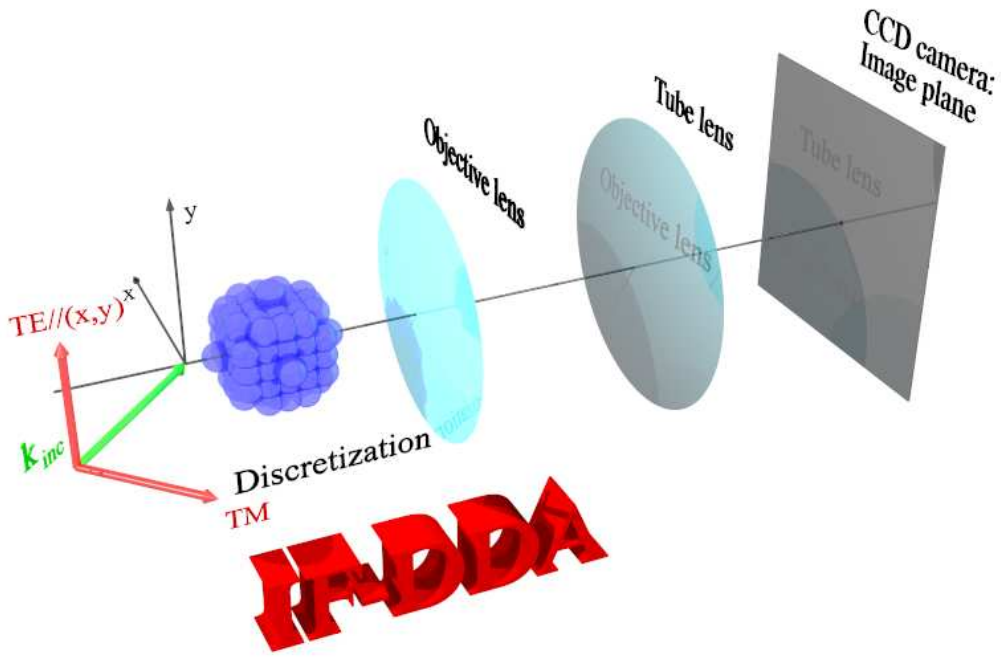
Institut Fresnel

Patrick C. Chaumet
Daniel Sentenac
Anne Sentenac

IFDDA
Idiot Friendly Discrete Dipole Approximation

version : 2.5.3



Contents

List of figures **vi**

1 Generality **1**
- 1.1 Introduction . . . 1
- 1.2 The principle of discrete dipole approximation . . . 2
- 1.3 A word about the code . . . 4
- 1.4 How to compile the code . . . 4
  - 1.4.1 CPU . . . 4
  - 1.4.2 GPU . . . 5
  - 1.4.3 Using command line code . . . 6
    - 1.4.3.1 Compile and run the code . . . 6
    - 1.4.3.2 How Command-Line Parsing Works . . . 6
    - 1.4.3.3 Implementation Notes and Important Behaviours 7
    - 1.4.3.4 Defaults & Initialization . . . 7
    - 1.4.3.5 Command-Line Important Behaviours . . . 8
    - 1.4.3.6 Some examples . . . 8
- 1.5 A word about the authors . . . 8
- 1.6 Licence . . . 9
- 1.7 How to quote the code . . . 9

2 Approximated method **11**
- 2.1 Introduction . . . 11
- 2.2 Approximated method . . . 13
  - 2.2.1 Born . . . 13
  - 2.2.2 Renormalized Born . . . 13
  - 2.2.3 Born at the order 1 . . . 13
  - 2.2.4 Rytov . . . 14
  - 2.2.5 Renormalized Rytov . . . 14
  - 2.2.6 Beam propagation method (BPM) . . . 14
  - 2.2.7 Renormalized BPM . . . 15

2.2.8 Classical scalar approximation 15
2.2.9 Scalar Approximation revisited 15
2.2.10 Multilayer Born approximation 16
2.2.10.1 Scalar MLB 16
2.2.10.2 Vectorial MLB 16
2.2.10.3 MLB with reflection 17
2.2.10.4 Summary of the different possibilities with the MLB 18

3 Numerical details 19
3.1 Polarizability 19
3.2 Correction to the tensor of susceptibility 20
3.3 Filtered Green's function 21
3.4 Solve the system of linear equation 21
3.5 Change of the initial guess 22
3.6 Preconditioning the system of linear equations 23
3.7 The default options and how to change them 23

4 Managing of the configurations 24
4.1 Introduction 24
4.2 Creation and saving of a new configuration 24
4.3 Managing of the configurations 24

5 Properties of the illumination 26
5.1 Introduction 26
5.2 Beam 27
5.2.1 Introduction 27
5.2.2 Linear plane wave 27
5.2.3 Circular plane wave 28
5.2.4 Multiplane wave 28
5.2.5 Antenna 28
5.2.6 Green's tensor inside the object 29
5.2.7 Linear Gaussian 30
5.2.8 Circular Gaussian 30
5.2.9 Circular and linear Gaussian (FFT) 31
5.2.10 Linear Gaussian (para) 31
5.2.11 Circular Gaussian (para) 32
5.2.12 Speckle $k_z > 0$ 32
5.2.13 Speckle 32
5.2.14 Confocal $k_z > 0$ 32
5.2.15 Confocal 33
5.2.16 Arbitrary wave 33

6 Definition of the object **35**
6.1 Introduction . . . 35
6.2 Type of the object . . . 36
6.2.1 Sphere . . . 36
6.2.2 Inhomogeneous sphere . . . 37
6.2.3 Random sphere (length) . . . 37
6.2.4 Random sphere (meshsize) . . . 38
6.2.5 Cube . . . 38
6.2.6 Cuboid (length) . . . 39
6.2.7 Cuboid (meshsize) . . . 39
6.2.8 Inhomogeneous Cuboid (length) . . . 39
6.2.9 Inhomogeneous Cuboid (meshsize) . . . 40
6.2.10 Ellipsoid . . . 40
6.2.11 Multiple spheres . . . 41
6.2.12 Multiple spheres surrounded by random spheres . . . 41
6.2.13 Cylinder . . . 42
6.2.14 Concentric spheres . . . 43
6.2.15 Arbitrary object . . . 43
6.3 Choose the relative permittivity . . . 44
6.4 Choose the discretization . . . 44

7 Possible study with the code **46**
7.1 Introduction . . . 46
7.2 Study in far field . . . 47
7.3 Microscopy . . . 49
7.4 Confocal microscopy . . . 52
7.5 Study in near field . . . 54
7.6 Optical force and torque . . . 55

8 Representation of the results **56**
8.1 Introduction . . . 56
8.2 Digital exits . . . 56
8.3 Graphics . . . 58
8.3.1 Plot epsilon/dipoles . . . 58
8.3.2 Far field and microscopy . . . 58
8.3.2.1 Plot Poynting vector . . . 58
8.3.2.2 Plot microscopy . . . 58
8.3.3 Study of the near field . . . 59
8.3.4 optical force and torque . . . 59

9 Ouput files for matlab, octave, scilab,... **60**
9.1 Introduction . . . 60
9.2 List of all exit files . . . 61
9.3 List of figure names created by ifdda.m . . . 63

10 Some examples **65**
10.1 Introduction 65
10.2 Test1 65
10.3 Test2 70
10.4 Test3 75
10.5 Test4 77

Bibliography **80**

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. 3

2.1 Sketch of the MLB. An incident field impinging on layer k, and the field at layer $k+1$ is the sum of the incident field propagated from layer k to $k+1$ and the field radiated by the elements of layer k. 16

2.2 Sketch of the MLB in reflection. The field propagating in $k_z < 0$ at the layer k is the sum of three components: the field radiated by the dipoles at the layer $k+1$ due to the field propagating in $k_z > 0$ (red); the incident field propagating in $k_z < 0$ and the field radiated by the dipoles at the layer $k+1$ due to this incident field (blue). 17

5.1 Definition of the beam's direction 27

5.2 Definition of the beam's polarization. 28

6.1 Definition of the angles of Euler according to the convention $Z - X - Z$. Scheme taken from Wikipedia 36

6.2 Presentation of the fields for the case of 2 spheres with radii of 500 and 300 nm surrounded by small spheres with a radius of 100 nm, randomly distributed within a cube of 2 μm on each side centered at the origin. Below, we have plotted the relative permittivity in the plane $y = 0$. 42

7.1 Simplified figure of the microscope. The object focus of the objective lens is at the origin of the frame. The axis of the lens is confounded with the z axis and at the side of the positive z. 49

7.2 Different masks in the Fourier domain for the illumination to simulate different types of microscopes. (a) Illumination in a NA cone. (b) Illumination in a NA-NA$_{centralaperture}$ cone. (c) Illumination along the generatrix of an NA aperture cone. (d) Illumination in a pupil of NA aperture centred on the point $(k_x/k_0, k_y/k_0)$. 52

10.1 Test1: configuration taken. 66
10.2 Modulus of the local field in (x, y) plane. 67
10.3 Modulus of the macroscopic field in (x, y) plane. 68
10.4 Modulus of the Poynting vector. 69
10.5 Optical force in the (x, y) plane and in 3D. 69
10.6 Optical torque in the (x, y) plane and in 3D. 70
10.7 Microscopy in transmission: Modulus of the diffracted field in the Fourier plane (left), modulus of the diffracted field in the image plane (middle), and modulus of the total field in the image plane (right). 70
10.8 Test2: configuration taken. 71
10.9 Object in 3D (left) and map of permittivity in the (x, y) plane (right). 71
10.10 Modulus of the incident field in (x, y) plane. 72
10.11 Modulus of the local field in (x, y) plane. 73
10.12 Modulus of the macroscopic field in (x, y) plane. 74
10.13 Modulus of the Poynting vector. 74
10.14 Microscopy in reflection: Modulus of the diffracted field in the Fourier plane (left) and in image plane (right). 75
10.15 Test3: configuration taken. 76
10.16 Microscopy in transmission: incident taken to make the image (left). Modulus of the diffracted field in the image plane (middle), and modulus of the total field in the image plane (right). . 77
10.17 Test4: configuration taken. 78
10.18 Microscope in reflection: incident taken to make the image (left). Modulus of the diffracted field in the image plane (right). 79

Chapter 1

Generality

Contents

1.1 Introduction **1**
1.2 The principle of discrete dipole approximation **2**
1.3 A word about the code **4**
1.4 How to compile the code **4**
1.4.1 CPU 4
1.4.2 GPU 5
1.4.3 Using command line code 6
1.4.3.1 Compile and run the code 6
1.4.3.2 How Command-Line Parsing Works 6
1.4.3.3 Implementation Notes and Important Behaviours 7
1.4.3.4 Defaults & Initialization 7
1.4.3.5 Command-Line Important Behaviours 8
1.4.3.6 Some examples 8
1.5 A word about the authors **8**
1.6 Licence **9**
1.7 How to quote the code **9**

1.1 Introduction

This software computes the diffraction of an electromagnetic wave by a three-dimensional object. This interaction is taken into account rigorously by solving the Maxwell's equations, but can also do with the approximation of Born, Rytov or the BPM. 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, optical forces and torques, diffraction near field and far field as well as microscopy.

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] The DDA has been subsequently widened to objects in presence of a plane substrate or in a multilayer system, see for instance Ref. [3]. These past few years, we endeavoured, on the one hand, to widen the DDA to more complex geometries (grating with or without any default) and, on the other hand, to increase its precision. These improvements tend to make this chapter a little technical, but they are going to be applied in the next chapters. Before studying more in details the last improvements of DDA, though, let us remind first of its principle.

1.2 The principle of discrete dipole approximation

Take an object of arbitrary form and relative permittivity in a homogeneous space that we suppose here being the vacuum. This object is submitted to an incident electromagnetic wave of wavelength λ ($k_0 = 2\pi/\lambda$). 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 incident wave is going to get polarized, and as such, to acquire a dipolar moment, whose value is going to depend on the incident 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 other $N-1$ dipoles :

$$\boldsymbol{E}(\boldsymbol{r}_i) = \boldsymbol{E}_0(\boldsymbol{r}_i) + \sum_{j=1, i\neq j}^{N} \boldsymbol{T}(\boldsymbol{r}_i, \boldsymbol{r}_j)\alpha(\boldsymbol{r}_j)\boldsymbol{E}(\boldsymbol{r}_j). \tag{1.1}$$

$\boldsymbol{E}_0$ is the incident wave, $\boldsymbol{T}$ the linear susceptibility of the field in homogeneous space:

$$\boldsymbol{T}(\boldsymbol{r}_i, \boldsymbol{r}_j) = e^{ik_0 r}\left[\left(3\frac{\boldsymbol{r}\bigotimes\boldsymbol{r}}{r^2} - \boldsymbol{I}\right)\left(\frac{1}{r^3} - \frac{ik_0}{r^2}\right) + \left(\boldsymbol{I} - \frac{\boldsymbol{r}\bigotimes\boldsymbol{r}}{r^2}\right)\frac{k_0^2}{r}\right] \tag{1.2}$$

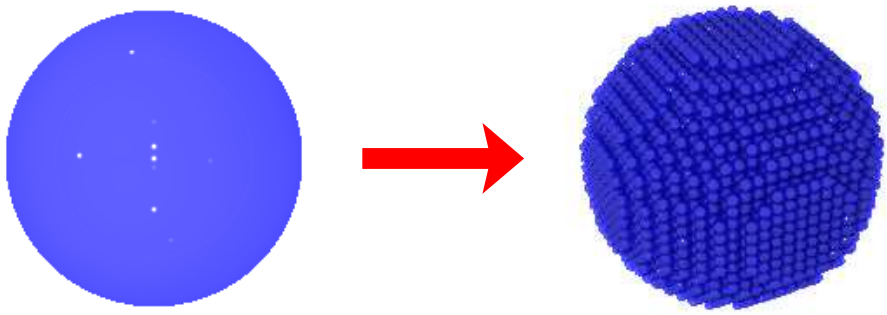


Figure 1.1 : *Principle of the DDA : the object under study (on the left) is discretized in a set of small dipoles (on the right)l.*

with $\boldsymbol{I}$ the unity matrix and $\boldsymbol{r} = \boldsymbol{r}_i - \boldsymbol{r}_j$. 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.[4] 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 :

$$\boldsymbol{E}(\boldsymbol{r}) = \sum_{j=1}^{N} \boldsymbol{T}(\boldsymbol{r}, \boldsymbol{r}_j)\alpha(\boldsymbol{r}_j)\boldsymbol{E}(\boldsymbol{r}_j). \tag{1.3}$$

When the object is in presence of a plane substrate or within a multilayer system, it is just necessary to replace $\boldsymbol{T}$, by the linear susceptibility of the referential system.

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.[5].

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. There are ways to accelerate the resolution of a system of linear equations very important in size as the method of conjugated gradients, but, besides all, values of $N > 10^6$ in homogeneous space are difficult to deal with.

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 optical forces, 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 userguide 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

1.4.1 CPU

The application is based on Qt (version 6) and gfortran To install it you need : qt, qt-devel, gcc-c++ and gfortran, also the code needs the libraries blas and lapack. Notice that there is four versions of the code, the first one is sequential and uses FFTE (Fast Fourier Transform in the east), 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. The last one to use the code in simple precision. Then to compile:

Étape	qmake	Commande
Code par défaut	qmake	make cp cdm/cdm bin/cdm
Code avec FFTW	qmake “CONFIG+=fftw”	
Code avec FFTW et HDF5	qmake “CONFIG+=fftw hdf5”	
Code avec FFTW et HDF5 en simple précision	qmake “CONFIG+=fftw hdf5 simp”	

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

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.

Notice that the code can be installed without the graphical interface. Go to the directory tests, and then write ./comp (or /compfftw or ./compfftwhdf5 depending on the packages installed), then in the four test directories four executables each are created corresponding to four different configuration. It is quite clear that to change the configuration you need to open the file main.f and change the options inside the fortran, which is more tedious than with the interface. graphic, but permit to run the code without Qt.

1.4.2 GPU

The code can also run on the GPU. In this case compilation is done with:

$ qmake “CONFIG+=fftw hdf5 gpu”

$ make

$ cp cdm/cdm bin/cdm

$ cd bin

$ ./cdm

To use the GPU, you need to have an NVIDA card that supports the GPU, and the nvcc compiler with CUDA installed.

You can also use CUDA in single precision, which allows you to gain an additional speed factor of 3 or 4 (and you also save on RAM, obviously). Be careful in this case not to request a tolerance lower than 10^{-4}. In this case, you should use: qmake ”CONFIG+=fftw hdf5 gpu simp”.

Notice that all the iterative method are not done. Mr. M. Gross provided significant assistance in porting part of the code to CUDA.

1.4.3 Using command line code

1.4.3.1 Compile and run the code

This possibility has been done by C. Argentin and M. Yurkin. To get the code do:

$ cd tests

Code used by default	Code with FFTW	Code with FFTW and HDF5	Code with FFTW and HDF5 GPU
$ ./comp	$ ./compfftw	$ ./compfftwhdf5	$ ./compfftwhdf5gpu

$ cd test_command

$ ./ifdda

By default, this runs a simulation of a sphere with a linearly polarized plane wave. To display all available command-line options:

$ ./ifdda -h

1.4.3.2 How Command-Line Parsing Works

The command-line interface supports multiple flags (like ‘-beam‘, ‘-object‘, etc.), each optionally followed by one or more parameters. Internally, the code parses the command line by:

1. Counting the total number of arguments with ‘IARGC()‘.
2. Reading each argument using ‘GETARG()‘.
3. When a known flag (like ‘-beam‘) is detected, the parser (endless ‘IF‘ ‘ELSE IF‘):
   - Advances to the next argument to read its value (e.g., ‘pwavelinear‘, ‘wavelinearmulti‘, etc.).
   - Calls a helper function ‘GET_PARAM_COUNT()‘ (defined at the end of the file) to determine how many additional parameters follow, stopping at the next flag (which begins with ‘-‘ unless it’s a negative number).
   - Reads those values accordingly and assigns them to variables like ‘pp‘, ‘ss‘, etc.

For example, in this command:

$ ./ifdda -beam pwavelinear 0 1

The ‘-beam‘ flag is followed by the beam type ‘pwavelinear‘, and two real values ‘pp=0‘, ‘ss=1‘.

The code can also distinguish between negative numbers and actual flags (e.g. ‘-epsmulti -8.5 0.76‘, ‘-8.5‘ is considered as a negative number and not a flag).

1.4.3.3 Implementation Notes and Important Behaviours

- The command-line options replicate the GUI settings.
- Command-line flags match the GUI names or variable names in ‘main.f‘.
- All ‘PARAMETER‘ and ‘DIMENSION‘ arrays depending on them have been converted to ‘ALLOCATABLE‘, **except**:
- ‘numberobjectmax‘, which remains a compile-time parameter and the arrays that depend on it (e.g. ‘xgmulti‘, ‘ygmulti‘, etc.).
- Arrays depending on ‘nbinc‘ (e.g. ‘E0m‘, ‘thetam‘, etc.).

1.4.3.4 Defaults & Initialization

- Running ‘./ifdda‘ without arguments defaults to a sphere and a linearly polarised plane wave.
- Orientation angles (‘thetaobj‘, ‘phiobj‘, ‘psiobj‘) are initialised to ‘0.0‘, as in the GUI. You can override them with the optional ‘-orient‘ flag.
- Beam direction angles (e.g. ‘thetam‘) are also initialised to ‘0.0‘. You only need ‘-prop‘ if you want to change the direction.
- ‘materiaumulti‘ is set to ‘’xx’‘ by default for all objects (no command-line option).
- ‘numberobjectmax‘ is still a parameter to change at compile time.
- ‘numberobject‘ defaults to ‘1‘ to avoid uninitialised random garbage values that might exceed ‘numberobjectmax‘. This avoids triggering the ‘if‘ condition: ‘if (numberobjet.gt.numberobjetmax) then‘ which will stop the code.
- ‘epsmulti(1)‘ is initialised to a default value to avoid garbage random value like ‘6.7E-310‘. If ‘numberobject $>$ 1‘, you must use ‘-epsmulti‘, otherwise the code will stop with:

  Program finished with problem:

  No relative permittivity given
- ‘epsanimulti‘ is only used when ‘-trope‘ is specified (for anisotropic materials), so it’s not initialised by default.

1.4.3.5 Command-Line Important Behaviours

- If using ‘-wavelinearmulti‘, the ‘-prop‘ flag (to set beam direction) **must be placed after** if you want to override the default propagation direction (i.e. 0 0 0).
- The ‘-epsmulti‘ flag **must be placed after** ‘-object nsphere‘ or ‘-object concentricsphere‘, because ‘numberobject‘ is defined by the object type. Otherwise, the default values will be used for ‘numberobject = 1‘ but the code will stop if ‘numberobject > 1‘, as indicated above.

1.4.3.6 Some examples

Warning: the command must be typed on a single line without line breaks.

$ ./ifdda -lambda 500 -object sphere 500 -methodeit BICGSTAB -epsmulti 1.69 0.0 -tolinit 1.d-4 -polarizability RR -nnnr 16 -ninitest 0

This command is for a sphere of radius a =500 nm with $\varepsilon = 1.69$ illuminated by a wavelength $\lambda = 500$ nm with a mehsize $d = 2a/16$. The polarizability choosen is RR. The iterative method is BICGSTAB with a tolerance of 10^{-4} with an initial guess null.

$ ./ifdda -lambda 500 -object cube 1000 -methodeit BICGSTAB -epsmulti 1.69 0.0 -tolinit 1.d-4 -polarizability RR -nnnr 16 -ninitest 0

As previously but for a cube.

$ ./ifdda -lambda 500 -object cylinder 500 500 -methodeit BICGSTAB -epsmulti 1.69 0.0 -tolinit 1.d-4 -polarizability RR -nnnr 16 -ninitest 0 -prop 35 30 -beam pwavelinear 1 1

As previously but for a cylinder with an incident plane wave defined by $\theta = 35°$ and $\varphi = 30°$.

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.
- 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.
- D. Sentenac develops the convivial interface of the code.

1.6 Licence

Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

You are free to:

- Share - copy and redistribute the material in any medium or format
- Adapt - remix, transform, and build upon the material

The licensor cannot revoke these freedoms as long as you follow the license terms.

- Attribution - You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
- NonCommercial - You may not use the material for commercial purposes.
- ShareAlike - If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

1.7 How to quote the code

- If only the basic functions of the code are used:

  P. C. Chaumet, D. Sentenac, G. Maire, M. Rasedujjaman, T. Zhang and A. Sentenac,
  IFDDA, an easy-to-use code for simulating the field scattered by 3D inhomogeneous objects in a stratified medium: tutorial.
  J. Opt. Soc. Am. A **38**, 1841 (2021).

- If the microscopy is used.

  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).

- If the calculation of the optical forces is used, then:

  P.C. Chaumet, A. Rahmani, A. Sentenac, and G. W. Bryant,
  Efficient computation of optical forces with the coupled dipole method.
  Phys. Rev. E **72**, 046708 (2005).

- If the calculation of optical couples is used:

  P. C. Chaumet and C. Billaudeau,
  Coupled dipole method to compute optical torque: Application to a micropropeller.
  J. Appl. Phys. **101**, 023106 (2007).

- If the rigorous Gaussian beam is used:

  P. C. Chaumet,
  Fully vectorial highly non paraxial beam close to the waist.
  J. Opt. Soc. Am. A **23**, 3197 (2006).

Chapter 2

Approximated method

Contents

2.1 Introduction **11**
2.2 Approximated method **13**
2.2.1 Born 13
2.2.2 Renormalized Born 13
2.2.3 Born at the order 1 13
2.2.4 Rytov 14
2.2.5 Renormalized Rytov 14
2.2.6 Beam propagation method (BPM) 14
2.2.7 Renormalized BPM 15
2.2.8 Classical scalar approximation 15
2.2.9 Scalar Approximation revisited 15
2.2.10 Multilayer Born approximation 16
2.2.10.1 Scalar MLB 16
2.2.10.2 Vectorial MLB 16
2.2.10.3 MLB with reflection 17
2.2.10.4 Summary of the different possibilities with the MLB 18

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 $\varepsilon = 1 + 4\pi\chi$, where χ denotes the linear field susceptibility, we have:

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

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

$$\boldsymbol{\nabla} \times (\boldsymbol{\nabla} \times \boldsymbol{T}(\boldsymbol{r}, \boldsymbol{r}')) - k_0^2 \boldsymbol{T}(\boldsymbol{r}, \boldsymbol{r}') = 4\pi k_0^2 \boldsymbol{I} \delta(\boldsymbol{r} - \boldsymbol{r}'), \tag{2.5}$$

and the solution of Eq. (2.4) reads:

$$\boldsymbol{E}^{\mathrm{m}}(\boldsymbol{r}) = \boldsymbol{E}^0(\boldsymbol{r}) + \int_\Omega \boldsymbol{T}(\boldsymbol{r}, \boldsymbol{r}') \chi(\boldsymbol{r}') \boldsymbol{E}^{\mathrm{m}}(\boldsymbol{r}') \mathrm{d}\boldsymbol{r}', \tag{2.6}$$

where $\boldsymbol{E}^0$ is the incident field and Ω the support of the object under study. When we solve Eq. (2.4) the field $\boldsymbol{E}^{\mathrm{m}}$ corresponds to 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:

$$\boldsymbol{E}^{\mathrm{m}}(\boldsymbol{r}_i) = \boldsymbol{E}^0(\boldsymbol{r}_i) + \sum_{j=1}^{N} \int_{V_j} \boldsymbol{T}(\boldsymbol{r}_i, \boldsymbol{r}') \chi(\boldsymbol{r}') \boldsymbol{E}^{\mathrm{m}}(\boldsymbol{r}') \mathrm{d}\boldsymbol{r}', \tag{2.7}$$

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

$$\boldsymbol{E}^{\mathrm{m}}(\boldsymbol{r}_i) = \boldsymbol{E}^0(\boldsymbol{r}_i) + \sum_{j=1}^{N} \boldsymbol{T}(\boldsymbol{r}_i, \boldsymbol{r}_j) \chi(\boldsymbol{r}_j) \boldsymbol{E}^{\mathrm{m}}(\boldsymbol{r}_j) d^3. \tag{2.8}$$

Using, in first approximation (the radiative reaction term neglected) $\int_{V_i} \boldsymbol{T}(\boldsymbol{r}_i, \boldsymbol{r}') \mathrm{d}\boldsymbol{r}' = -4\pi/3$[6]), we get:

$$\boldsymbol{E}^{\mathrm{m}}(\boldsymbol{r}_i) = \boldsymbol{E}^0(\boldsymbol{r}_i) + \sum_{j=1, i \neq j}^{N} \boldsymbol{T}(\boldsymbol{r}_i, \boldsymbol{r}_j) \chi(\boldsymbol{r}_j) d^3 \boldsymbol{E}^{\mathrm{m}}(\boldsymbol{r}_j) - \frac{4\pi}{3} \chi(\boldsymbol{r}_i) \boldsymbol{E}^{\mathrm{m}}(\boldsymbol{r}_i), \tag{2.9}$$

then we can write

$$\boldsymbol{E}(\boldsymbol{r}_i) = \boldsymbol{E}^0(\boldsymbol{r}_i) + \sum_{j=1, i \neq j}^{N} \boldsymbol{T}(\boldsymbol{r}_i, \boldsymbol{r}_j) \alpha_{\mathrm{CM}}(\boldsymbol{r}_j) \boldsymbol{E}(\boldsymbol{r}_j) \tag{2.10}$$

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

$$\alpha_{\mathrm{CM}}(\boldsymbol{r}_j) = \frac{3}{4\pi} d^3 \frac{\varepsilon(\boldsymbol{r}_i) - 1}{\varepsilon(\boldsymbol{r}_i) + 2}. \tag{2.12}$$

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

$$\boldsymbol{E} = \boldsymbol{E}^0 + \boldsymbol{A}\boldsymbol{D}_\alpha \boldsymbol{E}, \tag{2.13}$$

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

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

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 incident field for each element of discretization:

$$\boldsymbol{E}^{\mathrm{m}}(\boldsymbol{r}_i) = \boldsymbol{E}^0(\boldsymbol{r}_i), \tag{2.15}$$

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 incident field :

$$\boldsymbol{E}(\boldsymbol{r}_i) = \boldsymbol{E}^0(\boldsymbol{r}_i). \tag{2.16}$$

In that case the macroscopic field reads:

$$\boldsymbol{E}^{\mathrm{m}}(\boldsymbol{r}_i) = \frac{3}{\varepsilon(\boldsymbol{r}_i) + 2}\boldsymbol{E}^0(\boldsymbol{r}_i). \tag{2.17}$$

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:

$$\boldsymbol{E}(\boldsymbol{r}_i) = \boldsymbol{E}^0(\boldsymbol{r}_i) + \sum_{j=1, i\neq j}^{N} \boldsymbol{T}(\boldsymbol{r}_i, \boldsymbol{r}_j)\alpha(\boldsymbol{r}_j)\boldsymbol{E}^0(\boldsymbol{r}_j). \tag{2.18}$$

In that case we take into account the simple scattering.

2.2.4 Rytov

The Rytov approximation consist to take into account the phase variation inside the object:

$$E_\beta^{\mathrm{m}}(\boldsymbol{r}_i) \quad = \quad E_\beta^0(\boldsymbol{r}_i)e^{E_\beta^{\mathrm{d}}(\boldsymbol{r}_i)/E_\beta^0(\boldsymbol{r}_i)}, \tag{2.19}$$

with $\beta = x, y, z$. Notice that when a component of the incident field is null, then $E_\beta^{\mathrm{m}} = 0$. This approximation permits to deal with large object compare to the wavelength of illumination, but always with low contrast of permittivity. The diffracted field reads:

$$\boldsymbol{E}^{\mathrm{d}}(\boldsymbol{r}_i) \quad = \quad \sum_{j=1}^{N} \boldsymbol{T}(\boldsymbol{r}_i, \boldsymbol{r}_j)\chi(\boldsymbol{r}_j)\boldsymbol{E}^0(\boldsymbol{r}_j), \tag{2.20}$$

2.2.5 Renormalized Rytov

The renormalized Rytov approximation deals with the local field:

$$E_\beta(\boldsymbol{r}_i) \quad = \quad E_\beta^0(\boldsymbol{r}_i)e^{E_\beta^{\mathrm{d}}(\boldsymbol{r}_i)/E_\beta^0(\boldsymbol{r}_i)}, \tag{2.21}$$

and the diffracted field reads:

$$\boldsymbol{E}^{\mathrm{d}}(\boldsymbol{r}_i) \quad = \quad \sum_{j=1, i\neq j}^{N} \boldsymbol{T}(\boldsymbol{r}_i, \boldsymbol{r}_j)\alpha(\boldsymbol{r}_j)\boldsymbol{E}^0(\boldsymbol{r}_j). \tag{2.22}$$

2.2.6 Beam propagation method (BPM)

BPM is a class of algorithms designed for calculating the optical field distribution in space for very large object compare to the wavelength of illumination. BPM allows to obtain the electromagnetic field via alternating evaluation of diffraction and refraction steps handled in the Fourier and space domains It is important to note that BPM ignores reflections, for more details see Ref. 7. In final the field reads

$$\boldsymbol{E}^{\mathrm{m}}(x, y, z + d) = e^{ik_0 n(x,y,z+d)d}\mathcal{F}^{-1}\left[\mathcal{F}[\boldsymbol{E}^{\mathrm{m}}(x, y, z)]e^{-i(k_0-k_z)d}\right], \tag{2.23}$$

where the field at the position $(x, y, z+d)$ is computed with the permittivity at the same position and the field at the previous plane z. It is clear with this relation that the field is propagated only in the direction of the positive z. Note that the size of the FFT is given by the drop down menu and to avoid angle of incidence too high. Notice that the diffracted field is computed like the other methods which is more precise that the Kirchhoff's equation.

2.2.7 Renormalized BPM

We can do the same but with the local field:

$$\boldsymbol{E}(x,y,z+d) = e^{ik_0 n(x,y,z+d)d}\mathcal{F}^{-1}\left[\mathcal{F}[\boldsymbol{E}(x,y,z)]e^{-i(k_0-k_z)d}\right]. \tag{2.24}$$

2.2.8 Classical scalar approximation

The scalar approximation taken usually consist to approximate the dyadic Green function by the scalar Green function $g(\boldsymbol{r},\boldsymbol{r}') = \frac{e^{ik|\boldsymbol{r}-\boldsymbol{r}'|}}{|\boldsymbol{r}-\boldsymbol{r}'|}$, *i.e.* $\boldsymbol{T}(\boldsymbol{r},\boldsymbol{r}') \approx g(\boldsymbol{r},\boldsymbol{r}')\boldsymbol{I}$. This approximation is based on the assumption that the gradient of the relative permittivity is weak.

2.2.9 Scalar Approximation revisited

We only consider configurations where the reference field in Ω can be written as $\boldsymbol{E}^0(\boldsymbol{r}) = E^0(\boldsymbol{r})\boldsymbol{u}$ with $\boldsymbol{u}$ a complex vector such that $\boldsymbol{u}\cdot\boldsymbol{u}^* = 1$ where $*$ stands for the complex conjugate and $E^0(\boldsymbol{r})$ a complex function. This is the case if the reference field in Ω is a plane wave or a sum of plane waves with the same polarization.

In our approach, we assume that the field inside Ω is directed along $\boldsymbol{u}$ so that $\boldsymbol{E}(\boldsymbol{r}) \approx E_u(\boldsymbol{r})\boldsymbol{u}$ where E_u is a complex function. In this case, taking the scalar product of Eq. (2.6) with $\boldsymbol{u}^*$ yields an integral scalar equation for E_u,

$$\begin{aligned}\boldsymbol{u}^*\cdot\boldsymbol{E}(\boldsymbol{r}) &= \boldsymbol{u}^*\cdot\boldsymbol{E}^0(\boldsymbol{r})\boldsymbol{u}^*\cdot\int_\Omega \boldsymbol{T}(\boldsymbol{r},\boldsymbol{r}')\chi(\boldsymbol{r}')E_u(\boldsymbol{r}')\boldsymbol{u}\mathrm{d}\boldsymbol{r}' \\ E_u(\boldsymbol{r}) &= E^0(\boldsymbol{r}) + \int_\Omega \left[\boldsymbol{u}^*\cdot\boldsymbol{T}(\boldsymbol{r},\boldsymbol{r}')\boldsymbol{u}\right]\chi(\boldsymbol{r}')E_u(\boldsymbol{r}')\mathrm{d}\boldsymbol{r}',\end{aligned} \tag{2.25}$$

where the green tensor has been replaced by the scalar function, $g_u(\boldsymbol{r},\boldsymbol{r}') = \boldsymbol{u}^*\cdot\boldsymbol{T}(\boldsymbol{r},\boldsymbol{r}')\boldsymbol{u}$. It is worth noting that g_u is different from g as it depends on $\boldsymbol{u}$ and contains near field term in $1/R^3$ and $1/R^2$ with $R = |\boldsymbol{r}-\boldsymbol{r}'|$. Then the field inside Ω is computed through:

$$E_u(\boldsymbol{r}_i) = E^0(\boldsymbol{r}_i) + \sum_{j=1}^{N} g_u(\boldsymbol{r}_i,\boldsymbol{r}_j)\chi(\boldsymbol{r}_j)E_u(\boldsymbol{r}_j)d^3, \tag{2.26}$$

with $i = 1,\cdots,N$. It is obvious that the size of the vector and the matrix are decreased by a factor 3. Then, when we solve iteratively the linear system, as it needs only to treat one component, we guess that the numerical computation will be faster by a factor 3 at least. Notice that once the near field is obtained, we transform it in vectorial form with $\boldsymbol{E}(\boldsymbol{r}_i) = E_u(\boldsymbol{r}_i)\boldsymbol{u}$ and we use it to obtain the far field.

2.2.10 Multilayer Born approximation

2.2.10.1 Scalar MLB

The Multilayer Born approximation (MLB) has been introduced recently in the framework of the scalar approximation[8]. Here we present the MLB in an intuitive way: the object is divided into layers of thickness d along the optical axis (here z axis). Layer k receives an incident field $E^k(\boldsymbol{\rho})$. The elements of the layer k, under the action of the incident field, polarize with the dipole moment αE^k and therefore radiate a field to layer $k+1$. The total field at layer $k+1$ is thus composed of the incident field at layer k that has propagated to $k+1$, and the field diffracted by the layer k, see Fig. 2.1:

$$\begin{aligned} E^{k+1}(\boldsymbol{\rho}_i, z_{k+1}) &= \mathrm{FFT}_{2D}^{-1}\left[e^{ik_z d}\mathrm{FFT}_{2D}[E^k(z_k)]\right](\boldsymbol{\rho}_i) \\ &+ \sum_j g(\boldsymbol{\rho}_i, z_{k+1}, \boldsymbol{\rho}_j, z_k)\alpha(\boldsymbol{\rho}_j, z_k)E^k(\boldsymbol{\rho}_j, z_k) \end{aligned} \quad (2.27)$$

Note that we use the renormalized Born approximation to increase the precision.

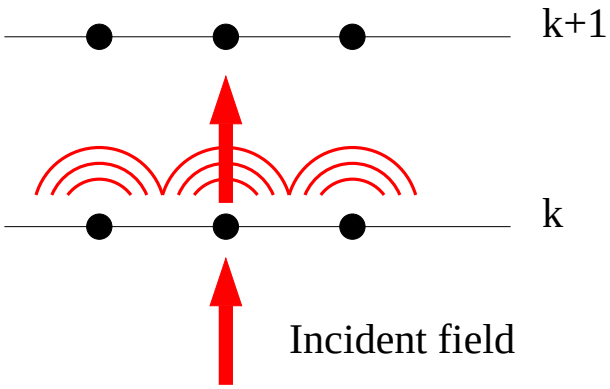


Figure 2.1 : *Sketch of the MLB. An incident field impinging on layer k, and the field at layer $k+1$ is the sum of the incident field propagated from layer k to $k+1$ and the field radiated by the elements of layer k.*

2.2.10.2 Vectorial MLB

Contrarily to the BPM, it is very easy for the MLB to switch to vectorial. It is the same principle but keeping the Green's tensor and the vector electric

field. In this case, the MLB reads:

$$\begin{aligned} \boldsymbol{E}^{k+1}(\boldsymbol{\rho}_i, z_{k+1}) &= \mathrm{FFT}_{2D}^{-1}\left[e^{ik_z d}\mathrm{FFT}_{2D}[\boldsymbol{E}^k(z_k)]\right](\boldsymbol{\rho}_i) \\ &+ \sum_j \boldsymbol{T}(\boldsymbol{\rho}_i, z_{k+1}, \boldsymbol{\rho}_j, z_k)\alpha(\boldsymbol{\rho}_j, z_k)\boldsymbol{E}^k(\boldsymbol{\rho}_j, z_k). \quad (2.28) \end{aligned}$$

It should be noted that the Green's tensor contains both evanescent and propagating waves[9]. Therefore, the propagation of evanescent waves from one layer to another is taken into account. However, the calculation time is multiplied by three because the three components of the electric field are calculated.

2.2.10.3 MLB with reflection

In their article, Chen *et al.* proposed to compute the electric field in reflection when modeling backward scattering with MLB[8]. We are going to use the same principle to evaluate the electric field propagating in the negative z direction inside the object. The principle is as follows, see Fig. 2.2: The

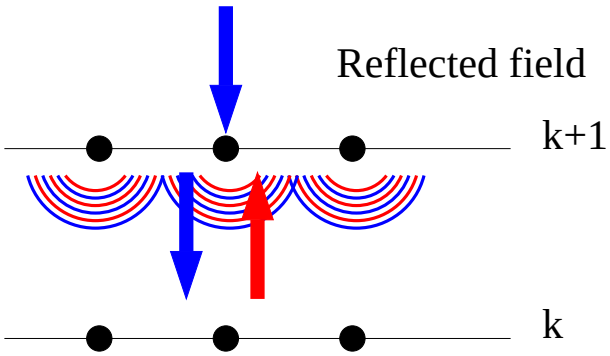


Figure 2.2 : *Sketch of the MLB in reflection. The field propagating in $k_z < 0$ at the layer k is the sum of three components: the field radiated by the dipoles at the layer $k+1$ due to the field propagating in $k_z > 0$ (red); the incident field propagating in $k_z < 0$ and the field radiated by the dipoles at the layer $k+1$ due to this incident field (blue).*

field reflected at layer k is composed of the sum of the fields radiated by the dipoles of layer $k+1$ due to field propagating towards the positive z direction, the incident field propagated from the layer $k+1$ to k and the field radiated by the dipoles of layer $k+1$ due to this incident field. Then,

the reflected field at layer k is written as:

$$\begin{aligned} \boldsymbol{E}^k_{\text{ref}}(\boldsymbol{\rho}_i, z_k) &= \text{FFT}^{-1}_{2D}\left[e^{ik_z d}\text{FFT}_{2D}[\boldsymbol{E}^k_{\text{ref}}(z_{k+1})]\right](\boldsymbol{\rho}_i) \\ &+ \sum_j \boldsymbol{T}(\boldsymbol{\rho}_i, z_k, \boldsymbol{\rho}_j, z_{k+1})\alpha(\boldsymbol{\rho}_j, z_{k+1}) \\ &\times \left[\boldsymbol{E}^k_{\text{ref}}(\boldsymbol{\rho}_j, z_{k+1}) + \boldsymbol{E}^k(\boldsymbol{\rho}_j, z_{k+1})\right]. \end{aligned} \tag{2.29}$$

The total field in the object is then the sum of the field propagating in the positive z and the field propagating in the negative z: $\boldsymbol{E}_{\text{tot}} = \boldsymbol{E} + \boldsymbol{E}_{\text{ref}}$. Once the field inside the object has been obtained, the diffracted field is computed like the other methods which is more precise that the Kirchhoff's equation. It should be noted that Eq. (2.29) can of course be be written in the scalar case, just replace $\boldsymbol{T}$ by g.

2.2.10.4 Summary of the different possibilities with the MLB

Method	Green's function used	Reflection with Eq. (2.29)
MLB scalar: exp(ikr)/r	$\frac{e^{ikr}}{r}$	no
MLB scalar: exp(ikr)/r with reflection	$\frac{e^{ikr}}{r}$	yes
MLB scalar: u.Gu	$\boldsymbol{u}^*.\boldsymbol{T}\boldsymbol{u}$	no
MLB scalar: u.Gu with reflection	$\boldsymbol{u}^*.\boldsymbol{T}\boldsymbol{u}$	yes
MLB vectorial	$\boldsymbol{T}$	no
MLB vectorial with reflection	$\boldsymbol{T}$	yes

Table 2.1 : *The different possibilities with the multilayer Born approximation. The first column corresponds to the names in the menu and the following columns show the Green's function used and whether or not there is reflection.*

Chapter 3

Numerical details

Contents

3.1 Polarizability 19
3.2 Correction to the tensor of susceptibility 20
3.3 Filtered Green's function 21
3.4 Solve the system of linear equation 21
3.5 Change of the initial guess 22
3.6 Preconditioning the system of linear equations 23
3.7 The default options and how to change them 23

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]:

$$\alpha_{\mathrm{CM}} = \frac{3}{4\pi}\frac{\varepsilon-1}{\varepsilon+2}d^3 = \frac{\varepsilon-1}{\varepsilon+2}a^3, \tag{3.1}$$

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 [4]:

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

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[10],

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

by Lakhtakia[11]:

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

and Draine and Goodman[12]

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

with:

- $b_1 = -1.8915316529870796511106114030718259$
- $b_2 = -0.16484691508771947306079362778185226$
- $b_3 = 1.7700004019321371908592738404451742$

and $S = 1/5$ (precise values of b have been given by M. Yurkin and C. Argentin). We also define α_{LS} with $S = \sum_j |a_j e_j^{(0)}|^2$ as defined by Draine and Goodman[12].

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 precised 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.[13]

3.2 Correction to the tensor of susceptibility

The tensor of susceptibility (or dyadic Green function) of the field connects the dipole to the position $\boldsymbol{r}_j$ to the field radiated at the position $\boldsymbol{r}_i$ by the relation : $\boldsymbol{E}(\boldsymbol{r}_i) = \boldsymbol{T}(\boldsymbol{r}_i, \boldsymbol{r}_j)\boldsymbol{p}(\boldsymbol{r}_i)$. But inside the DDA, considering the fact that the dipoles are associated with a certain volume, the following integration should be written[5]:

$$\boldsymbol{E}(\boldsymbol{r}_i) = \int_{V_j} \boldsymbol{T}(\boldsymbol{r}_i, \boldsymbol{r})\boldsymbol{p}(\boldsymbol{r})\mathrm{d}\boldsymbol{r} \approx \left[\int_{V_j} \boldsymbol{T}(\boldsymbol{r}_i, \boldsymbol{r})\mathrm{d}\boldsymbol{r}\right]\boldsymbol{p}(\boldsymbol{r}_j), \tag{3.6}$$

supposing the meshsize small enough to be able to consider the field as being uniform in it. So, the tensor must be integrated upon the meshsize V_j. This integration is not analytic (it has to be done numerically and this takes time) and, in fact, it only serves for the dipoles which are the nearest to the observation, after that, the integration does not bring any more precision. So, in the code, we propose the possibility to integrate upon the nearest mesh sizes:

$$\int_{V_j} \boldsymbol{T}(\boldsymbol{r}_i, \boldsymbol{r})\mathrm{d}\boldsymbol{r} \qquad \text{if} \qquad \frac{\|\boldsymbol{r}_i - \boldsymbol{r}_j\|}{d} \leq n \tag{3.7}$$

$$\boldsymbol{T}(\boldsymbol{r}_i, \boldsymbol{r}_j) \qquad \text{if} \qquad \frac{\|\boldsymbol{r}_i - \boldsymbol{r}_j\|}{d} > n. \tag{3.8}$$

n may take the value entire 0 (by default) until 5.

3.3 Filtered Green's function

For the polarizability one can choose the filtered Green's function with the FG option in the polarizability menu. The filtered Green's function function was introduced by Martin *et al.*[14,15] and revisited by Yurkin[16]. The polarizability is calculated by taking into account the modification of the Green function[16].

3.4 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:

$$\boldsymbol{E} = \boldsymbol{E}_0 + \boldsymbol{A} D_\alpha \boldsymbol{E}, \tag{3.9}$$

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 [17]. The code stops when the residue,

$$r = \frac{\|\boldsymbol{E} - \boldsymbol{A} D_\alpha \boldsymbol{E} - \boldsymbol{E}_0\|}{\|\boldsymbol{E}_0\|}, \tag{3.10}$$

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. 18
- GPBICG2 : Ref. 18
- GPBICGsafe : Ref. 19
- GPBICGAR1 : Ref. 18
- GPBICGAR2 : Ref. 18
- QMRCLA : Ref. 20
- TFQMR : Ref. 20
- CG : Ref. 20
- BICGSTAB : Ref. 20
- QMRBICGSTAB1 : Ref. 21
- QMRBICGSTAB2 : Ref. 21
- GPBICOR : Ref. 22
- CORS : Ref. 23
- BiCGstar-plus Ref. 24
- IDR(s) Ref. 25
- BICGSTABL Ref. 26
- GPBICGSTABL Ref. 27

3.5 Change of the initial guess

When the system of linear equations is solved iteratively, we have the possibility to choose the starting point, *i.e.* the initial field $\boldsymbol{E}_i$ used to start the iterative method. The closer the solution chosen at the beginning will be close to the "good solution", the more the number of iterations will be reduced. We therefore propose the possibility to choose as initial guess for the field:

- $\boldsymbol{E}_i = \mathbf{0}$: null field at the beginning.
- $\boldsymbol{E}_i = \boldsymbol{E}_0$: Born approximation.
- Rytov approximation.
- Field obtained by the BPM.

- Use of the scalar approximation $\boldsymbol{u}.\boldsymbol{G}\boldsymbol{u}$.[28] Note that the scalar approximation is also solved iteratively but for $r = 0.01$. An additional precision would not be of interest, because we just want a correct starting point.

3.6 Preconditioning the system of linear equations

Another solution is to precondition the matrix to be inverted on the left to make the iterative method faster. That is to say instead of solving $(\boldsymbol{I} - \boldsymbol{A}\boldsymbol{D}_\alpha)\boldsymbol{E} = \boldsymbol{E}_0$, we must then solve $\boldsymbol{P}^{-1}(\boldsymbol{I} - \boldsymbol{A}\boldsymbol{D}_\alpha)\boldsymbol{E} = \boldsymbol{P}^{-1}\boldsymbol{E}_0$ where $\boldsymbol{P}$ is a matrix close to $(\boldsymbol{I} - \boldsymbol{A}\boldsymbol{D}_\alpha)$ and whose inverse can be easily computed. For the preconditioner we have chosen a matrix of Chan[29] on the two dimensions of the space x and y[30]. This preconditioning is particularly efficient when the object under study is homogeneous or weakly inhomogeneous and when it has a small thickness in z compared to its dimensions in x and y. The preconditioning can also be done on the right hand side, *i.e.* we have to find $\boldsymbol{X}$ such that $(\boldsymbol{I} - \boldsymbol{A}\boldsymbol{D}_\alpha)\boldsymbol{P}^{-1}\boldsymbol{X} = \boldsymbol{E}_0$, then deduce the field with $\boldsymbol{E} = \boldsymbol{P}^{-1}\boldsymbol{X}$.

Note that this preconditioning is also implemented for the scalar approximation.

3.7 The default options and how to change them

The default options chosen are:

- The polarizability: α_{RR}.
- Iterative method: GPBICG1.
- Tolerance of the iterative method: 10^{-4}.
- Maximum number of iterations for the iterative method: 1000
- Initial guess for the iterative method: Born approximation.
- No preconditioning.
- Integration of the Green's function: no integration.

All these options can be changed. To do this you must click on "Advanced interface", and there appears at the bottom a whole section section called "Numerical parameters" where all the parameters related to the iterative method and the polarizability can be changed.

It is of course obvious that if one has chosen the Born, Rytov or BPM approximation then all the choices related to the iterative method have no influence.

Chapter 4

Managing of the configurations

Contents

4.1 Introduction 24
4.2 Creation and saving of a new configuration 24
4.3 Managing of the configurations 24

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 config-

urations. 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 **26**
5.2 Beam **27**
5.2.1 Introduction 27
5.2.2 Linear plane wave 27
5.2.3 Circular plane wave 28
5.2.4 Multiplane wave 28
5.2.5 Antenna 28
5.2.6 Green's tensor inside the object 29
5.2.7 Linear Gaussian 30
5.2.8 Circular Gaussian 30
5.2.9 Circular and linear Gaussian (FFT) 31
5.2.10 Linear Gaussian (para) 31
5.2.11 Circular Gaussian (para) 32
5.2.12 Speckle $k_z > 0$ 32
5.2.13 Speckle 32
5.2.14 Confocal $k_z > 0$ 32
5.2.15 Confocal 33
5.2.16 Arbitrary wave 33

5.1 Introduction

In the section properties of the illumination, the field *Wavelength* enables us to enter the using wavelength. 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.

5.2 Beam

5.2.1 Introduction

There are many beams predefined, their propagation direction is always defined in the same way, with two angles θ and φ when possible. 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,

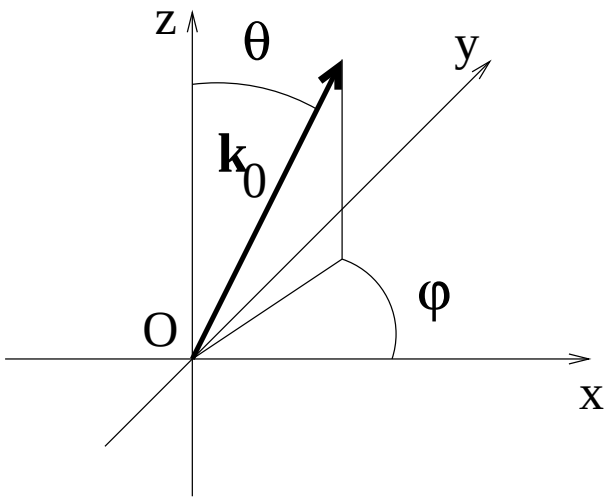


Figure 5.1 : *Definition of the beam's direction*

we use the plane (x, y) as referential surface. Then, we can determine a polarization TM (p) and TE (s) with the presence of a surface, see Fig. 5.2. The frame (x, y, z) is used as an absolute referential.

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, pola=1 en TM and pola=0 in TE. 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 $E^2_{\mathrm{TE}} = (1 - \mathrm{pola}^2)E^2$.

Note that the phase is always taken null at the origin of the frame:

$$\boldsymbol{E}(\boldsymbol{r}) = \boldsymbol{E}_0 e^{i\boldsymbol{k}.\boldsymbol{r}}, \tag{5.4}$$

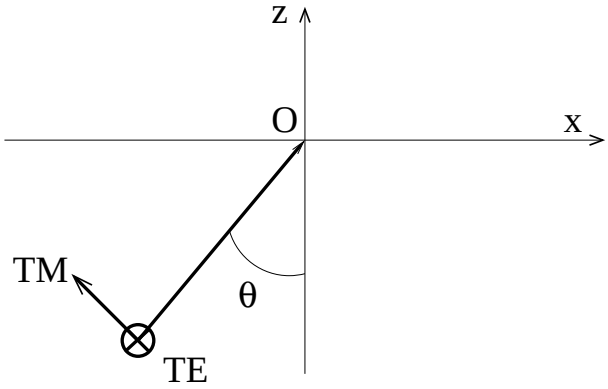


Figure 5.2 : *Definition of the beam's polarization.*

with Irradiance $= P_0/S$ where $S = \pi w_0^2$ is the surface of the beam and $E_0 = \sqrt{2\text{Irradiance}/c/\varepsilon_0}$.

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.

$$\boldsymbol{E}(\boldsymbol{r}) = \boldsymbol{E}_0 e^{i\boldsymbol{k}.\boldsymbol{r}}, \tag{5.5}$$

with Irradiance $= P_0/S$ where $S = \pi w_0^2$ is the surface of the beam and $E_0 = \sqrt{2\text{Irradiance}/c/\varepsilon_0}$.

5.2.4 Multiplane wave

Multiplane 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 Antenna

The incident beam can be a dipole where the user defines the position and orientation. Notice that the antenna can be inside or outside the object. The magnitude is chosen such that the power radiated by the dipole is equal to P_0:

$$P_0 = \frac{1}{4\pi\varepsilon_0}\frac{k^4 c}{3}\|\boldsymbol{p}\|^2. \tag{5.6}$$

5.2.6 Green's tensor inside the object

This option allows to calculate the Green's tensor, $\boldsymbol{G}'\boldsymbol{r}, \boldsymbol{r}_0)$, inside the object for a $\boldsymbol{r}_0$ position of the source. Hence, this option consists of choosing the incident field as well as choosing the calculation we want to make. All the other options are then deactivated, because it is necessary to calculate the field for three orientations of a dipole located in $\boldsymbol{r}_0$ and calculate the radiated field for each, and then deduce the Green's tensor in the object.

The principle is as follows, we have

$$\boldsymbol{E}(\boldsymbol{r}) = \boldsymbol{E}_0(\boldsymbol{r}) + \int \boldsymbol{T}(\boldsymbol{r}, \boldsymbol{r}')\chi(\boldsymbol{r}')\boldsymbol{E}(\boldsymbol{r}')\mathrm{d}\boldsymbol{r}'. \tag{5.7}$$

The source is a dipole located at $\boldsymbol{r}_0$, i.e.

$$\boldsymbol{E}(\boldsymbol{r}) = \boldsymbol{T}(\boldsymbol{r}, \boldsymbol{r}_0)\boldsymbol{p}(\boldsymbol{r}_0) + \int \boldsymbol{T}(\boldsymbol{r}, \boldsymbol{r}')\chi(\boldsymbol{r}')\boldsymbol{E}(\boldsymbol{r}')\mathrm{d}\boldsymbol{r}'. \tag{5.8}$$

We can also say that the field in $\boldsymbol{r}$ is written as $\boldsymbol{E}(\boldsymbol{r}) = \boldsymbol{G}(\boldsymbol{r}, \boldsymbol{r}_0)\boldsymbol{p}(\boldsymbol{r}_0)$ with $\boldsymbol{G}$ the Green's function of the vacuum plus the object, i.e:

$$\boldsymbol{G}(\boldsymbol{r}, \boldsymbol{r}_0)\boldsymbol{p}(\boldsymbol{r}_0) = \boldsymbol{T}(\boldsymbol{r}, \boldsymbol{r}_0)\boldsymbol{p}(\boldsymbol{r}_0) + \int \boldsymbol{T}(\boldsymbol{r}, \boldsymbol{r}')\chi(\boldsymbol{r}')\boldsymbol{G}(\boldsymbol{r}', \boldsymbol{r}_0)\boldsymbol{p}(\boldsymbol{r}_0)\mathrm{d}\boldsymbol{r}'. \tag{5.9}$$

We can therefore deduce that

$$\boldsymbol{G}(\boldsymbol{r}, \boldsymbol{r}_0)\boldsymbol{p}(\boldsymbol{r}_0) = \boldsymbol{T}(\boldsymbol{r}, \boldsymbol{r}_0)\boldsymbol{p}(\boldsymbol{r}_0) + \sum_{i=1}^{N} \boldsymbol{T}(\boldsymbol{r}, \boldsymbol{r}_i)\alpha(\boldsymbol{r}_i)\boldsymbol{G}(\boldsymbol{r}_i, \boldsymbol{r}_0)\boldsymbol{p}(\boldsymbol{r}_0), \tag{5.10}$$

with

$$\boldsymbol{G}(\boldsymbol{r}_i, \boldsymbol{r}_0)\boldsymbol{p}(\boldsymbol{r}_0) = \boldsymbol{T}(\boldsymbol{r}_i, \boldsymbol{r}_0)\boldsymbol{p}(\boldsymbol{r}_0) + \sum_{j=1, i\neq j}^{N} \boldsymbol{T}(\boldsymbol{r}_i, \boldsymbol{r}_j)\alpha(\boldsymbol{r}_j)\boldsymbol{G}(\boldsymbol{r}_j, \boldsymbol{r}_0)\boldsymbol{p}(\boldsymbol{r}_0). \tag{5.11}$$

We therefore have to solve the system of linear equations Eq. (5.11) and then calculate Eq. (5.10) if we want to calculate the Green's function outside the object. In the case where $\boldsymbol{r}_0 \in \Omega$ then it is enough to solve only Eq. (5.11). But, $\boldsymbol{T}(\boldsymbol{r}_i, \boldsymbol{r}_0)$ is not defined when $\boldsymbol{r}_i = \boldsymbol{r}_0$. To solve the problem we use the fact that $\int_{V_i} \boldsymbol{T}(\boldsymbol{r}_i, \boldsymbol{r}_0)\mathrm{d}V \approx \left(-\frac{4}{3} + \frac{2}{3}ik_0^3 d^3\right)\boldsymbol{I}$, that is $\boldsymbol{T}(\boldsymbol{r}_i, \boldsymbol{r}_0) \approx \left(-\frac{4}{3} + \frac{2}{3}ik_0^3 d^3\right)/d^3\boldsymbol{I}$. In the end, we have therefore calculated the Green's tensor $\boldsymbol{G}(\boldsymbol{r}_i, \boldsymbol{r}_0)$ at the position of all the discretization elements. Note that in the graphical interface, we plot only the field due to the dipole directed along the z axis (which allows us to see the G_{xz} components, G_{yz} and G_{zz} of the tensor), but the 9 elements of the Green's tensor are all in the HDF5 file under the name Greentensor or greentensor.mat depending on the option chosen.

5.2.7 Linear Gaussian

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

The three following lines help to fix the position of the centre 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. The definition of the waist, for a beam propagating along the z axis is :[31]

$$E(x, y, 0) = E_0 e^{-\rho^2/(2w_0^2)}, \tag{5.12}$$

with $\rho = \sqrt{x^2+y^2}$. From this definition of the beam at $z = 0$, for a beam polarized along the x axis we get :[32]

$$E_x = E_0 \int_0^{k_0} w_0^2 \exp\left(-\frac{w_0^2(k_0^2-k_z^2)}{2}\right) \exp(ik_z z) J_0\left(\rho\sqrt{k_0^2-k_z^2}\right) k_z \mathrm{d}k_z \tag{5.13}$$

$$\begin{aligned} E_z &= -iE_0 \frac{x}{\rho} \int_0^{k_0} w_0^2 \exp\left(-\frac{w_0^2(k_0^2-k_z^2)}{2}\right) \exp(ik_z z) \\ &\times J_1\left(\rho\sqrt{k_0^2-k_z^2}\right) \sqrt{k_0^2-k_z^2}\mathrm{d}k_z, \end{aligned} \tag{5.14}$$

with J_1 and J_0 the Bessel's function. The irradiance is computed at the center of the Gaussian beam and the relationship between the power and the magnitude E_0 is:

$$P_0 = \frac{\pi w_0^2}{4} c\varepsilon_0 E_0^2 \left(1 + \frac{(k_0 w_0)^2 - 1}{k_0 w_0} \frac{\sqrt{\pi}}{2} \mathrm{Im}[w(k_0 w_0)]\right) \tag{5.15}$$

$$\text{Irradiance} = \frac{E_0^2}{4} c\varepsilon_0 \left(1 + \frac{(k_0 w_0)^2 - 1}{k_0 w_0} \frac{\sqrt{\pi}}{2} \mathrm{Im}\left[w(k_0 w_0/\sqrt{2})\right]\right) \tag{5.16}$$

where $w()$ denotes the Faddeeva's function. If we suppose $w() \approx 0$, we obtain $P_0 = \pi w_0^2 \mathrm{Irradiance}$ and we find the relation given for a plane wave.

5.2.8 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.

5.2.9 Circular and linear Gaussian (FFT)

Circular and linear Gaussian (FFT) is a Gaussian wave based on the previous computation for the Circular and linear Gaussian, respectively. In this case, the incident wave is computed at the bottom of the object and then the beam is propaged with FFT as for the beam propagation method. This computation is quicker than the rigourous one. However, one needs to choose the number of points for the FFT enough large to not truncate the Gaussian beam and avoid periodicity problem.

5.2.10 Linear Gaussian (para)

Linear Gaussian (para) 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 polarization, pola=1 with TM and pola=0 with TE. 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 $E^2_{\mathrm{TE}} = (1 - \mathrm{pola}^2)E^2$.

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

Note that this Gaussian wave is calculated in accordance with the paraxial approximation and as such does not satisfy rigorously the Maxwell's equations. For a wave propagating along the z direction and polarized along the x axis we have[?] :

$$E_x = E_0\sqrt{2}\frac{w_0}{w}e^{-\rho^2/w^2}e^{ik_0\rho^2 R(z)/2}e^{i(k_0 z+\eta)} \tag{5.17}$$

$$w = \sqrt{2}w_0\sqrt{1+\frac{z^2}{z_0^2}} \tag{5.18}$$

$$z_0 = k_0 w_0^2 \tag{5.19}$$

$$R(z) = \frac{z}{z^2+z_0^2} \tag{5.20}$$

$$\eta = \tan^{-1}(z/z_0). \tag{5.21}$$

We remark that for $z = 0$ the Gaussian beam has the same magnitude that those computed rigorously. The field and the irradiance at the center of the waist are computed through

$$E_0 = \sqrt{\frac{2P_0}{\pi c\varepsilon_0 w_0^2}} \tag{5.22}$$

$$\text{irradiance} = c\varepsilon_0 E_0^2/2 = \frac{P_0}{\pi w_0^2}. \tag{5.23}$$

5.2.11 Circular Gaussian (para)

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

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 is calculated in accordance with the paraxial approximation and as such does not satisfy rigorously the Maxwell's equations.

5.2.12 Speckle $k_z > 0$

The speckle is done as for a Gaussian beam. For a speckle polarized along the x axis, the Fourier component is then written as :

$$\boldsymbol{A}(k_x, k_y) = E_0(k_z\boldsymbol{i} - k_x\boldsymbol{k})\frac{1}{\sqrt{k_x^2 + k_z^2}}e^{i\varphi}, \tag{5.24}$$

with φ a random variable between 0 and 2π. We calculate the incident field as :

$$\boldsymbol{E}_{\mathrm{ref}}(x, y, z) = \int\int_{k_0\mathrm{NA}} \boldsymbol{A}_{\mathrm{ref}}(k_x, k_y, z)e^{i(k_x(x-x_0)+k_y(y-y_0)-k_z z_0)}\mathrm{d}\boldsymbol{k}_{\parallel}, \tag{5.25}$$

with NA the numerical aperture of the microscope. $\boldsymbol{r}_0$ allows the speckle to be shifted in one direction and the seed allows to change the speckle distribution. The power is calculated in the Fourier domain and E_0 is fixed to match that given in the in the graphical interface. This speckle is calculated for $k_z > 0$ for all the Fourier component, the irradiance is not calculated because it has no meaning for a speckle.

5.2.13 Speckle

This speckle is made exactly like the $k_z > 0$ speckle plus a speckle obtained for $k_z < 0$. We then have a three-dimensional speckle.

5.2.14 Confocal $k_z > 0$

The $k_z > 0$ confocal is made following the same principle as the speckle $k_z > 0$:

$$\boldsymbol{A}(k_x, k_y) = E_0(k_z\boldsymbol{i} - k_x\boldsymbol{k})\frac{1}{\sqrt{k_x^2 + k_z^2}}. \tag{5.26}$$

We then calculate the incident field as :

$$\boldsymbol{E}_{\mathrm{ref}}(x, y, z) = \int\int_{k_0} \boldsymbol{A}_{\mathrm{ref}}(k_x, k_y, z)e^{i(k_x(x-x_0)+k_y(y-y_0)-k_z z_0)}\mathrm{d}\boldsymbol{k}_{\parallel}, \tag{5.27}$$

with the power calculated in the Fourier plane and the irradiance estimated at $P_0/\pi k_0^2$. $\boldsymbol{r}_0$ is used to position the the focus of the confocal.

5.2.15 Confocal

The total confocal is made with the confocal $k_z > 0$ to which we add a confocal $k_z < 0$.

5.2.16 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

Definition of the object

Contents

6.1 Introduction **35**
6.2 Type of the object **36**
6.2.1 Sphere 36
6.2.2 Inhomogeneous sphere 37
6.2.3 Random sphere (length) 37
6.2.4 Random sphere (meshsize) 38
6.2.5 Cube 38
6.2.6 Cuboid (length) 39
6.2.7 Cuboid (meshsize) 39
6.2.8 Inhomogeneous Cuboid (length) 39
6.2.9 Inhomogeneous Cuboid (meshsize) 40
6.2.10 Ellipsoid 40
6.2.11 Multiple spheres 41
6.2.12 Multiple spheres surrounded by random spheres . . 41
6.2.13 Cylinder 42
6.2.14 Concentric spheres 43
6.2.15 Arbitrary object 43
6.3 Choose the relative permittivity **44**
6.4 Choose the discretization **44**

6.1 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.

6.2 Type of the object

The list of the predefined objects is the following:

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

When the objects as the cube or the cuboid have their edges turned with respect to the axes of the system of coordinates, the angles of Euler are used as defined in Fig. 6.1. The rotation centre being the inertia centre of the object and the matrix of rotation reads:

$$\boldsymbol{A} = \begin{pmatrix} \cos(\psi)\cos(\varphi) - \sin(\psi)\cos(\theta)\sin(\varphi) & -\cos(\psi)\sin(\varphi) - \sin(\psi)\cos(\theta)\cos(\varphi) & \sin(\psi)\sin(\theta) \\ \sin(\psi)\cos(\varphi) + \cos(\psi)\cos(\theta)\sin(\varphi) & -\sin(\psi)\sin(\varphi) + \cos(\psi)\cos(\theta)\cos(\varphi) & -\cos(\psi)\sin(\theta) \\ \sin(\theta)\sin(\varphi) & \sin(\theta)\cos(\varphi) & \cos(\theta) \end{pmatrix} \tag{6.1}$$

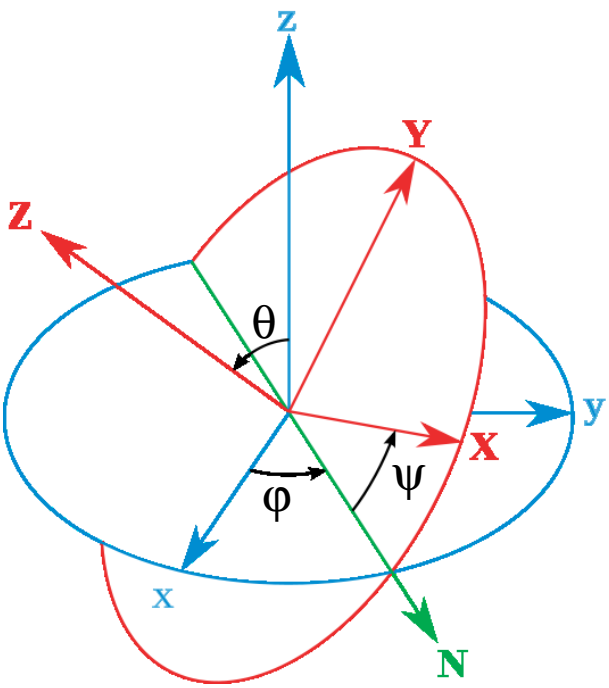


Figure 6.1 : *Definition of the angles of Euler according to the convention $Z - X - Z$. Scheme taken from Wikipedia*

6.2.1 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

6.2.2 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 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 seed
- The correlation length l_c
- The magnitude of oscillation A

6.2.3 Random sphere (length)

All the spheres are constituted with the same permittivity and the same radius but are distributed randomly in 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 radius of the spheres
- The density of sphere, *i.e.* d =volume of the sphere divided by the volume of the cuboid. d should satisfy the inequality $0 < d < 0.5$. If d is above 2, then it corresponds to the number of sphere in the box.

6.2.4 Random sphere (meshsize)

All the spheres are constituted with the same permittivity and the same radius but are distributed randomly in cuboid. There are ten 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 radius of the spheres
- The seed
- The density of sphere, *i.e.* d =volume of the sphere divided by the volume of the cuboid. d should satisfy the inequality $0 < d < 0.5$. If d is above 2, then it corresponds to the number of sphere in the box.

6.2.5 Cube

For the cube, there are seven 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
- 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

6.2.6 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

6.2.7 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

6.2.8 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 magnitude of oscillation A

6.2.9 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 magnitude of oscillation A

6.2.10 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

6.2.11 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.

6.2.12 Multiple spheres surrounded by random spheres

For multiple spheres whose position and radius are known surrounded by random spheres (all random spheres have the same permittivity and radius), it is first necessary to choose the number N of desired spheres using the line below labeled *number of objects*. Then, when you click on *Props*, N windows appear that you need to fill in. The first $N-1$ windows are dedicated to the spheres whose position and radius are known, while the last object pertains to the random medium.

For example, if we want two spheres, one with a radius of 500 nm located at $x = z = -500$ nm and another with a radius of 300 nm located at $x = z = 300$ nm surrounded by random spheres of 100 nm with a density of 0.1 in a cube of 2 μm on each side centered at the origin. We choose $N = 3$, then we enter the first sphere in object 1, the second sphere in object 2, and in object 3, we enter everything related to the random medium, see Fig. 6.2. Note that in objects 1 and 2, the characteristics of the random medium are recalled, but the fields are not active. The field X (and Y and Z) center corresponds to the position of the sphere for the first $N-1$ objects and to the position of the center of the box that contains the random spheres for the last object.

Note that all the small random spheres are entirely contained within the box and never within the known spheres. There can be no truncated random spheres.

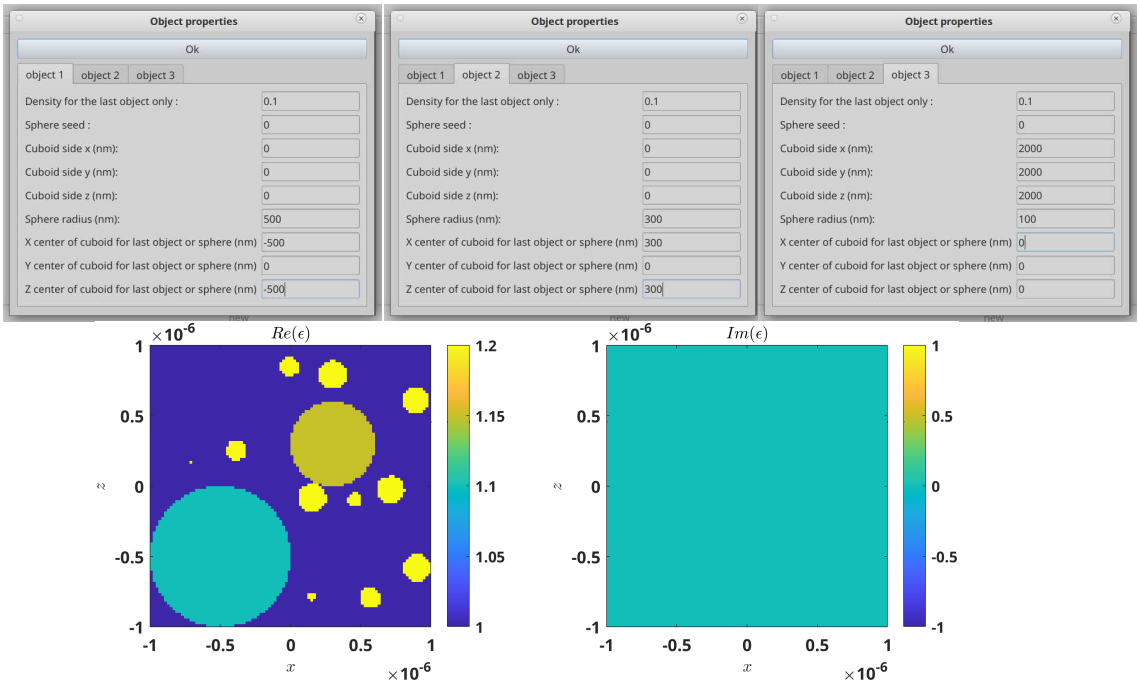


Figure 6.2 : *Presentation of the fields for the case of 2 spheres with radii of 500 and 300 nm surrounded by small spheres with a radius of 100 nm, randomly distributed within a cube of 2 μm on each side centered at the origin. Below, we have plotted the relative permittivity in the plane $y = 0$.*

6.2.13 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

6.2.14 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.

6.2.15 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.
- 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

6.3 Choose the relative permittivity

When the object or objects are chosen, it is then convenient to enter the relative permittivity. Apart from the arbitrary object, all the defined objects by default in the code are considered as being homogeneous. 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.

Please note that the distances must be entered in metres, and relative permittivities are complex numbers and must be written in the following form: (1.0,0.0) for example for the permittivity of vacuum. The total number of meshes is therefore $N = n_x n_y n_z$. It is also advisable not to take high prime numbers for the number of meshes in a given direction if one does not want the FFTs to be slow.

6.4 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 7

Possible study with the code

Contents

7.1 Introduction 46
7.2 Study in far field 47
7.3 Microscopy 49
7.4 Confocal microscopy 52
7.5 Study in near field 54
7.6 Optical force and torque 55

7.1 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.

Note also that if the calculation asked has a large number of discretization and that we are not interested by the output files in .mat (needs to use matlab), then we have the option "Do not write mat file". This requires the code to write no .mat file, and allows the code to go faster, less fill the hard drive and be better parallelized.

7.2 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:

$$C_{\text{ext}} = \frac{4\pi k_0}{\|\boldsymbol{E}_0\|^2} \sum_{j=1}^{N} \text{Im}\left[\boldsymbol{E}_0^*(\boldsymbol{r}_j).\boldsymbol{p}(\boldsymbol{r}_j)\right] \tag{7.1}$$

$$C_{\text{abs}} = \frac{4\pi k_0}{\|\boldsymbol{E}_0\|^2} \sum_{j=1}^{N} \left[\text{Im}\left[\boldsymbol{p}(\boldsymbol{r}_j).(\alpha^{-1}(\boldsymbol{r}_j))^*\boldsymbol{p}^*(\boldsymbol{r}_j)\right] - \frac{2}{3}k_0^3\|\boldsymbol{p}^*(\boldsymbol{r}_j)\|^2\right]. \tag{7.2}$$

- *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 and calculates differential cross section, *i.e.* $\langle \boldsymbol{S} \rangle .\boldsymbol{n}R^2$ with $\boldsymbol{S}$ the Poynting vector, $\boldsymbol{n}$ the direction of observation. The software gives a 2D representation of the Poynting vector in the plane plane (k_x, k_y) for $k_z > 0$ and $k_z < 0$, and to have a 3D representation 3D representation you have to use matlab where 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.

$$C_{\text{sca}} = \frac{k_0^4}{\|\boldsymbol{E}_0\|^2} \int \left\| \sum_{j=1}^{N} \left[\boldsymbol{p}(\boldsymbol{r}_j) - \boldsymbol{n}(\boldsymbol{n}.\boldsymbol{p}(\boldsymbol{r}_j))\right] e^{-ik_0 \boldsymbol{n}.\boldsymbol{r}_j} \right\|^2 \mathrm{d}\Omega \tag{7.3}$$

$$\begin{aligned} g &= \frac{k_0^3}{C_{\text{sca}}\|\boldsymbol{E}_0\|^2} \\ &\times \int \boldsymbol{n}.\boldsymbol{k}_0 \left\| \sum_{j=1}^{N} \left[\boldsymbol{p}(\boldsymbol{r}_j) - \boldsymbol{n}(\boldsymbol{n}.\boldsymbol{p}(\boldsymbol{r}_j))\right] e^{-ik_0 \boldsymbol{n}.\boldsymbol{r}_j} \right\|^2 \mathrm{d}\Omega \end{aligned} \tag{7.4}$$

$$\langle \boldsymbol{S} \rangle .\boldsymbol{n} R^2 = \frac{ck_0^4}{8\pi} \left\| \sum_{j=1}^{N} \left[\boldsymbol{p}(\boldsymbol{r}_j) - \boldsymbol{n}(\boldsymbol{n}.\boldsymbol{p}(\boldsymbol{r}_j))\right] e^{-ik_0 \boldsymbol{n}.\boldsymbol{r}_j} \right\|^2 \tag{7.5}$$

Another 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. The N chosen for the moment is $N = 256$. 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 with matlab is done as previously, *i.e.* with *Ntheta* and *Nphi* starting with an interpolation upon the calculated points with the FFT.

- *Energy Conservation.* 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.

  In the case of plane waves, which by definition have infinite energy, the w_0 term (waist) is redefined so that the laser the laser beam is distributed over the size of the FFT, *i.e.* an area of $(N\Delta x)^2$.

7.3 Microscopy

This option first asks for the type of microscope required: Holographic microscope holographic microscope, brightfield microscope, darkfield and phase microscope and phase microscope, etc. Depending on the microscope chosen, there are different field to fill in, such as the numerical aperture of the lens objective used (necessarily between 0 and 1). By default, the lenses are placed parallel to the plane (x, y) and at the side of the positive z. The focus of the microscope is placed to the origin of the frame but can be changed via the field "Position of the focal plane". (Fig. 7.1). The magnification of the microscope is G and should be above 1.

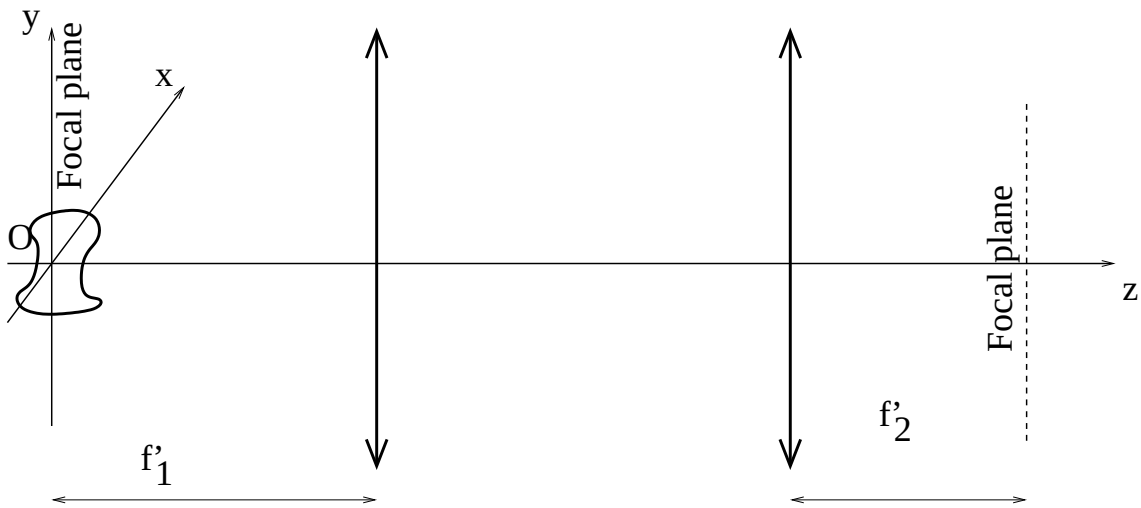


Figure 7.1 : *Simplified figure of the microscope. The object focus of the objective lens is at the origin of the frame. The axis of the lens is confounded with the z axis and at the side of the positive z.*

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 $L = N\Delta x$.

The diffracted field in far field at a distance r of the origin can written as $\boldsymbol{E} = \boldsymbol{S}(k_x, k_y, \boldsymbol{r}_{\text{object}})\frac{e^{ikr}}{r}$. The field after the first lens is then defined as: $\boldsymbol{E}^f = \frac{\boldsymbol{S}(k_x,k_y,\boldsymbol{r}_{\text{object}})}{-2i\pi\gamma}$ with $\gamma = \sqrt{k_0^2 - k_x^2 - k_y^2}$ and the image through the microscope is given by its Fourier transform, $\boldsymbol{E}^i = \mathcal{F}(\boldsymbol{E}^f)$.

To take into account the magnification of the microscope in the image we perform a rotation of the vector $\boldsymbol{E}^f$ before its Fourier transform as:

$$\boldsymbol{E}^i = \mathcal{F}(R(\theta)\boldsymbol{E}^f) \tag{7.6}$$

$$\text{with } R(\theta) = \begin{pmatrix} u_x^2 + \cos\theta(1-u_x^2) & u_x u_y(1-\cos\theta) & u_y \sin\theta \\ u_x u_y(1-\cos\theta) & u_y^2 + \cos\theta(1-u_y^2) & -u_x \sin\theta \\ -u_y \sin\theta & u_x \sin\theta & \cos\theta \end{pmatrix} \tag{7.7}$$

$$\theta = \sin^{-1}[\sin(-\beta)/G] - \beta \tag{7.8}$$

$$\beta = \cos^{-1}(k_z/k_0) \tag{7.9}$$

$$u_x = -k_y/k_{\|} \tag{7.10}$$

$$u_y = k_x/k_{\|}. \tag{7.11}$$

The code offers the possibility to simulate a transmission microscope ($k_z > 0$) or a reflection microscope ($k_z < 0$). Note, in transmission only because there is no specular in reflection (except for illumination with a confocal and total speckle), that when the total field is calculated (diffracted field plus incident field) in the case of one or plane waves, the plane wave in Fourier space is a Dirac which we place at the nearest pixel corresponding to the incident wave vector of the incident wave. In this case the dynamics of the image is such that only the incident plane wave is visible and the diffracted field too weak to appear. This option is especially interesting in the case of case of a Gaussian beam.

In the microscopy menu, different microscopes are proposed.

The first microscope in the list is the holographic microscope, *Holographic*, which is a rather a special microscope because it illuminated by a coherent incident (often a plane wave or a Gaussian beam but this is not not mandatory), then the diffracted field is measured in modulus and phase through an interferential system (off-axis for example) and this for the different x, y or z components (generally the experimenter does not have access to the z component, but this component is very small due to the high magnification of the microscope). The result given by the code is therefore the field diffracted by the object (Fourier plane) in modulus and phase and the image obtained through the microscope image obtained through the microscope at the image focus position (Plane image) in modulus and phase with and without the presence of the incident field. The incident field is the one defined in the code in the the illumination properties section.

The other proposed microscopes are more classical in the sense that the illumination is incoherent and we finally obtain the light intensity in the image focal plane of the microscope. To obtain the incoherent illumination we illuminate by numerous plane waves with polarizations and we sum up all the images obtained in intensity. This calculation requires many illuminations and can therefore be very time consuming. The step of discretization of the incident illuminations in the Fourier domain is chosen such

that $\Delta k_{\text{inc}} < \pi/l$ where l is the maximum size of the sample and with the condition that $\Delta k_{\text{inc}} = m\Delta k$ with $m \in \mathbb{N}^*$. Note that if we use the matlab interface to plot the images then the incidents chosen by the code will be indicated in figure 560.

- *Brightfield* For this microscope it is necessary to define NA=the numerical aperture of the condenser, see Fig. 7.2(a). Then the intensity diffracted by the object alone is calculated, as well as the total intensity which corresponds to the brightfield microscope.

- *Darkfield & phase contrast*: Darkfield microscopy microscopy illuminates the object along a ring between NA (NA condenser in the graphical interface) and NA (central aperture), see Fig. 7.2(b). The incoherent sum of all the fields diffracted fields between NA and NA (central aperture) is made. The result is given in the "plane image" without the incident field (dark field) and for the phase microscope the incident field is added to the diffracted field obtained with the dark field. The incident field out of phase by $\pi/2$.

- *Darkfield cone& phase contrast*: This is the same as the same as the previous microscope except that the sum is made on the generators of the cone, see Fig. 7.2(c). To be preferred if NA (condenser)-NA (central aperture) is very small.

- *Schieren*: The illumination pupil of radius NA can be off-centre at any position, see Fig. 7.2(d). The code returns the intensity diffracted by the object alone and the total intensity (incident+diffracted).

- Experimental phase contrast: This microscope uses the Darkfield illumination with an illumination between NA (codenser)-NA (central aperture), see Fig. 7.2(b). Then the phase of the diffracted field plus the incident is shifted of $\pi/2$ in the Fourier domain between NA (condenser) and NA (central aperture). The shift of the phase of the diffracted field plus incident field in the illumination ring corresponds exactly to what happens experimentally.

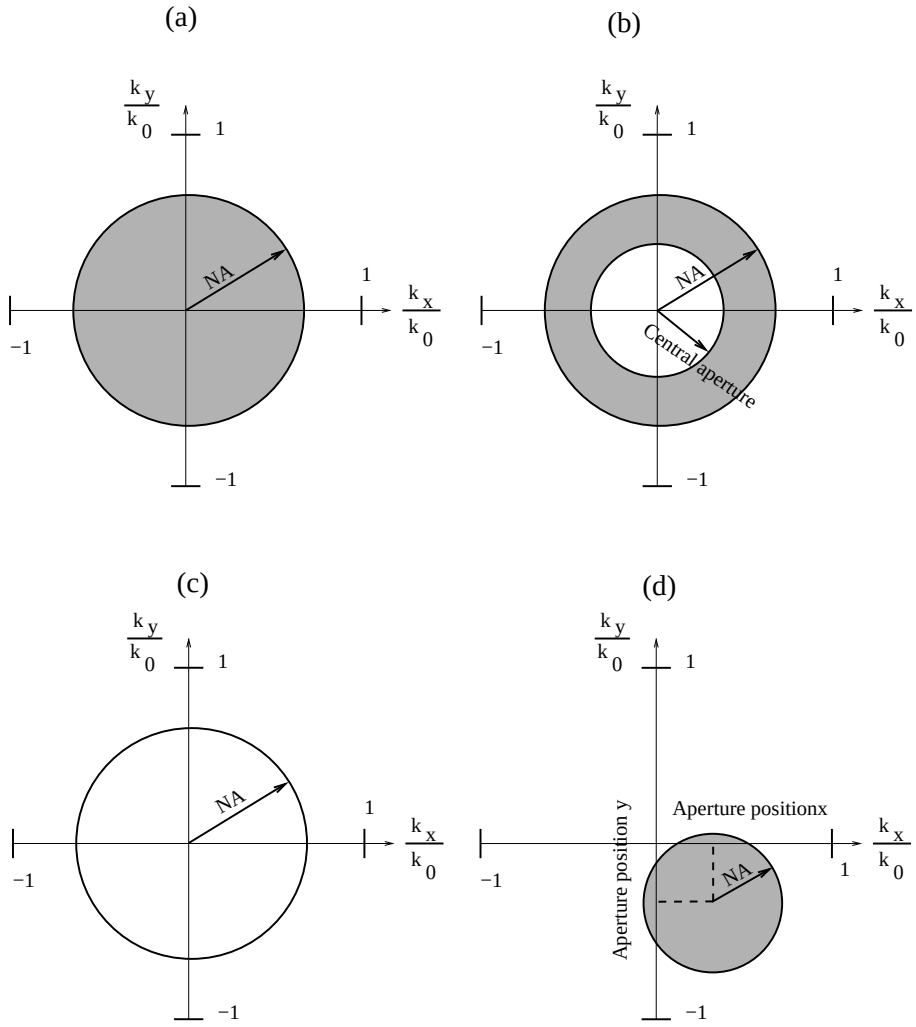


Figure 7.2 : *Different masks in the Fourier domain for the illumination to simulate different types of microscopes. (a) Illumination in a NA cone. (b) Illumination in a NA-NA$_{centralaperture}$ cone. (c) Illumination along the generatrix of an NA aperture cone. (d) Illumination in a pupil of NA aperture centred on the point* $(k_x/k_0, k_y/k_0)$.

7.4 Confocal microscopy

Confocal microscopy is an optical imaging technique that enables biological samples in three dimensions at high resolution. The principle of confocal microscopy is based on the use of a scanning system that enables the laser to be focus on a precise point on the sample. This point is then scanned to form a two-dimensional image. Then, the focus is moved in depth to form a series of three-dimensional images.

Confocal microscopy makes it possible to suppress signals from sample planes outside the focal plane, thus enabling images with high resolution.

The code simulates an ideal confocal microscope in reflection, *i.e.* the pinhole in the focal plane conjugate to the objective's focal plane is considered infinitely small. In a confocal microscope we get image stack (x, y) as a function of z. In the code, we enter the upper and lower limits of z as well as the desired step in z and the numerical aperture of the microscope's objective lens.

If we take a zero z step, only one plane will be calculated, and the same applies if the upper and lower bounds are identical.

In the case of the confocal transmission microscope, the incident field is not calculated, only the field diffracted by the object is used to compute the image.

The graphical interface shows only a (x, y) plane chosen in the middle of the image stack at z. You need to use matlab to access the image stack and be able to observe what's happening along the z axis.

Be careful, confocal microscopy should not be confused with confocal illumination.

The principle of calculation is as follows. For a confocal illumination polarized along the x axis, the microscope is successively illuminated by N plane waves (taken in the numerical aperture of the microscope) with the following structure:

$$\boldsymbol{E}_0(\boldsymbol{k}, \boldsymbol{r}) = \begin{pmatrix} \frac{k_z}{\sqrt{k_z^2+k_x^2}} \\ 0 \\ \frac{-k_x}{\sqrt{k_z^2+k_x^2}} \end{pmatrix} e^{i\boldsymbol{k}.\boldsymbol{r}}, \tag{7.12}$$

with $k_z = \sqrt{k_0^2 - k_x^2 - k_y^2}$ and $\boldsymbol{k}$ the incident wave vector wave vector of the plane wave, which corresponds to the confocal illumination seen Eq. (5.26). For each plane wave, the field diffracted by the object in the image plane, $\boldsymbol{E}_{\text{image}}(\boldsymbol{k}, \boldsymbol{r})$, is calculated with the appropriate magnification using numerical aperture of the microscope. The field measured by the confocal microscope at a point $\boldsymbol{r}_0$ is written as:

$$\boldsymbol{E}_{\text{confocal}}(\boldsymbol{r}_0) = \sum_{\boldsymbol{k}} \boldsymbol{E}_{\text{image}}(\boldsymbol{k}, \boldsymbol{r}_0) e^{i(k_x x_0 + k_y y_0 - k_z z_0)}. \tag{7.13}$$

The incident field is focused at a point $\boldsymbol{r}_0$ and the field is measured in the focal image plane at the conjugate point of $\boldsymbol{r}_0$. In this computation the diaphragm of the confocal is infinitely small. Note that in the exponential $e^{i(k_x x_0 + k_y y_0 - k_z z_0)}$, we have $+(k_x x_0 + k_y y_0)$ as the microscope inverts the final image.

7.5 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.[5] :

$$\boldsymbol{E}_{\text{macro}} = 3\left(\varepsilon + 2 - i\frac{k_0^3 d^3}{2\pi}(\varepsilon - 1)\right)^{-1} \boldsymbol{E}_{\text{local}} \tag{7.14}$$

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

- *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 cube 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 sideband (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 sideband x of 2, 3 for y and 4 for z, we get a box of size:

$$l_x = 100 + 2 \times 2 \times 10 = 140 \text{ nm} \tag{7.15}$$
$$l_y = 100 + 2 \times 3 \times 10 = 160 \text{ nm} \tag{7.16}$$
$$l_z = 100 + 2 \times 4 \times 10 = 180 \text{ nm} \tag{7.17}$$
$$\tag{7.18}$$

item *MDDA*: This option allows the field inside the object to be calculated by placing the object in an absorbing background of the form $\varepsilon = 1 + i\eta$.[33,34] This option improves the convergence of iterative methods by improving the conditioning of the matrix. When this option is selected, four fields appear. The "Absorbing factor" field corresponds

to the value of η. The other field corresponds to the desired "no reflective boundary condition" (NRBC):

$$n(\boldsymbol{r}) = \frac{n_{\text{air}} + n_{\text{ref}}}{2} + \frac{(n_{\text{air}} - n_{\text{ref}})}{2} \frac{\tanh[\alpha(2\delta/l - 1)]}{\tanh \alpha}, \tag{7.19}$$

with $\alpha = 1,\ 2,\ 3$, and tanh can be replaced by the erf function. The two fields below correspond to the width of the NRBC (l) and the width of the apodization (L). These two widths are given in units of wavelength. The apodization corresponds to a distance L over which the incident field is apodized using a cosine function.

When using this function, it is necessary to ensure that l, L, and η are properly chosen. Note that only three objects are currently implemented with this option (inhomogeneous and sphere), and only the plane wave and the paraxial Gaussian beam are currently programmed. When the discretization value is entered, the mesh size is calculated based on the total length, *i.e.* $d = (2L + 2l + a)$, where a is the size of the object. Therefore, the width of the NRBC and the apodization must be taken into account in order to determine the final meshsize.

7.6 Optical force and torque

When the force option is selected, four possibilities appear:

- *Optical force*: Calculation of the optical force exerting on one or more objects.
- *Optical force density*: Enables us to draw the density of the optical force.
- *Optical torque*: Calculation of the optical torque exerting on one or more objects. The torque is computed for an origin placed in the gravity center of the object.
- *Optical torque density*: Enables us to draw the density of the optical force torque.

The net optical force and torque experienced by the object are computed with[35,36]:

$$F_u = \frac{1}{2} \sum_{j=1}^{N} \text{Re} \left(\sum_{v=1}^{3} p_v(\boldsymbol{r}_j) \frac{\partial (E_v(\boldsymbol{r}_j))^*}{\partial u} \right) \tag{7.20}$$

$$\boldsymbol{\Gamma} = \sum_{j=1}^{N} \left[\boldsymbol{r}_j \times \boldsymbol{F}(\boldsymbol{r}_j^g) + \frac{1}{2} \text{Re} \left\{ \boldsymbol{p}(\boldsymbol{r}_j) \times [\boldsymbol{p}(\boldsymbol{r}_j)/\alpha_{\text{CM}}(\boldsymbol{r}_j)]^* \right\} \right]. \tag{7.21}$$

where u or v, stand for either x ,y, or z. The symbol $*$ denotes the complex conjugate. $\boldsymbol{r}_j^g$ is the vector between j and the center of mass of the object.

Chapter 8

Representation of the results

Contents

8.1 Introduction **56**
8.2 Digital exits **56**
8.3 Graphics **58**
8.3.1 Plot epsilon/dipoles 58
8.3.2 Far field and microscopy 58
8.3.2.1 Plot Poynting vector 58
8.3.2.2 Plot microscopy 58
8.3.3 Study of the near field 59
8.3.4 optical force and torque 59

8.1 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.

8.2 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.
- $\lambda/(10|n|)$: 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.
- *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.
- *Optical force x*: Optical force according to the axis x.
- *Optical force y*: Optical force according to the axis y.
- *Optical force z*: Optical force according to the axis z.
- *Optical force modulus*: Modulus of the optical force.
- *Optical torque x*: Optical torque according to the axis x.
- *Optical torque y*: Optical torque according to the axis x.
- *Optical torque z*: Optical torque according to the axis x.
- *Optical torque modulus* Modulus of the optical torque.

8.3 Graphics

8.3.1 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.

8.3.2 Far field and microscopy

8.3.2.1 Plot Poynting vector

Plot Poynting: enables us to draw the modulus of the Poynting vector in the plane (k_x, k_y) for $k_z > 0$ and $k_z < 0$.

8.3.2.2 Plot microscopy

Plot microscopy :enables us to draw the diffracted field in far field by the object may this be either of the modulus of the field or of the x, y or z. Then, the vectorial field on the picture plane is represented by considering a magnification G for the microscope.

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 with the origin of the phase at the origin of the frame (x, y, z). If the computation is done by radiation of the dipoles, then, the obtained picture has a size $k_0 NA$ and discretized as $\Delta k_x = 2k_0 NA/N$, and if this one is done with Fourier transform, then, the size of the picture is fixed by discretization of the object Δx with the relation $\Delta x \Delta k = 2\pi/N$.

The field inside the picture plane is calculated with Fourier transform. So, we have with the calculation by radiation of the dipoles:

$$\Delta x \Delta k_x = \frac{2\pi}{N} \tag{8.1}$$

$$\Delta x 2k_0 \mathrm{NA} = 2\pi \tag{8.2}$$

$$\Delta x = \frac{\lambda}{2\mathrm{NA}} \tag{8.3}$$

The size of the picture is then $\lambda/(2\mathrm{NA})$.

If the calculation of the diffracted field has been made by FFT, then, the discretization is that of the mesh.

For the holographic microscope we present the Fourier plane (with and without the incident field) and the image plane (with and without the incident field). For the brightfield options we present the image plane with the incident field (brightfield) and without the incident field (a kind of darkfield). For the darkfield & phase option we present the image plane without the incident (darkfield) and with the incident phase shifted of $\pi/2$ (phase).

8.3.3 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.

8.3.4 optical force and torque

- The first button *Field* enables us to choose to represent the optical force or the optical torque.
- The button *Type* enables us to choose 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 azimuth). *Plot* x (y or z) draws the cur. *Plot all* x draws all the cuts at once.

Chapter 9

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

Contents

9.1 Introduction 60
9.2 List of all exit files 61
9.3 List of figure names created by ifdda.m 63

9.1 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 and to read them through other softwares 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.

9.2 List of all exit files

- 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 (x,y) 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 (x,y) 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) mlocalfieldx (x,y) 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 (x,y) 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 (x,y) 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 (x,y) 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.
- poynting pointing contains the modulus of the vector of Poynting in theta and phi direction of size (Ntheta+1)*Nphi.
- (3D) forcex (y,z) contains the x component of the optical force only inside the object.
- (3D) torquex (y,z) contains the x component of the optical torque force only inside the object.
- (2D) fourier(x,y,z) contains the diffracted field in the fourier plane in modulus (x,y,z).
- (2D) fourierinc(x,y,z) contains the total field in the fourier plane in modulus (x,y,z).
- kxfourier contains the coordinates of the Fourier plane.
- (2D) image(x,y,z) contains the diffracted field in the image plane in modulus (x,y,z).
- (2D) imageinc(x,y,z) contains the total field in the image plane in modulus (x,y,z).

- (2D) imagebf(x,y,z) contains the diffracted field in the image plane in modulus (x,y,z) for a dark field.
- (2D) imageincbf(x,y,z) contains the total field in the image plane in modulus (x,y,z) for a brightfield microscope.
- (2D) imagedf(x,y,z) contains the diffracted field in the image plane in modulus (x,y,z) for a dark field.
- (2D) imageincdf(x,y,z) contains the total field in the image plane in modulus (x,y,z) for a phase microscope.
- Imageconf (x,y,z) contains the intensity obtained with a confocal microscope. The size of the file is the size of the FFT squared by the number of slices in z chosen.
- ximage contains the position of the pixel for all the microscopy proposed by the code.
- Poynting contains the Poynting modulus fo the direction θ and ϕ.
- theta and phi contains the direction θ and ϕ for Poynting.
- 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.
- fieldconfm (3D) contain the modulus of the field for the confocal in reflexion or transmission.
- fieldconfx(y,z) (3D) contain the component x (y,z) of the field for the confocal in reflexion or transmission.

9.3 List of figure names created by ifdda.m

When using ifdda.m, you can select the "print" option to print the figures in any format (default eps). Figures are printed without sliders and other uiinterface matlab. Without the uiinterface commands, we use exportgraphics with a resolution of 300 dpi, and with the uiinterface commands, imwrite is used, and in this case the resolution cannot be selected. Note that the image formats available are not the same depending on whether imwrite or exportgraphics is chosen.

Figure name	Description
dipolepos.eps	Represents selected dipoles in three dimensions with a color level depending on the value of ε
epsilon.eps	Plot relative permittivity cross-sections of the object
incident.eps	Plot the incident field
local.eps	Plot the local field
macroscopic.eps	Plot the macroscopic field
poynting2d.eps	Plot the modulus of the Poynting vector in 2 dimensions for $k_>0$ and $k_z < 0$
poynting3d.eps	Plot the modulus of the Poynting vector in 3 dimensions
force2d.eps	Plot optical force density cross-sections
force3d.eps	Plot optical force density in 3 dimensions
torque2d.eps	Plot optical torque density cross-sections
torque3d.eps	Plot optical torque density in 3 dimensions
fourier.eps	Plot the diffracted field in the Fourier plane for the holographic microscope
fourierinc.eps	Plot in the Fourier plane the diffracted field plus the incident field for the holographic microscope (no incident field in reflection)
image.eps	Plot in the image plane the diffracted field for the holographic microscope
imageinc.eps	Plot in the image plane the diffracted field plus the incident field for the holographic microscope (no incident field in reflection)
angleincmic.eps	Plot all the plane waves taken to simulate the different microscopes (bright field, wide field, dark field, schieren, etc)
imagemic.eps	Plot the intensity in the image plane of the diffracted field for the different microscopes
imageincmic.eps	Plot the intensity in the image plane of the diffracted field plus the incident field for the different microscopes (no incident field in reflection)
confocal.eps	Plot the intensity in the image plane of the confocal microscope
angleincconfocal.eps	Plot all plane waves taken to simulate the confocal microscope

Table 9.1 : *Name of figures saved by the ifdda.m code with their description.*

Chapter 10

Some examples

Contents

10.1 Introduction 65
10.2 Test1 65
10.3 Test2 70
10.4 Test3 75
10.5 Test4 77

10.1 Introduction

In bin/tests there is the file options.db3. You should copy it in the directory bin as "cp options.db3 ../.", and then you launch the code. After four test configurations appear that allow you to see all the options in action.

10.2 Test1

The aim of test1 is to test a simple case and many options of the code to validate them. Figure 10.1 shows the options of the chosen configuration.

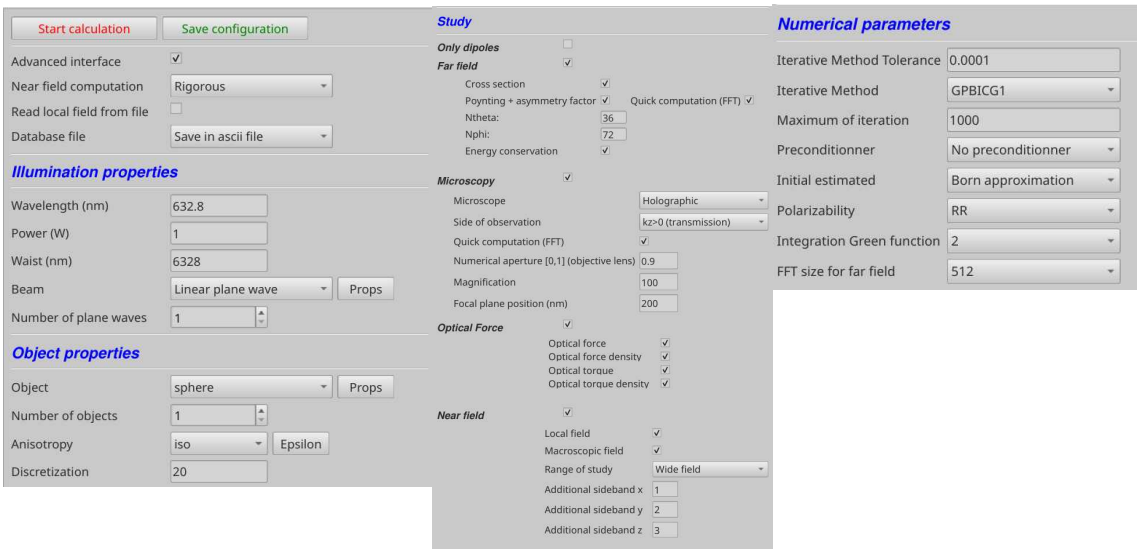


Figure 10.1 : *Test1: configuration taken.*

The following figures show the results obtained. The plots are done with Matlab and these are directly the eps files from the ifdda.m script that are used. The advantage of matlab in this case is to give all the figures in one go.

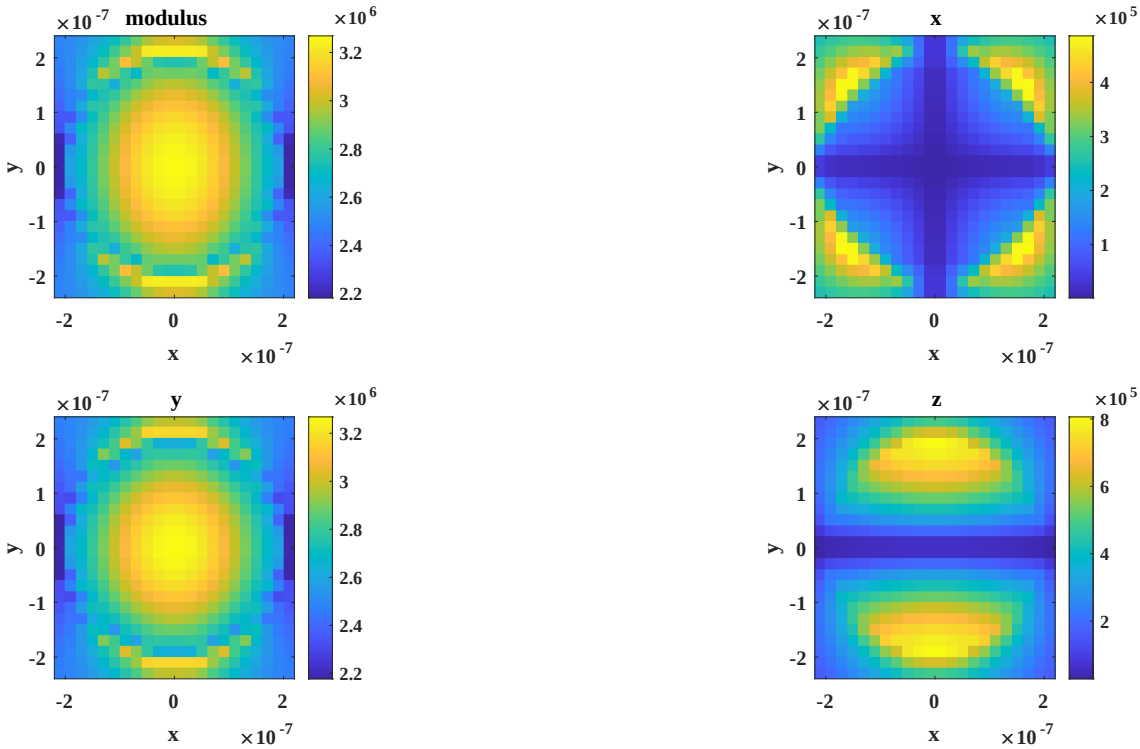


Figure 10.2 : *Modulus of the local field in* (x, y) *plane.*

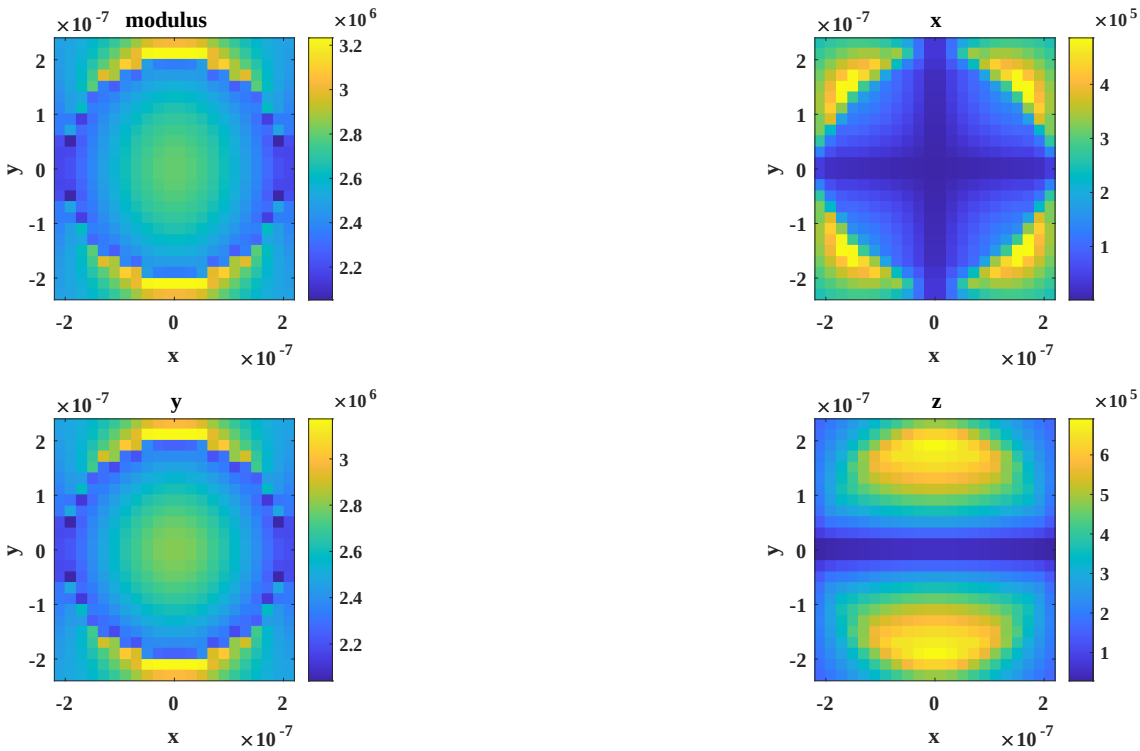


Figure 10.3 : *Modulus of the macroscopic field in* (x, y) *plane.*

Because the incident field is polarized along the y direction (TE), hence the y component of the field inside the sphere is the largest.

Poynting Modulus in k_x and k_y plane

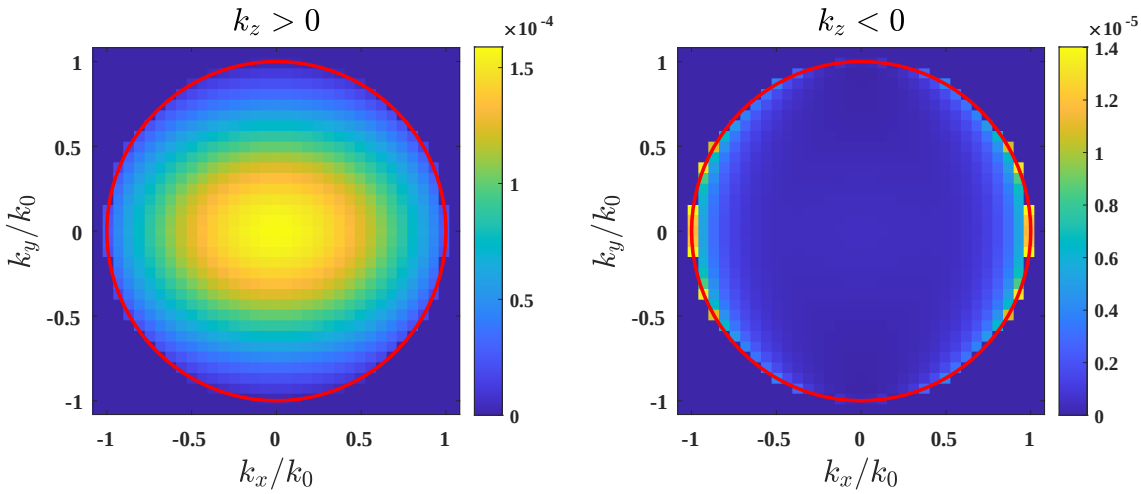


Figure 10.4 : *Modulus of the Poynting vector.*

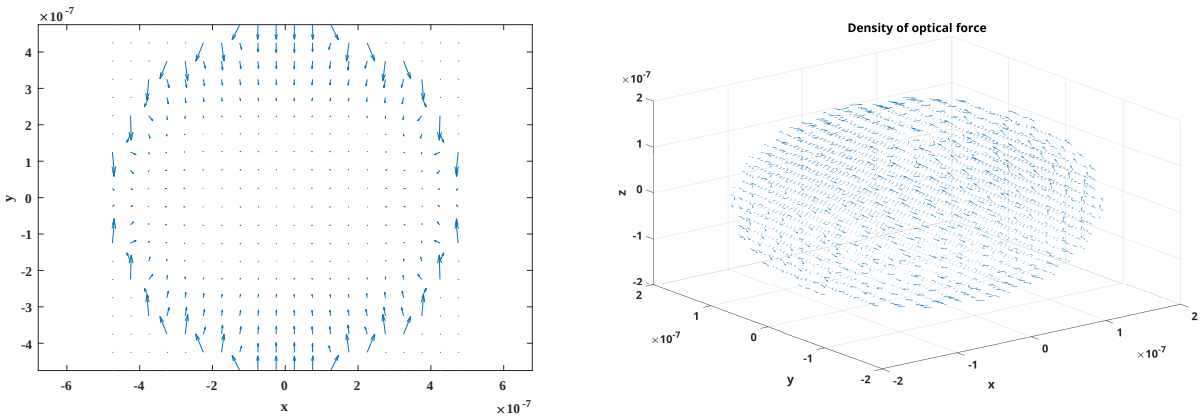


Figure 10.5 : *Optical force in the (x, y) plane and in 3D.*

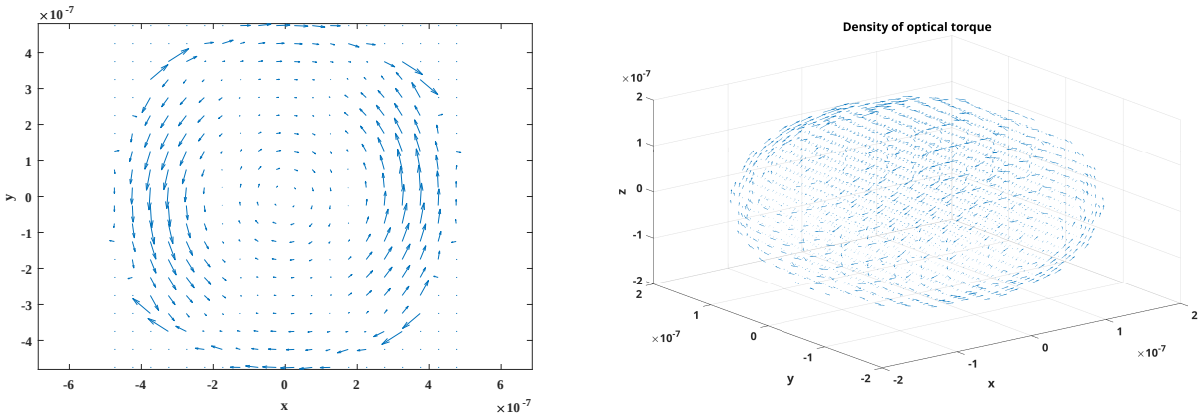


Figure 10.6 : *Optical torque in the (x, y) plane and in 3D.*

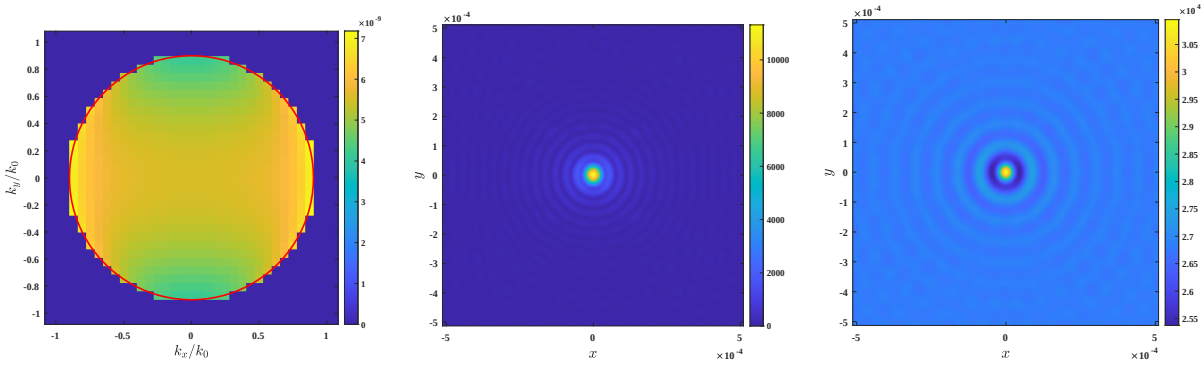


Figure 10.7 : *Microscopy in transmission: Modulus of the diffracted field in the Fourier plane (left), modulus of the diffracted field in the image plane (middle), and modulus of the total field in the image plane (right).*

10.3 Test2

The aim of the test2 is to test a simple case and many options code to validate them. Figure 10.8. shows the options of the chosen configuration. The illumination is done by two plane waves.

Figure 10.8 : *Test2: configuration taken.*

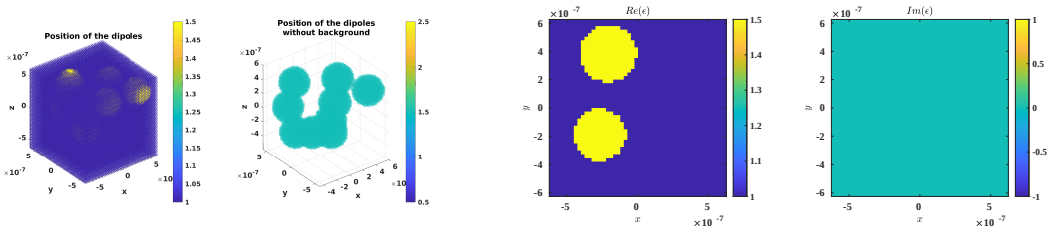


Figure 10.9 : *Object in 3D (left) and map of permittivity in the (x, y) plane (right).*

The following figures show the results obtained.

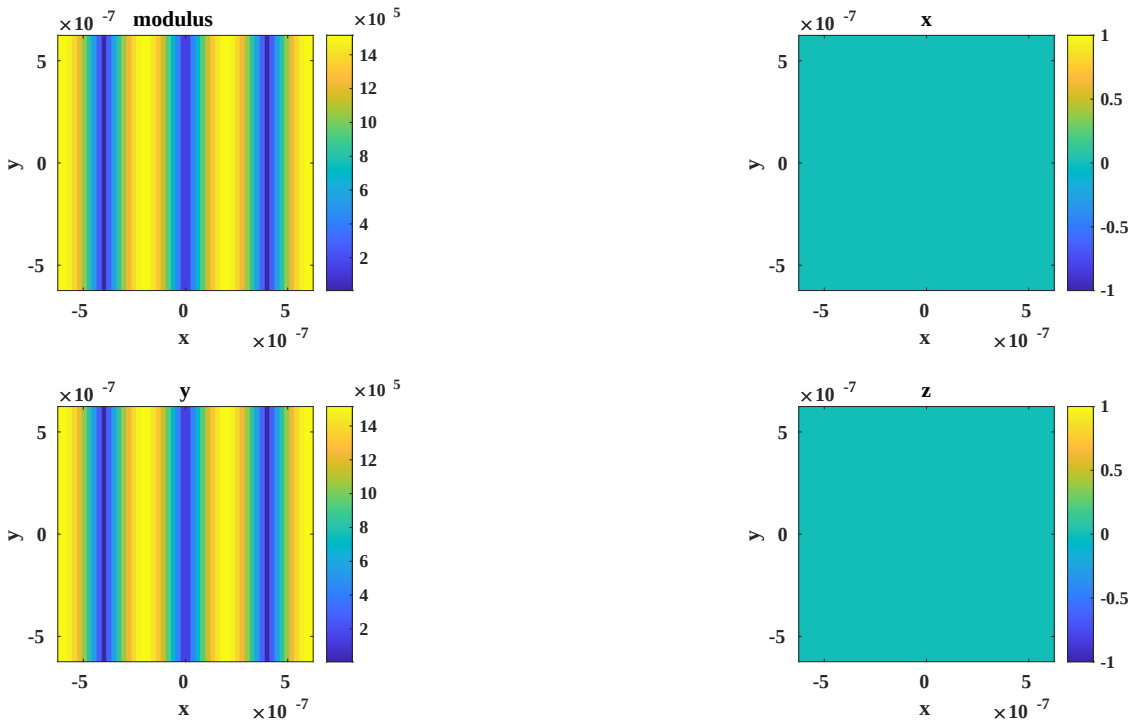


Figure 10.10 : *Modulus of the incident field in* (x, y) *plane.*

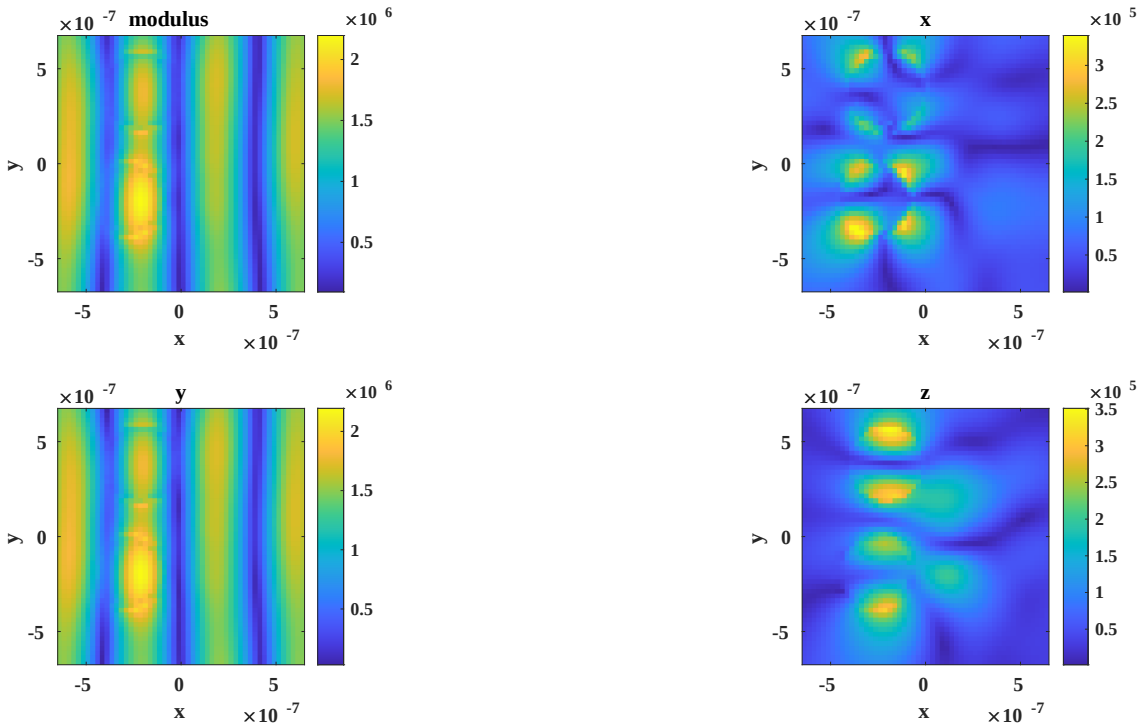


Figure 10.11 : *Modulus of the local field in* (x, y) *plane.*

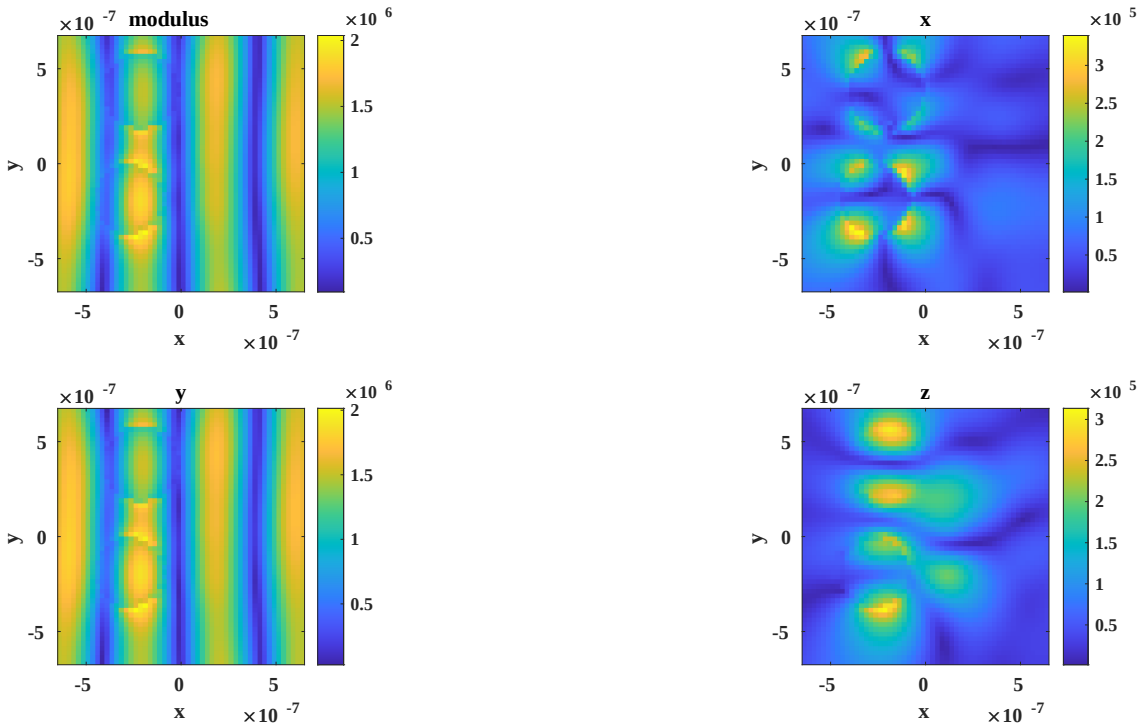


Figure 10.12 : *Modulus of the macroscopic field in (x, y) plane.*

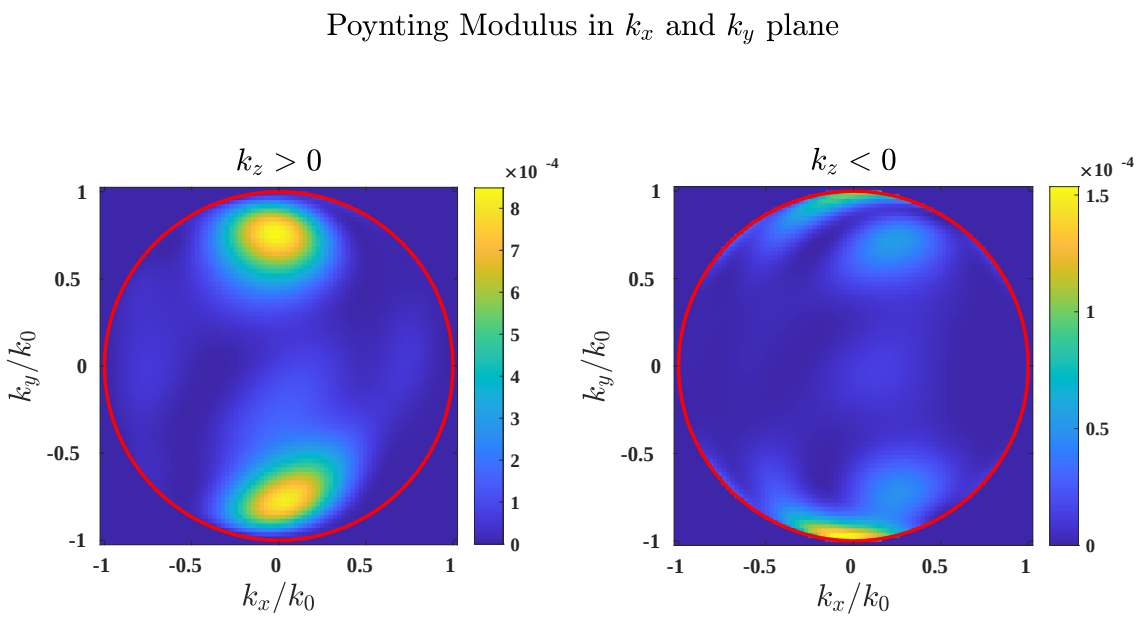


Figure 10.13 : *Modulus of the Poynting vector.*

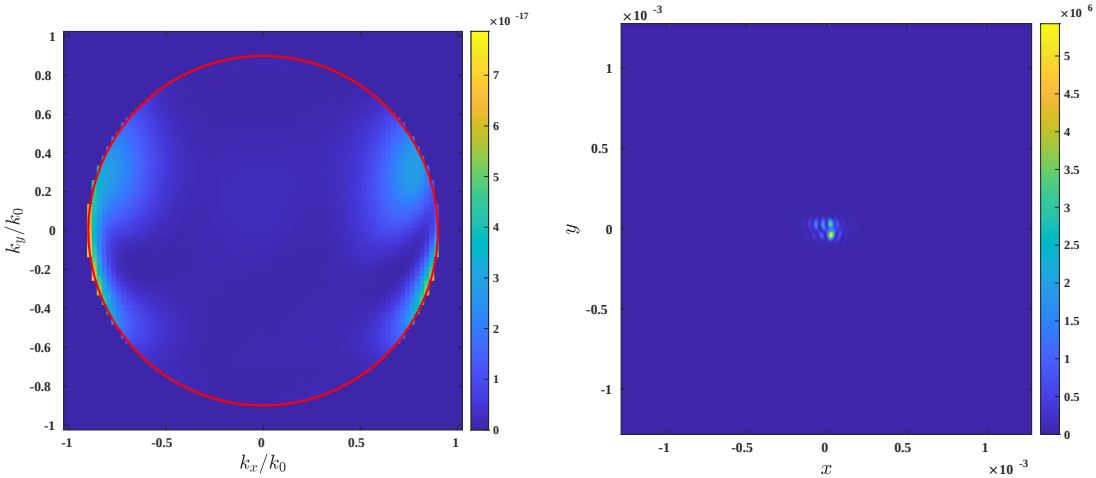


Figure 10.14 : *Microscopy in reflection: Modulus of the diffracted field in the Fourier plane (left) and in image plane (right).*

10.4 Test3

The aim of the test3 is to test the microscopy in dark field and bright field in transmission. One studies a sphere with a radius of 500 nm and permittivity 1.5.

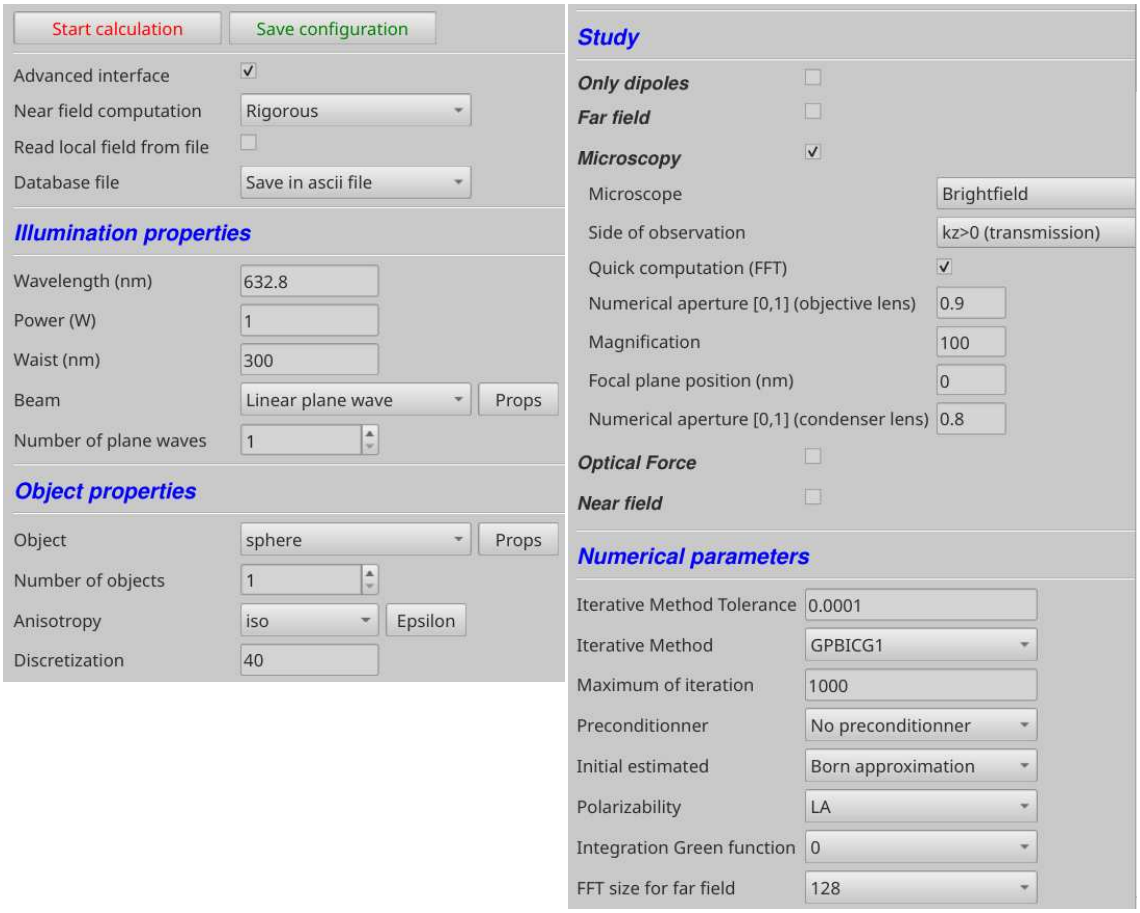


Figure 10.15 : *Test3: configuration taken.*

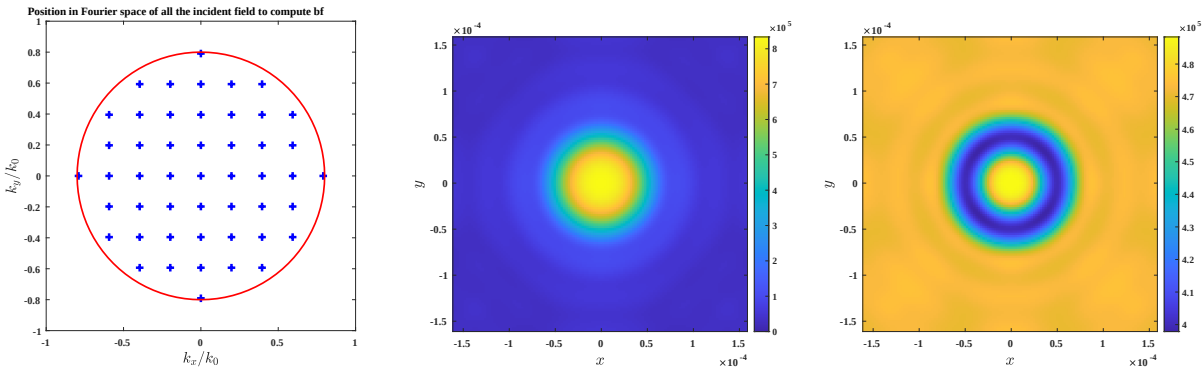


Figure 10.16 : *Microscopy in transmission: incident taken to make the image (left). Modulus of the diffracted field in the image plane (middle), and modulus of the total field in the image plane (right).*

10.5 Test4

Same configuration as in test3 for a dark field microscope.

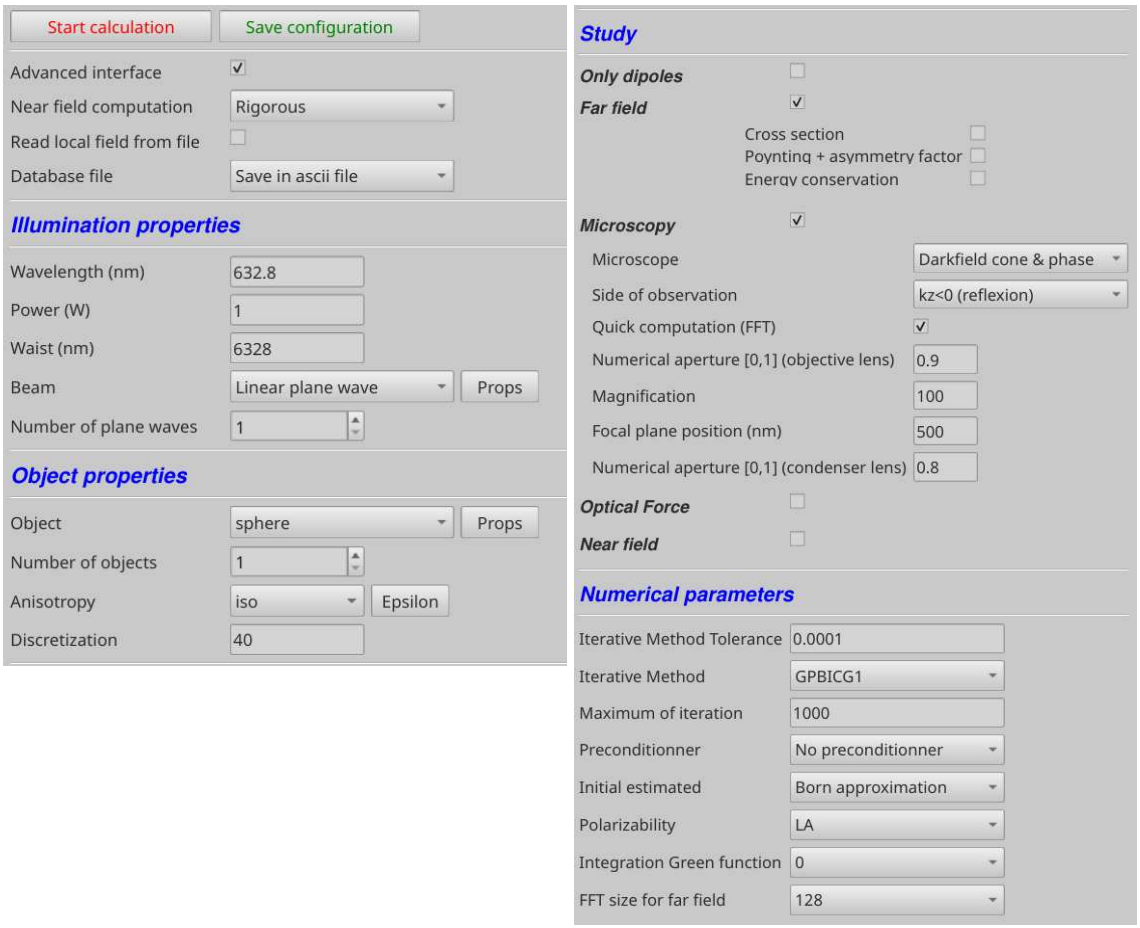


Figure 10.17 : *Test4: configuration taken.*

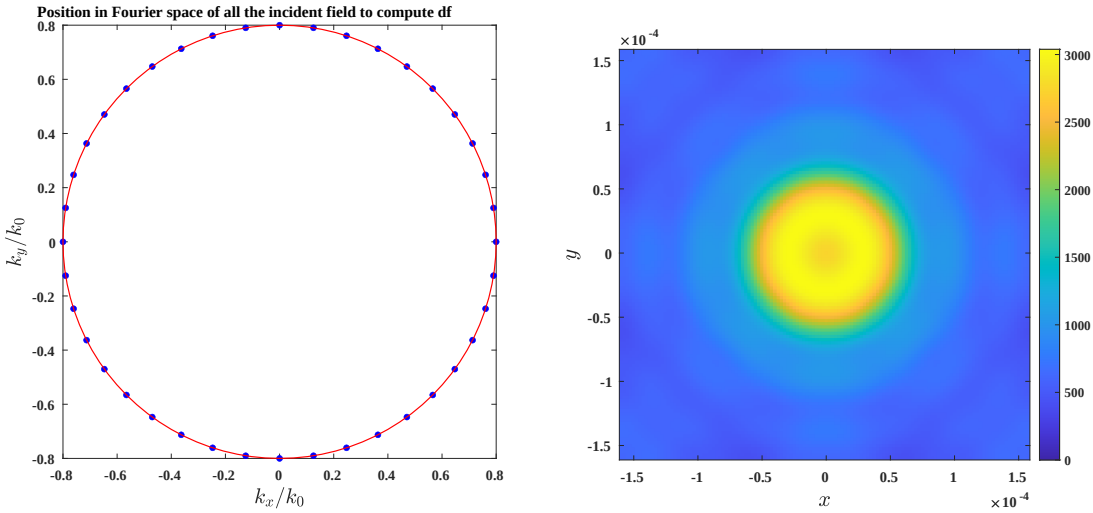


Figure 10.18 : *Microscope in reflection: incident taken to make the image (left). Modulus of the diffracted field in the image plane (right).*

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] A. Rahmani, P. C. Chaumet, F. de Fornel, and C. Girard, Phys. Rev. A **56**, 3245 (1997).

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

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

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

[7] U. S. Kamilov, I. N. Papadopoulos, M. H. Shoreh, A. Goy, C. Vonesch, M. Unser, and D. Psaltis, IEEE Transactions on Computational Imaging **2**, 59 (2016), ISSN 2333-9403.

[8] M. Chen, D. Ren, H.-Y. Liu, S. Chowdhury, and L. Waller, Optica **7**, 394 (2020), URL https://opg.optica.org/optica/abstract.cfm?URI=optica-7-5-394.

[9] A. Rahmani and G. W. Bryant, Opt. Lett. **25**, 433 (2000), URL https://opg.optica.org/ol/abstract.cfm?URI=ol-25-7-433.

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

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

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

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

[14] N. Piller and O. Martin, IEEE Transactions on Antennas and Propagation **46**, 1126 (1998).

[15] P. Gay-Balmaz and O. J. F. Martin, Appl. Opt. **40**, 4562 (2001).

[16] M. A. Yurkin, M. Min, and A. G. Hoekstra, Phys. Rev. E **82**, 036703 (2010).

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

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

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

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

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

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

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

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

[25] P. Sonneveld and M. B. van Gijzen, SIAM Journal on Scientific Computing **31**, 1035 (2009).

[26] G. L. G. Sleijpen and D. R. Fokkema, Electronic Transactions on Numerical Analysis **1**, 11 (1993).

[27] K. Aihara, Numerical Linear Algebra with Applications **27**, e2298 (2020).

[28] P. C. Chaumet, G. Maire, and A. Sentenac, J. Opt. Soc. Am. A **39**, 1462 (2022).

[29] T. F. Chan, SIAM Journal on Scientific and Statistical Computing **9**, 766 (1988).

[30] S. P. Groth, A. G. Polimeridis, and J. K. White, Journal of Quantitative Spectroscopy and Radiative Transfer **240**, 106689 (2020), ISSN 0022-4073.

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

[32] P. C. Chaumet, J. Opt. Soc. Am. A **23**, 3197 (2006).

[33] G. Osnabrugge, S. Leedumrongwatthanakun, and I. M. V. p, Journal of Computational Physics **322**, 113 (2016), ISSN 0021-9991, URL https://www.sciencedirect.com/science/article/pii/S0021999116302595.

[34] G. Osnabrugge, M. Benedictus, and I. M. Vellekoop, Opt. Express **29**, 1649 (2021), URL https://opg.optica.org/oe/abstract.cfm?URI=oe-29-2-1649.

[35] P. C. Chaumet and M. Nieto-Vesperinas, Opt. Lett. **25**, 1065 (2000).

[36] P. C. Chaumet and C. Billaudeau, J. Appl. Phys. **101**, 023106 (2007).