Institut Fresnel

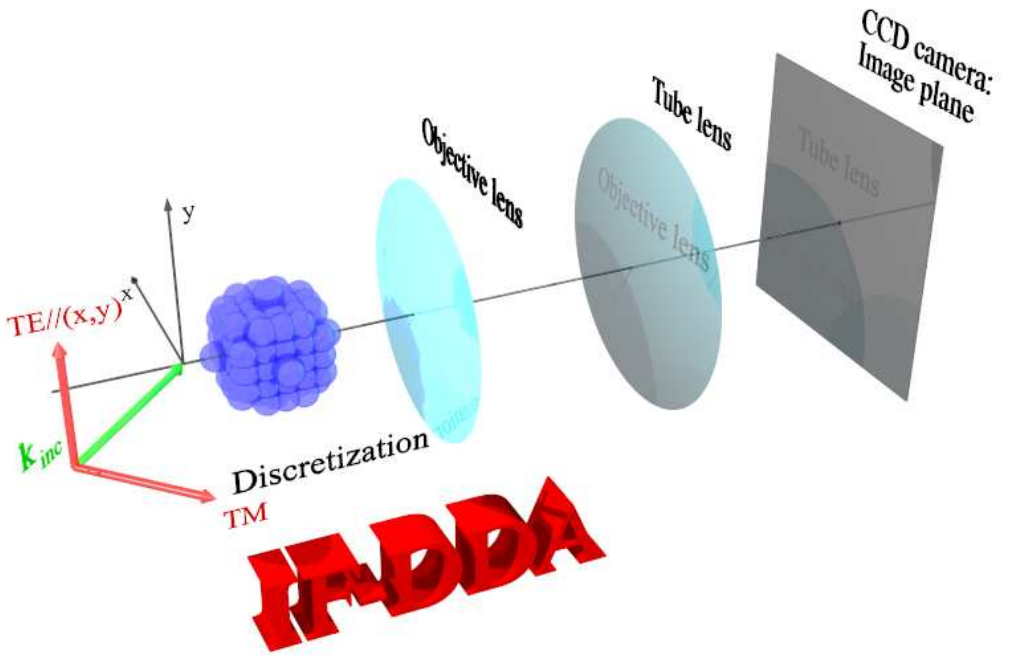
Patrick C. Chaumet

Daniel Sentenac

Anne Sentenac

IF-DDA

Idiot Friendly-Discrete Dipole Approximation



Contents

List of figures iv

1 Generality 1
- 1.1 Introduction . . . 1
- 1.2 The principle of discrete dipole approximation in a multilayer system . . . 2
- 1.3 A word about the code . . . 3
- 1.4 How to compile the code . . . 3
- 1.5 A word about the authors . . . 4
- 1.6 Licence . . . 4
- 1.7 How to quote the code . . . 4

2 Approximation to increase the efficiency of the code 5
- 2.1 Introduction . . . 5
- 2.2 Approximated method . . . 7
  - 2.2.1 Born . . . 7
  - 2.2.2 Renormalized Born . . . 7
  - 2.2.3 Born at the order 1 . . . 7
- 2.3 Computation of the Green function . . . 7

3 Numerical details 9
- 3.1 Polarizability . . . 9
- 3.2 Solve the system of linear equation . . . 10

4 Managing of the configurations 11
- 4.1 Introduction . . . 11
- 4.2 Creation and saving of a new configuration . . . 11
- 4.3 Managing of the configurations . . . 11

5 Properties of the illumination 12
- 5.1 Introduction . . . 12
- 5.2 Beam . . . 12
  - 5.2.1 Introduction . . . 12
  - 5.2.2 Linear plane wave . . . 13
  - 5.2.3 Circular plane wave . . . 14
  - 5.2.4 Multiple plane wave . . . 14
  - 5.2.5 Linear Gaussian beam . . . 14
  - 5.2.6 Circular Gaussian . . . 14

5.2.7 Speckle . . . 15
5.2.8 Arbitrary wave . . . 15

6 Properties of the multilayer 17
6.1 Introduction . . . 17
6.2 Remarks . . . 17

7 Definition of the object 19
7.1 Introduction . . . 19
7.2 Type of the object . . . 19
7.2.1 Sphere . . . 20
7.2.2 Inhomogeneous sphere . . . 20
7.2.3 Cube . . . 21
7.2.4 Cuboid (length) . . . 21
7.2.5 Cuboid (meshsize) . . . 21
7.2.6 Inhomogeneous Cuboid (length) . . . 22
7.2.7 Inhomogeneous Cuboid (meshsize) . . . 22
7.2.8 Ellipsoid . . . 22
7.2.9 Multiple spheres . . . 23
7.2.10 Cylinder . . . 23
7.2.11 Concentric spheres . . . 23
7.2.12 Arbitrary object . . . 23
7.3 Choose the relative permittivity . . . 24
7.4 Choose the discretization . . . 25

8 Possible study with the code 26
8.1 Introduction . . . 26
8.2 Study in far field . . . 26
8.3 Microscopy . . . 28
8.4 Study in near field . . . 29

9 Representation of the results 31
9.1 Introduction . . . 31
9.2 Digital exits . . . 31
9.3 Graphics . . . 32
9.3.1 Plot epsilon/dipoles . . . 32
9.3.2 Far field and microscopy . . . 32
9.3.2.1 Plot Poynting vector . . . 32
9.3.2.2 Plot microscopy . . . 32
9.3.3 Study of the near field . . . 33

10 Ouput files for matlab, octave, scilab,... 34
10.1 Introduction . . . 34
10.2 List of all exit files . . . 35

List of Figures

1.1 Principle of the DDA : the object under study (on the left) is discretized in a set of small dipoles (on the right)l. 2

5.1 Definition of the beam's direction 13
5.2 Definition of the beam's polarization. 13

6.1 How to configure the layer. 17

7.1 Definition of the angles of Euler according to the convention $Z - X - Z$. Scheme taken from Wikipedia 20

8.1 Simplified figure of the microscope. The object focus of the objective lens are at the origin of the frame but can be changed. The axis of the lens is confounded with the z axis. 28

Chapter 1

Generality

Contents

1.1 Introduction 1
1.2 The principle of discrete dipole approximation in a multilayer system 2
1.3 A word about the code 3
1.4 How to compile the code 3
1.5 A word about the authors 4
1.6 Licence 4
1.7 How to quote the code 4

1.1 Introduction

This software computes the diffraction of an electromagnetic wave by a three-dimensional object in a multilayer system. This interaction is taken into account rigorously by solving the Maxwell's equations, but can also do with the approximation of Born at the order 0 or 1. The code has an user-friendly interface and allows you to choose canonical objects (sphere, cube, ...) as well as predefined incident waves (plane wave, Gaussian beam, ...) or arbitrary objects and incidents waves. After by drop-down menus, it is easy to study cross sections, diffraction near field and far field as well as microscopy in transmission or reflection (holography, brigthfield, dark field,...).

There are numerous methods that enable the study of the diffraction of an electromagnetic wave by an object of arbitrary form and relative permittivity. We are not going here to set up an exhaustive list of these methods, but the curious reader may refer to the article by F. M. Kahnert who details the advantages and weaknesses of the most common methods.[1]

The method we use is called coupled dipoles method (CDM) or the discrete dipole approximation (DDA). This method is a volume method, because the diffracted field is obtained from an integral, the support of which is the volume of the considered object. It had been introduced by E. M. Purcell and C. R. Pennypacker in 1973, in order to study the scattering of light by grains in interstellar medium.[2]

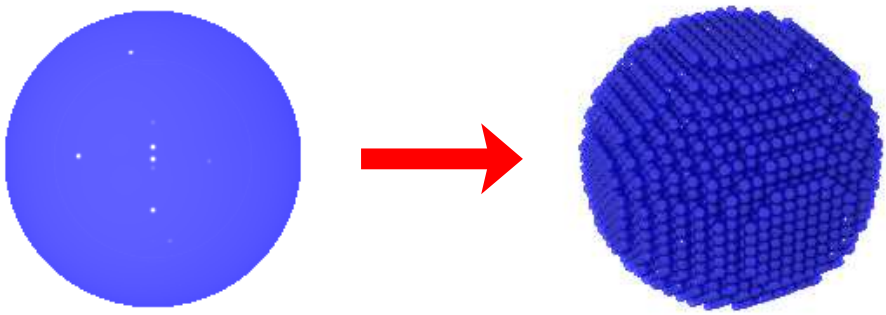


Figure 1.1 : *Principle of the DDA : the object under study (on the left) is discretized in a set of small dipoles (on the right)l.*

1.2 The principle of discrete dipole approximation in a multilayer system

Take an object of arbitrary form and relative permittivity in a multilayer system. This multilayer is submitted to a incident field electromagnetic wave of wavelength λ ($k_0 = 2\pi/\lambda$). In the absence of the object under study a reference field takes place in the multilayer. The principle of the DDA consists in representing the object as a set of N small cubes of an edge a [by little, we mean smaller than the wavelength in the object : $a \ll \lambda/\sqrt{\varepsilon}$ (Fig. 1.1)]. Each one of the small cubes under the action of the reference wave is going to get polarized, and as such, to acquire a dipolar moment, whose value is going to depend on the reference field and on its interaction with its neighbours. The local field of a dipole located at $\boldsymbol{r}_i$, $\boldsymbol{E}(\boldsymbol{r}_i)$, is the sum of the incident wave and the field radiated by the dipoles :

$$\boldsymbol{E}(\boldsymbol{r}_i) = \boldsymbol{E}_{\text{ref}}(\boldsymbol{r}_i) + \sum_{j=1}^{N} \boldsymbol{G}(\boldsymbol{r}_i, \boldsymbol{r}_j)\alpha(\boldsymbol{r}_j)\boldsymbol{E}(\boldsymbol{r}_j). \tag{1.1}$$

$\boldsymbol{E}_{\text{ref}}$ is the reference wave, $\boldsymbol{G}$ the linear susceptibility of the field of the multilayer system. α is the polarizability of each discretization element obtained from the Clausius-Mossotti relation. Note that the polarizability α, in order to respect the optical theorem, needs to contain a term called the radiative reaction term.[3] Equation (1.1) is valid for $i = 1, \cdots, N$, and so represents a system of $3N$ linear equations where the local fields, $\boldsymbol{E}(\boldsymbol{r}_i)$, being the unknowns. Once the system of linear equation is solved, the field scattered by the object at an arbitrary position $\boldsymbol{r}$ is obtained by making the sum of all the radiated fields by each one of the dipoles :

$$\boldsymbol{E}(\boldsymbol{r}) = \sum_{j=1}^{N} \boldsymbol{G}(\boldsymbol{r}, \boldsymbol{r}_j)\alpha(\boldsymbol{r}_j)\boldsymbol{E}(\boldsymbol{r}_j). \tag{1.2}$$

We have just presented the DDA as E. M. Purcell and C. R. Pennypacker had presented it earlier.[2] Note that another method very close to the DDA does exist. This method called the method of the moments starts from the integral equation of Lippman Schwinger, which is strictly identical to the DDA. The demonstration of the equivalence between these two methods being a little technical, it is explained in Ref.[4].

The advantages of the DDA are that it is applicable to objects of arbitrary forms, inhomogeneous (that is hardly achievable in case of surface method), and anisotropic (the polarizability associated to the mesh becomes a tensor). The outgoing wave condition is automatically satisfied through the linear susceptibility of the field. Finally, note that only the object is discretized unlike the methods of finite differences and finite elements.[1] The main inconvenience of the DDA consists in the fast increase of computation time together with the increase of the number of discretization elements, *i.e.*, the increase in size of the system of linear equations to be solved.

1.3 A word about the code

The code is thought to have a user-friendly interface so that everyone can use it without any problems including non specialists. This allows undergraduate students to study, for example, the basics of microscopy (Rayleigh's criteria, notion of numerical aperture, ...) or diffraction without any problem; and researchers, typically biologists, having no notion of Maxwell's equations to simulate what gives a microscope (brightfield, phase microscope, dark field, ...) in function of the usual parameters and the object. Nevertheless, this code can also serve physicists specializing in electromagnetism in performing, for example, calculations of diffraction, cross sections, near field and this with many incident beams.

The code thus has by default a simple interface where all numerical parameters are hidden and where many options are then chosen by default. But it's easy to access all Code options by checking the Advanced Interface option. This user guide explains how to use the advanced interface in starting with the different approaches used by the code to solve the Maxwell equations.

Note that the usability of the code is made to the detriment of the optimization of the RAM and the code can used large memory for large objects.

1.4 How to compile the code

The application is based on Qt-4.8 and gfortran To install it you need : qt, qt-devel, gcc-c++ et gfortran. Then to compile it:

- qmake-qt4
- make
- make install

To run the application, cd bin, and ./cdm.

Notice that there is three versions of the code, the first one is sequential and uses FFTS (Singleton's Fast Fourier Transform), the second one uses FFTW (Fast Fourier Transform in the west) which needs openmp 4.5 minimum and the third uses HDF5 format to save data file.

On linux system with the library FFTW, it requires to install FFTW packages with " dnf install * fftw * ". For the version that uses HDF5 file you should install the following packages "dnf install hdf hdf5 hdf5-static hdf5-devel".

The code works on windows system but it is tricky to compile it if you want to use FFTW.

1.5 A word about the authors

- P. C. Chaumet is Professor at Fresnel Institute of Aix-Marseille University, and deals with the development of the fortran source code.
- D. Sentenac develops the convivial interface of the code.
- A. Sentenac is research director at the CNRS, and works at Fresnel Institute of Aix-Marseille University, and participates to the development of the code connected to the far field diffraction.

1.6 Licence

1.7 How to quote the code

- P. C. Chaumet, A. Sentenac, and A. Rahmani,
  Coupled dipole method for scatterers with large permittivity.
  Phys. Rev. E **70**, 036606 (2004).
- S. Khadir, P. C. Chaumet, G. Baffou and A. Sentenac,
  Quantitative model of the image of a radiating dipole through a microscope.
  J. Opt. Soc. Am. A **36**, 478 (2019).

Chapter 2

Approximation to increase the efficiency of the code

Contents

2.1 Introduction 5
2.2 Approximated method 7
2.2.1 Born 7
2.2.2 Renormalized Born 7
2.2.3 Born at the order 1 7
2.3 Computation of the Green function 7

2.1 Introduction

In the previous chapter we have presented the DDA in a simple way where the object under study is a set of radiating dipole. In an approach more rigorous, with the Maxwell's equation, we get in Gaussian unit:

$$\boldsymbol{\nabla} \times \boldsymbol{E}^{\mathrm{m}}(\boldsymbol{r}) = i\frac{\omega}{c}\boldsymbol{B}(\boldsymbol{r}) \tag{2.1}$$

$$\boldsymbol{\nabla} \times \boldsymbol{B}(\boldsymbol{r}) = -i\frac{\omega}{c}\varepsilon(\boldsymbol{r})\boldsymbol{E}^{\mathrm{m}}(\boldsymbol{r}), \tag{2.2}$$

where $\varepsilon(\boldsymbol{r})$ denotes the relative permittivity of the object and $\boldsymbol{E}^{\mathrm{m}}$ the macroscopic field inside the object, then we get

$$\boldsymbol{\nabla} \times (\boldsymbol{\nabla} \times \boldsymbol{E}^{\mathrm{m}}(\boldsymbol{r})) = \varepsilon(\boldsymbol{r})k_0^2\boldsymbol{E}^{\mathrm{m}}(\boldsymbol{r}), \tag{2.3}$$

with $k_0 = \omega^2/c^2$. Using the relationship $\varepsilon = \varepsilon_{\mathrm{mul}} + 4\pi\chi$, where χ denotes the linear field susceptibility and $\varepsilon_{\mathrm{mul}}$ the relative permittivity of the multilayer system. Note that $\varepsilon_{\mathrm{mul}}$ depends only of z. Then, we have:

$$\boldsymbol{\nabla} \times (\boldsymbol{\nabla} \times \boldsymbol{E}^{\mathrm{m}}(\boldsymbol{r})) - \varepsilon_{\mathrm{mul}}k_0^2\boldsymbol{E}^{\mathrm{m}}(\boldsymbol{r}) = 4\pi\chi(\boldsymbol{r})k_0^2\boldsymbol{E}^{\mathrm{m}}(\boldsymbol{r}). \tag{2.4}$$

To solve this equation one needs the Green function defined as:

$$\boldsymbol{\nabla} \times (\boldsymbol{\nabla} \times \boldsymbol{G}(\boldsymbol{r}, \boldsymbol{r}')) - \varepsilon_{\mathrm{mul}}k_0^2\boldsymbol{G}(\boldsymbol{r}, \boldsymbol{r}') = 4\pi k_0^2\boldsymbol{I}\delta(\boldsymbol{r} - \boldsymbol{r}'), \tag{2.5}$$

and the solution of Eq. (2.4) reads:

$$\boldsymbol{E}^{\mathrm{m}}(\boldsymbol{r}) = \boldsymbol{E}_{\mathrm{ref}}(\boldsymbol{r}) + \int_{\Omega} \boldsymbol{G}(\boldsymbol{r},\boldsymbol{r}')\chi(\boldsymbol{r}')\boldsymbol{E}^{\mathrm{m}}(\boldsymbol{r}')\mathrm{d}\boldsymbol{r}', \tag{2.6}$$

where $\boldsymbol{E}_{\mathrm{ref}}$ is the field in the absence of the object and Ω the support of the object under study. When we solve Eq. (2.4) the field $\boldsymbol{E}^{\mathrm{m}}$ corresponds to the macroscopic field inside the object. To solve Eq. (2.4) we discretize the object in a set of N subunits with a cubic meshsize d, then the integral equation becomes the sum of N integrals:

$$\boldsymbol{E}^{\mathrm{m}}(\boldsymbol{r}_i) = \boldsymbol{E}_{\mathrm{ref}}(\boldsymbol{r}_i) + \sum_{j=1}^{N} \int_{V_j} \boldsymbol{G}(\boldsymbol{r}_i,\boldsymbol{r}')\chi(\boldsymbol{r}')\boldsymbol{E}^{\mathrm{m}}(\boldsymbol{r}')\mathrm{d}\boldsymbol{r}', \tag{2.7}$$

with $V_j = d^3$. Assuming the field, the Green function and the susceptibility constant over a subunit we get:

$$\boldsymbol{E}^{\mathrm{m}}(\boldsymbol{r}_i) = \boldsymbol{E}_{\mathrm{ref}}(\boldsymbol{r}_i) + \sum_{j=1}^{N} \boldsymbol{G}(\boldsymbol{r}_i,\boldsymbol{r}_j)\chi(\boldsymbol{r}_j)\boldsymbol{E}^{\mathrm{m}}(\boldsymbol{r}_j)d^3. \tag{2.8}$$

We can share into two parts the Green function, $\boldsymbol{G} = \boldsymbol{M} + \boldsymbol{T}$, where $\boldsymbol{M}$ is the Green function who take into account of the multiple reflection between the different layers and $\boldsymbol{T}$ is the Green function of the homogeneous space. Using, in first approximation (the radiative reaction term neglected) $\int_{V_i} \boldsymbol{T}(\boldsymbol{r}_i,\boldsymbol{r}')\mathrm{d}\boldsymbol{r}' = -4\pi/(3\varepsilon_{\mathrm{mul}}(\boldsymbol{r}_i))$[5], we get:

$$\boldsymbol{E}^{\mathrm{m}}(\boldsymbol{r}_i) = \boldsymbol{E}_{\mathrm{ref}}(\boldsymbol{r}_i) + \sum_{j=1}^{N} \boldsymbol{G}'(\boldsymbol{r}_i,\boldsymbol{r}_j)\chi(\boldsymbol{r}_j)d^3\boldsymbol{E}^{\mathrm{m}}(\boldsymbol{r}_j) - \frac{4\pi}{3\varepsilon_{\mathrm{mul}}(\boldsymbol{r}_i)}\chi(\boldsymbol{r}_i)\boldsymbol{E}^{\mathrm{m}}(\boldsymbol{r}_i), \tag{2.9}$$

where $\boldsymbol{G}' = \boldsymbol{M} + \boldsymbol{T}$ for $i \neq j$ and $\boldsymbol{G}' = \boldsymbol{M}$ for $i = j$, then we can write

$$\boldsymbol{E}(\boldsymbol{r}_i) = \boldsymbol{E}_{\mathrm{ref}}(\boldsymbol{r}_i) + \sum_{j=1}^{N} \boldsymbol{G}'(\boldsymbol{r}_i,\boldsymbol{r}_j)\alpha_{\mathrm{CM}}(\boldsymbol{r}_j)\boldsymbol{E}(\boldsymbol{r}_j) \tag{2.10}$$

$$\text{with} \qquad \boldsymbol{E}(\boldsymbol{r}_i) = \frac{\varepsilon(\boldsymbol{r}_i) + 2\varepsilon_{\mathrm{mul}}(\boldsymbol{r}_i)}{3\varepsilon_{\mathrm{mul}}(\boldsymbol{r}_i)}\boldsymbol{E}^{\mathrm{m}}(\boldsymbol{r}_i) \tag{2.11}$$

$$\alpha_{\mathrm{CM}}(\boldsymbol{r}_j) = \frac{3}{4\pi}\varepsilon_{\mathrm{mul}}(\boldsymbol{r}_i)d^3\frac{\varepsilon(\boldsymbol{r}_i) - \varepsilon_{\mathrm{mul}}(\boldsymbol{r}_i)}{\varepsilon(\boldsymbol{r}_i) + 2\varepsilon_{\mathrm{mul}}(\boldsymbol{r}_i)}. \tag{2.12}$$

The field $\boldsymbol{E}(\boldsymbol{r}_i)$ is the local field, *i.e.* the field at the position i in the absence of the subunit i. Then the linear system can be written formally as

$$\boldsymbol{E} = \boldsymbol{E}_{\mathrm{ref}} + \boldsymbol{A}\boldsymbol{D}_{\alpha}\boldsymbol{E}, \tag{2.13}$$

where $\boldsymbol{A}$ is a matrix which contains all the Green function and $\boldsymbol{D}_{\alpha}$ is a tridiagonal matrix with the polarizabilities of each element of discretization. In the next chapter we detail how to solve Eq. (2.13) rigorously, but in this present chapter we detail different approached methods to avoid the tedious resolution of Eq. (2.13). The scattered field is computed through

$$\boldsymbol{E}^{\mathrm{d}}(\boldsymbol{r}) = \sum_{j=1}^{N} \boldsymbol{G}(\boldsymbol{r},\boldsymbol{r}_j)\alpha(\boldsymbol{r}_j)\boldsymbol{E}(\boldsymbol{r}_j). \tag{2.14}$$

2.2 Approximated method

2.2.1 Born

The most simple approximation is the Born approximation which consists to assume the field inside the object equal to the reference field for each element of discretization:

$$\boldsymbol{E}^{\mathrm{m}}(\boldsymbol{r}_i) = \boldsymbol{E}_{\mathrm{ref}}(\boldsymbol{r}_i), \tag{2.15}$$

This approximation hold if the contrast is weak and the object small compare to the wavelength of illumination.

2.2.2 Renormalized Born

The renormalized Born approximation consists to assume the local field inside the object equal to the reference field :

$$\boldsymbol{E}(\boldsymbol{r}_i) = \boldsymbol{E}_{\mathrm{ref}}(\boldsymbol{r}_i). \tag{2.16}$$

In that case the macroscopic field reads:

$$\boldsymbol{E}^{\mathrm{m}}(\boldsymbol{r}_i) = \frac{3\varepsilon_{\mathrm{mul}}}{\varepsilon(\boldsymbol{r}_i) + 2\varepsilon_{\mathrm{mul}}} \boldsymbol{E}_{\mathrm{ref}}(\boldsymbol{r}_i). \tag{2.17}$$

This approximation is better that the classical Born approximation when the permittivity is high.

2.2.3 Born at the order 1

To be more precise that the renormalized Born approximation, one can perform the Born series at the order one:

$$\boldsymbol{E}(\boldsymbol{r}_i) = \boldsymbol{E}_{\mathrm{ref}}(\boldsymbol{r}_i) + \sum_{j=1}^{N} \boldsymbol{G}'(\boldsymbol{r}_i, \boldsymbol{r}_j)\alpha(\boldsymbol{r}_j)\boldsymbol{E}_{\mathrm{ref}}(\boldsymbol{r}_j). \tag{2.18}$$

In that case we take into account the simple scattering.

2.3 Computation of the Green function

Even though we are using an efficient integration scheme to evaluate the Green tensor, see Ref. 6, it takes a lot of time to evaluate it for all the different pair of points covering the object. Note that due to the translational invariance of the reference medium in the (x, y) plane, $\boldsymbol{G}(\boldsymbol{r}, \boldsymbol{r}') = \boldsymbol{G}(\|\boldsymbol{r}_{\|} - \boldsymbol{r}'_{\|}\|, z, z')$. For each couple (z, z'), the number of pairs with different distances of a Cartesian (x, y) mesh with $n_x \times n_y$ points $(n_x > n_y)$ is equal to $n_y(2n_x - n_y + 1)/2$. To accelerate the computation, we approximate $\boldsymbol{G}(\|\boldsymbol{r}_{\|} - \boldsymbol{r}'_{\|}\|, z, z')$ using an interpolation of a discrete set of points, $\boldsymbol{G}(qd/n_d, z, z')$ with $q = 1, \cdots, \mathrm{int}\sqrt{n_x^2 + n_y^2}$ and n_d a natural number. Linear and polynomial interpolations could not evaluate the Green tensor properly when $\|\boldsymbol{r}_{\|} - \boldsymbol{r}'_{\|}\| < \lambda$, as the fast decay of the evanescent waves was not accounted for accurately. We obtained much better results using rational functions, that is quotients of polynomials. Rational functions have the ability to model functions with poles (Press et al. 1986) and permit an accurate approximation of the $1/r^3$ behavior of the Green tensor in the near field range.

In the code, in the section numerical parameters, the drop menu Green function permits to choose to compute rigorously the Green function or evaluate the Green tensor with interpolation for $n_d = 1, 2, 3, 4$. Notice that by default the code uses $n_d = 2$ which is enough accurate. Note that only the computation with interpolation is parallelized.

Chapter 3

Numerical details

Contents

3.1 Polarizability 9
3.2 Solve the system of linear equation 10

3.1 Polarizability

The DDA discretizes the object into a set of punctual dipoles, where a polarizability α is associated to each punctual dipoles. There are different forms for this polarizability. The first to have been used, and the simplest, is the relation of Clausius Mossotti (CM)[2]:

$$\alpha_{\mathrm{CM}} = \frac{3}{4\pi}\varepsilon_{\mathrm{mul}}\frac{\varepsilon-\varepsilon_{\mathrm{mul}}}{\varepsilon+2\varepsilon_{\mathrm{mul}}}d^3 = \varepsilon_{\mathrm{mul}}\frac{\varepsilon-\varepsilon_{\mathrm{mul}}}{\varepsilon+2\varepsilon_{\mathrm{mul}}}a^3, \tag{3.1}$$

where ε denotes the permittivity of the object, d the size of the cubic meshsize and $a = \left(\frac{3}{4\pi}\right)^{\frac{1}{3}} d$ the radius of the sphere of the same volume than the cubic meshsize of the side d. Unfortunately, this relation does not keep the energy and, then, it is necessary to introduce a radiative reaction term that takes into account the fact that charges in movement lose energy, and the polarizability is, then, written as [3]:

$$\alpha_{\mathrm{RR}} = \frac{\alpha_{\mathrm{CM}}}{1-\frac{2}{3}ik_0^3 n_{\mathrm{mul}}\alpha_{\mathrm{CM}}}. \tag{3.2}$$

After different forms of the polarizability have been established in order to improve the precision of the DDA and take into account the non-punctual character of the dipole, and we may quote, among the best known, the ones by Goedecke and O'Brien[7],

$$\alpha_{\mathrm{GB}} = \frac{\alpha_{\mathrm{CM}}}{1-\frac{2}{3}ik_0^3 n_{\mathrm{mul}}\alpha_{\mathrm{CM}} - k_0^2\alpha_{\mathrm{CM}}/a}, \tag{3.3}$$

by Lakhtakia[8]:

$$\alpha_{\mathrm{LA}} = \frac{\alpha_{\mathrm{CM}}}{1-2\frac{\varepsilon-\varepsilon_{\mathrm{mul}}}{\varepsilon+2\varepsilon_{\mathrm{mul}}}\left[(1-ik_0 n_{\mathrm{mul}}a)e^{ik_0 n_{\mathrm{mul}}a}-1\right]} \tag{3.4}$$

and Draine and Goodman[9]

$$\alpha_{\mathrm{LR}} = \frac{\alpha_{\mathrm{CM}}}{1+\alpha_{\mathrm{CM}}\left[\frac{(b_1+\varepsilon b_2/\varepsilon_{\mathrm{mul}}+\varepsilon b_3/\varepsilon_{\mathrm{mul}}S)k_0^2}{d}-\frac{2}{3}in_{\mathrm{mul}}k_0^3\right]}, \tag{3.5}$$

with $b_1 = -1.891531$, $b_2 = 0.1618469$, $b_3 = -1.7700004$ and $S = 1/5$.

Inside the code by default, it is α_{RR} which is used. In the case when the permittivity is anisotropic only α_{RR} is going to be used.

A last polarizability is introduced (PS) that only works for homogeneous spheres and is particularly precise for metals. This consists of making a change of the polarizability of the elements on the edge of the sphere taking into account the factor of depolarization of the sphere.[10] Note that the sphere should be embedded in only one layer.

3.2 Solve the system of linear equation

In order to know the electric field in the object, *i.e.* the field at the position of the N elements of discretization, we have to solve the following system of linear equation:

$$\boldsymbol{E} = \boldsymbol{E}_0 + \boldsymbol{A} D_\alpha \boldsymbol{E}, \tag{3.6}$$

where $\boldsymbol{E}_0$ is a vector of size $3N$ which contains the incident field at the discretization elements. $\boldsymbol{A}$ is a matrix $3N \times 3N$ which contains all the field tensor susceptibility and D_α is a diagonal matrix $3N \times 3N$, if the object is isotropic, or diagonal block 3×3 if the object is anisotropic. $\boldsymbol{E}$ is the vector $3N$ which contains the unknown electric local fields. The equation is solved by a non-linear iterative method. The code proposes numerous iterative methods, and the one used by default is GPBICG because it is the most efficient in most cases [11]. The code stops when the residue,

$$r = \frac{\|\boldsymbol{E} - \boldsymbol{A} D_\alpha \boldsymbol{E} - \boldsymbol{E}_0\|}{\|\boldsymbol{E}_0\|}, \tag{3.7}$$

is under the tolerance given by the user. 10^{-4} is the tolerance used by default, because it is a good compromise between speed and precision. Please find below the different iterative method possible in the code:

- GPBICG1 : Ref. 12
- GPBICG2 : Ref. 12
- GPBICGsafe : Ref. 13
- GPBICGAR1 : Ref. 12
- GPBICGAR2 : Ref. 12
- QMRCLA : Ref. 14
- TFQMR : Ref. 14
- CG : Ref. 14
- BICGSTAB : Ref. 14
- QMRBICGSTAB1 : Ref. 15
- QMRBICGSTAB2 : Ref. 15
- GPBICOR : Ref. 16
- CORS : Ref. 17
- BiCGstar-plus Ref. 18

Chapter 4

Managing of the configurations

Contents

4.1 Introduction 11
4.2 Creation and saving of a new configuration 11
4.3 Managing of the configurations 11

4.1 Introduction

The Code is launched by ./cdm inside the bin folder for a Linux configuration. It has been created to be as convenient as possible and so needing few explanations for its use. However, certain conventions have been taken and need to be clarified.

4.2 Creation and saving of a new configuration

In order to start a new calculation, go to the tab *calculation* and *New*. A new configuration shows up with values by default. Once the new configuration is chosen, in order to be saved, the tab *Calculation* and *Save* have to be selected again. Then, we select the name of the configuration, and we may add a short description of the calculation that has been made. Another way to save a configuration is to click directly on the panel of the configuration *Save configuration*. Then, two fields appear, one for the name of the configuration and the second one for its description.

4.3 Managing of the configurations

In order to manage all the selected configurations, we have go to the tab *Calculation* and *Load*. So, a new window appears with all the saved configurations. For each configuration there is a short description that the user has entered, the date, when the configuration file has been saved, then the principal characteristics of the configuration (wave length, power, the beam's waist, object, material, discretization and tolerance of the iterative method). It is enough to click on a configuration and to click on *load* in order to load a configuration.

The *delete* button is used to delete a saved configuration and the *export* enables to export inside a file (name of the configuration.opt) all the characteristics of the configuration.

Note that by double clicking on the line, we can modify the description field.

Chapter 5

Properties of the illumination

Contents

5.1 Introduction 12
5.2 Beam 12
5.2.1 Introduction 12
5.2.2 Linear plane wave 13
5.2.3 Circular plane wave 14
5.2.4 Multiple plane wave 14
5.2.5 Linear Gaussian beam 14
5.2.6 Circular Gaussian 14
5.2.7 Speckle 15
5.2.8 Arbitrary wave 15

5.1 Introduction

In the section properties of the illumination, the field *Wavelength* enables us to enter the using wavelength in vacuum. This one is entered in nanometer. The field P_0 enables to enter the power of the laser beam in Watt. The field W_0 in nanometer enables to enter for a plane wave the radius of the laser beam and for a Gaussian beam, the waist of the beam.

Note that the beam always propagates in the direction of the positive z axis, hence for $k_z > 0$ whatever the beam chosen.

5.2 Beam

5.2.1 Introduction

There are six beams predefined, their propagation direction is always defined in the same way, with two angles θ and φ, except for he speckle. They are connected to the given direction by the wave vector as follows:

$$k_x = k_0 \sin\theta \cos\varphi \tag{5.1}$$
$$k_y = k_0 \sin\theta \sin\varphi \tag{5.2}$$
$$k_z = k_0 \cos\theta \tag{5.3}$$

where $\boldsymbol{k}_0 = (k_x, k_y, k_z)$ is the wave vector parallel to the direction of the incident beam and k_0 the wave number, see Fig. 5.1. For the polarization, we use the plane (x, y) as

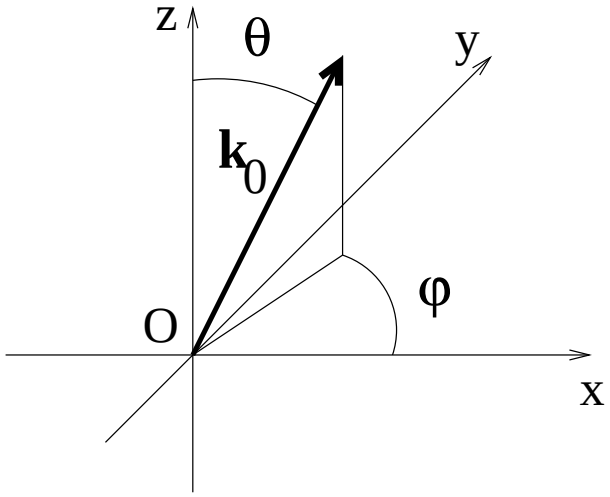


Figure 5.1 : *Definition of the beam's direction*

referential surface. Then, we can determine a polarization TM (p) and TE (s) with the presence of a surface, see Fig. 5.2 or a polarization along he x or y axis, depending of the beam. The frame (x, y, z) is used as an absolute referential. We define the polarization

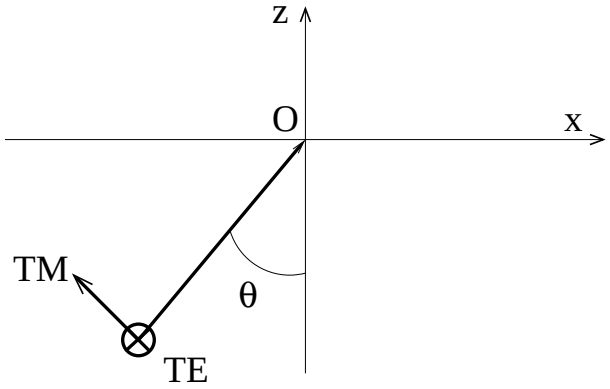


Figure 5.2 : *Definition of the beam's polarization.*

vector $\boldsymbol{s}$ corresponding to TE polarisation and $\boldsymbol{p}$ corresponding to TM as,

$$\boldsymbol{s} = \frac{\hat{\boldsymbol{z}} \times \boldsymbol{k}_0}{\|\hat{\boldsymbol{z}} \times \boldsymbol{k}_0\|} \tag{5.4}$$

$$\boldsymbol{p} = \frac{\boldsymbol{k}_0 \times \boldsymbol{s}}{\|\boldsymbol{k}_0 \times \boldsymbol{s}\|}. \tag{5.5}$$

5.2.2 Linear plane wave

Linear plane wave is a plane wave linearly polarized. The first line is relative to θ and the second to φ. The third line is connected to the polarization, polarization TM(1) or TE(0) as $\boldsymbol{E}_0 = E_p\boldsymbol{p} + E_s\boldsymbol{s}$ with $E_p = \text{pola}$ and $E_s = \sqrt{1 - \text{pola}^2}$. Note that the

polarization is not necessarily purely in TE or TM: pola $\in$ [0 1] such as $E^2_{\mathrm{TM}} = \mathrm{pola}^2 E^2$ and $E^2_{\mathrm{TE}} = (1 - \mathrm{pola}^2)E^2$.

Notice that if one wants that $\boldsymbol{E}_0 \cdot \hat{\boldsymbol{x}} = 0$ ($\boldsymbol{E_0} \cdot \hat{\boldsymbol{y}} = 0$), one can choose pola=3 (2) and the code will compute the right value of pola to get the polarization asked.

Note that the phase is always taken null at the origin of the frame, with Irradiance = P_0/S where $S = \pi w_0^2$ is the surface of the beam and $E_0 = \sqrt{2\mathrm{Irradiance}/c/\varepsilon_0}$.

5.2.3 Circular plane wave

pwavecircular is a plane wave circularly polarized. The first line is relative to θ and the second to φ. The third line is connected to the polarization that we can choose right (1) or left (-1) circular.

Note that the phase is taken null at the origin of the frame, with Irradiance = P_0/S where $S = \pi w_0^2$ is the surface of the beam and $E_0 = \sqrt{2\mathrm{Irradiance}/c/\varepsilon_0}$.

5.2.4 Multiple plane wave

Multiple wave consists to take many planes waves. The first thing to do is to choose the number of plane wave, and then for each plane wave we choose θ and φ and the polarization. We have to write also the complex magnitude of each plane wave. The sum of the power of all the plane wave is equal to P_0.

5.2.5 Linear Gaussian beam

Linear Gaussian is a Gaussian wave polarized linearly. The first line is relative to θ and the second to φ. The third line is connected to the angle between the polarization and the x axis (0) or along the y axis (90).

The three following lines help to fix the position of the centre (x_0, y_0, z_0) of the waist in nanometers in the frame (x, y, z).

Note that this Gaussian beam may have a very weak waist, because it is calculated without any approximation through an angular spectrum representation done with FFT with always $k_z > 0$ whatever the inclination θ of the Beam. The definition of the waist, for a beam propagating along the z axis is :[19]

$$E(x, y, 0) = E_0 e^{-\rho^2/(2w_0^2)}, \tag{5.6}$$

with $\rho = \sqrt{x^2 + y^2}$. Then to compute a Gaussian beam polarized along the x axis, we have the magnitude of the Fourier component as:

$$\boldsymbol{A}(k_x, k_y) = E_0(k_z \boldsymbol{i} - k_x \boldsymbol{k}) \frac{1}{\sqrt{k_x^2 + k_z^2}} w_0 e^{-(k_x^2 + k_y^2)w_0^2/2}, \tag{5.7}$$

then we compute the reference field, $\boldsymbol{A}_{\mathrm{ref}}(k_x, k_y, z)$, through the multilayer system, and the reference field reads:

$$\boldsymbol{E}_{\mathrm{ref}}(x, y, z) = \int\int_{k_0} \boldsymbol{A}_{\mathrm{ref}}(k_x, k_y, z) e^{i(k_x(x-x_0) + k_y(y-y_0) - k_z z_0)} \mathrm{d}\boldsymbol{k}_{\|}. \tag{5.8}$$

5.2.6 Circular Gaussian

Circular Gaussian is a Gaussian wave circularly polarized. The first line is relative to θ and the second to φ. The third line is connected to the polarization that we can choose right (1) or left (-1) circular.

The next three lines enable us to fix the position of the centre of the waist in nanometers in the frame (x, y, z).

Note that this Gaussian wave may have a very weak waist, because it is calculated without any approximation through a plane wave spectrum.

5.2.7 Speckle

Speckle is done as the Gaussian beam with a FFT but with a magnitude with a random phase. For a speckle polarized along the x axis the magnitude of the Fourier component is:

$$\boldsymbol{A}(k_x, k_y) = E_0(k_z \boldsymbol{i} - k_x \boldsymbol{k}) \frac{1}{\sqrt{k_x^2 + k_z^2}} e^{i\varphi}, \tag{5.9}$$

where φ is random variable between 0 and 2π. Then we compute the reference field, $\boldsymbol{A}_{\text{ref}}(k_x, k_y, z)$, through the multilayer system, and the reference field reads:

$$\boldsymbol{E}_{\text{ref}}(x, y, z) = \int\int_{k_0 \text{NA}} \boldsymbol{A}_{\text{ref}}(k_x, k_y, z) e^{i(k_x(x-x_0)+k_y(y-y_0)-k_z z_0)} \mathrm{d}\boldsymbol{k}_{\parallel}, \tag{5.10}$$

where NA is the numerical aperture of the microscope. $\boldsymbol{r}_0$ permits to shift the speckle and the seed to change the distribution of the speckle.

5.2.8 Arbitrary wave

In the case of an arbitrary field, the characteristic are determined by the user. In other words, he has to create the field himself, and it is mandatory to create these files respecting the chosen conventions by the code.

The description of the discretization of the incident field is done within a file which is asked for when we click on *Props*. For example, for the real part of the component x of the field, it has to be constructed as follows:

nx,ny,nz
dx,dy,dz
xmin,ymin,zmin

- nx is the number of meshsize according to the axis x
- ny is the number of meshsize according to the axis y
- nz is the number of meshsize according to the axis z
- dx is the step according to the axis x
- dy is the step according to the axis y
- dz is the step according to the axis z
- xmin the smallest abscissa
- ymin the smallest ordinate
- zmin the smallest azimuth

Then, the files of the electric field are created as follows for each of the components of the real part and separated imaginary field:

```
open(11, file='Exr.mat', status='new', form='formatted', access='direct', recl=22)
do k=1,nz
    do j=1,ny
        do i=1,nx
            ii=i+nx*(j-1)+nx*ny*(k-1)
            write(11,FMT='(D22.15)',rec=ii) dreal(Ex)
        enddo
    enddo
enddo
```

Be careful, the mesh size of the discretization of the object has to be larger than the meshsize of the discretization of the field.

Chapter 6

Properties of the multilayer

Contents

6.1 Introduction 17
6.2 Remarks 17

6.1 Introduction

In the section properties of the multilayer, you should first choose the number of interface. Note that you can not choose zero, as in that case the free space code is more efficient. Once the number n of interface is fixed click on *Props*. It appears a new window with n interfaces and $n+1$ media. The first medium corresponds to the substrate, *i.e.* the medium through which the light comes. The permittivity of this medium is obviously transparent (no absorption). Then the position of the first interface in nanometer is asked followed by the second medium, etc.

An example is given in Fig. 6.1

The last medium corresponds to the superstrate.

6.2 Remarks

- The last medium if one studies a microscope in transmission can not be absorbing.

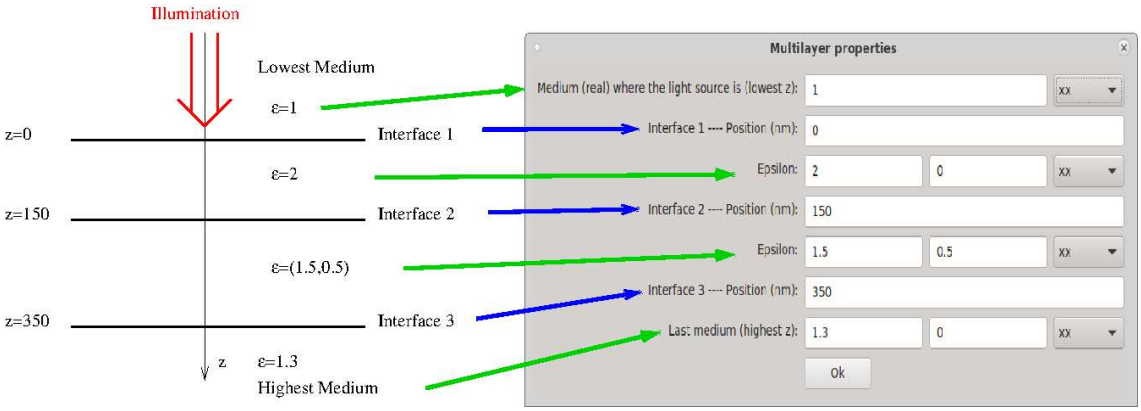


Figure 6.1 : *How to configure the layer.*

- If the object under study contains one or more interfaces, note that any dipole (element of discretization) can be on a interface, then the code slightly moves the interface (less than one half mesh) to avoid this.
- The code is limited to 10 interfaces.
- Notice that if there is a large distance between two layers (several hundred of wavelength) the computation of the Green function can fail.
- The multilayer support guided mode resonance as the Green function is computed with the residue theorem.

Chapter 7

Definition of the object

Contents

7.1 Introduction . . . 19
7.2 Type of the object . . . 19
7.2.1 Sphere . . . 20
7.2.2 Inhomogeneous sphere . . . 20
7.2.3 Cube . . . 21
7.2.4 Cuboid (length) . . . 21
7.2.5 Cuboid (meshsize) . . . 21
7.2.6 Inhomogeneous Cuboid (length) . . . 22
7.2.7 Inhomogeneous Cuboid (meshsize) . . . 22
7.2.8 Ellipsoid . . . 22
7.2.9 Multiple spheres . . . 23
7.2.10 Cylinder . . . 23
7.2.11 Concentric spheres . . . 23
7.2.12 Arbitrary object . . . 23
7.3 Choose the relative permittivity . . . 24
7.4 Choose the discretization . . . 25

7.1 Introduction

The code proposes several predefined objects, and we are going to precise in this section how to enter their optogeometrical characteristics. Note that all the distances have to be entered in nanometers. The code is doing the conversion in meters.

7.2 Type of the object

The list of the predefined objects is the following:

sphere, cube, cuboid, ellipsoid, cylinder, concentric spheres, multiple spheres, inhomogeneous sphere, inhomogeneous cuboid and arbitrary object.

When the objects as the cuboid, cylinder or ellipsoid have their edges turned with respect to the axes of the system of coordinates, the angles of Euler are used as defined

in Fig. 7.1. The rotation centre being the inertia centre of the object and the matrix of rotation reads:

$$\boldsymbol{A} = \begin{pmatrix} \cos(\psi)\cos(\varphi) - \sin(\psi)\cos(\theta)\sin(\varphi) & -\cos(\psi)\sin(\varphi) - \sin(\psi)\cos(\theta)\cos(\varphi) & \sin(\psi)\sin(\theta) \\ \sin(\psi)\cos(\varphi) + \cos(\psi)\cos(\theta)\sin(\varphi) & -\sin(\psi)\sin(\varphi) + \cos(\psi)\cos(\theta)\cos(\varphi) & -\cos(\psi)\sin(\theta) \\ \sin(\theta)\sin(\varphi) & \sin(\theta)\cos(\varphi) & \cos(\theta) \end{pmatrix}$$

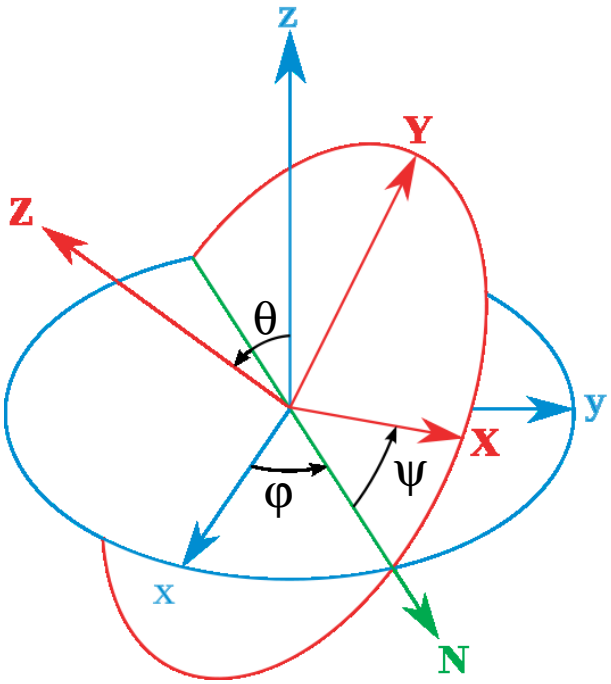


Figure 7.1 : *Definition of the angles of Euler according to the convention $Z - X - Z$. Scheme taken from Wikipedia*

7.2.1 Sphere

For the sphere, there are four fields to be filled:

- The radius of the sphere in nanometer
- The abscissa of the centre of the sphere in nanometer
- The ordinate of the centre of the sphere in nanometer
- The azimuth of the centre of the sphere in nanometer

7.2.2 Inhomogeneous sphere

The permittivity of the sphere have a Gaussian noise with a correlation length l_c, standard deviation, A and an average ε_r.

For the inhomogeneous sphere there are seven fields to be filled:

- The radius of the sphere in nanometer
- The seed

- The abscissa of the centre of the sphere in nanometer
- The ordinate of the centre of the sphere in nanometer
- The azimuth of the centre of the sphere in nanometer
- The correlation length l_c
- The standard deviation A

7.2.3 Cube

For the cube, there are four fields to be filled:

- The edge of the cube in nanometer
- The abscissa of the centre of the sphere in nanometer
- The ordinate of the centre of the sphere in nanometer
- The azimuth of the centre of the sphere in nanometer

7.2.4 Cuboid (length)

For the cuboid, there are nine fields to be filled:

- The edge of the cube in nanometer according to the axis x
- The edge of the cube in nanometer according to the axis y
- The edge of the cube in nanometer according to the axis z
- The abscissa of the centre of the cuboid in nanometer
- The ordinate of the centre of the cuboid in nanometer
- The azimuth of the centre of the sphere in nanometer
- First angle of Euler ψ by rotation around the axis z
- Second angle of Euler θ by rotation around the axis x
- Third angle of Euler φ by rotation around the axis z

7.2.5 Cuboid (meshsize)

For the cuboid, there are seven fields to be filled:

- The abscissa of the centre of the cuboid in nanometer
- The ordinate of the centre of the cuboid in nanometer
- The azimuth of the centre of the sphere in nanometer
- Number of meshsize long x
- Number of meshsize long y
- Number of meshsize long z
- Meshsize in nanometer

7.2.6 Inhomogeneous Cuboid (length)

The permittivity of the cuboid have a Gaussian noise with a correlation length l_c, standard deviation A and an average ε_r. For the cuboid, there are nine fields to be filled:

- The edge of the cube in nanometer according to the axis x
- The edge of the cube in nanometer according to the axis y
- The edge of the cube in nanometer according to the axis z
- The abscissa of the centre of the cuboid in nanometer
- The ordinate of the centre of the cuboid in nanometer
- The azimuth of the centre of the sphere in nanometer
- The seed
- The correlation length l_c
- The standard deviation A

7.2.7 Inhomogeneous Cuboid (meshsize)

The permittivity of the cuboid have a Gaussian noise with a correlation length l_c, standard deviation A and an average ε_r. For the cuboid, there are nine fields to be filled:

- The abscissa of the centre of the cuboid in nanometer
- The ordinate of the centre of the cuboid in nanometer
- The azimuth of the centre of the sphere in nanometer
- Number of meshsize long x
- Number of meshsize long y
- Number of meshsize long z
- Meshsize in nanometer
- The seed
- The correlation length l_c
- The standard deviation A

7.2.8 Ellipsoid

For the ellipsoid, there are nine fields to be fulfilled:

- The half axis in nanometer according to the axis x
- The half axis in nanometer according to the axis y
- The half axis in nanometer according to the axis z
- The abscissa of the centre of the ellipse in nanometer

- The ordinate of the centre of the ellipse in nanometer
- The azimuth of the centre of the ellipse in nanometer
- First angle of Euler ψ by rotation around the axis z
- Second angle of Euler θ by rotation around the axis x
- Third angle of Euler φ by rotation around the axis z

7.2.9 Multiple spheres

For multiple spheres, it is convenient first to choose with the line from the under *number of objects* the number N of the expected spheres. Then, when we click on *Props N* windows, that we fill in the same way as for the unique sphere, appear. Beware, the spheres must be disconnected, otherwise, the code stops and shows error.

7.2.10 Cylinder

For the cylinder, there are eight fields to be fulfilled:

- The radius of the cylinder in nanometers
- The length of the cylinder in nanometer
- The abscissa of the centre of the cylinder in nanometer
- The ordinate of the centre of the cylinder in nanometer
- The azimuth of the centre of the cylinder in nanometer
- First angle of Euler ψ by rotation around the axis z
- Second angle of Euler θ by rotation around the axis x
- Third angle of Euler φ by rotation around the axis z

7.2.11 Concentric spheres

For concentric spheres, it is convenient first to choose with the under line *number of objects* the number N of concentric spheres. Then, when we click on *Props N* windows appear. The first window is filled the same way as for the sphere, and for the next windows, it is enough to enter the radius in nanometer. The radii must be entered in increasing order, otherwise, the code shows the error.

7.2.12 Arbitrary object

In the case of an arbitrary object, it is defined by the user. In other words, he has to create the object himself, and then, it is convenient to create this entry file by respecting the conventions chosen by the code. *namefile* is the name of the file containing the arbitrary object and it is asked for when we choose the arbitrary object. It is coded in sequential and in ascii, and is necessarily described inside a cuboid box. Below are given the lines of the code enabling to create this file:

```
open(15,file=namefile,status='old',iostat=ierror)
write(15,*) nx,ny,nz
```

```
write(15,*) aretecube
do i=1,nz
   do j=1,ny
      do k=1,nx
         write(15,*) xs(i,j,k),ys(i,j,k),zs(i,j,k)
      enddo
   enddo
enddo
do i=1,nz
   do j=1,ny
      do k=1,nx
         if objet isotrope
            write(15,*) eps(i,j,k)
         elseif objet anisotrope
            do ii=1,3
               do jj=1,3
                  write(15,*) epsani(ii,jj,i,j,k)
               enddo
            enddo
         endif
      enddo
   enddo
enddo
```

- nx : size of the cuboid according to the axis x.
- ny : size of the cuboid according to the axis y.
- nz : size of the cuboid according to the axis z.
- aretecube : size of the meshsize of discretization in nanometers.
- x : abscissa of the mesh of discretization according the axis x.
- y : ordinate of the mesh of discretization according the axis y.
- z : azimuth of the mesh of discretization according the axis z.
- eps : epsilon of the object if isotropic
- epsani : epsilon of the object if anisotropic

7.3 Choose the relative permittivity

When the object or objects are chosen, it is then convenient to enter the relative permittivity. For the homogeneous object they may be isotropic or anisotropic. So, we choose *iso* or *aniso* and we click on *Epsilon*.

- *iso*: A board appears, where either we enter the relative permittivity by hand (real and imaginary part) or we choose a material in the data base.
- *aniso*: A board appears where we enter the relative permittivity by hand (real and imaginary part) for all the components of anisotropic tensor.

7.4 Choose the discretization

The number N_c entered in the field of the discretization corresponds to the number of layers forming the object in its largest direction.

A few examples:

- For an ellipse of half axis (a, b, c), it is going to be the greatest half axis a that is going to be selected and the edge of discretization is going to be of $2a/N_c$.
- For a cube the number of meshsize is so going to be of $N = N_c^3$.

Chapter 8

Possible study with the code

Contents

8.1 Introduction 26
8.2 Study in far field 26
8.3 Microscopy 28
8.4 Study in near field 29

8.1 Introduction

To determine the object with the appropriate orientation is not an easy task. That is why the first option *Only dipoles with epsilon*, enables us to check quickly if the object entered is well the one intended without any calculation being launched. Once this has been done, there are three great fields: the study in far field, the study in near field and the optical forces.

Important: Note that in the DDA the computation that takes the longest time is the calculation of the local field due to the necessity to solve the system of linear equations. One option has been added which consists in reading again the local field starting with a file. When this option is selected, the name of a file is asked for; either we enter an old file or a new name:

- If this is a new name, the calculation of the local field is going to be accomplished, then, stored together with the chosen configuration.
- If this is an old name, the local field is going to be read again with a checking that the configuration has not been changed between the writing and the second reading. This makes it easier to relaunch calculations very quickly for the same configuration but for different studies.

8.2 Study in far field

When the option far field is selected, three possibilities appear:

- *Cross section*: This option enables us to calculate the extinction (C_{ext}), absorbing (C_{abs}) and scattering cross section (C_{sca}). The scattering cross section is obtained

through $C_{\text{sca}} = C_{\text{ext}} - C_{\text{abs}}$. The extinction and absorption cross sections may be evaluated as:

$$C_{\text{ext}} = \frac{4\pi k_0}{\|\boldsymbol{E}_0\|^2} \sum_{j=1}^{N} \text{Im}\left[\boldsymbol{E}_0^*(\boldsymbol{r}_j).\boldsymbol{p}(\boldsymbol{r}_j)\right] \tag{8.1}$$

$$C_{\text{abs}} = \frac{4\pi k_0}{\|\boldsymbol{E}_0\|^2} \sum_{j=1}^{N} \left[\text{Im}\left[\boldsymbol{p}(\boldsymbol{r}_j).(\alpha^{-1}(\boldsymbol{r}_j))^* \boldsymbol{p}^*(\boldsymbol{r}_j)\right] - \frac{2}{3} k_0^3 \|\boldsymbol{p}^*(\boldsymbol{r}_j)\|^2\right]. \tag{8.2}$$

Note that the cross section can only be computed if the background is homogeneous. The free space code is more adapted in that case, but it permits to check easily if the discretization is well adapted when a sphere is studied. Notice that the cross section can be computed if there is no interface.

- *Cross section+Poynting*: This option calculates also the scattering cross section from the integration of the far field diffracted by the object upon 4π steradians, the asymmetric factor when the background is homogeneous, else it computes only the differential cross section, *i.e.* $\langle \mathcal{P} \rangle = \langle \boldsymbol{S} \rangle .\boldsymbol{n} R^2$ with $\boldsymbol{S}$ the Poynting vector, $\boldsymbol{n}$ the direction of observation, which is going to be represented in 3D. The values *Ntheta* and *Nphi* enable us to give the number of points used in order to calculate the scattering cross and to represent the Poynting vector. The larger the object is, the larger *Ntheta* and *Nphi* must be, which leads to time consuming calculations for objects of several wavelengths.

$$C_{\text{sca}} = \frac{k_0^4}{\|\boldsymbol{E}_0\|^2} \int \left\| \sum_{j=1}^{N} \left[\boldsymbol{p}(\boldsymbol{r}_j) - \boldsymbol{n}(\boldsymbol{n}.\boldsymbol{p}(\boldsymbol{r}_j))\right] e^{-ik_0 \boldsymbol{n}.\boldsymbol{r}_j} \right\|^2 \text{d}\Omega \tag{8.3}$$

$$g = \frac{k_0^3}{C_{\text{sca}} \|\boldsymbol{E}_0\|^2} \int \boldsymbol{n}.\boldsymbol{k}_0 \left\| \sum_{j=1}^{N} \left[\boldsymbol{p}(\boldsymbol{r}_j) - \boldsymbol{n}(\boldsymbol{n}.\boldsymbol{p}(\boldsymbol{r}_j))\right] e^{-ik_0 \boldsymbol{n}.\boldsymbol{r}_j} \right\|^2 \text{d}\Omega \tag{8.4}$$

$$\frac{\text{d}\langle \mathcal{P} \rangle}{\text{d}\Omega} = \frac{1}{2} c \varepsilon_0 n \left\| \boldsymbol{E}_{\text{d}}(\boldsymbol{k}_{\|}) \right\|^2, \tag{8.5}$$

where $\boldsymbol{E}_{\text{d}}(\boldsymbol{k}_{\|})$ is the diffracted field in far field.

A solution in order to go faster (option *quick computation*) and to pass by FFT for the calculation of the diffracted field. In this case, of course, it is convenient to discretize keeping in mind that the relation $\Delta x \Delta k = 2\pi/N$ connects the mesh size of the discretization with the size of the FFT. This is convenient for objects larger than the wavelength. Indeed, $L = N\Delta x$ corresponds to the size of the object which gives $\Delta k = 2\pi/L$, and if the size of the object is too small, then, the Δk is too large, and the quadrature is imprecise. Note that since the integration is performed on two planes parallel to the plane (x, y), is not convenient if the incident makes an angle more than 70 degrees with the z axis. The 3D representation of the vector of Poynting is done as previously, i.e. with *Ntheta* and *Nphi* starting with an interpolation upon the calculated points with the FFT.

- *Emissivity*. This study computes the reflectance, transmittance and absorptance. If the object under study is no absorbing then the absorptance should be zero. Then it traduces the level of energy conservation of our solver. It can depend of the precision of the iterative method and of the polarizability chosen.

8.3 Microscopy

This option permits to compute the image obtained for different microscope (holographic, brightfield, darkfield and phase). We consider a microscope made of an objective lens and a tube lens in $4f$ configuration and sine-Abbe condition[20] It asks for the numerical aperture of the objective lens (NA $= n_{\text{obj}} \sin\theta_{\text{obj}}$) and the numerical aperture of the condenser lens ($sin\theta_{\text{cond}}$) for brightfield, darkfield and phase microscope. By default, the lenses are placed parallel to the plane (x, y) and their optical axis are confounded with the z axis. The focus plane of the lenses are placed to the origin of the frame but can be changed via the field "Position of the focal plane" for the microscope in transmission and reflection (Fig. 8.1). The magnification of the microscope is M and should be above 1. The drop menu propose three different microscope.

- *Holographic*: This option computes the diffracted field (Fourier plane) with the incident field defined in the section illumination properties. It computes the image plane with or without the presence of the incident field.
- *Brightfield*: This microscope uses a condenser lens, which focuses light from the light source onto the sample with a numerical aperture defined below the magnification. It consists to sum incoherently the image obtained with many incident field inside this numerical aperture with different polarization, hence it can take time as it needs to solve many direct problem. The result is given in the image plane without the incident field (a kind of dark field) and with the incident field (brightfield).
- *Darkfield & phase*: In darkfield microscopy the condenser is designed to form a hollow cone of light with a numerical aperture equal to the condenser lens, as apposed to brightfield microscopy that illuminates the sample with a full cone of light. The result is given in the image plane (scattered field). In the phase microscopy the ring-shaped illuminating light that passes the condenser annulus is focused on the specimen by the condenser exactly as in the dark field microscope and then the incident field with a phase shifted of $\pi/2$ is added to the scattered field.

The drop menu side computation permits to simulate microscope in transmission (Side $k_Z > 0$), in reflection (Side $k_Z > 0$), or both cases.

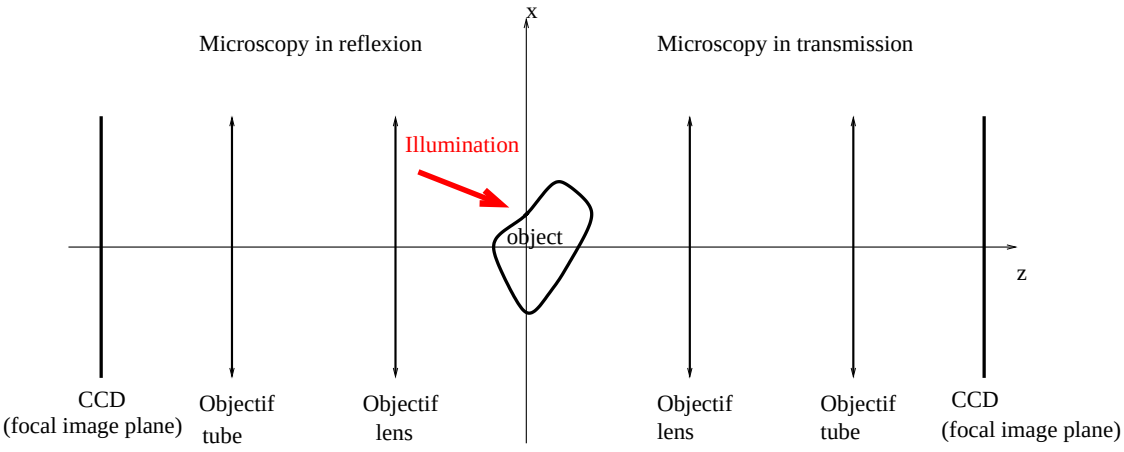


Figure 8.1 : *Simplified figure of the microscope. The object focus of the objective lens are at the origin of the frame but can be changed. The axis of the lens is confounded with the z axis.*

The calculation for the diffracted field may be completed starting with the sum of the radiation of the dipoles (very long when the object has a lot of dipoles) or with FFT (option *quick computation*) with a value $N = 128$ by default here as well. In this case, $\Delta x \Delta k = 2\pi/N$ with Δx the mesh size of discretization of the object which corresponds also to the discretization of the picture plane. Consequently, this one has a size of $L = GN\Delta x$.

The principle of the computation of the far field diffracted by the object and how to get the image through a microscope with a magnifying factor $M > 1$ has been detailed in Ref[21]. Then we recall it briefly. The diffracted field in far field at a distance r of the origin in the direction (k_x, k_y) can be written as $\boldsymbol{E} = \boldsymbol{S}(k_x, k_y)\frac{e^{ikr}}{r}$. The field after the first lens (field in the Fourier space) is then defined as: $\boldsymbol{e}(k_\parallel) = \frac{\boldsymbol{S}(k_x,k_y)}{-2i\pi\gamma}$ with $\gamma = \sqrt{\epsilon_{\text{mul}}k_0^2 - k_x^2 - k_y^2}$ where the value of ϵ_{mul} corresponds to the permittivity of the substrate for the microscope in reflection and to the permittivity of the superstrate for the microscope in transmission. A microscope transforms a plane wave with wavevector $\boldsymbol{k}$ into a plane wave with wavevector $\boldsymbol{k}'$ with $\boldsymbol{k}' = [\boldsymbol{k}'_\parallel, \gamma']$ where $\boldsymbol{k}'_\parallel = (-k_x/M, -k_y/M)$ and $\gamma' = \sqrt{k_0^2 - k_\parallel'^2}$ (the refractive index in the image space is considered equal to 1). Then the field in the image space, after the tube lens, reads as:

$$\boldsymbol{E}_{\text{ob}}(\boldsymbol{r}) = \frac{1}{M}\iint \sqrt{\frac{\gamma}{\gamma'}}\tilde{h}(\boldsymbol{k}_\parallel)\boldsymbol{e}'(\boldsymbol{k}_\parallel)\exp[i\boldsymbol{k}'\cdot(\boldsymbol{r}-\boldsymbol{r}_f)]\mathrm{d}\boldsymbol{k}_\parallel, \tag{8.6}$$

where $\tilde{h}(\boldsymbol{k}_\parallel)$ is cutoff function allowing the transmission of only the signal included in the numerical aperture (NA) of the objective lens, it reads $\tilde{h}(\boldsymbol{k}_\parallel) = 1$ for $\mid \boldsymbol{k}_\parallel \mid < k_0\text{NA}$ and 0 elsewhere and $\boldsymbol{r}_f$ the position of the lens. We have

$$\boldsymbol{e}'(\boldsymbol{k}_\parallel) = \boldsymbol{R}(\boldsymbol{k}_\parallel)\boldsymbol{e}(\boldsymbol{k}_\parallel), \tag{8.7}$$

and $\boldsymbol{R}(\boldsymbol{k}_\parallel)$ is given by:

$$\boldsymbol{R}(\boldsymbol{k}_\parallel) = \begin{pmatrix} u_x^2(1-\cos\theta)+\cos\theta & u_xu_y(1-\cos\theta) & u_y\sin\theta \\ u_xu_y(1-\cos\theta) & u_y^2(1-\cos\theta)+\cos\theta & -u_x\sin\theta \\ -u_y\sin\theta & u_x\sin\theta & \cos\theta \end{pmatrix}, \tag{8.8}$$

where $\boldsymbol{u} = \frac{\hat{\boldsymbol{k}}\times\boldsymbol{z}}{|\hat{\boldsymbol{k}}\times\boldsymbol{z}|}$ is the rotation axis. Notice that $\boldsymbol{u}$ has no component along the z direction. θ is defined as $\cos\theta = \hat{\boldsymbol{k}}.\hat{\boldsymbol{k}}'$ and $\sin\theta = \|\hat{\boldsymbol{k}}\times\hat{\boldsymbol{k}}'\|$

8.4 Study in near field

When the option near field is selected, two possibilities appear:

- *Local field*: This option enables us to draw the local field to the position of each element of discretization. The local field being the field at the position of each element of discretization in absence of itself.
- *Macroscopic field*: This option enables us to draw the macroscopic field to the position of each element of discretization. The connection between the local field and the macroscopic field is given Ref.[4] :

$$\boldsymbol{E}_{\text{macro}} = 3\varepsilon_{\text{mul}}\left(\varepsilon + 2\varepsilon_{\text{mul}} - i\frac{k_0^3 d^3 \varepsilon_{\text{mul}}^{3/2}}{2\pi}(\varepsilon - \varepsilon_{\text{mul}})\right)^{-1}\boldsymbol{E}_{\text{local}} \tag{8.9}$$

The last option enables us to choose the mesh in which the local and macroscopic fields are represented.

- *Object*: Only the field in the object is represented. Notice that when FFT is used for the beam or for the computation of the diffracted field then this options is passed in the option *Cube*. This is same for the computation of the emissivity, the reread option and the use of the BPM(R).
- *Cube*: The field is represented within a cuboid containing the object.
- *Wide field*: The field is represented within a box greater than the object. The size of the box correspond to the size of the object plus the Additional side band (x, y or z) on each side. For example for a sphere with a radius $r = 100$ nm and discretization of 10, *i.e.* a meshsize of 10 nm, with an Additional side band x of 2, 3 for y and 4 for z, we get a box of size:

$$l_x = 100 + 2 \times 2 \times 10 = 140 \text{ nm} \tag{8.10}$$
$$l_y = 100 + 2 \times 3 \times 10 = 160 \text{ nm} \tag{8.11}$$
$$l_z = 100 + 2 \times 4 \times 10 = 180 \text{ nm} \tag{8.12}$$

The field inside the wide field area in near field is computed with

$$\boldsymbol{E} = \boldsymbol{E}_0 + \boldsymbol{ADE}, \tag{8.13}$$

which gives the field inside the near field zone and in the object.

Chapter 9

Representation of the results

Contents

9.1 Introduction **31**
9.2 Digital exits **31**
9.3 Graphics **32**
9.3.1 Plot epsilon/dipoles 32
9.3.2 Far field and microscopy 32
9.3.3 Study of the near field 33

9.1 Introduction

Three windows enable us to manage and represent the requested results. The one on the top enables us to manage the different figures; the one at the bottom on the left present the digital values requested, and the one at the bottom on the right is kept for the graphic representations.

9.2 Digital exits

All the results are given in the SI system.

- *Object subunits*: Number of elements of discretization of the object under study.
- *Mesh subunits* : Number of elements of discretization of the cuboid containing the object under study.
- *Mesh size* : Size of the element of discretization.
- $\lambda/(10\mathsf{n})$: In order to obtain a good precision, it is advised to have a discretization under the value of $\lambda/10$ in the considered material of optical index n.
- k_0 :Wave number.
- *Irradiance*: Beam irradiance, for a Gaussian beam, it is estimated at the center of the waist.
- *Field modulus*: Modulus of the field, for a Gaussian beam, it is estimated at the center of the waist.

- *Tolerance obtained*: Tolerance obtained for the chosen iterative method. Logically under the requested value.
- *Number of products Ax (iterations)*: Number of matrix vector products completed by the iterative method. Between brackets the iteration number of the iterative method.
- *Absorptivity* Fraction of radiation absorbed in %, equal to zero if all the permittivity are real.
- *Reflectivity* Fraction of radiation reflected in %.
- *Tansmittivity* Fraction of radiation transmitted in %.
- *Extinction cross section*: Value of the extinction cross section.
- *Absorbing cross section*: Value of the absorbing cross section.
- *Scattering cross section*: Value of the scattering cross section obtained by = extinction cross section- absorbing cross section.
- *Scattering cross section with integration*: Value of the scattering cross section obtained by integration of the far field field radiated by the object.
- *Scattering asymmetric parameter*: Asymmetric factor.

9.3 Graphics

9.3.1 Plot epsilon/dipoles

The button *Plot epsilon/dipoles* enables us to see the position of each element of discretization. The colour of each point is associated with the value of the permittivity of the considered meshsize.

9.3.2 Far field and microscopy

9.3.2.1 Plot Poynting vector

Plot Poynting: enables us to draw the modulus of the Poynting vector in 3D. If the option quick computation is taken, then the results come from a interpolation of the points in the (k_x, k_y) space: if it is not smooth increase the number of point of the FFT.

9.3.2.2 Plot microscopy

Plot microscopy : enables us to draw the diffracted field in the Fourier plane for holographic microscope or the image through the dark field, brightfield, phase or holographic microscope. We can plot the modulus, intensity or the component x, y or z.

We have 8 different plot:

- *Fourier plane: Scattered field:* $k_z > 0$ The diffracted field by the object in the Fourier plane for a holographic microscope in transmission.
- *Fourier plane: Total field:* $k_z > 0$ The diffracted field by the object plus the incident field (if the incident field is a plane wave, then we have a Dirac) in the Fourier plane for a holographic microscope in transmission.

- *Fourier plane: Scattered field:* $k_z < 0$ The diffracted field by the object in the Fourier plane for a holographic microscope in reflection.
- *Fourier plane: Total field:* $k_z < 0$ The diffracted field by the object plus the incident field reflected on the multilayer system (if the incident field is a plane wave, then we have a Dirac) in the Fourier plane for a holographic microscope in reflection.
- *Image plane: Scattered field:* $z > 0$ Image of the diffracted field by the object through the microscope in transmission.
- *Image plane: Total field:* $z > 0$ Image of the diffracted field by the object plus the incident field through the microscope in transmission.
- *Image plane: Scattered field:* $z < 0$ Image of the diffracted field by the object through the microscope in reflection.
- *Image plane: Total field:* $z < 0$ Image of the diffracted field by the object plus the incident field through the microscope in reflection.

Remarks

- The diffracted field is represented upon a regular mesh in $\Delta k_x = \Delta k_y$ such as $\sqrt{k_x^2 + k_y^2} \leq k_0$ NA. If the computation is done by radiation of the dipoles, then the code choose to have at least 21 points in the numerical aperture. If quick computation is chosen then he code used FFT transform, then, the size of the picture is fixed by discretization of the object d with the relation $d\Delta k = 2\pi/N$ and N the size of the FFT.
- The computation in the image plane is always done with FFT.
- For the brightfield, darkfield and phase microscope only the field in the image plane can be plotted. When the option x, y and z component is chosen, the phase can not be plotted as we sum incoherently all the incidences, then we get only the modulus.

9.3.3 Study of the near field

- The first button *Field* enables us to choose to represent the incident field, local field or macroscopic field.
- The button *Type* enables us to represent the modulus or the component x, y or z of the studied field.
- The button *Cross section* x (y or z) enables us to choose the abscissa of the cut (ordinate or dimension). *Plot* x (y or z) draws the cut in plane x. *Plot all* x draws all the cut at once.

Chapter 10

Ouput files for matlab, octave, scilab,...

Contents

10.1 Introduction 34
10.2 List of all exit files 35

10.1 Introduction

It is not necessary to use the graphic interface of the program to watch the results. For the scalar, all the results are in the output file and for the pictures, it is possible to use directly the exit files in ascii or in one hdf5 file and to read them through other software such as Matlab, Octave, Scilab,...For example in the directory bin the field ifdda.m uses matlab to represent the different data.

When the advanced option is chosen, it is possible to choose to save the data either in separate .mat files or in a single hdf5 file.

- In the case of the hdf5 file, there are six created groups: option (the options chosen by the user), near field (the near field data), microscopy (data from the microscopy), far field (data from the far field option), and dipole (position of the elements of discretization and permittivity).
- In the case of .mat file, all the output are formatted in the form of a unique column vector or two column vectors if the number is a complex (the real part being associated with the first column and the imaginary part with the second column).
- In the hdf5 file all the data are formatted under the form of a single column vector and with two separate tables in the case of complex numbers.
- In the case where the file contains three-dimensional data, these ones are always stored as follows:

```
do i=1,nz
  do j=1,ny
    do k=1,nx
      write(*,*) data(i,j,k)
```

enddo

enddo

enddo

Three-dimensional data are going to be recognized by 3D at the beginning of the line.

10.2 List of all exit files

All the files have the extension .mat if you choose the output in ascii file.

- x,y,z represent the different used coordinates.
- (3D) epsilon contain the permittivity of the object.
- (3D) xc,yc,zc contain the coordinates of all the points of the mesh.
- (3D) xwf,ywf,zwf contain the coordinates of all the points of the mesh in which the near field is calculated when the wide field option (wide field) is used.
- (3D complex) incidentfieldx (y,z) contains the component x(y,z) of the incidental field only inside the object.
- (3D) incidentfield contains the modulus of the incident field only inside the object.
- (3D complexe) macroscopicfieldx (y,z) contains the component x(y,z) of the macroscopic field only inside the object.
- (3D) contains the modulus of the macroscopic field only inside the object.
- (3D complexe) localfieldx (y,z) contains the component x(y,z) of the local field only inside the object.
- (3D) localfield contains the modulus of the local field only inside the object.
- (3D complexe) incidentfieldxwf (y,z) contains the component x(y,z) of the incident field inside the box of near field in wide field.
- (3D) incidentfieldwf contains the modulus of the incidental field inside the box of near field in wide field.
- (3D complexe) macroscopicfieldxwf (y,z) contains the component x (y,z) of the macroscopic field inside the box of near field in wide field.
- (3D) macroscopicfieldwf contains the modulus of the macroscopic field inside the box of near field in wide field.
- (3D complexe) localfieldxwf (y,z) contains the component x(y,z) of the local field inside the box of near field in wide field.
- (3D) localfieldwf contains the modulus of the local field inside the box of near field in wide field.
- theta is a board which contains all the theta angles corresponding to all the directions in which the vector of Poynting is calculated. Its size is (Ntheta+1)*Nphi.

- phi is a board which contains all the theta angles corresponding to all the directions in which the vector of Poynting is calculated. Its size is (Ntheta+1)*Nphi.
- (2D) poynting poynting contains the modulus of the vector of Poynting in theta and phi direction of size (Ntheta+1)*Nphi.
- (2D) poyntingneg and poyntingpos contain the Poynting modulus for $k_z < 0$ and $k_z > 0$ respectively in the plane (k_x, k_y).
- kx and ky contain the coordinates for poyntingneg and poyntingpos.
- (2D) fourierpos(x,y,z) contains the diffracted field in the fourier plane in modulus (x,y,z) for $k_z > 0$.
- (2D) fourierposinc(x,y,z) contains the total field in the fourier plane in modulus (x,y,z) for $k_z > 0$.
- (2D) fourierneg(x,y,z) contains the diffracted field in the fourier plane in modulus (x,y,z) for $k_z < 0$.
- (2D) fourierneginc(x,y,z) contains the total field in the fourier plane in modulus (x,y,z) for $k_z < 0$.
- kxfourier (kyfourier) contains the abscissa (ordinate) of the Fourier plane.
- (2D) imagepos(x,y,z) contains the diffracted field in the image plane in modulus (x,y,z) for $z > 0$.
- (2D) imageposinc(x,y,z) contains the total field in the image plane in modulus (x,y,z) for $z > 0$.
- (2D) imageneg(x,y,z) contains the diffracted field in the image plane in modulus (x,y,z) for $z < 0$.
- (2D) imageneginc(x,y,z) contains the total field in the image plane in modulus (x,y,z) for $z < 0$.
- (2D) imagebfpos(x,y,z) contains the diffracted field in the image plane in modulus (x,y,z) for a dark field with illumination inside the numerical aperture of the condenser for $z > 0$.
- (2D) imageincbfpos(x,y,z) contains the total field in the image plane in modulus (x,y,z) for a brightfield microscope for $z > 0$.
- (2D) imagebfneg(x,y,z) contains the diffracted field in the image plane in modulus (x,y,z) for a dark field with illumination inside the numerical aperture of the condenser for $z < 0$.
- (2D) imageincbfneg(x,y,z) contains the total field in the image plane in modulus (x,y,z) for a brightfield microscope for $z < 0$.
- (2D) imagedfpos(x,y,z) contains the diffracted field in the image plane in modulus (x,y,z) for a dark field with illumination with a hollow cone of light with a numerical aperture equal to the condenser lens for $z > 0$.
- (2D) imageincdfpos(x,y,z)contains field in the image plane in modulus (x,y,z) for a phase microscope for $z > 0$.

- (2D) imagedfneg(x,y,z) contains the diffracted field in the image plane in modulus (x,y,z) for a dark field with illumination with a hollow cone of light with a numerical aperture equal to the condenser lens for $z < 0$.
- (2D) imageincdfneg(x,y,z)contains field in the image plane in modulus (x,y,z) for a phase microscope for $z < 0$.
- (2D) Ediffkposx (y,z) contains the diffracted field by the object for $k_z > 0$ in the (k_x, k_y) plane.
- (2D) Ediffknegx (y,z) contains the diffracted field by the object for $k_z < 0$ in the (k_x, k_y) plane.

Bibliography

[1] F. M. Kahnert, J. Quant. Spect. Rad. Transf. **79-80**, 775 (2003).

[2] E. M. Purcell and C. R. Pennypacker, Astrophys. J. **186**, 705 (1973).

[3] B. T. Draine, Astrophys. J. **333**, 848 (1988).

[4] P. C. Chaumet, A. Sentenac, and A. Rahmani, Phys. Rev. E **70**, 036606 (2004).

[5] A. D. Yaghjian, Proceedings of the IEEE **68**, 248 (1980), ISSN 0018-9219.

[6] M. Paulus, P. Gay-Balmaz, and O. J. F. Martin, Phys. Rev. E **62**, 5797 (2000).

[7] G. H. Goedecke and S. G. O'Brien, Appl. Opt. **27**, 2431 (1988).

[8] A. Lakhtakia, Int. J. Mod. Phys. C **3**, 583 (1992).

[9] B. T. Draine and J. Goodman, Astrophys. J. **405**, 685 (1993).

[10] A. Rahmani, P. C. Chaumet, and G. W. Bryant, Astrophys. J. **607**, 873 (2004).

[11] P. C. Chaumet and A. Rahmani, Opt. Lett. **34**, 917 (2009).

[12] M. Thuthu, S. Fujino, and Y. Onoue, IMECS **1** (2009).

[13] S. Fujino and T. Sekimoto, IMECS **2** (2012).

[14] R. D. Da Cunha and T. Hopkins, Appl. Numer. Math. **19**, 33 (1995).

[15] T. F. Chan, E. Gallopoulos, V. Simoncini, T. Szeto, and C. H. Tong, SIAM J. Sci. Comput. **15**, 338 (1994).

[16] L. Zhao, T.-Z. Huang, Y.-F. Jing, and L.-J. Deng, Computers & Mathematics with Applications **66**, 1372 (2013), ISSN 0898-1221.

[17] B. Carpentieri, Y.-F. Jing, T.-Z. Huang, W.-C. Pi, and X.-Q. Sheng, Computational Electromagnetics International Workshop pp. 85–90 (2011).

[18] S. Fujino and K. Iwasato, Proc. of the World Congress on Engineering **1** (2015).

[19] G. P. Agrawal and D. N. Pattanayak, J. Opt. Soc. Am. A **69**, 575 (1979).

[20] E. Abbe, Journal of the Royal Microscopical Society **1**, 388 (1981).

[21] S. Khadir, P. C. Chaumet, G. Baffou, and A. Sentenac, J. Opt. Soc. Am. A **36**, 478 (2019).