

# “pyGDM” - new functionalities and major improvements to the python toolkit for nano-optics full-field simulations ☆,☆☆



Peter R. Wiecha<sup>a,\*</sup>, Clément Majorel<sup>b</sup>, Arnaud Arbouet<sup>b</sup>, Adelin Patoux<sup>a,b,c</sup>, Yoann Brûlé<sup>d</sup>, Gérard Colas des Francs<sup>d</sup>, Christian Girard<sup>b</sup>

<sup>a</sup> LAAS, Université de Toulouse, CNRS, Toulouse, France

<sup>b</sup> CEMES-CNRS, Université de Toulouse, CNRS, UPS, Toulouse, France

<sup>c</sup> Airbus Defence and Space, Toulouse, France

<sup>d</sup> ICB, UMR 6303 CNRS - Université Bourgogne-Franche Comté, Dijon, France

## ARTICLE INFO

### Article history:

Received 10 May 2021

Received in revised form 30 July 2021

Accepted 14 August 2021

Available online 24 August 2021

### Keywords:

Electrodynamical simulations

Green dyadic method

Coupled dipoles approximation

Nano-optics

Photonic nanostructures

Nano plasmonics

## ABSTRACT

pyGDM is a python toolkit for electro-dynamical simulations of individual nano-structures, based on the Green Dyadic Method (GDM). pyGDM uses the concept of a generalized propagator, which allows to solve cost-efficiently monochromatic problems with a large number of varying illumination conditions such as incident angle scans or focused beam raster-scan simulations. We provide an overview of new features added since the initial publication [Wiecha, *Comput. Phys. Commun.* 233 (2018) 167–192]. The updated version of pyGDM is implemented in pure python, removing the former dependency on fortran-based binaries. In the course of this re-write, the toolkit's internal architecture has been completely redesigned to offer a much wider range of possibilities to the user such as the choice of the dyadic Green's functions describing the environment. A new class of dyads allows to perform 2D simulations of infinitely long nanostructures. While the Green's dyads in pyGDM are based on a quasistatic description for interfaces, we also provide as new external python package “pyGDM2\_retard” a module with retarded Green's tensors for an environment with two interfaces. We have furthermore added functionalities for simulations using fast-electron excitation, namely electron energy loss spectroscopy and cathodoluminescence. Along with several further new tools and improvements, the update includes also the possibility to calculate the magnetic field and the magnetic LDOS inside nanostructures, field-gradients in- and outside a nanoparticle, optical forces or the chirality of nearfields. All new functionalities remain compatible with the evolutionary optimization module of pyGDM for nano-photonics inverse design.

### Program summary

Program Title: pyGDM2

CPC Library link to program files: <https://doi.org/10.17632/5zdrppdc3j.1>

Licensing provisions: GPLv3

Programming language: python

**Nature of problem:** Full-field electrodynamical simulations of photonic nanostructures. This includes calculations of the optical extinction, scattering and absorption, as well as the near-field distribution or the interaction of quantum emitters with nanostructures as well as fast electron beam simulations. The toolkit includes a module for automated evolutionary optimization of nanostructure geometries to obtain a user-defined optical response.

**Solution method:** The optical response of photonic nanostructures is calculated using field susceptibilities (“Green Dyadic Method”, GDM) via a volume discretization. The approach is formally similar to the coupled dipole approximation.

☆ The review of this paper was arranged by Prof. Blum Volker.

☆☆ This paper and its associated computer program are available via the Computer Physics Communications homepage on ScienceDirect (<http://www.sciencedirect.com/science/journal/00104655>).

\* Corresponding author.

E-mail address: [pwiecha@laas.fr](mailto:pwiecha@laas.fr) (P.R. Wiecha).

*Additional comments including restrictions and unusual features:* 2D and 3D nanostructures. On typical office PCs (8–16GB RAM) the discretization is limited to about 10000–15000 meshpoints, therefore it applies best to single, small nanostructures.

© 2021 Elsevier B.V. All rights reserved.

## 1. Introduction

Solving Maxwell's equations [1] for arbitrary nano-structure geometries in complex environments is a key challenge in modern nano-optics. In such nano-optics problems, effects of light-matter interaction can be computed only numerically. Different methods can be used to this end, which have distinct strengths and drawbacks, depending on the respective configuration. To name a few, popular methods are for example the finite difference time domain method (FDTD) [2], the coupled dipole approximation (CDA) [3,4] or the boundary element method (BEM, also called surface integral equation method) [5,6] or the finite element method (FEM) [7,8]. Detailed comparisons between the methods can be found in literature [9–11].

We describe here pyGDM, a python toolkit implementing a volume integral approach based on field-susceptibilities, namely the Green's Dyadic Method (GDM). It is well suited for the numerical description of nano-particles deposited on a substrate. The full optical response of both dielectric [12] or plasmonic nanostructures is accessible [13,14]. The GDM can describe well nano-plasmonics in general [15,16], but in particular it is an almost ideal method to describe "flat plasmonics", hence large planar nanostructures of thicknesses smaller than the skin depth [17–22]. Concerning dielectric materials, low-index dielectric nanostructures can be described very accurately [9], but also high-index dielectric nanostructures are within the scope of the GDM [23–25]. Beyond linear optics it is also capable to describe non-linear effects such as multi-photon photoluminescence or harmonics generation [25–27].

One of the major modifications in the new version is the entirely rewritten internal API. While the initial version was based on fortran routines for a fast computation of Green's tensors, the new pyGDM is fully written in python and accelerated using the *numba* package [28]. In addition to easier installation and better platform independence, the python re-write offers an enormous gain in flexibility and renders pyGDM fully modular, which means that external packages can be easily written and fully integrate in the toolkit without technical barriers. A huge advantage is that the environment is no longer hard-coded with the nano-structure class, but is now fully described through a separate *dyads* class, in which the set of Green's tensors is implemented. This means that the environment can be flexibly modified in the new pyGDM. With the new version, we currently provide an alternative *dyads* class for 2D simulations based on a quasistatic approximation for a layered environment. As an external package we provide another *dyads* class offering a fully retarded description of layered environments with one or two interfaces. In the future we plan to implement further sets of Green's dyads, for instance for periodic structures.

Further new features, among others, are several focused vectorbeams, the generalization of the plane wave incident field, supporting arbitrary illumination angles and polarizations in environments containing up to 3 interfaces. We added the computation of optical chirality, a fast electron illumination for electron energy loss spectroscopy (EELS) and cathodoluminescence (CL) within the new *electron* submodule, or routines for field gradient calculations and optical forces. In the appendix we provide a full description of the Green's tensors used in pyGDM for the different simulation functions. In a further appendix section we also provide more technical documentation as well as instructions for the installation and the use of pyGDM.

## 2. Summary of pyGDM functionalities

We provide here a non-exhaustive list of the functionalities available in the pyGDM toolkit:

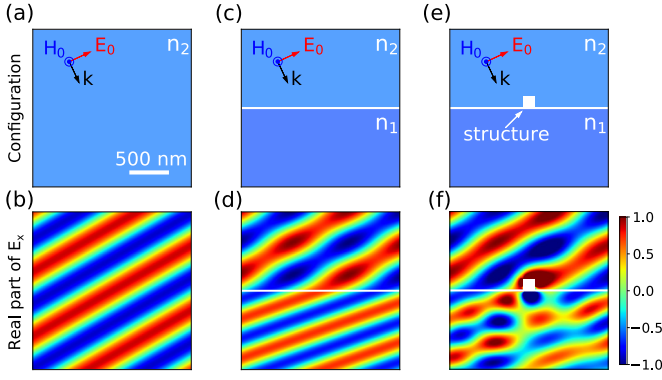
- Easy to use, easy to install, open source. Depends exclusively on open-source python libraries available via [pip](#) (numpy, numba, scipy, matplotlib)
- Fast: Performance-critical parts are accelerated by just-in-time compilation via numba and are parallelized. Efficient and parallelized scipy libraries are used whenever possible. An MPI-parallelized routine for the calculation of spectra is available
- Electro-dynamical simulations including a substrate, either in a computationally fast quasistatic approximation, or computationally slower including retardation
- 2D and 3D simulations
- Various illumination sources such as plane waves, tightly focused beams with linear, radial or azimuthal polarization, dipolar emitters
- Very efficient calculation of large monochromatic problems (raster-scans, incident angle scans, ...)
- Fast electron simulations (EELS, cathodoluminescence)
- Includes various tools to rapidly post-process the simulations and derive physical quantities such as
  - extinction, absorption and scattering cross-sections
  - optical near-field inside and around nanostructures
  - polarization- and spatially resolved far-field scattering
  - optical chirality of the near-field
  - heat generation, temperature distribution around nano-objects
  - photonic local density of states (LDOS)
  - field gradients and optical forces
  - multi-photon photoluminescence
  - electric / magnetic dipole decomposition of local fields and the extinction cross section
- Evolutionary optimization of the nano-particle geometry with regards to specific optical properties
- Easy to use visualization tools including animations of the electro-magnetic fields

## 3. The Green's dyadic method

### 3.1. From Maxwell's equations to Lippmann-Schwinger equation

We will first recapitulate the underlying theory behind the Green's Dyadic Method (GDM) [29]. Compared to the initial pyGDM paper [30], we present here the formalism in a more general way, including the self-consistent equations describing both, electric and magnetic optical fields, since one new feature of the toolkit is the calculation of the magnetic field inside nanostructures.

The GDM allows to treat light-matter interaction problems such as depicted in Fig. 1, where the progressive increase in complexity of the scenario is illustrated by the example of the perturbation of a plane wave. Fig. 1a shows the field amplitude of a plane wave passing through a homogeneous environment, in 1b an interface is added and in 1c an additional small nano-structure. Problems



**Fig. 1.** Different environment configurations, through which an oblique incident plane wave passes. Angle of incidence  $\theta_{in} = 30^\circ$ , wavelength  $\lambda_0 = 710$  nm. (a-b) air ( $n_2 = 1$ ). (c-d) plane wave incident from air ( $n_2 = 1$ ) onto a dielectric substrate ( $n_1 = 1.5$ ). (e-f) same as in (c-d) with a nanostructure deposited on top of the interface (silicon nano-cube of size  $160 \times 160 \times 160$  nm<sup>3</sup>). (a,c,e) illustration of the respective simulated configuration. (b,d,f) color plot of the real part of  $E_x$ . Panels show  $2 \times 2 \mu\text{m}^2$  large areas. In contrast to most other electrodynamics solvers, in (e-f) only the silicon cube needs to be discretized.

such as the one shown in Fig. 1 require to solve Maxwell's equations, the GDM does this for monochromatic fields (fixed frequency  $\omega$ ). In cgs units (centimeter, gram, second), the frequency domain Maxwell's equations are

$$\nabla \times \mathbf{E}(\mathbf{r}, \omega) = ik_0[\mu_{\text{env}}(\omega)\mathbf{H}(\mathbf{r}, \omega) + 4\pi\mathbf{M}(\mathbf{r}, \omega)] \quad (1)$$

$$\nabla \cdot \mathbf{H}(\mathbf{r}, \omega) = -\frac{4\pi}{\mu_{\text{env}}(\omega)} \nabla \cdot \mathbf{M}(\mathbf{r}, \omega) \quad (2)$$

$$\nabla \times \mathbf{H}(\mathbf{r}, \omega) = -ik_0[\epsilon_{\text{env}}(\omega)\mathbf{E}(\mathbf{r}, \omega) + 4\pi\mathbf{P}(\mathbf{r}, \omega)] \quad (3)$$

$$\nabla \cdot \mathbf{E}(\mathbf{r}, \omega) = -\frac{4\pi}{\epsilon_{\text{env}}(\omega)} \nabla \cdot \mathbf{P}(\mathbf{r}, \omega) \quad (4)$$

where  $\mathbf{E}(\mathbf{r}, \omega)$  (respectively  $\mathbf{H}(\mathbf{r}, \omega)$ ) is the electric (respectively magnetic) field at the position  $\mathbf{r}$ .  $\mathbf{P}(\mathbf{r}, \omega)$  and  $\mathbf{M}(\mathbf{r}, \omega)$  are the electric and magnetic polarizations,  $\epsilon_{\text{env}}$  and  $\mu_{\text{env}}$  are respectively the dielectric constant and the permeability of the surrounding medium and  $k_0 = \omega/c$  is the wavenumber of the light in vacuum.

By applying the rotational operator to the Maxwell-Faraday (Eq. (1)) and Maxwell-Ampere (Eq. (3)) equations and by reorganizing the obtained equations, we can define two wave-equations [31,32]. One for  $\mathbf{E}(\mathbf{r}, \omega)$  and one for  $\mathbf{H}(\mathbf{r}, \omega)$ :

$$(\Delta + k^2)\mathbf{E}(\mathbf{r}, \omega) = \quad (5a)$$

$$-4\pi \left[ \frac{1}{\epsilon_{\text{env}}} (\nabla \nabla + k^2) \mathbf{P}(\mathbf{r}, \omega) + ik_0 \nabla \times \mathbf{M}(\mathbf{r}, \omega) \right]$$

$$(\Delta + k^2)\mathbf{H}(\mathbf{r}, \omega) = \quad (5b)$$

$$-4\pi \left[ \frac{1}{\mu_{\text{env}}} (\nabla \nabla + k^2) \mathbf{M}(\mathbf{r}, \omega) + ik_0 \nabla \times \mathbf{P}(\mathbf{r}, \omega) \right]$$

with  $k = \sqrt{\epsilon_{\text{env}} \mu_{\text{env}}} k_0$  the wavenumber of light in the environment medium. In the approximation of a local and linear optical response, the electric and magnetic polarizations are proportional to the electric and magnetic field as  $\mathbf{P} = \chi_e \cdot \mathbf{E}$  and  $\mathbf{M} = \chi_m \cdot \mathbf{H}$ . The electric (respectively magnetic) susceptibility is a tensor of rank 2, corresponding to the difference between the relative permittivity (respectively permeability) of the structure and the environment  $\chi_e(\mathbf{r}, \omega) = (\epsilon_r - \epsilon_{\text{env}})/4\pi$  (respectively  $\chi_m(\mathbf{r}, \omega) = (\mu_r - \mu_{\text{env}})/4\pi$ ).

If we replace  $\mathbf{P}$  and  $\mathbf{M}$  by their expressions in Eq. (5a) and Eq. (5b), we obtain a system of coupled equations relating  $\mathbf{E}(\mathbf{r}, \omega)$  and  $\mathbf{H}(\mathbf{r}, \omega)$ . To lighten the notation, we define the super vector  $\mathbf{F}(\mathbf{r}, \omega) = \{\mathbf{E}(\mathbf{r}, \omega), \mathbf{H}(\mathbf{r}, \omega)\}$ . We can then write the solution of

Eq. (5), as a single vectorial Lippman-Schwinger equation [33,34,4,35]

$$\mathbf{F}(\mathbf{r}, \omega) = \mathbf{F}_0(\mathbf{r}, \omega) + \int_V \mathbf{G}(\mathbf{r}, \mathbf{r}', \omega) \cdot \chi(\mathbf{r}', \omega) \cdot \mathbf{F}(\mathbf{r}', \omega) d\mathbf{r}'. \quad (6)$$

This equation establishes a self-consistent relation between the local fields  $\mathbf{F}(\mathbf{r}, \omega)$  and the illumination fields  $\mathbf{F}_0(\mathbf{r}, \omega)$  at the position  $\mathbf{r}$  inside the structure. The integral is performed over the nanostructure volume  $V$ .  $\chi(\mathbf{r}, \omega)$  is a  $(6 \times 6)$  tensor comprising the electric and magnetic susceptibilities:

$$\chi(\mathbf{r}, \omega) = \begin{pmatrix} \chi_e(\mathbf{r}, \omega) & 0 \\ 0 & \chi_m(\mathbf{r}, \omega) \end{pmatrix}. \quad (7)$$

The  $(6 \times 6)$  “super-propagator”  $\mathbf{G}$  describes light propagation in the environment:

$$\mathbf{G}(\mathbf{r}, \mathbf{r}', \omega) = \begin{pmatrix} \mathbf{G}_{\text{tot}}^{\text{EE}}(\mathbf{r}, \mathbf{r}', \omega) & \mathbf{G}_{\text{tot}}^{\text{EH}}(\mathbf{r}, \mathbf{r}', \omega) \\ \mathbf{G}_{\text{tot}}^{\text{HE}}(\mathbf{r}, \mathbf{r}', \omega) & \mathbf{G}_{\text{tot}}^{\text{HH}}(\mathbf{r}, \mathbf{r}', \omega) \end{pmatrix}. \quad (8)$$

It is composed of four Green's dyads (called field susceptibilities in the context of electrodynamics). The two “direct” field susceptibilities  $\mathbf{G}_{\text{tot}}^{\text{EE}}(\mathbf{r}, \mathbf{r}', \omega)$  and  $\mathbf{G}_{\text{tot}}^{\text{HH}}(\mathbf{r}, \mathbf{r}', \omega)$  describe, respectively, electric-electric and magnetic-magnetic coupling. The terms  $\mathbf{G}_{\text{tot}}^{\text{EH}}(\mathbf{r}, \mathbf{r}', \omega)$  and  $\mathbf{G}_{\text{tot}}^{\text{HE}}(\mathbf{r}, \mathbf{r}', \omega)$  on the other hand are the mixed field susceptibilities which describe coupling between the electric and magnetic fields.

Since all materials in nature interact only marginally with the optical magnetic field, the implementation in pyGDM is so far limited to treat only non-magnetic media, hence  $\chi_e \neq 0$ ,  $\chi_m = 0$ . In consequence, equation (6) involves only the field susceptibilities  $\mathbf{G}_{\text{tot}}^{\text{EE}}(\mathbf{r}, \mathbf{r}', \omega)$  and  $\mathbf{G}_{\text{tot}}^{\text{HE}}(\mathbf{r}, \mathbf{r}', \omega)$ . The local electric and magnetic fields are then defined by the two following Lippmann-Schwinger equations

$$\begin{aligned} \mathbf{E}(\mathbf{r}, \omega) &= \mathbf{E}_0(\mathbf{r}, \omega) + \int \mathbf{G}_{\text{tot}}^{\text{EE}}(\mathbf{r}, \mathbf{r}', \omega) \cdot \chi_e(\mathbf{r}', \omega) \cdot \mathbf{E}(\mathbf{r}', \omega) d\mathbf{r}' \\ \mathbf{H}(\mathbf{r}, \omega) &= \mathbf{H}_0(\mathbf{r}, \omega) + \int \mathbf{G}_{\text{tot}}^{\text{HE}}(\mathbf{r}, \mathbf{r}', \omega) \cdot \chi_e(\mathbf{r}', \omega) \cdot \mathbf{E}(\mathbf{r}', \omega) d\mathbf{r}' \end{aligned} \quad (9)$$

By default, pyGDM uses a Green's tensor  $\mathbf{G}_{\text{tot}}^{\text{EE}} = \mathbf{G}_0^{\text{EE}} + \mathbf{G}_{\text{3-layer}}$ , composed of a field susceptibility for the homogeneous medium  $\mathbf{G}_0^{\text{EE}}$  and a field susceptibility  $\mathbf{G}_{\text{3-layer}}$  associated with a substrate and/or cladding layer [30] (for computational efficiency, the latter is based on a quasi-static approximation). Setting  $\mu = 1$  everywhere,  $\mathbf{G}_{\text{tot}}^{\text{HE}} = \mathbf{G}_0^{\text{HE}}$  (no contribution from the layered environment), which writes explicitly [36]

$$\mathbf{G}_0^{\text{HE}}(\mathbf{r}, \mathbf{r}', \omega) = -ik_0 \nabla \times G_0(\mathbf{r}, \mathbf{r}', \omega) \quad (10)$$

with  $G_0(\mathbf{r}, \mathbf{r}', \omega) = e^{ik|\mathbf{r}-\mathbf{r}'|}/|\mathbf{r}-\mathbf{r}'|$  being the scalar Green's function.

#### 4. Numerical resolution of the Lippmann-Schwinger equation – volume discretization

In the following we will briefly describe how the above theory can be numerically solved for structures of arbitrary shape. In such case, the set of equations (9) can in general not be solved analytically and needs to be approached with numerical methods. Here, we solve Eqs. (9) via a volume discretization of the structure on a regular mesh of  $N$  unit cells centered at positions  $\mathbf{r}_i$ . Consequently, the integrals become discrete sums over the ensemble of

mesh-cells and the differential term  $d\mathbf{r}'$  is replaced by the volume of the unit cell  $V_{\text{cell}}$ . Note that we omit the dependency on  $\omega$  in the following for better readability.

$$\begin{aligned} \mathbf{E}(\mathbf{r}_i) &= \mathbf{E}_0(\mathbf{r}_i) \\ &+ \sum_{j=1}^N \mathbf{G}_{\text{tot}}^{\text{EE}}(\mathbf{r}_i, \mathbf{r}_j) \cdot \chi_e(\mathbf{r}_j) \cdot \mathbf{E}(\mathbf{r}_j) V_{\text{cell}} \\ \mathbf{H}(\mathbf{r}_i) &= \mathbf{H}_0(\mathbf{r}_i) \\ &+ \sum_{j=1}^N \mathbf{G}_{\text{tot}}^{\text{HE}}(\mathbf{r}_i, \mathbf{r}_j) \cdot \chi_e(\mathbf{r}_j) \cdot \mathbf{E}(\mathbf{r}_j) V_{\text{cell}}. \end{aligned} \quad (11)$$

It is worth emphasizing that the incident electromagnetic field ( $\mathbf{E}_0$ ,  $\mathbf{H}_0$ ) can be e.g. a plane wave, a highly focused beam with linear, radial or azimuthal polarization, as well as a dipolar field. A key advantage of the GDM is that different excitation conditions can be considered by computationally highly efficient post-processing, once the Green's dyadic tensor (also "field susceptibility tensor") of the full structure has been determined.

We now define  $6N$ -dimensional super-vectors  $\mathbf{F}_{\text{obj}}$  and  $\mathbf{F}_{0,\text{obj}}$ , which are composed of the electromagnetic fields at each unit cell's position  $\mathbf{r}_i$  as

$$\begin{aligned} \mathbf{F}_{0,\text{obj}} &= (\mathbf{E}_0(\mathbf{r}_1), \mathbf{E}_0(\mathbf{r}_2), \dots, \dots, \mathbf{E}_0(\mathbf{r}_N), \\ &\quad \mathbf{H}_0(\mathbf{r}_1), \mathbf{H}_0(\mathbf{r}_2), \dots, \dots, \mathbf{H}_0(\mathbf{r}_N)) \\ \mathbf{F}_{\text{obj}} &= (\mathbf{E}(\mathbf{r}_1), \mathbf{E}(\mathbf{r}_2), \dots, \dots, \mathbf{E}(\mathbf{r}_N), \\ &\quad \mathbf{H}(\mathbf{r}_1), \mathbf{H}(\mathbf{r}_2), \dots, \dots, \mathbf{H}(\mathbf{r}_N)). \end{aligned}$$

Using those super-vectors, the system of equations defined by Eq. (11) can be written in matrix form:

$$\mathbf{F}_{0,\text{obj}} = \begin{pmatrix} \mathbf{M}^{\text{EE}} & \mathbf{0} \\ \mathbf{M}^{\text{HE}} & \mathbf{I}_{3N} \end{pmatrix} \cdot \mathbf{F}_{\text{obj}} = \mathbf{M} \cdot \mathbf{F}_{\text{obj}} \quad (12)$$

with the identity matrix  $\mathbf{I}_{3N}$  of size  $3N$ .  $\mathbf{M}^{\text{EE}}$  and  $\mathbf{M}^{\text{HE}}$  are  $(3N \times 3N)$  matrices composed of  $N$  ( $3 \times 3$ ) matrices defined by equation (11):

$$\begin{aligned} \mathbf{M}_{ij}^{\text{EE}}(\mathbf{r}_i, \mathbf{r}_j) &= \mathbf{I}_{\delta_{ij}} - \sum_{j=1}^N \mathbf{G}_{\text{tot}}^{\text{EE}}(\mathbf{r}_i, \mathbf{r}_j) \cdot \chi_e(\mathbf{r}_j) \cdot V_{\text{cell}} \\ \mathbf{M}_{ij}^{\text{HE}}(\mathbf{r}_i, \mathbf{r}_j) &= - \sum_{j=1}^N \mathbf{G}_{\text{tot}}^{\text{HE}}(\mathbf{r}_i, \mathbf{r}_j) \cdot \chi_e(\mathbf{r}_j) \cdot V_{\text{cell}}. \end{aligned} \quad (13)$$

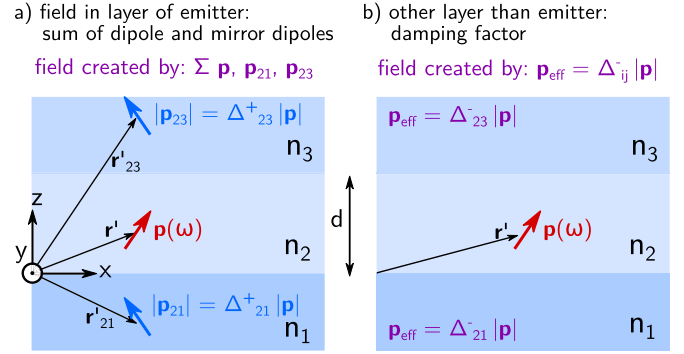
$\mathbf{I}$  is the unitary tensor. Now, by inverting the matrix  $\mathbf{M}$ , we can deduce the local electromagnetic field  $\mathbf{F}_{\text{obj}}$  inside the structure from the incident fields  $\mathbf{F}_{0,\text{obj}}$  from the inverse  $\mathbf{K} = \mathbf{M}^{-1}$  via a single matrix-vector product:

$$\mathbf{F}_{\text{obj}} = \mathbf{K} \cdot \mathbf{F}_{0,\text{obj}}. \quad (14)$$

pyGDM supports conventional CPU based inversion algorithms as well as LU decomposition, both through `scipy`. A new functionality is graphics processing unit (GPU) accelerated inversion with CUDA compatible GPUs through the optional `cupy` package. On high-end GPUs, up to 10-fold speed-up can be reached.

#### 4.1. Interfaces: near-field, far-field and retardation

The Lippmann-Schwinger equation (6) allows to determine the electromagnetic field everywhere in space. Indeed, as mentioned previously, we determine the electromagnetic field inside



**Fig. 2.** Illustration of the image dipole approximation. Schematic representation of an electric dipole  $\mathbf{p}(\omega)$  (red) at the position  $\mathbf{r}'$  in a layer of refractive index  $n_2$  and thickness  $d$ . The layer is sandwiched between two semi-infinite layers of indices  $n_1$  (below) and  $n_3$  (above). a)  $\mathbf{p}_{21}(\omega)$  and  $\mathbf{p}_{23}(\omega)$  are the image dipoles in the lower and upper layers, used to approximate the field inside the layer of the emitting physical dipole. Their orientation is along  $\mathbf{p}_{21} = \mathbf{p}_{23} = (-p_x, -p_y, +p_z)$ , their  $z$ -distance to  $\mathbf{p}$  is twice the distance between the physical dipole and the respective interface. b) in order to calculate the fields in the layers where no physical emitter is present, no mirror dipoles are required. The field corresponds to the original dipole's field, attenuated by a screening factor  $\Delta^-$ . (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

the structure by solving self-consistently the equation (6). Then, to determine the electric field outside the structure, we reuse the same equation to repropagate the local internal field. The super-propagator  $\mathbf{G}$  introduced in the last section (Eq. (8)) contains the four field susceptibilities ( $\mathbf{G}_{\text{tot}}^{\text{EE}}(\mathbf{r}, \mathbf{r}', \omega)$ ,  $\mathbf{G}_{\text{tot}}^{\text{EH}}(\mathbf{r}, \mathbf{r}', \omega)$ ,  $\mathbf{G}_{\text{tot}}^{\text{HE}}(\mathbf{r}, \mathbf{r}', \omega)$ ,  $\mathbf{G}_{\text{tot}}^{\text{HH}}(\mathbf{r}, \mathbf{r}', \omega)$ ) describing the propagation of light emitted by an electric or a magnetic dipole.

For a dipole in the vicinity of a plane interface these susceptibilities can be written as a sum of a free space contribution  $\mathbf{G}_0^{\alpha\beta}(\mathbf{r}, \mathbf{r}', \omega)$  and a surface dyad  $\mathbf{G}_{2\text{-layer}}^{\alpha\beta}(\mathbf{r}, \mathbf{r}', \omega)$ . The superscript index  $\alpha$  represents the nature of the induced field at  $\mathbf{r}$  (electric "E" or magnetic "H") and  $\beta$  represents the nature of the dipole at position  $\mathbf{r}'$ . The free space susceptibilities, solutions of the wave-equations Eq. (5), are proportional to the scalar Green's function  $G_0(\mathbf{r}, \mathbf{r}', \omega)$  (see appendix Eq. (A.1)). These susceptibilities can be decomposed in a sum of dyadic tensors representing the near field, intermediate and far field contributions (Eq. (A.2)) [37,38].

For a single plane interface, the susceptibilities  $\mathbf{G}_{2\text{-layer}}^{\alpha\beta}(\mathbf{r}, \mathbf{r}', \omega)$ , associated with the surface, can be obtained by expanding the electric and magnetic fields using plane waves while applying the boundary conditions for the tangential component of the field at the interface [31,37,39]. We summarize the main results of the calculation in Appendix A.2.

##### 4.1.1. Near-field in the electrostatic region: mirror-charge approximation

In case of a low index dielectric layer or if the emitted field is evaluated at distances to the dipole, significantly smaller than the wavelength (i.e. in the electrostatic limit with no retardation), we can use an approximation based on the method of mirror charges to describe the effect of the interface. As illustrated in Fig. 2a for the presence of two interfaces, for an evaluation inside the layer of the emitter dipole a virtual dipole is added to the configuration, with an amplitude vector of opposite sign and which is placed at identical distance behind the interface. This ensures the continuity of the electromagnetic field at the interface. Inside the empty medium, the emitted field corresponds to the single dipole, however attenuated by a screening factor (see Fig. 2). This leads to the expressions (A.9) and (A.10) for the electric-electric contribution. The magnetic-electric susceptibility of the interface is zero (Eq. (A.11)). The mirror dipole method provides a good approxima-



tion for the contribution of the interface if retardation effects are negligible [40].

In pyGDM the environment can consist of at most three layers of refractive indexes  $n_1$ ,  $n_2$  and  $n_3$  for the bottom, center and top layer, respectively. The lower and upper layers in Fig. 2 are semi-infinite and the intermediate layer has a thickness  $d$  (also called “spacing”). In the shown example with an electric dipole  $\mathbf{p}(\omega)$  in the intermediate layer, the two mirror dipoles are represented by the blue arrows  $\mathbf{p}_{21}$  and  $\mathbf{p}_{23}$ .

#### 4.1.2. Asymptotic far-field approximation

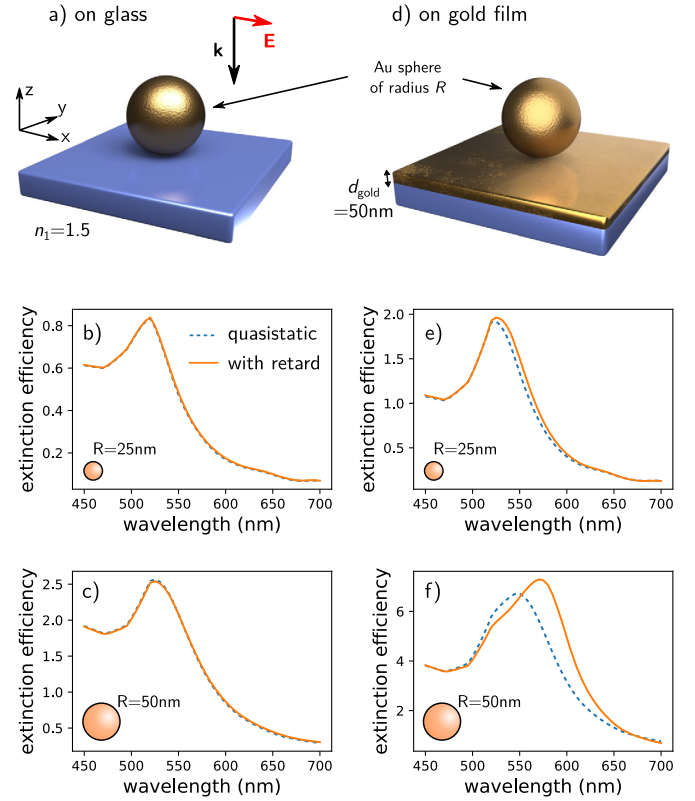
Conversely if the evaluation position is at a large distance from the emitting dipole, we are in the far-field region. In this case the surface contributions above and below the interface have also explicit analytical expressions. They can be obtained using the asymptotic limit, defined in equation (A.7). For the electric-electric surface term  $\mathbf{G}_{2\text{-layer}}^{\text{EE}}(\mathbf{r}, \mathbf{r}', \omega)$ , the analytical asymptotic expressions implemented in pyGDM are (A.12) and (A.15). In the far-field the electromagnetic wave is transverse, therefore it is possible to deduce the magnetic field  $\mathbf{H}$  via the cross product of  $\mathbf{k}$  and  $\mathbf{E}$ .

#### 4.1.3. Intermediate region - retardation

If the field is to be evaluated at intermediate distances to the dipole (roughly at distances of less than a few wavelengths), no general approximation exists and the integral in Eq. (A.5) must be performed numerically [41], requiring integration in the complex plane to avoid the singularities on the real axis. This can be interpreted as an integration over propagating (real valued wavevectors) and evanescent modes (complex valued wavevectors) [42]. Reference [43] provides a detailed discussion about how to choose the best integration path in the complex plane, and extends the procedure to layered media with arbitrary numbers of interfaces. Due to the necessity of the complex integration, the evaluation of the retarded Green's tensors is computationally relatively expensive. It is usually necessary to include retardation effects when considering large structures deposited on high index dielectric, thin film or plasmonic substrates, in order to take into account the effect of, for example, propagating modes or surface plasmons.

A comparison between the mirror dipole approximation and a fully retarded description of the substrate is given in Fig. 3. Panels 3a-c show simulation results of a gold nanosphere lying on a glass substrate ( $n_1 = 1.5$ ), illuminated by a plane wave polarized along X. A small nanosphere ( $R = 25$  nm in Fig. 2b) can be very accurately described in the quasistatic approximation, and even for larger particles (Fig. 2c,  $R = 50$  nm), only small deviations occur in comparison with the retarded simulation. Fig. 3d-f show simulations of an identical gold sphere as in 3a-c, but now lying on a thin gold film ( $d_{\text{gold}} = 50$  nm). Even in this case, small particles ( $R = 25$  nm, Fig. 3e) can be described accurately using quasistatic Green's dyads. But when the radius of the sphere is increased ( $R = 50$  nm, Fig. 3f), significant deviations occur between the non-retarded and retarded descriptions. For a script to reproduce the results in Fig. 3 see [this link](#).

Because of poor performance of pure python code and to avoid a dependency of pyGDM on external binaries, the retarded Green's tensors are not implemented in the main pyGDM module. However, we provide a separate python module based on a fortran implementation of the retarded Green's tensors for the 2-interface environment, used in pyGDM. Please note that this extension of pyGDM is still in a development version, and not all aspects have been implemented yet. The “pyGDM2\_retard” module requires compilation of the fortran part. Via the python package index *pypi*, we provide pre-compiled binaries for windows, on other platforms the module needs to be compiled from the source code (see Appendix D.3). For technical details on the numerical calculation of the retarded Green's tensors, we refer the interested reader to



**Fig. 3.** Comparison between quasistatic mirror-dipole approximation (dashed blue lines) and Green's dyads with retardation (solid orange lines) to describe the substrate. (a-c) Extinction simulations of a gold nanosphere on a semi-infinite dielectric substrate ( $n_1 = 1.5$ ) as depicted in (a). The sphere is illuminated by a plane wave from the top, with linear polarization along X. (b) extinction efficiency for a gold sphere of radius  $R = 25$  nm and (c) for  $R = 50$  nm. (d-f) same as (a-c) but the substrate is composed of an additional  $d_{\text{gold}} = 50$  nm thin gold layer on top of the dielectric substrate.

Ref. [43]. We note that the “pyGDM2\_retard” package requires the nanostructure to be placed in the top-most layer (layer “3” in Fig. 2). Moreover, field calculations are only possible in the layer of the nanostructure. We plan to extend the retarded module in the future.

If for reasons of computation speed, the mirror dipole approximation is used to take into account a layered environment, the above described limitations should be kept in mind.

#### 4.2. Mesh-type, self-terms

We will recall here briefly the meshing in pyGDM. For details, please refer to Ref. [30]. pyGDM can currently handle discretizations on regular cubic and hexagonal compact meshes in 3D and on square meshes in 2D. The divergence of the Green's function at  $\mathbf{r}_i = \mathbf{r}_j$  needs to be accounted for by analytically evaluating the so-called “self-term” at that location, which depends on the geometry of the mesh-cells.

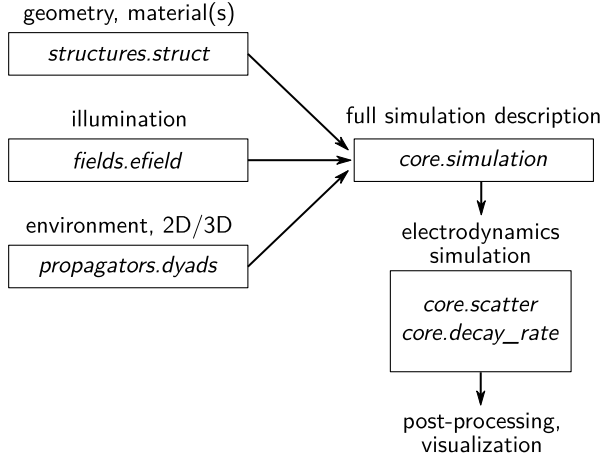
For the cubic, respectively hexagonal mesh, in 3D, we find (see [17], section 3.1)

$$\mathbf{G}_{0,\text{cube}}^{\text{EE}}(\mathbf{r}_i, \mathbf{r}_i) = -\frac{4\pi}{3\epsilon_{\text{env}}d^3} \mathbf{I} \quad (15)$$

$$\mathbf{G}_{0,\text{hex}}^{\text{EE}}(\mathbf{r}_i, \mathbf{r}_i) = -\frac{4\pi\sqrt{2}}{3\epsilon_{\text{env}}d^3} \mathbf{I}. \quad (16)$$

For the self-terms of a 2D square mesh, see Appendix B.

Self-terms for more complex mesh-cell geometries like cuboids or tetrahedrons can be calculated numerically [44,20,45]. This is



**Fig. 4.** Structure of a simulation in pyGDM and the typical workflow. The geometry and material(s) of the nano-object are defined in the *struct*-class, the illumination field by the *efield*-class and the simulation environment (environment refractive index, substrate, 2D/3D, ...) is configured in a *dyads* object. The *simulation* class bundles all information and serves also as a container for the simulation results, which is then used for further evaluation and post-processing or visualizations.

currently not implemented but might be added in a future version of pyGDM.

We note here that the structure needs to be discretized with a sufficiently small step to resolve all fields and their gradients inside the nanostructure. For best convergence the step should be in the order of  $d \lesssim \lambda_0/10n_{\text{struct}}$ , but significantly finer meshes may be required for certain configurations, for instance in cases of strong field gradients.

#### 4.3. Outlook: magnetic media

As mentioned before, the current version of pyGDM assumes media with no direct magnetic response ( $\chi_m = 0$ ), but it might be implemented in a future version of pyGDM. It is technically straightforward to include the  $\chi_m \neq 0$  terms in the calculation. While natural materials don't interact with the optical magnetic field, this may be interesting to model meta-materials with a macroscopic effective magnetic response [46]. Every mesh-cell would correspond to a meta-atom of the meta-material. In such case the upper and lower right sub-matrices in  $\mathbf{M}$  would be filled with *electric-magnetic* and *magnetic-magnetic* coupling terms  $\mathbf{M}_{ij}^{\text{EH}}$  and  $\mathbf{M}_{ij}^{\text{HH}}$  [4].

#### 4.4. pyGDM simulation description

A pyGDM simulation is organized as depicted in Fig. 4. A first class describes the geometry and material(s) of the nanostructure (class *struct*). A second class describes the incident field (class *efield*), and a third one the environment and simulation type (new class *dyads*). These three objects are assembled in a *simulation* class, which also serves as a container to store the results of the GDM simulation. The simulation object is also used by all further post-processing and visualization functions provided by pyGDM.

### 5. New functionalities

#### 5.1. Decomposition in electric and magnetic dipole modes

We have implemented in the updated version the decomposition of the internal field in electric and magnetic multipoles, following the work of Evlyukhin et al. [47]. By default, the new

functions `linear.multipole_decomp` and `linear.multipole_decomp_extinct` return only the electric dipole (ED) and magnetic dipole (MD) contributions, those are stable and extensively tested. The quadrupoles can be returned by passing the optional keyword argument `quadrupoles=True`, but those are still experimental and have not been sufficiently tested. The effective dipole moments  $\mathbf{p}_{\text{tot}}$  (ED) and  $\mathbf{m}_{\text{tot}}$  (MD) are obtained from the internal field distribution as

$$\mathbf{p}_{\text{tot}} = \sum_i \mathbf{p}_i = \sum_i V_{\text{cell}} \chi_e(\mathbf{r}_i) \cdot \mathbf{E}(\mathbf{r}_i) \quad (17)$$

and

$$\mathbf{m}_{\text{tot}} = -\frac{ik_0}{2} \sum_i (\mathbf{r}_i - \mathbf{r}_0) \times \mathbf{p}_i. \quad (18)$$

We note that the choice of the position  $\mathbf{r}_0$  of the total dipole moment is crucial in the calculation of the MD moment. By default, pyGDM uses the center of mass of the nanostructure, but  $\mathbf{r}_0$  can be specified manually using the optional keyword argument “*r0*”.

The respective contributions to the extinction cross section can be calculated as [47]

$$\begin{aligned} \sigma_{\text{ext,ED}} &= \frac{4\pi k_0}{|\mathbf{E}_0|^2 n_{\text{env}}} \text{Im} \left( \sum \mathbf{E}_0^* \cdot \mathbf{p}_{\text{tot}} \right) \\ \sigma_{\text{ext,MD}} &= \frac{4\pi k_0}{|\mathbf{E}_0|^2 n_{\text{env}}} \text{Im} \left( \sum \mathbf{H}_0^* \cdot \mathbf{m}_{\text{tot}} \right), \end{aligned} \quad (19)$$

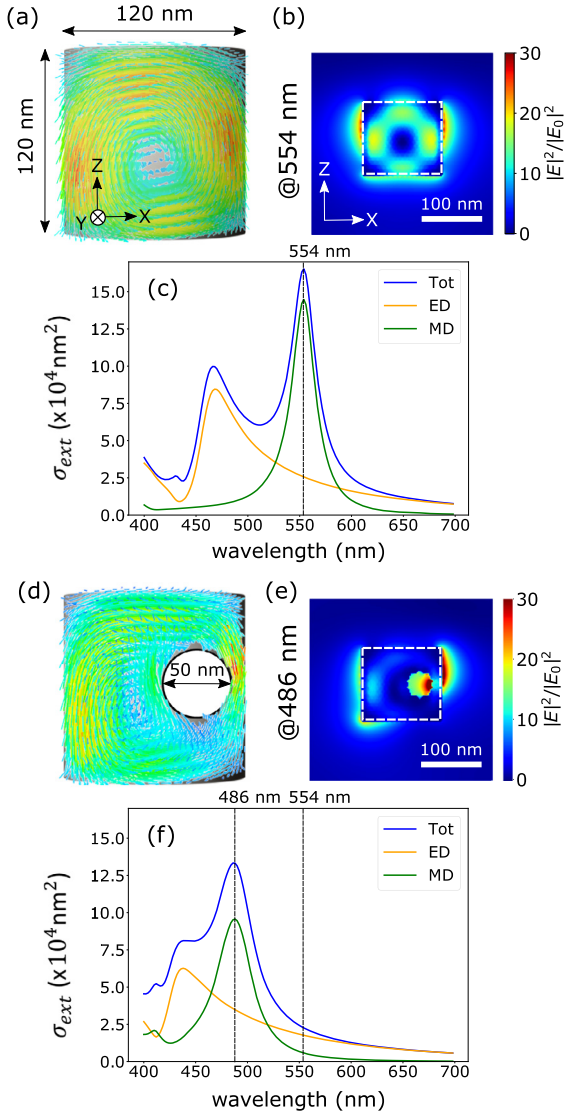
where  $n_{\text{env}}$  is the refractive index of the environment.

We demonstrate the decomposition of the extinction cross section on a silicon nano-disc, as illustrated in Fig. 5. Fig. 5a-c show the case of a full cylinder in vacuum, illuminated from the top by a plane wave of linear polarization along X. The internal field vectors and the field intensity in and around the nano-disc are shown in Fig. 5a-b for the MD resonance at  $\lambda_0 = 554$  nm, which can be easily identified in the extinction spectra shown in Fig. 5c. The electric field vectors inside the nano-disc form a vortex in the XZ plane, which induces an effective magnetic dipole moment along Y. We now cut a hole in the silicon, through the vortex, as illustrated in Fig. 5d (hole at the right side of the disc, parallel to Y, with a diameter of  $D_{\text{hole}} = 50$  nm). We see that the vortex at the magnetic resonance wavelength is now strongly perturbed (see in particular Fig. 5e). This results in a significant reduction of the magnetic dipole contribution to the extinction, as can be seen in the spectra of Fig. 5f. Simultaneously, an electric near-field hotspot is induced around the hole. The shift of the resonances to shorter wavelengths can be attributed to the reduced amount of material in comparison with the full disc.

The pyGDM-script to reproduce the results shown in Fig. 5 can be found [in the online documentation at this link](#).

#### 5.2. Magnetic field inside the nanostructure

Calculating only the electric response can be done by inversion of  $\mathbf{M}^{\text{EE}}$  in equation (13). However, if the magnetic field inside a nanostructure needs to be known, we require the inverse of matrix  $\mathbf{M}$  in equation (12). Standard matrix inversion techniques scale with the third power. In consequence, doubling the size of the matrix means the inversion takes 8 times longer (while the matrix occupies four times the memory). Fortunately, a block-matrix can be efficiently inverted block-wise using the Hans Boltz method [48]. In our case, assuming non-magnetic media ( $\chi_m = 0$ ), we obtain the inverse of  $\mathbf{M}$  in equation (12) as



**Fig. 5.** Influence of a hole inside a silicon nanodisc on the electric and magnetic dipole resonances. (a-c) Full nanodisc of diameter  $D = 120$  nm and height  $H = 120$  nm. (a) Field vectors inside (real part) and (b) near-field intensity inside and around the disc at the magnetic resonance position  $\lambda_0 = 554$  nm. (c) Total extinction cross section (blue line) as well as the electric (orange line) and magnetic (green line) dipole contributions. The illumination is a plane wave traveling along  $Z$ , polarized along  $X$ , the environment is vacuum ( $n_{\text{env}} = 1$ ). (d-f) The same nanodisc but with a hole of diameter  $D_{\text{hole}} = 50$  nm along the  $(Oy)$  direction, cutting through the field vortex which induces the magnetic dipole moment. (d-e) show the near-field as in (a-b), here at the resonance wavelength  $\lambda_0 = 486$  nm. (f) extinction cross sections similar to (c).

$$\mathbf{K} = \begin{pmatrix} \mathbf{K}^{\text{EE}} & 0 \\ -\mathbf{M}^{\text{HE}} \cdot \mathbf{K}^{\text{EE}} & \mathbf{I}_{3N} \end{pmatrix} \quad (20)$$

with  $\mathbf{K}^{\text{EE}}$  the *electric-electric* generalized field propagator [29]. Hence, the  $6N \times 6N$  problem reduces to the inversion of a  $3N \times 3N$  matrix and an additional product of two matrices of the same size. It is therefore of comparable computational cost as the resolution of the electric-electric problem alone [30]. However, calculating the internal magnetic field in addition to the electric field requires around twice as much memory, in order to store the additional  $3N \times 3N$  block at the lower left of Eq. (20). Therefore the H-field calculation is by default deactivated in pyGDM but can be enabled by passing `calc_H=True` as parameter to the `core.scatter` function.

As an example we reproduce in Fig. 6 the internal magnetic field in a silicon disc and in silicon disc-dimers as reported earlier by Wang et al. [49]. The silicon cylinders are placed on a silica substrate ( $n_1 = 1.45$ ) and are embedded in air ( $n_2 = 1$ ). The silicon cylinders have a diameter  $D = 140$  nm and a height  $H = 160$  nm. In case of the disc-dimer both cylinders are separated by a gap of  $G = 30$  nm. We discretize the cylinders on a hexagonal compact lattice (with  $N = 1490$  mesh cells). We use plane wave illumination at normal incidence from inside the substrate, polarized along the  $Ox$ -axis (Fig. 6a, 6d and 6b, 6e) or along the  $Oy$ -axis (Fig. 6c and 6f).

Fig. 6a-c show the total extinction curves (blue spectra), and additionally the contributions due to the effective electric and magnetic dipole moments (orange and green curves, respectively). This allows us to determine where the structures have electric dipole (ED) and magnetic dipole (MD) resonances.

While the  $X$ -polarized dimer has a totally different optical response, the dimer illuminated by  $Y$ -polarized light yields similar optical spectra as the isolated cylinder. The reason is that under  $Y$ -polarization, the electric field is perpendicular to the axis of the dimer, which reduces coupling. At the respective electric and magnetic dipole resonances we computed the normalized amplitudes of the electric and magnetic fields, shown in the bottom subplots of Fig. 6. The spectra and field amplitude maps are in good agreement with results obtained by Finite Difference Time Domain (FDTD) simulations [50,51,49].

The pyGDM-script to reproduce the results shown in Fig. 6 can be found in the online documentation under this link.

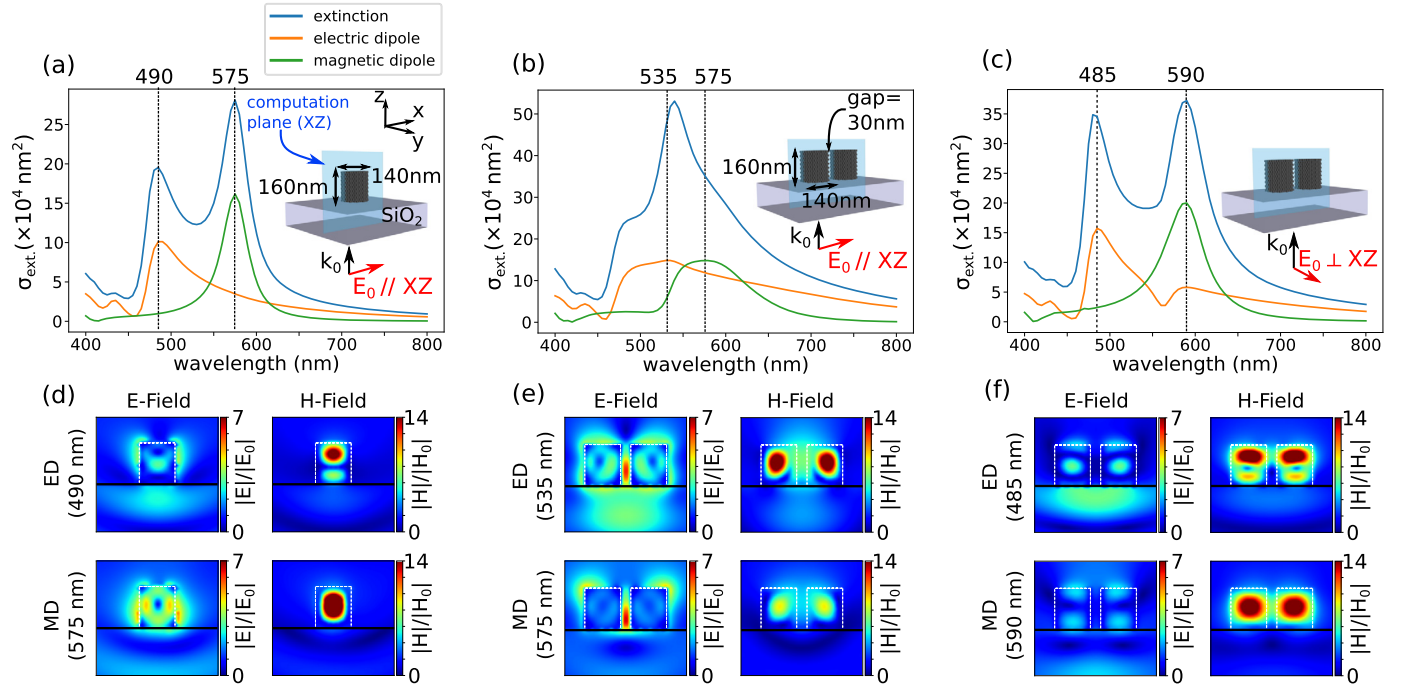
### 5.3. 2D simulations

pyGDM was entirely rewritten in pure python. In this process, the definitions of the Green's dyads used to describe the environment was implemented in a new class. This allows to flexibly use different sets of field susceptibilities. So far this comprises the original set of tensors for 3D simulations in an environment with up to 3 layers in the quasistatic near-field approximation. New in pyGDM is now a class providing field susceptibilities for 2D simulations, hence with one axis being infinitely long. In this case the nanostructure is discretized on a 2D square mesh using "line-dipoles" and light-emission is described by cylindrical waves. The according vacuum Green's dyads can be derived by an integration of the 3D Green's dyad along the infinite axis (for details see Appendix B). Like the 3D dyads, we provide a quasi-static approximation for a layered environment with up to three layers based on the method of mirror dipoles. Consequently the same limitations hold as in the case of the quasistatic 3D tensors.

In pyGDM the usage of 2D tensors is very simple and is basically nothing more than replacing the 3D *dyads* class by the equivalent class for 2D. An example of the simulation of silicon nanowires of different shapes is depicted in Fig. 7. The simulations show silicon nanowires (SiNWs) of circular (7a), square (7b) and triangular (7c) cross section. The SiNWs are illuminated from above by a plane wave of wavelength  $\lambda_0 = 550$  nm, linearly polarized along the NW axis (transverse magnetic, TM). Fig. 7d-f show the electric (top panels) and magnetic (bottom panels) field intensity distributions inside the nanowire. The nanowire dimensions are chosen such that the magnetic dipole mode is excited. We note that the results agree qualitatively and quantitatively with FDTD simulations [24]. We emphasize that the full 3D optical response of infinitely long nanowires is completely described considering 2D simulations, which is of strong interest for investigating e.g. propagating modes [52,53].

The pyGDM-script to reproduce the results shown in Fig. 7 can be found in the online documentation under this link.





**Fig. 6.** Simulated extinction spectra and near-field amplitude maps for a single silicon cylinder and a cylinder dimer lying on a silica substrate ( $n_1 = 1.45$ ) surrounded by air ( $n_2 = 1$ ). Each cylinder, discretized by  $N = 1490$  unit cells on a hexagonal compact grid, has a diameter  $D = 140$  nm, a height  $H = 160$  nm and, for the dimer, a gap  $G = 30$  nm. The structures are illuminated by a plane wave from below (through the substrate). (a) total extinction cross section (blue curve), as well as the electric (orange) and magnetic (green) dipole contributions are calculated. Linear polarization along X (see inset of (a)). (d) Normalized amplitude maps of the electric and magnetic fields inside and around the structures (in the XZ plane) at the electric and magnetic resonances. The positions of the resonances are highlighted by the vertical dashed lines in (a), (b), (e) and (c), (f): Same as (a), (d) but for a dimer illuminated by a parallel (b), (e) or a perpendicular linear polarization (c), (f). The maps show areas of  $450 \times 425$  nm<sup>2</sup>. The outline of the respective structure is indicated by a dashed white line, the air-silica interface with a black solid line.

#### 5.4. EELS/CL

Another new feature is the support of fast-electron illumination and the implementation of corresponding evaluation functions through the new electron submodule. These new functionalities are based on prior theoretical work [55]. They allow to simulate electron energy loss spectroscopy (EELS) and cathodoluminescence (CL) experiments. A recapitulation is given in Appendix C, the formalism to calculate the energy loss probability is derived in Appendix C.1.

Fig. 8 shows an example in which we reproduce results from Ref. [54]. A flat (thickness of 40 nm), regular prism made of aluminum with a side length of 700 nm lies on a  $\text{Si}_3\text{N}_4$  membrane ( $n_1 = 2$ ), typically used in electron microscopy. It is surrounded by vacuum ( $n_2 = 1$ ). A normally incident beam of fast electrons (of 100 keV energy) passes through the sample, as illustrated in Fig. 8a by a red dashed line. In a first simulation we focus the electron beam on two positions: either on the tip (blue marker) or on the center of one side (orange marker). At those positions we calculate the cathodoluminescence (Fig. 8b) and EELS spectra (Fig. 8c). In a second step we fix the energy loss to 1.45 eV and perform a raster-scan simulation, in which the electron beam scans across the entire structure, as shown in Fig. 8d. We note that when the beam is moving over the structure, it is important to keep a constant distance between the electron beam and the meshcells' centers, in order to obtain a smooth mapping. To this end we implemented a tool `tools.adapt_map_to_structure_mesh`, which re-maps positions close to a meshcell exactly onto the respective closest discretization dipole.

Finally, we combine both types of simulations. With the fast electron beam we perform a line-scan along one side of the Al triangle. At each position we calculate an EELS spectrum, which allows to visualize the spatial dispersion of the plasmonic edge modes in the prism (Fig. 8e).

pyGDM-scripts to reproduce the results shown in Fig. 8 can be found in the online documentation under [link to example 1](#), [link to example 2](#) and [link to example 3](#).

#### 5.5. Optical chirality

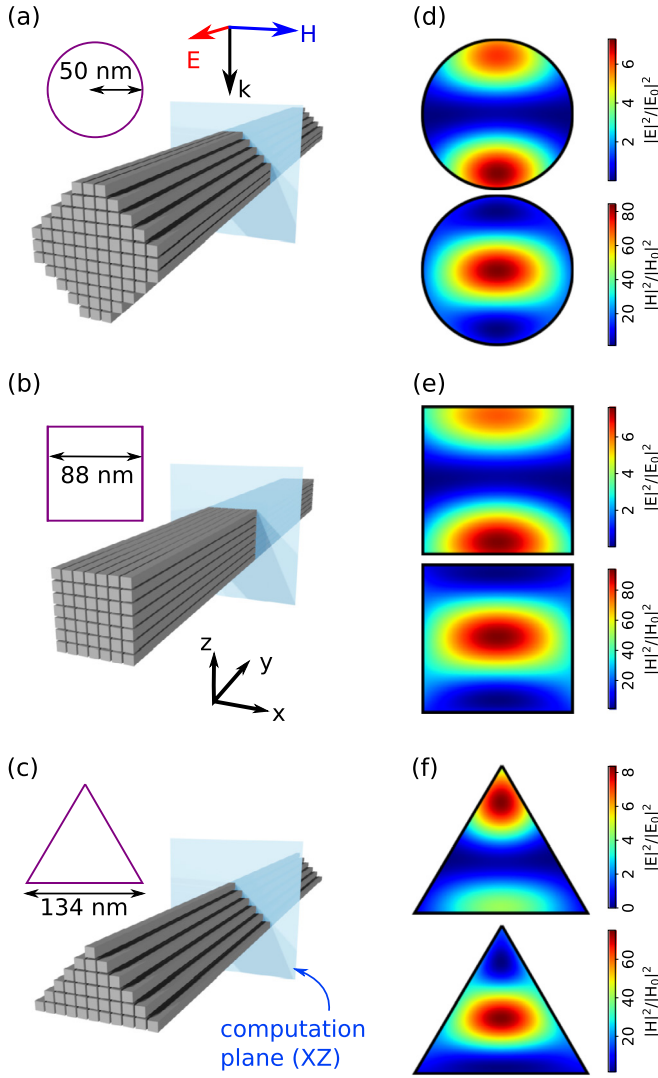
Another new function in pyGDM is the calculation of optical near-field chirality. The optical chirality is a scalar quantity which can be interpreted as a measure of the asymmetry in the coupling of left and right handed chiral emitters with the available photonic modes [57,58]. It is defined as [59]

$$\frac{C}{C_{\text{LCP}}} = -\text{Im}(\mathbf{E}^* \cdot \mathbf{H}), \quad (21)$$

where the superscript asterisk stands for complex conjugation. pyGDM follows the convention to normalize the chirality  $C$  to the value of a left circularly polarized plane wave ( $C_{\text{LCP}}$ ). Consequently, left circularly polarized (LCP), respectively right circularly polarized (RCP) plane waves have a chirality of  $C_{\text{LCP}} = +1$ , respectively  $C_{\text{RCP}} = -1$ . Nanostructures inducing an optical chirality larger than that of a circularly polarized plane wave ( $|C| > 1$ ) are often referred to as *super-chiral*.

To demonstrate the new function, we reproduce in Fig. 9 selected examples of chiral near-fields from literature. Fig. 9a reproduces the chirality around a point dipole excited by a linearly polarized plane wave. This is illustrated by a small gold nanosphere in vacuum. The calculated optical chirality is in very good agreement with literature [56]. Fig. 9b illustrates the chirality above and below a gold nanorod, illuminated by a linearly polarized plane wave at the fundamental localized plasmon resonance, again reproducing results from literature with high fidelity [56]. In a third example shown in Fig. 9c, we calculate the chirality at the center of a silver nanorod dimer, which is either in a left-handed





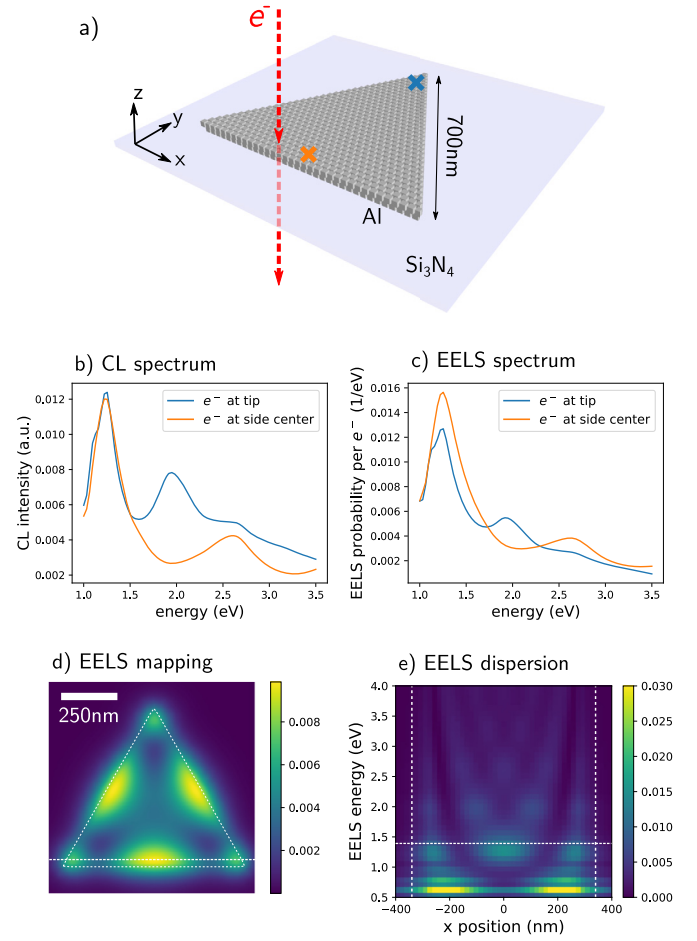
**Fig. 7.** 2D GDM simulations of the field intensity distribution inside infinitely long silicon nanowires (SiNWs) of different shapes: circular (a), square (b) and triangular (c). The sizes are chosen in order to excite the magnetic dipole Mie resonance at  $\lambda_0 = 550$  nm (normal incidence plane wave illumination with linear polarization along the NW axis). The corresponding electric and magnetic field intensity maps inside the SiNWs are shown in (d-f) in the upper, respectively lower subplots. The SiNWs are in vacuum. Note that for illustrative purposes the sketches (a-c) depict a significantly coarser discretization as used for the actual simulations.

(“L-dimer”) or right-handed (“R-dimer”) configuration. The bottom plot shows the spectrum of the respective chiralities (black line: L-dimer, green dashed line: R-dimer). In the insets above the spectra, we show the spatial mapping of  $C$  calculated 15 nm above the nanorod’s top surface. The L-handed (left plot) and R-handed (right plot) dimers are illuminated at the wavelength  $\lambda_0 = 770$  nm, leading to a peak of the chirality. The results agree well with literature [57].

pyGDM-scripts to reproduce the results shown in Fig. 9 can be found in the online documentation under [link to example 1](#), [link to example 2](#) and [link to example 3](#).

### 5.6. Field gradients and optical forces

A further new feature in pyGDM is a function to calculate the gradients of near-fields in- and outside the nanostructure. These gradients are required for instance to calculate the charge distribution inside plasmonic nanostructures [61] or to compute optical forces [62,60]. In pyGDM, we implement a simple numerical cen-



**Fig. 8.** Example of electron energy loss spectroscopy (EELS) and cathodoluminescence (CL) simulations, reproducing results from Ref. [54]. A regular aluminum prism of 700 nm side length lies on an Si<sub>3</sub>N<sub>4</sub> membrane ( $n_1 = 2$ ). A normally incident, focused beam of 100 keV electrons is raster-scanned along the sample. From the interaction with the plasmonic structure we calculate (b) the CL spectrum at the tip (blue) respectively side center (orange), (c) the EELS spectrum (again at the tip and side center), (d) the spatially resolved EELS probability for an energy loss of 1.45 eV (indicated by a horizontal white dotted line in (e)), as well as (e) the spatial dispersion of the EELS signal for various energies along the bottom side (indicated by a horizontal white dotted line in (d)). The colorbar scales in (d-e) represent the energy loss probability per electron and per eV.

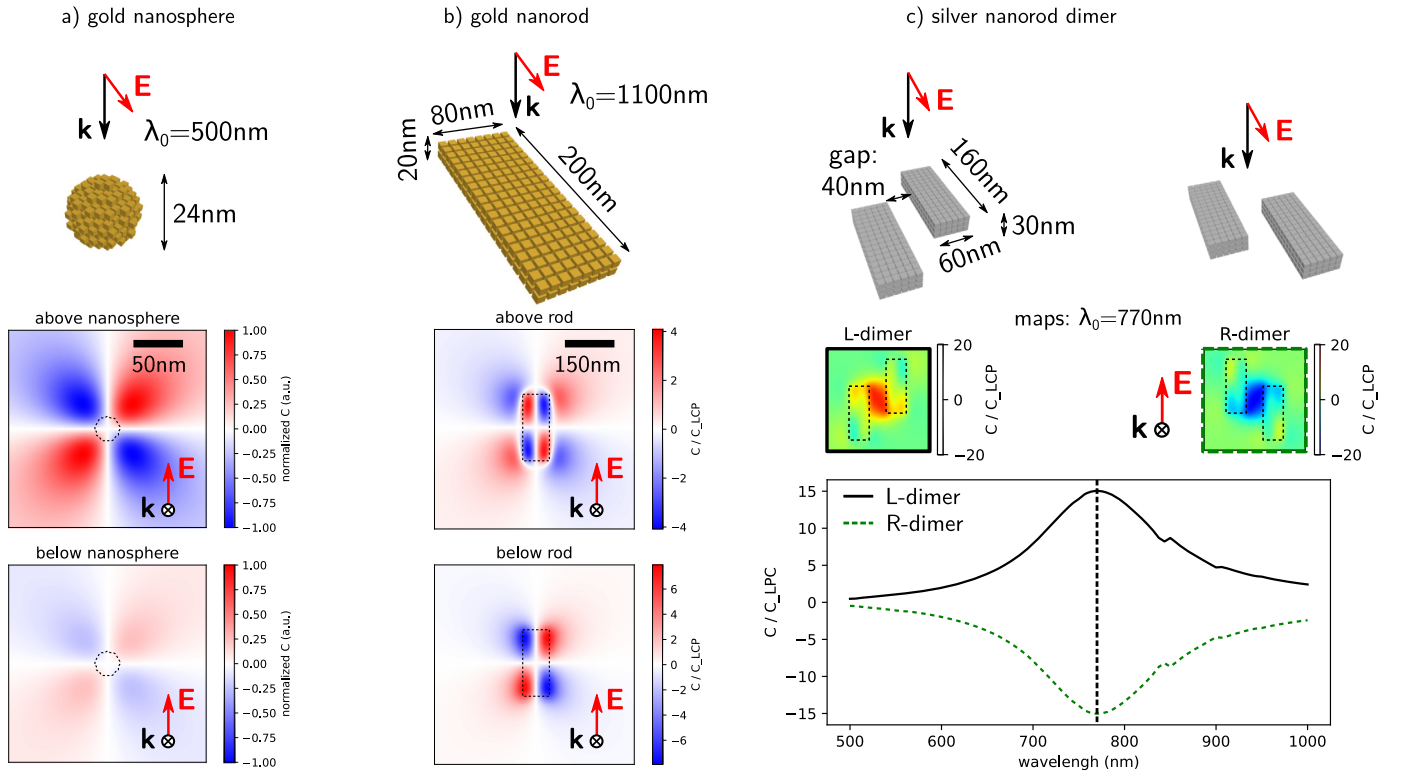
ter differentiation, which approximates the derivative with second order accuracy [63]:

$$\frac{\partial f(x)}{\partial x} \approx \frac{f(x + \Delta) - f(x - \Delta)}{2\Delta} \quad (22)$$

By default, the discretization stepsize is used as value for  $\Delta$ , so inside a structure the fields are taken at each meshpoint. Outside the structure the field can be evaluated at arbitrary positions around the evaluation location. Here the user may optionally choose a value smaller than the stepsize for  $\Delta$ . Note that the gradients inside the structure can be obtained also using a self-consistent formulation, which would however require a coupled system of size  $12N \times 12N$ , involving the second derivative of the field susceptibilities ( $3 \times 3 \times 3 \times 3$  tensors) [60].

Being able to compute the field gradients at the position of a dipole, the optical force acting on this dipole of moment  $\mathbf{p}(\omega)$  can be obtained through [60]

$$F_i = \frac{1}{2} \text{Re} \left( \sum_{j=1}^3 p_j(\omega) \frac{\partial E_j^*(\omega)}{\partial e_i} \right), \quad (23)$$



**Fig. 9.** Examples of calculations of the chirality of the optical near-field. (a-b) Reproduction of results from [56]. (a) Plane wave illuminated gold nano-sphere as a model for a point dipole response, resulting in a symmetric pattern of the optical chirality close to the structure (here calculated at  $\pm 25$  nm above and below the nanosphere). The illumination wavelength (here  $\lambda_0 = 500$  nm) does not have a significant impact on the chirality pattern. (b) Gold nanorod illuminated by a plane wave at the wavelength of the dipolar plasmon resonance ( $\lambda_0 = 1100$  nm), linearly polarized along the rod's long axis. The chirality is calculated at  $\pm 25$  nm above and below the surface of the nanorod. (c) Reproduction of results from [57]. Chirality spectra calculated at a distance of 15 nm above the top surface in the center between the two silver nanorods. The colormaps show spatial mappings of the chirality in the L-dimer and R-dimer case at the wavelength of maximum chirality ( $\lambda_0 = 770$  nm, indicated by a vertical dashed line in the bottom plot). For simplicity all structures in (a-c) are placed in vacuum, yet the qualitative trends are very well reproduced, in comparison with the original publications [56,57].

where the superscript asterisk  $*$  indicates complex conjugation,  $i, j \in \{1, 2, 3\}$  stand for the Cartesian directions ( $x, y$  and  $z$ ), and  $e_i$  corresponds to the Cartesian direction of the force vector, along which we also differentiate the electric field at the position of the dipole. The force acting on a larger nanostructure can be obtained by summation of the forces on each individual mesh cell. Note that the force on a nanoparticle can also be obtained through the Maxwell stress tensor by integration of the electromagnetic fields on a surface enclosing the particle [60]. However, in the GDM the gradients method can take advantage of the pre-calculated internal fields and is therefore usually faster.

In Fig. 10a we illustrate a system of two dielectric nano-cubes of refractive index  $n = 2$ , 165 nm side length and variable distance  $d$  between each other. The cubes are placed in vacuum and illuminated by a plane wave from below, polarized along  $x$ . Due to symmetry the  $E_y$  component is always zero. In Fig. 10b and 10d we show the real part of the non-zero field components  $E_x$ , respectively  $E_z$ . Fig. 10c and 10e show the according spatial derivative along  $z$ , required to calculate the  $z$  component of the optical force. Fig. 10f shows the force  $F_z$  on the upper of the two dielectric cubes.

A second example is illustrated in Fig. 11a. A dielectric cube of index  $n_{\text{cube}} = 1.5$  and side length  $a = 200$  nm is kept at short distance  $d$  above a dielectric substrate of same refractive index ( $n_1 = 1.5$ ) and illuminated from below with a plane wave ( $\lambda_0 = 633$  nm, polarization along  $X$ ). In Fig. 11b the force along  $z$  is shown for increasing distance  $d$ , either calculated with the mirror-charge approximation (blue dashed line) or using the retarded propagator for the dielectric interface via the `pyGDM2_retard` package. The re-

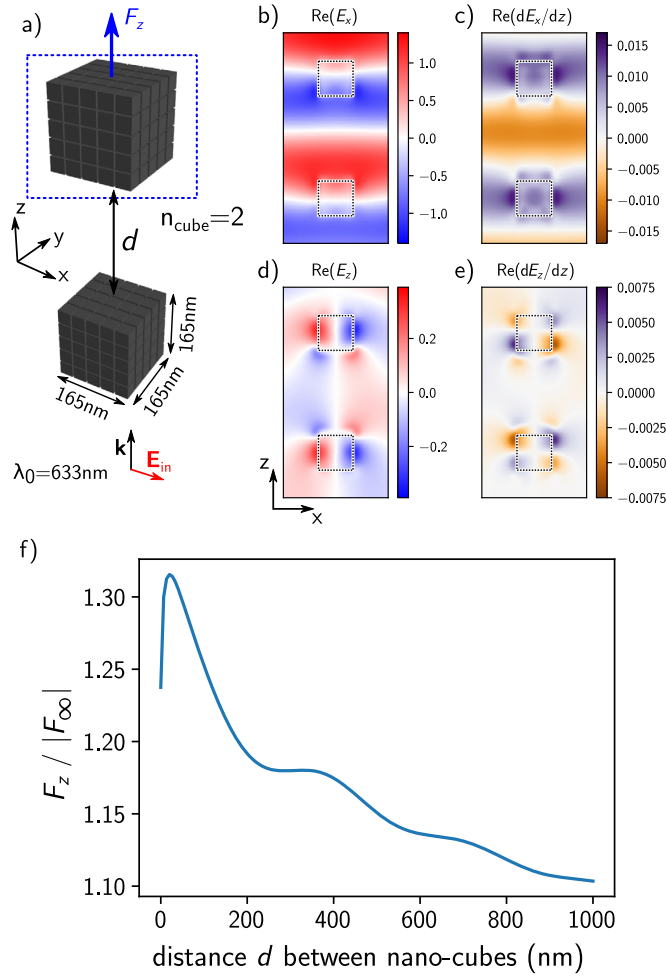
sults are in excellent agreement with a similar configuration from literature, using a sphere instead of a cube [60]. The force between the two cubes (Fig. 10) behaves actually very similarly to the optical force acting on the nano-cube close to a substrate (Fig. 11), with the difference of an additional decreasing component in the former case. This can be explained by the small size of the second nano-cube in comparison with the infinite extension of the surface.

We note, that the gradients inside plasmonic nanostructures can also be used to calculate the charge distribution [61]. Due to the very short skin depth in metals, the field gradients can be very strong which poses difficulties together with the usually relatively coarse volume discretization in the GDM. While this often does not affect the accuracy of results that average the internal fields (like the optical forces, far-fields or also near-fields *outside* the structure), the internal fields and their gradients can appear noisy which may pose problems for computations of quantities *inside* the nanostructure on a small scale, such as the charge density, in which cases fine meshings need to be used.

The `pyGDM`-script to reproduce the results shown in Fig. 10 can be found in the [online documentation under this link](#). For a script to reproduce Fig. 11 see [this link](#).

### 5.7. Scattering into a substrate

By calculating the scattered intensity in the far-field region on a spherical surface around the nanostructure, the radiation pattern of a nanostructure can be deduced. In the new version of `pyGDM` the asymptotic far-field propagators are fully implemented for a single

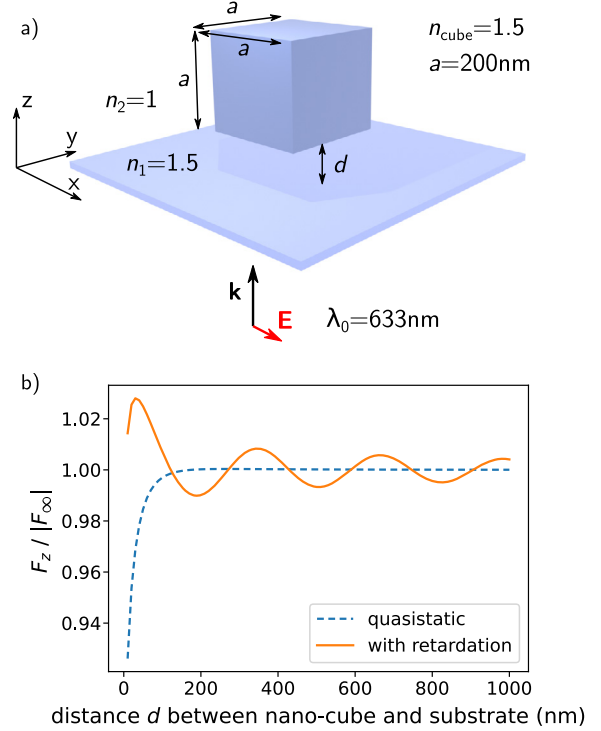


**Fig. 10.** Calculation of field gradients to deduce optical forces. (a) illustration of the problem. Two dielectric nano-cubes ( $n_{\text{cube}} = 2$ ) in vacuum are aligned on the  $z$ -axis with an inter-cube distance  $d$ . The cubes are illuminated from below with a plane wave of wavelength  $\lambda_0 = 633$  nm, with linear polarization along  $X$ . We seek to calculate the optical force  $F_z$  acting in  $z$ -direction on the upper cube. (b-e) real part of the optical fields  $E_x$  (b) and  $E_z$  (d) as well as their spatial derivatives in  $z$ -direction  $\partial E_x / \partial z$  (c) and  $\partial E_z / \partial z$  (e). Shown areas are  $500 \times 1000 \text{ nm}^2$  in the  $y = 0$  plane, cutting through the center of the dielectric cubes. The nano-cube outlines are indicated by white/black dotted lines. All fieldmaps are in units of the incident field amplitude  $E_0$ . (f) optical force on the upper nano-cube as function of the distance  $d$  between the two cubes, normalized to the force at large distance.

interface (see Appendix A.2), hence far-field patterns of scattering into a substrate can now be calculated.

In Fig. 12, we present a simulation of the radiation pattern of a gold Yagi-Uda antenna in vacuum ( $n_2 = 1$ ) laying on a silica substrate ( $n_1 = 1.5$ ), similar to the configuration described in the reference [64]. It consists of 5 nanorods of radius  $R = 30$  nm. From left to right, we have the reflector rod of length  $L_1 = 170$  nm, the feed rod of length  $L_2 = 145$  nm and the last three, of length  $L_3 = 115$  nm, are parasitic elements serving as directors (see Fig. 12a). An electric dipole  $\mathbf{p}$ , of emission wavelength  $\lambda_0 = 820$  nm, is positioned in the vicinity of the feed. The rods are discretized on cubic mesh with a discretization step of 10 nm.

We obtain results in agreement with literature [64–66], as we find that the emission of the emitter-antenna system points unidirectionally towards the positive  $X$  axis (see Fig. 12b). Additionally, the change in refractive index between the vacuum and the substrate bends the direction of radiation to the critical angle at  $\theta_c = \arcsin(n_1/n_2) = 222^\circ$  (Fig. 12c). The nano Yagi-Uda antenna makes it possible to emit predominantly in the direction of the positive  $X$  axis. The dipole emitter couples to the feed rod, which



**Fig. 11.** Calculation of the optical force of a dielectric cube above a substrate. (a) illustration of the configuration. A dielectric nano-cube (side length  $a = 200$  nm,  $n_{\text{cube}} = 1.5$ ) is placed in vacuum ( $n_2 = 1$ ) at a small distance  $d$  above a dielectric substrate of same index ( $n_1 = 1.5$ ). The system is illuminated from below with a plane wave of linear polarization along  $X$  and vacuum wavelength  $\lambda_0 = 633$  nm. We calculate the optical force  $F_z$  on the cube. (b) optical force  $F_z$  on the nano-cube as function of the distance  $d$  to the dielectric interface, normalized to the force at large distance to the surface. The optical force is calculated either using the quasistatic mirror-charge approximation for the interface (blue dashed line) or with a Green's tensor for the surface including retardation (orange solid line). The results are in good agreement with similar simulations for a nano-sphere [60].

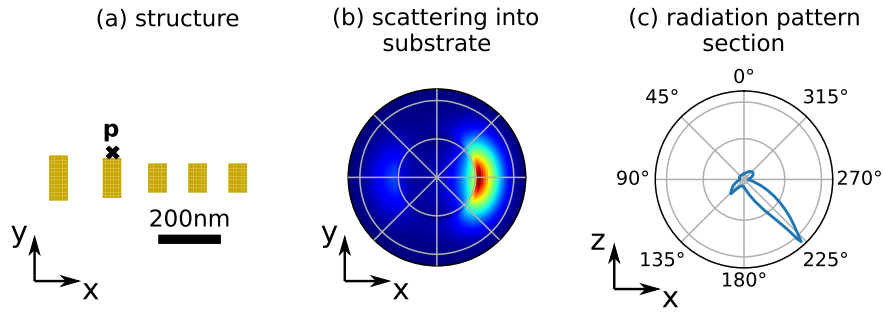
strongly enhances the signal due to the Purcell effect. Then, as a result of the specifically tailored inter-rods distances, the other elements impose the directivity of emission due to constructive interference only towards the positive  $X$  direction.

The pyGDM-script to reproduce the results shown in Fig. 12 can be found in the online documentation under this link.

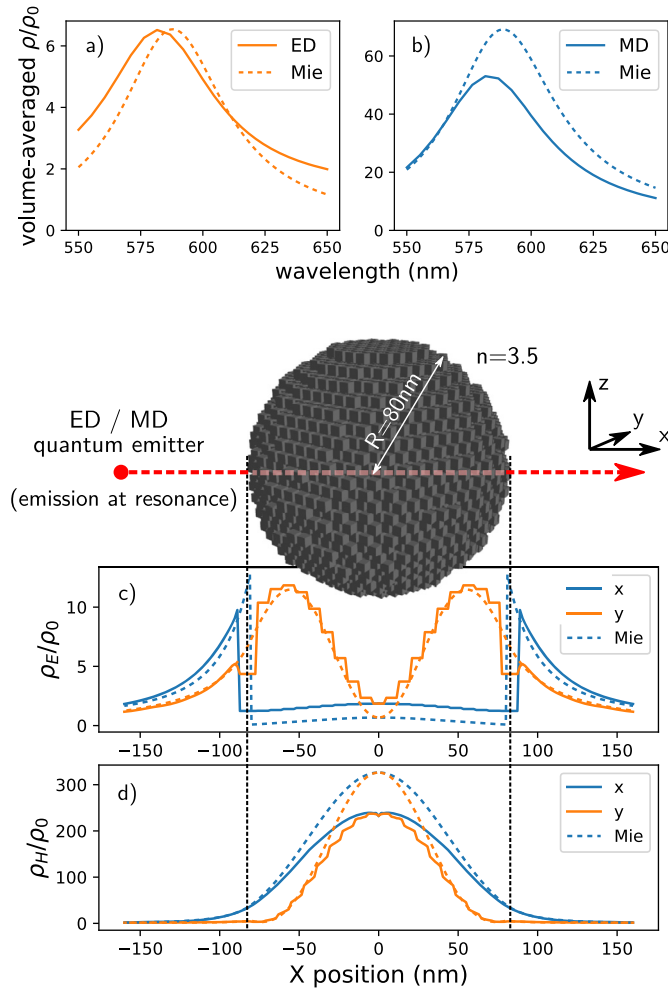
Please note that pyGDM includes a module for evolutionary optimization of photonic nanostructures. Using this functionality we have demonstrated inverse design of directional nano-antennas [66]. The analysis of the near-field amplitude and phase clearly reveals the role of the antenna elements even for the complex optimized nanostructure that we obtained.

### 5.8. Quantum emitter decay rate inside nanostructures

Like the rest of the code, in the new version of pyGDM the `core.decay_rate` function has been entirely rewritten in pure python. A first noteworthy improvement is a significant optimization of the algorithm, in which now the required field susceptibilities are pre-calculated, which hugely accelerates the numerical evaluation of the volume integral. Furthermore, like all other pyGDM functions, the decay-rate calculation is now capable to handle multi-material structures and materials with non-isotropic permittivity. Finally, it is now possible to calculate both, the electric and magnetic decay rates also *inside* nanostructures. pyGDM uses a formalism based on field-susceptibilities for the derivation of the photonic local density of states (LDOS), as detailed in references [30,67–69].



**Fig. 12.** Radiation pattern of a Yagi Uda antenna in vacuum ( $n_2 = 1$ ) lying on a substrate of refractive index  $n_1 = 1.5$ . (a) Top view of the structure. The structure consists in five nanorods of radius  $R = 30$  nm, the lengths are respectively  $L_1 = 170$  nm for the reflector rod (left most),  $L_2 = 145$  nm for the feed rod (closest to the electric dipole) and  $L_3 = 115$  nm for the guide rods (three others). The dipole is oriented along the Oy axis and emits light at  $\lambda_0 = 820$  nm. (b) Top view of the radiation pattern intensity distribution in the substrate. (c) Angular radiation pattern in the plane XZ, the maximum of emission is centered around the critical angle  $\theta_c = 222^\circ$ .



**Fig. 13.** Decay rate inside a dielectric sphere (refractive index  $n = 3.5$ ) of radius  $R = 80$  nm, placed in vacuum. Spectrally resolved decay rate  $\rho$ , averaged over the entire volume of the sphere for (a) electric dipole (ED) transitions and (b) magnetic dipole (MD) transitions. Solid lines correspond to the GDM simulation, dashed lines are values obtained by Mie theory. (c-d) show the decay rate modification along a line parallel to the X axis through the center of the sphere. Blue lines indicate an x-oriented, orange lines a y-oriented dipole transition, values from Mie theory are again shown by dashed lines. (c) electric, and (d) magnetic dipole transition, calculated at the respective maximum decay rate enhancement.

We demonstrate the possibility to calculate magnetic decay rates inside nanostructures by a comparison to Mie theory [70,71]. We calculate the normalized electric and magnetic LDOS  $\rho/\rho_0$  averaged over the particle volume of a dielectric sphere of refractive index  $n = 3.5$  and radius  $R = 80$  nm in air. This models the decay rate of a dielectric particle doped with electric or magnetic emitters.

We note that the sphere is discretized by around 7000 mesh-points, which would have resulted in many days of simulation time with the former code for numerical integration. It becomes feasible with the optimized implementation, which runs fully parallelized and for the described problem takes in the order of 15 minutes per wavelength running on 8 CPU cores (3rd generation AMD Ryzen). The corresponding spectra are shown in Fig. 13a, respectively 13b.



Despite a slight resonance shift the agreement is very good for the electric LDOS, and of the correct order of magnitude for the magnetic case. To investigate the origin of the discrepancy in the magnetic case, we calculate the E- and H-LDOS along a line parallel to OX through the center of the nanosphere (Figs. 13c-d). The pyGDM-simulated electric LDOS reproduces well Mie theory (Fig. 13c), but more than 20% error is found for the magnetic contribution (Fig. 13d). On the one hand we attribute this discrepancy to the non-perfect spherical shape of the particle that is meshed using cells with a stepsize of 8 nm, because the resonance position strongly depends on the shape. On the other hand, the magnetic LDOS is deduced by repropagating the electric Green's tensor [67], which can explain the larger error in the interior of the structure. In conclusion, the LDOS can be calculated with a good qualitative accuracy. However, in demanding situations like when high refractive indices are used or for round geometries like spheres, which are a challenging geometry for the GDM's volume discretization, the associated limitations of pyGDM should be kept in mind.

The pyGDM-script to reproduce the results shown in Fig. 13 can be found [in the online documentation under this link](#).

### 5.9. Plasmonic properties of doped dielectric nanostructures

Doped semi-conductors nanostructure has open the way towards tunable plasmonic [72]. Doping consists of introducing impurities into a pure material in order to modify its electronic and optical properties. The impurities provide free carriers which form a free electron cloud, that enables a plasmonic response of the material. A new class of such doped dielectric materials has been added to pyGDM, which adds a supplemental Drude term to the permittivity of the undoped material, based on a Drude-Lorentz model [73]

$$\epsilon(\omega) = \epsilon_{\text{int}}(\omega) - \frac{\omega_p^2}{\omega^2 + i\gamma\omega}, \quad (24)$$

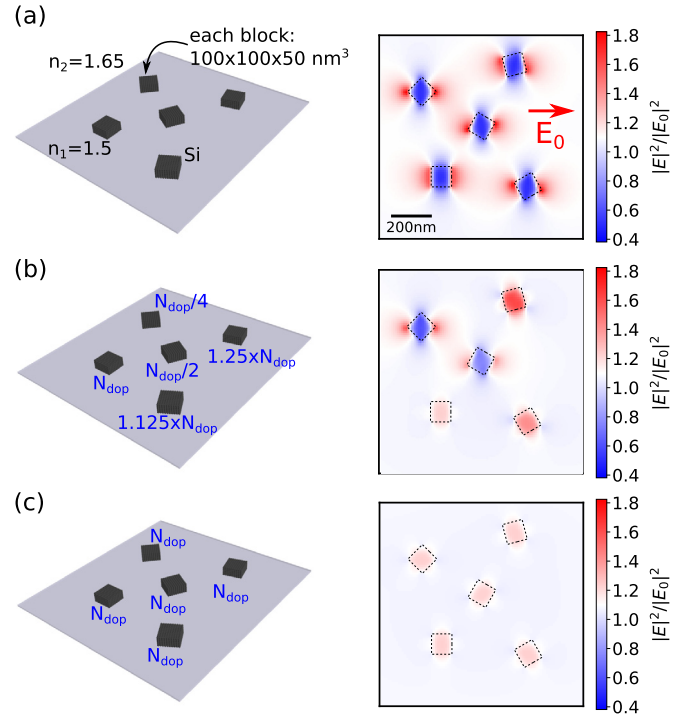
where  $\epsilon_{\text{int}}(\omega)$  is the dielectric constant of the intrinsic dielectric,  $\gamma$  is the damping rate due to the collisions, and the plasma frequency  $\omega_p$  is related to the concentration of free carriers  $N_{\text{dop}}$  according to

$$\omega_p = \sqrt{\frac{4\pi N_{\text{dop}} e^2}{m^*}}, \quad (25)$$

where  $e$  and  $m^*$  are respectively the elementary charge and the effective mass of the carriers.

We added three material-classes `materials.hyperdoped_xxx` to pyGDM. One adds the doping-induced plasmonic response to a tabulated dielectric permittivity which is loaded from an external user-defined text file. The other two are specific cases, namely silicon as well as a dielectric material of constant refractive index.

In Fig. 14a, we show a simulation of the electric near-field intensity maps 10 nm above the top surface of an array of five undoped silicon pads. The structures are embedded in an environment of refractive index  $n_2 = 1.65$ , laying on a substrate with  $n_1 = 1.5$ , all illuminated by a descending plane wave of wavelength  $\lambda_0 = 2754$  nm. In Fig. 14b, we observe a strong change in the near-field pattern as we gradually increase the doping of the pads. The optical contrast decreases as the doping approaches  $N_{\text{dop}} = 5 \times 10^{20} \text{ cm}^{-3}$ . When the doping is further increased, the near-field contrast increases again, yet with an opposite sign. The fact that the nanostructures with a doping of  $N_{\text{dop}} = 5 \times 10^{20} \text{ cm}^{-3}$  seem to disappear (Fig. 14c) is due to an index matching of the real part of the refractive index of the object with that of the environment at a specific doping concentration [74].



**Fig. 14.** Evolution of the near-field pattern when progressively doping five silicon pads until a carrier density of  $N_{\text{dop}} = 5 \times 10^{20} \text{ cm}^{-3}$  is reached (window size:  $1000 \times 1000 \text{ nm}^2$ ). (a) Geometry consisting of five undoped silicon blocks of size  $100 \times 100 \times 50 \text{ nm}^3$  in an environment  $n_2 = 1.65$  and lying on a substrate  $n_1 = 1.5$ , illuminated by a descending plane wave. The electric near-field intensity map computed 10 nm above the top surface of the pads at  $\lambda_0 = 2754 \text{ nm}$  is shown on the right. (b) same simulation as (a), but with various different doping concentrations. Each pad's dopant concentration relative to  $N_{\text{dop}}$  is indicated by a blue label. (c) same as (a), but each silicon pad is doped with the concentration  $N_{\text{dop}}$ , leading to minimum near-field contrast.

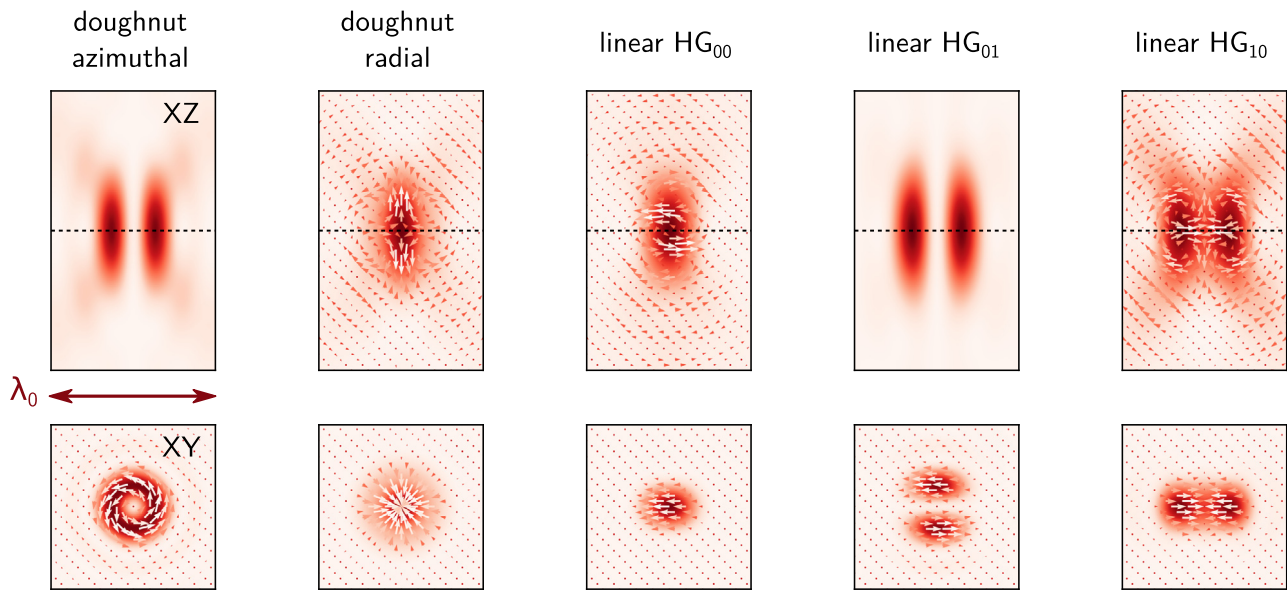
The pyGDM-script to reproduce the results shown in Fig. 14 can be found [in the online documentation under this link](#).

### 5.10. New illuminations: vectorbeams

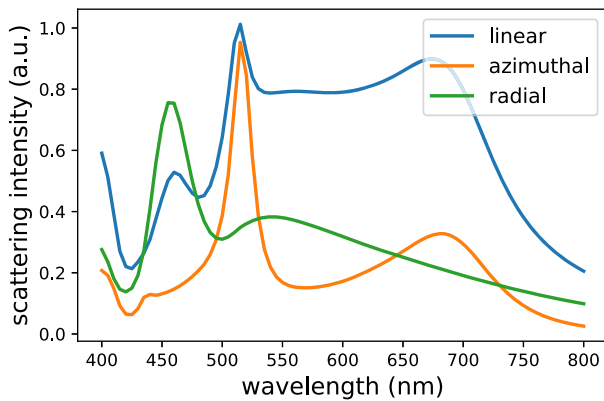
In the pyGDM update we implemented a few additional focused vectorbeams, following chapter 3.6 of the textbook of Novotny and Hecht [39]. We now provide generator functions for doughnut beams with azimuthal and radial polarization as well as linearly polarized Hermite-Gauss beams for the orders  $\text{HG}_{00}$ ,  $\text{HG}_{01}$  and  $\text{HG}_{10}$ . Focusing through a numerical aperture of  $\text{NA} = 1.4$  is shown in Fig. 15. We note that pyGDM implements a full tight focus model of these beams, including the correct  $E_z$  component. So far the vector-beams are limited to homogeneous environments, but we are developing an implementation of taking into account transmission and reflection at an interface, for which an experimental version is already available, which currently undergoes extensive testing.

To demonstrate the focused vectorbeam illuminations, we show in Fig. 16 scattering spectra of a silicon nanosphere of diameter  $D = 170 \text{ nm}$  in a homogeneous oil environment ( $n_{\text{oil}} = 1.48$ ). The sphere is illuminated by either a Gaussian beam of  $\text{HG}_{00}$  mode and linear polarization (blue line), or by a doughnut mode of either azimuthal (orange line) or radial (green line) polarization. The pyGDM simulations reproduce well recently reported experimental results [75].

A pyGDM-script to reproduce the results shown in Figs. 15 and 16 can be found [in the online documentation under this link](#).



**Fig. 15.** Electric field intensity plots of the new added focused vectorbeams. From left to right: Doughnut modes (azimuthal and radial polarizations) and Hermite-Gauss modes of the first orders, all calculated for focus by a numerical aperture  $NA = 1.4$  and incidence from positive towards negative  $z$ . Top panels show the plane  $y = 0$ , bottom plots show the focal plane at  $z = 0$ , which is indicated by a dotted black line in the top panels. The real part of the electric field vectors is indicated by small arrows.

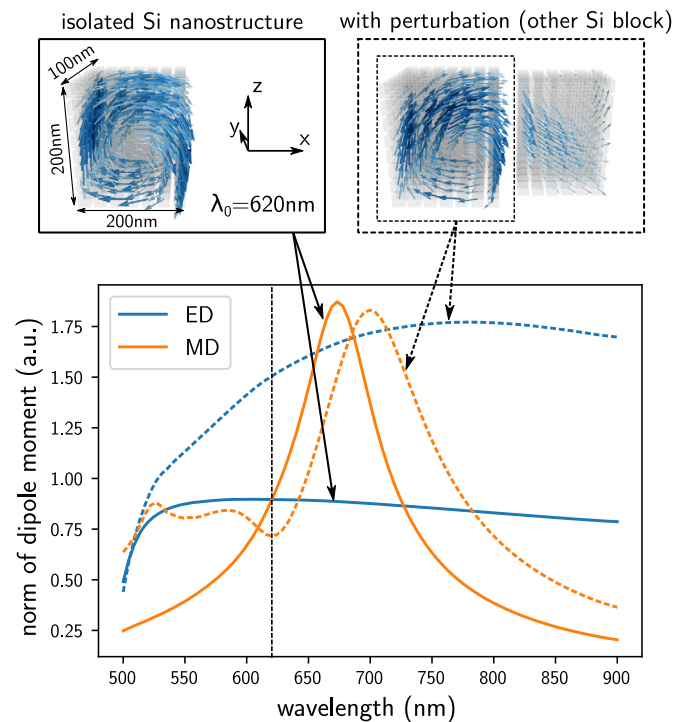


**Fig. 16.** Scattering from a silicon nanosphere of diameter  $D = 170$  nm in oil ( $n_{oil} = 1.48$ ), illuminated by different polarized vectorbeams through a numerical aperture of  $NA = 1.2$ . The blue line corresponds to a linear polarized  $TEM_{00}$  beam, the orange line to azimuthal polarization and the green line to radial polarization.

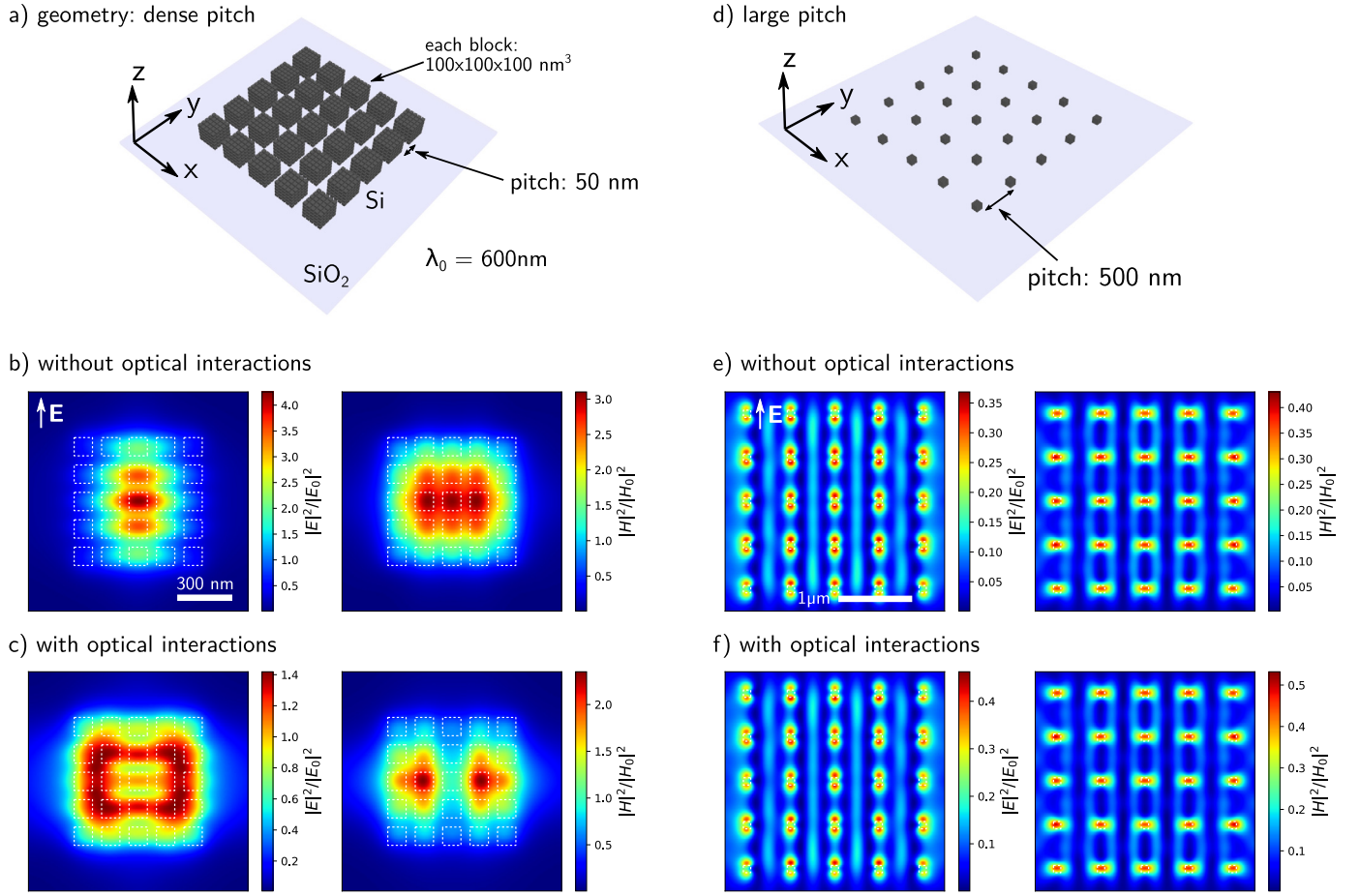
### 5.11. Tools for the analysis of optical interaction effects

Several tools have been added to pyGDM in the past 3 years. Along with tools to rotate and shift structures (`structures.rotate` and `structures.shift`), we added a tool to combine simulations (`tools.combine_simulations`, assuming identical configurations for each simulation), as well as to split the geometry of a simulation (`tools.split_simulation`). These tools allow in particular to manipulate simulation objects with already calculated fields, splitting or combining geometries whilst preserving the formerly calculated optical response. This allows to analyze the impact of optical interactions and near-field coupling in an ensemble of several nanostructures.

We demonstrate a possible use scenario of the new function `tools.split_simulation` in Fig. 17. On the left we show the real part of the electric field inside an isolated silicon cube under  $X$  polarized plane wave illumination at  $\lambda_0 = 620$  nm. On the right we show a second simulation with the same silicon object but including a second, smaller silicon block close to the original one. This additional nanostructure perturbs the optical response of the



**Fig. 17.** Electric (ED) and magnetic dipole moment (MD) analysis of a silicon nanoblock. The block is either isolated (top left sketch), or perturbed by a second, smaller silicon nanoblock (top right illustration). The simulations are performed in vacuum with  $X$  polarized plane wave illumination from the top. To assess the impact of the perturbation on the electric and magnetic dipole moments (MD) inside the larger Si block, the tool “`tools.split_simulation`” is used to split off a sub-part of the structure after simulating the coupled system of two blocks (dashed box in top right sketch) in order to analyze the effect of the perturbation. The bottom plot compares the dipole moment magnitudes of the isolated case (solid lines) with the perturbed silicon nanoblock (dashed lines). This analysis reveals a shift of the magnetic dipole resonance, some new emerging magnetic contributions (between 500 nm and 600 nm) as well as a significant increase in the induced electric dipole moment as result of the perturbation by a second silicon nanostructure. A vertical dashed line indicates the wavelength  $\lambda_0 = 620$  nm of the field distributions shown in the top figures.



**Fig. 18.** Impact of optical interactions on an array of silicon nano-blocks. (a) shows the considered geometry, consisting of a square array of 25 silicon nano-cubes of 100 nm side length, lying on a glass substrate ( $n_1 = 1.45$ ). The spacing between the cubes is 50 nm. The array is illuminated from below by a  $\lambda_0 = 600$  nm plane wave, linearly polarized along  $y$ . (b–c) show electric (left) and magnetic (right) near-field intensity maps, calculated 50 nm above the top-surface of the blocks, on  $1.2 \times 1.2 \mu\text{m}^2$  large areas. (b) simulation without optical interactions at the excitation, using the new “tools.combine\_simulations” function. (c) full simulation with all optical interactions. (d–f) same as (a–c) but with a larger pitch of 500 nm. Maps in (e–f) are  $3 \times 3 \mu\text{m}^2$  large.

larger block. Using `tools.split_simulation` we can remove the small block from the simulation once the internal fields are calculated, and calculate further observables from the perturbed internal fields, that have been modified by the optical interactions. The bottom plot in Fig. 17 shows the norm of the electric (blue) and magnetic (orange) dipole moments of the unperturbed (solid lines) as well as of the perturbed block (dashed lines). We find that the small perturbation has mainly a quantitative effect on the electric dipole resonance, whereas the magnetic dipole resonance is also strongly modified in qualitative sense. The resonance is redshifted and a few side-modes appear in the short wavelength region as a result of the presence of the small neighbor silicon block.

In a reciprocal approach to splitting simulations, it is possible to combine several simulations using `tools.combine_simulations`. Those simulations can contain the already calculated optical response of their respective nanostructures, which will be all combined in a new simulation object. The fields in such a combined simulation represent then an approximation without optical interactions between the constituents. This is illustrated in Fig. 18 by the example of an array of  $5 \times 5$  silicon nano-cubes. As depicted in Fig. 18a, the cubes of 100 nm side length and with a mutual spacing of 50 nm are deposited on a glass substrate, surrounded by air. A plane wave of  $\lambda_0 = 600$  nm and linear polarization along  $Y$  illuminates the structure from below. First, each cube is described by a separate pyGDM simulation object. After calculating

the individual cubes' optical responses, these 25 simulation objects are combined via `tools.combine_simulations`. The resulting simulation contains the optical response in the Born approximation, hence without multiple scattering or near-field coupling between the cubes. This combined simulation can subsequently be used in pyGDM to derive further quantities. Fig. 18b shows intensity maps of the electric (left) and magnetic (right) near-field, calculated 50 nm above the top surface of the (non-interacting) blocks. By running `scatter()` on the combined simulation object, we can then compute the self-consistent response of the full structure, hence including all optical interactions between the cubes. The resulting intensity maps are shown in Fig. 18c. The comparison with 18b demonstrates that near-field interactions can play a crucial role in the optical excitation of dense assemblies of nano-structures.

Fig. 18d–f shows the exact same simulations as 18a–c, with the difference of a significantly larger pitch between the silicon nano-cubes (500 nm distance instead of 50 nm). The comparison of simulations without and with optical interactions in Fig. 18e and 18f indicates a very weak optical interaction between the distant cubes, since the near-field maps are almost identical in this case. In such a de-coupled scenario, the `tools.combine_simulations` function can also be used to cost-efficiently calculate large arrays of non-interacting particles to evaluate interference effects, for instance in gratings.



**Table 1**

Summary of new functionalities implemented in pyGDM.

New Functionalities
<ul style="list-style-type: none"> <li>• 2D simulations</li> <li>• Substrates with retardation effects</li> <li>• Fast electrons illumination: EELS and CL</li> <li>• Internal H-field</li> <li>• Fields gradients, optical forces</li> <li>• Decay-rate for multi-material structures</li> <li>• Magnetic decay-rate inside structures</li> <li>• Multipole decomposition (ED / MD)</li> <li>• Combine and split simulations</li> <li>• pyGDM-UI (Graphical user interface, <i>still experimental</i>)</li> </ul>

**Table 2**

Summary of improvements compared to the initial version of pyGDM.

Improvements
<ul style="list-style-type: none"> <li>• Automatic mesh-type recognition</li> <li>• Multi-materials nanostructures</li> <li>• Proper distinction inside / outside object in NF calculations</li> <li>• Structures supported in any layer of the environment</li> <li>• Dispersive environments <math>n_{env}(\omega)</math></li> <li>• New hardcoded materials: SiO<sub>2</sub>, TiO<sub>2</sub>, doped dielectrics</li> <li>• Optimized LDOS (decay rate) calculation</li> <li>• Structure consistency test at initialization</li> <li>• Support for GPU solver to accelerate simulations</li> <li>• Avoid unnecessary calculations in <code>nearfield</code> / <code>farfield</code></li> </ul>

**Table 3**

Summary of major bug fixes to different pyGDM functions.

function	bugfix description
extinct	Prefactor for environments $\epsilon \neq 1$
farfield	Asymptotic propagator for environments $\epsilon \neq 1$
heat	Correct equations implementation Tensorial alpha support
material	Correct $\epsilon$ for Ag
visu	Correct field-intensity calculation Non-cubic mesh in <code>visu.visu_contour</code>
fields	Correct H-field phase in plane wave and gaussian arbitrary incident angles for plane wave correct 3-layer environments in plane wave arbitrary polarization states

The pyGDM-script to reproduce the results shown in Fig. 17 can be found [in the online documentation under this link](#). For Fig. 18, see [this link](#).

## 6. All changes and bugfixes since version 1.0

The new features and improvements are summarized in Table 1, respectively Table 2. Important bugfixes concerning the physics are listed in Table 3.

## 7. Outlook

In the future, from the physical modeling point of view we plan to implement Green's tensors for periodic structures [76], retarded tensors for 2D simulations in layered environments and possibly for 3D systems with arbitrary numbers of interfaces [43,77], functions to calculate the non-radiative part of the LDOS [78,22] and the possibility to model femto-second laser pulses for ultra-fast optics applications [79]. We also plan to implement nonlinear nano-optics functions from earlier work into pyGDM [26,25]. On the long term we want to add support for meta-material unit-cells

having a direct magnetic response [4]. We are constantly implementing further improvements. For instance we work on including transmission and reflection coefficients at an interface for the new focused vector-beams. In the `pyGDM2_retard` module for the retarded description of a layered environment we are working on including field calculations in other zones than the one hosting the nanostructure.

From the technical side, we plan to implement a conjugate gradients solver to avoid full inversion of systems with large amounts of mesh-cells [3]. Furthermore, such iterative solver could be strongly accelerated using GPU-based fast Fourier transforms [80,81].

More flexible meshing is also a feature that we plan to implement in the future [44,20,45]. Furthermore, we are beginning to conceptualize a module to evaluate complex problems with pre-trained deep learning models, and we think about how to construct an interface which allows the easy creation of such deep learning accelerated models for specific applications [82].

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

We gratefully thank Aurélien Cuche, Otto Muskens and Frank Mersch for many inspiring discussions. This work was supported by the German Research Foundation (DFG) through a research fellowship (WI 5261/1-1), and by the computing facility center CALMIP of the University of Toulouse under grant P20010. YB and GCF acknowledge support from the French National Research Agency (ANR) project HiLight (ANR-19-CE24-0026), the French "Investissements d'Avenir" program (EUR-EIPHI 17-EURE-0002) and access to the HPC facilities of the University of Bourgogne (ccuB).

## Appendix A. Expressions of the field susceptibilities used in pyGDM

### A.1. Cartesian representation of the free space field susceptibilities

The solutions of the wave equations (5) are [33,38,67]

$$\mathbf{G}_0^{\text{EE}}(\mathbf{r}, \mathbf{r}', \omega) = \left[ \frac{\nabla \nabla}{\epsilon_{\text{env}}} + \mu_{\text{env}} k_0^2 \mathbf{I} \right] \frac{e^{ik|\mathbf{r}-\mathbf{r}'|}}{|\mathbf{r}-\mathbf{r}'|} \quad (\text{A.1a})$$

$$\mathbf{G}_0^{\text{EH}}(\mathbf{r}, \mathbf{r}', \omega) = ik_0 \nabla \times \frac{e^{ik|\mathbf{r}-\mathbf{r}'|}}{|\mathbf{r}-\mathbf{r}'|} \quad (\text{A.1b})$$

$$\mathbf{G}_0^{\text{HH}}(\mathbf{r}, \mathbf{r}', \omega) = \left[ \frac{\nabla \nabla}{\mu_{\text{env}}} + \epsilon_{\text{env}} k_0^2 \mathbf{I} \right] \frac{e^{ik|\mathbf{r}-\mathbf{r}'|}}{|\mathbf{r}-\mathbf{r}'|} \quad (\text{A.1c})$$

$$\mathbf{G}_0^{\text{HE}}(\mathbf{r}, \mathbf{r}', \omega) = -ik_0 \nabla \times \frac{e^{ik|\mathbf{r}-\mathbf{r}'|}}{|\mathbf{r}-\mathbf{r}'|} \quad (\text{A.1d})$$

with  $\mathbf{r} = (x, y, z)$ ,  $\mathbf{r}' = (x', y', z')$ ,  $k_0$  is the wave vector in the vacuum and  $k = \sqrt{\mu_{\text{env}} \epsilon_{\text{env}}} k_0$  the wave vector in the environment. As mentioned in the main document, all materials (gases or solids) encountered in nature have no intrinsic response to the magnetic field at optical frequencies. That means  $\mu_{\text{env}} = 1$ , which leads to a wave vector in the environment  $k = \sqrt{\epsilon_{\text{env}}} k_0 = n_{\text{env}} k_0$ . In Eq. (A.1), if we apply the different nabla operators to the scalar Green's function, we obtain the field susceptibilities as a sum of tensors

$$\mathbf{G}_0^{\text{EE}}(\mathbf{r}, \mathbf{r}', \omega) = \left[ -k^2 \mathbf{T}_1^{\text{EE}}(\mathbf{R}) - ik \mathbf{T}_2^{\text{EE}}(\mathbf{R}) + \mathbf{T}_3^{\text{EE}}(\mathbf{R}) \right] \frac{e^{ikR}}{\epsilon_{\text{env}}} \quad (\text{A.2a})$$



$$\mathbf{G}_0^{\text{HH}}(\mathbf{r}, \mathbf{r}', \omega) = \epsilon_{\text{env}} \mathbf{G}_0^{\text{EE}}(\mathbf{r}, \mathbf{r}', \omega) \quad (\text{A.2b})$$

$$\mathbf{G}_0^{\text{HE}}(\mathbf{r}, \mathbf{r}', \omega) = \left[ n_{\text{env}} k_0^2 \mathbf{T}_1^{\text{HE}}(\mathbf{R}) + i k_0 \mathbf{T}_2^{\text{HE}}(\mathbf{R}) \right] e^{i k_0 R} \quad (\text{A.2c})$$

$$\mathbf{G}_0^{\text{EH}}(\mathbf{r}, \mathbf{r}', \omega) = -\mathbf{G}_0^{\text{HE}}(\mathbf{r}, \mathbf{r}', \omega) \quad (\text{A.2d})$$

with  $\mathbf{R} = \mathbf{r} - \mathbf{r}' = (\Delta_x, \Delta_y, \Delta_z)$  the relative distance between the source at the position  $\mathbf{r}'$  and the evaluation point  $\mathbf{r}$ . The different dyadic tensors give the electromagnetic field contributions in the far field ( $\mathbf{T}_1^{\text{EE}}(\mathbf{R}), \mathbf{T}_1^{\text{HE}}(\mathbf{R}) \propto 1/R$ ), the intermediate ( $\mathbf{T}_2^{\text{EE}}(\mathbf{R}), \mathbf{T}_2^{\text{HE}}(\mathbf{R}) \propto 1/R^2$ ) and near field ( $\mathbf{T}_3^{\text{EE}}(\mathbf{R}) \propto 1/R^3$ ) regions. They are defined as [38,39]

$$\mathbf{T}_1^{\text{EE}}(\mathbf{R}) = \frac{\mathbf{R}\mathbf{R} - \mathbf{I}R^2}{R^3}, \quad (\text{A.3a})$$

$$\mathbf{T}_2^{\text{EE}}(\mathbf{R}) = \frac{3\mathbf{R}\mathbf{R} - \mathbf{I}R^2}{R^4}, \quad (\text{A.3b})$$

$$\mathbf{T}_3^{\text{EE}}(\mathbf{R}) = \frac{3\mathbf{R}\mathbf{R} - \mathbf{I}R^2}{R^5}, \quad (\text{A.3c})$$

and

$$\mathbf{T}_1^{\text{HE}}(\mathbf{R}) = \frac{1}{R^2} \begin{pmatrix} 0 & -\Delta_z & \Delta_y \\ \Delta_z & 0 & -\Delta_x \\ -\Delta_y & \Delta_x & 0 \end{pmatrix}, \quad (\text{A.4a})$$

$$\mathbf{T}_2^{\text{HE}}(\mathbf{R}) = \frac{1}{R^3} \begin{pmatrix} 0 & -\Delta_z & \Delta_y \\ \Delta_z & 0 & -\Delta_x \\ -\Delta_y & \Delta_x & 0 \end{pmatrix}. \quad (\text{A.4b})$$

## A.2. Weyl representation of the field susceptibilities of a flat surface

We consider in the following a single, flat interface at  $z = 0$ , separating the dipole environment (of refractive index  $n_2$ ) and a substrate (of refractive index  $n_1$ ). For the description of radiation from oscillating dipoles across planar interfaces, the Cartesian forms of the free space susceptibilities are exact but not well suited for this boundary conditions problem. It is easier to use electromagnetic fields with wavefronts parallel to the interface (plane wave for a flat interface; spherical wave for a spherical particle interface). Due to the broken symmetry along the  $z$ -axis, a plane wave expansion of the surface wave vector  $\mathbf{k}_{\parallel} = (\mathbf{k}_x, \mathbf{k}_y)$  is performed. Thus the susceptibilities associated with the surface are, as presented in the references [31,39], developed in 2D-integrals of the shape

$$\mathbf{G}_{2\text{-layer}}^{\alpha\beta+/-}(\mathbf{r}, \mathbf{r}', \omega) = \frac{i}{2\pi} \int d\mathbf{k}_{\parallel} F^{+/-}(\mathbf{r}, \mathbf{r}', \omega) \mathbf{N}_{\alpha\beta}^{+/-}(\mathbf{k}_{\parallel}, \omega). \quad (\text{A.5})$$

For all the terms, the “+” superscript means that we consider the case above the interface (in the medium containing the structure) and “-” under it (in the substrate). The  $\mathbf{N}_{\alpha\beta}^{+/-}(\mathbf{k}_{\parallel}, \omega)$  terms are  $(3 \times 3)$  matrices,  $k_z^- = \sqrt{\epsilon_1 k_0^2 - k_{\parallel}^2}$  and  $k_z^+ = \sqrt{\epsilon_2 k_0^2 - k_{\parallel}^2}$ , with  $\epsilon_1 = n_1^2$ . For the electric-electric coupling term, the definition of the matrices  $\mathbf{N}_{\alpha\beta}^{+/-}(\mathbf{k}_{\parallel}, \omega)$  are mentioned in the references [38] ( $\mathbf{N}_{\alpha\beta}^+$ , see Eq. (51) in [38]) and [83] ( $\mathbf{N}_{\alpha\beta}^-$ , Eq. (A5) in [83]). The spatial term  $F^+(\mathbf{r}, \mathbf{r}', \omega) = e^{i[\mathbf{k}_{\parallel} \cdot (\mathbf{I} - \mathbf{I}') + k_z^+(z + z')]}$  depends on the position of the dipole  $\mathbf{r}'$  above the interface, as well as on the evaluation position  $\mathbf{r}$  above the surface. The term  $F^-(\mathbf{r}, \mathbf{r}', \omega) = e^{i[\mathbf{k}_{\parallel} \cdot (\mathbf{I} - \mathbf{I}') - (k_z^- z - k_z^+ z')]}$  corresponds to the case of an evaluation location below the interface.

The equation (A.5) gives the electric field everywhere in space, nevertheless that requires numerical integrations in the complex plane to deduce the nine elements of the matrix. In the reference

[43], all the details on how to perform the calculation efficiently are described. The expression (A.5) is an integral of the shape:

$$G_s(\mathbf{r}, \mathbf{r}', \omega) = \frac{i}{2\pi} \int_0^{+\infty} dk_{\parallel} \int_0^{2\pi} g(k_{\parallel}, \alpha) e^{i r f(k_{\parallel}, \alpha)} d\alpha \quad (\text{A.6})$$

where the vector  $\mathbf{k}_{\parallel}$  is expressed in cylindrical coordinates  $(k_{\parallel}, \alpha)$ . This integration can be avoided in the near-field and far-field regions. In the near-field region (small values of  $r$ ) we use the electrostatic limit ( $c \rightarrow \infty$ ) and in the far-field region (large values of  $r$ ) we use the retarded asymptotic limit defined in the references [84]:

$$G_s(\mathbf{r}, \mathbf{r}', \omega) \sim_{\infty} \frac{1}{|\tan \theta|} g(\tilde{k}_{\parallel}, \tilde{\alpha}) \frac{e^{i n k_0 r}}{r}. \quad (\text{A.7})$$

In the previous equation,  $r$  and  $\theta$  are the spherical components of the evaluation position  $\mathbf{r} = (r, \theta, \varphi)$ ,  $n$  is the refractive index of the medium considered and  $(\tilde{k}_{\parallel}, \tilde{\alpha})$  defined the saddle point of the function  $f(k_{\parallel}, \alpha)$

$$\frac{\partial f}{\partial k_{\parallel}}(\tilde{k}_{\parallel}, \tilde{\alpha}) = 0 \quad \text{and} \quad \frac{\partial f}{\partial \alpha}(\tilde{k}_{\parallel}, \tilde{\alpha}) = 0. \quad (\text{A.8})$$

The calculus is fully explained in Ref. [37], here we present only the electrostatic and asymptotic expressions of the surface Green's dyadic tensor, which are implemented in pyGDM. As previously, due to the non-magnetic response of the environment, we get interest only to the “EE” and “HE” surface dyadic tensors.

### A.2.1. Non-retarded form and electrostatic limit

In the electrostatic limit we suppose  $c \rightarrow \infty$ , hence  $k_0 \rightarrow 0$ , which simplifies the problem drastically. For the electric-electric tensors that leads to the following expressions

$$\mathbf{G}_{2\text{-layer}}^{\text{EE}, \text{stat}}(\mathbf{r}, \mathbf{r}', \omega) = \Delta_{\text{surf}}^+ \frac{\mathbf{T}_3(\mathbf{R})}{\epsilon_{\text{env}}} \cdot \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad (\text{A.9})$$

$$\mathbf{G}_{2\text{-layer}}^{\text{EE}, \text{stat}}(\mathbf{r}, \mathbf{r}', \omega) = \Delta_{\text{surf}}^- \mathbf{T}_3(\mathbf{R}), \quad (\text{A.10})$$

with  $\Delta_{\text{surf}}^+ = (\epsilon_1 - \epsilon_2)/(\epsilon_1 + \epsilon_2)$  and  $\Delta_{\text{surf}}^- = 2/(\epsilon_1 + \epsilon_2)$ .

The magnetic-electric case vanishes in the electrostatic limit

$$\begin{aligned} \mathbf{G}_{2\text{-layer}}^{\text{HE}, \text{stat}}(\mathbf{r}, \mathbf{r}', \omega) &= \mathbf{0} \\ \mathbf{G}_{2\text{-layer}}^{\text{HE}, \text{stat}}(\mathbf{r}, \mathbf{r}', \omega) &= \mathbf{0}. \end{aligned} \quad (\text{A.11})$$

These results can be obtained equally with the image dipole theory [32].

### A.2.2. Retarded asymptotic form

The retarded expressions of the electric-electric Green's dyad, above and below the surface, are obtained by using the equation (A.7). These tensors make it possible to easily obtain the electric field emitted by an electric dipole, in the vicinity of an interface, far from it. The tensors depend mainly of the spherical components  $(r, \theta, \varphi)$  of the evaluation position, above the surface it is expressed as

$$G_{2\text{-layer}, \text{xx}}^{\text{EE}+\infty} = \mathcal{F}_0(r_p \cos^2 \theta \cos^2 \varphi - r_s \sin^2 \varphi) \quad (\text{A.12a})$$

$$G_{2\text{-layer}, \text{xy}}^{\text{EE}+\infty} = \mathcal{F}_0(r_p \cos^2 \theta + r_s) \cos \varphi \sin \varphi \quad (\text{A.12b})$$

$$G_{2\text{-layer}, \text{xz}}^{\text{EE}+\infty} = \mathcal{F}_0 r_p \sin \theta \cos \theta \cos \varphi \quad (\text{A.12c})$$

$$G_{2\text{-layer}, \text{yx}}^{\text{EE}+\infty} = G_{2\text{-layer}, \text{xy}}^{\text{EE}+\infty} \quad (\text{A.12d})$$

$$G_{2\text{-layer},yy}^{EE+\infty} = \mathcal{F}_0(r_p \cos^2 \theta \sin^2 \varphi - r_s \cos^2 \varphi) \quad (\text{A.12e})$$

$$G_{2\text{-layer},yz}^{EE+\infty} = \mathcal{F}_0 r_p \sin \theta \cos \theta \sin \varphi \quad (\text{A.12f})$$

$$G_{2\text{-layer},zx}^{EE-\infty} = -G_{2\text{-layer},xz}^{EE+\infty} \quad (\text{A.12g})$$

$$G_{2\text{-layer},zy}^{EE+\infty} = -G_{2\text{-layer},yz}^{EE+\infty} \quad (\text{A.12h})$$

$$G_{2\text{-layer},zz}^{EE+\infty} = -\mathcal{F}_0 r_p \sin^2 \theta \quad (\text{A.12i})$$

with

$$\mathcal{F}_0 = -k_0^2 \frac{e^{in_2 k_0 r}}{r} e^{-in_2 k_0 \sin \theta (x' \cos \varphi + y' \sin \varphi)} e^{in_2 k_0 \cos \theta z'} \quad (\text{A.13})$$

$r_p$  and  $r_s$  are the reflection coefficients, depending on the refractive index of the media from either side of the interface, given by [39, 37]

$$r_p = \frac{\epsilon_1 n_2 \cos \theta - \epsilon_2 (\epsilon_1 - \epsilon_2 \sin^2 \theta)^{1/2}}{\epsilon_1 n_2 \cos \theta + \epsilon_2 (\epsilon_1 - \epsilon_2 \sin^2 \theta)^{1/2}}, \quad (\text{A.14})$$

$$r_s = \frac{n_2 \cos \theta - (\epsilon_1 - \epsilon_2 \sin^2 \theta)^{1/2}}{n_2 \cos \theta + (\epsilon_1 - \epsilon_2 \sin^2 \theta)^{1/2}}.$$

A completely similar form is obtained for the electric-electric surface susceptibility below the interface

$$G_{2\text{-layer},xx}^{EE-\infty} = \mathcal{F}_1 \left( \frac{\Phi_p}{\epsilon_2} \cos^2 \varphi + \Phi_s \sin^2 \varphi \right) \cos \theta \quad (\text{A.15a})$$

$$G_{2\text{-layer},xy}^{EE-\infty} = \mathcal{F}_1 \left( \frac{\Phi_p}{\epsilon_2} - \Phi_s \right) \cos \theta \sin \varphi \cos \varphi \quad (\text{A.15b})$$

$$G_{2\text{-layer},xz}^{EE-\infty} = \mathcal{F}_1 \tau_p \frac{\epsilon_1}{\epsilon_2} \sin \theta \cos \theta \cos \varphi \quad (\text{A.15c})$$

$$G_{2\text{-layer},yx}^{EE-\infty} = G_{2\text{-layer},xy}^{EE-\infty} \quad (\text{A.15d})$$

$$G_{2\text{-layer},yy}^{EE-\infty} = \mathcal{F}_1 \left( \frac{\Phi_p}{\epsilon_2} \sin^2 \varphi + \Phi_s \cos^2 \varphi \right) \cos \theta \quad (\text{A.15e})$$

$$G_{2\text{-layer},yz}^{EE-\infty} = \mathcal{F}_1 \tau_p \frac{\epsilon_1}{\epsilon_2} \sin \theta \cos \theta \sin \varphi \quad (\text{A.15f})$$

$$G_{2\text{-layer},zx}^{EE-\infty} = -\mathcal{F}_1 \frac{\Phi_p}{\epsilon_2} \sin \theta \cos \varphi \quad (\text{A.15g})$$

$$G_{2\text{-layer},zy}^{EE-\infty} = -\mathcal{F}_1 \frac{\Phi_p}{\epsilon_2} \sin \theta \sin \varphi \quad (\text{A.15h})$$

$$G_{2\text{-layer},zz}^{EE-\infty} = -\mathcal{F}_1 \tau_p \frac{\epsilon_1}{\epsilon_2} \sin^2 \theta \quad (\text{A.15i})$$

with

$$\mathcal{F}_1 = k_0^2 \frac{e^{in_1 k_0 r}}{r} e^{-in_1 k_0 \sin \theta (x' \cos \varphi + y' \sin \varphi)} e^{ik_0 (\epsilon_2 - \epsilon_1 \sin^2 \theta)^{1/2} z'} \quad (\text{A.16})$$

and

$$\Phi_s = \frac{n_1 \tau_s}{(\epsilon_2 - \epsilon_1 \sin^2 \theta)^{1/2}}, \quad \tau_s = 1 - \Delta_s, \quad (\text{A.17})$$

$$\Phi_p = n_1 \tau_p (\epsilon_2 - \epsilon_1 \sin^2 \theta)^{1/2}, \quad \tau_p = \Delta_p + 1$$

where

$$\Delta_p = \frac{-n_1 \epsilon_2 \cos \theta - \epsilon_1 (\epsilon_2 - \epsilon_1 \sin^2 \theta)^{1/2}}{-n_1 \epsilon_2 \cos \theta + \epsilon_1 (\epsilon_2 - \epsilon_1 \sin^2 \theta)^{1/2}} \quad (\text{A.18})$$

$$\Delta_s = \frac{-n_1 \cos \theta - (\epsilon_2 - \epsilon_1 \sin^2 \theta)^{1/2}}{-n_1 \cos \theta + (\epsilon_2 - \epsilon_1 \sin^2 \theta)^{1/2}}.$$

In the far region, or propagating region, the wave vector, the electric field and magnetic field respect the right-hand rule. Therefore,

we easily calculate the magnetic field in this region by using the well-known equation

$$\mathbf{H} = \mathbf{k} \times \mathbf{E}. \quad (\text{A.19})$$

## Appendix B. 2D field susceptibilities

A further new feature in pyGDM is the possibility to simulate two-dimensional nanostructures like nanowires, which are infinitely long along one dimension. In this case, the geometry is discretized using “line-dipoles”, and light-emission is described by cylindrical waves.

We assume a nanostructure geometry which is “infinitely long”, hence is invariant along one axis. We will use the  $y$  axis in the following. In this case, all dependencies along this axis are identical to the time-harmonicity of the incident field and are therefore explicit. With  $\mathbf{r}^{2D} = x\hat{e}_x + z\hat{e}_z$  the two-dimensional electric polarization at a “3D” location  $\mathbf{r}$  can be written:

$$\mathbf{p}^{2D}(\mathbf{r}) = \mathbf{p}^{2D}(\mathbf{r}^{2D}) \exp(ik_{y,0}y) \quad (\text{B.1})$$

where  $k_{y,0}$  is the wavevector component along the infinite axis. In the 2D field calculations, we discretize the nanostructure into lines of identical dipoles, infinitely extending along the  $y$ -axis. Hence an integration along these dipole lines needs to be performed. This integration can be included in the 2D Green's Dyads. Using the Dyad  $\mathbf{G}_0^{EE}$  from equation (A.1), gives:

$$\begin{aligned} \mathbf{G}_0^{2D,EE}(\mathbf{r}^{2D}, \mathbf{r}'^{2D}) &= \\ &= \int_{-\infty}^{\infty} \mathbf{G}_0^{EE}(\mathbf{r}, \mathbf{r}') e^{ik_{y,0}y'} dy' \\ &= \frac{1}{\epsilon_{\text{env}}} \left( k^2 \mathbf{I} + \nabla \nabla \right) \int_{-\infty}^{\infty} G_0(\mathbf{r}, \mathbf{r}') e^{ik_{y,0}y'} dy' \end{aligned} \quad (\text{B.2})$$

where  $\mathbf{r}^{2D}$  and  $\mathbf{r}'^{2D}$  are the 2D positions of the observer and the emitting dipole-line, respectively.

The  $y'$ -integral over the vacuum Green's function can be expressed by the following identity via the zero-order Hankel function of the first kind  $H_0^{(1)}$  [85,86]:

$$\int_{-\infty}^{\infty} \frac{e^{ik\sqrt{R^2+(y-y')^2}}}{\sqrt{R^2+(y-y')^2}} e^{ik_{y,0}y'} dy' = i\pi H_0^{(1)}(k_r R) e^{ik_{y,0}y} \quad (\text{B.3})$$

where we define the in-plane distance as  $R = |\mathbf{R}^{2D}| = |\mathbf{r}^{2D} - \mathbf{r}'^{2D}|$ , and

$$k_r = \sqrt{k^2 - k_{y,0}^2}. \quad (\text{B.4})$$

Now, by including the explicit  $y$ -dependence in the differential operator for the 2D problem

$$\begin{aligned} \nabla &= \partial \hat{e}_x / \partial x + \partial \hat{e}_z / \partial z + ik_{y,0} \hat{e}_y \\ &= \nabla_\rho + ik_{y,0} \hat{e}_y \end{aligned} \quad (\text{B.5})$$

we obtain

$$\begin{aligned} \mathbf{G}_0^{2D,EE}(\mathbf{r}^{2D}, \mathbf{r}'^{2D}) &= \\ &= \frac{i\pi}{\epsilon_{\text{env}}} \begin{pmatrix} k^2 + \frac{\partial^2}{\partial x^2} & ik_{y,0} \frac{\partial}{\partial x} & \frac{\partial^2}{\partial x \partial z} \\ ik_{y,0} \frac{\partial}{\partial x} & k^2 - k_{y,0}^2 & ik_{y,0} \frac{\partial}{\partial z} \\ \frac{\partial^2}{\partial z \partial x} & ik_{y,0} \frac{\partial}{\partial z} & k^2 + \frac{\partial^2}{\partial z^2} \end{pmatrix} \\ &\quad \times H_0^{(1)}(k_r R) e^{ik_{y,0}y}. \end{aligned} \quad (\text{B.6})$$

With the derivatives of the  $n$ -th order Hankel functions of the first kind

$$\frac{d}{da} H_n^{(1)}(a) = \frac{n H_n^{(1)}(a)}{a} - H_{n+1}^{(1)}(a), \quad (\text{B.7})$$

and the recurrence relation

$$H_n^{(1)}(a) = \frac{a}{2n} \left( H_{(n-1)}^{(1)}(a) + H_{(n+1)}^{(1)}(a) \right), \quad (\text{B.8})$$

we arrive finally at the following set of components for the 2D Green's dyad, where we assume that the observer is located in the plane  $y = 0$ :

$$G_{0,xx}^{2D,EE} = i\pi \left( k_0^2 - k_r^2 / (2\epsilon_{\text{env}}) \right) H_0^{(1)}(k_r R) + i\pi \frac{k_r^2 (2(\Delta x/R)^2 - 1)}{2\epsilon_{\text{env}}} H_2^{(1)}(k_r R) \quad (\text{B.9})$$

$$G_{0,xy}^{2D,EE} = \pi \frac{k_r k_{y,0} \Delta x / R}{\epsilon_{\text{env}}} H_1^{(1)}(k_r R) \quad (\text{B.10})$$

$$G_{0,xz}^{2D,EE} = i\pi \frac{k_r^2 \Delta x \Delta z / R^2}{\epsilon_{\text{env}}} H_2^{(1)}(k_r R) \quad (\text{B.11})$$

$$G_{0,yy}^{2D,EE} = i\pi \left( k_0^2 - k_{y,0}^2 / \epsilon_{\text{env}} \right) H_0^{(1)}(k_r R) \quad (\text{B.12})$$

$$G_{0,yz}^{2D,EE} = \pi \frac{k_r k_{y,0} \Delta z / R}{\epsilon_{\text{env}}} H_1^{(1)}(k_r R) \quad (\text{B.13})$$

$$G_{0,zz}^{2D,EE} = i\pi \left( k_0^2 - k_r^2 / (2\epsilon_{\text{env}}) \right) H_0^{(1)}(k_r R) + i\pi \frac{k_r^2 (2(\Delta z/R)^2 - 1)}{2\epsilon_{\text{env}}} H_2^{(1)}(k_r R) \quad (\text{B.14})$$

$$\begin{aligned} G_{0,yx}^{2D,EE} &= G_{0,xy}^{2D,EE} \\ G_{0,zx}^{2D,EE} &= G_{0,xz}^{2D,EE} \\ G_{0,zy}^{2D,EE} &= G_{0,yz}^{2D,EE} \end{aligned} \quad (\text{B.15})$$

with  $\Delta x = x - x'$  and  $\Delta z = z - z'$ .

Analogously, for the mixed electric-magnetic Green's tensor we obtain:

$$\begin{aligned} \mathbf{G}_0^{2D,HE}(\mathbf{r}^{2D}, \mathbf{r}',^{2D}) &= \\ &= \int_{-\infty}^{\infty} \mathbf{G}_0^{HE}(\mathbf{r}, \mathbf{r}') e^{ik_{y,0} y'} dy' \\ &= -ik_0 \nabla \times \int_{-\infty}^{\infty} G_0(\mathbf{r}, \mathbf{r}') e^{ik_{y,0} y'} dy' \\ &= k_0 \pi \begin{pmatrix} 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \end{pmatrix} H_0^{(1)}(k_r R) e^{ik_{y,0} y} \end{aligned} \quad (\text{B.16})$$

which explicitly gives

$$G_{0,xy}^{2D,HE} = \pi \frac{k_0 k_r \Delta z}{R} H_1^{(1)}(k_r R) \quad (\text{B.17a})$$

$$G_{0,xz}^{2D,HE} = i\pi k_0 k_{y,0} H_0^{(1)}(k_r R) \quad (\text{B.17b})$$

$$G_{0,yz}^{2D,HE} = \pi \frac{k_0 k_r \Delta x}{R} H_1^{(1)}(k_r R) \quad (\text{B.17c})$$

$$G_{0,yx}^{2D,HE} = -G_{0,xy}^{2D,HE} \quad (\text{B.17d})$$

$$G_{0,zx}^{2D,HE} = -G_{0,xz}^{2D,HE} \quad (\text{B.17e})$$

$$G_{0,zy}^{2D,HE} = -G_{0,yz}^{2D,HE} \quad (\text{B.17f})$$

$$G_{0,xx}^{2D,HE} = G_{0,yy}^{2D,HE} = G_{0,zz}^{2D,HE} = 0. \quad (\text{B.17g})$$

Without loss of generality, 2D simulations in pyGDM are restricted to the  $y = 0$  plane. For the evaluation of the fields at positions  $y \neq 0$ , the tensors simply need to be multiplied by the  $y$ -dependent phase factor of the incident field.

### B.1. 2D self-terms

The electric-electric self-term components write [85]:

$$\text{norm}_x = \text{norm}_z = \frac{-4\pi}{2\epsilon_{\text{env}} S_{\text{cell}}} + 4i\pi^2 \left( 2k_0^2 - \frac{k_r^2}{\epsilon_{\text{env}}} \right) \frac{H_1^{(1)}(k_r d / \sqrt{\pi}) + 2i/(\pi k_r)}{4S_{\text{cell}} k_r} \quad (\text{B.18})$$

and

$$\text{norm}_y = 4i\pi^2 \left( k_0^2 - \frac{k_{y,0}^2}{\epsilon_{\text{env}}} \right) \frac{H_1^{(1)}(k_r d / \sqrt{\pi}) + 2i/(\pi k_r)}{2S_{\text{cell}} k_r} \quad (\text{B.19})$$

where  $S_{\text{cell}} = d^2$  is the surface covered by a square mesh-cell with side length  $d$ . Similar to the 3D Green's tensors, the magnetic-electric self-terms equal zero. Note that in the current version of pyGDM, 2D simulations are only possible on a square mesh.

## Appendix C. Fast electron illumination: computation of electron energy losses and cathodoluminescence

The pyGDM toolbox now provides built-in routines to simulate the results of Electron Energy Loss Spectroscopy (EELS) or Cathodoluminescence (CL) experiments.

We consider in the following the situation represented in Fig. C.19. A fast electron with charge  $(-e)$  is traveling along the (OZ) axis towards positive  $z$  i.e.  $\mathbf{v} = +v\mathbf{e}_z$ . The electron passes in the vicinity of a nanostructure. It is incident normally on the sample plane at coordinates  $\mathbf{R}_{\perp} = (x_e, y_e)$ .

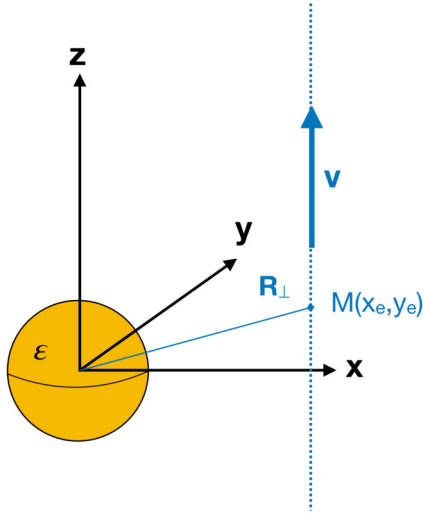
Classically, the energy losses experienced by a fast electron are interpreted as follows: (i) A moving charge is the source of an electromagnetic field which polarizes the neighboring nano-objects. (ii) In return, the polarized nano-objects radiate an electric field which acts on the electron movement. The energy losses are a due to the work of the Lorentz force on the moving charge. The loss probability per incident electron per unit angular frequency is given by the following formula [6]:

$$\Gamma(\omega) = \frac{e}{\hbar\pi\omega} \int \text{Re} \left( e^{-i\omega t} \mathbf{v} \cdot \mathbf{E}_{\text{ind}}(\mathbf{r}(t), \omega) \right) dt \quad (\text{C.1})$$

$\Gamma(\omega)$  can be directly computed by (i) repropagating the electric field at each point of the electron trajectory and (ii) summing up the different contributions to the energy losses along the trajectory. In a previous version of our simulation tools, this was done using adaptative discretization of the trajectory of the electron [55]. In the following, we derive a concise expression for the energy losses in the framework of the Green Dyadic Method. This expression is now used in the EELS function of the pyGDM toolbox enabling rapid computations of EELS spectra and maps.

### C.1. Electron energy losses using the Green dyadic method

The following alternative strategy can be adopted to obtain  $\Gamma(\omega)$ . The induced electric field can be expressed as a function of the vacuum propagator  $S_0$  and the polarization distribution inside the nanoparticle  $\mathbf{p}(\mathbf{r}', \omega)$  [38]:



**Fig. C.19.** A fast electron is incident on a nanostructure and crosses the (OXY) plane at  $\mathbf{R}_\perp = (x_e, y_e)$ . We compute the electron energy losses experienced by the swift particle in the framework of the Green Dyadic Method.

$$\begin{aligned}\Gamma(\omega) &= \frac{e}{\hbar\pi\omega} \int \text{Re} \left( e^{-i\omega t} \mathbf{v} \cdot \mathbf{E}_{\text{ind}}(\mathbf{r}(\mathbf{t}), \omega) \right) dt \\ &= \frac{e}{\hbar\pi\omega} \int_{-\infty}^{+\infty} \text{Re} \left\{ e^{-i\omega t} \mathbf{v} \cdot \left( \int_{V'} S_0(\mathbf{r}(\mathbf{t}), \mathbf{r}', \omega) \cdot \mathbf{p}(\mathbf{r}', \omega) d\mathbf{r}' \right) \right\} dt\end{aligned}$$

In the latter,  $V'$  is the volume of the nanoparticle. We can then exchange the integration on time and  $\mathbf{r}'$ . Taking into account the fact that  $z = vt$  we have:

$$\begin{aligned}\Gamma(\omega) &= \frac{ev}{\hbar\pi\omega} \text{Re} \left\{ \int_{V'} \mathbf{e}_z \cdot \left( \int_t e^{-i\omega z(t)/v} S_0((\mathbf{R}, z(t)), \mathbf{r}', \omega) dt \right) \cdot \mathbf{p}(\mathbf{r}', \omega) d\mathbf{r}' \right\}\end{aligned}$$

The electric-electric vacuum Green's tensor is given by (Eq. (A.1)):

$$S_0(\mathbf{r}, \mathbf{r}', \omega) = \mathbf{G}_0^{\text{EE}}(\mathbf{r}, \mathbf{r}', \omega) = \left\{ k_0^2 \mathbf{I} + \frac{1}{\varepsilon} \nabla \nabla \right\} \frac{e^{ik|\mathbf{r}-\mathbf{r}'|}}{|\mathbf{r}-\mathbf{r}'|} \quad (\text{C.2})$$

which leads to:

$$\begin{aligned}\Gamma(\omega) &= \frac{e}{\hbar\pi\omega} \text{Re} \left\{ \int_{V'} \mathbf{e}_z \cdot \left( \int_z e^{-i\omega z/v} \left\{ k_0^2 \mathbf{I} + \frac{1}{\varepsilon} \nabla \nabla \right\} \frac{e^{ik|\mathbf{r}-\mathbf{r}'|}}{|\mathbf{r}-\mathbf{r}'|} dz \right) \cdot \mathbf{p}(\mathbf{r}', \omega) d\mathbf{r}' \right\}\end{aligned}$$

or more explicitly:

$$\begin{aligned}\Gamma(\omega) &= \frac{e}{\hbar\pi\omega} \text{Re} \left\{ \int_{V'} \left( \int_z e^{-i\omega z/v} \mathbf{e}_z \cdot \begin{bmatrix} k_0^2 + \frac{1}{\varepsilon} \frac{\partial^2}{\partial x^2} & \frac{1}{\varepsilon} \frac{\partial^2}{\partial x \partial y} & \frac{1}{\varepsilon} \frac{\partial^2}{\partial x \partial z} \\ \frac{1}{\varepsilon} \frac{\partial^2}{\partial y \partial x} & k_0^2 + \frac{1}{\varepsilon} \frac{\partial^2}{\partial y^2} & \frac{1}{\varepsilon} \frac{\partial^2}{\partial y \partial z} \\ \frac{1}{\varepsilon} \frac{\partial^2}{\partial z \partial x} & \frac{1}{\varepsilon} \frac{\partial^2}{\partial z \partial y} & k_0^2 + \frac{1}{\varepsilon} \frac{\partial^2}{\partial z^2} \end{bmatrix} \right) \cdot \mathbf{p}(\mathbf{r}', \omega) d\mathbf{r}' \right\}\end{aligned}$$

$$\left. \frac{e^{ik|\mathbf{r}-\mathbf{r}'|}}{|\mathbf{r}-\mathbf{r}'|} dz \cdot \begin{bmatrix} p_x(\mathbf{r}', \omega) \\ p_y(\mathbf{r}', \omega) \\ p_z(\mathbf{r}', \omega) \end{bmatrix} \right) d\mathbf{r}' \Bigg\}$$

The term between parenthesis can be expressed as:

$$\begin{aligned}A &= A_1 + A_2 + A_3 + A_4 \\ &= \frac{1}{\varepsilon} p_x \int_{-\infty}^{+\infty} e^{-i\omega z/v} \frac{\partial^2}{\partial z \partial x} \left( \frac{e^{ikR}}{R} \right) dz \\ &\quad + \frac{1}{\varepsilon} p_y \int_{-\infty}^{+\infty} e^{-i\omega z/v} \frac{\partial^2}{\partial z \partial y} \left( \frac{e^{ikR}}{R} \right) dz \\ &\quad + k_0^2 p_z \int_{-\infty}^{+\infty} e^{-i\omega z/v} \frac{e^{ikR}}{R} dz \\ &\quad + \frac{1}{\varepsilon} p_z \int_{-\infty}^{+\infty} e^{-i\omega z/v} \frac{\partial^2}{\partial z^2} \left( \frac{e^{ikR}}{R} \right) dz\end{aligned}$$

Each of these 4 terms  $A_i$  can be calculated by integration by parts and connected to the following integral:

$$I = \int_{-\infty}^{+\infty} e^{-i\omega z/v} \frac{e^{ikR}}{R} dz$$

which gives

$$I = 2e^{-i\omega z'/v} K_0 \left( \sqrt{\left( \frac{\omega}{v} \right)^2 - k^2} |\mathbf{R} - \mathbf{R}'| \right) \quad (\text{C.3})$$

with  $R = |\mathbf{r} - \mathbf{r}'|$  and  $K_n$  the modified Bessel function of the second kind and order  $n$ . We get the following expressions for the  $A_i$  terms:

$$A_1 = -2i e^{-i\omega z'/v} \frac{\omega^2}{\varepsilon \gamma v^2} \frac{x - x'}{|\mathbf{R} - \mathbf{R}'|} K_1(k_\rho |\mathbf{R} - \mathbf{R}'|) p_x \quad (\text{C.4})$$

$$A_2 = -2i e^{-i\omega z'/v} \frac{\omega^2}{\varepsilon \gamma v^2} \frac{y - y'}{|\mathbf{R} - \mathbf{R}'|} K_1(k_\rho |\mathbf{R} - \mathbf{R}'|) p_y \quad (\text{C.5})$$

$$A_3 = 2k_0^2 e^{-i\omega z'/v} K_0(k_\rho |\mathbf{R} - \mathbf{R}'|) p_z \quad (\text{C.6})$$

$$A_4 = -2 \left( \frac{\omega^2}{\varepsilon v^2} \right) e^{-i\omega z'/v} K_0(k_\rho |\mathbf{R} - \mathbf{R}'|) p_z \quad (\text{C.7})$$

with  $k_z = \omega/v$ ,  $k_\rho = \sqrt{k_z^2 - k^2}$ ,  $k = k_0 \sqrt{\varepsilon}$ ,  $\gamma = 1/\sqrt{1 - \varepsilon v^2/c^2}$  and  $k_\rho = \omega/v\gamma$ .

After some algebra we obtain finally the following expression for the energy losses:

$$\begin{aligned}\Gamma(\omega) &= \frac{2e}{\hbar\pi\omega} \text{Re} \left\{ \int_{V'} e^{-i\omega z'/v} \left[ \begin{aligned} & - \left( \frac{\omega^2}{\varepsilon \gamma^2 v^2} \right) K_0(k_\rho |\mathbf{R} - \mathbf{R}'|) p_z \\ & - i \frac{\omega^2}{\varepsilon \gamma v^2} \frac{p_x(x - x') + p_y(y - y')}{|\mathbf{R} - \mathbf{R}'|} \\ & K_1(k_\rho |\mathbf{R} - \mathbf{R}'|) \end{aligned} \right] d\mathbf{r}' \right\}\end{aligned}$$



In GDM, the used Fourier transform is defined with a  $2\pi$  factor difference from the convention used for instance by J. Garcia de Abajo (see [55] for more information). The latter defines the Fourier transform as:

$$E(r, t) = \int \frac{d\omega}{2\pi} E(r, \omega) e^{-i\omega t}$$

$$E(r, \omega) = \int dt E(r, t) e^{i\omega t}$$

whereas in the GDM it is defined as:

$$E(r, t) = \int d\omega E(r, \omega) e^{-i\omega t}$$

$$E(r, \omega) = \frac{1}{2\pi} \int dt E(r, t) e^{i\omega t}$$

The formula for the loss probability per unit frequency that is implemented in pyGDM is therefore:

$$\Gamma(\omega) = \frac{4e}{\hbar\omega} \text{Re} \left\{ \int_{V'} e^{-i\omega z'/v} \left[ - \left( \frac{\omega^2}{\varepsilon \gamma^2 v^2} \right) K_0(k_\rho |\mathbf{R} - \mathbf{R}'|) p_z - i \frac{\omega^2}{\varepsilon \gamma v^2} \frac{p_x(x - x') + p_y(y - y')}{|\mathbf{R} - \mathbf{R}'|} K_1(k_\rho |\mathbf{R} - \mathbf{R}'|) \right] d\mathbf{r}' \right\} \quad (\text{C.8})$$

It connects the electron energy losses to the three components of the polarization induced inside the nanostructure by the moving charge.

Please note that equation (C.8) diverges for meshpoints on the beam trajectory because of the divergence of the modified Bessel function of the second kind (see also [55]). To calculate equation (C.8) for an electron beam passing through the structure, we therefore simply remove all meshpoints from the numerical integration, that are closer to the beam than one discretization step. Furthermore, in order to avoid numerical noise due to the discrete mesh when scanning the beam above the structure, we added a new function `tools.adapt_map_to_structure_mesh` to pyGDM, which adjusts all (X,Y) raster-scan positions onto the closest mesh coordinate. We recommend using this tool for EELS and CL raster-scan simulations. The example scripts contain an accompanying description.

We also want to mention that electron propagation through a lossy material yields additional losses, the probability of which is described by the so-called “bulk loss probability”  $\Gamma_{\text{bulk}}(\omega)$ , that is proportional to the thickness of the traversed material and to the imaginary part of  $1/\varepsilon(\omega)$ . This contribution is not taken into account in the pyGDM simulations, but it could be straightforwardly computed, if required. More details can be found in Refs. [6,87].

In the next section we give a simpler expression of this result for a dipolar particle.

## C.2. Case of a very small particle

We demonstrate the calculation now for the case of a small nanoparticle that can be approximated by a polarizability  $\alpha$  located at the origin. Then we can write the components of the polarization induced inside this nanoparticle by an electron beam crossing the (OXY) plane at location (d, 0) as:

$$\mathbf{p} = \alpha \mathbf{E}_{\text{el}}$$

The electric field  $\mathbf{E}_{\text{el}}$  created at the location of the nano-object by a swift electron travelling along (OZ) towards positive z is:

$$p_x = \frac{e\omega}{\pi v^2 \gamma \varepsilon} \alpha K_1 \left( \frac{\omega d}{\gamma v} \right)$$

$$p_y = 0$$

$$p_z = i \frac{e\omega}{\pi v^2 \gamma^2 \varepsilon} \alpha K_0 \left( \frac{\omega d}{\gamma v} \right)$$

Inserting these expressions in formula (C.8) yields:

$$\begin{aligned} \Gamma(\omega) &= \frac{4e}{\hbar\omega} \text{Re} \left\{ \left[ - \left( \frac{\omega^2}{\varepsilon \gamma^2 v^2} \right) K_0(k_\rho d) p_z - i \frac{\omega^2}{\varepsilon \gamma v^2} (p_x K_1(k_\rho d)) \right] \right\} \\ &= \frac{4e}{\hbar\omega} \text{Re} \left\{ \left[ - i \frac{e\omega}{\pi v^2 \gamma^2 \varepsilon} \alpha \left( \frac{\omega^2}{\varepsilon \gamma^2 v^2} \right) K_0^2(k_\rho d) - i \frac{e\omega}{\pi v^2 \gamma \varepsilon} \alpha \frac{\omega^2}{\varepsilon \gamma v^2} K_1^2(k_\rho d) \right] \right\} \\ \Gamma(\omega) &= \frac{1}{\pi \hbar} \left( \frac{2e\omega}{v^2 \gamma \varepsilon} \right)^2 \frac{K_0^2(k_\rho d)}{\gamma^2} \text{Im}(\alpha) + \frac{1}{\pi \hbar} \left( \frac{2e\omega}{v^2 \gamma \varepsilon} \right)^2 K_1^2(k_\rho d) \text{Im}(\alpha) \end{aligned}$$

Which finally leads to

$$\Gamma(\omega) = \frac{1}{\pi \hbar} \left( \frac{2e\omega}{v^2 \gamma \varepsilon} \right)^2 \text{Im}(\alpha) \left\{ K_1^2(k_\rho d) + \frac{K_0^2(k_\rho d)}{\gamma^2} \right\} \quad (\text{C.9})$$

We see here that assuming a dipolar response for the nanoparticle in formula (C.8) yields identical results as formulas (34) and (35) of reference [6].

The computation of cathodoluminescence spectra and maps follows the same lines as detailed in reference [55]. In short, pyGDM computes the power radiated by the polarization distribution, induced inside the sample by the moving charge from the integral of the Poynting vector on a sphere, centered on the sample.

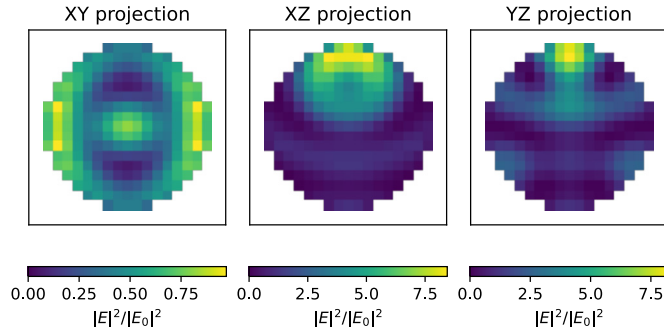
## Appendix D. Technical details

### D.1. Simulation size

The maximum number of discretization cells depends on the available RAM. The memory requirement scales with the square of the discretization points. 8GB of RAM are enough to treat around 7000 meshpoints. 32GB can host around 15k meshpoints, and 128GB will suffice for around 30k meshpoints. The computation time scales with the cube of the number of cells, therefore for larger structures the simulation time will become the bottleneck in systems with enough memory. For instance, a model with 25k meshpoints takes several days per wavelength on a recent 36 core intel server CPU. In consequence the number of meshpoints for practical simulations is limited to  $\approx 10000 - 15000$ .

### D.2. Dependencies

pyGDM is written in pure python. It uses several third-party tools and libraries for acceleration, processing and visualization. All dependencies are exclusively open-source and can be installed via “pip”:



**Fig. D.20.** Graphical output of the example script given in Listing 1. A dielectric sphere ( $n = 2$ ,  $r = 160$  nm) in vacuum is illuminated from below by a plane wave of linear polarization along  $X$  and wavelength  $\lambda_0 = 400$  nm. Each plot shows the E-field intensity inside the sphere on a slice through its center on an area of  $400 \times 400$  nm<sup>2</sup>.

#### Required third party python packages

- “numba” (just-in-time compiler for acceleration)
- “numpy” (numerical stack)
- “scipy” (various efficient scientific algorithm)

#### Optional python packages

- “matplotlib” (for 2D visualizations)
- “mayavi” (for 3D visualizations)
- “cupy” (CUDA-based GPU solver)
- “pytables” (to save simulations results in the efficient hdf5 data format)
- “mpi4py” (for MPI parallelized spectra calculation)
- “pyGMO” (required by the evolutionary optimization submodule)
- “PIL” (required by some image processing tools)

#### D.3. Compiling, installation

We provide the latest stable version via the pypi repository, installation is therefore easiest via “pip”:

```
pip install -U pygdm2
```

To install the latest development version, download the source code from the gitlab repository and run in a terminal in the pyGDM sources root directory

```
pip install -U.
```

Attention, the trailing point “.” is significant in the command, pointing to the current working directory. For a local installation, add “--user”.

#### D.4. Compiling, installation of the retarded Green's dyads module

The retarded dyads module `pyGDM2-retard` depends on fortran based binaries and is available as an external python package in order to avoid a binary dependency in the main pyGDM package. For windows we provide pre-compiled packages of the latest stable version. On other operating systems like linux the package needs to be compiled. Installation is easiest via “pip”:

```
pip install -U pyGDM2-retard
```

If no binaries are available on your platform, the code must be locally compiled. If the gfortran compiler is correctly installed this should be done automatically by pip and installation should be possible with the above command. Details and tips concerning the installation can be found in the [pyGDM online documentation](#).

#### D.5. Installation of the graphical user interface pyGDM-UI

We currently work on a [graphical user interface for pyGDM](#) based on Qt, it therefore requires `pyqt5`. Currently, also `mayavi` is required, as it is used for several visualizations. `pyGDM-UI` is still under development but we provide a working beta-version. It can be installed from `pypi` via `pip`

```
pip install -U pygdmUI
```

#### D.6. Minimum working example script

```
1 from pyGDM2 import materials
2 from pyGDM2 import fields
3 from pyGDM2 import propagators
4 from pyGDM2 import core
5 from pyGDM2 import visu
6
7 ## --- simulation setup ---
8 ## structure: sphere of 160nm radius,
9 ## constant dielectric function (n=2),
10 step = 20
11 geometry = structures.sphere(step, R=8.2, mesh='cube')
12 material = materials.dummy(2.0)
13 struct = structures.struct(step, geometry, material)
14
15 ## incident field: plane wave, 400nm, lin. pol. || x
16 field_generator = fields.plane_wave
17 wavelengths = [400] # nm
18 kwargs = dict(inc_angle=0, inc_plane='xz', theta=0)
19 efield = fields.efield(field_generator,
20 wavelengths=wavelengths, kwargs=kwargs)
21
22 ## environment: vacuum
23 n1 = 1.0
24 dyads = propagators.DyadsQuasistatic123(n1)
25
26 ## create simulation object
27 sim = core.simulation(struct, efield, dyads)
28
29
30 ## --- run the simulation ---
31 core.scatter(sim)
32
33
34 ## --- plot nearfield intensity inside sphere ---
35 ## using first (of one) field-config (=index 0)
36 ## slice through sphere center
37 visu.vectorfield_color_by_fieldindex(sim, 0,
38 projection='XY', slice_level=160)
39 visu.vectorfield_color_by_fieldindex(sim, 0,
40 projection='XZ', slice_level=0)
41 visu.vectorfield_color_by_fieldindex(sim, 0,
42 projection='YZ', slice_level=0)
```

Listing 1: Minimum example script. The plots generated by the script are shown in Fig. D.20.

#### D.7. Technical documentation of main classes and functions

For a detailed documentation of all functions and classes as well as many commented examples and tutorials, please visit the online documentation <https://wiechapeter.gitlab.io/pyGDM2-doc>. We provide here merely a very brief description of the core classes and functions.

#### D.8. Main classes

**core.simulation**  
(class)

constructor arguments:

- `struct`: instance of **structures.struct**
- `efield`: instance of **fields.efield**
- `dyads`: class defining simulation environment

**structures.struct**

(class)

constructor arguments:

- *step*: discretization stepsize (in nm)
- *geometry*: list of meshpoint coordinates (x, y, z) (in nm)
- *material*: structure material dispersion, instance of **materials.CLASS** or list of materials (one for each meshpoint)

**fields.efield**

(class)

constructor arguments:

- *field\_generator*: field generator function (e.g. from **fields** module)
- *wavelengths*: list of wavelengths at which to do the simulation (in nm)
- *kwargs* (optional): dict (or list of dict) with further kwargs for the field generator

**propagators.DyadsQuasistatic123**

(class)

constructor arguments:

- *n1*: refractive index for layer 1 (substrate)
- *n2*: ref. index for layer 2 (center layer)
- *n3*: ref. index for layer 3 (top cladding)
- *spacing*: thickness of layer 2
- *radiative\_correction*: whether or not to use the radiative correction term

**D.9. Main core functions****core.scatter / core.scatter\_mpi**

(function)

arguments:

- *sim*: instance of **core.simulation**
- *method* (optional): inversion method, default: "lu"
- *calc\_H* (optional): calculate internal magnetic field, default: "False"

**core.decay\_rate**

(function)

arguments:

- *sim*: instance of **core.simulation**
- *wavelength* target wavelength
- *r\_probe* list of coordinates to evaluate
- *method* (optional): inversion method, default: "scipyinv"

**References**

- [1] J.C. Maxwell, Philos. Trans. R. Soc. Lond. 155 (1865) 459–512.
- [2] A.F. Oskooi, et al., Comput. Phys. Commun. 181 (2010) 687–702.
- [3] B.T. Draine, P.J. Flatau, J. Opt. Soc. Am. A 11 (1994) 1491.
- [4] P.C. Chaumet, A. Rahmani, J. Quant. Spectrosc. Radiat. Transf. 110 (2009) 22–29.
- [5] U. Hohenester, A. Trügler, Comput. Phys. Commun. 183 (2012) 370–381.
- [6] F.J. García de Abajo, Rev. Mod. Phys. 82 (2010) 209–275, arXiv:0903.1669.
- [7] G. Demésy, F. Zolla, A. Nicolet, M. Commandré, JOSA A 27 (2010) 878–889.
- [8] J. Hoffmann, C. Hafner, P. Leidenberger, J. Hesselbarth, S. Burger, in: Modeling Aspects in Optical Metrology II, vol. 7390, International Society for Optics and Photonics, 2009, p. 73900J.
- [9] D. Barchiesi, C. Girard, O.J.F. Martin, D. Van Labeke, D. Courjon, Phys. Rev. E 54 (1996) 4285–4292.
- [10] J. Parsons, C.P. Burrows, J.R. Sambles, W.L. Barnes, J. Mod. Opt. 57 (2010) 356–365.
- [11] B. Gallinet, J. Butet, O.J.F. Martin, Laser Photonics Rev. 9 (2015) 577–603.
- [12] A.I. Kuznetsov, A.E. Miroshnichenko, M.L. Brongersma, Y.S. Kivshar, B. Luk'yanchuk, Science 354 (2016).
- [13] P. Mühlischlegel, H.-J. Eisler, O.J.F. Martin, B. Hecht, D.W. Pohl, Science 308 (2005) 1607–1609.
- [14] P. Bharadwaj, B. Deutsch, L. Novotny, Adv. Opt. Photonics 1 (2009) 438–483.
- [15] P.R. Wiecha, et al., Sci. Rep. 7 (2017) 40906.
- [16] C. Girard, P.R. Wiecha, A. Cuche, E. Dujardin, J. Opt. 20 (2018) 075004.
- [17] C. Girard, E. Dujardin, G. Baffou, R. Quidant, New J. Phys. 10 (2008) 105016.
- [18] S. Viarbitskaya, et al., Nat. Mater. 12 (2013) 426–432.
- [19] S. Viarbitskaya, et al., Appl. Phys. Lett. 103 (2013) 131112.
- [20] Y. Ould Agha, O. Demichel, C. Girard, A. Bouhelier, G. Colas des Francs, Prog. Electromagn. Res. 146 (2014) 77–88.
- [21] A. Cuche, et al., Opt. Commun. 387 (2017) 48–54.
- [22] A. Cuche, et al., Phys. Rev. B 95 (2017) 121402.
- [23] P.R. Wiecha, et al., Nat. Nanotechnol. 12 (2017) 163–169.
- [24] P.R. Wiecha, et al., ACS Photonics 4 (2017) 2036–2046.
- [25] P.R. Wiecha, A. Arbouet, C. Girard, T. Baron, V. Paillard, Phys. Rev. B 93 (2016) 125421.
- [26] L.-J. Black, et al., ACS Photonics 2 (2015) 1592–1601.
- [27] P.R. Wiecha, Linear and Nonlinear Optical Properties of High Refractive Index Dielectric Nanostructures, PhD thesis, Université de Toulouse, Université Toulouse III - Paul Sabatier, 2016.
- [28] S.K. Lam, A. Pitrou, S. Seibert, in: Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC, LLVM '15, Association for Computing Machinery, New York, NY, USA, 2015, pp. 1–6.
- [29] O.J.F. Martin, C. Girard, A. Dereux, Phys. Rev. Lett. 74 (1995) 526–529.
- [30] P.R. Wiecha, Comput. Phys. Commun. 233 (2018) 167–192.
- [31] G.S. Agarwal, Phys. Rev. A 11 (1975) 230–242.
- [32] J.D. Jackson, Classical Electrodynamics, third edn., Wiley, 1999.
- [33] A. Patoux, et al., Phys. Rev. B 101 (2020) 235418, arXiv:1912.04124.
- [34] I. Sersic, C. Tuambalangana, T. Kampfrath, A.F. Koenderink, Phys. Rev. B 83 (2011) 245102.
- [35] U. Schröter, Eur. Phys. J. B 33 (2003) 297–310.
- [36] C. Girard, J.-C. Weeber, A. Dereux, O.J.F. Martin, J.-P. Goudonnet, Phys. Rev. B 55 (1997) 16487–16497.
- [37] G. Colas des Francs, Optique Sub-Longueur d'onde et Fluorescence Moléculaire Perturbée, PhD thesis, Université Paul Sabatier Toulouse, CEMES-CNRS, 2002.
- [38] C. Girard, Rep. Prog. Phys. 68 (2005) 1883–1933.
- [39] L. Novotny, B. Hecht, Principles of Nano-Optics, Cambridge University Press, Cambridge, New York, 2006.
- [40] P. Gay-Balmaz, O.J.F. Martin, Opt. Commun. 184 (2000) 37–47.
- [41] L. Novotny, B. Hecht, D.W. Pohl, J. Appl. Phys. 81 (1997) 1798–1806.
- [42] A. Rahmani, P.C. Chaumet, F. de Fornel, C. Girard, Phys. Rev. A 56 (1997) 3245–3254.
- [43] M. Paulus, P. Gay-Balmaz, O.J.F. Martin, Phys. Rev. E 62 (2000) 5797–5807.
- [44] J. Kottmann, O. Martin, IEEE Trans. Antennas Propag. 48 (2000) 1719–1726.
- [45] D.A. Smunov, P.C. Chaumet, M.A. Yurkin, J. Quant. Spectrosc. Radiat. Transf. 156 (2015) 67–79.
- [46] J.B. Pendry, Phys. Rev. Lett. 85 (2000) 3966–3969.
- [47] A.B. Evlyukhin, C. Reinhardt, B.N. Chichkov, Phys. Rev. B 84 (2011) 235429.
- [48] T.H. Cormen, T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, MIT Press, 2001.
- [49] C. Wang, et al., J. Appl. Phys. 115 (2014) 244312.
- [50] R.M. Bakker, et al., Nano Lett. 15 (2015) 2137–2142.
- [51] P. Albella, et al., J. Phys. Chem. C 117 (2013) 13573–13584.
- [52] G. Colas des Francs, et al., Phys. Rev. B 80 (2009) 115419.
- [53] J. Barthes, G. Colas des Francs, A. Bouhelier, J.-C. Weeber, A. Dereux, Phys. Rev. B 84 (2011) 073403.
- [54] A. Campos, et al., ACS Photonics 4 (2017) 1257–1263.
- [55] A. Arbouet, A. Mlayah, C. Girard, G. Colas des Francs, New J. Phys. 16 (2014) 113012.
- [56] M. Schäferling, X. Yin, H. Giessen, Opt. Express 20 (2012) 26326–26336.
- [57] N. Meinzer, E. Hendry, W.L. Barnes, Phys. Rev. B 88 (2013) 041407.
- [58] Y. Gorodetski, A. Drezet, C. Genet, T.W. Ebbesen, Phys. Rev. Lett. 110 (2013) 203906.
- [59] Y. Tang, A.E. Cohen, Phys. Rev. Lett. 104 (2010) 163901.
- [60] P.C. Chaumet, M. Nieto-Vesperinas, Phys. Rev. B 61 (2000) 14119–14127.
- [61] R. Marty, G. Baffou, A. Arbouet, C. Girard, R. Quidant, Opt. Express 18 (2010) 3035–3044.
- [62] C. Girard, A. Dereux, O.J.F. Martin, Phys. Rev. B 49 (1994) 13872–13881.
- [63] W. Press, Numerical Recipes 3rd Edition: The Art of Scientific Computing, Cambridge University Press, 2007.
- [64] A.G. Curto, et al., Science 329 (2010) 930–933.
- [65] T.H. Taminiau, F.D. Stefani, N.F. van Hulst, Opt. Express 16 (2008) 10858–10866.
- [66] P.R. Wiecha, et al., Opt. Express 27 (2019) 29069–29081.
- [67] P.R. Wiecha, A. Arbouet, A. Cuche, V. Paillard, C. Girard, Phys. Rev. B 97 (2018) 085411.
- [68] C. Majorel, C. Girard, A. Cuche, A. Arbouet, P.R. Wiecha, J. Opt. Soc. Am. B 37 (2020) 1474–1484, arXiv:1912.06023.

- [69] R. Carminati, et al., *Surf. Sci. Rep.* 70 (2015) 1–41.
- [70] Y.S. Kim, P.T. Leung, T.F. George, *Surf. Sci.* 195 (1988) 1–14.
- [71] M.K. Schmidt, et al., *Opt. Express* 20 (2012) 13636.
- [72] J.M. Luther, P.K. Jain, T. Ewers, A.P. Alivisatos, *Nat. Mater.* 10 (2011) 361–366.
- [73] C. Majorel, et al., *Opt. Commun.* 453 (2019) 124336.
- [74] A. Teulle, R. Marty, C. Girard, A. Arbouet, E. Dujardin, *Opt. Commun.* 291 (2013) 412–415.
- [75] U. Manna, et al., *J. Appl. Phys.* 127 (2020) 033101.
- [76] P.C. Chaumet, A. Rahmani, G.W. Bryant, *Phys. Rev. B* 67 (2003) 165404.
- [77] M. Paulus, O.J.F. Martin, *Phys. Rev. E* 63 (2001) 066615.
- [78] G. Baffou, C. Girard, E. Dujardin, G. Colas des Francs, O.J.F. Martin, *Phys. Rev. B* 77 (2008) 121101.
- [79] A. Arbouet, F. Houdellier, R. Marty, C. Girard, *J. Appl. Phys.* 112 (2012) 053103.
- [80] N.K. Govindaraju, B. Lloyd, Y. Dotsenko, B. Smith, J. Manferdelli, in: *SC '08: Proceedings of the 2008 ACM/IEEE Conference on Supercomputing*, 2008, pp. 1–12.
- [81] M. Huntemann, G. Heygster, G. Hong, *J. Comput. Sci.* 2 (2011) 262–271.
- [82] P.R. Wiecha, O.L. Muskens, *Nano Lett.* 20 (2020) 329–338, arXiv:1909.12056.
- [83] G. Colas des Francs, C. Girard, A. Dereux, *J. Chem. Phys.* 117 (2002) 4659–4666.
- [84] M. Born, E. Wolf, *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light*, seventh edn., Cambridge University Press, Cambridge, 1999.
- [85] O.J.F. Martin, N.B. Piller, *Phys. Rev. E* 58 (1998) 3909–3915.
- [86] I.S. Gradshteyn, I.M. Ryzhik, *Table of Integrals, Series, and Products*, seventh edn., Elsevier/Academic Press, Amsterdam, 2007.
- [87] G.D. Bernasconi, et al., *ACS Photonics* 4 (2017) 156–164.