F. Martin. Pitfalls in the determination of optical cross sections from surface integral equations simulations. IEEE Transactions on antennas and propagation, 58: 2158 – 2161, 2010.

[Kön11] M. König. Discontinuous Galerkin Methods in Nanophotonics. PhD thesis, Fakultät für Physik des Karlsruher Instituts für Technologie, 2011.

[KS05] V. I. Krylov and A. H. Stroud. Approximate calculation of integrals. Dover Publications Inc., 2005.

[Lam91] J. D. Lambert. Numerical Methods for Ordinary Differential Systems. Wiley, 1991.

[LBU+08] F. Le, D. W. Brandl, Y. A. Urzhumov, H. Wang, J. Kundu, N. J. Halas, J. Aizpurua, and P. Nordlander. Metallic nanoparticle arrays: a common substrate for both surface-enhanced Raman scattering and surface-enhanced infrared absorption. ACS Nano, 2: 707 – 718, 2008.

[LC09] J. Y. Lu and Y. H. Chang. Implementation of an efficient dielectric function into the finite difference time domain method for simulating the coupling between localized surface plasmons of nanostructures. Superlattices and Microstructures, 47: 60 – 65, 2009.

[LL60] L. D. Landau and E. M. Lifshitz. Electrodynamics of Continuous Media. Pergamon Press, 1960.

[LLS+14] S. Lanteri, R. Léger, C. Scheid, J. Viquerat, T. Cabel, and G. Hautreux. Hybrid MIMD/SIMD high order DGTD solver for the numerical modeling of light/matter interaction on the nanoscale. PRACE, 2014.

[LMDL15] R. Léger, M. Alvarez Mallon, A. Duran, and S. Lanteri. Assessing the DEEP-ER cluster-/booster architecture with a finite-element type solver for bioelectromagnetics. PARCO Conference 2015, 2015.

[LP11] S. Lanteri and R. Perrussel. An implicit hybridized discontinuous Galerkin method for time-domain Maxwell's equations. Technical report, INRIA, 2011.

[LS76] W. Liu and A. Sherman. Comparative analysis of the Cuthill-McKee and the Reverse Cuthill-McKee ordering algorithms for sparse matrices. SIAM Journal on Numerical Analysis, 13: 198 – 213, 1976.

[LSB03] K. Li, M. I. Stockman, and D. J. Bergman. Self-similar chains of metal nanospheres as an efficient nanolens. Physical Review Letters, 91: 227402, 2003.

[LSV] S. Lanteri, C. Scheid, and J. Viquerat. Analysis of a generalized dispersive model coupled to a DGTD method with application to nanophotonics. Submitted, 2015.

[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.

[LZM+10] B. Luk'yanchuk, N. I. Zheludev, S. A. Maier, N. J. Halas, P. Nordlander, H. Giessen, and C. T. Chong. The Fano resonance in plasmonic nanostructures and metamaterials. Nature Materials, 9: 707 – 715, 2010.

[Mai07] S. A. Maier. Plasmonics: Fundamentals and Applications. Springer, 2007.

[Max65] J. C. Maxwell. A dynamical theory of the electromagnetic field. Philosophical Transactions of the Royal Society of London, 155: 459 – 512, 1865.

[MCM+12] A. Moreau, C. Ciraci, J. J. Mock, R. T. Hill, Q. Wang, B. J. Wiley, A. Chilkoti, and D. R. Smith. Controlled-reflectance surfaces with film-coupled colloidal nanoantennas. Nature, 492: 86 – 90, 2012.

[MCS13] A. Moreau, C. Ciraci, and D. R. Smith. The impact of nonlocal response on metallo-dielectric multilayers and optical patch antennas. Physical Review B, 87: 045401, 2013.

[MG81] M. G. Moharam and T. K. Gaylord. Rigorous coupled-wave analysis of planar-grating diffraction. Journal of the Optical Society of America, 71: 811 – 818, 1981.

[MNHB11] C. Matyssek, J. Niegemann, W. Hergert, and K. Busch. Computing electron energy loss spectra with the discontinuous Galerkin time-domain method. Photonics and Nanostructures, 9: 367 – 373, 2011.

[Mon03] P. Monk. Finite Element Methods for Maxwell's Equations. Oxford Science Publications, 2003.

[Moy12] L. Moya. Temporal convergence of a locally implicit discontinuous Galerkin method for Maxwell's equations. ESAIM : Mathematical Modelling and Numerical Analysis, 46: 1225 – 1246, 2012.

[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.

[Nie09] J. Niegemann. Higher-Order Methods for Solving Maxwell's Equations in the Time-Domain. PhD thesis, Fakultät für Physik des Karlsruher Instituts für Technologie, 2009.

[NKP+10] J. Niegemann, M. König, C. Prohm, R. Diehl, and K. Busch. Using curved elements in the discontinuous Galerkin time-domain approach. AIP Conference Proceedings, 1291: 76, 2010.

[NKSB09] J. Niegemann, M. König, K. Stannigel, and K. Busch. Higher-order time-domain methods for the analysis of nano-photonic systems. Photonics and Nanostructures - Fundamentals and Applications, 7: 2 – 11, 2009.

[NPC11] N. C. Nguyen, J. Peraire, and B. Cockburn. Hybridizable discontinuous Galerkin methods for time-harmonic Maxwell's equations. Journal of Computational Physics, 230: 7151 – 7175, 2011.

[Pal98] E. D. Palik. Handbook of Optical Constants of Solids. Academic Press, 1998.

[Pav13] P. Pavaskar. Electromagnetic modeling of plasmonic nanostructures. PhD thesis, University of Southern California, 2013.

[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.

[Pip05] S. Piperno. Symplectic local time-stepping in non-dissipative DGTD methods applied to wave propagation problems. Technical report, Inria Sophia Antipolis, Project-team Caïman, 2005.

[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.

[PPM+14] Z. Pirzadeh, T. Pakizeh, V. Miljkovic, C. Langhammer, and A. Dmitriev. Plasmon-interband coupling in Nickel nanoantennas. ACS Photonics, 1: 158 – 162, 2014.

[RC01] E. J. Rothwell and M. J. Cloud. Electromagnetics. CRC Press, 2001.

[RDEM98] A. D. Rakic, A. B. Djurisic, J. M. Elazar, and M. L. Majewski. Optical properties of metallic films for vertical-cavity optoelectronic devices. Applied Optics, 37: 5271 – 5283, 1998.

[RF98] M. Remaki and L. Fezoui. Une méthode de Galerkin discontinu pour la résolution des équations de Maxwell en milieu hétérogene. Technical report, Inria Sophia Antipolis, Project-team Caïman, 1998.

[RFH28] Courant Richard, K. O. Friedrichs, and Lewy Hans. Uber die partiellen Differenzengleichungen der mathematischen Physik. Mathematische Annalen, 100: 32 – 74, 1928.

[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.

[RH73] W. H. Reed and T. R. Hill. Triangular mesh method for the neutron transport equation. Technical report, Los Alamos National Laboratory, 1973.

[SCG10] B. Salski, M. Celuch, and W. Gwarek. FDTD for nanoscale and optical problems. IEEE Microwave Magazine, 10: 50 – 59, 2010.

[Sch97] J. Schöberl. NETGEN : An advancing front 2D/3D-mesh generator based on abstract rules. Computing and Visualization in Science, 1: 41 – 52, 1997.

[Sil69] P. Silvester. Finite element solution of homogeneous waveguide problems. Alta Frequenza, 38: 313 – 317, 1969.

[SKNB09] K. Stannigel, M. Koenig, J. Niegemann, and K. Busch. Discontinuous Galerkin time-domain computations of metallic nanostructures. Optics Express, 17: 14934 – 14947, 2009.

[SL11] C. Scheid and S. Lanteri. Convergence of a discontinuous Galerkin scheme for the mixed time domain maxwell's equations in dispersive media. Technical report, INRIA, 2011.

[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.

[SSD+14] G. A. Sotirou, F. Starsich, A. Dasargyri, M. C. Wurning, 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: 2818 – 2827, 2014.

[SSL+] N. Schmitt, C. Scheid, S. Lanteri, A. Moreau, and J. Viquerat. A DGTD method for the numerical modeling of the interaction of light with nanometer scale metallic structures taking into account non-local dispersion effects. Submitted, 2015.

[SSS13] M. Siebenborn, V. Schulz, and S. Schmidt. A curved-element unstructured discontinuous Galerkin method on GPUs for the Euler equations. arXiv, 1208.4772, 2013.

[Str04] J. Strikwerda. Finite difference schemes and partial differential equations. SIAM, second edition, 2004.

[SW12] S. M. Schnepp and T. Weiland. Efficient large scale electromagnetic simulations using dynamically adapted meshes with the discontinuous Galerkin method. Journal of Computational and Applied Mathematics, 236: 4909 – 4924, 2012.

[TDMS09] A. Taube, M. Dumbser, C.-D. Munz, and R. Schneider. A high-order discontinuous Galerkin method with time-accurate local time stepping for the Maxwell equations. International Journal of Numerical Modelling: Electronic Networks, Devices and Fields, 22: 77 – 103, 2009.

[TGRL12] T. Toulorge, C. Geuzaine, J.-F. Remacle, and J. Lambrechts. Robust untangling of curvilinear meshes. Proceedings of the 21st International Meshing Roundtable, pages 71 – 83, 2012.

[TH05] A. Taflove and S. Hagness. Computational Electrodynamics: The Finite-Difference Time-Domain Method. Artech House, Boston, third edition, 2005.

[Tho73] V. Thomée. Polygonal domain approximation in Dirichlet's problem. Journal of the Institute of Mathematics and its Applications, 11: 33 – 44, 1973.

[Tor09] E. F. Toro. Riemann solvers and numerical methods for fluid dynamics. Springer, third edition, 2009.

[VB09] J. G. Verwer and M. A. Botchev. Unconditionaly stable integration of maxwell's equations. Linear Algebra and Applications, 431: 300 – 317, 2009.

[vdH81] H. C. van de Hulst. Light scattering by small particles. Dover Pulibcations, Inc., 1981.

[Vel] J. Velimsky. Gestikulator - Generator of a tetrahedral mesh on a sphere.

[Ver10] J. G. Verwer. Component splitting for semi-discrete maxwell's equations. BIT Numerical Mathematics, 51: 427 – 445, 2010.

[VKLS13] J. Viquerat, M. Klemm, S. Lanteri, and C. Scheid. Theoretical and numerical analysis of local dispersion models coupled to a discontinuous Galerkin time-domain method for Maxwell's equations. Technical report, INRIA, 2013.

[VLDC11] A. Vial, T. Laroche, M. Dridi, and L. Le 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.

[War06] T. Warburton. An explicit construction of interpolation nodes on the simplex. Journal of Engineering Mathematics, 6: 247 – 262, 2006.

[War10] T. Warburton. A low storage curvilinear discontinuous Galerkin time-domain method for electromagnetics. URSI International Symposium on Electromagnetic Theory, pages 996 – 999, 2010.

[Wil56] C. H. Wilcox. An expansion theorem for electromagnetic fields. Communications on pure and applied mathematics, 9: 115 – 134, 1956.

[Wil80] J. H. Williamson. Low-storage Runge-Kutta scheme. Journal of Computational Physics, 35: 48 – 56, 1980.

[WK89] J. P. Webb and V. N. Kanellopoulos. Absorbing boundary conditions for the finite element solution of the vector wave equation. Microwave and optical technology letters, 2: 370 – 372, 1989.

[WROB13] C. Wolff, R. Rodriguez-Oliveros, and K. Busch. Simple magneto-optic transition metal models for time-domain simulations. Optics Express, 21: 12022 – 12037, 2013.

[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.

[ZCL09] L. Zhang, T. Cui, and H. Liu. A set of symmetric quadrature rules on triangles and tetrahedra. Journal of Computational Mathematics, 27: 89 – 96, 2009.

[ZIC14] J. Zhang, M. Irannejad, and B. Cui. Bowtie nanoantenna with single-digit nanometer gap for surface-enhanced Raman scattering (SERS). Plasmonics, 2014.

[ZLGW+14] L. Zou, M. Lopez-Garcia, W. Withayachumnankul, C. M. Shah, A. Mitchell, M. Bhaskaran, S. Sriram, R. Oulton, M. Klemm, and C. Fumeaux. Spectral and angular characteristics of dielectric resonator metasurface at optical frequencies. Applied Physics Letters, 105: 191109, 2014.

[ZWS+13] L. Zou, W. Withayachumnankul, C. Shah, A. Mitchell, M. Bhaskaran, S. Sriram, and C. Fumeaux. Dielectric resonator nanoantennas at visible frequencies. Optics Express, 21: 1344 – 1352, 2013.