PhD Dissertation

International Doctoral School in Information and Communication Technologies

DISI - University of Trento

Parallel FDTD Electromagnetic Simulation of Dispersive Plasmonic Nanostructures and Opal Photonic Crystals in the Optical Frequency Range

Antonino Calà Lesina

Advisor: Dott. Alessandro Vaccari

FBK - Fondazione Bruno Kessler

REET - Renewable Energies and Environmental Technologies

XXV Cycle - April 2013

Abstract

In the last decade, nanotechnology has enormously and rapidly developed. The technological progress has allowed the practical realization of devices that in the past have been studied only from a theoretical point of view. In particular we focus here on nanotechnologies for the optical frequency range, such as plasmonic devices and photonic crystals, which are used in many areas of engineering. Plasmonic noble metal nanoparticles are used in order to improve the photovoltaic solar cell efficiency for their forward scattering and electromagnetic field enhancement properties. Photonic crystals are used for example in low threshold lasers, biosensors and compact optical waveguide. The numerical simulation of complex problems in the field of plasmonics and photonics is cumbersome. The dispersive behavior has to be modeled in an accurate way in order to have a detailed description of the fields. Besides the code parallelization is needed in order to simulate large and realistic problems. Finite Difference Time Domain (FDTD) is the numerical method used for solving the Maxwell's equations and simulating the electromagnetic interaction between the optical radiation and the nanostructures. A modified algorithm for the Drude dispersion is proposed and validated in the case of noble metal nanoparticles. The modified approach is extended to other dispersion models from a theoretical point of view. A parallel FDTD code with a mesh refinement (subgridding) for the more detailed regions has been developed in order to speed up the simulation time. The parallel approach is also needed for the large amount of required memory due to the dimension of the analysis domain. Plasmonic nanostructures of different shapes and dimensions on the front surface of *a silicon layer have been simulated. The forward field scattering has been evaluated in order to optimize the concentration of the light inside the active region of the solar cell. Some design parameters have been deduced from this study. Opal photonic crystals with different filling factors have been simulated in order to tune the optical transmittance band-gap and find a theoretical explanation to the experimental evidences.*

Acknowledgements

Ringrazio Alessandro, il mio advisor, è suo il merito di questo traguardo. Ne ho ammirato la dedizione, attinto la passione. Sono stato seguito con responsabilità. Mi sono nutrito di umanità prima che di nozioni.

E' stato un cammino lungo e faticoso.
Ho incontrato persone sincere, respirato aria di casa.
Ho colto il mio riflesso in molti sguardi.
Mi sono arricchito incrociando vari mondi.

Negli occhi di Maurizio, Luca, Lucia, Alessio e Andrea
ho visto risplendere la gioia della conoscenza.
Il sapere senza amore non può dirsi tale.
Li ringrazio perché sono persone di inestimabile bellezza.

Ringrazio l'unità REET di FBK per avermi accolto
e fatto crescere supportando ogni mio slancio.
Ringrazio la commissione di dottorato, mi ha dato emozioni e gratificazione.
Ringrazio la mia famiglia, presenza costante nella mia vita.
Ringrazio i nuovi e i vecchi amici, rendono i miei giorni felici.
Ringrazio chi mi ha regalato serenità e affetto. E anche chi, passandomi accanto, ha inflitto dolore alla mia esistenza. Ciò che non uccide fortifica.

Ringrazio me stesso poiché nulla mi spaventa.
E' già ora di andare, è tempo di partenza...

Glossary

ABC: Absorbing Boundary Conditions

API: Application Programming Interface

CPU: Central Processing Unit

DFT: Discrete Fourier Transform

FCC: Face-Centered Cubic

f.f. : filling factor

FDTD: Finite-Difference Time-Domain

HPC: High Performance Computing

IFT: Inverse Fourier Transform

ILT: Inverse Laplace Transform

LSP: Localized Surface Plasmon

LSPR Localized Surface Plasmon Resonance

MPI: Message Passing Interface

PDE: Partial Differential Equation

PC: Photonic Crystal

RBC: Radiation Boudary Conditions

RC: Recursive Convolution

SEM: Scanning Electron Microscope

SPMD: Single-Program Multiple-Data

SPP: Surface Plasmon Polariton

SMP: Symmetrical MultiProcessor

TFSF: Total Field/Scattered Field

1D: one-dimensional

2D: two-dimensional

3D: three-dimensional

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 The FDTD method for electromagnetics

Simulation allows us to realize a virtual laboratory, it stays in fact in the middle between the theoretical and the experimental approach. Simulation is the only way to investigate a large quantity of physical problems. For example when we want to do a measure related to an object where it is impossible to go inside. In electromagnetics the numerical simulation is needed for solving almost all the problems of the real life, for designing new devices, measure the electromagnetic field inside the objects over which an electromagnetic radiation impings, evaluate the scattered field, calculate the radiation pattern of a radiating structure, simulate the electromagnetic propagation in a complex environment, etc. The analytical solution indeed is not available for complex problems. The *Finite-Difference Time-Domain* (FDTD) method is the numerical technique we deal with. The early Yee formulation [1] of the FDTD method [2] and its successive enrichments, such as the pulsed FDTD method, the absorbing (ABC) or radiation (RBC) boundary conditions for appropriate mesh truncation, are a powerful tool in computational electromagnetics [3]. FDTD is a fully explicit numerical solution method of the hyperbolic first order Maxwell's curl PDEs system. The computational domain is discretized by cubic Yee cells [1] (Fig. 1.1) and all the spatial values of the electromagnetic field components are updated step by step. The electric field is calculated basing on the values of the fields at the previous instants in time, then the magnetic field is calculated basing on the previous values of the fields, etc. The temporal evolution of the fields is then obtained. A frequency domain analysis is not precluded, because a *Discrete Fourier Transform* (DFT) can be performed in-line. With a single run of the code it is possible to obtain the frequency behavior of the system. In this thesis we deal with the electromagnetic simulation of complex problems in the field of plasmonics and photonics. These problems require a certain amount of memory, a detailed description at the nano-scale, the study of dispersive media, etc. The absorbing boundary conditions are those described in [4, 5]. A parallel approach is used for the code development in order to reduce the simulation time using *High Performance Computing* (HPC) systems. The

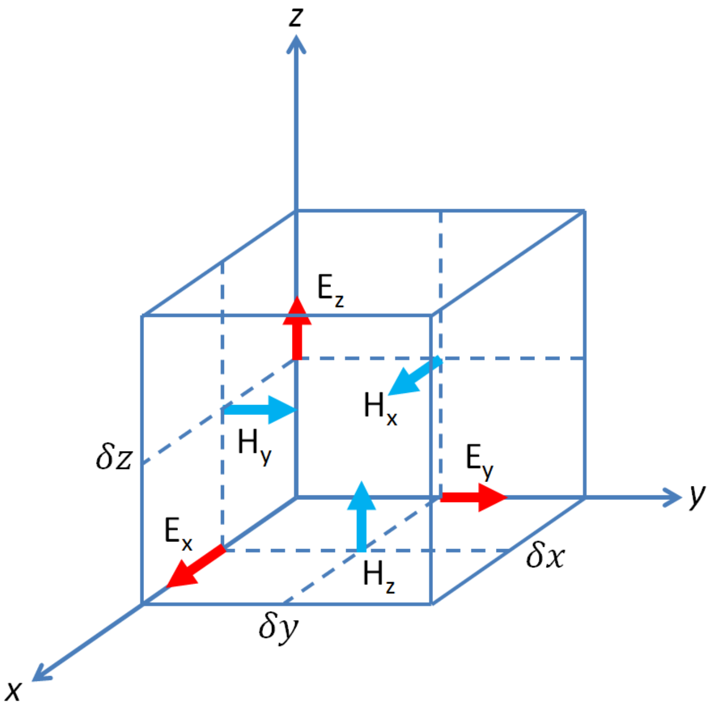


Figure 1.1: Yee FDTD cell.

dispersion in the optical frequency range of noble metals is also treated in order to simulate in the more accurate way as possible the plasmonic resonances. Plasmonics in particular is applied to the photovoltaics in order to increase the photon concentration within the solar cell. The problems of the FDTD simulation are also treated in the case of opal photonic crystals for simulation and design purposes.

1.2 The simulation of dispersive media

In the field of nanotechnology the noble metal nanoparticles have an important role. In the optical range noble metals act as plasma and surface waves appear along the metal-dielectric interface. These waves originate collective oscillations of the electrons and are called Localized Surface Plasmons (LSP). The main phenomena are the field confinement, the field enhancement, the scattering cross section increase. They are maximized at the resonance condition, known as Localized Surface Plasmon Resonance (LSPR), which has also the property of the tunability. The plasmonic behavior appearing in the light spectrum region can be used in optoelectronics devices and photonic applications. These phenomena are important also for the photovoltaic efficiency improvement. The dispersive behavior of the noble metal has to be modeled in an accurate way in order to have a detailed description of the plasmonic resonances. We focus at the beginning on the Drude model for describing the noble metals dispersion. Although some other approaches for the simulation of Drude dispersive media already exist [6–17], we consider the *Recursive Convolution* (RC) algorithm [18], that we call standard, as reference. We propose then a modified algorithm for improving the accuracy at the plasmonic resonance frequencies. Our approach time-discretizes directly the convolution integral expressing the temporal non-locality between the $\boldsymbol{D}$ and $\boldsymbol{E}$ fields. In order to minimize the truncation error, we propose here to find a closed form solution of the Ampère-Maxwell equation, and only after to proceed with the time discretization. We calculate explicitly the kernel of such a closed form solution in the case of Drude media deducing the modified RC algorithm, and show how it can be updated recursively with the same memory requirements than the standard RC scheme. The evaluation of some error parameters and the comparison of the electromagnetic fields highlight the better accuracy of the proposed modified RC algorithm with respect to the standard one at the plasmonic resonances. The test has been done for Au and Ag noble metals [19] nanospheres, comparing the numerical solution with the analytical one [20] in the optical frequency range. The modified RC algorithm appears suitable for the simulation of plasmonic resonant nanostructures and optical antennas. The modified approach was then extended from a theoretical point of view to the case of the Drude model with the two critical points correction [21–23] and to the case of the Lorentz model [24]. We used the *Convolutional Perfectly Matched Layer* (CPML) boundary conditions described in [4, 5]

1.3 The parallelization for large simulations

At the nano-scale the simulation parameters as the space-step used for the domain discretization determines a memory requirement very high for being supported by a single calculator. Due to the large sizes of the models, a serial encoding of the FDTD method may result in long calculation time. The increase of the computational efficiency is then necessary. The need to solve complex and large problems has been dealt with a subgridding approach besides a code parallelization. The simulation time is then reduced and more realistic scenarios can be studied. We start on the ground of a subgridding algorithm for the three-dimensional (3D) FDTD explicit numerical solution method of the Maxwell's Equations, proposed in a preceding paper [25]. This algorithm is efficient, non-recursive, and incorporates a spatial filtering technique of the numerical signals and was described using a serial programming approach. A parallel code version of the same algorithm is here proposed, for very large size FDTD calculations. The algorithm enables the embedding of finer meshes into coarse ones with a reduction of the space step by factors of 3, 5, 7 or greater, maintaining also a good accuracy and the long term stability. The basic assumption of the subgridding (or mesh refinement) algorithm is that increased mesh densities are introduced only in sub-regions where an accurate discretized description of the geometrical surface details are really needed [26–29] or subregions where the wavelength is shortened by the presence of high permittivity media. It is assumed that such subregions are embedded in homogeneous, isotropic, air-like regions where the accuracy criterion is easily verified with a larger space step. The subgridding is thus confined to volumes of minimum possible extent. For a given coarse/fine ratio (which we assume the best from the viewpoint of the geometrical/electrical constraints), the shortest simulation time is achieved with an accurate load balancing obtained by an appropriate domain decomposition (slicing). The execution time depends also on the cluster network bandpass, due to the data exchange between processes executing contiguous domain portions. Besides the subgridding for accuracy purposes another step is to increase the computational efficiency running the code on HPC systems, i.e. parallel or massive parallel machines with large RAM and storage resources. The parallelization is made by means of an *Message Passing Interface* (MPI) *Application Programming Interface* (API) implementation. To keep the parallel part of the code at the speed level of the bulk FDTD calculations, suitable user defined MPI data types were introduced, along with an adequate domain decomposition of the coarse and refined meshes and their interfaces organization. MPI realizes the *Single-Program Multiple-Data* (SPMD) programming approach: many instances of a single program, the processes, are executed autonomously on distinct physical processors of a cluster. This is a collection of interconnected SMP (Symmetrical MultiProcessor) *nodes*, each of them with a certain number of Central Processing Unit (CPU). Previous works on FDTD code parallelization by means of the MPI Library are reported in [30–33]. These papers describe 2D or 3D Cartesian topologies of processes for the FDTD grid decomposition, along with user defined MPI data structures for the data exchange at the subdomain boundaries. They also report some performance analysis. We start giving some details on how the MPI parallel version of our subgridding algorithm [25] is implemented, in particular it is described how to dynamically allocate the data structures in memory in order to have a fast access and an efficient management of the data transition at the coarse/coarse, refined/refined and coarse/refined domain interfaces. According to the intrinsic modularity of this algorithm, all the code is object-oriented C++ [34], which makes easier its understanding, development and maintenance. Our domain decomposition is through slices, i.e. 1D. We analyze quantitatively the performances of our parallel code applying it to some FDTD test configurations involving the electromagnetic field human exposure in a complex propagation environment. The code can be adapted to the study of plasmonic nanostructures and opal photonic crystals as shown in the next sections. We have assumed cubic uniform grid cell shapes everywhere. The absorbing boundary conditions are those described in [35].

1.4 Simulation of plasmonic nanostructures

In the last years there has been a growing interest in the application of plasmonic light-trapping techniques to photovoltaic cells [36–42]. The main inefficiency causes due to the photon absorption are that:

- not all the incident photons go deep into the lattice, some photons are reflected at the surface. The use of plasmonic nanoparticles can be an alternative method to the simple anti-reflective coating.
- the photon energy is not completely used for the electron-hole pairs creation and a fraction is lost in thermal phenomena.
- not all the electron-hole pairs contribute to the electric current because of the recombination.

The presence of metal nanostructures on the front and rear surfaces or in the bulk of a cell is a promising tool to enhance the efficiency through surface plasmon resonances, especially in thin film (e.g. a-Si:H) solar cells. Plasmon resonances from metal nanoparticles is a wide and popular sector of investigations, with many applications [43–45] also in other fields. In solar cell devices light can be converted to electricity using the plasmon resonances of the noble metal nanostructures in three ways [46]:

- nanoparticles can be used as subwavelenght scattering elements that couple and trap within the solar device the propagating plane waves through a far-field effect that extends the optical path length of the radiation into the cell (Fig. 1.2).
- nanoparticles can be used as subwavelenght antennas in which the plasmonic near-field is coupled to the semiconductor increasing its effective cross-section (Fig. 1.3). The result is a local absorption increase due to the local field enhancement.

- a corrugated metallic film on the back surface of a thin photovoltaic absorber layer can couple sunlight into surface plasmon polaritons (SPP) modes supported at the metal-semiconductor interface as well as guided modes in the semiconductor slab (Fig. 1.4).

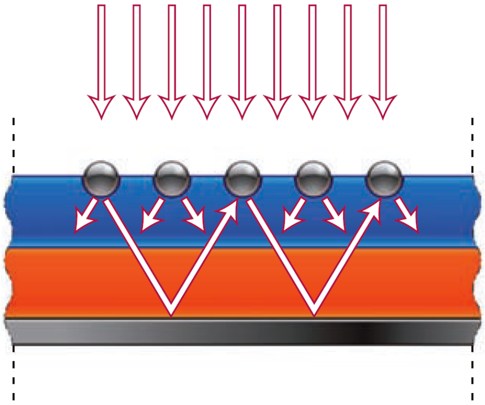


Figure 1.2: Light trapping by scattering from metal nanoparticles at the surface of the solar cell.

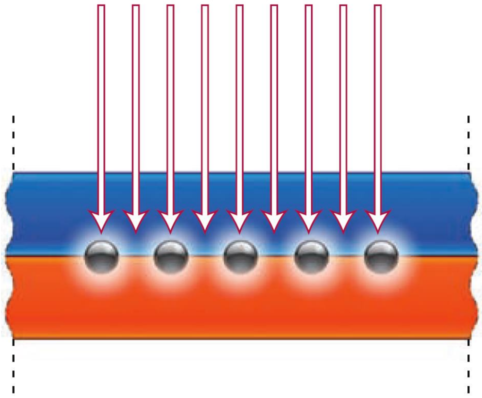


Figure 1.3: Light trapping by the excitation of localized surface plasmons in metal nanoparticles embedded in the semiconductor.

Among the different physical mechanisms leading to better performing photovoltaic cells, we discuss here the enhancement in the optical path

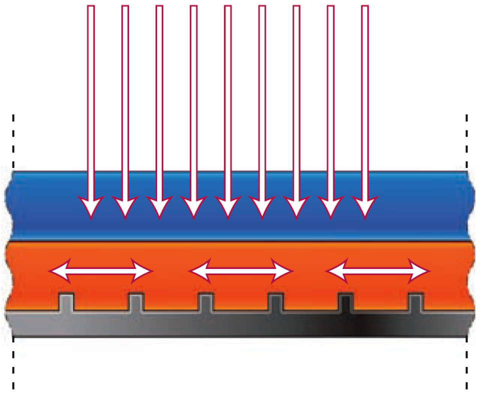


Figure 1.4: Light trapping by the excitation of surface plasmon polaritons at the metal-semiconductor interface.

length due to the far-field scattering from nanoparticles of different size, shapes and material. Although this method has been experimentally confirmed many times [47–51], a theoretical treatment of the topic, with comprehensive comparisons among the possible choices of the parameters is still required. So, it is not completely clear how each parameter influences the efficiency, how they interplay and which combination determines the optimal cell design. Independently from the type of solar cell, metal nanoparticles should satisfy two general conditions in order to improve the efficiency in a significant way:

- their scattering cross section should be as large as possible, and, at the same time, their absorption cross section should be as small as possible;

- they should scatter most of the radiation into the cell, with the largest angular spread to maximize the optical path length.

To achieve these features, we discuss the case of a single particle in a homogeneous, infinite dielectric medium as a function of several parameters, namely the shape and size of the particles, the material they are composed of, and the optical characteristics of the surrounding environment. Since the number of parameters is very large, it is important to treat the scattering analytically whenever possible, before starting time-expensive numerical simulations or cell fabrication. The scattering problem is exactly solvable only for few particle shapes, as ellipsoid (spheres [52], in particular) and infinitely long cylinders. We discuss here the equations governing the extinction mechanism by a metallic sphere in order to investigate the scattering and the absorption properties, and study the effects of changing the material, the dimensions and the media which embed the spheres. The analytical discussion can go so far and cannot extend to other shapes. Numerical simulations fill this gap; in particular the FDTD method is used to simulate the evolution of the electromagnetic field during the extinction process. The method will be applied to study the response of hemispheres, cylinders and spheroids of various size. An analysis of the angular distribution of the scattered radiation is also included.

1.5 Simulation of opal photonic crystals

Another complex problem dealt with a simulation approach is the analysis and the characterization of opal photonic crystals [53–55] (PC) is described. These crystals are modeled by an ordered face-centered cubic (FCC) lattice structure of spherical particles with a given filling factor ($f.f.$). The filling factor is the ratio between the filled space by the nanospheres and the total space. It can be evaluated over a single FCC cell. These particles are made of a non-absorptive and non-dispersive material with a constant dielectric permittivity. The physical phenomenon to investigate is the multiple scattering by the spheres of an impinging monochromatic light beam, which is obtained in laboratory through a monochromator. The modeling of the exciting signal is based on a plane wave electromagnetic pulse and the interaction with the opal PC is simulated through the numerical solution of the Maxwell's Equations using the FDTD method [3]. The resulting field is calculated in a discretized fashion on a three dimensional grid of sampling points. Frequency Domain results are obtained by means of a DFT of the system response to the transient excitation due to the impinging pulse. The simulation target is the opal PC reflectance and transmittance calculation in the $300 \div 700$ nm wavelength range. The scattering by the spheres originates a non-omnidirectional pattern for the transmitted and the reflected light. Through an analytical approach (Kirchhoff's formula [56]) the angular distribution of the scattered field in the far-zone is evaluated and the total scattered power calculated. The structure, due to the nanoscale and to the fine space-discretization needed for accuracy reasons, is cumbersome and challenging from a simulation point of view and a parallel approach is required for that. By using the MPI API the simulation can run on HPC systems. A cubic slicing approach is used for the decomposition of the analysis domain into blocks. MPI allows the field components exchange between processes running adjacent blocks. The computational result for the transmittance as function of the wavelength in the $300 \div 700$ nm range reveals the presence of a band-gap in the curve. The comparison with the experimental measures carried out in our laboratory shows a satisfactory agreement. Although some modeling approximation has to be done the proposed numerical technique seems appear as a promising tool for the theoretical characterization of opal PCs.

1.6 Summary of publications

This thesis is based on the following publications.

- A. Vaccari, A. Calà Lesina, L. Cristoforetti, and R. Pontalti: *Parallel implementation of a 3D subgridding FDTD algorithm for large simulations*, Progress In Electromagnetics Research, Vol. 120, 263-292, 2011.
  `http://www.jpier.org/PIER/pier.php?paper=11063004`

- A. Paris, A. Vaccari, A. Calà Lesina, E. Serra, and L. Calliari: *Plasmonic scattering by metal nanoparticles for solar cells*, Plasmonics, Vol. 7, no. 3, pp. 525-534, 2012.
  `http://link.springer.com/article/10.1007%2Fs11468-012-9338-4?LI=true`

- A. Calà Lesina, A. Vaccari, and A. Bozzoli: *A modified RC-FDTD algorithm for plasmonics in Drude dispersive media nanostructures*, 6th European Conference on Antennas and Propagation (EuCAP), Prague, Czech Republic, 2012.
  `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6206111`

- A. Calà Lesina, A. Vaccari, and A. Bozzoli, *A novel RC-FDTD algorithm for the Drude dispersion analysis*, Progress In Electromagnetics Research M, Vol. 24, 251-264, 2012.
  `http://www.jpier.org/PIERM/pier.php?paper=12041904`

- A. Calà Lesina, A. Vaccari, and A. Bozzoli, *A modified RC-FDTD algorithm for plasmonics in Drude dispersive media*, Proceedings of the XIX RiNEm (Riunione Nazionale di Elettromagnetismo), Roma, Italy, 2012.

- A. Vaccari, A. Calà Lesina, L. Cristoforetti, A. Chiappini, and M. Ferrari, *A computational approach to the optical characterization of photonic crystals and photonic glasses*, Proceedings of the XIX RiNEm (Riunione Nazionale di Elettromagnetismo), Roma, Italy, 2012.

- A. Chiappini, A. Armellini, A. Calà Lesina, A. Vaccari, L. Cristoforetti, F. Prudenzano, G. C. Righini, and M. Ferrari, *Colloidal plasmonic photonic crystals*, 1st International workshop on Metallic Nano-objects, Sant-Etienne, France, 2012.

- A. Vaccari, A. Calà Lesina, L. Cristoforetti, A. Chiappini, F. Prudenzano, A. Bozzoli, M. Ferrari, *A parallel computational FDTD approach to the analysis of the light scattering from an opal photonic crystal*, Proceedings of SPIE 8781, Integrated Optics: Physics and Simulations, p. 87810, Prague, Czech Republic, 2013.

- A. Vaccari, A. Calà Lesina, L. Cristoforetti, A. Chiappini, F. Prudenzano, A. Bozzoli, and M. Ferrari, *A parallel FDTD computation of the photonic crystal transmittance*, 15th National Conference on Photonic Technologies (Fotonica 2013), Milano, Italy, 2013.

Chapter 2

The simulation of dispersive media

2.1 Theoretical Approach

The temporal non-locality relation between the $\boldsymbol{D}$ and $\boldsymbol{E}$ fields in dispersive media is expressed by means of the convolution integral

$$\boldsymbol{D}(t) = \epsilon_0 \epsilon_\infty \boldsymbol{E}(t) + \epsilon_0 \int_0^t \boldsymbol{E}(t-\tau)\chi(\tau) d\tau \tag{2.1}$$

which exhibits a Dirac-delta contribution representing the instantaneous response at the infinite frequency through the (relative) permittivity ϵ_∞ term. In (2.1) $\chi(\tau)$ is the Inverse Fourier Transform (IFT) of the electric susceptibility $\chi(\omega)$. This measures the media polarization and enters the complex permittivity $\epsilon(\omega)$ in

$$\boldsymbol{D}(\omega) = \epsilon_0 \left[\epsilon_\infty + \chi(\omega)\right] \boldsymbol{E}(\omega) = \epsilon(\omega)\boldsymbol{E}(\omega), \tag{2.2}$$

i.e. the proportionality coefficient between $\boldsymbol{D}$ and $\boldsymbol{E}$ in the angular frequency domain ω after a Fourier transform with respect to the time variable. In (2.2) the same letters are used to denote the fields both in the time and frequency domain and the space dependence is understood. The Ampère-Maxwell equation which is time stepped in the FDTD method, along with the Faraday-Maxwell curl equation for the $\boldsymbol{E}$ and $\boldsymbol{B}$ fields, is

$$\boldsymbol{\nabla} \times \boldsymbol{H} = \frac{\partial \boldsymbol{D}}{\partial t} + \sigma \boldsymbol{E}, \tag{2.3}$$

where σ is the static conductivity contribution. Before discretizing (2.3) for time stepping however, we analytically solve it with respect to the time variable by the Laplace transform method, to get a closed form solution for the electric field $\boldsymbol{E}$ at a given time instant t. If we denote by s the dual of the time variable t, omit an understood space dependence as before, and now use a tilde for a Laplace transformed quantity, we have from (2.1)

$$\boldsymbol{D}(s) = \epsilon_0 \left[\epsilon_\infty + \chi(s)\right] \boldsymbol{E}(s) \tag{2.4}$$

and from (2.3)

$$\boldsymbol{\nabla} \times \boldsymbol{H}(s) = s\boldsymbol{D}(s) - \boldsymbol{D}(0) + \sigma \boldsymbol{E}(s), \tag{2.5}$$

where we assumed that the curl operator $\boldsymbol{\nabla}\times$, acting on the space variables, commutes with the Laplace transform operator. By solving for $\boldsymbol{D}(s)$ the first of the two previous equations, inserting the result in the second one, where an initial time zero-field condition has been assumed, and then solving for $\boldsymbol{E}(s)$ we have

$$\boldsymbol{E}(s) = G(s) \cdot \boldsymbol{\nabla} \times \boldsymbol{H}(s), \tag{2.6}$$

where $G(s)$ stands for

$$G(s) = \frac{1}{s\epsilon_0 \left[\epsilon_\infty + \chi(s)\right] + \sigma} \quad . \tag{2.7}$$

Note that $\chi(s = -i\omega) = \chi(\omega)$ where $i = \sqrt{-1}$ is the imaginary unit. By returning to the time domain through an Inverse Laplace Transform (ILT), we get the aforementioned closed form exact solution for the electric field

$$\boldsymbol{E}(t) = \int_0^t G(t-\tau) \cdot \boldsymbol{\nabla} \times \boldsymbol{H}(\tau) d\tau, \tag{2.8}$$

where the convolutional kernel $G(\tau)$ depends on the medium dispersion characteristic. Note that if in (2.1) $\chi(\tau)$ were identically zero, we would recover the usual non-dispersive behavior with the absolute dielectric constant $\epsilon = \epsilon_0\epsilon_\infty$. This would imply an identically zero χ in (2.7) too. For the corresponding original G we would then get

$$G(\tau) = \frac{1}{\epsilon} e^{-\frac{\sigma}{\epsilon}\tau}, \tag{2.9}$$

where a Heaviside step function factor of argument is understood. Using this result in (2.8) and, as is usual in FDTD, sampling at discrete times $n\delta t$, where δt is the time step, with $\boldsymbol{\nabla} \times \boldsymbol{H}$ and $\boldsymbol{H}$ temporally sampled halfway at $(n + 1/2)\delta t$, one gets an updating equation for $\boldsymbol{E}$ with exponential coefficients

$$\boldsymbol{E}^{n+1} = e^{-\frac{\sigma\delta t}{\epsilon}} \boldsymbol{E}^n + \frac{(1 - e^{-\frac{\sigma\delta t}{\epsilon}})}{\sigma} \boldsymbol{\nabla} \times \boldsymbol{H}^{n+\frac{1}{2}}, \tag{2.10}$$

where superscripts denote time levels. By Taylor expanding to first order the coefficients in (2.10) with respect to the small quantity $\sigma\delta t/\epsilon$, they equal their FDTD discrete counterparts expanded to the same order. With this in mind we think that our approach based on (2.8) is less prone to time truncation errors, mainly for highly absorptive media with rapidly time decaying fields, than the standard method based on an early discretization of the convolution integral (2.1).

2.2 The Drude model

We now calculate explicitly the convolutional kernel $G(s)$ shown in (2.7) in the case of Drude dispersion. This is formulated by the single term electric susceptibility

$$\chi(s) = \frac{\omega_D^2}{s(s+\gamma)}, \tag{2.11}$$

where ω_D and γ are the plasma frequency and the damping coefficient. This gives

$$G(s) = \frac{s+\gamma}{\epsilon_0\epsilon_\infty(s - s_+)(s - s_-)}, \tag{2.12}$$

where

$$s_{\pm} = -P \pm iQ \tag{2.13}$$

and

$$P = \frac{1}{2}\left(\gamma + \frac{\sigma}{\epsilon_0\epsilon_\infty}\right), \quad Q = \sqrt{\frac{\epsilon_0\omega_D^2 + \sigma\gamma}{\epsilon_0\epsilon_\infty} - P^2} \quad . \tag{2.14}$$

After returning to the time-domain we get

$$G(\tau) = \frac{1}{\epsilon_0\epsilon_\infty}\left[\frac{e^{s_+\tau}(s_+ + \gamma)}{s_+ - s_-} + \frac{e^{s_-\tau}(s_- + \gamma)}{s_- - s_+}\right] \tag{2.15}$$

which, putting

$$S = \frac{1}{2}(\gamma - \frac{\sigma}{\epsilon_0\epsilon_\infty}), \tag{2.16}$$

has the following form

$$G(\tau) = Im\left\{Ke^{-W\tau + i\Phi}\right\}, \tag{2.17}$$

where

$$K = \frac{1}{\epsilon_0\epsilon_\infty}\sqrt{1 + \left(\frac{S}{Q}\right)^2}, \tag{2.18}$$

$$W = P - iQ, \tag{2.19}$$

$$\Phi = \arctan\frac{Q}{S}, \tag{2.20}$$

with K and Φ real quantities. Re and Im denote the real and imaginary parts of a complex quantity. By defining the complex vector

$$\boldsymbol{\Psi}(t) = \int_0^t Ke^{-W(t-\tau)+i\Phi} \cdot \boldsymbol{\nabla} \times \boldsymbol{H}(\tau)d\tau \quad . \tag{2.21}$$

one sees that, according with (2.8) and (2.17), the electric field $\boldsymbol{E}$ results to be

$$\boldsymbol{E}(t) = Im\left\{\boldsymbol{\Psi}(t)\right\} = \int_0^t Im\left\{Ke^{-W(t-\tau)+i\Phi}\right\} \cdot \boldsymbol{\nabla} \times \boldsymbol{H}(\tau)d\tau. \tag{2.22}$$

Sampling (2.21) at the discrete times $n\delta t$ $(n = 1, 2, ...)$ we have

$$\boldsymbol{\Psi}^{n+1} = \int_0^{(n+1)\delta t} Ke^{-W((n+1)\delta t-\tau)+i\Phi} \cdot \boldsymbol{\nabla} \times \boldsymbol{H}(\tau) d\tau. \tag{2.23}$$

Separating the integration interval we obtain

$$\begin{aligned} \boldsymbol{\Psi}^{n+1} = e^{-W\delta t} \int_0^{n\delta t} Ke^{-W(n\delta t-\tau)+i\Phi} \cdot \boldsymbol{\nabla} \times \boldsymbol{H}(\tau) d\tau + \\ + \boldsymbol{\nabla} \times \boldsymbol{H}^{n+\frac{1}{2}} \cdot \int_{n\delta t}^{(n+1)\delta t} Ke^{-W((n+1)\delta t-\tau)+i\Phi} d\tau \end{aligned} \tag{2.24}$$

and then

$$\boldsymbol{\Psi}^{n+1} = e^{-W\delta t} \cdot \boldsymbol{\Psi}^n + A \cdot \boldsymbol{\nabla} \times \boldsymbol{H}^{n+\frac{1}{2}}, \tag{2.25}$$

where the complex coefficient A is given by

$$A = \int_{n\delta t}^{(n+1)\delta t} Ke^{-W((n+1)\delta t-\tau)+i\Phi} d\tau = K\frac{e^{i\Phi}(1 - e^{-W\delta t})}{W}. \tag{2.26}$$

Thus storing, as in the RC traditional scheme [18], one extra complex variable for each sampling point and each electric field component, updating it according to (2.25), and using its imaginary part as a new electric field value, allows us to include dispersive media in a simpler and more accurate recursive procedure. By expanding the exponential factor according to the Euler formula

$$e^{-W\delta t} = e^{-P\delta t} [\cos(Q\delta t) + i \sin(Q\delta t)] \tag{2.27}$$

and taking the imaginary part of both sides of (2.25) we have the modified form of the electric field updating equation

$$\begin{aligned} \boldsymbol{E}^{n+1} = e^{-P\delta t} \sin(Q\delta t) \cdot Re\{\boldsymbol{\Psi}^n\} + \\ + e^{-P\delta t} \cos(Q\delta t) \cdot \boldsymbol{E}^n + Im\{A\} \cdot \boldsymbol{\nabla} \times \boldsymbol{H}^{n+\frac{1}{2}}. \end{aligned} \tag{2.28}$$

For completeness we report the updating equation of the standard method [18]:

$$\boldsymbol{E}^{n+1} = C_1 \cdot \boldsymbol{\Phi}^n + C_2 \cdot \boldsymbol{E}^n + C_3 \cdot \boldsymbol{\nabla} \times \boldsymbol{H}^{n+\frac{1}{2}}, \tag{2.29}$$

where

$$\mathbf{\Phi}^n = C_4 \cdot \boldsymbol{E}^{n-1} + e^{-\gamma\delta t} \cdot \mathbf{\Phi}^{n-1} \tag{2.30}$$

and

$$C_1 = \frac{1}{\epsilon_0 + \chi_0}, \tag{2.31}$$

$$C_2 = (\epsilon_\infty + \Delta\chi_0)C_1, \tag{2.32}$$

$$C_3 = \frac{\delta t}{\epsilon_0}C_1, \tag{2.33}$$

$$C_4 = e^{-\gamma\delta t}\Delta\chi_0, \tag{2.34}$$

$$\chi_0 = \int_0^{\delta t} \chi(\tau)d\tau, \tag{2.35}$$

$$\Delta\chi_0 = -\frac{\omega_D^2}{\gamma}[1 - e^{-\gamma\delta t}]^2. \tag{2.36}$$

2.2.1 Simulations

To test the modified RC algorithm previously proposed, we apply it to a $96nm$ radius nanosphere, made of gold or silver, in a plane wave light beam. The static conductivity is assumed null. We have tested our modified algorithm in particular for Au at $\lambda = 480nm$ and for Ag at $\lambda = 336nm$ and $\lambda = 380nm$, i.e. the resonance wavelengths evidenced in Fig. 2.1 through the extinction coefficient Q_{ext} defined in [56]. This coefficient is an efficiency parameter defined as the particle cross section C_{ext} (effective surface) reported in (4.1) normalized to the geometrical projection of the particle on a plane perpendicular to the incoming field (real surface). The extinction, scattering and absorption coefficients for a small spherical particle [20] are:

$$Q_{ext} = \frac{2}{r^2k^2}\sum_{n=1}^{\infty}(2n+1)Re(a_n + b_n), \tag{2.37}$$

$$Q_{sca} = \frac{2}{r^2k^2}\sum_{n=1}^{\infty}(2n+1)(|a_n|^2 + |b_n|^2), \tag{2.38}$$

$$Q_{abs} = Q_{ext} - Q_{sca}, \tag{2.39}$$

where $k = (2\pi N)/\lambda$, N is the refractive index of the medium surrounding the sphere, λ is the wavelength of the incident radiation, a_n and b_n are combinations of Riccati-Bessel functions [20]. These functions are explicitly dependent on the radius r of the sphere and on the complex permittivity of the medium. We used a $200 \times 200 \times 200$ cubic Yee cell discretization to accommodate the nanosphere. The cell edge (space step $\delta = \delta x = \delta y = \delta z$) amounts to $2nm$ for a good representation of the geometrical details. The time step was set to $\delta/(2c_0)$, with c_0 the vacuum light velocity, to satisfy the Courant stability condition [3] in three dimensions. We also used a Total Field/Scattered Field (TFSF) source [3], placed 8 cells inward from the outer boundary of the FDTD lattice, to create a plane wave linearly polarized (along the z-axis), impinging along the positive y-direction on the nanostructure. The FDTD lattice was completed with an extra layer, 15 cells thick, supporting the CPML boundary conditions [4] to simulate an open to infinity surrounding media. We used the CPML parameters reported in [5]. We used a compact pulse exciting signal, i.e. of finite duration and with zero value outside a given time interval [25] [57]. The signal duration $T = 1/f_{max}$ is suitably chosen to get spectral distribution results in the range $200 \div 1000nm$, where f_{max} is the maximum frequency, as obtained by the DFT which is updated at every FDTD time iteration, until the excitation is extinguished inside the whole numerical lattice. The excitation signal is a raised cosine

$$s(t) = (1 - \cos(\frac{2\pi t}{T}))^3, \tag{2.40}$$

where T is the duration of the pulse. It can be considered extinguished in $3\div4$ the time the radiation needs for propagating along the lattice diagonal.

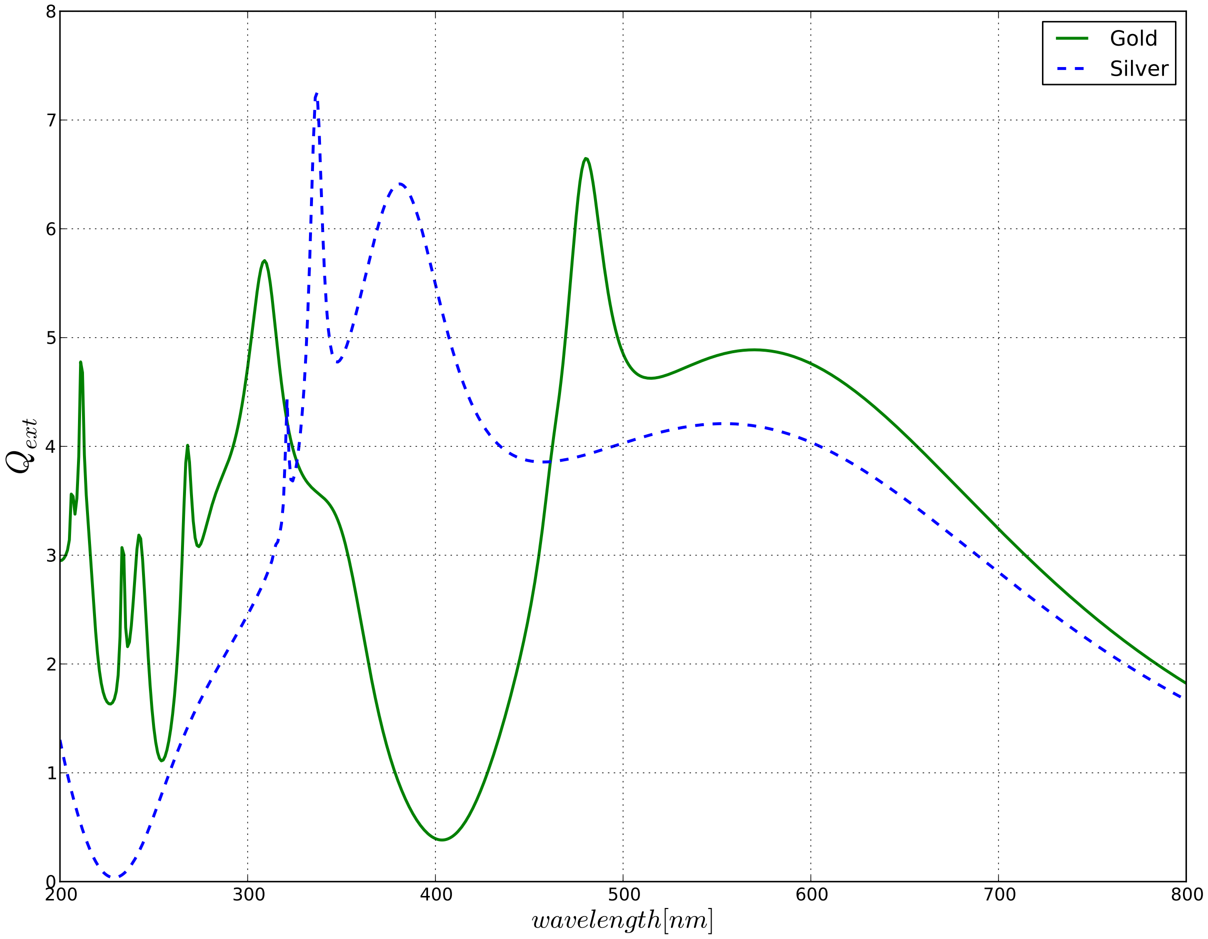


Figure 2.1: Analytical Q_{ext} for Au and Ag nanospheres ($r = 96nm$) modeled by Drude dispersion.

The Drude parameters for Au and Ag were taken from the literature [19] and are reported in Tab. 2.1.

	ϵ_∞	$\omega_D\,[rad/s]$	$\gamma\,[s^{-1}]$
Au	9.84	$1.3819 \cdot 10^{16}$	$1.09387 \cdot 10^{14}$
Ag	3.70	$1.3521 \cdot 10^{16}$	$3.19050 \cdot 10^{13}$

Table 2.1: Noble Metals Drude Parameters

	$Au(480nm)$	$Ag(336nm)$	$Ag(380nm)$
$L_{m,x}\ (L_{s,x})$	0.0386 (0.0548)	0.1275 (0.1533)	0.0374 (0.0421)
$L_{m,y}\ (L_{s,y})$	0.0592 (0.0872)	0.2085 (0.2485)	0.0585 (0.0687)
$L_{m,z}\ (L_{s,z})$	0.0440 (0.0633)	0.1718 (0.2024)	0.0430 (0.0478)
$L_m\ (L_s)$	0.0693 (0.1011)	0.2670 (0.3282)	0.0594 (0.0684)
$LQ_m\ (LQ_s)$	0.2216 (0.3376)	0.3265 (0.3646)	0.4579 (0.5052)

Table 2.2: Error Evaluation at the resonance frequencies

	Au	Ag
$LQ_m\ (LQ_s)$	0.0515 (0.0747)	0.0926 (0.1019)

Table 2.3: LQ_ξ over the total frequency range.

The electric fields, by means of the DFT, are computed for each sampling point of the lattice at the frequency of interest and are normalized with respect to the incident electric field component at the same frequency. The numerical results for the electric field and the extinction coefficient have been compared with those from the standard RC method [18] through the analytical solutions obtained by implementing the methods described in [56]. The numerical counterpart of Q_{ext} is calculated by adding the absorption coefficient Q_{abs} to the scattering coefficient Q_{sca}, indeed only these two are evaluable through a numerical approach. The first is the Poynting vector flux through a closed surface containing the sphere in the total field domain, the second is calculated by means of the same flux through a closed surface located in the scattered field region. In order to evaluate the deviation of the numerical results from the exact solution we considered the average error for each electric field component

$$L_{\xi,\eta} = \frac{1}{N^3} \sum_{i,j,k=1}^{N} \left| E_\eta^\xi(i,j,k) - E_\eta^a(i,j,k) \right| , \tag{2.41}$$

the average error for the electric field module

$$L_\xi = \frac{1}{N^3} \sum_{i,j,k=1}^{N} \left| |\boldsymbol{E}^\xi(i,j,k)| - |\boldsymbol{E}^a(i,j,k)| \right| \tag{2.42}$$

and the average error for C_{ext}

$$LQ_\xi = \frac{1}{N_\lambda} \sum_{i=1}^{N_\lambda} \left| Q_{ext_i}^\xi - Q_{ext_i}^a \right| , \tag{2.43}$$

where $\eta = \{x, y, z\}$ indicates the cartesian component, $\xi = \{s, m\}$, the letters s, m, a denote standard, modified and analytical solution, and N_λ is the number of wavelengths at which Q_{ext} has been evaluated. The values in Tab. 2.2 were obtained with 12000 time iterations simulations. For the gold resonance the error parameters (2.41) and (2.42) are reported as a function of the number of FDTD iterations (Fig. 2.2). We can observe that the convergence is reached after the same number of FDTD time iterations than in the standard case and in all the cases the level of accuracy is better. The better numerical accuracy is also evidenced with a comparison of the total electric field extracted from the lattice along one direction in x, y and z (Fig. 2.3, 2.4 and 2.5). In each figure the three components of the

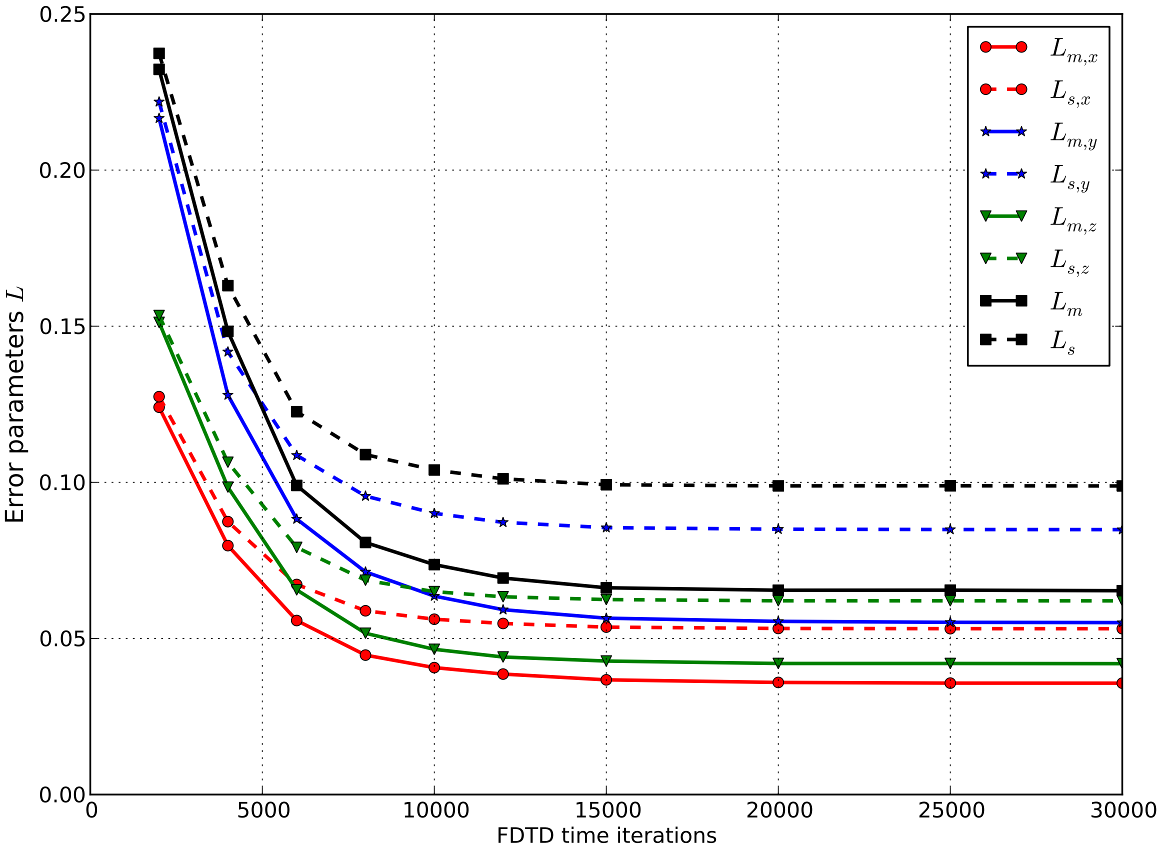


Figure 2.2: Error parameters $L_{\xi,\eta}$ and L_ξ comparison for Au (DFT at $\lambda = 480nm$).

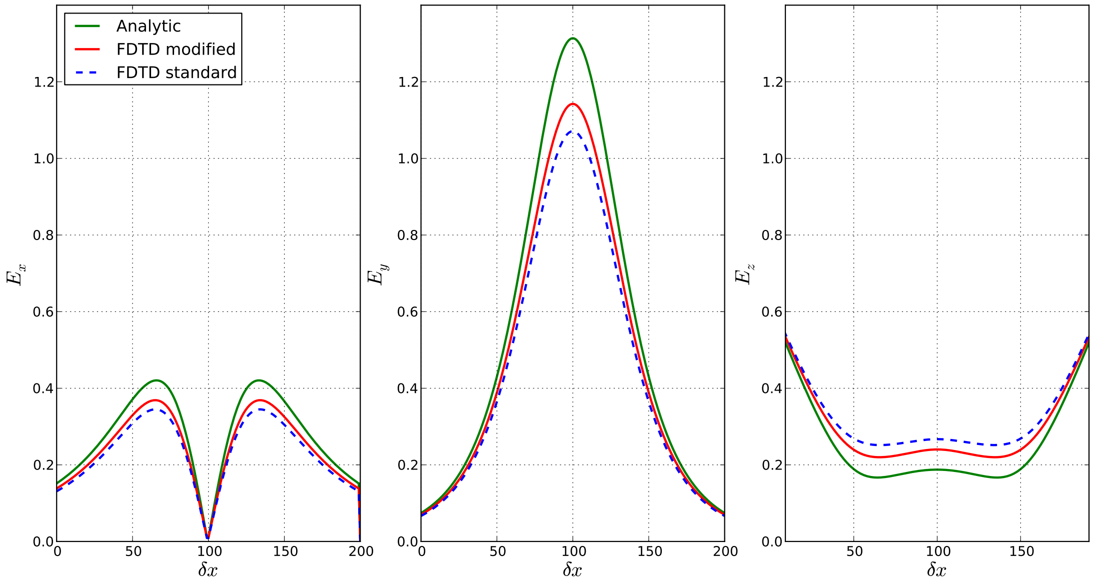


Figure 2.3: Total field E_x, E_y, E_z for Au nanosphere (DFT at $\lambda = 480nm$) along the x axis ($y = 170$, $z = 120$).

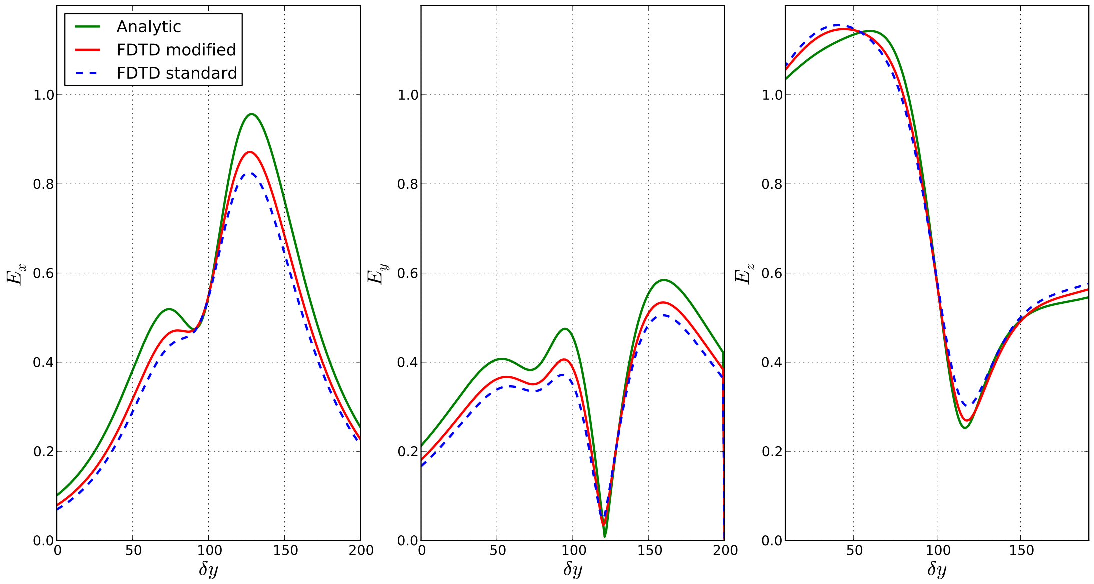


Figure 2.4: Total field E_x, E_y, E_z for Au nanosphere (DFT at $\lambda = 480nm$) along the y axis ($x = 150$, $z = 150$).

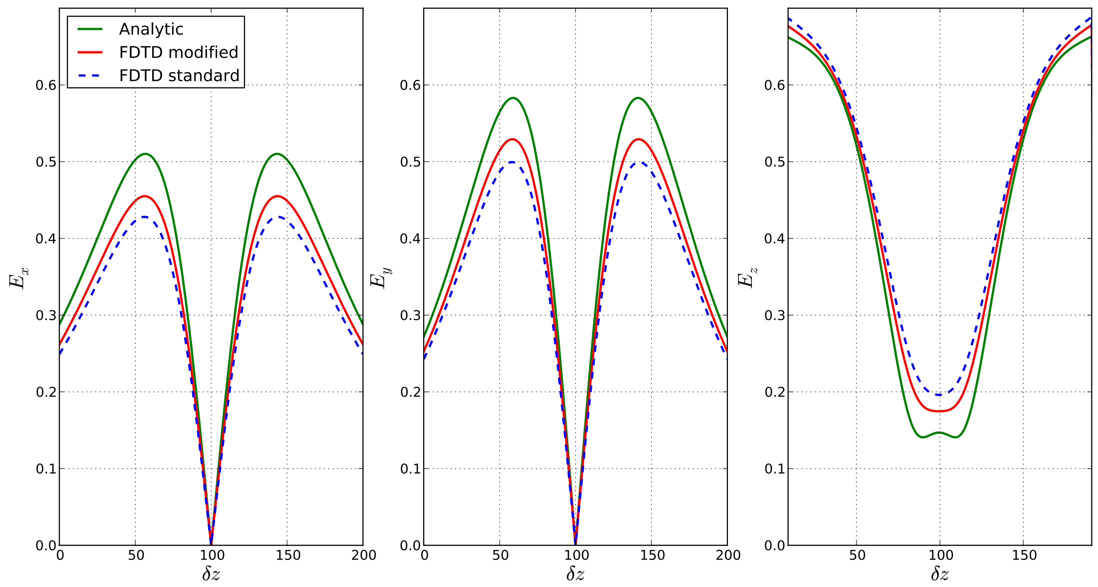


Figure 2.5: Total field E_x, E_y, E_z for Au nanosphere (DFT at $\lambda = 480nm$) along the z axis ($x = 150$, $y = 170$).

electric field (E_x on the left, E_y in the middle and E_z on the right) are represented. The sharp peaks are due to the field inside the sphere, that is very low (metallic sphere), and to some symmetry planes where the field is zero. For gold and silver the extinction coefficient has been calculated and the error parameter (2.43) at the resonance frequencies (Tab. 2.2) and over the total frequency range are reported (Tab. 2.3). For gold the error parameter (2.43) is evaluated for the total frequency range and in the peak regions ($\lambda = 400nm$ and $\lambda = 480nm$) varying the number of time iterations (Fig. 2.6), while the Q_{ext} comparison is shown in Fig. 2.7.

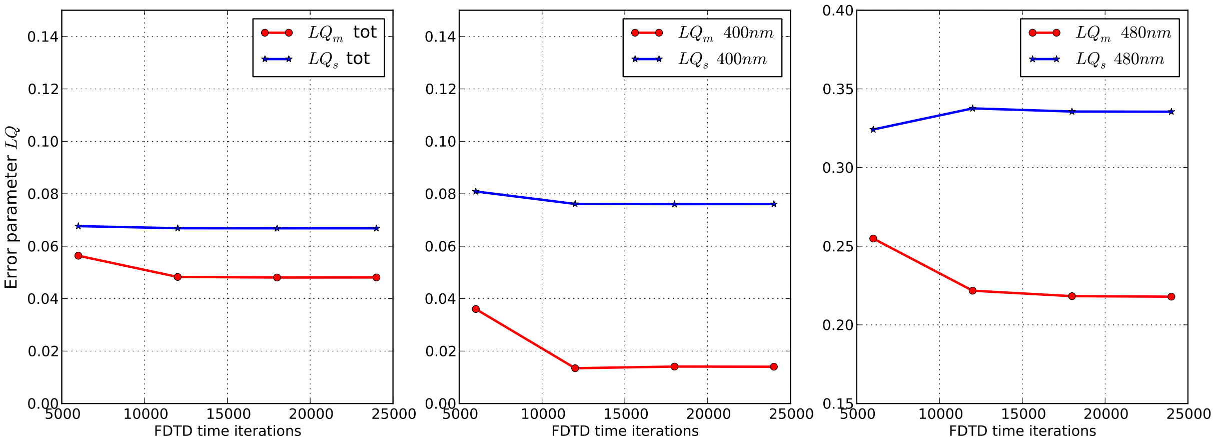


Figure 2.6: LQ_ξ comparison for gold in the total frequency range, and in the region of Q_{ext} minimum and maximum.

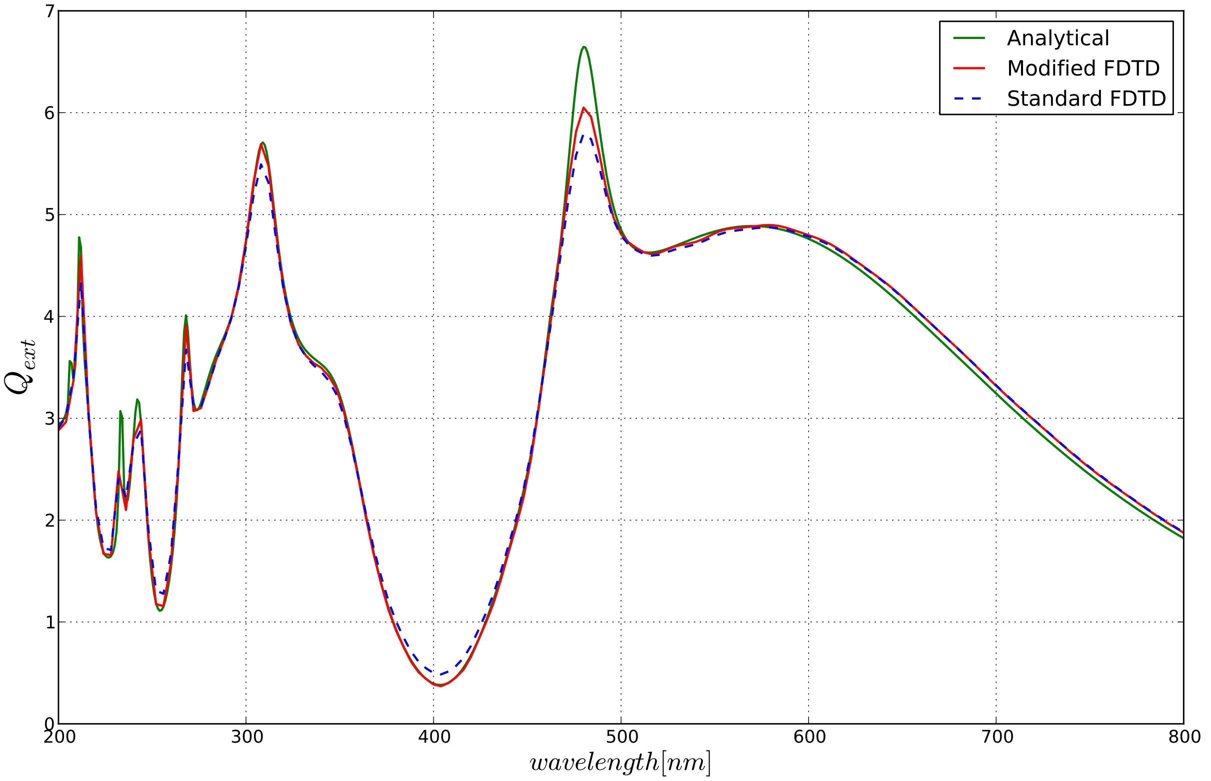


Figure 2.7: Q_{ext} for a gold nanosphere ($r = 96nm$) modeled by Drude dispersion.

2.3 The N-order Drude model

Let assume a dispersive medium described by a Drude multiple poles model. Let assume also a time dependence $e^{-i\omega t}$. If N is the order of the poles the complex susceptibility is

$$\chi(\omega) = \sum_{p=1}^{N} \frac{-\omega_p^2}{\omega(i\gamma_p + \omega)}, \tag{2.44}$$

where ω_p is the pth plasma frequency, γ_p is the pth damping coefficient and i the imaginary unit. In the Laplace domain ($s = -i\omega$) we have

$$\chi(s) = \sum_{p=1}^{N} \frac{\omega_p^2}{s(s + \gamma_p)}. \tag{2.45}$$

Substituting (2.45) into (2.7) we obtain

$$G(s) = \frac{\prod_{p=1}^{N}(s + \gamma_p)}{\sum_{k=0}^{N+1} C_k s^{N+1-k}}, \tag{2.46}$$

where $C_k = f_k(\epsilon_0, \epsilon_\infty, \sigma, \gamma_p, \omega_p)$ for $\{p = 1, ...N\}$ and $\{k = 0, ...N + 1\}$ are real coefficients, f_k is a kth function depending on the polynomial development. Let's distinguish two cases: even and odd N-order.

If N is odd the grade of the denominator of (2.46) is even. Developing it whit $\frac{N+1}{2}$ couples of complex conjugate poles we have

$$G(s) = \frac{\prod_{p=1}^{N}(s + \gamma_p)}{C_0 \prod_{k=1}^{\frac{N+1}{2}}(s + s_k)(s + s_k^*)}, \tag{2.47}$$

where $s_k = A_k + iB_k$ is the kth pole. Acting the ILT of (2.47) we obtain

$$G(t) = \sum_{k=1}^{\frac{N+1}{2}} e^{-A_k t} \left[M_k \cos(B_k t) + N_k \sin(B_k t) \right], \tag{2.48}$$

where M_k and N_k $\left\{k = 1, ...\frac{N+1}{2}\right\}$ are real coefficients that depend on γ_p $\{p = 1, ..N\}$, C_0, A_k and B_k $\left\{k = 1, ...\frac{N+1}{2}\right\}$. Setting

$$\frac{M_k}{N_k} = \tan \Theta_k = \frac{\sin \Theta_k}{\cos \Theta_k} \tag{2.49}$$

we obtain

$$G(t) = \sum_{k=1}^{\frac{N+1}{2}} \frac{N_k e^{-A_k t}}{\cos \Theta_k} \sin(B_k t + \Theta_k). \tag{2.50}$$

If we put

$$F(t) = \sum_{k=1}^{\frac{N+1}{2}} F_k(t), \tag{2.51}$$

where

$$F_k(t) = \frac{N_k e^{-A_k t}}{\cos \Theta_k} e^{i(B_k t + \Theta_k)} \tag{2.52}$$

and

$$\boldsymbol{\Psi}(t) = \int_0^t F(t - \tau) \cdot \boldsymbol{\nabla} \times \boldsymbol{H}(\tau) d\tau, \tag{2.53}$$

we have

$$\boldsymbol{E}(t) = \Im \left\{ \boldsymbol{\Psi}(t) \right\}. \tag{2.54}$$

Developing $\boldsymbol{\Psi}(t)$ we obtain

$$\boldsymbol{\Psi}(t) = \int_0^t \sum_{k=1}^{\frac{N+1}{2}} F_k(t - \tau) \cdot \boldsymbol{\nabla} \times \boldsymbol{H}(\tau) d\tau = \sum_{k=1}^{\frac{N+1}{2}} \boldsymbol{\Phi}_k(t) \tag{2.55}$$

with

$$\boldsymbol{\Phi}_k(t) = \int_0^t F_k(t - \tau) \cdot \boldsymbol{\nabla} \times \boldsymbol{H}(\tau) d\tau. \tag{2.56}$$

Sampling at the discrete times $n\delta t$ $\{n = 1, 2, ...\}$ we have

$$\begin{aligned}\boldsymbol{\Psi}^{n+1} = \sum_{k=1}^{\frac{N+1}{2}} \boldsymbol{\Phi}_k^{n+1} = \sum_{k=1}^{\frac{N+1}{2}} \int_0^{(n+1)\delta t} F_k\left[(n+1)\delta t - \tau\right] \cdot \boldsymbol{\nabla} \times \boldsymbol{H}(\tau) d\tau = \\ = \sum_{k=1}^{\frac{N+1}{2}} \int_0^{n\delta t} e^{-A_k\delta t} e^{iB_k\delta t} F_k(n\delta t - \tau) \cdot \boldsymbol{\nabla} \times \boldsymbol{H}(\tau) d\tau + \\ + \sum_{k=1}^{\frac{N+1}{2}} \int_{n\delta t}^{(n+1)\delta t} F_k\left[(n+1)\delta t - \tau\right] \cdot \boldsymbol{\nabla} \times \boldsymbol{H}(\tau) d\tau = \\ = \sum_{k=1}^{\frac{N+1}{2}} e^{-A_k\delta t} e^{iB_k\delta t} \boldsymbol{\Phi}_k^n + A \cdot \boldsymbol{\nabla} \times \boldsymbol{H}^{n+\frac{1}{2}},\end{aligned} \tag{2.57}$$

where

$$A = \sum_{k=1}^{\frac{N+1}{2}} \int_{n\delta t}^{(n+1)\delta t} F_k\left[(n+1)\delta t - \tau\right] d\tau = \sum_{k=1}^{\frac{N+1}{2}} \frac{e^{i\Theta_k} N_k}{\cos\Theta_k} \frac{e^{(-A_k+iB_k)\delta t} - 1}{-A_k + iB_k}. \tag{2.58}$$

The expression for the electric field then becomes

$$\begin{aligned}\boldsymbol{E}^{n+1} = \Im\left\{\boldsymbol{\Psi}^{n+1}\right\} = \sum_{k=1}^{\frac{N+1}{2}} e^{-A_k\delta t}\left[\sin(B_k\delta t)\Re\left\{\boldsymbol{\Phi}_k^n\right\} + \cos(B_k\delta t)\Im\left\{\boldsymbol{\Phi}_k^n\right\}\right] + \\ + \Im\left\{A\right\} \cdot \boldsymbol{\nabla} \times \boldsymbol{H}^{n+\frac{1}{2}}\end{aligned} \tag{2.59}$$

and the number of required extra memorizations for the auxiliary variables needed for the code execution is $N - 1$.

If N is even the grade of the denominator of (2.46) is odd. Developing it whit $N/2$ couples of complex conjugate poles and one real pole we have

$$G(s) = \frac{\prod_{p=1}^{N}(s + \gamma_p)}{C_0 \cdot (s + \Gamma) \prod_{k=1}^{\frac{N}{2}}(s + s_k)(s + s_k^*)}, \tag{2.60}$$

where $s_k = A_k + iB_k$ is the kth pole. Acting the ILT of (2.60) we obtain

$$G(t) = M_0 e^{-\Gamma t} + \sum_{k=1}^{\frac{N}{2}} e^{-A_k t} \left[M_k \cos(B_k t) + N_k \sin(B_k t) \right], \tag{2.61}$$

where $M_k \left\{k = 0, ... \frac{N}{2}\right\}$ and $N_k \left\{k = 1, ... \frac{N}{2}\right\}$ are real coefficients and they depend on γ_p $\{p = 1, ...N\}$, C_0, A_k and B_k $\left\{k = 1, ... \frac{N}{2}\right\}$. Setting

$$\frac{N_k}{M_k} = \tan \Theta_k = \frac{\sin \Theta_k}{\cos \Theta_k} \tag{2.62}$$

we obtain

$$G(t) = M_0 e^{-\Gamma t} + \sum_{k=1}^{\frac{N}{2}} \frac{M_k e^{-A_k t}}{\cos \Theta_k} \cos(B_k t - \Theta_k). \tag{2.63}$$

If we put

$$F(t) = \sum_{k=0}^{\frac{N}{2}} F_k(t), \tag{2.64}$$

where

$$F_0(t) = M_0 e^{-\Gamma t} \tag{2.65}$$

$$F_k(t) = \frac{M_k e^{-A_k t}}{\cos \Theta_k} e^{i(B_k t - \Theta_k)} \quad (k = 1, 2 ... \frac{N}{2}) \tag{2.66}$$

and

$$\boldsymbol{\Psi}(t) = \int_0^t F(t - \tau) \cdot \boldsymbol{\nabla} \times \boldsymbol{H}(\tau) d\tau, \tag{2.67}$$

we have

$$\boldsymbol{E}(t) = \Re \left\{ \boldsymbol{\Psi}(t) \right\}. \tag{2.68}$$

Developing $\boldsymbol{\Psi}(t)$ we obtain

$$\boldsymbol{\Psi}(t) = \int_0^t \sum_{k=0}^{\frac{N}{2}} F_k(t - \tau) \cdot \boldsymbol{\nabla} \times \boldsymbol{H}(\tau) d\tau = \sum_{k=0}^{\frac{N}{2}} \boldsymbol{\Phi}_k(t), \tag{2.69}$$

where

$$\boldsymbol{\Phi}_k(t) = \int_0^t F_k(t - \tau) \cdot \boldsymbol{\nabla} \times \boldsymbol{H}(\tau) d\tau. \tag{2.70}$$

Sampling at the discrete times $n\delta t$ $\{n = 1, 2, ...\}$ we have

$$\begin{aligned}\boldsymbol{\Psi}^{n+1} = \sum_{k=0}^{\frac{N}{2}} \boldsymbol{\Phi}_k^{n+1} = \sum_{k=0}^{\frac{N}{2}} \int_0^{(n+1)\delta t} F_k\left[(n+1)\delta t - \tau\right] \cdot \boldsymbol{\nabla} \times \boldsymbol{H}(\tau) d\tau = \\ = \int_0^{n\delta t} e^{-\Gamma\delta t} F_0(n\delta t - \tau) \cdot \boldsymbol{\nabla} \times \boldsymbol{H}(\tau) d\tau + \\ + \sum_{k=1}^{\frac{N}{2}} \int_0^{n\delta t} e^{-A_k\delta t} e^{iB_k\delta t} F_k(n\delta t - \tau) \cdot \boldsymbol{\nabla} \times \boldsymbol{H}(\tau) d\tau + \\ + \sum_{k=0}^{\frac{N}{2}} \int_{n\delta t}^{(n+1)\delta t} F_k\left[(n+1)\delta t - \tau\right] \cdot \boldsymbol{\nabla} \times \boldsymbol{H}(\tau) d\tau = \\ = e^{-\Gamma\delta t} \boldsymbol{\Phi}_0^n + \sum_{k=1}^{\frac{N}{2}} e^{-A_k\delta t} e^{iB_k\delta t} \boldsymbol{\Phi}_k^n + A \cdot \boldsymbol{\nabla} \times \boldsymbol{H}^{n+\frac{1}{2}},\end{aligned} \tag{2.71}$$

where

$$A = \sum_{k=1}^{\frac{N}{2}} \int_{n\delta t}^{(n+1)\delta t} F_k\left[(n+1)\delta t - \tau\right] d\tau = \sum_{k=1}^{\frac{N}{2}} \frac{e^{-i\Theta_k} M_k}{\cos\Theta_k} \frac{e^{(-A_k + iB_k)\delta t} - 1}{-A_k + iB_k}. \tag{2.72}$$

The expression for the electric field then becomes

$$\begin{aligned}\boldsymbol{E}^{n+1} = \Re\left\{\boldsymbol{\Psi}^{n+1}\right\} = e^{-\Gamma\delta t} \boldsymbol{\Phi}_0^n + \\ + \sum_{k=1}^{\frac{N}{2}} e^{-A_k\delta t} \left[\cos(B_k\delta t) \Re\left\{\boldsymbol{\Phi}_k^n\right\} - \sin(B_k\delta t) \Im\left\{\boldsymbol{\Phi}_k^n\right\}\right] + \Re\left\{A\right\} \cdot \boldsymbol{\nabla} \times \boldsymbol{H}^{n+\frac{1}{2}}\end{aligned} \tag{2.73}$$

and the number of required extra memorizations is N with respect to the non-dispersive case.

2.4 The Lorentz model

Let assume a dispersive medium described by a Lorentz function (single pair of complex conjugate poles) and a time dependence $e^{-i\omega t}$. If ω_0 is the resonant frequency of the medium and δ_0 is the damping coefficient the complex susceptibility is

$$\chi(\omega) = (\epsilon_s - \epsilon_\infty)\frac{\omega_0^2}{\omega_0^2 - 2i\omega\delta_0 - \omega^2}, \tag{2.74}$$

where ϵ_s is the static permettivity. In the Laplace domain ($s = -i\omega$) it becomes

$$\chi(s) = (\epsilon_s - \epsilon_\infty)\frac{\omega_0^2}{\omega_0^2 + 2s\delta_0 + s^2}. \tag{2.75}$$

This expression can be simplified in the form

$$\chi(s) = \frac{\Omega_0}{(s + \alpha_0 + i\beta_0)(s + \alpha_0 - i\beta_0)}, \tag{2.76}$$

where $\alpha_0 = \delta_0$, $\beta_0 = \sqrt{\omega_0^2 - \delta_0^2}$ and $\Omega_0 = (\epsilon_s - \epsilon_\infty)\omega_0^2$. Putting (2.75) into (2.7), considering $\sigma = 0$ and developing some calculation it results

$$G(s) = \frac{(s + \alpha_0 + i\beta_0)(s + \alpha_0 - i\beta_0)}{Cs[(s + A_0 + iB_0)(s + A_0 - iB_0)]}, \tag{2.77}$$

where $C = \epsilon_0\epsilon_\infty$, $A_0 = \alpha_0$ and $B_0 = \sqrt{\beta_0^2 + \Omega_0/\epsilon_\infty}$.

Acting the ILT of (2.77) we obtain

$$G(t) = K + e^{-A_0 t}\left[M_0\cos(B_0 t) + N_0\sin(B_0 t)\right], \tag{2.78}$$

where

$$K = \frac{1}{\epsilon_0\epsilon_s}, \tag{2.79}$$

$$M_0 = \frac{\epsilon_s - \epsilon_\infty}{\epsilon_0\epsilon_\infty\epsilon_s}, \tag{2.80}$$

and

$$N_0 = \frac{\delta_0(\epsilon_s - \epsilon_\infty)}{B_0\epsilon_0\epsilon_\infty\epsilon_s} \tag{2.81}$$

are real quantities.

Setting

$$\Theta_0 = \arctan(\frac{N_0}{M_0}) \tag{2.82}$$

we have

$$G(t) = K + \frac{M_0 e^{-A_0 t}}{\cos\Theta_0}\cos(B_0 t - \Theta_0). \tag{2.83}$$

If we put

$$F(t) = K + F_0(t), \tag{2.84}$$

where

$$F_0(t) = \frac{M_0 e^{-A_0 t}}{\cos\Theta_0} e^{i(B_0 t - \Theta_0)} \tag{2.85}$$

and

$$\boldsymbol{\Psi}(t) = \int_0^t F(t-\tau)\cdot\boldsymbol{\nabla}\times\boldsymbol{H}(\tau)d\tau, \tag{2.86}$$

$$\boldsymbol{\Phi}(t) = \int_0^t F_0(t-\tau)\cdot\boldsymbol{\nabla}\times\boldsymbol{H}(\tau)d\tau, \tag{2.87}$$

it results

$$\boldsymbol{E}(t) = \Re\left\{\boldsymbol{\Psi}(t)\right\}. \tag{2.88}$$

Sampling at the discrete times $n\delta t$ $\{n = 1, 2, ...\}$ we have

$$\begin{aligned}\boldsymbol{\Psi}^{n+1} = \int_0^{(n+1)\delta t} \left[K + F_0\left((n+1)\delta t - \tau\right)\right]\cdot\boldsymbol{\nabla}\times\boldsymbol{H}(\tau)d\tau = \\ = \int_0^{(n+1)\delta t} K\cdot\boldsymbol{\nabla}\times\boldsymbol{H}(\tau)d\tau + \\ + \int_0^{n\delta t} e^{-A_0\delta t}e^{iB_0\delta t}F_0(n\delta t - \tau)\cdot\boldsymbol{\nabla}\times\boldsymbol{H}(\tau)d\tau + \\ + \int_{n\delta t}^{(n+1)\delta t} F_0\left((n+1)\delta t - \tau\right)\cdot\boldsymbol{\nabla}\times\boldsymbol{H}(\tau)d\tau = \\ = e^{-A_0\delta t}e^{iB_0\delta t}\boldsymbol{\Phi}^n + ((n+1)K\delta t + Z)\cdot\boldsymbol{\nabla}\times\boldsymbol{H}^{n+\frac{1}{2}},\end{aligned} \tag{2.89}$$

where

$$Z = \int_{n\delta t}^{(n+1)\delta t} F_0\left((n+1)\delta t - \tau\right)d\tau = \frac{M_0(e^{(-A_0+iB_0)\delta t} - 1)}{(-A_0 + iB_0)\cos\Theta_0}e^{-i\Theta_0}. \tag{2.90}$$

Taking the real part of (2.89), the expression for the electric field becomes

$$\begin{aligned}\boldsymbol{E}^{n+1} = e^{-A_0\delta t}\left[\cos(B_0\delta t)\Re\left\{\boldsymbol{\Phi}^n\right\} - \sin(B_0\delta t)\Im\left\{\boldsymbol{\Phi}^n\right\}\right] + \\ +\Re\left\{(n+1)K\delta t + Z\right\} \cdot \boldsymbol{\nabla} \times \boldsymbol{H}^{n+\frac{1}{2}}\end{aligned} \tag{2.91}$$

and the number of extra variables for each electric field component is 2 as in the traditional algorithm [24]. The magnetic field indeed does not need auxiliary variables and the updating equations are the same than in the standard FDTD algorithm.

2.5 The N-order Lorentz model

Let assume a dispersive medium described with a Lorentz multiple second-order poles. If N is the number of natural frequencies ω_p $\{p = 1, 2, ...N\}$ the complex susceptibility is

$$\chi(\omega) = (\epsilon_s - \epsilon_\infty)\sum_{p=1}^{N} \frac{G_p\omega_p^2}{\omega_p^2 - 2i\omega\delta_p - \omega^2}, \tag{2.92}$$

where ϵ_s is the static permittivity, ω_p is the pth resonant frequency and δ_p is the pth dumping coefficient, with the condition

$$\sum_{i=1}^{N} G_p = 1. \tag{2.93}$$

In the Laplace domain ($s = -i\omega$) we have

$$\chi(s) = (\epsilon_s - \epsilon_\infty)\sum_{p=1}^{N} \frac{G_p\omega_p^2}{\omega_p^2 + 2s\delta_p + s^2}, \tag{2.94}$$

that can be simplified in the form

$$\chi(s) = \sum_{p=1}^{N} \frac{\Omega_p}{(s+\xi_p)(s+\xi_p^*)}, \tag{2.95}$$

where $\xi_p = \alpha_p + i\beta_p$ is a complex number (the pth pole), ξ_p^* its conjugate and $\Omega_p = (\epsilon_s - \epsilon_\infty)G_p\omega_p^2$. Putting (2.94) into (2.7) and developing some calculation we obtain

$$G(s) = \frac{\prod_{p=1}^{N}(s+\xi_p)(s+\xi_p^*)}{\sum_{k=0}^{2N+1} C_k s^{2N+1-k}}, \tag{2.96}$$

where $C_k = f_k(\epsilon_0, \epsilon_\infty, \epsilon_s, \Omega_p, \xi_p, \xi_p^*)$ for $\{p = 1, ...N\}$ and $\{k = 0, ...2N+1\}$ are real coefficients because the dependence from ξ_p and ξ_p^* is of the type $\xi_p + \xi_p^*$ and $\xi_p \cdot \xi_p^*$, which are real quantities. f_k is a kth function depending on the polynomial calculation.

Developing the denominator of (2.96) with N conjugate poles and a real pole we have

$$\begin{aligned} G(s) &= \frac{\prod_{p=1}^{N}(s+\xi_p)(s+\xi_p^*)}{C_0 \cdot (s+\Gamma) \cdot \prod_{k=1}^{N}(s+s_k)(s+s_k^*)} = \\ &= \frac{\prod_{p=1}^{N}(s+\alpha_p+i\beta_p)(s+\alpha_p-i\beta_p)}{C_0 \cdot (s+\Gamma) \cdot \prod_{k=1}^{N}(s+A_k+iB_k)(s+A_k-iB_k)}, \end{aligned} \tag{2.97}$$

where $s_k = A_k + iB_k$ $\{k = 1, ...N\}$ are complex quantities. Acting the ILT of (2.97) it results

$$G(t) = M_0 e^{-\Gamma t} + \sum_{k=1}^{N} e^{-A_k t}\left[M_k \cos(B_k t) + N_k \sin(B_k t)\right], \tag{2.98}$$

where M_k $\{k = 0, ...N\}$, N_k $\{k = 1, ...N\}$ are real coefficients and are functions of α_p, β_p, A_k, B_k, Γ and C_0 $\{i, k = 1, ...N\}$. Setting

$$\frac{N_k}{M_k} = \tan\Theta_k = \frac{\sin\Theta_k}{\cos\Theta_k} \tag{2.99}$$

we obtain

$$G(t) = M_0 e^{-\Gamma t} + \sum_{k=1}^{N} \frac{M_k e^{-A_k t}}{\cos\Theta_k}\cos(B_k t - \Theta_k). \tag{2.100}$$

If we put

$$F(t) = \sum_{k=0}^{N} F_k(t), \tag{2.101}$$

where

$$F_0(t) = M_0 e^{-\Gamma t}, \tag{2.102}$$

$$F_k(t) = \frac{M_k e^{-A_k t}}{\cos \Theta_k} e^{i(B_k t - \Theta_k)} \quad (k = 1, 2...N) \tag{2.103}$$

and

$$\boldsymbol{\Psi}(t) = \int_0^t F(t-\tau) \cdot \boldsymbol{\nabla} \times \boldsymbol{H}(\tau) d\tau, \tag{2.104}$$

we have

$$\boldsymbol{E}(t) = \Re \left\{ \boldsymbol{\Psi}(t) \right\}. \tag{2.105}$$

Developing $\boldsymbol{\Psi}(t)$ we obtain

$$\boldsymbol{\Psi}(t) = \int_0^t \sum_{k=0}^{N} F_k(t-\tau) \cdot \boldsymbol{\nabla} \times \boldsymbol{H}(\tau) d\tau = \sum_{k=0}^{N} \boldsymbol{\Phi}_k(t), \tag{2.106}$$

where

$$\boldsymbol{\Phi}_k(t) = \int_0^t F_k(t-\tau) \cdot \boldsymbol{\nabla} \times \boldsymbol{H}(\tau) d\tau. \tag{2.107}$$

Sampling at the discrete times $n\delta t$ $\{n = 1, 2, ...\}$ we have

$$\begin{aligned}\boldsymbol{\Psi}^{n+1} = \sum_{k=0}^{N} \boldsymbol{\Phi}_k^{n+1} = \sum_{k=0}^{N} \int_0^{(n+1)\delta t} F_k\left[(n+1)\delta t - \tau\right] \cdot \boldsymbol{\nabla} \times \boldsymbol{H}(\tau) d\tau = \\ = \int_0^{n\delta t} e^{-\Gamma\delta t} F_0(n\delta t - \tau) \cdot \boldsymbol{\nabla} \times \boldsymbol{H}(\tau) d\tau + \\ + \sum_{k=1}^{N} \int_0^{n\delta t} e^{-A_k\delta t} e^{iB_k\delta t} F_k(n\delta t - \tau) \cdot \boldsymbol{\nabla} \times \boldsymbol{H}(\tau) d\tau + \\ + \sum_{k=0}^{N} \int_{n\delta t}^{(n+1)\delta t} F_k\left[(n+1)\delta t - \tau\right] \cdot \boldsymbol{\nabla} \times \boldsymbol{H}(\tau) d\tau = \\ = e^{-\Gamma\delta t} \boldsymbol{\Phi}_0^n + \sum_{k=1}^{N} e^{-A_k\delta t} e^{iB_k\delta t} \boldsymbol{\Phi}_k^n + A \cdot \boldsymbol{\nabla} \times \boldsymbol{H}^{n+\frac{1}{2}},\end{aligned} \tag{2.108}$$

where

$$A = \sum_{k=1}^{N} \int_{n\delta t}^{(n+1)\delta t} F_k\left[(n+1)\delta t - \tau\right] d\tau = \sum_{k=1}^{N} \frac{e^{-i\Theta_k} M_k}{\cos\Theta_k} \frac{e^{(-A_k + iB_k)\delta t} - 1}{-A_k + iB_k}. \tag{2.109}$$

The expression for the electric field then becomes

$$\begin{aligned}\boldsymbol{E}^{n+1} = \Re\left\{\boldsymbol{\Psi}^{n+1}\right\} = e^{-\Gamma\delta t} \boldsymbol{\Phi}_0^n + \\ + \sum_{k=1}^{N} e^{-A_k\delta t} \left[\cos(B_k\delta t) \Re\left\{\boldsymbol{\Phi}_k^n\right\} - \sin(B_k\delta t) \Im\left\{\boldsymbol{\Phi}_k^n\right\}\right] + \Re\left\{A\right\} \cdot \boldsymbol{\nabla} \times \boldsymbol{H}^{n+\frac{1}{2}}\end{aligned} \tag{2.110}$$

and the number of required extra memorizations is $2N$.

2.6 The Drude with N critical points model

Starting from the description of noble metals by the Drude model together with 2 critical points [21–23, 58], we develop the FDTD theory in the general case of N critical points. Let assume a complex susceptibility

$$\chi(\omega) = \frac{-\omega_D^2}{\omega(\omega + i\gamma)} + \sum_{p=1}^{N} A_p\Omega_p \left(\frac{e^{i\phi_p}}{\Omega_p - \omega - i\Gamma_p} + \frac{e^{-i\phi_p}}{\Omega_p + \omega + i\Gamma_p} \right), \quad (2.111)$$

where ω_D and γ are the Drude parameters and the time dependence is of the type $e^{-i\omega t}$. A_p, Ω_p, ϕ_p and Γ_p for $\{p = 1, ...N\}$ are the parameters of the critical points model. In the Laplace domain ($s = -i\omega$) we have

$$\chi(s) = \frac{\omega_D^2}{s(s+\gamma)} + 2\sum_{p=1}^{N} A_p\Omega_p \left(\frac{\Omega_p \cos\phi_p - \Gamma_p \sin\phi_p - \sin\phi_p \cdot s}{s^2 + 2\Gamma_p \cdot s + \Gamma_p^2 + \Omega_p^2} \right), \quad (2.112)$$

that can be expressed in the form

$$\chi(s) = \frac{\omega_D^2}{s\Gamma(s)} + \sum_{p=1}^{N} \frac{R_p(s)}{Q_p(s)}. \quad (2.113)$$

Putting (2.112) into (2.7) and developing some calculation we obtain

$$G(s) = \frac{\Gamma(s)\prod_{p=1}^{N} Q_p(s)}{P(s)}, \quad (2.114)$$

with

$$P(s) = (\epsilon_0\epsilon_\infty s + \sigma)\Gamma(s)\prod_{p=1}^{N} Q_p(s) + \epsilon_0 \left[\omega_D^2 \prod_{p=1}^{N} Q_p(s) + \Gamma(s)\sum_{q=1}^{N} R_q(s) \prod_{p=1, p\neq q}^{N} Q_p(s) \right], \quad (2.115)$$

that can be simplified in

$$G(s) = \frac{\sum_{n=0}^{2N+1} a_n s^{2N+1-k}}{\sum_{k=0}^{2(N+1)} c_k s^{2(N+1)-k}}, \quad (2.116)$$

where a_n and c_k are real coefficients.

Putting the denominator of (2.116) in the monic form and developing it with $N+1$ conjugate poles through a complex expansion we have

$$G(s) = \sum_{k=1}^{N+1} \left(\frac{r_k}{s - s_k} + \frac{r_k^*}{s - s_k^*} \right), \tag{2.117}$$

where $s_k = A_k + iB_k$ is the kth pole and $r_k = U_k + iV_k$ is the residual of (2.116) in s_k

$$r_k = (s - s_k) \cdot G(s)|_{s=s_k}.$$

Acting the ILT of (2.117) we obtain

$$G(t) = \sum_{k=1}^{N+1} M_k e^{A_k t} \cos(B_k t + \Theta_k), \tag{2.118}$$

where $M_k = 2\sqrt{U_k^2 + V_k^2}$ and $\Theta_k = \arctan(V_k/U_k)$. If we set

$$F(t) = \sum_{k=1}^{N+1} F_k(t), \tag{2.119}$$

where

$$F_k(t) = M_k e^{A_k t} e^{i(B_k t + \Theta_k)} \quad \{k = 1, ...N+1\} \tag{2.120}$$

and

$$\boldsymbol{\Psi}(t) = \int_0^t F(t-\tau) \cdot \boldsymbol{\nabla} \times \boldsymbol{H}(\tau) d\tau, \tag{2.121}$$

we have

$$\boldsymbol{E}(t) = \Re \left\{ \boldsymbol{\Psi}(t) \right\}. \tag{2.122}$$

Developing $\boldsymbol{\Psi}(t)$ it results

$$\boldsymbol{\Psi}(t) = \sum_{k=1}^{N+1} \boldsymbol{\Phi}_k(t), \tag{2.123}$$

where

$$\boldsymbol{\Phi}_k(t) = \int_0^t F_k(t-\tau) \cdot \boldsymbol{\nabla} \times \boldsymbol{H}(\tau) d\tau. \tag{2.124}$$

Sampling (2.123) at the discrete times $n\delta t$ $\{n = 1, 2, ...\}$ it results

$$\begin{aligned}\boldsymbol{\Psi}^{n+1} = \sum_{k=1}^{N+1} \boldsymbol{\Phi}_k^{n+1} &= \sum_{k=1}^{N+1} \int_0^{(n+1)\delta t} F_k\left[(n+1)\delta t - \tau\right] \cdot \boldsymbol{\nabla} \times \boldsymbol{H}(\tau) d\tau = \\ &= \sum_{k=1}^{N+1} \int_0^{n\delta t} e^{A_k \delta t} e^{iB_k \delta t} F_k(n\delta t - \tau) \cdot \boldsymbol{\nabla} \times \boldsymbol{H}(\tau) d\tau + \\ &+ \sum_{k=1}^{N+1} \int_{n\delta t}^{(n+1)\delta t} F_k\left[(n+1)\delta t - \tau\right] \cdot \boldsymbol{\nabla} \times \boldsymbol{H}(\tau) d\tau = \\ &= \sum_{k=1}^{N+1} \left(e^{A_k \delta t} e^{iB_k \delta t} \boldsymbol{\Phi}_k^n + Z_k \cdot \boldsymbol{\nabla} \times \boldsymbol{H}^{n+\frac{1}{2}} \right), \end{aligned} \tag{2.125}$$

where

$$Z_k = \int_{n\delta t}^{(n+1)\delta t} F_k\left[(n+1)\delta t - \tau\right] d\tau = e^{i\Theta_k} M_k \frac{e^{(A_k + iB_k)\delta t} - 1}{A_k + iB_k}. \tag{2.126}$$

The expression for the electric field then becomes

$$\begin{aligned}\boldsymbol{E}^{n+1} = \Re\left\{\boldsymbol{\Psi}^{n+1}\right\} = \sum_{k=1}^{N+1} \left(e^{A_k \delta t} \cos(B_k \delta t) \Re\left\{\boldsymbol{\Phi}_k^n\right\} - e^{A_k \delta t} \sin(B_k \delta t) \Im\left\{\boldsymbol{\Phi}_k^n\right\}\right) + \\ + \Re\left\{Z_k\right\} \cdot \boldsymbol{\nabla} \times \boldsymbol{H}^{n+\frac{1}{2}}\end{aligned} \tag{2.127}$$

and the number of required extra memorizations is $2N + 2$.

Chapter 3

The parallelization for large simulations

3.1 MPI parallelization strategy for the Subgridding algorithm

We start with a parallelepiped domain $\mathcal{D}$, the coarser one, sampled with a uniform *space step* δ and a corresponding *time step* $\delta t = \delta/(2c_0)$ satisfying the Courant stability criterion [59–61], c_0 being the vacuum light velocity. The resulting space lattice $\mathcal{D}$ consists of $N_x \times N_y \times N_z$ cubic Yee cells of edge size δ. $\mathcal{D}$ has to be augmented with extra cells of the same size to get a bigger $\mathcal{D}'$ parallelepiped lattice, in such a way that the set $\mathcal{D}' \setminus \mathcal{D}$ is an external shell used to accomodate the absorbing boundary conditions. We use a fixed number of $b = 20$ extra cells for each face of $\mathcal{D}$, so that the total memory cost for $\mathcal{D}'$ amounts to $(N_x + 2b) \times (N_y + 2b) \times (N_z + 2b)$ Yee cells.

The $\mathcal{D}'$ domain is partitioned (see Fig. 3.1) into $L \geq 1$ subdomains, by "slicing" it along the z-axis through cutting planes parallel to the xy coordinate plane. Each one of the resulting L slices, referred to as a "z-slice", will have a thickness of given $Z_\ell > 0$ cells in the z direction ($1 \leq$

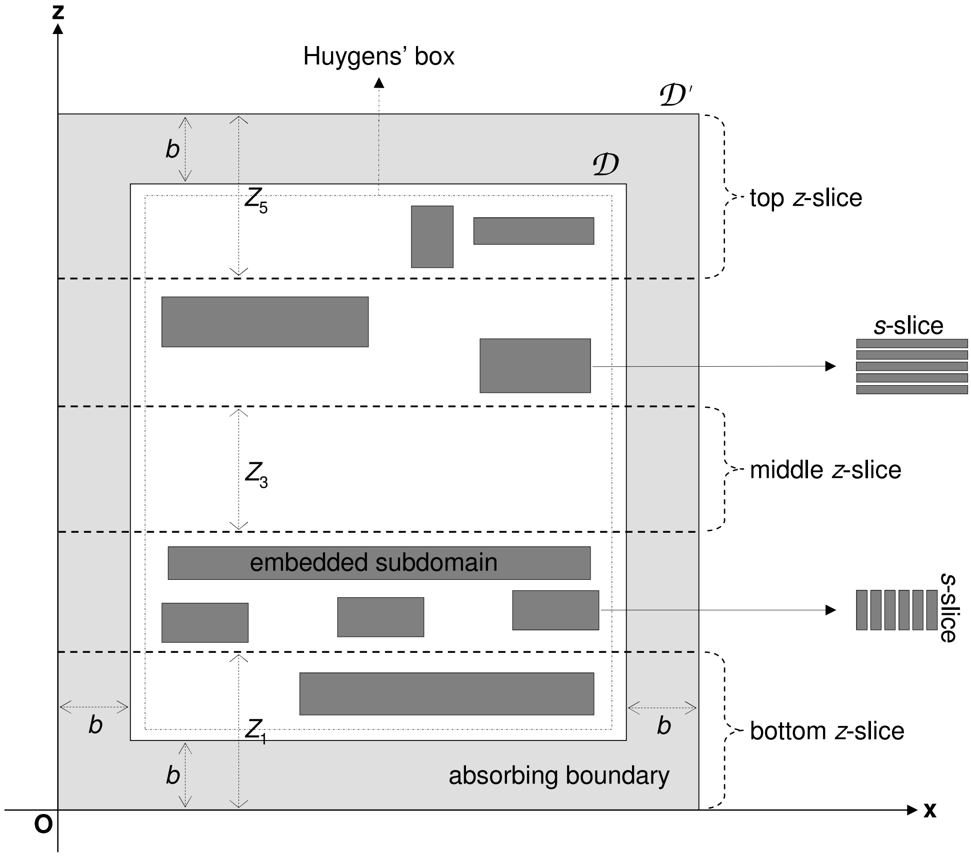


Figure 3.1: 2D view example of a $\mathcal{D}'$ with $L = 5$. The $\ell = 3$ z-slice is empty.

$\ell \leq L$), such that:

$$Z_1 + Z_2 + \ldots + Z_L = N_z + 2b.$$

Into each z-slice with index ℓ ($1 \leq \ell \leq L$) a given number $M_\ell \geq 0$ of sublattices can be embedded. Any two of them do not intersect and all are strictly contained inside the z-slice intersection with $\mathcal{D}$. The various sublattices are characterized by a *mesh refinement factor* $R_{\ell m}$ ($1 \leq \ell \leq L; 1 \leq m \leq M_\ell$). Their space and time steps are $\delta/R_{\ell m}$ and $\delta t/R_{\ell m}$ respectively so that the Courant limit remains unchanged. $R_{\ell m}$ are assumed to be odd integers: $3, 5, 7, \ldots$ without loss of generality. Every sublattice can be partitioned through cutting planes into $S_{\ell m} \geq 1$ slices (called "s-slice") along one, arbitrarily chosen, of the three coordinate directions (see Fig. 3.1). If the positive integers $T_{\ell mq}$ are the refined-cell thicknesses of the s-slices ($1 \leq q \leq S_{\ell m}$), they have to be chosen so that $R_{\ell m}$ is, for each q, a divisor of $T_{\ell mq}$ and consequently

$$C_{\ell m} = \sum_{q=1}^{S_{\ell m}} \frac{T_{\ell mq}}{R_{\ell m}}$$

gives the coarse-cell span, along the slicing direction, of the mth sublattice in the ℓth z-slice. Each slice, being it of the z-type or of the s-type, will be assigned to a single MPI process. The total number P of processes required is therefore given by (see Fig. 3.2):

$$P = L + \sum_{\ell=1}^{L} \sum_{m=1}^{M_\ell} S_{\ell m}.$$

Each process executes the usual FDTD "bulk" algorithm on its own lattice variables, inside a private local memory space. The overall space-grid continuity is achieved by *message passing* data at the various interfaces between the slices (cutting planes) and at the coarse/fine mesh interfaces. The task of data sharing is accomplished by means of MPI point-to-point communication routines. The FDTD bulk algorithm cannot advance to the next time step, until the data sharing in the whole space grid has been completed. In the MPI environment, the rank ρ of each of the P processes ($0 \leq \rho \leq P-1$) is assigned as follows: processes with $\rho = 0$ to

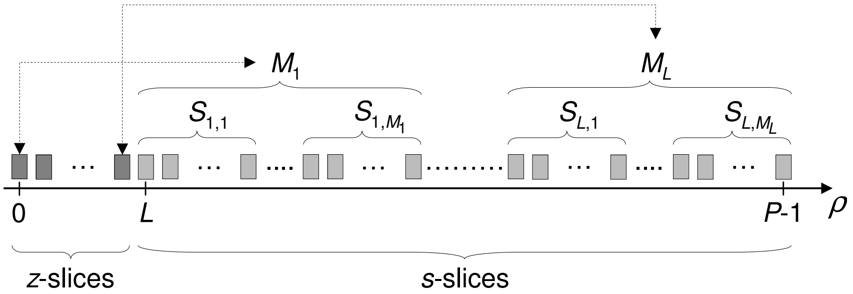


Figure 3.2: Scheme of the process rank ρ assignment for z-slices and s-slices.

$\rho = L - 1$ contain the L z-slices; the process running the q' s-slice, of the m' sublattice, in the ℓ' z-slice ($1 \leq q' \leq S_{\ell' m'}$), has rank:

$$\rho = L + \sum_{\ell=1}^{\ell'-1} \sum_{m=1}^{M_\ell} S_{\ell m} + \sum_{m=1}^{m'-1} S_{\ell' m} + q'.$$

Processes could be grouped into $1 + \sum_{\ell=1}^{L} M_\ell$ disjoint subsets. There is "intra-communication" between their members, but no "inter-communication" between any two of the subsets. One of such subsets is formed from the first L processes running the various z-slices; the others are formed from

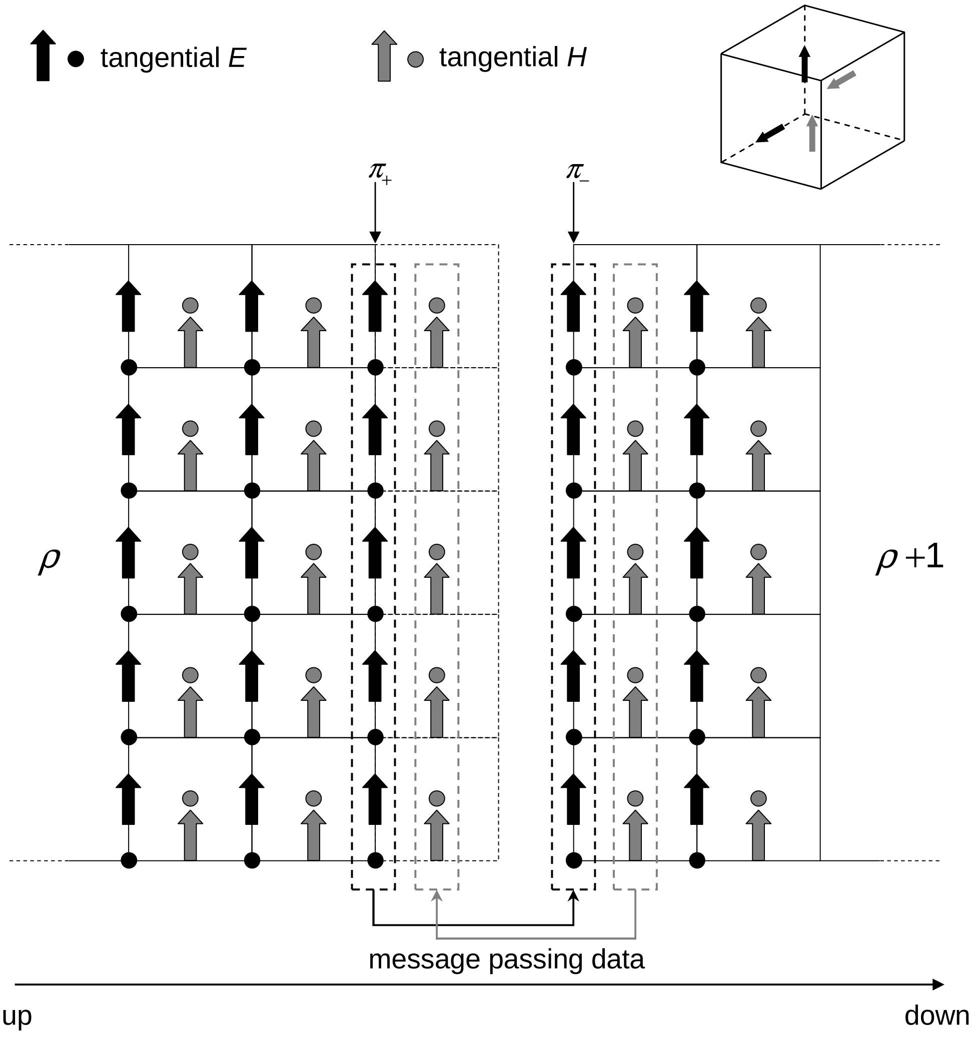


Figure 3.3: Data sharing at the cutting plane between two s-slices (2D view).

the various $S_{\ell m}$ processes running all the s-slices of the mth generic sublattice in the ℓth generic z-slice. Inside each of the subsets, a process ρ communicates with its two neighbors $\rho - 1$ and $\rho + 1$ only. It must share the *tangential* electric (E) and magnetic (H) fields components of the Yee cells lying on its two boundary cutting planes: the upstream one, π_-, and the downstream one, π_+. More precisely, it *sends* H from π_- to $\rho - 1$ and E from π_+ to $\rho + 1$; it *receives* E in π_- from $\rho - 1$ and H in π_+ from $\rho + 1$ (see Fig. 3.3). To avoid deadlocks, this task is best accomplished by calling the `Sendrecv` MPI routine. If ρ implements a first or last s-slice, then `MPI::PROC_NULL` is passed to the routine, instead of $\rho - 1$ or $\rho + 1$ respectively. The subgridding algorithm [25] is now described through the following cyclic sequence of steps (E and H denote the fields in the coarse grid, e and h denote the fields in the refined grid):

(1) Storing of the current time step tangential coarse E field components.

(2) Updating of the coarse bulk E field values (using also the H values in step (10) below). Add excitations if any.

(3) Updating of the coarse E field Discrete Fourier Transforms.

(4) Updating of the coarse bulk H field values (normal components to the coarse/refined interfaces included). Add excitations if any.

(5) Updating of the interior refined e field values.

(6) Calculation of the refined tangential e field components by bilinear space interpolation (needed to fill in missing refined locations) and linear time inter/extra-polation (needed to get values at refined times). The valued stored in step (1) and those obtained in step (2) are used.

(7) Updating of the refined e field Discrete Fourier Transforms.

(8) Updating the refined h field values.

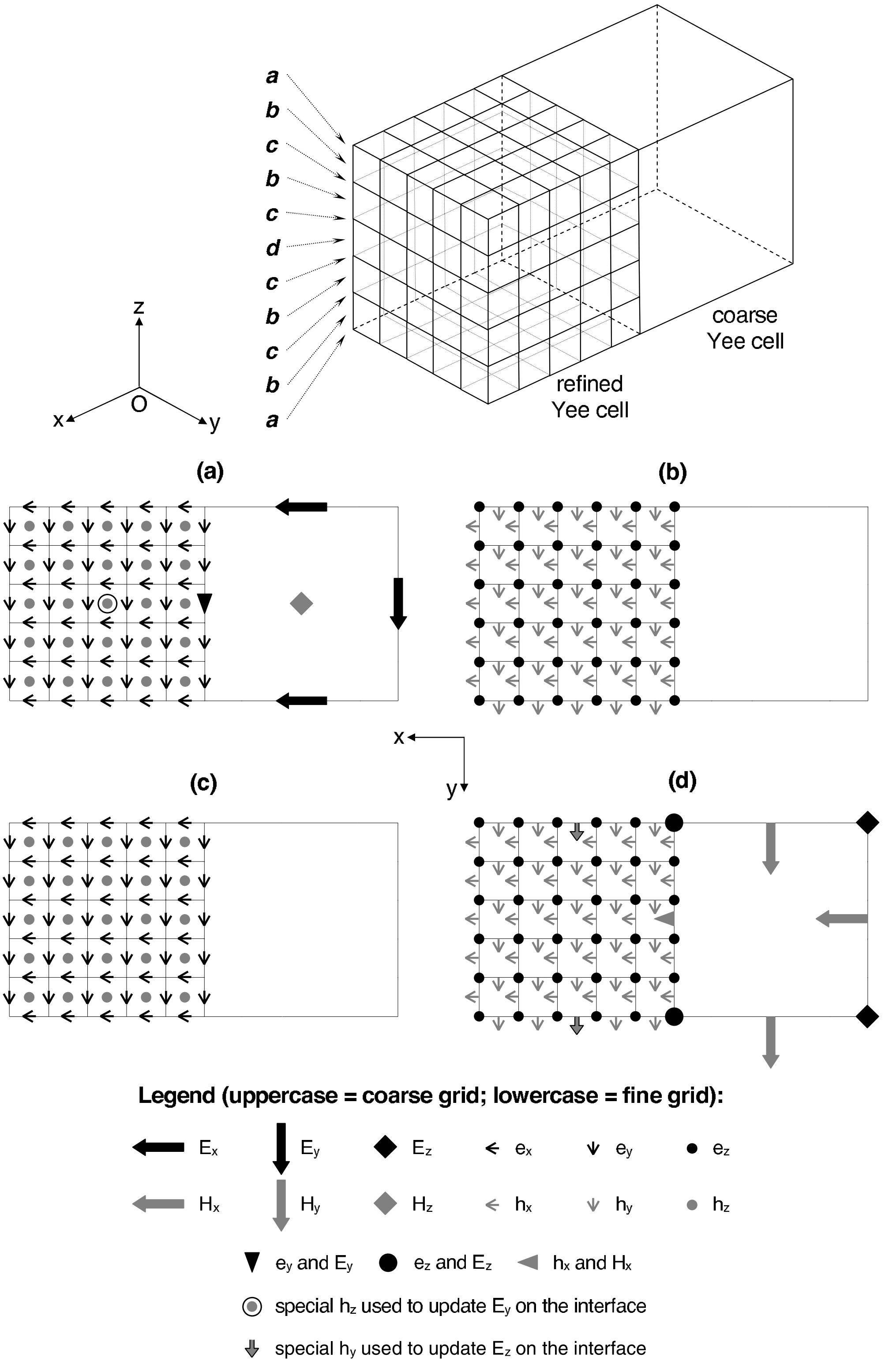

a
b
c
b
c
d
c
b
c
b
a
z
x
O
y
coarse
Yee cell
refined
Yee cell
(a)
(b)
x
y
(c)
(d)
Legend (uppercase = coarse grid; lowercase = fine grid):
Ex
Ey
Ez
ex
ey
ez
Hx
Hy
Hz
hx
hy
hz
ey and Ey
ez and Ez
hx and Hx
special hz used to update Ey on the interface
special hy used to update Ez on the interface

Figure 3.4: Fields layout at a coarse/fine grid interface in the yz-plane with a mesh refinement factor $R = 5$.

(9) Repeat steps from (5) to (8) R times (R being the refinement factor).

(10) Application of the spatial filtering technique to a subset (corresponding to the coarse grid sampling) of the refined tangential h field components and storing of the values.

(11) Repetition of steps from (1) to (10): an FDTD iteration.

Extra communication is then required at the coarse/fine grid interfaces (see Fig. 3.4). To this end, each z-slice manages the data sharing for all the s-slice interfaces of every sublattice it contains. In turn, a process ρ running a s-slice has to know the rank of the z-slice in which it is contained. Tangential E values on the interfaces have to be passed from the coarse side to the refined one and a space-time interpolation must be performed. In our parallelized implementation, the interpolation task is left to the process receiving the data, after the program has returned from the MPI `Recv` call. Moreover, the refined side has to pass *part* of its tangential H interface values, back to the coarse one. Such values have to undergo a "spatial" filtering procedure, according to formula (5) of [25], before they can be used in the FDTD bulk algorithm of the coarse side. Such task is left again to the refined side and then the program calls the MPI `Send` function. The functional scheme for the parallel version of the subgridding algorithm [25] is shown in Fig. 3.5 (labels 1, 2, $1'$, $2'$, A, B, A', B' in the figure will be referenced later on in Subsections 3.2.3 and 3.2.4).

Before the FDTD algorithm is entered, initial data for antenna structures or target complex permittivities are loaded from binary files. There is a data file for every sublattice. The loaded data are partitioned and dispatched to the appropriate s-slice processes by means of pointers. Additional initial data files are loaded, one for each z-slice: they contain parameters for the numbers of cells, numbers of sublattices, numbers of slices, placement of the sublattices, filtering order, and so on. At the end, the resulting data are collected from the various slices and packed, in such a way that there is a file of results for the $\mathcal{D}$ domain and each of its refined subdomains. As far as accuracy is concerned, we use the $\lambda_{min}/10 \div \lambda_{min}/20$ criterion [3], where λ_{min} is the shortest wavelength of interest in the response spectrum. This should assure an accuracy amounting to at least 5%. In early subgridding tests with experimental to analytical comparison, we found [25] an accuracy of about 3% or better for canonical objects 20 space step apart from the outer boundary of the FDTD grid, where Higdon-Mur absorbing boundary conditions [62] were imposed. We also did not found any relevant effect of the coarse/refined transition boundaries, if an adequate spatial filtering order is used, provided that the stencil of the filtering procedure involves homogeneous Yee cells only. For problems of electromagnetic field interaction with complex structures then, the subgridding should be applied only to those space regions where the local value of the relative permittivity reduces the local wavelength and thus the local space step value needed for a satisfactory accuracy.

3.1.1 Dynamic memory allocation and user defined MPI types

This subsection gives basic information about memory management specific of our MPI parallel implementation of our subgridding algorithm [25]. In the C++ encoding we use pointers and dynamic memory allocation to implement the data structures for the electric and magnetic field components lattice variables. In fact, no previous knowledge of the actual size of the model is available before compilation. Let then `F` be a generic field component and suppose we want for it a 3D lattice of `(M+1)` × `(N+1)` × `(P+1)` points along the x, y and z axes respectively. As has been seen in Section 3.1, data corresponding to the faces of parallelepipeds embedded in the `F` lattice have to be exchanged between pairs of processes at the coarse/fine grids interfaces. From the viewpoint of either the sender or receiver, the

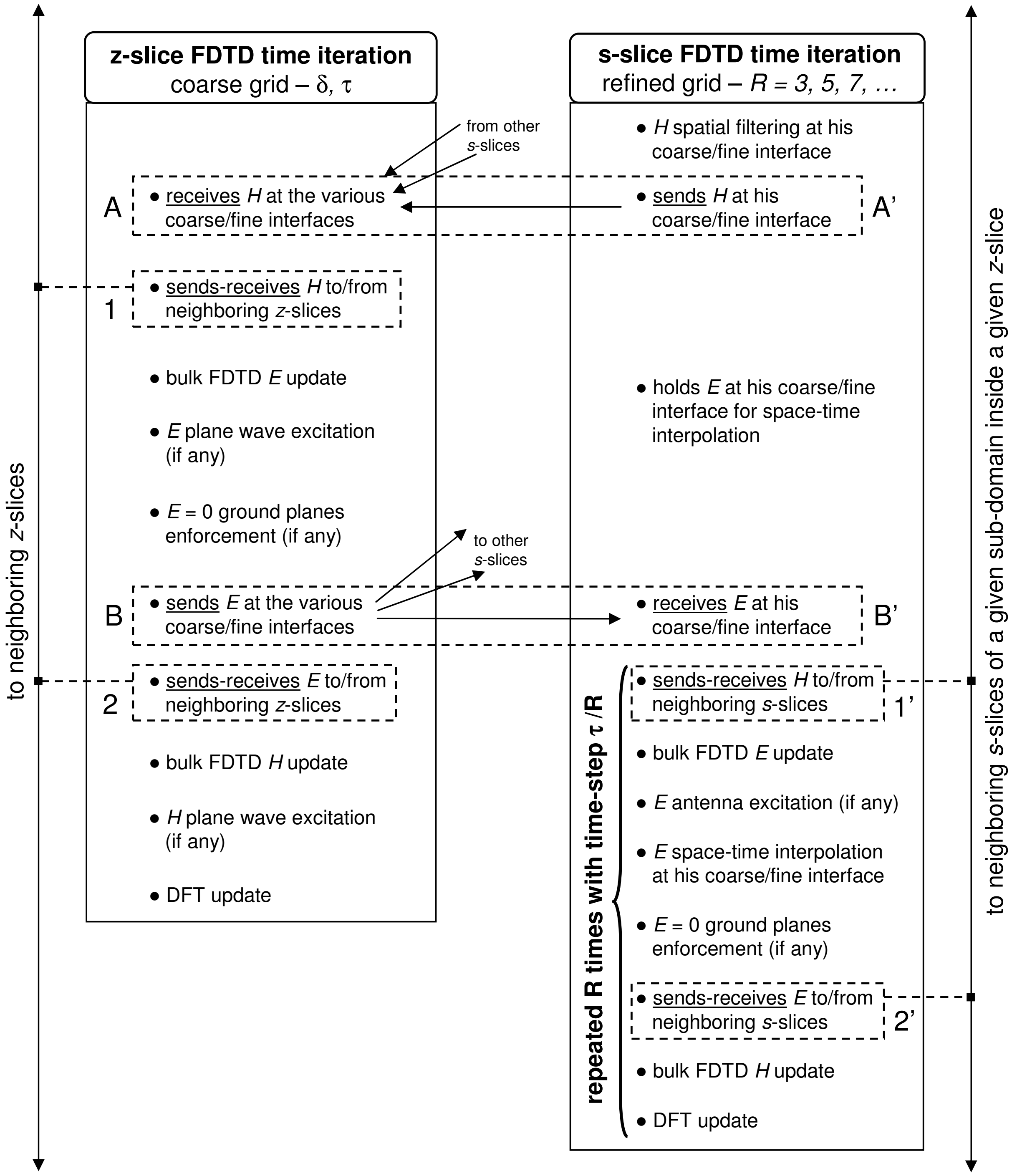

z-slice FDTD time iteration
coarse grid – δ, τ
s-slice FDTD time iteration
refined grid – R = 3, 5, 7, ...
• H spatial filtering at his coarse/fine interface
from other s-slices
A
• receives H at the various coarse/fine interfaces
• sends H at his coarse/fine interface
A'
1
• sends-receives H to/from neighboring z-slices
• bulk FDTD E update
• E plane wave excitation (if any)
• E = 0 ground planes enforcement (if any)
• holds E at his coarse/fine interface for space-time interpolation
to other s-slices
B
• sends E at the various coarse/fine interfaces
• receives E at his coarse/fine interface
B'
2
• sends-receives E to/from neighboring z-slices
• bulk FDTD H update
• H plane wave excitation (if any)
• DFT update
repeated R times with time-step τ /R
• sends-receives H to/from neighboring s-slices
1'
• bulk FDTD E update
• E antenna excitation (if any)
• E space-time interpolation at his coarse/fine interface
• E = 0 ground planes enforcement (if any)
• sends-receives E to/from neighboring s-slices
2'
• bulk FDTD H update
• DFT update
to neighboring z-slices
to neighboring s-slices of a given sub-domain inside a given z-slice

Figure 3.5: Parallel FDTD functional scheme for the subgridding algorithm.

best way to do this is by *defining* new MPI types [63,64], derived from the original `MPI::FLOAT` one. Such definitions require additional parameters, used by MPI to recognize regular patterns of subsets of `F` and to group and collect them out of `F` in a `Send` routine, or to fill in `F` appropriately with an incoming data flush in a `Recv` routine. In defining those new types, it is important to keep in mind the original ordering of the `F` array, that is to say the contiguity pattern of its memory locations (by appropriately changing the above lines of code, one could equally well allocate memory for `F` to include different orderings).

- For a face parallel to the xy-plane, consisting of $\mathtt{m} \times \mathtt{n}$ lattice points and lower left vertex at `I`, `J`, `K` $(0 < \mathtt{I}, \mathtt{J}; \mathtt{I+m} < \mathtt{M}; \mathtt{J+n} < \mathtt{N}; 0 < \mathtt{K} < \mathtt{P})$, one can create the derived type by invoking the method `Create_vector` with parameters values: $\mathtt{count} = \mathtt{n}$ blocks, each of $\mathtt{blocklength} = \mathtt{m}$ and with a $\mathtt{stride} = \mathtt{M}$ of old types between the blocks. The whole face can then be sent or received in a single call, starting at the address `&F[K][J][I]`.

- For a face parallel to the xz-plane, consisting of $\mathtt{m} \times \mathtt{p}$ lattice points and lower left vertex at `I`, `J`, `K` $(0 < \mathtt{I}, \mathtt{K}; \mathtt{I+m} < \mathtt{M}; \mathtt{K+p} < \mathtt{P}; 0 < \mathtt{j} < \mathtt{N})$, one can invoke again the `Create_vector` method, with parameters values: $\mathtt{count} = \mathtt{p}$ blocks, each of $\mathtt{blocklength} = \mathtt{m}$ and with a $\mathtt{stride} = \mathtt{M} \times \mathtt{N}$ of old types between the blocks. The whole face can then be sent or received in a single call, starting at the address `&F[K][J][I]`.

- Data in a face parallel to the yz-plane, having $\mathtt{n} \times \mathtt{p}$ lattice points and lower left vertex at `I`, `J`, `K` $(0 < \mathtt{J}, \mathtt{K}; \mathtt{J+n} < \mathtt{N}; \mathtt{K+p} < \mathtt{P}; 0 < \mathtt{i} < \mathtt{M})$, could be sent in `p` chunks, each starting at the address `&F[K+k][J][I]`, $0 \leq \mathtt{k} \leq \mathtt{p}$. Each chunk would correspond to an MPI derived type with $\mathtt{count} = \mathtt{n}, \mathtt{blocklength} = 1$ and $\mathtt{stride} = \mathtt{M}$. The drawback of such a choice is that it requires `p` calls to the `Send` and `Recv` routines, with the bad side effect of amplifying the latency time by the same factor `p`. A better way is to create a derived MPI type by invoking the method

`Create_indexed` [63, 64]. `Create_vector` does not work here, because the parameters must be constant integers. With `Create_indexed` one can specify an array of `blocklengths` and, instead of a constant `stride`, an array of corresponding `displacements`, in units of the old type from the starting location.

3.2 Parallel Subgridding Algorithm Performance Analysis

To analyze the parallel subgridding algorithm implementation performances, we defined three 3D FDTD test configurations, here conventionally referred to as T_1, T_2 and T_3.

- T_1 has an outer $\mathcal{D}'$ domain with a single z-slice ($L = 1$). $\mathcal{D} \subset \mathcal{D}'$ has $N_x = 200$, $N_y = 100$, $N_z = 100$ coarse Yee cells with a space step of 1.4 cm. It embeds a single subdomain ($M_1 = 1$) made of $875 \times 252 \times 147$ Yee cells with a space step of 0.2 cm., corresponding to a refinement factor $R_{1,1} = 7$. The latter models a standing human male body through the complex permittivity (dielectric constant in Farad/m and electric conductivity in Siemens/m) values, at a frequency of 2 GHz, of its anatomical structures. The human has its feet on a ground yz plane and is exposed to a x linearly polarized electromagnetic plane pulse of that frequency, impinging on it along the decreasing z direction. Such a model, without subgridding, was developed and used in [65]. The refined space step value has been chosen to get a good accuracy at the anatomical space scale for the given wavelength. If one uses this space step value for the entire model, there would be an increase by a factor of about 18 in the memory and simulation time requests.

- T_2 has the same outer $\mathcal{D}'$ domain than T_1. It embeds two subdomains ($M_1 = 2$). One modeling a commercial Radio Base Station antenna, made of 700 × 154 × 112 Yee cells with a space step of 0.2 cm. The antenna operates at a frequency of 1862.5 MHz and is made of a double array (for a double polarization of ±45°) of radiating elements, each array consisting of 8 dipole pairs. Moreover, there is a metallic shield in the back. The other subdomain models the above mentioned standing human male body on a grounded yz plane, made of 875 × 252 × 147 Yee cells with a space step of 0.2 cm. Both sublattices have the same refinement factor: $R_{1,1} = R_{1,2} = 7$. T_2 increases the complexity of T_1 by adding an e.m. source-containing sublattice. Now the human body is not in a plane wave regime, but exposed to the antenna electromagnetic near-field (see Fig. 3.6).

- T_3 is the same as T_2, but realized with two z-slices ($L = 2$): one for each of the sublattices ($M_1 = M_2 = 1$). From a serial programming point of view, there is no difference between T_2 and T_3. However, in parallel programming, T_3 adds complexity to T_2, because one more process is required in the $\mathcal{D}'$ modelization. T_3 has been introduced here to test the z-slice communication, which also affects the absorbing boundary conditions [35] implementation.

The parallel runs of T_1, T_2 and T_3 are made with variable s-slice numbers ($S_{\ell m}$) for the embedded sublattices. Also, various possible slicing directions (along the x or y or z axes) are taken into account. The code was compiled and executed on three types of systems, here conventionally referred to as WS, SP5 and BCX.

- WS is a dual processor Linux workstation, with two 3 GHz Xeon$^{\mathrm{TM}}$ EM64T Hyper-Threading Technology$^{\mathrm{TM}}$ CPUs, having a cache size of 1024 KB each and sharing a 6 GB memory space. Each CPU is treated by the operating system as two virtual processors. Installed are the GNU C++ ver. 3.4.6 compiler and the MPICH2 ver. 0.4 implementation [66] of the Message Passing API.

- SP5 is an IBM SP Cluster 1600, made of 64 SMP nodes intercon-

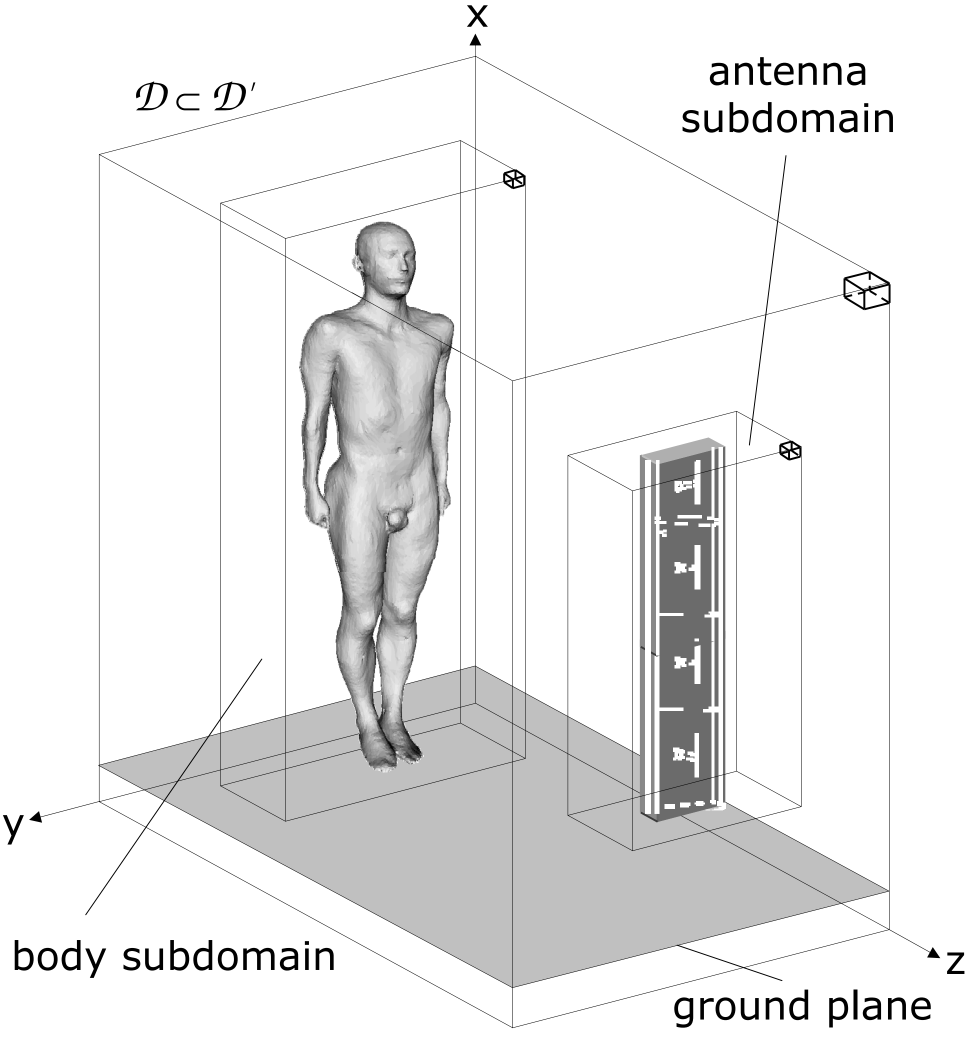


Figure 3.6: Schematic of the $\mathrm{T}_2 - \mathrm{T}_3$ FDTD test configurations.

nected through a Federation High Performance Switch with a bandwidth of 2 GB/s. Each node contains 8 IBM Power5 1.9 GHz processors, for a total of 512 CPUs: 60 of the nodes have 16 GB of shared memory, the remaining 4 have 64 GB. Installed are the IBM AIX ver. 5.3 operating system and the IBM `xlC` C++ serial compiler. SP5 has the IBM Parallel Operating Environment (POE) and the LoadLeveler job management system to submit and schedule batch jobs. The SP5 peak performance amounts to 3.89 Tflop/s.

- BCX is an IBM BCX Cluster, made of 1280 SMP nodes, each containing 2 AMD Opteron$^{\mathrm{TM}}$ 2.6 GHz Dual Core processors (i.e., 4 cores per node) and with 8 GB of shared memory, for a total of 5120 cores. The nodes are interconnected through an InfiniBand network with a bandwidth of 5 Gb/s. Installed is the Intel C++ serial compiler `iCC`. The OpenMPI (not to be confused with OpenMP) implementation of the Message Passing API can be used [67]. The queuing system is LSF. The BCX peak performance amounts to 26.6 Tflop/s.

3.2.1 Profiling the serial version. OpenMP loop-level parallelism.

The T_1 and T_2 -T_3 test configurations ran for 1200 time iterations using the *serial* code version of the subgridding algorithm [25] on a single processor (remember that T_2 and T_3 coincide in this case). T_1 required about 1.8 GB of memory, while T_2 -T_3 took about 2.4 GB of memory. The running times reported in Tab. 3.1 refer to the program calculation kernels only. In fact, times for initial data loading and final results output are negligible. A lot of time is spent (the top of list) in the space loop structures which update, in the *refined* grid, the: *a*) time domain values of the electric fields by the finite difference equations (44.66% of the total execution time); *b*) time domain values of the magnetic fields by the finite difference equations

	SP5	BCX	WS
T_1	3.0×10^4 ($\simeq$ 8.3 h)	4.9×10^4 ($\simeq$ 13.6 h)	3.7×10^4 ($\simeq$ 10.3 h)
T_2-T_3	4.0×10^4 ($\simeq$ 11.1 h)	6.3×10^4 ($\simeq$ 17.6 h)	4.6×10^4 ($\simeq$ 12.8 h)

Table 3.1: Mean serial execution time for 1200 FDTD time iterations (*sec*).

(25.41% of the total execution time); *c*) Discrete Fourier Transform values of the electric field (19.45% of the total execution time). The results obtained varying the number of threads are listed in Tab. 3.2. Inside each SMP node, the CPUs can *share* a common address space using the multithreading OpenMP method [68, 69]. So one can specify more processes than the physical processors for tuning up the parallel code before running it on a bigger machine.

Nr. of threads	T_1	T_2-T_3
2	1.5×10^4 ($\simeq$ 4.2 h)	2.1×10^4 ($\simeq$ 5.8 h)
3	1.1×10^4 ($\simeq$ 3.1 h)	1.8×10^4 ($\simeq$ 5.0 h)
4	9.8×10^3 ($\simeq$ 2.7 h)	1.3×10^4 ($\simeq$ 3.6 h)
5	8.8×10^3 ($\simeq$ 2.4 h)	1.1×10^4 ($\simeq$ 3.1 h)
6	8.0×10^3 ($\simeq$ 2.2 h)	9.1×10^3 ($\simeq$ 2.5 h)
7	6.6×10^3 ($\simeq$ 1.8 h)	8.7×10^3 ($\simeq$ 2.4 h)
8	5.9×10^3 ($\simeq$ 1.6 h)	8.4×10^3 ($\simeq$ 2.3 h)

Table 3.2: Mean SP5 OpenMP execution times for 1200 time iterations (*sec*).

3.2.2 MPI parallel runs. Communication overhead.

The T_1, T_2 *and* T_3 test configurations ran for 1200 time iterations using the previously described *parallel* MPI implementation of the subgridding algorithm [25], with a varying number of processors and different sublattice decompositions, i.e., slicing directions. Actually, the sublattice decompositions here considered are a “longitudinal” one, made of *s*-slices which cut the subdomains through planes *perpendicular* to the body or antenna height (yz planes in Fig. 3.6, corresponding to a x slicing direction) and a “transversal” one, made of s-slices which cut the subdomains through planes *parallel* to the body or antenna height (xy planes in Fig. 3.6, corresponding to a z slicing direction). The former, hereafter referred to as the Longitudinal Slicing (LS, see left of Fig. 3.7), involves an interface exchange area between neighboring s-slices of 252×147 (human) or 154×112 (antenna) Yee cells. The latter, hereafter referred to as the Transversal Slicing (TS, see right of Fig. 3.7), involves an interface exchange area between neighboring s-slices of 875×252 (human) or 700×154 (antenna) Yee cells. As to the coarse cell s-slice thicknesses $\mathcal{T}_{\ell mq} = T_{\ell mq}/R_{\ell m}$, their

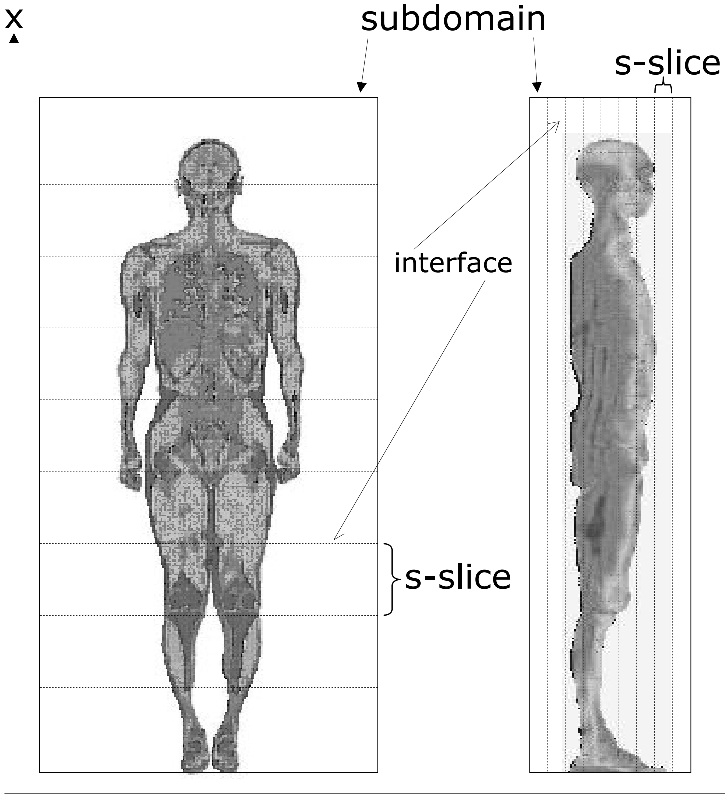


Figure 3.7: Schematic of the human male model LS (left) and TS (right) slicings.

values are all chosen to be about the same, within ±1 or ±2. This way, we are trying to avoid large unbalances in the processor loading. For example, in the TS slicing of the human body sublattice with $S_{\ell m} = 5$ s-slices, we alternate 3 s-slices of thickness 5 with 2 s-slices of thickness 3. This gives a total of $C_{\ell m} = 21$ coarse cells which amount, for a refinement factor of $R_{\ell m} = 7$, to the given 147 refined cells along the z-axis. In the human body LS slicing with $S_{\ell m} = 35$ s-slices, we shuffle 20 s-slices of thickness 4 with 15 s-slices of thickness 3, for a total of $C_{\ell m} = 125$ coarse cells which amount, for a refinement factor of $R_{\ell m} = 7$, to the given 875 refined cells along the x-axis. There is, however, a lower limit for the s-slice thickness. A thickness $\mathcal{T}_{\ell m q} < 2$ ($1 \leq q \leq S_{\ell m}$, for fixed ℓ and m) would make inapplicable the spatial filtering procedure [25] at the coarse/fine interface. As can be seen from Fig. 3.8 the T_1 configuration has been tested for a wide range of s-slice numbers and for both LS and TS decompositions. A number of s-slices equal to 0 means *serial* execution: the values are taken from Tab. 3.1. Parallel MPI executions start with a number of s-slices at least equal to 1, which means a single embedding coarse FDTD lattice — the z-slice — and an embedded *uncut* refined FDTD sublattice, i.e., two processes which communicate between them at the coarse/fine interface. One sees LS is faster than TS. This is due to the bigger amount of data passed between processes in TS slicing. From Fig. 3.8 it is also apparent the saturation effect due to the communication overhead with increasingly P. In Fig. 3.9 the T_2 and T_3 models are compared for LS only. They are more complex than T_1, including the extra antenna subdomain. For T_2, P is given by the number read on the horizontal axis plus *one* and for T_3, P is given by the horizontal axis value plus *two*. This explains the faster behavior of the T_3 model compared with that of T_2. Both Fig.s 3.8 and 3.9 evidence fluctuations due to the non-exclusive access to the computing machine.

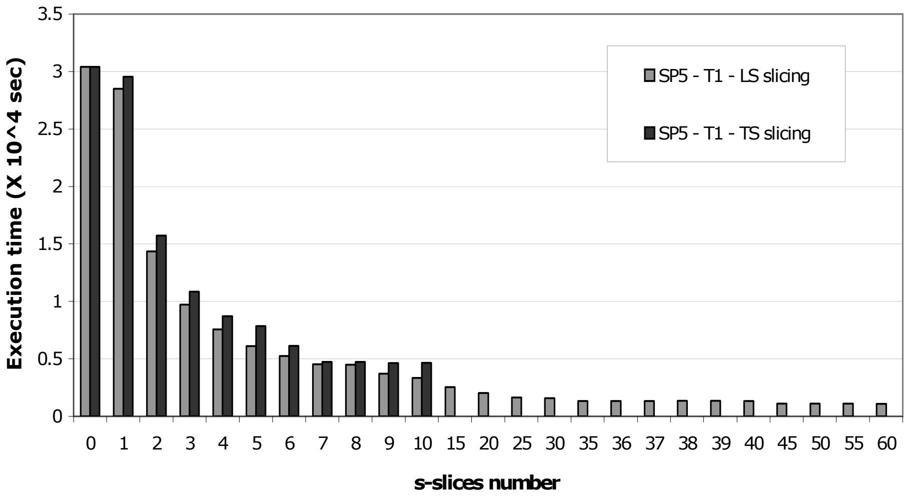


Figure 3.8: LS-TS comparison for the T_1 test model on SP5.

3.2.3 Scalability analysis.

This Subsection deals with the analysis of the expected performances of the proposed MPI parallel implementation of the subgridding algorithm [25] and their comparison with the numerical experiments in the previously two Subsections. Let Ω be the *input size* [63] of the FDTD model, which in our case is given by the pair:

$$\Omega = (\Omega_c \, , \Omega_s) = (\Omega_c \, , \sum_{i=1}^{\mathcal{N}} \Omega_{s,i})$$

where Ω_c refers to the outer embedding coarse grid size and $\Omega_{s,i}$ to the ith sublattice refined grid size with $\mathcal{N} \geq 1$ ($\mathcal{N} = \sum_{\ell=1}^{L} M_\ell$) the total sublattice number inside it. Then:

$$\Omega_c = N'_x \times N'_y \times N'_z$$

where primed quantities include the extra 20 + 20 cells needed for the absorbing boundary conditions [35], and:

$$\Omega_{s,i} = n_{x,i} \times n_{y,i} \times n_{z,i}$$

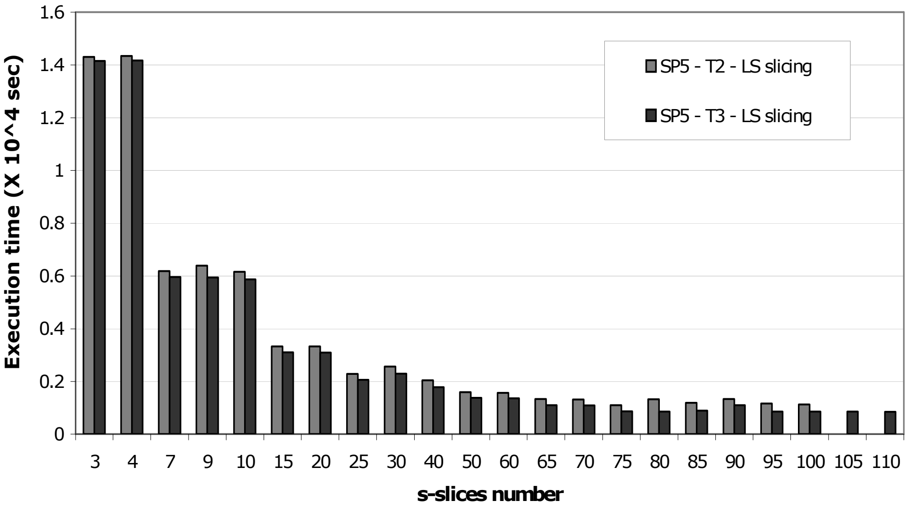


Figure 3.9: T_2-T_3 test models comparison for LS on SP5.

where $n_{\rho,i}$ are refined sizes ($\rho = x, y, z$ and $i = 1, \ldots, \mathcal{N}$). Let $T_\sigma(\Omega)$ and $T_\pi(\Omega, P)$ be the serial and parallel running times respectively (the latter referring to the slowest of $P \geq 2$ processes on separate physical processors) of the subgridding algorithm [25]. Important parameters of interest to measure performances ([63, 64, 70, 71]) are the *speedup* $\mathcal{S}$:

$$\mathcal{S}(\Omega, P) \equiv \frac{T_\sigma(\Omega)}{T_\pi(\Omega, P)} \tag{3.1}$$

and *efficiency* of process utilization $\mathcal{E}$:

$$\mathcal{E}(\Omega, P) \equiv \frac{\mathcal{S}(\Omega, P)}{P} = \frac{T_\sigma(\Omega)}{P \times T_\pi(\Omega, P)}. \tag{3.2}$$

Taking into account message-passing communication overhead but, as already pointed out, discarding any input/output or initialization code time contributions, we have [63]:

$$T_\sigma(\Omega) = T_\sigma^{calc}(\Omega) \tag{3.3}$$

and

$$T_\pi(\Omega, P) = T_\pi^{calc}(\Omega, P) + T_\pi^{comm}(\Omega, P)\ . \tag{3.4}$$

Moving data among processes is a task which is a priori more expensive than local calculation times. In fact, the cost of a single `Send/Recv` pair with α `floats` of data transmitted may be written as [63, 64]:

$$\tau_{la} + \alpha \times \tau_{bw} \tag{3.5}$$

where τ_{la} is the *latency time* (a start-up time needed to initiate a message) of the MPI communication system, while τ_{bw} depends inversely on the communication *bandwidth* and its units in (3.5) are *sec*/`|float|`.Commonly one has: $\tau_{la} \gg \tau_{bw} \gtrsim \tau_{fp}$, where τ_{fp} is the typical cost of an arithmetic floating point operation. It is also useful to introduce a P processes *total overhead* T_o with respect to ideal parallelism:

$$T_o(\Omega, P) \equiv T_\pi(\Omega, P) - \frac{T_\sigma(\Omega)}{P} \,. \tag{3.6}$$

Through T_o, (3.2) can be rewritten as:

$$\mathcal{E}(\Omega, P) = \left[1 + P\frac{T_o(\Omega, P)}{T_\sigma(\Omega)}\right]^{-1} \,. \tag{3.7}$$

According to [63] we will then say that a parallel program is *scalable* if it, as Ω and P vary, admits a suitable $T_o(\Omega, P)$ adjustment that keeps efficiency $\mathcal{E}$ constant and as close to 1 as possible. The complexity of the bulk FDTD algorithm plus the electric field DFT updating code is $O(N^3)$, where N is a typical grid linear size. Since 39 or 42 (single precision) floating point operations (sums and multiplications) are required per FDTD time iteration per vacuum and dielectric/metallic cell respectively. We have for an average estimate of T_σ in (3.3) (accounting for about the 94% of the total computation time):

$$T_\sigma(\Omega) \equiv T_\sigma^{calc}(\Omega) \simeq \left[42 \sum_{i=1}^{\mathcal{N}} R_i \, \Omega_{s,i} + 39 \, \Omega_c\right] \tau_{fp} \,, \tag{3.8}$$

where R_i is the refinement factor of the ith sublattice (for each coarse FDTD time iteration, there are R_i FDTD time iterations inside the ith sublattice). Turning now to T_π in (3.4) and differentiating between coarse grid calculations (c extra subscript) and refined grids (i.e. sublattices) calculations (s extra subscript), for a parallel code running with P processes ($P \geq \mathcal{N} + L$) we can write:

$$T^{calc}_{\pi,c}(\Omega, P) \simeq \frac{39\,\Omega_c\,\tau_{fp}}{L} \tag{3.9}$$

and

$$T^{calc}_{\pi,s}(\Omega, P) \simeq \frac{1}{P-L}\left[42\sum_{i=1}^{\mathcal{N}} R_i\,\Omega_{s,i}\right]\tau_{fp} \tag{3.10}$$

respectively. By introducing the average number Γ of s-slices per sublattice:

$$\Gamma \simeq \frac{P-L}{\mathcal{N}}\,,$$

(L will depend on P through $\mathcal{N}$ and Γ), we deduce the following three contributions for the communication time overhead per FDTD iteration. The first one:

$$T^{comm}_{\pi,c}(\Omega, P) = 2 \times 4\,(L-1)\left(\tau_{la} + N'_x\,N'_y\,\tau_{bw}\right) \tag{3.11}$$

is for the coarse grid computations (labels 1 and 2 in the left column of Fig. 3.5). The second one:

$$T^{comm}_{\pi,s}(\Omega, P) = 2 \times 4\,(\Gamma-1)\sum_{i=1}^{\mathcal{N}} R_i\,(\tau_{la} + n_{\alpha,i}\,n_{\beta,i}\,\tau_{bw}) \tag{3.12}$$

for the refined grid computations (labels $1'$ and $2'$ in the right column of Fig. 3.5). The third contribution is:

$$T^{comm}_{\pi,both}(\Omega, P) = 6\left[\mathcal{N}\,\Gamma\,\tau_{la} + \sum_{i=1}^{\mathcal{N}}\left(\frac{n_{x,i}\,n_{y,i} + n_{x,i}\,n_{z,i} + n_{y,i}\,n_{z,i}}{R_i^2}\right)\tau_{bw}\right] \tag{3.13}$$

and affects both the coarse and refined grid computations (labels AA' and BB' in Fig. 3.5). As previously pointed out, these data are passed efficiently by means of suitable data templates. We thus have:

$$T_{\pi,c}(\Omega,P) = T^{calc}_{\pi,c}(\Omega,P) + T^{comm}_{\pi,c}(\Omega,P) + T^{comm}_{\pi,both}(\Omega,P)$$

and

$$T_{\pi,s}(\Omega,P) = T^{calc}_{\pi,s}(\Omega,P) + T^{comm}_{\pi,s}(\Omega,P) + T^{comm}_{\pi,both}(\Omega,P)\ .$$

We now formulate the general criterion which would give our parallel subgridding algorithm optimal scalability:

$$\boxed{T_{\pi,c}(\Omega,P) \overset{(1)}{=} T_{\pi,s}(\Omega,P) \overset{(2)}{=} \frac{T_\sigma(\Omega)}{P\,\mathcal{E}_t}} \tag{3.14}$$

where $\mathcal{E}_t$ is a target value (necessarily $\mathcal{E}_t < 1$), for the attainable efficiency $\mathcal{E}(\Omega,P)$ (3.2) or (3.7). The first equation of system (3.14) will make all of the P processes (of both kind, those running z-slice calculations and those running s-slice calculations) to take, on average, an equal running time. The second equation in (3.14), on the other hand, should reduce the overhead $T_o(\Omega,P)$ — as defined in (3.6) — according to the chosen value $\mathcal{E}_t$ for $\mathcal{E}(\Omega,P)$. Tab. 3.3 summarizes the values for the size parameters just introduced. From them one can easily deduce that (3.8) evaluates to:

$$T_\sigma(\Omega) \simeq 9.71 \times 10^{09} \times \tau_{fp} \tag{3.15}$$

for the T_1 test model. For the T_2 -T_3 test model it evaluates to:

$$T_\sigma(\Omega) \simeq 1.33 \times 10^{10} \times \tau_{fp}\ . \tag{3.16}$$

Comparing with the results of Tab. 3.1, which hold for 1200 FDTD time iterations, empirical estimates for τ_{fp} are obtained. We also got values for τ_{la} and τ_{bw} by looking at elapsed times when running ad hoc MPI pieces of code. The values found are:

Size Param.	T$_1$	T$_2$	T$_3$
N'_x	$20+200+20$	$20+200+20$	$20+200+20$
N'_y	$20+100+20$	$20+100+20$	$20+100+20$
L	1	1	2
N'_z	$20+100+20$	$20+100+20$	$(20+50)+(50+20)$
$\mathcal{N}$	1	2	2
$n_{x,1}$	875	875	875
$n_{y,1}$	252	252	252
$n_{z,1}$	147	147	147
R_1	7	7	7
$n_{x,2}$	–	700	700
$n_{y,2}$	–	154	154
$n_{z,2}$	–	112	112
R_2	–	7	7

Table 3.3: Size parameter values for the paper test models.

	τ_{fp}	τ_{la}	τ_{bw}
SP5	$\simeq$ 2.5 nsec	$\simeq$ 40 μsec	$\simeq$ 2 nsec/`float`
BCX	$\simeq$ 4.2 nsec	$\simeq$ 200 μsec	$\simeq$ 4 nsec/`float`
WS	$\simeq$ 3.0 nsec	–	–

Table 3.4: Floating point, latency and bandwidth time

We then proceed to extrapolate, accordingly to the above analysis, the expected performances of the algorithm proposed by applying it to the three general test configurations T$_1$, T$_2$ and T$_3$ previously mentioned. We point out here that, with the values in Tab. 3.3 and Tab. 3.4, we are not able to satisfy the first constraint in the general criterion (3.14) and always get:

$$T_{\pi,s}(\Omega, P) > T_{\pi,c}(\Omega, P) ,$$

meaning the coarse embedding grid code runs faster than that for the refined grids, with a limiting parallel execution time, per FDTD iteration, given by the middle member $T_{\pi,s}$ of (3.14). For the T$_1$, T$_2$ cases it has the general form:

$$T_\pi(\Omega, P) \equiv T_{\pi,s}(\Omega, P) = \frac{a}{P-1} + b\,(P-1) + c\,, \tag{3.17}$$

while for the T$_3$ case it has the general form:

$$T_\pi(\Omega, P) \equiv T_{\pi,s}(\Omega, P) = \frac{a}{P-2} + b\,(P-2) + c\,. \tag{3.18}$$

where a, b, c are coefficients depending on the model size, on τ_{fb}, τ_{la} and τ_{bw}. Asymptotically both these two times grow linearly with P, thus indicating that the communication overhead ends up overwhelming the speedup achievable by reducing further and further the s-slice bulk sizes.

T$_1$ test model: here we have $\mathcal{N} = 1$ and $L = 1$ (thus implying $\Gamma = P - 1$), with $P \geq 2$. The coefficients in (3.17) amount to:

$$a = 9.53 \times 10^9 \times \tau_{fp}\, sec.$$

$$b = (56 \times \tau_{la} + 2074464 \times \tau_{bw})\ sec.$$

$$c = (-56 \times \tau_{la} + 10242 \times \tau_{bw})\ sec.$$

By means of (3.15) and (3.17), expressions for the efficiency $\mathcal{E}(\Omega, P)$ and the speedup $\mathcal{S}(\Omega, P) = P\,\mathcal{E}(\Omega, P)$ can be calculated as functions of P, starting from their definitions (3.1), (3.2). Graphs of $T_\pi(\Omega, P)$, efficiency and speedup are shown in Figs. 3.10, 3.11 and 3.12 under the "theoretical" label. From those expressions, values for $P_{\mathcal{S}_{max}}$, $\mathcal{E}_{max}$, $P_{\mathcal{E}_{max}}$ and $\mathcal{S}_{max}$ are obtained by deriving with respect to P. Omitting cumbersome calculations, we get for the SP5 and BCX machines the following results:

	$P_{\mathcal{E}_{max}}$	$\mathcal{E}_{max}$	$P_{\mathcal{S}_{max}}$	$\mathcal{S}_{max}$
SP5	$\simeq 13$	$\simeq 0.91$	$\simeq 61$	$\simeq 30.8$
BCX	$\simeq 11$	$\simeq 0.89$	$\simeq 45$	$\simeq 23.0$

to which corresponds the following times per FDTD iteration:

	$T_\pi(P_{\mathcal{S}_{max}})$	$T_\pi(P_{\mathcal{E}_{max}})$
SP5	$\simeq 0.79$ sec	$\simeq 2.0$ sec
BCX	$\simeq 1.8$ sec	$\simeq 4.2$ sec

It would be possible to get a better efficiency, say $\mathcal{E} = 0.95$, by increasing ad hoc the size Ω_c of the coarse embedding grid and/or reducing the size $\Omega_{s,1}$ of the embedded refined sublattice, to force the $T_{\pi,s}$ and $T_{\pi,c}$ equalization in the first equation of system (3.14).

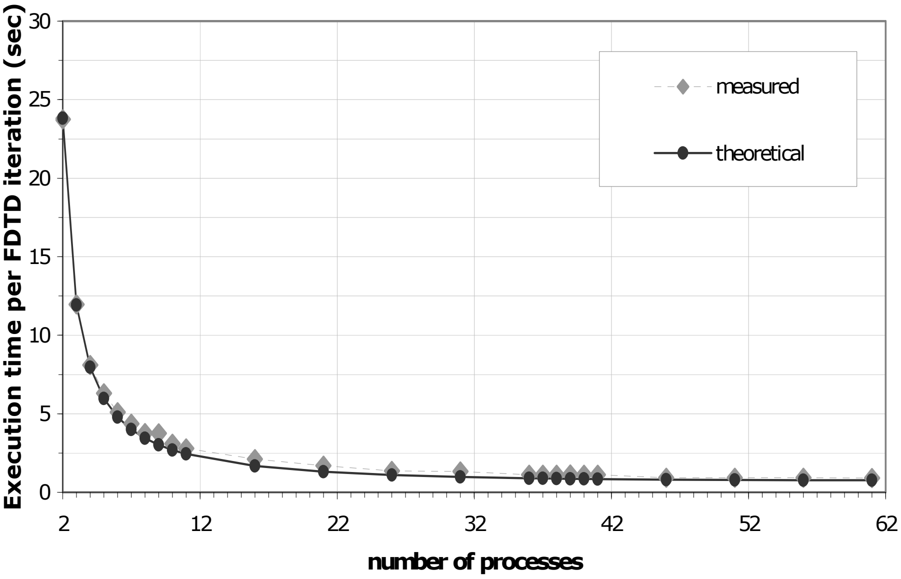


Figure 3.10: T_1 test model timing on SP5.

T$_2$ test model: here we have $\mathcal{N} = 2$ and $L = 1$ (thus implying $\Gamma = (P-1)/2$), with $P \geq 3$. The set of coefficients in (3.17) is now:

$$a = 1.31 \times 10^{10} \times \tau_{fp}\; sec.$$

$$b = (62 \times \tau_{la} + 1520176 \times \tau_{bw})\; sec.$$

$$c = (-112 \times \tau_{la} - 2968154 \times \tau_{bw})\; sec.$$

Going on as in the T_1 case, we get:

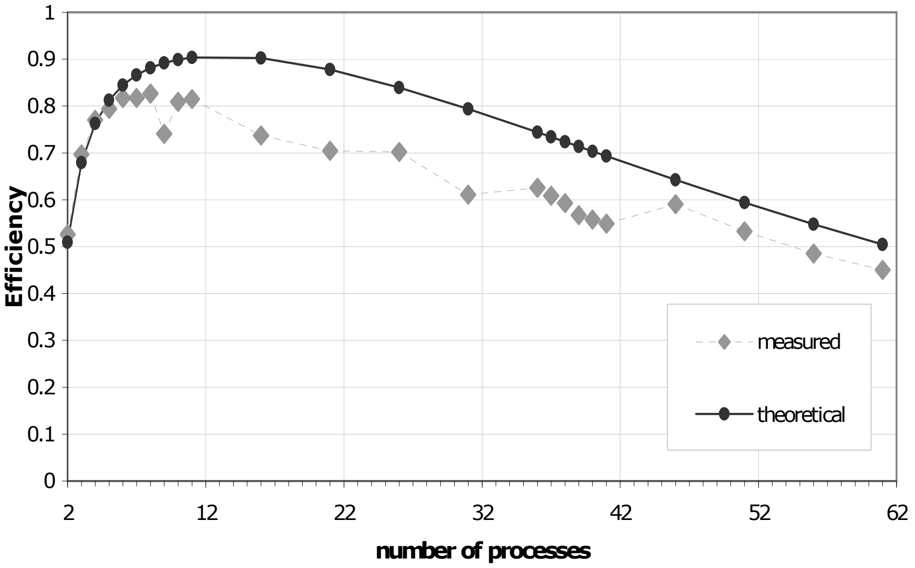


Figure 3.11: T_1 test model efficiency on SP5.

	$P_{\mathcal{E}_{max}}$	$\mathcal{E}_{max}$	$P_{\mathcal{S}_{max}}$	$\mathcal{S}_{max}$
SP5	$\simeq 16$	$\simeq 0.92$	$\simeq 78$	$\simeq 39.5$
BCX	$\simeq 13$	$\simeq 0.90$	$\simeq 56$	$\simeq 28.1$

	$T_\pi(P_{\mathcal{S}_{max}})$	$T_\pi(P_{\mathcal{E}_{max}})$
SP5	$\simeq 0.84$ sec	$\simeq 2.3$ sec
BCX	$\simeq 2.0$ sec	$\simeq 4.9$ sec

Relevant graphs are shown in Figs. 3.13, 3.14 and 3.15.

$\underline{\text{T}_3 \text{ test model:}}$ here we have $\mathcal{N} = 2$ and $L = 2$ (thus implying $\Gamma = (P-2)/2$), with $P \geq 4$. The set of coefficients in (3.18) is now:

$$a = 1.31 \times 10^{10} \times \tau_{fp} \; sec.$$

$$b = (62 \times \tau_{la} + 1520176 \times \tau_{bw}) \; sec.$$

$$c = (-112 \times \tau_{la} - 2968154 \times \tau_{bw}) \; sec.$$

Going on as in the T_1 case, we get:

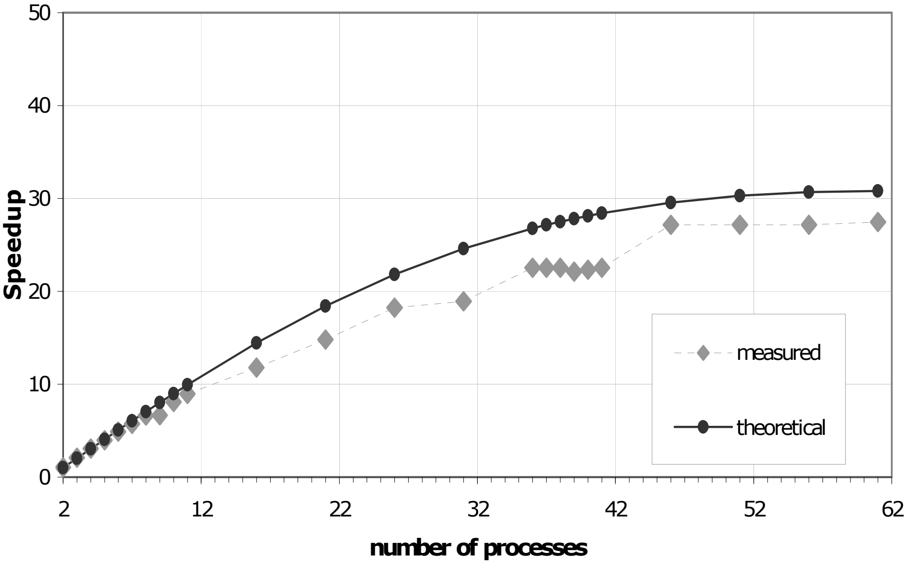


Figure 3.12: T$_1$ test model speedup on SP5.

comm.	$P_{\mathcal{E}_{max}}$	$\mathcal{E}_{max}$	$P_{\mathcal{S}_{max}}$	$\mathcal{S}_{max}$
SP5	$\simeq 19$	$\simeq 0.87$	$\simeq 79$	$\simeq 39.5$
BCX	$\simeq 16$	$\simeq 0.84$	$\simeq 57$	$\simeq 28.1$

comm.	$T_\pi(P_{\mathcal{S}_{max}})$	$T_\pi(P_{\mathcal{E}_{max}})$
SP5	$\simeq 0.84$ sec	$\simeq 1.98$ sec
BCX	$\simeq 1.98$ sec	$\simeq 4.28$ sec

Relevant graphs are shown in Figs. 3.16, 3.17 and 3.18. Also for T$_2$ and T$_3$ configurations it would be possible to calculate more efficient sizings, as has been mentioned for the T$_1$ case, by forcing $T_{\pi,s}$ and $T_{\pi,c}$ equalization in the first equation of system (3.14). To end this Subsection, we point out that the scattering and jaggedness in the measured "experimental" data shown in the graphs along with their "theoretical" counterparts are due to non-reproducibility, mainly from a lack of exclusive access to the computing machine. The constant loading of the computing machine by extraneous jobs concretizes in an extra amount of overhead time T_o, other than that arising simply from the non instantaneous communication

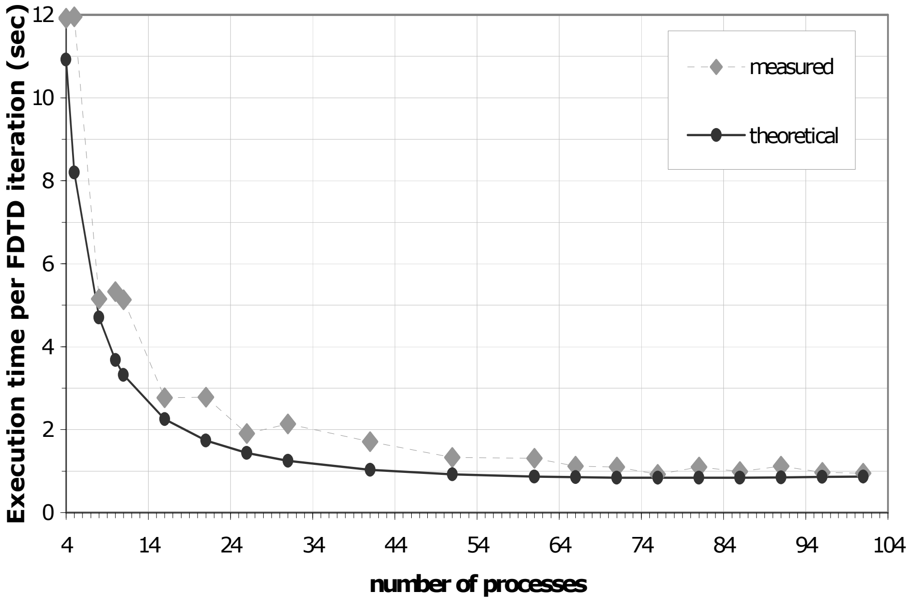


Figure 3.13: T_2 test model timing on SP5.

between processes. Although unpredictable (and apart the possibly uncorrected estimates for τ_{la}, τ_{bw} and τ_{fp}), such effect let one to appreciate the fairly good efficiency and scalability of the proposed parallelization algorithm, once the most suitable amount of slicing has been chosen for a given geometry and model complexity. There is no limitation on the number of slices decomposing a given domain, but it is shown that, for a given initial size, there is an optimum of slices beyond which communication overwhelms computation.

3.2.4 Load balancing. Nonblocking communication.

With a better *load balancing* — a fine tuning of the amount of work made by each process — one may hope to lower further the overhead time T_o (3.6), by reducing the *idle time* contribution due to a lack of synchronism between processes execution. A good way to achieve such a reduction is, if the program structure allows it, by overlapping computation and communication. This is accomplished by using the nonblocking `MPI::Isend`

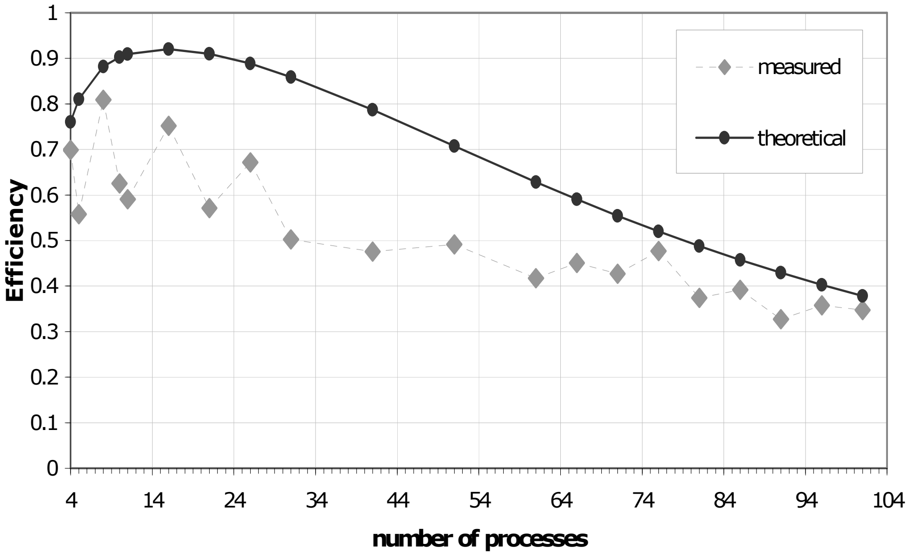


Figure 3.14: T_2 test model efficiency on SP5.

and MPI::Irecv routines [63, 64] and [70, 71], were the I prefix stands for "Immediate" return, in place of the more common blocking ones, i.e., MPI::Send and MPI::Recv. As already pointed out, however, the FDTD algorithm iteration cycle is poorly prone to such an overlapping, because the field values are passed among the processes in a domain decomposition context, and are immediately used in the difference equations. As a consequence, a little gain is got from the use of the nonblocking communication functions. The better strategy remains an optimal choice of the size parameter values, so that the processes happen to be evenly loaded. The only point in the task list of Fig. 3.5 which could suit a "small" communication/computation overlapping is that labeled B' in the right column. After the Recv call, an R sequence of subcycles (R being the refinement factor) iterating on the refined sublattice grid is started. The *first* of such iterations could indeed start immediately without waiting the Recv completion, at least until the space-time interpolation routine is called. Hence the Recv completion can be slightly forward shifted. The better way to load balancing the code is by the fulfillment, as much as possible, of the

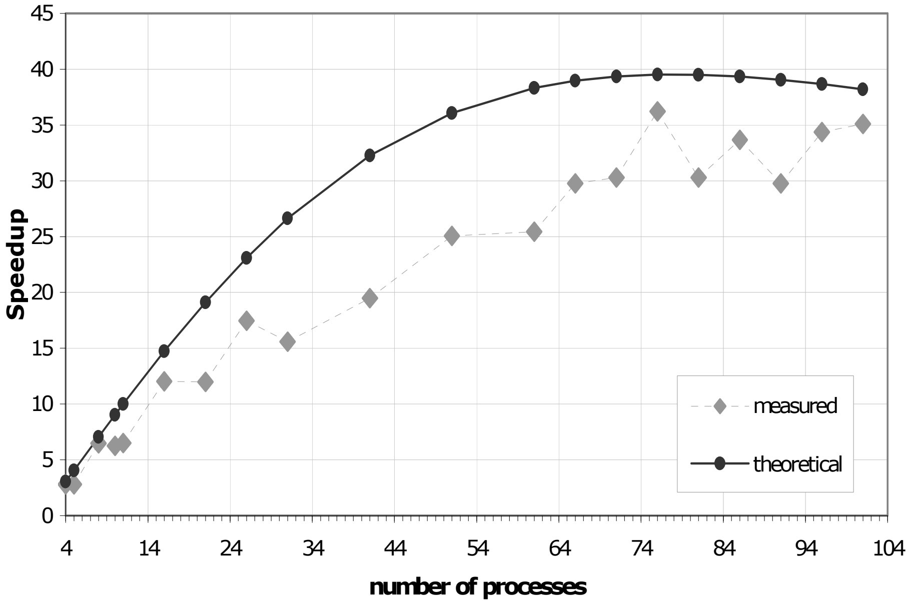


Figure 3.15: T_2 test model speedup on SP5.

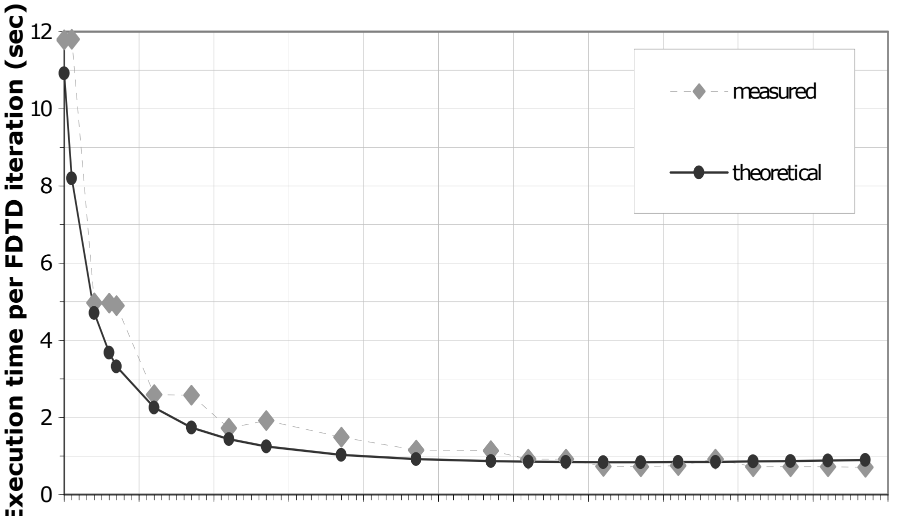


Figure 3.16: T_3 test model timing on SP5.

criterion (3.14), particularly with respect to the size of the coarse grid relatively to the embedded refined ones, because a great amount of inefficiency is seen by processes running coarse grid computations spending a lot of time in communication. Keeping in mind that the subgridding is fixed, i.e.

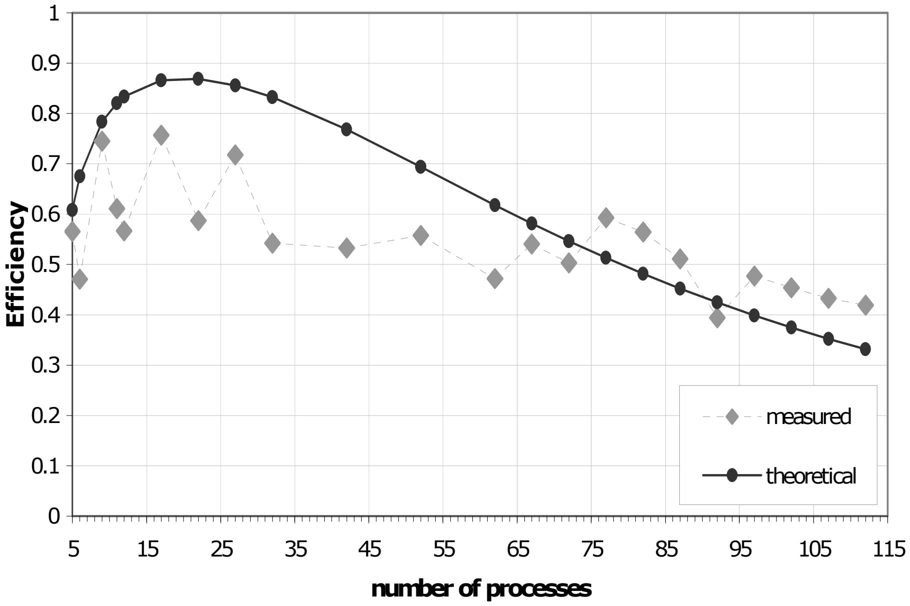


Figure 3.17: T_3 test model efficiency on SP5.

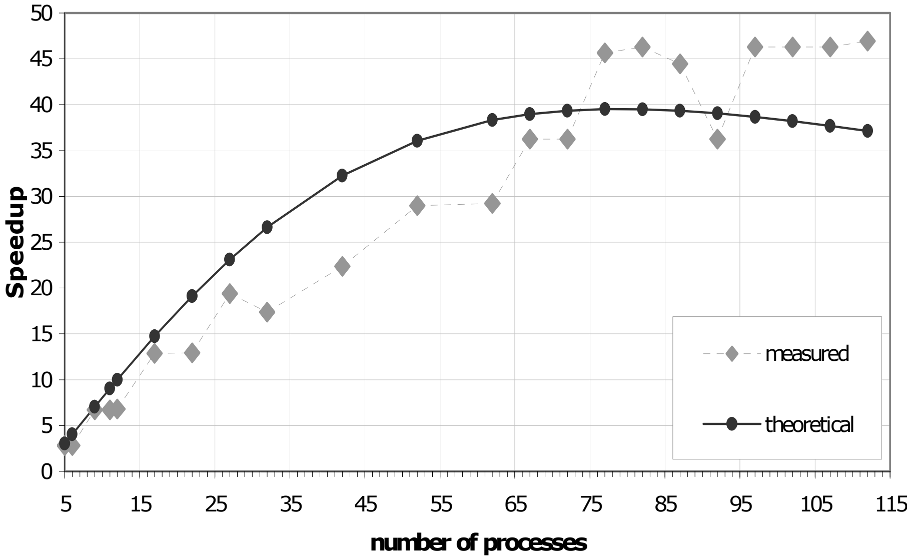


Figure 3.18: T_3 test model speedup on SP5.

not dynamically varying during the code runs, a rough criterion for load balancing, given the refinement factor R, is that the flop number in the refined grid is R^4 greater than the flop number in the coarse one.

3.3 A large FDTD numerical experiment

In this Section we apply the above described parallel version of the subgridding algorithm [25] to a moderately large FDTD model, made of five standing men on a perfectly conducting ground plane and exposed to the e.m. near-field of an antenna (Fig.3.19). The constitutive elements of the model are the same as those previously used in the test configurations T_1, T_2 or T_3, but now arranged to give an estimated *serial* execution time of $\simeq$ 216 sec per FDTD iteration on BCX, and of $\simeq$ 129 sec per FDTD time iteration on SP5. The modeled space volume has sizes of: 2.80 m (Height, along x-axis) × 1.40 m (Width, along y-axis) × 2.80 m (Depth, along z-axis) and thus, assuming that 1200 FDTD iterations suffice to stimulate every part of the computational domain by the excitation antenna signal — for good frequency domain results via Semi-Discrete Fourier transforms —, about 72 hours (3 days) on BCX and about 43 hours (1.8 days) on SP5 should be required for completing the calculations with conventional sequential programming. The *parallel* execution of the model uses 2 z-slices (like the T_3 test), each taking 100 coarse cells along the z-axis. Each z-slice contains 3 sublattices, each refined by a factor of 7 with respect to the coarse cubic cells which have an edge of 1.4 cm. Five sublattices contain, at a refined level, the relative dielectric constant and conductivity (Siemens/m) data values for the standing man ($875 \times 252 \times 147$ cubic cells), conformed to the anatomical structures as described in [65]. The remaining one contains data for the antenna metallic structures ($700 \times 154 \times 112$ cubic cells). On the final field values, we also perform some energetic checks, both to assess the reliability of the numerical calculations for dosimetric analysis and as an inspection of the goodness of the numerical solution, being lacking for such a complex model the analytic reference solution. For pure MPI code the running time results are shown in Tab. 3.5, from which we can see how the parallelization technique here proposed allows for a strong shortening of the execution FDTD times and with a fairly good scalability.

	# proc.s: $105 + 2$	# proc.s: $140 + 2$	# proc.s: $350 + 2$
BCX	$4,137$ ($\simeq 1.15$ h)	$3,862$ ($\simeq 1.07$ h)	$1,934$ ($\simeq 0.54$ h)
SP5	$2,470$ ($\simeq 0.69$ h)	N/A	N/A

Table 3.5: MPI parallel execution times for 1200 FDTD iterations (*sec*)

Emitted Poynting (antenna-refined)	79.96	79.96
Escaped Poynting/Kirchhoff (outer-coarse)	43.57	43.56
Absorbed Poyn./Dissip. (sublatt.1-refined)	6.03	6.02
Absorbed Poyn./Dissip. (sublatt.2-refined)	1.44	1.44
Absorbed Poyn./Dissip. (sublatt.3-refined)	1.55	1.55
Absorbed Poyn./Dissip. (sublatt.4-refined)	12.72	12.63
Absorbed Poyn./Dissip. (sublatt.5-refined)	12.22	12.13
Subtotal Absorbed (1 to 5)	33.96	33.77
Subtotal Absorbed + Escaped	77.53	77.33
Deviation from Emitted	2.43 (3%)	2.63 (3.2%)

Table 3.6: Power balance (Watt) for the large FDTD numerical experiment

In Tab. 3.6 it is summarized the mentioned power balance check at the antenna working frequency of 1862.5 MHz (human relative dielectric constants and conductivities are given for this frequency value too). The first row gives the total e.m. power emitted by the antenna, in equilibrium with the loads represented by the five humans in front to it. That value is the real part of the complex Poynting vector flux on a parallelepiped surface completely surrounding the antenna inside the refined sublattice (five refined cells apart from its outer surface). The second row gives the e.m. power leaving the computational domain. Of the two value in the row, the first is the real part of the complex Poynting vector flux on a parallelepiped surface completely surrounding the antenna and the humans inside the coarse lattice (five coarse cells apart from its outer surface). The second value is obtained by means of a near-to-far field transformation — a Kirchhoff integral formula [72] implementation — and an integration over the 4π steradian angle at an extrapolated infinite distance from the center of the model. Rows from the third to the seventh have values of the e.m. power absorbed by the human loads. The absorbed power can be calculated in either of two ways. The first value is obtained as the real part of the complex Poynting vector *inflow* on a parallelepiped surface completely surrounding the human inside the refined sublattice (five refined cells apart from its outer surface). The alternative is to calculate the power dissipation in the subregions where the electric conductivity $\sigma \neq 0$, as $\sigma\|\vec{E}\|^2/2$ and summing (integrating) up. The eighth row gives the total amount of e.m. absorbed power. The ninth row gives the sum of the e.m. powers absorbed and leaving the volume space under consideration, a value which has to equal the e.m. power emitted from the antenna. As can be seen, a fairly good result is obtained (within the 3% of the emitted power reference), despite the single precision floating point format, the absorbing boundary conditions effect, the coarse/refined mesh sizes coupling (with filtering) effect, the 1200 FDTD iterations (which become $1200 \times 7 = 8400$ inside the refined sublattices) cumulative truncation error effect. Fig. 3.20 is a perspective view of three simultaneous mutually perpendicular planes from the coarse domain. The test has given good accuracy and performance results and the algorithm could be proposed for solving accurately the problems described in [73] and [74] with bigger sizes. There is no limitation on the number of slices decomposing a given domain, but it is shown that, for a given initial size, there is an optimum of slices beyond which communication overwhelms computation.

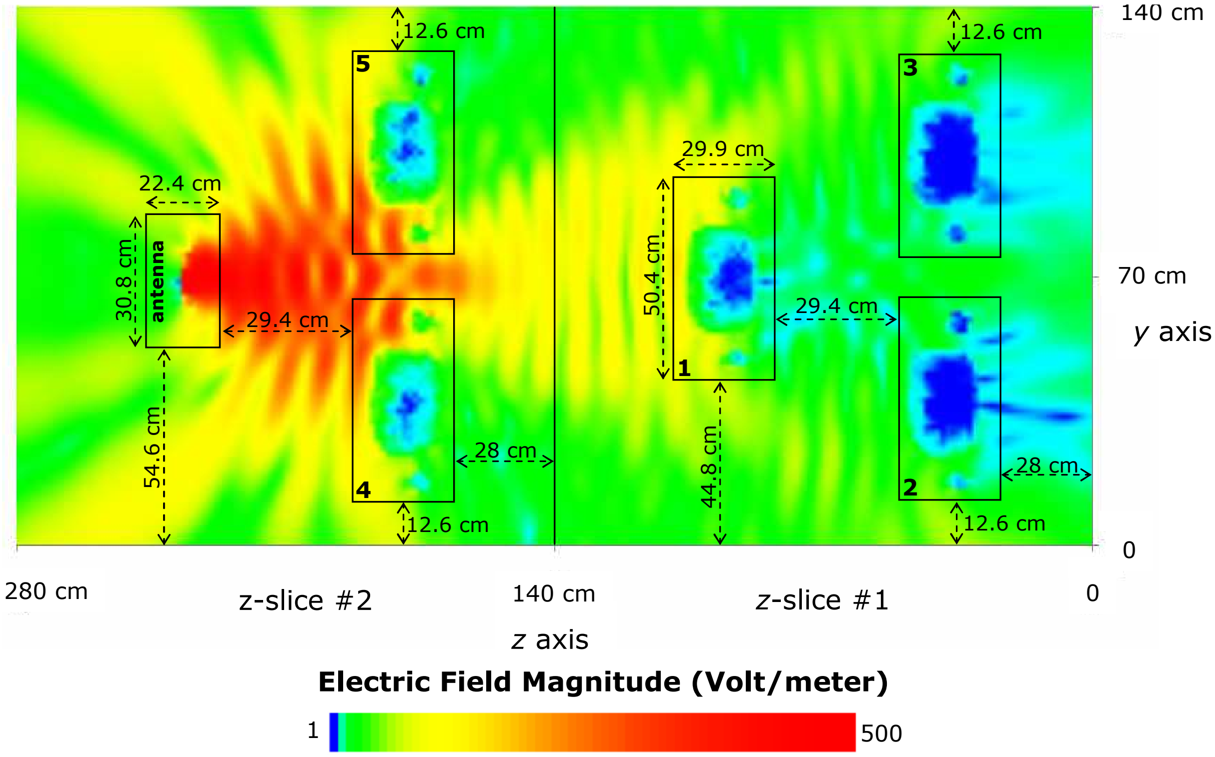


Figure 3.19: Top view of a plane from the outer coarse lattice with information about geometry and displacements of the various embedded refined sublattices for the antenna and the humans. Log. scale.

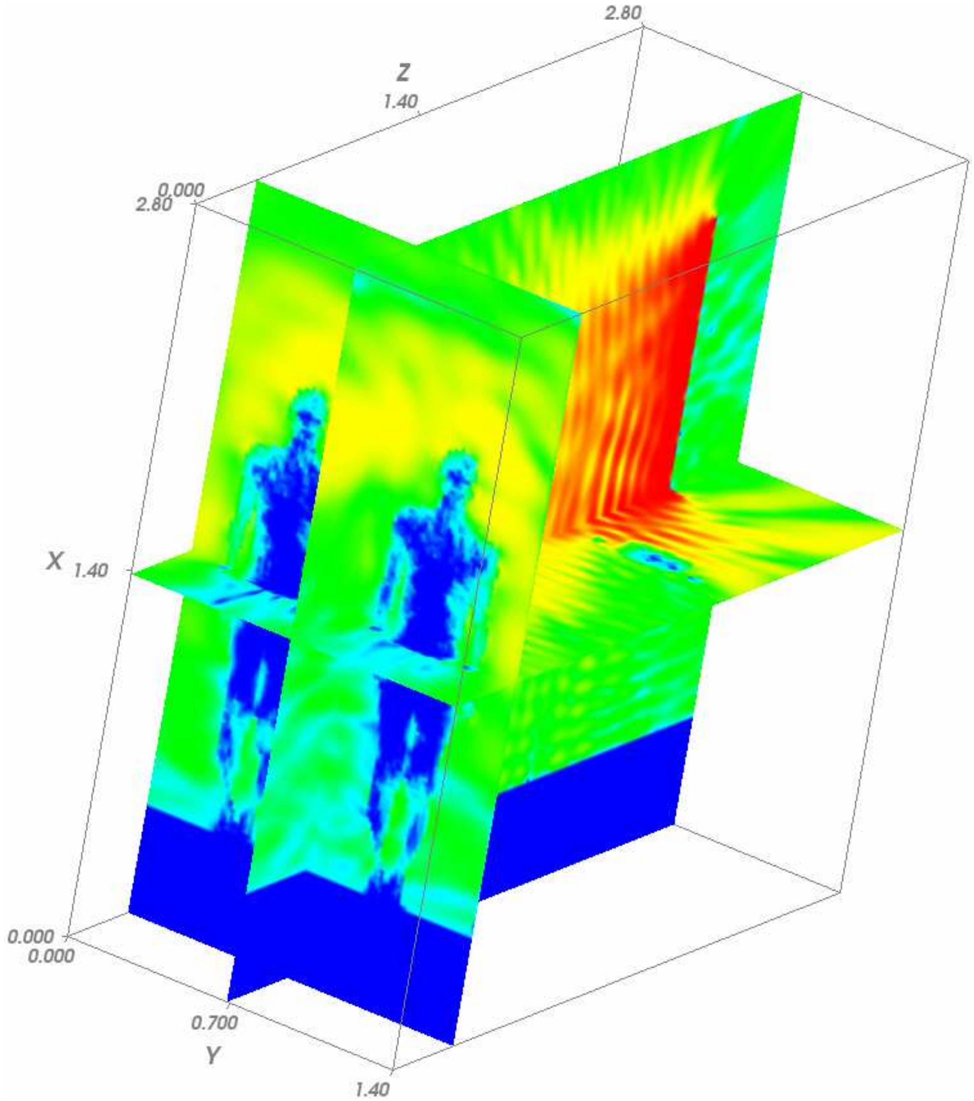


Figure 3.20: Three planes from the coarse grid. Perspective back view. Log. scale.

Chapter 4

Simulation of plasmonic nanostructures

4.1 Scattering by spherical nanoparticles

4.1.1 Theory of extinction

Scattering and absorption of light by a sphere is a well-known problem whose solution is given by the Lorentz-Mie theory [20]. Within this theory, the extinction, scattering and absorption cross sections for a spherical particle are

$$
\begin{aligned}
C_{ext} &= \frac{2\pi}{k^2}\sum_{n=1}^{\infty}(2n+1)Re(a_n+b_n), && (4.1)\\
C_{sca} &= \frac{2\pi}{k^2}\sum_{n=1}^{\infty}(2n+1)(|a_n|^2+|b_n|^2), && (4.2)\\
C_{abs} &= C_{ext}-C_{sca}, && (4.3)
\end{aligned}
$$

where $k = (2\pi N)/\lambda$, N is the refractive index of the medium around the particle and λ is the wavelength of the incident radiation. a_n and b_n

are defined as

$$a_n = \frac{m\psi_n(mkr)\psi_n'(kr) - \psi_n(kr)\psi_n'(mkr)}{m\psi_n(mkr)\chi_n'(kr) - \chi_n(kr)\psi_n'(mkr)}, \tag{4.4}$$

$$b_n = \frac{\psi_n(mkr)\psi_n'(kr) - m\psi_n(kr)\psi_n'(mkr)}{\psi_n(mkr)\chi_n'(kr) - m\chi_n(kr)\psi_n'(mkr)}, \tag{4.5}$$

where ψ and χ are Riccati-Bessel functions of the first and third kind, respectively. r is the radius of the particle, and $m = N_p/N$, where N_p is the complex refractive index of the particle [20]. When dealing with the behavior of nanoparticles illuminated by light in the visible range, the *small particle* approximation is often taken for granted, leading to stop at the lowest order term of the previous equations, which amounts to treat the particle as a simple dipole. As shown in [20], the cross sections (4.1), (4.2) and (4.3) can also be written as a power series in the adimensional parameter $x = 2\pi r/\lambda$; for wavelengths in the visible region and spheres with $r \geq 100nm$, $x \sim 1$. The dipole approximation, which is then valid only for very small particles ($x \ll 1$), is not appropriate and would neglect an important part of the scattering mechanism. Hence, in our analysis we will truncate the series at $n = 10$, a choice that guarantees satisfactory convergence of the results.

An efficiency parameter is usually defined as the ratio of the particle cross section over a surface which is the geometrical projection of the particle on a plane perpendicular to the incoming light. For a sphere it is

$$Q_i = \frac{C_i}{\pi r^2} \tag{4.6}$$

where $i = \{ext, sca, abs\}$.

Note that the efficiency parameter Q_{sca} defines a minimum distance between neighbor particles if they are arranged in an array. In fact, a sphere scatters the radiation contained in a circle of area $C_{sca} = \pi r^2 Q_{sca}$; if we require the effective area of neighboring particles not to overlap, the distance d between them should satisfy the condition $d > 2r\sqrt{Q_{sca}}$. Examining our data presented below, where in most cases $Q \sim 3 - 4$, we found a good agreement with the results reported in [75], where it is claimed that a 30% surface coverage is enough to scatter almost all of the incident radiation. Moreover, our results are not in conflict with those presented in [76], given the difference in the choice of the parameters.

Although the efficiency parameters contain all the dynamics of the extinction process, they are unrelated to the spectrum of the incoming light. In order to consider a realistic situation functional to the solar cell design, we introduce an averaged efficiency defined as

$$\hat{Q}_i = \frac{\int I(\lambda) Q_i(\lambda)\, \mathrm{d}\lambda}{\int I(\lambda)\, \mathrm{d}\lambda}, \tag{4.7}$$

where $I(\lambda)$ is the standard AM1.5G spectrum intensity [77] (Fig. 4.1) and the integration is extended from 300 to 1100nm. The figure reveals

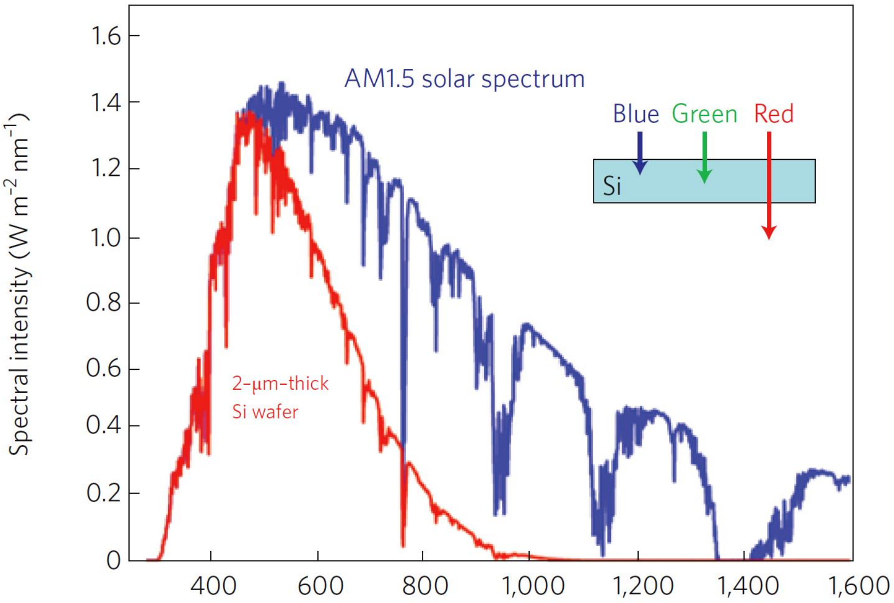


Figure 4.1: Solar spectrum together with a graph that indicates the solar energy absorbed by a crystalline Si film.

the high absorbance of a thin silicon layer in most of the optical frequency range. The silicon solar cell works in the infrared and red regions ($600 \div 1100 nm$). Here a thin silicon layer cannot absorb most of the radiation. Tipically thicker silicon layer are required for increasing the absorption also in this frequency range. This is not convenient from a business viewpoint. The presence of plasmonic nanostructures on the top of a thin silicon layer improves its performance for different reasons [78]:

- can prolonge the optical path of the photons due to the far-field scattering (Fig. 4.2a).
- can locally cause an increased absorption due to the near-field scattering and the resulting local field enhancement (Fig. 4.2b).
- can originate a direct injection of photo-excited carriers into the semiconductor (Fig. 4.2c).

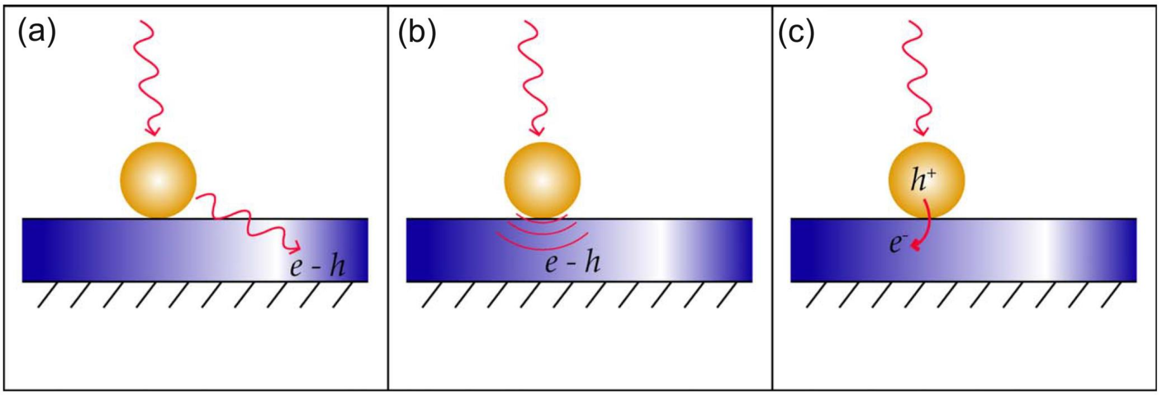


Figure 4.2: Different types of plasmonic nanoparticle effects on photovoltaic.

The new parameter $\hat{Q}_i$ weights the scattering efficiency of a sphere at each wavelength with the intensity of the incident radiation. $\hat{Q}_i$ is then a useful tool to compare different configurations and represents a sort of global response of the particle to a given illumination spectrum (since we worked on a set of discrete data, we used the trapezoidal formula for the evaluation of the integral). As a reference, for an opaque particle, whose cross section corresponds simply to its geometrical size, we have $Q_{ext} = 1$ all over the entire spectrum and consequently $\hat{Q}_{ext} = 1$. It is worth stressing that, though $\hat{Q}_i$ combines the effect of all the parameters and of the solar spectrum features, thereby allowing a quick comparison among several configurations, the design of specific solar cells should nonetheless consider non-averaged features like the position of the resonance peaks or their absolute magnitude.

Since scattering should be maximized and, at the same time, absorption should be minimized, a useful quantity to describe this feature is the albedo, defined as the fraction of the extinct light which is re-emitted as radiation. Written in percentage form the albedo is

$$\alpha = \frac{C_{sca}}{C_{ext}} \times 100 \; . \tag{4.8}$$

If the nanoparticles are placed on the front surface of a solar cell, we want the scattered radiation to be highly peaked in the forward direction. In principle, this feature could depend on the particle size, the surrounding medium and the wavelength of the incident radiation. The fraction of light scattered in the forward direction is

$$f_{sub} = \frac{\int_0^{2\pi} \int_0^{\pi/2} \vec{S}_s \cdot \hat{\vec{e}}_r r^2 \sin\theta \, \mathrm{d}\theta \, \mathrm{d}\phi}{\int_0^{2\pi} \int_0^{\pi} \vec{S}_s \cdot \hat{\vec{e}}_r r^2 \sin\theta \, \mathrm{d}\theta \, \mathrm{d}\phi} , \tag{4.9}$$

where $\vec{S}_s$ is the Poynting vector of the scattered field. The coordinates (r, θ, ϕ) form a spherical system of reference, centred on the particle and with the propagation vector of the incoming radiation aligned along the $\theta = 0$ direction. f_{sub} varies between 0 and 1 by definition and it should be maximized for solar cell applications. For $f_{sub} < 0.5$ light is mostly scattered backward. As done with the extinction efficiency, we calculate an averaged parameter

$$\hat{f}_{sub} = \frac{\int I(\lambda) f_{sub}(\lambda)\, \mathrm{d}\lambda}{\int I(\lambda)\, \mathrm{d}\lambda}, \tag{4.10}$$

which represents the total amount of radiation scattered in the forward direction.

4.1.2 Results and discussion

The quantities introduced above are calculated for different combinations of the parameters:

- the metals chosen are silver, gold, aluminium and copper;
- the radius of the spheres is 50, 100 and 150nm;
- nanospheres are embedded in air, silica (SiO_2) and silicon nitride (Si_3N_4).

The choice of different media could reflect the presence of a dielectric passivating layer in the solar cell. The complex dielectric functions of the metals are taken from Palik [79], while we assume a constant value for the refractive index of each medium, $N = 1.05$ for air, $N = 1.54$ for silica and $N = 2.02$ for silicon nitride. Although the true refractive index of the three environments is not constant over the optical range, this approximation should give an error of only a few percent. We also neglect any absorption in the media surrounding the nanoparticles. Results for the extinction efficiency are shown in Figures 4.3, 4.4 and 4.5, and in Tables 4.1, 4.2 and 4.3.

From the combination of tables and plots we draw several conclusions:

- the optimal size for scattering is around 150nm in air, but tends to decrease toward 100nm in SiO_2 and Si_3N_4. In general more than one

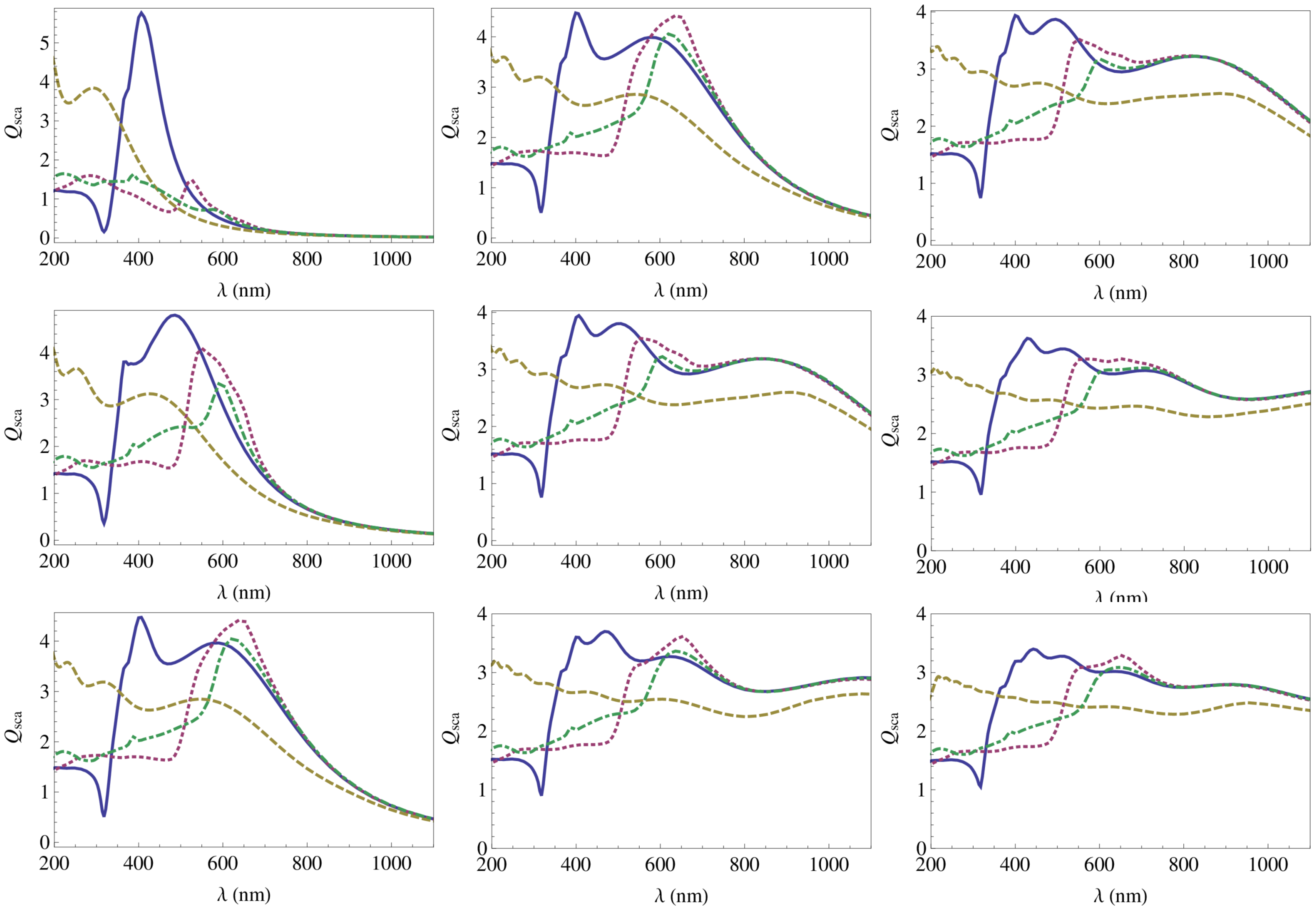


Figure 4.3: Scattering efficiency Q_{sca} as a function of the wavelength for different choices of the parameters. Top row: air, middle row: SiO_2, bottom row: Si_3N_4. Left column: $r = 50nm$, central column: $r = 100nm$, right column: $r = 150nm$. Blue line: silver, red dotted: gold, yellow dashed: aluminium, green dashed-dotted: copper.

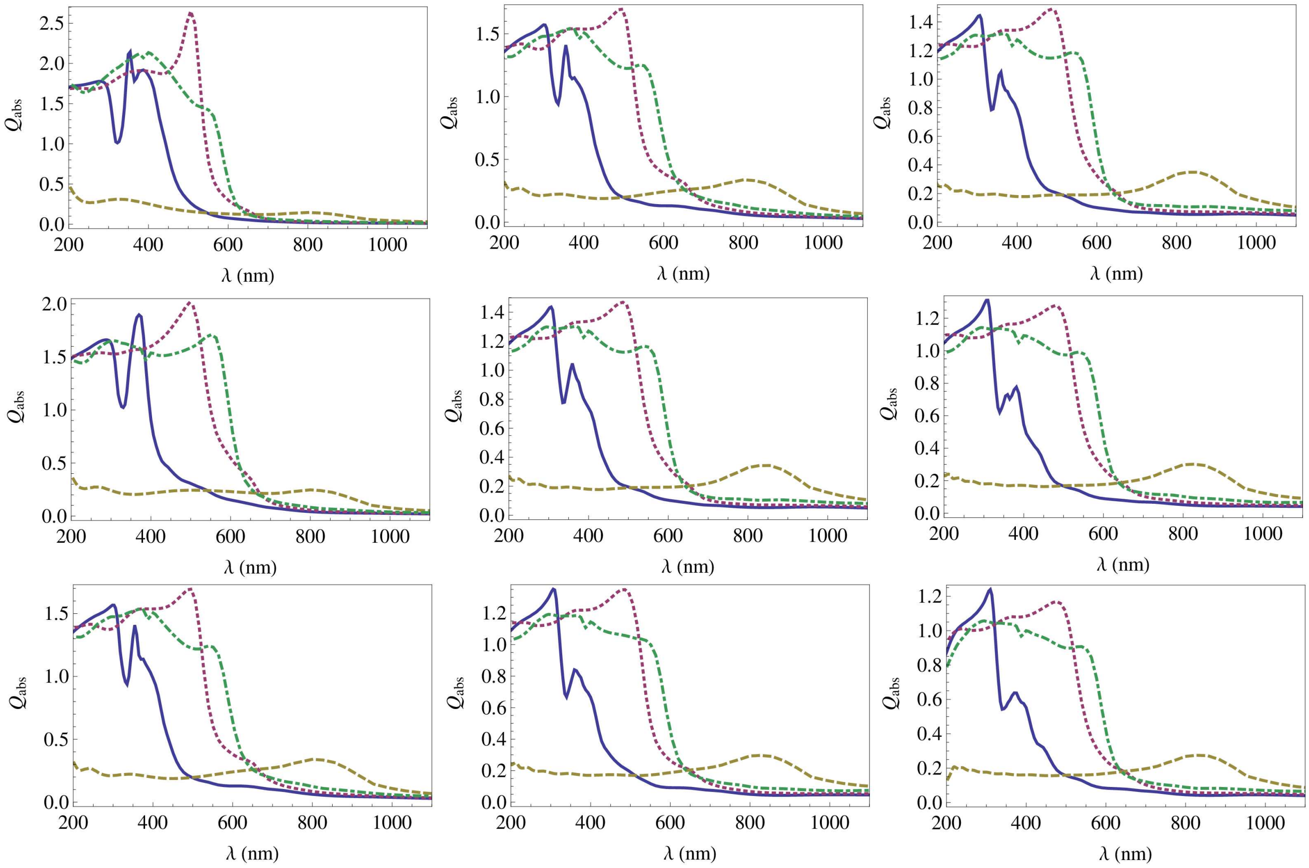


Figure 4.4: Absorption efficiency Q_{abs} as a function of the wavelength for different choices of the parameters. Top row: air, middle row: SiO_2, bottom row: Si_3N_4. Left column: $r = 50nm$, central column: $r = 100nm$, right column: $r = 150nm$. Blue line: silver, red dotted: gold, yellow dashed: aluminium, green dashed-dotted: copper.

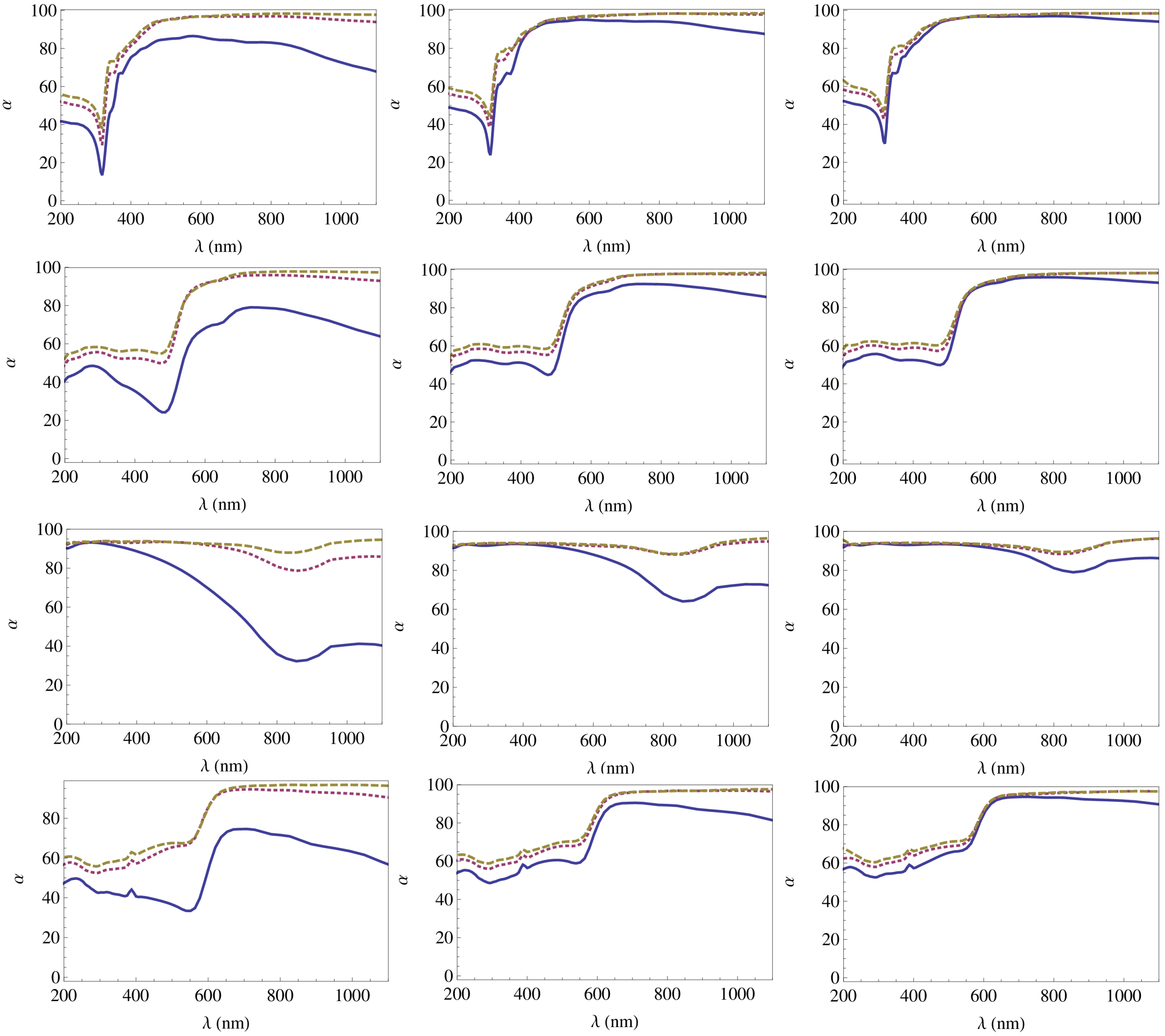


Figure 4.5: Albedo α as a function of the wavelength for different choices of the parameters. Top row: silver, second row: gold, third row: aluminium, bottom line: copper. Left column: air, central column: SiO_2, right column: Si_3N_4. Blue line: $50nm$, red dotted: $100nm$, yellow dashed: $150nm$.

Metal	r (nm)	$\hat{Q}_{sca}$	$\hat{Q}_{abs}$	$\hat{f}_{sub}$
Ag	50	0.98	0.26	0.46
	100	2.80	0.19	0.50
	150	3.21	0.16	0.61
Au	50	0.49	0.67	0.48
	100	2.44	0.56	0.52
	150	2.84	0.50	0.63
Al	50	0.60	0.15	0.40
	100	2.27	0.24	0.45
	150	2.56	0.22	0.59
Cu	50	0.41	0.69	0.47
	100	2.10	0.73	0.52
	150	2.71	0.67	0.63

Table 4.1: Averaged scattering and absorption, and $\hat{f}_{sub}$ in air for metal spheres as a function of the particle metal and size.

Metal	r (nm)	$\hat{Q}_{sca}$	$\hat{Q}_{abs}$	$\hat{f}_{sub}$
Ag	50	2.19	0.22	0.47
	100	3.20	0.16	0.63
	150	3.03	0.14	0.75
Au	50	1.74	0.66	0.49
	100	2.85	0.49	0.66
	150	2.72	0.42	0.77
Al	50	1.70	0.22	0.41
	100	2.55	0.22	0.61
	150	2.47	0.20	0.74
Cu	50	1.40	0.80	0.49
	100	2.72	0.66	0.65
	150	2.69	0.58	0.77

Table 4.2: Averaged scattering and absorption, and $\hat{f}_{sub}$ in SiO_2 for metal spheres as a function of the particle metal and size.

Metal	r (nm)	$\hat{Q}_{sca}$	$\hat{Q}_{abs}$	$\hat{f}_{sub}$
Ag	50	2.82	0.19	0.50
	100	3.09	0.15	0.70
	150	2.95	0.13	0.78
Au	50	2.45	0.56	0.52
	100	2.76	0.45	0.72
	150	2.67	0.39	0.80
Al	50	2.28	0.24	0.45
	100	2.48	0.20	0.69
	150	2.42	0.19	0.77
Cu	50	2.12	0.73	0.52
	100	2.74	0.60	0.72
	150	2.63	0.54	0.80

Table 4.3: Averaged scattering and absorption, and $\hat{f}_{sub}$ in Si_3N_4 for metal spheres as a function of the particle metal and size.

peak, due to the multipole excitation, is observable for large spheres. This leads to scattering of a larger part of the spectrum;

- embedding particles in a high-refractive environment slightly increases, especially at long wavelengths, the averaged scattering efficiency, but it tends to reduce the maximum intensity and to widen the profile in the scattering efficiency as a function of λ, as shown in Figure 4.3. This effect is due to the polarization of the medium, which tends to weaken the restoring force inside the polarized particles, to excite different oscillation modes and to shift the resonance peaks;

- increasing the particle size or the refractive index of the external medium reduces absorption, possibly by decreasing the damping of plasmon excitations. Copper and aluminium are partial exceptions to this rule. As a consequence, the albedo is maximum for Si_3N_4 and for $r = 150nm$. Absorption profiles are less affected than scattering profiles by the change in the parameters, as shown in figure 4.4;

- silver is by far the best scattering materials, both in absolute and integrated magnitude, followed by aluminium, gold and copper; the spread among different materials is reduced increasing the refractive index, as the polarization of the surrounding environment becomes more and more important;

- silver and aluminium in particular, are very poor absorbers and in general have a very high albedo. On the contrary, gold and copper absorb more light, especially at the peak of the solar spectrum between 400 and $500nm$, as shown in Figures 4.4 and 4.5.

Regarding the angular distribution of the scattered light, Figure 4.6 shows how f_{sub} varies with different combinations of sphere size, material and surrounding medium. Values for $\hat{f}_{sub}$ are reported in the last column of Tables 4.1, 4.2 and 4.3. As a general feature, short-wavelength radiation is scattered in the forward direction more than long-wavelength radiation. Spheres with larger size tend to scatter more in the forward direction, a well-known fact; less appreciated is that the same behavior is observed if the dielectric constant of the surrounding medium is increased. The presence of a surrounding dielectric, in fact, helps to excite higher-multipole scattering, for which the re-emitted radiation is peaked in the forward direction more than that of a dipole. The angular distribution is more or less unaffected by the material of which the spheres are composed, although aluminium is less efficient than other metals. This angular spread could enhance the optical path in a solar cell. We assume the enhancement to behave like $1/ < \cos\theta >$, where $< \cos\theta >$ is the average cosine of the scattered distribution and only the forward half-plane is considered in the averaging process. In the most favorable cases, this quantity is ≈ 2, i.e. the path length is doubled. It turns out that f_{sub} and $1/ < \cos\theta >$ cannot be maximized at the same time, so that a trade-off should be looked for

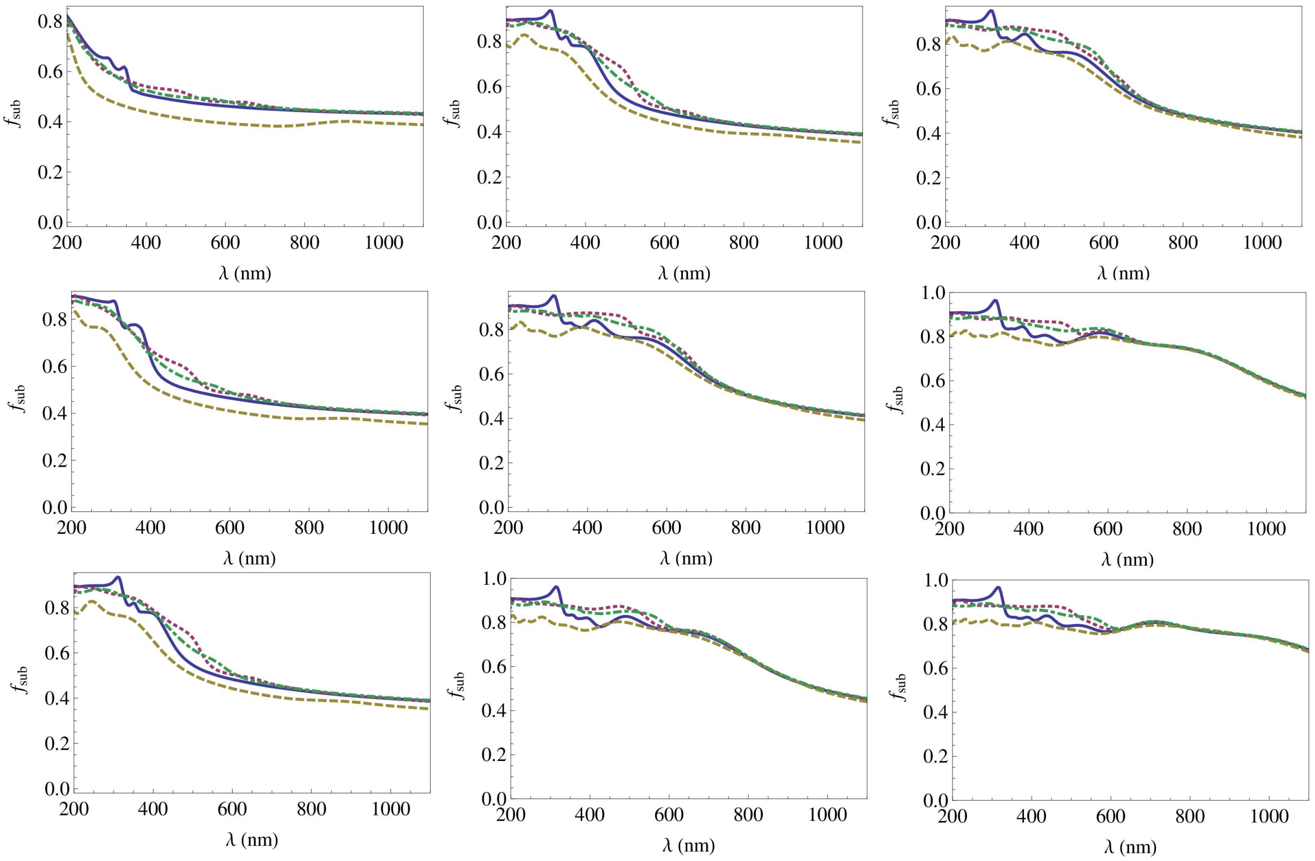


Figure 4.6: Fraction of forward scattered light as a function of the wavelength for different choices of the parameters. Top row: air, middle row: SiO_2, bottom row: Si_3N_4. Left column: $r = 50nm$, central column: $r = 100nm$, right column: $r = 150nm$. Blue line: silver, red dotted: gold, yellow dashed: aluminium, green dashed-dotted: copper.

in practical applications, e.g. solar cell design. From a naive combination of the two parameters, i.e. taking f_{sub} times $1/ < \cos\theta >$, f_{sub} appears to be the dominant factor. As a consequence, configurations for which f_{sub} is maximized should be preferred.

4.2 Scattering by non-spherical nanoparticles

In order to investigate the effect of changing the nanoparticle shape, we resorted to numerical simulation. We developed an in-house FDTD code for the solution of Maxwell's equations [1,3]. We used a $120 \times N_y \times 120$ cubic Yee [1] cell lattice, with an increasing number of cells along the y axis: N_y=120, 168, 216. We need a varying N_y because we want to accommodate metal nanostructures of different dimensions along the incident beam propagation direction, which we assume to lie along the y axis. The space step amounts to $2nm$ for a fairly good representation of the geometrical details. The time step is $\delta t = \delta/(2c_0)$, with c_0 the vacuum light speed and δ the space step, to satisfy the stability condition [3, 60] in three spatial dimensions. We use the TFSF source [3], placed 8 cells inward from each face of the outer boundary of the FDTD lattice, to create a plane wave linearly polarized electromagnetic pulse impinging on the nanostructures (polarization is along the z axis). The FDTD lattice is completed with an extra layer, 15 cells thick, simulating an infinite surrounding medium. This layer acts as an absorbing boundary condition [3]. It is implemented using the CPML boundary conditions [4]. We use a compact pulse exciting signal, i.e. of finite duration and with zero values outside a given time interval [25, 57]. The signal duration is suitably chosen to get spectral distribution results in the range $300 \div 1100nm$, as obtained by DFT, which is updated at every FDTD time iteration, until the excitation is extinguished inside the whole numerical lattice. To correctly track the metal behavior in the frequency domain with a single run, however, we also need to include frequency dispersion in the time domain algorithm. This is accomplished by generalizing the recursive convolution approach described in [18] to the "two critical points" correction of a Drude dielectric function as given in [58]. To calculate f_{sub} in (4.9), the angular distribution of the Poynting vector is obtained by implementing the Kirchhoff's integral formula [72], which transforms the near-fields, calculated with the FDTD algorithm around the nanoparticles, into plane scattered waves far away from them, one for each sampled direction. Angular values were sampled at one degree steps. Before applying the method to variable shapes, we verified that for spherical nanoparticles FDTD results reproduce those obtained by the analytical approach. Choosing Al as the nanoparticle metal, we have evaluated the scattering and absorption efficiency Q_i for spheroids, hemispheres and cylinders. For each geometry, the projected surface on the plane normal to the incident radiation is a circle of radius $100nm$, so that results are comparable to those obtained above for spheres of the same radius. For spheroids, the height is 100 and $3000nm$ (corresponding to oblate and prolate spheroids respectively, while $h = 200nm$ would correspond to the sphere discussed above). Cylinders are 100, 200 and $300nm$ high, while the hemisphere has, clearly, $r = h = 100nm$. Figure 4.7 shows the distribution of the field intensity inside and outside the nanoparticles. Figure 4.8 compares Q_{sca}, Q_{abs} and the albedo α for the different shapes to the results obtained for a sphere of radius $r = 100nm$. As done with spheres, we can extract the integrated efficiency $\hat{Q}_i$ and compare it to the previous results, as shown in Tab. 4.4. Spheroidal shapes, including spheres and hemispheres, show a similar behavior, with very similar extinction cross sections and albedo. The most remarkable effect of abandoning a spherical shape is a shift of the scattering peaks. Another interesting feature is the reduction of the absorption when the height is reduced. Cylinders

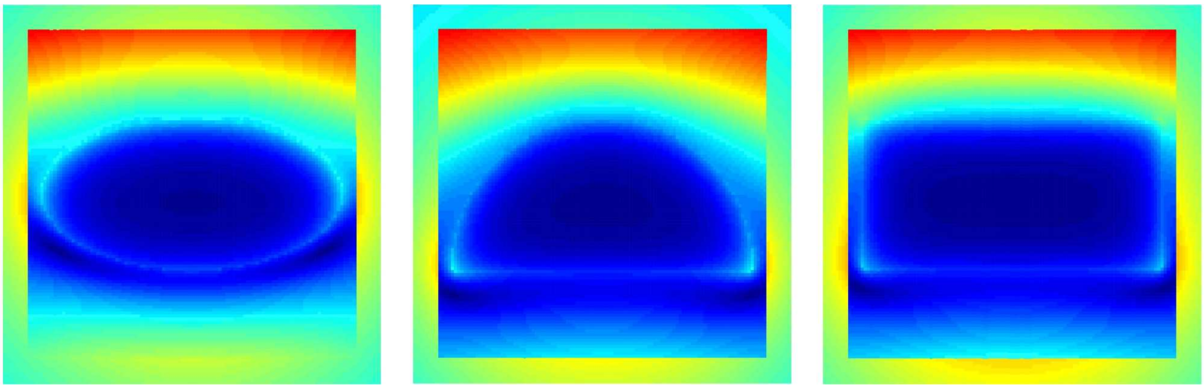


Figure 4.7: FDTD/TFSF simulation of the electric field intensity surrounding an aluminium ellipsoid, a hemisphere and a cylinder with $h = 100nm$. For each case, the polarized radiation comes from the top side. Colors represent a logarithmic scale of the intensity, normalized to a reference value of 1 V/m. The wavelength is $500nm$.

show a different behavior: independently from their height the scattering and absorption efficiency are increased with respect to the sphere, both in absolute and integrated value. On the contrary, the albedo resembles very close that of a sphere; cylinders seem then to be a better choice as scatterer.

Finally, the angular distributions of scattered light is shown in Figure 4.9. We see that cylinders and spheroids are less performing than spheres at short wavelengths, but, in most cases, they perform better at long wavelengths. As shown in Tab. 4.4, if the averaged values are considered, the sphere is outperformed by all the other shapes excluded the spheroids with $h = 300nm$. In conclusion we recognize that varying the metal has a very poor impact on the angular distribution of scattered light. For spheres, the particle radius should be confined to the range $100 \div 150nm$, where scattering dominates over absorption and the radiation is re-emitted mostly into the cell. A high dielectric constant medium has a positive impact in many respects. In particular, it drives a bigger fraction of the scattered radiation in the half-space which would be occupied by the cell. It also tends to decrease the restoring force and the damping of the plasmon excitations inside the nanoparticle, leading to lower absolute magnitude for scatter-

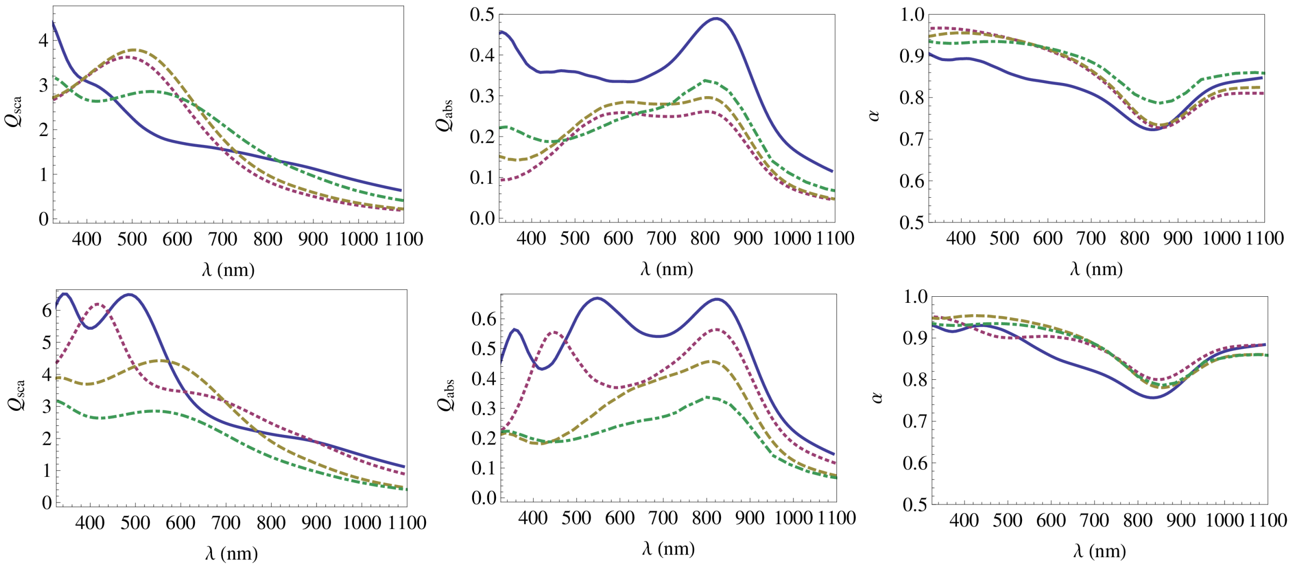


Figure 4.8: Scattering efficiency (left), absorbing efficiency (middle) and albedo (right) as a function of the wavelength for aluminium spheroids and cylinders. In the first row, blue line: spheroid with $h = 300nm$; red dotted: spheroid with $h = 100nm$; yellow dashed: semisphere with $r = 100nm$; green dashed-dotted: sphere with $r = 100nm$. In the second row, blue line: cylinder with $h = 300nm$; red dotted: cylinder with $h = 200nm$; yellow dashed: cylinder with $h = 100nm$; green dashed-dotted: sphere with $r = 100nm$. All particles are placed in air.

	h (nm)	$\hat{Q}_{sca}$	$\hat{Q}_{abs}$	$\hat{f}_{sub}$
Cylinders	100	2.98	0.30	0.49
	200	3.29	0.41	0.50
	300	3.57	0.53	0.49
Ellipsoids	100	2.00	0.20	0.49
	300	1.75	0.35	0.42
Semisphere	100	1.81	0.22	0.50
Sphere	200	2.27	0.24	0.45

Table 4.4: Extinction parameters, and integrated $\hat{f}_{sub}$ for aluminium cylinders and ellipsoid of different heights and with $r = 100nm$, compared with an aluminium sphere with $r = 100nm$. All particles are placed in air.

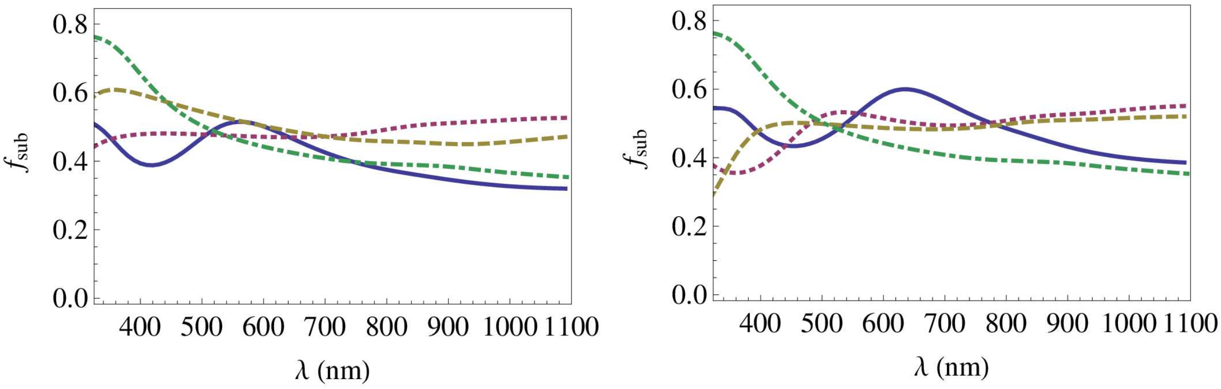


Figure 4.9: Fraction of light scattered into the substrate as a function of the wavelength for different choices of the parameters. On the left: spheroids, semisphere and sphere. Blue line: spheroid with $h = 300nm$, red dotted: $h = 100nm$, yellow dashed: sphere with $r = 100nm$. On the right: cylinders and sphere. Blue line: cylinder with $h = 300nm$, red dotted: $h = 200nm$, yellow dashed: $h = 100nm$, green dashed-dotted: sphere with $r = 100nm$. All particles are placed in air.

ing and absorption, but it allows the excitation of higher multipoles and the efficient scattering of larger parts of the spectrum. Regarding shapes, we show that cylinders have the greatest scattering efficiency, especially if the height of the cylinder is increased, but they also exhibit high absorption. There is not a big difference among sphere and spheroids in general, although for spheroids it is better to reduce the height in order to keep absorption low. We also show that albedo is almost independent from the chosen shape. The present analysis is independent from the choice of a specific cell design, and it neglects the presence of the air/cell interface. Including it might change some of the conclusions, without invalidating their general value. We leave this part of the investigation to a future work, where we aim at the complete simulation of a realistic cell.

Chapter 5

Simulation of opal photonic crystals

5.1 Modeling and Results

Artificial opals can be prepared by self-assembling of silica or polystyrene spheres, for example, in a colloidal solution. In Fig. 5.1 a SEM (Scanning Electron Microscope) image [80] of an artificial opal PC is shown as example. The analytical solution of the scattered field by a single sphere is well known [20], but the scattering behavior by a multiple spheres system can be obtained only through numerical approach. The opal PC modeled here is made of non-dispersive polystyrene nanospheres with radius $R = 122nm$, relative permittivity $\epsilon_r = 2.4$, conductivity $\sigma = 0$. They are ordered in a structure of $25 \times 25 \times 5$ FCC cells, where 5 is the number of lattice planes in the propagation direction. The structure of a single FCC cell is illustrated in Fig. 5.2. The edge of the FCC cell measures $340nm$. The filling factor used for simulating a light overlapping or crushing between the nanospheres is 0.774. The theoretical filling factor of the FCC cell is instead 0.74.

The impinging electromagnetic plane pulse in the optical range is realized by the TFSF plane source method (Fig. 5.3). The wave propagates along the y-direction and is polarized along the x-direction. The Huygens box separates the scattered from the total field region and allows us to

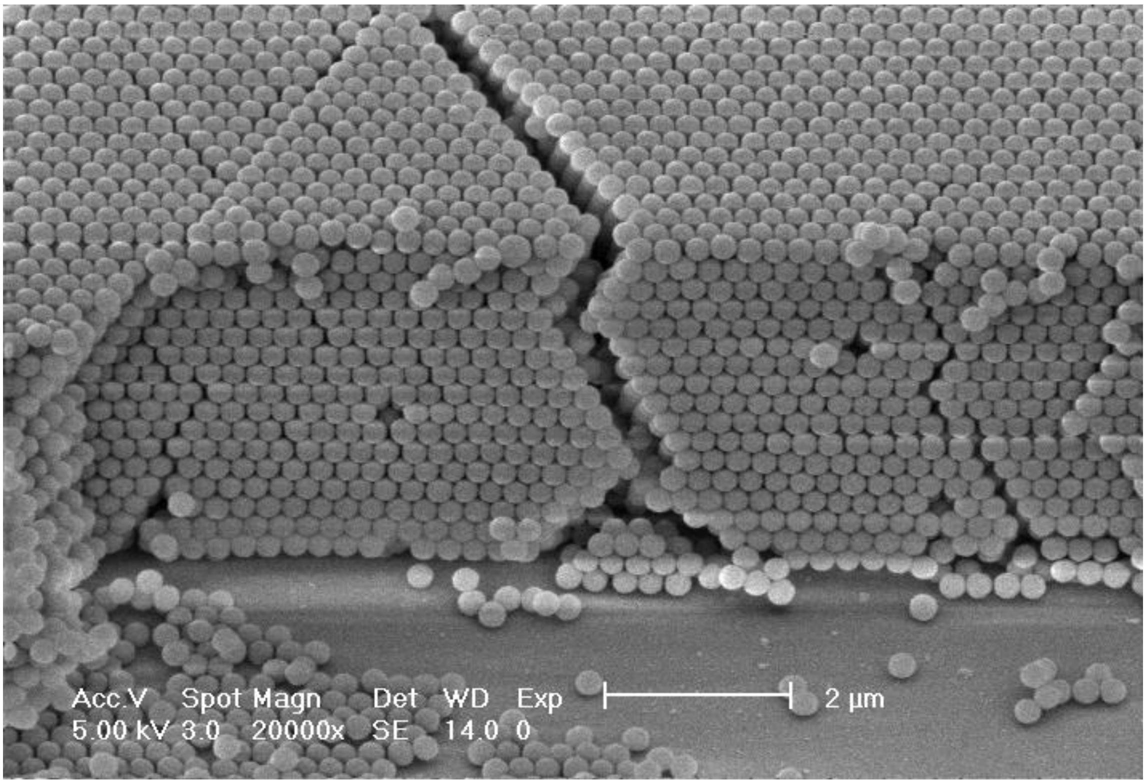


Figure 5.1: SEM example image of an artificial opal PC.

introduce the exciting plane wave signal $s(t)$ already mentioned in (2.40). The transmittance T of the opal PC is calculated by the equation (5.1)

$$T = 1 - \frac{P_s + P_a}{P_i}, \tag{5.1}$$

where P_i is the power associated to the incident field, P_s is the scattered power and P_a the absorbed power from the opal PC. The absorbed power is assumed to be zero because we are dealing with a non-absorptive dielectric. The scattered power is calculated by means of an integration of the scattered field in the far-field region. Starting from the values of the scattered field over a closed surface Σ_1, that is localized in the near-field region, it is possible to obtain the far-field for each propagation direction (Fig. 5.4). The near-to-far transformation is realized using the Kirchhoff's

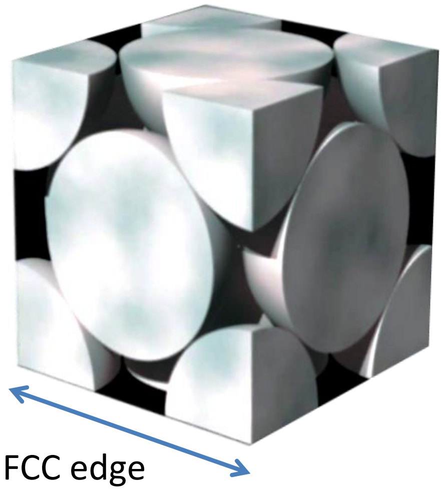


Figure 5.2: Face-Centered Cubic cell.

formula (5.2)

$$\vec{E}(\vec{r'},\omega) = \frac{e^{ikr'}}{r'}\frac{k\hat{n}'}{4\pi i}\oiint_{\Sigma_1}\left[\sqrt{\frac{\mu_0}{\epsilon}}(\hat{n}'\times(\hat{n}\times\vec{H})) - \hat{n}\times\vec{E}\right]e^{-ik\hat{n}'\cdot\vec{r}}dS, \quad (5.2)$$

where $\vec{E}$ and $\vec{H}$ are the electric and magnetic field respectively, $k = \omega/c = \omega\sqrt{\mu_0\epsilon}$ is the wavenumber, ω the angular frequency, c the speed of light, $\hat{n}$ the normal versor to the integration surface Σ_1, $\hat{n}'$ the versor of $\vec{r'}$, i the imaginary unit, $\vec{r}$ and $\vec{r'}$ are the vectors that scan the near-field and the far-field region respectively. This transformation allows us to have the angular power distribution or radiation pattern. Integrating the far-field with the Poynting vector flux we obtain the total scattered power P_s.

An in-house FDTD code has been developed for simulating the problem. The space step chosen for the discretization is $\delta = \delta x = \delta y = \delta z = 10nm$. The time step δt used for the simulation satisfies the Courant-Friedrichs-Lewy stability condition [60] and is equal to $\delta/(2c_0)$. The spatial discretization fixes the sampling points of the electric and magnetic fields. Depending on the position of each sampling point the local properties of the medium (permittivity and permeability) are assigned when the field component is calculated. The medium properties are not associated with

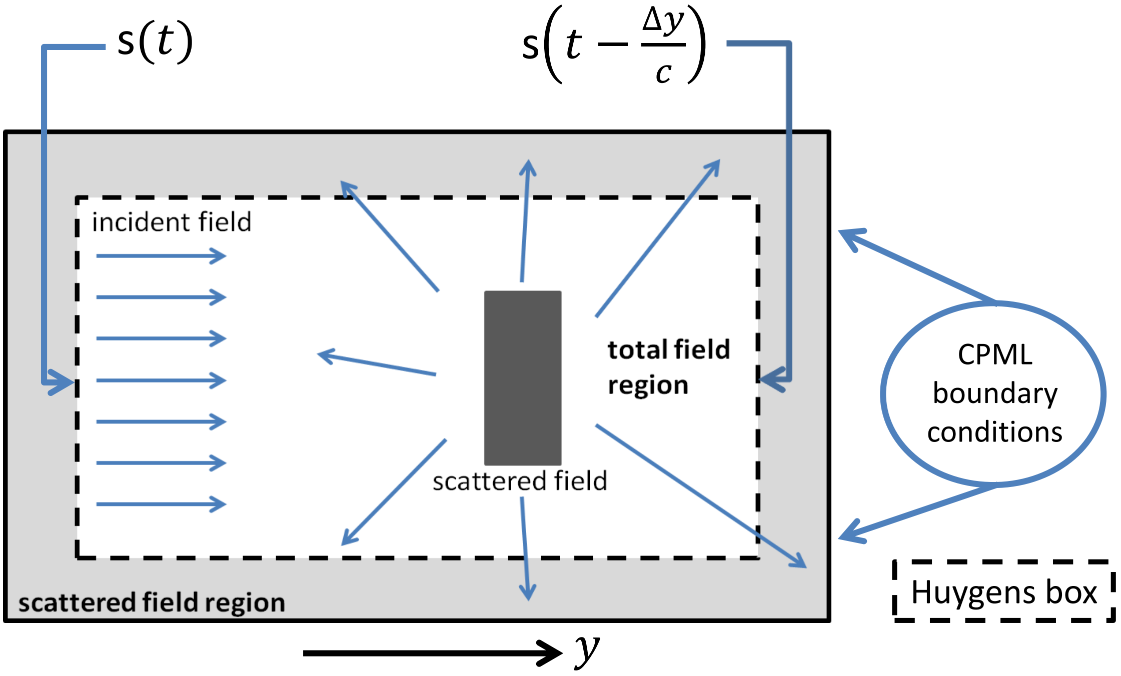


Figure 5.3: Total-Field/Scattered-Field plane source method.

the whole Yee cell, but with each component of the electromagnetic field. This per-component based assignation of the material parameters allowed us to obtain an accurate value of the filling factor. The total simulation domain, comprehensive also of the boundary conditions, is made of 999 × 999 × 300 Yee cells. We used the CPML boundary conditions [4, 5]. An HPC cluster is necessary for simulating such kind of problem and a parallel approach has been used in order to reduce the computational time. The total domain has been divided into 27 blocks, each one composed by 333 × 333 × 100 Yee cells. Each block runs as an MPI process on a different core of the cluster. The cores exchange information during the simulation through the InfiniBand network. At the end all the data are collected and the solution of the total domain is given. The solution for the message passing data between adjacent blocks has been proposed in chapter 3 in the case of a 1D-slicing. In this case a 3D-slicing has been made. The

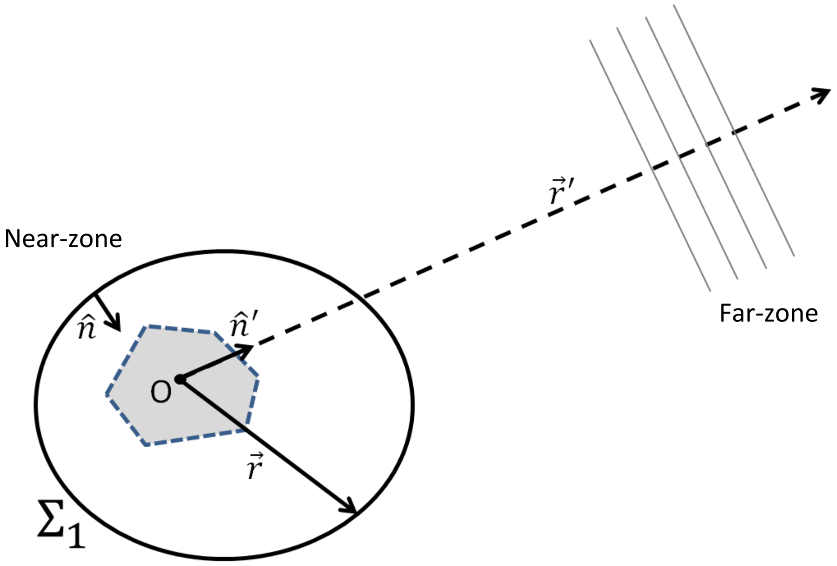


Figure 5.4: Near-to-far transformation.

exchange of the tangential electric and magnetic field components follows the scheme shown in Fig. 3.3. It is done at each face of the block with the neighbors as illustrated in Fig. 5.5. We observed a strong dependence of the band-gap position from the filling factor (Fig. 5.6). In Fig. 5.7 the experimental transmittance [53] of the opal photonic crystal we have simulated is shown.

The transmittance curves reported in Figs. 5.8 for different x-axis representations have been deduced by simulation with the aforementioned filling factor in the whole optical frequency range. They evidences an agreement with a 5% of error compared to the experimental curve in Fig. 5.7. The tolerance error is due to the FDTD staircasing and to the difficulty of knowing the exact experimental filling factor. The band-gap region is large because of the small number of lattice planes used for modeling the opal PC. The radiation pattern of the scattered field is obtained basing on the far-field calculated through the near-to-far transformation. The comparison between the scattered field in the band-gap region (outer blue curve at $\lambda = 530nm$) and outside the band-gap region (inner red curve

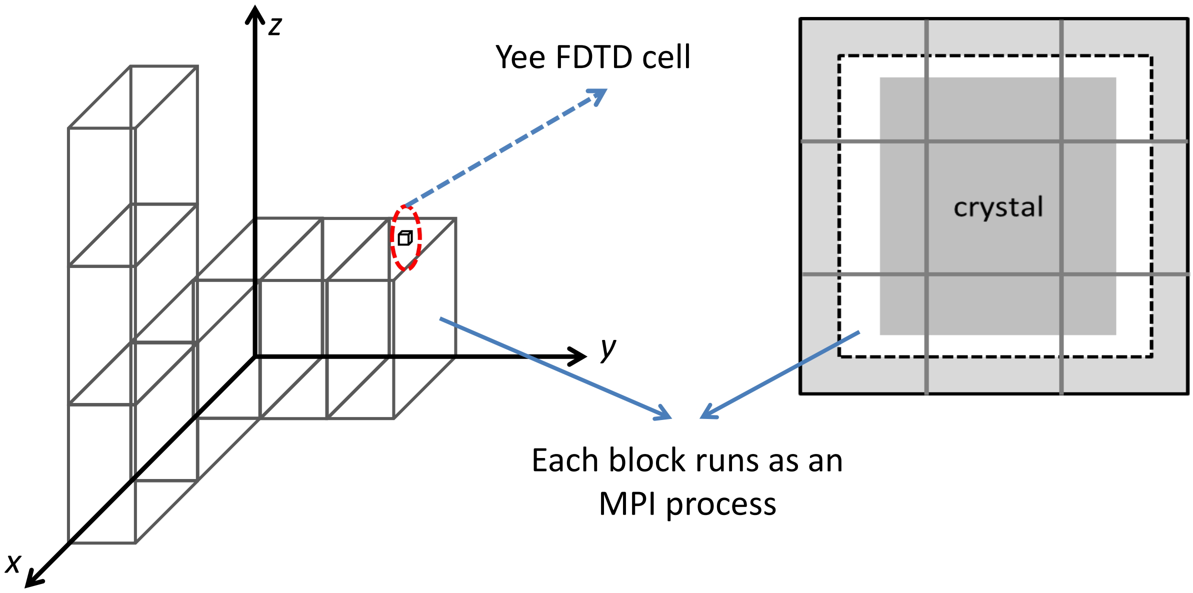


Figure 5.5: Domain decomposition and data exchange between adjacent blocks.

at $\lambda = 400nm$) is shown in Fig. 5.9. It highlights the different scattering behavior in the band-gap region, where the scattering is maximum, with respect to the external region. In Figs. 5.10 a side and a back view of the E_z electric field component on a cut plane of the analysis domain is shown. The maps result from the DFT at the wavelength $\lambda = 550nm$, which stays in the transmittance band-gap region.

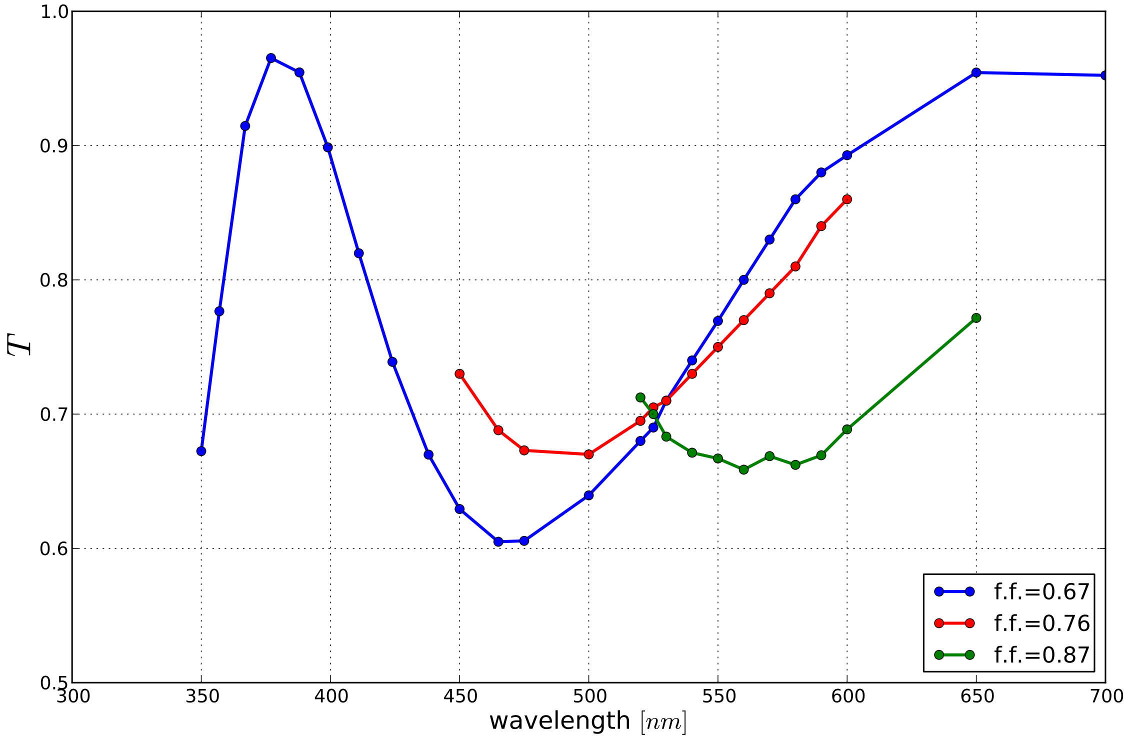


Figure 5.6: Transmittance varying the filling factor.

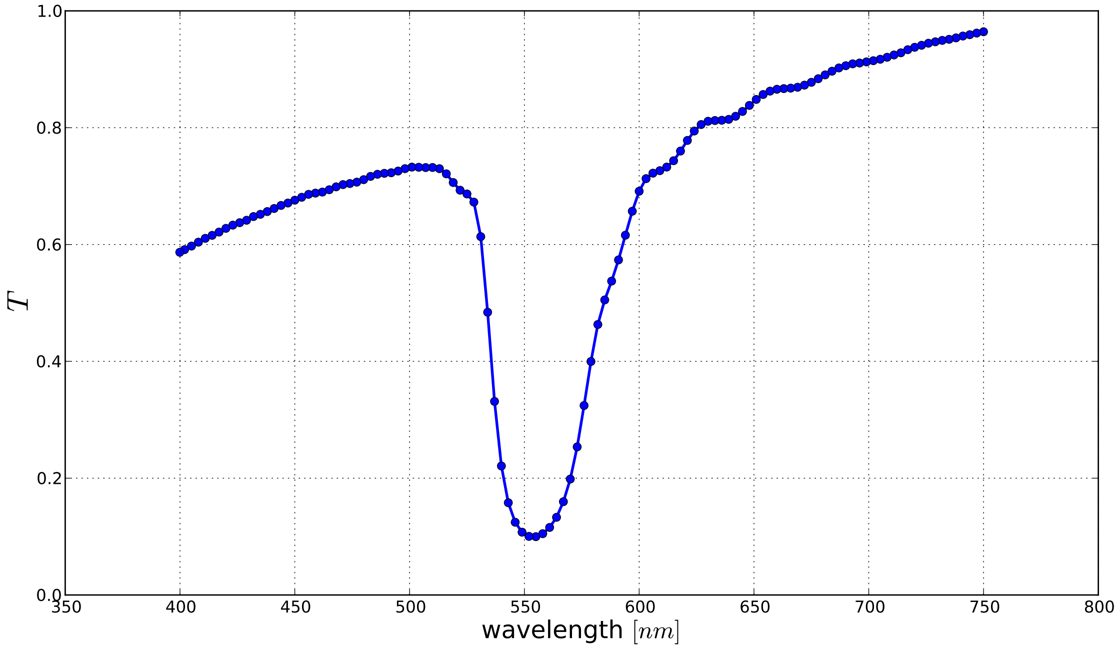
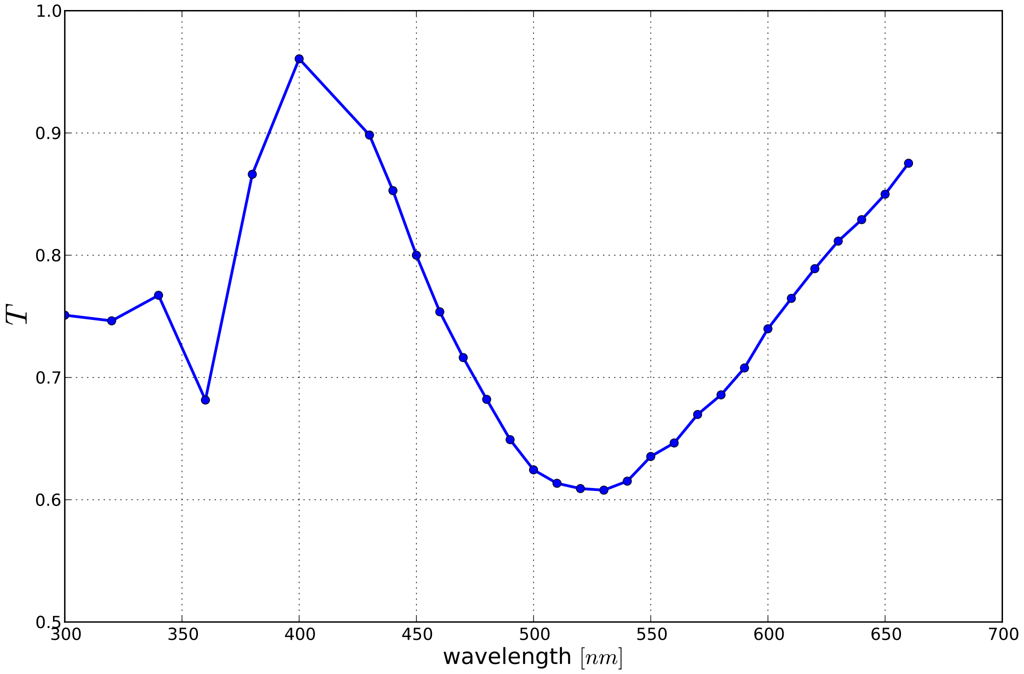


Figure 5.7: Experimental transmittance of the reference opal PC.



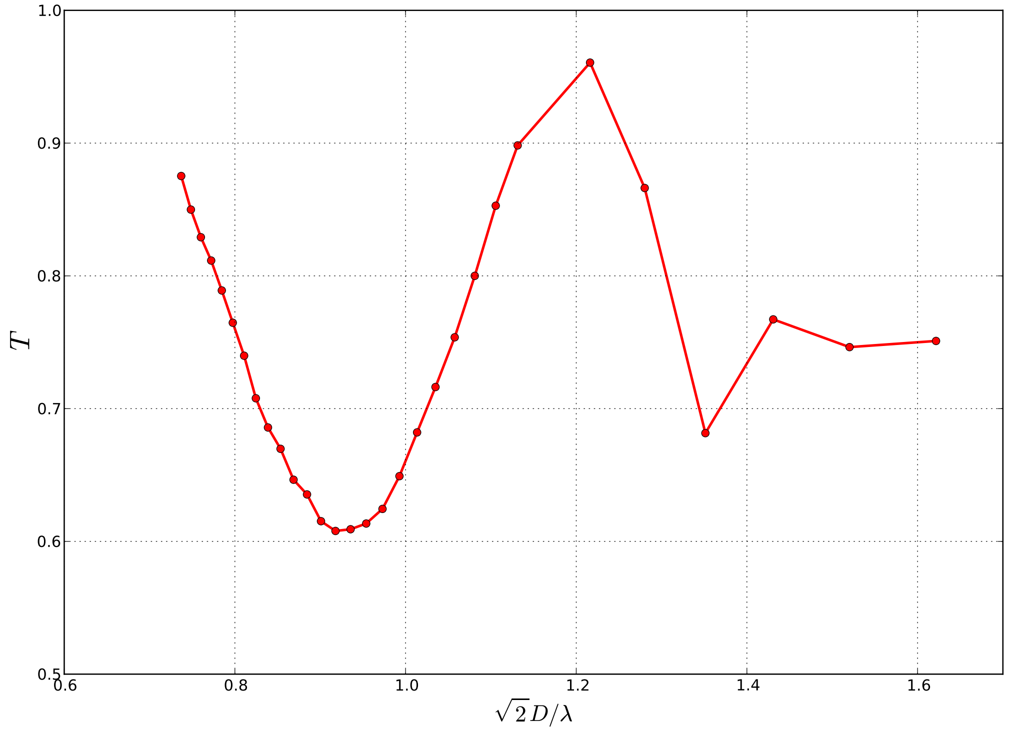


Figure 5.8: Transmittance of the opal PC with $f.f. = 0.774$.

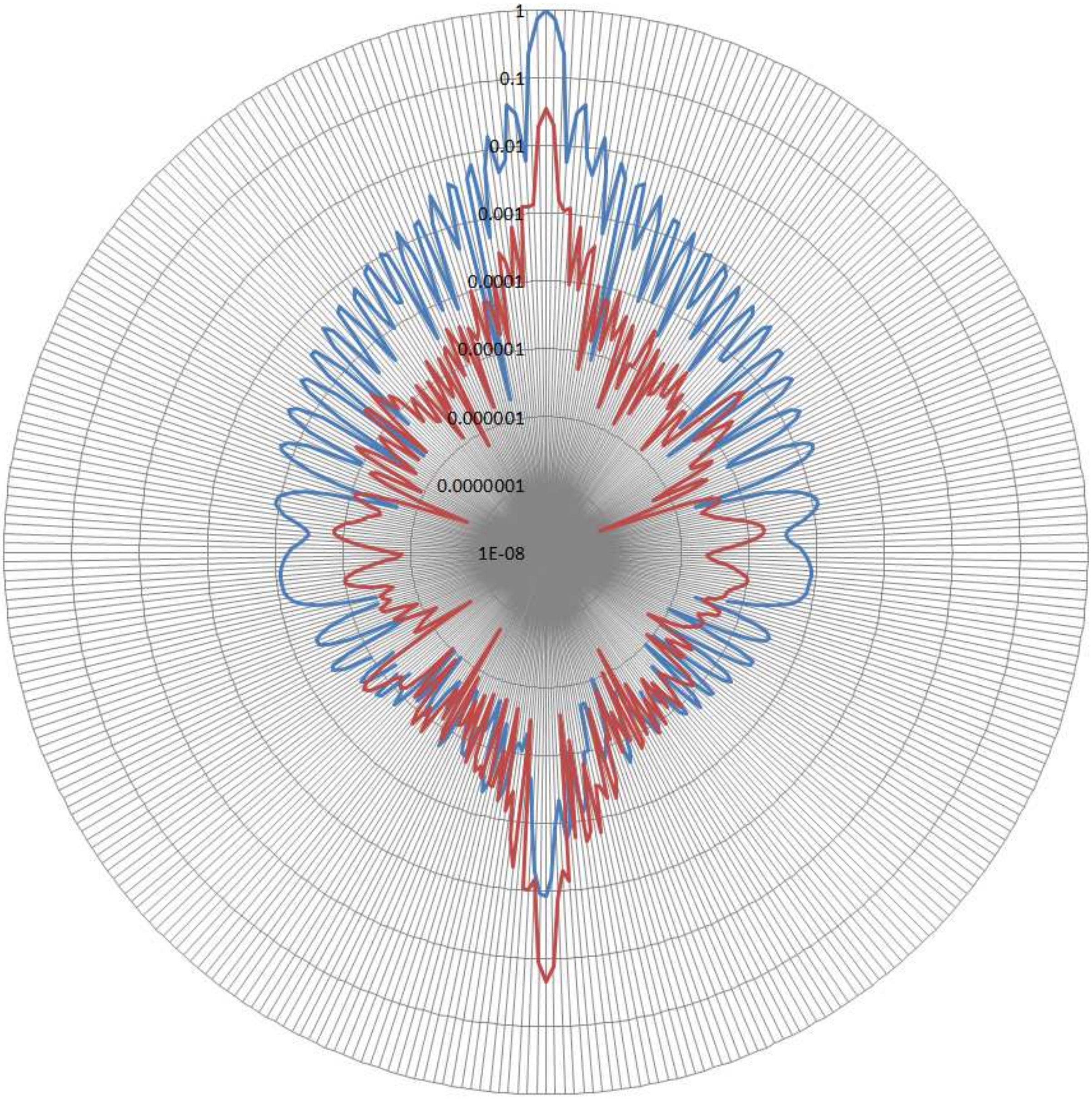


Figure 5.9: Scattered field inside (outer curve) and outside (inner curve) the band-gap region. Log scale.

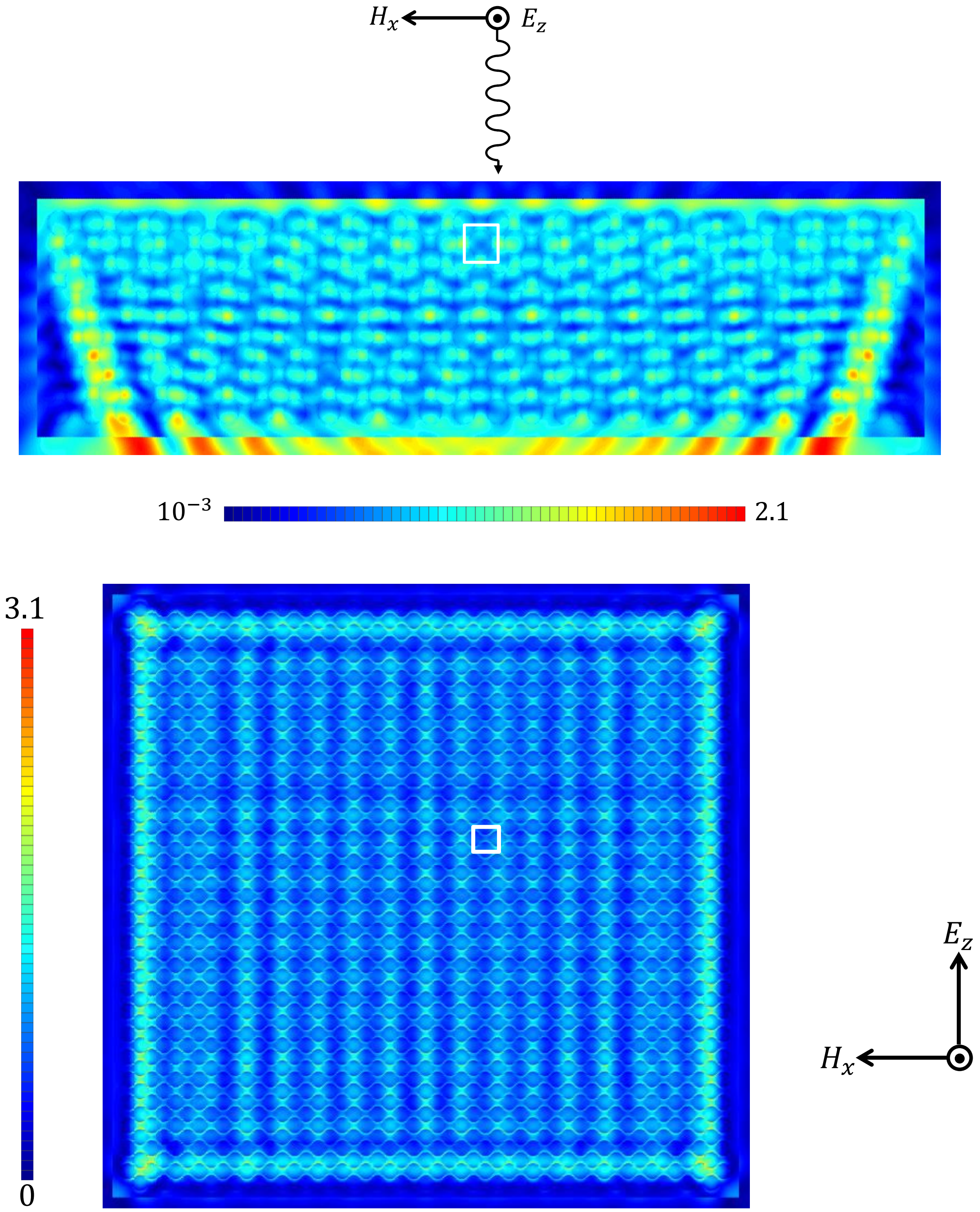


Figure 5.10: E_z component normalized to the incident field.

Chapter 6

Conclusions

Numerical simulation is the only way to analyze and make prediction about complex electromagnetic problems.

In the first chapter a modified Recursive Convolution algorithm for the FDTD analysis of Drude dispersive media is proposed. The algorithm has been tested by comparing the electric field and the extinction coefficient deviation from the analytical solution for gold and silver nanospheres. It evidences an accuracy improvement with respect to the standard RC method. The better precision is observable in particular at the plasmonic resonance frequencies and makes the modified algorithm suitable for plasmonic simulations. The modified approach was also extended to other dispersion models from a theoretical point of view revealing an increased ease of implementation. This approach is more general especially for the N-order models.

The second chapter is dedicated to the parallelization of a subgridding FDTD algorithm, previously proposed in a serial version. Complex problems can need a detailed discretization in some subregions. The subridding allows us to use a different space-step in these subdomains, while the parallel approach speeds up the computational time. The parallel code has been tested with moderately large FDTD models. Although affected by a lack of exclusive access to the cluster, it gives fairly good speedup and scalability. These parameters are also analyzed from a theoretical point of view.

Using analytical and numerical methods, in the third chapter, the light scattering and absorption from small metal particles embedded in an infinite and homogeneous medium have been investigated. Beside calculating the wavelength dependence of the relevant cross sections in the proximity of the plasmon resonance frequency, we have introduced the averaged quantities $\hat{Q}_i$ and $\hat{f}_{sub}$ which provide a quick means to evaluate the optical response of the nanoparticles under a given illumination. The aim is to provide a guide for choosing the particle size and composition as well as the characteristics of the surrounding dielectric in order to design the plasmonic layers for solar cell applications. As a general conclusion, silver and aluminium are the best choice for metals, the former for showing high scattering efficiency, the latter having poor absorption and high albedo.

In the fourth chapter a parallel FDTD code has been developed for simulating the transmittance of opal photonic crystals. The numerical results are in good agreement with the experimental ones. We are confident in future to apply this simulation tool to the random close packing sphere configuration for the analysis of photonic glasses. Besides optimized photonic crystals could be designed. The transmittance tuning could be obtained varying the permittivity, introducing *defects* as nanoparticles of different materials or plasmonic noble metal nanospheres, changing the filling factor, etc.

Bibliography

[1] K. S. Yee, "Numerical solution of initial boundary value problems involving Maxwell's Equations in isotropic media," *IEEE Transactions on Antennas and Propagation*, vol. 14, no. 3, pp. 302–307, 1966.

[2] A. Taflove, "Application of the finite-difference time-domain method to sinusoidal steady-state electromagnetic penetration problems," *IEEE Trans. Electromagnetic Compatibility*, vol. 22, pp. 191–202, 1980.

[3] A. Taflove and S. C. Hagness, *Computational Electrodynamics: The Finite-Difference Time-Domain Method, 3rd ed.* Norwood: MA: Artech House, 2005.

[4] J. A. Roden and S. D. Gedney, "Convolution PML (CPML): An efficient FDTD implementation of the CFS-PML for arbitrary media," *Microwave and Optical Technology Letters*, vol. 27, no. 5, pp. 334–339, 2000.

[5] I. Laakso, S. Ilvonen, and T. Uusitupa, "Performance of convolutional PML absorbing boundary conditions in finite-difference time-domain sar calculations," *Phys. Med. Biol.*, vol. 52, pp. 7183–7192, 2007.

[6] D. M. Sullivan, "Frequency-dependent FDTD methods using Z transforms," *IEEE Trans. Antennas Propagat.*, vol. 40, pp. 1223–1230, 1992.

[7] O. P. Gandhi, B.-H. Gao, and J.-Y. Chen, “A frequency-dependent Finite-Difference Time-Domain formulation for general dispersive media,” *IEEE Trans. Microwave Theory Tech.*, vol. 41, pp. 658–665, 1993.

[8] J. L. Young, “Propagation in linear dispersive media: Finite-difference time-domain methodologies,” *IEEE Trans. Antennas Propagat.*, vol. 43, pp. 422–426, 1995.

[9] J. A. Pereda, L. A. Vielva, A. Vegas, and A. Prieto, “State-space approach to the FDTD formulation for dispersive media,” *IEEE Trans. Magn.*, vol. 31, pp. 1602–1605, 1995.

[10] D. F. Kelly and R. J. Luebbers, “Piecewise linear recursive convolution for dispersive media using FDTD,” *IEEE Trans. Antennas Propagat.*, vol. 44, pp. 792–797, 1996.

[11] M. Okoniewski, M. Mrozowski, and M. A. Stuchly, “Simple treatment of multi-term dispersion in FDTD,” *IEEE Microwave Guided Wave Lett.*, vol. 7, pp. 121–123, 1997.

[12] Q. Chen, M. Katsuari, and P. H. Aoyagi, “An FDTD formulation for dispersive media using a current density,” *IEEE Trans. Antennas Propagat.*, vol. 46, pp. 1739–1746, 1998.

[13] J. A. Pereda, A. Vegas, and A. Prieto, “FDTD modeling of wave propagation in dispersive media by using the Mobius transformation technique,” *IEEE Trans on Microwave Theory and Techniques*, vol. 50, pp. 1689–1695, 2002.

[14] M. Okoniewski and E. Okoniewska, “Drude dispersion in ADE FDTD revisited,” *Electronics Letters*, vol. 42, no. 9, pp. 503–504, 2006.

[15] S. Kong, J. J. Simpson, and V. Backman, "ADE-FDTD scattered-field formulation for dispersive materials," *IEEE Microwave and Wireless Components Letters*, vol. 18, no. 1, pp. 4–6, 2008.

[16] J. S. et al, "Simple trapezoidal recursive convolution technique for the frequency-dependent FDTD analysis of a Drude-Lorentz model," *IEEE Photonics Technology Letters*, vol. 21, no. 2, pp. 100–102, 2009.

[17] M. A. Alsunaidi and A. A. Al-Jabr, "A general ADE-FDTD algorithm for the simulation of dispersive structures," *IEEE Photonics Technology Letters*, vol. 21, no. 12, pp. 817–819, 2009.

[18] R. J. Luebbers, F. Hunsberger, and K. S. Kunz, "A frequency-dependent finite-difference time-domain formulation for transient propagation in plasma," *IEEE Transactions on Antennas and Propagation*, vol. 39, no. 1, pp. 29–34, 1991.

[19] K. Kolwas, A. Derkachova, and M. Shopa, "Size characteristics of surface plasmons and their manifestation in scattering properties of metal particles," *Journal of Quantitative Spectroscopy & Radiative Transfer*, vol. 110, pp. 1490–1501, 2009.

[20] C. F. Bohren and D. R. Huffman, *Absorption and scattering of light by small particles.* New York: Wiley, 1998.

[21] A. Vial, A. S. Grimault, D. Macias, D. Barchiesi, and M. L. de la Chapelle, "Improved analytical fit of gold dispersion: Application to the modeling of extinction spectra with a finite-difference time-domain method," *Phys. Rev. B*, vol. 71, no. 8, pp. 085416–085422, 2005.

[22] A. Vial, "Implementation of the critical points model in the recursive convolution method for modelling dispersive media with the finite-

difference time domain method," *J. Opt. A-Pure Appl. Opt.*, vol. 9, pp. 745–748, 2007.

[23] A. Vial and T. Laroche, "Description of dispersion properties of metals by means of the critical points model and application to the study of resonant structures using the FDTD method," *J. Phys. D-Appl. Phys.*, vol. 40, no. 22, pp. 7152–7158, 2007.

[24] R. J. Luebbers and F. Hunsberger, "FDTD for Nth-order dispersive media," *IEEE Transactions on Antennas and Propagation*, vol. 40, no. 11, pp. 1297–1301, 1992.

[25] A. Vaccari, R. Pontalti, C. Malacarne, and L. Cristoforetti, "A robust and efficient subgridding algorithm for finite-difference time-domain simulations of Maxwell's equations," *J. Comput. Phys.*, vol. 194, no. 1, pp. 117–139, 2004.

[26] M. Okoniewski, E. Okoniewska, and M. A. Stuchly, "Three-dimensional subgridding algorithm for FDTD," *IEEE Trans. Antenna Propagat.*, vol. 45, pp. 422–429, 1997.

[27] W. Yu and R. Mittra, "A new subgridding method for finite difference time domain (FDTD) algorithm," *Microwave Opt. Technol. Lett.*, vol. 21, no. 5, pp. 330–333, 1999.

[28] B. Wang, Y. Wang, W. Yu, and R. Mittra, "A hybrid 2-D ADI-FDTD subgridding scheme for modeling on-chip interconnets," *IEEE Transactions on Advanced Packaging*, vol. 24, no. 4, pp. 528–533, 2001.

[29] M. Marrone, R. Mittra, and W. Yu, "A novel approach to deriving a stable hybridized FDTD algorithm using the cell method," (Columbus (OH)), June 2003.

[30] C. Guiffaut and K. Mahdjoubi, “A parallel FDTD algorithm using the MPI library,” *IEEE Antennas and Propagat. Mag.*, vol. 43, no. 2, pp. 94–103, 2001.

[31] W. Yu, Y. Liu, T. Su, N. Huang, and R. Mittra, “A robust parallel conformal finite-difference time-domain processing package using the MPI library,” *IEEE Antennas and Propagat. Mag.*, vol. 47, no. 3, pp. 39–59, 2005.

[32] X. Chen, M. Cracraft, Y. Zhang, J. Zhang, J. Drewniak, B. Archambeault, and S. Connor, “An efficient implementation of parallel FDTD,” EMC 2007, (Honolulu (HI)), July 2007.

[33] W. Yu, X. Yang, Y. Lin, L. Ma, T. Su, N. Huang, R. Mittra, R. Maaskant, Y. Lu, Q. Che, R. Lu, and Z. Su, “A new direction in computational electromagnetics: Solving large problems using the parallel FDTD on the blugene/l supercomputer providing teraflop-level performance,” *IEEE Antennas and Propagat. Mag.*, vol. 50, no. 2, pp. 26–44, 2008.

[34] N. M. Josuttis, *The C++ Standard Library. A Tutorial and Reference.* Addison-Wesley, 1999.

[35] S. D. Gedney, “An anisotropic perfectly matched layer-absorbing medium for the truncation of FDTD lattices,” *IEEE Trans. Antenna Propagat.*, vol. 44, no. 12, pp. 1630–1639, 1996.

[36] V. E. Ferry, A. Polman, and H. A. Atwater, “Modeling light trapping in nanostructured solar cells,” *ACS Nano*, vol. 5, no. 12, pp. 10055–10064, 2011.

[37] U. W. Paetzold, E. Moulin, B. E. Pieters, U. Rau, and R. Carius, “Optical simulations of microcrystalline silicon solar cells applying plas-

monic reflection grating back contacts," *J. Photon. Energy.*, vol. 2, no. 1, p. 027002, 2012.

[38] A. Polman and H. A. Atwater, "Photonic design principles for ultrahigh-efficiency photovoltaics," *Nature Materials*, vol. 11, pp. 174–177, 2012.

[39] P. Spinelli, V. E. Ferry, J. van de Groep, M. van Lare, M. A. Verschuuren, R. E. I. Schropp, H. A. Atwater, and A. Polman, "Plasmonic light trapping in thin-film si solar cells," *J. Opt.*, vol. 14, no. 2, p. 024002, 2012.

[40] K. R. Catchpole and A. Polman, "Plasmonic solar cells," *Optics Express*, vol. 16, no. 26, pp. 21793–21800, 2008.

[41] A. Polman, "Plasmonics applied," *Science Magazine*, vol. 322, no. 5903, pp. 868–869, 2008.

[42] S. Mokkapati, F. J. Beck, R. de Waele, A. Polman, and K. R. Catchpole, "Resonant nano-antennas for light trapping in plasmonic solar cells," *J. Phys. D: Appl. Phys.*, vol. 44, no. 18, p. 185101, 2011.

[43] K. L. Kelly, E. Coronado, L. L. Zhao, and G. Schatz, "The optical properties of metal nanoparticles: The influence of size, shape, and dielectric environment," *Journal of Physical Chemistry B*, vol. 107, no. 3, pp. 668–677, 2003.

[44] M. Pelton, J. Aizpurua, and G. Bryant, "Metal-nanoparticle plasmonics," *Laser and Photonic Review*, vol. 2, no. 3, pp. 136–159, 2008.

[45] M. Knight, Y. Wu, J. B. Lassiter, P. Nordlander, and N. J. Halas, "Substrates matter: Influence of an adjacent dielectric on an individual plasmonic nanoparticle," *Nanoletters*, vol. 9, no. 5, pp. 2188–2192, 2009.

[46] H. A. Atwater and A. Polman, “Plasmonics for improved photovoltaic devices,” *Nature Materials*, vol. 9, pp. 205–213, 2010.

[47] D. M. Schaadt, B. Feng, and E. T. Yu, “Enhanced semiconductor optical absorption via surface plasmon excitation in metal nanoparticles,” *Applied Physics Letters*, vol. 86, p. 063106, 2005.

[48] S. Pillai, K. R. Catchpole, T. Trupke, and M. A. Green, “Surface plasmon enhanced silicon solar cells,” *Journal of Applied Physics*, vol. 101, p. 093105, 2007.

[49] T. L. Temple, G. D. K. Mahanama, H. S. Reehal, and D. M. Bagnall, “Influence of localized surface plasmon excitation in silver nanoparticles on the performance of silicon solar cells,” *Solar Energy Materials and Solar Cells*, vol. 93, pp. 1978–1985, 2009.

[50] S. Pillai and M. A. Green, “Plasmonics for photovoltaic applications,” *Solar Energy Materials and Solar Cells*, vol. 94, pp. 1481–1486, 2010.

[51] S. Pillai, F. J. Beck, K. R. Catchpole, Z. Ouyang, and M. A. Green, “The effect of dielectric spacer thickness on surface plasmon enhanced solar cells front and rear side deposition,” *Journal of Applied Physics*, vol. 109, p. 073105, 2011.

[52] I. Diukman and M. Orenstein, “How front side plasmonic nanostructures enhance solar cell efficiency,” *Solar Energy Materials and Solar Cells*, vol. 95, pp. 2628–2631, 2011.

[53] A. Chiappini, C. Armellini, A. Chiasera, M. Ferrari, L. Fortes, M. C. Gonalves, R. Guider, Y. Jestin, R. Retoux, G. N. Conti, S. Pelli, R. M. Almeida, and G. C. Righini, “An alternative method to obtain direct opal photonic crystal structures,” *Journal of Non-Crystalline Solids*, vol. 355, pp. 1167–1170, 2009.

[54] A. Chiappini, S. Guddala, C. Armellini, S. Berneschi, I. Cacciari, C. Duverger-Arfuso, M. Ferrari, and G. C. Righini, “Fabrication and characterization of colloidal crystals infiltrated with metallic nanoparticles,” *Proceedings of SPIE*, vol. 7725, p. 77250, 2010.

[55] A. Chiappini, A. Chiasera, S. Berneschi, C. Armellini, A. Carpentiero, M. Mazzola, E. Moser, S. Varas, G. C. Righini, and M. Ferrari, “Sol-gel-derived photonic structures: fabrication, assessment, and application,” *Journal of Sol Gel Science and Technology*, vol. 60, pp. 408–425, 2011.

[56] J. A. Stratton, *Electromagnetic Theory.* New York and London: McGraw-Hill, 1941.

[57] R. Pontalti, L. Cristoforetti, and L. Cescatti, “The frequency dependent FD-TD method for multi-frequency results in microwave hyperthermia treatment simulation,” *Phys. Med. Biol.*, vol. 38, pp. 1283–1298, 1993.

[58] A. Vial and T. Laroche, “Comparison of gold and silver dispersion laws suitable for FDTD simulations,” *Applied Physics B*, vol. 93, pp. 139–143, 2008.

[59] R. Courant, K. O. Friedrichs, and H. Lewy, “Uber di partiellen differenzenglei-chungen der mathematischen physik,” *Math. Ann.*, vol. 100, pp. 32–74, 1928. German original paper.

[60] R. Courant, K. O. Friedrichs, and H. Lewy, “On the partial difference equations of mathematical physics,” *IBM Jour. of Res. and Dev.*, vol. 11, pp. 215–234, 1967. English translation of the original paper.

[61] A. Taflove and M. E. Brodwin, “Numerical solution of steady-state electromagnetic scattering problems using the time-dependent

Maxwell's equations," *IEEE Trans. Microwave Theory Tech.*, vol. 23, pp. 623–630, 1975.

[62] G. Mur, "Absorbing boundary conditions for the finite difference approximation of the time domain electromagnetic field equations," *IEEE Trans. Electromagn. Comp.*, vol. 23, pp. 377–382, 1981.

[63] P. S. Pacheco, *Parallel Programming with MPI.* San Francisco: Morgan Kaufmann Publishers, 1997.

[64] W. Gropp, E. Lusk, and A. Skjellum, *Using MPI. Portable Parallel Programming with the Message-Passing Interface. Second Edition.* Cambridge, Mass.: The MIT Press, 1999.

[65] M. Mazzurana, L. Sandrini, A. Vaccari, C. Malacarne, L. Cristoforetti, and R. Pontalti, "A semi-automatic method for developing an anthropomorphic numerical model of dielectric anatomy by MRI," *Physics in Medicine and Biology*, vol. 48, no. 19, pp. 3157–3170, 2004.

[66] `http://www.mcs.anl.gov/mpi`. The MPICH implementation of the MPI libraries.

[67] `http://www.open-mpi.org`. The OpenMPI implementation of the MPI libraries official web site.

[68] `http://www.openmp.org`. The OpenMP official web site.

[69] R. Chandra, L. Dagun, D. Kohr, D. Maydan, J. McDonald, and R. Menon, *Parallel Programming in OpenMP.* San Francisco: Morgan Kaufmann Publishers, 2001.

[70] M. J. Quinn, *Parallel Programming in C with MPI and OpenMP.* McGraw-Hill, 2003.

[71] G. E. Karniadakis and R. M. K. II, *Parallel Scientific Computing in C++ and MPI.* Cambridge University Press, 2005.

[72] J. D. Jackson, *Classical Electrodynamics. Third Edition.* Wiley, 1998.

[73] A. Hirata, H. Sugiyama, and O. Fujiwara, "Estimation of core temperature elevation in humans and animals for whole-body averaged SAR," *Progress In Electromagnetics Research*, vol. PIER 99, pp. 53–70, 2009.

[74] M. T. Islam, M. R. I. Faruque, and N. Misran, "Design analysis of ferrite sheet attachment for SAR reduction in human head," *Progress In Electromagnetics Research*, vol. PIER 98, pp. 191–205, 2009.

[75] J. R. Cole and N. J. Halas, "Optimized plasmonic nanoparticle distributions for solar spectrum harvesting," *Applied Physics Letters*, vol. 89, p. 153120, 2006.

[76] Y. A. Akimov, K. Ostrikov, and E. P. Li, "Surface plasmon enhancement of optical absorption in thin-film silicon solar cells," *Plasmonics*, vol. 4, pp. 107–113, 2009.

[77] `http://rredc.nrel.gov/solar/spectra/am1.5`. Reference Solar Spectral Irradiance: Air Mass 1.5.

[78] P. Bharadwaj, B. Deutsch, and L. Novotny, "Optical antennas," *Advances in Optics and Photonics*, vol. 1, no. 3, 2009.

[79] E. D. Palik, *Handbook of Optical Constants of Solids.* Boston: Academic Press, 1985.

[80] K. Wostyn, Y. Zhao, G. de Schaetzen, L. Hellemans, N. Matsuda, K. Clays, and A. Persoons, "Insertion of a two-dimensional cavity into a self-assembled colloidal crystal," *Langmuir*, vol. 19, no. 10, pp. 4465–4468, 2003.