treams – a T-matrix-based scattering code for nanophotonics

Dominik Beutel[a,*], Ivan Fernandez-Corbaton[b], Carsten Rockstuhl[a,b]

[a] *Institute of Theoretical Solid State Physics, Karlsruhe Institute of Technology (KIT), 76131 Karlsruhe, Germany*
[b] *Institute of Nanotechnology, Karlsruhe Institute of Technology (KIT), 76344 Eggenstein-Leopoldshafen, Germany*

Keywords:
T-matrix
Electromagnetic scattering
Lattice sums
Chiral media

A B S T R A C T

We report the publication of *treams*, a new software for electromagnetic scattering computations based on the T-matrix method. Besides conventional T-matrix calculations for individual scatterers and finite clusters of particles, a unique feature of the code is its full support for periodic boundaries in one, two, and all three spatial dimensions. We use highly efficient and quickly converging lattice summation techniques based on the Ewald method to evaluate the arising lattice sums in these cases. In addition to the common use of vector spherical waves as a basis set for the T-matrix, vector cylindrical waves are also implemented. To describe stratified media, vector plane waves are used with an S-matrix description of the electromagnetic scattering. All basis sets and the associated methods can be used together with chiral constitutive relations. Thereby, chiral embedding media are supported, as well as scatterers made from chiral materials.

This contribution outlines the basic methods implemented and the program structure. Two interfaces to the implemented functionality are available: a flexible and fast low-level interface and a high-level interface for added convenience and plausibility checks. We conclude with two examples: a demonstration of the field calculation in various lattices and the explorations of quasi-bound states in the continuum. The presented code was already used in calculations for various physical systems: from the mode properties of molecular arrays in cavities to analytical models for metasurfaces and from moiré lattices to the homogenization of artificial photonic materials. With the publication of *treams* and the associated documentation, we hope to empower more scientists to make an efficient, fast, and precise exploration of nanophotonic systems that can be described in the broader framework of scattering theory.

Program summary
Program title: treams
CPC Library link to program files: https://doi.org/10.17632/2np8snmzfx.1
Developer's repository link: https://github.com/tfp-photonics/treams
Licensing provisions: MIT
Programming language: Python and Cython
Nature of problem: Simulating electromagnetic scattering in periodic nanophotonic structures, when different length scales are present or for large parameter sweeps, require specialized tools that can solve Maxwell's equations more efficiently than general-purpose solvers. A possible tool for these computations is the T-matrix method, which needs to be amended by suitable lattice summation schemes in the presence of periodic boundary conditions.
Solution method: The properties of the individual scatterers are described by T-matrices, such that the interaction can be solved analytically. The slowly converging lattice sums that appear in the presence of periodic boundary conditions are computed by converting them to two quickly converging series using Ewald's method. Depending on the lattice dimensions and the geometry of the scatterers, vector spherical, cylindrical, or plane waves are

* Corresponding author.
E-mail address: dominik.beutel@kit.edu (D. Beutel).

used. Using modes of well-defined helicity enables the straightforward inclusion of chiral material parameters for scatterers and embedding media.

1. Introduction

Advances in fabrication technology for nanophotonic devices open the avenue for novel designs and applications. This progress in manufacturing needs to be accompanied by novel simulation techniques that can outperform general-purpose Maxwell solvers, e.g., when hugely different length scales or large parameter sweeps are involved. The T-matrix (transition matrix) method [1] has proven to be an exceptionally useful tool to tackle such scattering scenarios. Various programs to solve multi-scattering scenarios in different conditions have been developed [2–8], and a continuously expanding list of software is available online [9,10].

While each of these codes has its advantage and unique capabilities, there is a lack of support for arbitrary periodic boundary conditions, especially when complex unit cells, i.e., unit cells containing multiple particles, are involved. Furthermore, including modes with well-defined helicity required to treat chiral constitutive relations is usually missing. Therefore, we present *treams* a program to solve electromagnetic scattering problems. It combines the highly efficient description of scatterers by the T-matrix formalism with quickly convergent lattice sums to handle their possible periodic arrangement [11–13]. These types of lattice sum transformations are known as Ewald method [14]. Furthermore, stratified structures can be included with an S-matrix (scattering matrix) description. Besides the conventional T-matrices that build on a vector spherical wave (SW) basis, vector cylindrical wave (CW) T-matrices [15] are also part of *treams*. The combination with the lattice sums for arbitrary one-dimensional (1D), two-dimensional (2D), and three-dimensional (3D) arrays of the translation coefficients for these underlying basis sets allows an efficient description of all types of arrays. Moreover, typical T-matrix operations, such as coupling within finite clusters, rotations, or translations, are implemented to support the simulation of various scattering scenarios. The program is written using Python, while most calculation-heavy functions are accelerated using Cython and, thus, run as compiled C code.

We want to especially mention the software QPMS [7], which also performs multi-scattering calculations in periodic systems using the Ewald method. Moreover, it also provides a compiled core library of functions that are also available through a Python interface. However, *treams* is different from QPMS by consistently including modes with well-defined helicity. Thus, chiral materials in the embedding medium and the scatterers can be used. Furthermore, *treams* can use T-matrices in the CW basis and S-matrices in the PW basis, enlarging the range of potential applications. Another important related software is MULTEM [16] and its successor, MULTEM2 [2], which implements methods to compute the response of 2D lattices of spheres in the SW basis and their stacking using an S-matrix algorithm.

The rest of this manuscript is structured as follows. We first define the basis sets of well-defined parity and helicity in Section 2.1: SWs, CWs, and vector plane waves (PWs). These basis sets and their relations are important for most calculations in *treams*. Next, we will briefly overview the T-matrix method in Section 2.2. We particularly exploit a generalized concept of the usual T-matrix definition based on SWs to also include CWs. These two basis sets complement each other on the type of object they can describe. Then, we outline the S-matrix method used for stacked planar structures in Section 2.3, which is used for the PW description. We conclude the first part with an overview of additional operators implemented in Section 2.4.

Section 3.1 then gives an overview of how the main functions, methods, and operators are included in the framework of *treams*. The section also describes other added functionality, such as importing and exporting T-matrix data in a file based on the HDF5 format [17]. Moreover, the operators, such as translations or rotations, are explained more in-depth from the programming and usage perspective in Section 3.2. These operators are working closely together with the implemented classes for general “physics-aware” arrays, i.e., *numpy* arrays with added information about physical quantities such as the wave number, and the more specialized data structures for T-matrices and S-matrices. Those classes are the focus of Section 3.3. Finally, we give examples of the capabilities of *treams*. We start with technical examples on 1D and 2D arrays, highlighting how the different lattice sums and basis sets work together and complement each other in Section 4.1. Finally, we show a physics-inspired example of calculating quasi-bound states in the continuum (quasi-BICs) in Section 4.2.

2. Electromagnetic scattering in chiral media

This section summarizes the underlying electromagnetic scattering theory, which is the foundation of the program *treams*. It describes SW, CW, and PW analytical solutions to the scattering of electromagnetic waves in chiral media. Using the first two of these basis sets allows the use of the T-matrix method, which is introduced for multi-scattering scenarios with and without periodic boundary conditions. We formulate these scattering scenarios in a global and a local T-matrix basis [18]. Together with the lattice sums for the periodic boundary conditions, this allows us to describe complex unit cells, namely unit cells with multiple different particles. For PWs, we use the S-matrix method. Finally, we will shortly discuss additional operators, e.g., for rotations, the change of basis set, and other common transformations in electromagnetic scattering calculations.

2.1. Vector spherical, cylindrical, and plane waves

treams solves Maxwell’s equations without free charges and currents. We assume monochromatic waves of vacuum wave number $k_0 = \frac{\omega}{c}$ where the time domain fields are defined as $\mathrm{Re}(\boldsymbol{E}(\boldsymbol{r}, k_0)\mathrm{e}^{-\mathrm{i}ck_0t})$ with the angular frequency ω and speed of light in vacuum c. The assumption of monochromatic waves is justified due to the later choice of linear constitutive relations. With this ansatz, Maxwell’s equations are

$$\nabla \times \begin{pmatrix} \boldsymbol{E}(\boldsymbol{r}, k_0) \\ Z_0 \boldsymbol{H}(\boldsymbol{r}, k_0) \end{pmatrix} = \mathrm{i}k_0 \begin{pmatrix} 0 & \mathbb{1} \\ -\mathbb{1} & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{\epsilon_0} \boldsymbol{D}(\boldsymbol{r}, k_0) \\ c\boldsymbol{B}(\boldsymbol{r}, k_0) \end{pmatrix} \tag{1a}$$

$$\nabla \cdot \begin{pmatrix} \frac{1}{\epsilon_0} \boldsymbol{D}(\boldsymbol{r}, k_0) \\ c\boldsymbol{B}(\boldsymbol{r}, k_0) \end{pmatrix} = 0\,, \tag{1b}$$

where all fields are normalized such that they have the same units as the electric field $\boldsymbol{E}(\boldsymbol{r}, k_0)$: the magnetic field $\boldsymbol{H}(\boldsymbol{r}, k_0)$ is multiplied by the free space impedance Z_0, the displacement field $\boldsymbol{D}(\boldsymbol{r}, k_0)$ is divided by the vacuum permittivity ϵ_0, and the magnetic flux density $\boldsymbol{B}(\boldsymbol{r}, k_0)$ is multiplied by the speed of light. Describing the frequency by the wave number is convenient for the implementation later since no unit of frequency is used explicitly, and hence there is no need to keep track of it. Instead, all quantities, such as positions or wave numbers, are implicitly assumed to have the same unit of length or inverse length.

Maxwell's equations in this form need to be supplemented by constitutive relations that describe the interdependence of the four fields above. In a linear, isotropic, homogeneous, and reciprocal medium, these can be expressed by three scalar dimensionless quantities, the relative permittivity $\epsilon(k_0)$, the relative permeability $\mu(k_0)$, and the chirality parameter $\kappa(k_0)$ [19] as

$$\begin{pmatrix} \frac{1}{\epsilon_0}\boldsymbol{D}(\boldsymbol{r},k_0) \\ c\boldsymbol{B}(\boldsymbol{r},k_0) \end{pmatrix} = \begin{pmatrix} \epsilon(k_0) & \mathrm{i}\kappa(k_0) \\ -\mathrm{i}\kappa(k_0) & \mu(k_0) \end{pmatrix} \begin{pmatrix} \boldsymbol{E}(\boldsymbol{r},k_0) \\ Z_0\boldsymbol{H}(\boldsymbol{r},k_0) \end{pmatrix}. \tag{2}$$

Thus, we arrive at

$$\nabla \times \begin{pmatrix} \boldsymbol{E}(\boldsymbol{r},k_0) \\ Z_0\boldsymbol{H}(\boldsymbol{r},k_0) \end{pmatrix} = k_0 \begin{pmatrix} \kappa(k_0) & \mathrm{i}\mu(k_0) \\ -\mathrm{i}\epsilon(k_0) & \kappa(k_0) \end{pmatrix} \begin{pmatrix} \boldsymbol{E}(\boldsymbol{r},k_0) \\ Z_0\boldsymbol{H}(\boldsymbol{r},k_0) \end{pmatrix} \tag{3}$$

by combining Eqs. (1a) and (2). In the case of achiral materials, it becomes possible to decouple the electric and magnetic field equations by applying the curl twice, arriving at the Helmholtz wave equation. For non-zero $\kappa(k_0)$, the equations for these two fields can be decoupled by introducing the Riemann-Silberstein fields $\sqrt{2}\boldsymbol{G}_\pm(\boldsymbol{r},k_0) = \boldsymbol{E}(\boldsymbol{r},k_0) \pm \mathrm{i}Z_0 Z(k_0)\boldsymbol{H}(\boldsymbol{r},k_0)$ [20,21] that then need to fulfill the differential equation

$$\nabla \times \boldsymbol{G}_\pm(\boldsymbol{r},k_0) = \pm k_\pm(k_0)\boldsymbol{G}_\pm(\boldsymbol{r},k_0), \tag{4}$$

where $Z(k_0) = \sqrt{\frac{\mu(k_0)}{\epsilon(k_0)}}$ is the relative wave impedance and $k_\pm(k_0) = k_0(\sqrt{\epsilon(k_0)\mu(k_0)} \pm \kappa(k_0))$ is the wave number in the medium for each helicity. Applying the curl twice, we also obtain the Helmholtz equation for $\boldsymbol{G}_\pm(\boldsymbol{r},k_0)$

$$\left(\nabla^2 + k_\pm^2(k_0)\right)\boldsymbol{G}_\pm(\boldsymbol{r},k_0) = 0. \tag{5}$$

However, Eq. (4) is actually stricter than the Helmholtz equation, i.e., not all solutions of the Helmholtz equation necessarily solve Eq. (4). However, starting from the solution theory developed for the Helmholtz equation, suitable solutions can be constructed easily, as shown in the following.

A set of transverse solutions for the vector Helmholtz equation obtained by applying the curl twice can be derived from solutions of the scalar Helmholtz equation $f_\nu(\boldsymbol{r},k)$ [22]. These solutions are assumed to have a general index ν. Then, the functions

$$\boldsymbol{M}_\nu(\boldsymbol{r},k) = \nabla \times \left(\boldsymbol{w} f_\nu(\boldsymbol{r},k)\right) \tag{6a}$$

$$\boldsymbol{N}_\nu(\boldsymbol{r},k) = \frac{\nabla}{k} \times \boldsymbol{M}_\nu(\boldsymbol{r},k) \tag{6b}$$

are transverse solutions, where $\boldsymbol{w}$ is a pilot vector depending on the chosen basis set. For the planar and cylindrical case, it is $\boldsymbol{w} = \hat{\boldsymbol{z}}$, and in the spherical case, we use $\boldsymbol{w} = \boldsymbol{r}$. The solutions $\boldsymbol{M}_\nu(\boldsymbol{r},k)$ are tangential to the surface defined by $\boldsymbol{w}$. When used for the electric field, which is possible for achiral media, they are called transverse electric (TE) waves. The solutions $\boldsymbol{N}_\nu(\boldsymbol{r},k)$ are called transverse magnetic (TM). Note that the transversality always refers to the direction of $\boldsymbol{w}$. We will refer to the TE/TM modes also as parity modes. Solutions of well-defined helicity can be constructed by

$$\boldsymbol{A}_{\nu,\pm}(\boldsymbol{r},k) = \frac{\boldsymbol{N}_\nu(\boldsymbol{r},k) \pm \boldsymbol{M}_\nu(\boldsymbol{r},k)}{\sqrt{2}} \tag{7}$$

from the parity solutions. They then allow to expand the field

$$\boldsymbol{G}_\pm(\boldsymbol{r},k_\pm(k_0)) = \sum_\nu a_{\nu,\pm}\boldsymbol{A}_{\nu,\pm}(\boldsymbol{r},k_\pm(k_0)) \tag{8}$$

in modes of positive or negative helicity. Inverting the definitions of these fields $\boldsymbol{G}_\pm(\boldsymbol{r},k_\pm(k_0))$ leads to an expansion of the electric and magnetic fields in modes of well-defined helicity.

The first set of concrete solutions are PWs. The scalar solution $f_{\hat{\boldsymbol{k}}}(\boldsymbol{r},k) = \mathrm{e}^{\mathrm{i}\boldsymbol{kr}}$ is indexed by the three Cartesian components of the normalized wave vector $\boldsymbol{k} = k\hat{\boldsymbol{k}}$. The application of Eq. (6) results in

$$\boldsymbol{M}_{\hat{\boldsymbol{k}}}(\boldsymbol{r},k) = \mathrm{i}\frac{k_y\hat{\boldsymbol{x}} - k_x\hat{\boldsymbol{y}}}{\sqrt{k_x^2+k_y^2}}\mathrm{e}^{\mathrm{i}\boldsymbol{kr}} = -\mathrm{i}\hat{\boldsymbol{\varphi}}_{\hat{\boldsymbol{k}}}\mathrm{e}^{\mathrm{i}\boldsymbol{kr}} \tag{9a}$$

$$\boldsymbol{N}_{\hat{\boldsymbol{k}}}(\boldsymbol{r},k) = \frac{-k_xk_z\hat{\boldsymbol{x}} - k_yk_z\hat{\boldsymbol{y}} + (k_x^2+k_y^2)\hat{\boldsymbol{z}}}{k\sqrt{k_x^2+k_y^2}}\mathrm{e}^{\mathrm{i}\boldsymbol{kr}} = -\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{k}}}\mathrm{e}^{\mathrm{i}\boldsymbol{kr}}, \tag{9b}$$

where we divided the functions by $\sqrt{k_x^2+k_y^2}$ to normalize the resulting modes. As shown on the right-hand side, these modes can be written compactly by using unit vectors of spherical coordinates (see Appendix A).

The CWs can be obtained from the scalar wave solutions $f^{(n)}_{k_zm}(\boldsymbol{r},k) = Z^{(n)}_m(k_\rho\rho)\mathrm{e}^{\mathrm{i}m\varphi+\mathrm{i}k_zz}$, that are indexed by the wave's z-component k_z and the azimuthal index $m \in \mathbb{Z}$. The vector $\boldsymbol{r}$ is expressed in cylindrical coordinates. Similarly, we define $k_\rho = \sqrt{k^2-k_z^2}$. The function $Z^{(n)}_m$ can be the Bessel or Hankel functions of the first or second kind. As usual for T-matrices, we choose as the two independent solutions the Bessel functions of the first kind $Z^{(1)}_m = J_m$, that are regular in the whole space, and the Hankel functions of the first kind $Z^{(3)}_m = H^{(1)}_m$, that fulfill the radiation condition. After applying Eq. (6) the basis set for CWs is

$$\boldsymbol{M}^{(n)}_{k_z m}(\boldsymbol{r},k) = \left(\mathrm{i}m\frac{Z^{(n)}_m(k_\rho\rho)}{k_\rho\rho}\hat{\boldsymbol{\rho}} - Z^{(n)\prime}_m(k_\rho\rho)\hat{\boldsymbol{\varphi}}\right)\mathrm{e}^{\mathrm{i}m\varphi+\mathrm{i}k_z z} \tag{10a}$$

$$\boldsymbol{N}^{(n)}_{k_z m}(\boldsymbol{r},k) = \left(\mathrm{i}\frac{k_z}{k}Z^{(n)\prime}_m(k_\rho\rho)\hat{\boldsymbol{\rho}} - \frac{k_z m}{k}\frac{Z^{(n)}_m(k_\rho\rho)}{k_\rho\rho}\hat{\boldsymbol{\varphi}} + \frac{k_\rho}{k}Z^{(n)}_m(k_\rho\rho)\hat{\boldsymbol{z}}\right)\mathrm{e}^{\mathrm{i}m\varphi+\mathrm{i}k_z z}\,. \tag{10b}$$

As in the case of PWs, we normalize the fields. Here, this is done by dividing through k_ρ.

Finally, in analogy to the derivation of the CWs, the SWs can be derived. The scalar solution is $f_{lm}(\boldsymbol{r},k) = z^{(n)}_l(kr)Y_{lm}(\theta,\varphi)$, where the functions $z^{(n)}_l$ are the spherical Bessel and Hankel functions, analogously defined to the Bessel and Hankel functions for the CWs. The spherical harmonics $Y_{lm}(\theta,\varphi)$ (Appendix B) have the indices $l \in \mathbb{N}$, where $l = 0$ corresponding to monopoles is not included for transverse waves, and $m \in \{-l, -l+1, \dots, l\}$. With Eq. (6) using $\boldsymbol{w} = \boldsymbol{r}$ the SWs are

$$\boldsymbol{M}^{(n)}_{lm}(\boldsymbol{r},k) = N_{lm}\left(\mathrm{i}\pi_{lm}(\theta)\hat{\boldsymbol{\theta}} - \tau_{lm}(\theta)\hat{\boldsymbol{\varphi}}\right)\mathrm{e}^{\mathrm{i}m\varphi}z_l(kr) = \boldsymbol{X}_{lm}(\theta,\varphi)z_l(kr) \tag{11a}$$

$$\boldsymbol{N}^{(n)}_{lm}(\boldsymbol{r},k) = \left[\mathrm{i}\sqrt{l(l+1)}Y_{lm}(\theta,\varphi)\frac{z_l(kr)}{kr}\hat{\boldsymbol{r}} + \left(z^{(n)\prime}_l(kr) + \frac{z^{(n)}_l(kr)}{kr}\right)\hat{\boldsymbol{r}}\times\boldsymbol{X}_{lm}(\theta,\varphi)\right], \tag{11b}$$

where we use the prefactor

$$N_{lm} = \mathrm{i}\sqrt{\frac{(2l+1)}{4\pi l(l+1)}\frac{(l-m)!}{(l+m)!}} \tag{12}$$

and the angular functions

$$\pi_{lm}(\theta) = \frac{mP^m_l(\cos\theta)}{\sin\theta} \tag{13a}$$

$$\tau_{lm}(\theta) = \frac{\partial P^m_l(\cos\theta)}{\partial\theta} \tag{13b}$$

are derived from the associated Legendre polynomials (Appendix B). With the PWs, the CWs, and the SWs defined, we can now use the properties of these functions to formulate the multi-scattering equations using the T-matrix method. For the PWs, we use an S-matrix description.

2.2. *T-matrix method in the presence and absence of periodic boundaries*

The T-matrix method has been proven to be a suitable approach to efficiently perform electromagnetic scattering calculations for various systems. At its core, the T-matrix method relies on a set of functions that form a complete basis to Maxwell's equations together with the chosen constitutive relations, with – among others – two properties: First, the functions can be separated into incident and scattered waves and, second, simple translation properties of the individual waves, such that multiple scattering calculations can be done efficiently, are known. Typically, these properties are associated with the SWs. However, it is equally possible to apply this concept to CWs. Due to the analytically known properties of these functions, not only the scattering coefficients but also various quantities, such as scattering and extinction cross-sections, circular dichroism, or duality breaking, can be computed highly efficiently.

For the T-matrix method, the total electric field is separated into incident and scattered modes that can be expanded as

$$\begin{aligned}\boldsymbol{E}_{\mathrm{inc}}(\boldsymbol{r},k_0) &= \sum_\nu\sum_{\lambda=\pm1} a_{\nu\lambda}\boldsymbol{A}^{(1)}_{\nu,\lambda}(\boldsymbol{r},k_\lambda(k_0))\\ \boldsymbol{E}_{\mathrm{sca}}(\boldsymbol{r},k_0) &= \sum_\nu\sum_{\lambda=\pm1} p_{\nu\lambda}\boldsymbol{A}^{(3)}_{\nu,\lambda}(\boldsymbol{r},k_\lambda(k_0))\end{aligned} \tag{14}$$

using the previously derived solutions in the helicity basis. In the case of vanishing chirality parameters, the fields can be expanded equally well in the parity basis of $\boldsymbol{M}^{(n)}_\nu(\boldsymbol{r},k(k_0))$ and $\boldsymbol{N}^{(n)}_\nu(\boldsymbol{r},k(k_0))$. *treams* has both solution sets implemented, but for brevity, we will restrict the introduction here to the helicity modes. The sum over ν is a shorthand for the sum over l and m in the case of SWs and for $m \in \mathbb{Z}$ and $k_z \in \mathbb{R}$ for CWs. The continuous index k_z needs, in principle, an integral expression in the expansion. However, we will restrict the application of CWs to systems with periodicity along z where only discrete diffraction orders couple. In both cases, the sum has to be truncated to obtain a finite matrix. By choosing a large enough value, the error introduced by this truncation can be controlled. For SWs, we, therefore, assume a finite scatterer size, such that a finite number of multipoles is sufficient and only waves with $l \le l_{\mathrm{max}}$ are used. For CWs, we first assume periodicity in the z-direction. Then, the sum over k_z only includes values that differ by multiples of the reciprocal lattice. Thus, for objects with a finite extent in the x- and y-directions, the sum over diffraction orders k_z and the sum over $|m| \le m_{\mathrm{max}}$ can be truncated. The finite number of incident and scattered field coefficients constitute then the vectors $\boldsymbol{a}$ and $\boldsymbol{p}$. Now, for a linear response, a relationship between the incident and scattered field can be expressed by the T-matrix $\mathbf{T}$ through

$$\boldsymbol{p} = \mathbf{T}\boldsymbol{a}\,. \tag{15}$$

So, the T-matrix can be seen as a full description of the electromagnetic properties at a specified frequency. It can be obtained analytically for spheres in SWs and infinitely long cylinders in CWs, which can consist of multiple concentric shells [23–25]. The calculation of these coefficients is included in *treams*. For more complicated objects, e.g., objects without spherical or cylindrical symmetry, the T-matrix can be computed, in theory, by a range of numerical methods [26–29]. These methods are, however, not part of *treams*. The possibility to include chiral material parameters in the scatterer or embedding, then depends on the restrictions of the used method to precalculate the T-matrix.

To describe the interaction between N particles with T-matrices $\mathbf{T}_i$ at positions $\boldsymbol{r}_i$ with $i \in \{1, \dots, N\}$, we have to add to the primary incident field $\boldsymbol{a}_i$ also the scattered fields of all other particles by

$$\boldsymbol{p}_i = \mathbf{T}_i \left[\boldsymbol{a}_i + \sum_{j \neq i} \mathbf{C}^{(3)}(\boldsymbol{r}_i - \boldsymbol{r}_j)\boldsymbol{p}_j \right] , \tag{16}$$

where $\mathbf{C}^{(3)}(\boldsymbol{r}_i - \boldsymbol{r}_j) = \mathbf{C}^{(3)}_{i,j}$ contains the translation coefficients (Appendix C) along $\boldsymbol{r}_i - \boldsymbol{r}_j$ [30–32]. This set of linear equations can be solved in the local basis, where the field expansions at all positions $\boldsymbol{a}_i$ and $\boldsymbol{p}_i$ are combined in one vector $\boldsymbol{a}_{\text{local}}$ and $\boldsymbol{p}_{\text{local}}$ [18]. The total scattered field is then the superposition of all individual scattered fields. Using the translation properties of the SWs, we can write the multi-scattering equation as

$$\boldsymbol{p}_{\text{local}} = \mathbf{T}_{\text{diag}} \left[\boldsymbol{a}_{\text{local}} + \mathbf{C}^{(3)} \boldsymbol{p}_{\text{local}} \right] , \tag{17}$$

where $\mathbf{T}_{\text{diag}}$ is the block-diagonal matrix containing the T-matrices of all particles

$$\mathbf{T} = \begin{pmatrix} \mathbf{T}_1 & 0 & \dots & 0 \\ 0 & \mathbf{T}_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \mathbf{T}_N \end{pmatrix} \quad \text{and} \quad \mathbf{C}^{(3)} = \begin{pmatrix} 0 & \mathbf{C}^{(3)}_{1,2} & \dots & \mathbf{C}^{(3)}_{1,N} \\ \mathbf{C}^{(3)}_{2,1} & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{C}^{(3)}_{N-1,N} \\ \mathbf{C}^{(3)}_{N,1} & \dots & \mathbf{C}^{(3)}_{N,N-1} & 0 \end{pmatrix} \tag{18}$$

expands for each particle the scattered fields from all particles in regular waves using the translation coefficients. Equation (17) can be rewritten as

$$\boldsymbol{p}_{\text{local}} = \underbrace{\left[\mathbb{1} - \mathbf{T}_{\text{diag}} \mathbf{C}^{(3)} \right]^{-1} \mathbf{T}_{\text{diag}}}_{= \mathbf{T}_{\text{local}}} \boldsymbol{a}_{\text{local}} \tag{19}$$

to obtain the local T-matrix for the interacting particles. That local T-matrix for the interacting particles can already directly be used to compute quantities such as the scattering cross-section without needing to convert it to the global basis. The expansion of the incident and scattered field is taken at N positions. By expressing the incident fields at each particle in terms of the expansion at the origin $\boldsymbol{a}$ with $\boldsymbol{r}_0 = 0$

$$\boldsymbol{a}_{\text{local}} = \begin{pmatrix} C^{(1)}_{1,0} & \dots & C^{(1)}_{N,0} \end{pmatrix} \boldsymbol{a} \tag{20}$$

and, likewise, summing the scattered fields from all particles to express them in an expansion at the origin

$$\boldsymbol{p} = \begin{pmatrix} C^{(1)}_{0,1} & \dots & C^{(1)}_{0,N} \end{pmatrix} \boldsymbol{p}_{\text{local}} , \tag{21}$$

we can obtain a global T-matrix

$$\mathbf{T} = \begin{pmatrix} C^{(1)}_{0,1} & \dots & C^{(1)}_{0,N} \end{pmatrix} \mathbf{T}_{\text{local}} \begin{pmatrix} C^{(1)}_{1,0} & \dots & C^{(1)}_{N,0} \end{pmatrix} \tag{22}$$

from the local T-matrix. In practice, *treams* does not distinguish between local and global T-matrices: the global T-matrix is treated as a local T-matrix in the special case that only one origin is present.

Next, we want to consider periodic boundaries. We assume that there are N particles per unit cell, with $\boldsymbol{r}_i - \boldsymbol{r}_j$ pointing from particle j to particle i in the same unit cell. The d-dimensional lattice is defined by the basis vectors $\boldsymbol{u}_i$ with $i \in \{1, \dots, d\}$ and consists of the set $\Lambda_d = \left\{ \sum_{i=1}^{d} n_i \boldsymbol{u}_i | n_i \in \mathbb{Z} \right\}$. We assume that the illumination of the array has a fixed phase relation along the direction of the lattice given by the vector $\boldsymbol{k}_\parallel$. Thus, the phase difference between unit cells of distance $\boldsymbol{R} \in \Lambda_d$ is $\mathrm{e}^{\mathrm{i}\boldsymbol{k}_\parallel \boldsymbol{R}}$. We select one unit cell as our reference that is placed at the origin by convention. Then, the scattered fields amplitudes $\tilde{\boldsymbol{p}}_i$ are defined analogously to Eq. (16) but now including a sum over all lattice sites, by

$$\tilde{\boldsymbol{p}}_i = \mathbf{T}_i \left[\tilde{\boldsymbol{a}}_i + \sum_{j=1}^{N} \sum_{\boldsymbol{R} \in \Lambda_d}{}' \mathbf{C}^{(3)}(\boldsymbol{r}_i - \boldsymbol{r}_j - \boldsymbol{R}) \tilde{\boldsymbol{p}}_{j,\boldsymbol{R}} \right] . \tag{23}$$

The prime next to the lattice sum indicates the omission of the self-interaction term with $\boldsymbol{R} = 0$ if $i = j$. Making use of the fixed phase relation $\tilde{\boldsymbol{p}}_{j,\boldsymbol{R}} = \mathrm{e}^{\mathrm{i}\boldsymbol{k}_\parallel \boldsymbol{R}} \tilde{\boldsymbol{p}}_j$, we can convert this expression to

$$\tilde{\boldsymbol{p}}_i = \mathbf{T}_i \left[\tilde{\boldsymbol{a}}_i + \sum_{j=1}^{N} \underbrace{\sum_{\boldsymbol{R} \in \Lambda_d}{}' \mathbf{C}^{(3)}(\boldsymbol{r}_i - \boldsymbol{r}_j - \boldsymbol{R}) \mathrm{e}^{\mathrm{i}\boldsymbol{k}_\parallel \boldsymbol{R}}}_{= \tilde{\mathbf{C}}^{(3)}_{ij}} \tilde{\boldsymbol{p}}_j \right] , \tag{24}$$

with only quantities from the reference unit cell.

The lattice sums over translation coefficients converge notoriously slow [11]. However, they can be considerably accelerated using the Ewald summation technique [14,11,33–35]. For each of the basis sets, SWs and CWs, combined with each possible lattice dimension, they have to be evaluated individually. An additional difficulty that arises in these sums is the inclusion of complex unit cells, which additionally require shifts within the unit cell. We derived these lattice sums in a unified manner previously [13]. All the lattice sums are implemented in *treams*. These lattice sums assume a default orientation of the lattices. 2D lattices of SWs are placed in the x-y-plane, 1D lattices of SWs are placed along the z-direction, and 1D lattices of CWs are placed along the x-direction as shown in Fig. 1. Using the definition of the translation matrix with periodic boundaries $\tilde{\mathbf{C}}^{(3)}$ analogous to Eq. (18) but now with the diagonal blocks generally being non-zero and containing the interaction of the particle with its periodic equivalents, we can write

$$\tilde{\boldsymbol{p}}_{\text{local}} = \left[\mathbb{1} - \mathbf{T}_{\text{diag}} \tilde{\mathbf{C}}^{(3)} \right]^{-1} \mathbf{T}_{\text{diag}} \tilde{\boldsymbol{a}}_{\text{local}} , \tag{25}$$

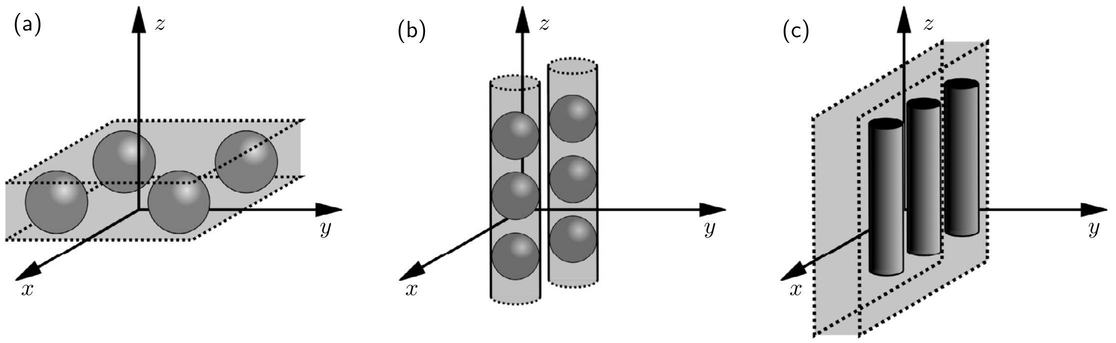

Fig. 1. The domains of validity for different expansion and lattices. For objects described by a T-matrix, i.e., the incident and scattered fields are expanded in SWs, the expansion is only valid outside the circumscribing sphere [36]. Then, for a 2D lattice in the x-y-plane of T-matrices, the expansion in PWs is valid above and below the planes that enclose the lattice and the scatterers, as shown in panel (a). Panel (b) shows the domain of validity when a 1D lattice of SW expansions along the z-axis is expanded in a CW. As shown, a local T-matrix in SWs can be expanded in a local CW basis, where each position around which the field is expanded has its own domain of validity. The CWs are only valid in the domain outside of a circumscribing cylinder. Similarly to the expansion of SWs in PWs in panel (a), the expansion of CWs on a 1D lattice along the x-axis in PWs is only valid outside the planes that enclose the lattice.

which is formally quite similar to Eq. (19). However, the matrix $\tilde{\mathbf{C}}^{(3)}$ includes a sum over lattice sites, that the incident field is assumed to fulfill the phase relation defined by $\boldsymbol{k}_\parallel$, and that the total scattered field is not only obtained after summing over all particle positions but also all lattice sites. To find lattice modes supported in the lattice without external illumination, we can solve

$$\left[\mathbb{1} - \mathbf{T}_\text{diag}\tilde{\mathbf{C}}^{(3)}\right]\tilde{\boldsymbol{p}}_\text{local} = 0\,, \tag{26}$$

which in practice can be done by calculating eigenmodes with vanishing or sufficiently small eigenvalues.

2.3. S-matrices for stratified structures

For PWs, we use an S-matrix description. This means, instead of incident and scattered modes, we use a separation into incoming and outgoing modes. These modes are defined concerning a reference plane that we assume to be perpendicular to $\hat{\boldsymbol{z}}$. Similarly to the CWs, we will assume periodicity in the x-y-plane. Then, we can make a slight modification of the mode definition. In Eq. (9), the PWs are indexed by the three components of $\hat{\boldsymbol{k}}$, now we use the following definition that is more suitable for systems with periodicity in the x-y-plane later: We index the modes with $\boldsymbol{k}_\parallel = k_x\hat{\boldsymbol{x}} + k_y\hat{\boldsymbol{y}}$ and additionally a direction $d = \pm 1$ that defines the sign of the remaining component $k_z = \pm\sqrt{k^2 - k_\parallel^2}$. We chose the same symbol $\boldsymbol{k}_\parallel$ as in Eq. (24) because the parallel component here can be immediately used in that equation as the phase difference between unit cells. The incoming fields are

$$\boldsymbol{E}_\text{in}(\boldsymbol{r}, k_0) = \sum_{\boldsymbol{k}_\parallel}\sum_{\lambda=\pm1} a_{\boldsymbol{k}_\parallel d\lambda}\boldsymbol{A}_{\boldsymbol{k}_\parallel d,\lambda}(\boldsymbol{r}, k_\lambda(k_0))\,, \tag{27}$$

with $d = \pm 1$ for $z \lessgtr 0$ and the outgoing fields are

$$\boldsymbol{E}_\text{out}(\boldsymbol{r}, k_0) = \sum_{\boldsymbol{k}_\parallel}\sum_{\lambda=\pm1} b_{\boldsymbol{k}_\parallel d\lambda}\boldsymbol{A}_{\boldsymbol{k}_\parallel d,\lambda}(\boldsymbol{r}, k_\lambda(k_0))\,. \tag{28}$$

Therefore, to describe the scattering of a particular PW in a 2D periodic system, it is sufficient to reduce the values of $\boldsymbol{k}_\parallel$ from a continuous spectrum to the countable set of diffraction orders indicated by the sum symbol. Furthermore, we can truncate the sum to a finite number of modes, neglecting only highly evanescent contributions to the fields. After sorting the expansion coefficients into four sets of modes $\boldsymbol{a}_\uparrow$, $\boldsymbol{a}_\downarrow$, $\boldsymbol{b}_\uparrow$, and $\boldsymbol{b}_\downarrow$, we can describe the scattering with

$$\begin{pmatrix}\boldsymbol{b}_\uparrow \\ \boldsymbol{b}_\downarrow\end{pmatrix} = \underbrace{\begin{pmatrix}\mathbf{S}_{\uparrow\uparrow} & \mathbf{S}_{\uparrow\downarrow} \\ \mathbf{S}_{\downarrow\uparrow} & \mathbf{S}_{\downarrow\downarrow}\end{pmatrix}}_{\mathbf{S}}\begin{pmatrix}\boldsymbol{a}_\uparrow \\ \boldsymbol{a}_\downarrow\end{pmatrix}. \tag{29}$$

To avoid confusion between the direction $d = \pm 1$ and the helicity λ, we used $\uparrow$ and $\downarrow$ to indicate the direction. We implicitly assume that the scattering takes place at the $z = 0$ plane. Except for planar interfaces, this is obviously only a model description. The domain of validity of the field expansion is then only above and below the planes that encase the scatterers geometrically, as shown in Fig. 1(a). The two diagonal blocks contain the coefficients for transmission. The off-diagonals contain the coefficients for reflection.

Two systems described by the matrices $\mathbf{S}^I$ and $\mathbf{S}^{II}$ can be stacked, where the resulting S-matrix is given by

$$\mathbf{S}_{\uparrow\uparrow} = \mathbf{S}^{II}_{\uparrow\uparrow}\left[1 - \mathbf{S}^I_{\uparrow\downarrow}\mathbf{S}^{II}_{\downarrow\uparrow}\right]^{-1}\mathbf{S}^I_{\uparrow\uparrow} \tag{30a}$$

$$\mathbf{S}_{\uparrow\downarrow} = \mathbf{S}^{II}_{\uparrow\uparrow}\left[1 - \mathbf{S}^I_{\uparrow\downarrow}\mathbf{S}^{II}_{\downarrow\uparrow}\right]^{-1}\mathbf{S}^I_{\uparrow\downarrow}\mathbf{S}^{II}_{\downarrow\downarrow} + \mathbf{S}^{II}_{\uparrow\downarrow} \tag{30b}$$

$$\mathbf{S}_{\downarrow\downarrow} = \mathbf{S}^I_{\downarrow\downarrow}\left[1 - \mathbf{S}^{II}_{\downarrow\uparrow}\mathbf{S}^I_{\uparrow\downarrow}\right]^{-1}\mathbf{S}^{II}_{\downarrow\downarrow} \tag{30c}$$

$$\mathbf{S}_{\downarrow\uparrow} = \mathbf{S}^I_{\downarrow\downarrow}\left[1 - \mathbf{S}^{II}_{\downarrow\uparrow}\mathbf{S}^I_{\uparrow\downarrow}\right]^{-1}\mathbf{S}^{II}_{\downarrow\uparrow}\mathbf{S}^I_{\uparrow\uparrow} + \mathbf{S}^I_{\downarrow\uparrow}\,, \tag{30d}$$

taking the multi-reflection between the two objects described by the S-matrices into account. For repeating structures along the z-direction, the layer doubling technique is highly efficient in calculating the response [37]. Furthermore, it is conveniently possible to calculate the band structure of periodic systems along the z-direction from the S-matrix of a single layer [16,37].

2.4. Further operators

Up to now, we already considered few operators, in the sense that they perform a particular action when applied to, e.g., a field expansion. The T-matrix or S-matrix can be seen as an operator, also the matrices that transform the scattered fields back into incident fields. Here, we will discuss several other common operations implemented in *treams*. These are translations and rotations of objects, the expansion of a field in another basis, and the expansion of the scattered fields in a lattice in another basis.

We already introduced the expansion of scattered fields in incident fields for CWs and SWs $\mathbf{C}^{(3)}$ and the matrices to expand those fields in the same type $\mathbf{C}^{(1)}$, i.e., expand incident fields at some point into incident fields at another point and, analogously, for the expansion of scattered fields in scattered fields. A missing part, especially for connecting different basis sets, is the expansions in one basis set into other basis sets. First, we have the expansion of PWs in SWs [38]

$$\boldsymbol{A}_{\boldsymbol{k},\pm}(\boldsymbol{r},k) = 4\pi \sum_{l=1}^{\infty}\sum_{m=-l}^{l} N_{lm}\mathrm{i}^{l}\left(\tau_{lm}(\cos\theta_{\boldsymbol{k}}) \pm \pi_{lm}(\cos\theta_{\boldsymbol{k}})\right)\mathrm{e}^{-\mathrm{i}m\varphi_{\boldsymbol{k}}}\boldsymbol{A}^{(1)}_{lm,\pm}(\boldsymbol{r},k) \tag{31}$$

and of PWs in CWs [39]

$$\boldsymbol{A}_{\boldsymbol{k},\pm}(\boldsymbol{r},k) = \sum_{m=-\infty}^{\infty} \mathrm{i}^{m}\mathrm{e}^{-\mathrm{i}m\varphi_{\boldsymbol{k}}}\boldsymbol{A}^{(1)}_{k_z m,\pm}(\boldsymbol{r},k)\,. \tag{32}$$

These expansions can be obtained from the scalar plane wave expansions and then applying the respective operations from Eq. (6). The PWs are regular in the whole space, so their expansion consists only of regular CWs or regular SWs. A benefit of the helicity basis is apparent in these equations: opposite helicities are not mixed by these operations. For a truncated expansion, the coefficients can be used in matrices to describe the change of basis. By equating and comparing the coefficients from the two PWs expansions, we get the missing expansion [40]

$$\boldsymbol{A}^{(1)}_{k_z m,\pm}(\boldsymbol{r},k) = \sum_{l=\max(|m|,1)}^{\infty} 4\pi N_{lm} i^{l-m}\left(\tau_{lm}(\cos\theta_{\boldsymbol{k}}) \pm \pi_{lm}(\cos\theta_{\boldsymbol{k}})\right)\boldsymbol{A}^{(1)}_{lm,\pm}(\boldsymbol{r},k) \tag{33}$$

of CWs in SWs. The inverse transformations of these expressions can be derived from the Fourier transforms of the scalar cylindrical and spherical waves [41]. They result in continuous spectra and, thus, require discretization of the spectrum to describe them with simple matrices.

We find a convenient connection between the chosen basis sets for periodic boundary conditions that can simplify and speed-up computations. As mentioned earlier, each type of solution has a geometric symmetry where it is most beneficial. SWs are well-suited for objects of finite size, CWs are well-suited for objects that are periodic along one axis and of finite extent in the remaining two directions, and PWs can be efficiently used for 2D periodic structures. We can easily find transformations between basis sets by taking these natural symmetries of the different solution sets as guidance. For example, summing singular SWs placed on a 1D lattice along the z-axis leads to CWs. In such a case, the typical trade-off is between the speed of the calculation and the convergence area of the solution, as shown in Fig. 1. While using SWs together with lattice sums provides convergence in the whole area outside the circumscribing spheres of each particle, the calculation, although considerably accelerated by Ewald's method, is comparably slow compared to the use of CWs. This basis set is inherently efficient for 1D periodic structures. However, the solution now only converges outside the circumscribing cylinder. In effect, these two methods complement each other. Analogously to the described transition from 1D lattices of SWs to CWs, it is possible to find transitions from SWs on a 2D lattice to PWs and from CWs on a 1D lattice to PWs. In the latter case, the inherent 1D periodicity of the CWs in one direction, in addition to their arrangement on a 1D lattice in a perpendicular direction, leads to the required 2D periodicity for the PWs. Moreover, under certain conditions, it is possible that by changing the basis, it becomes possible to expand the range of validity in certain parts of the domain [6].

By using the Fourier space representation of the CWs and the PWs together with Poisson's equation for lattice sums (Appendix E), we obtain

$$\sum_{\boldsymbol{R}\in\Lambda_1}\boldsymbol{A}^{(3)}_{lm,\lambda}(\boldsymbol{r}-\boldsymbol{R},k)\mathrm{e}^{\mathrm{i}\boldsymbol{k}_\parallel\boldsymbol{R}} = -\frac{\mathrm{i}N_{lm}\pi}{ak\mathrm{i}^{l-m}}\sum_{\boldsymbol{G}\in\Lambda_1^*}\left(\tau_{lm}(\theta_{\boldsymbol{k}}) \pm \pi_{lm}(\theta_{\boldsymbol{k}})\right)\boldsymbol{A}^{(3)}_{G+k_\parallel m,\lambda}(\boldsymbol{r},k)\,, \tag{34}$$

where, by assuming the lattice $\Lambda_1 = \left\{na\hat{\boldsymbol{z}} | n\in\mathbb{Z}\right\}$ to be in the z-direction with pitch a, we have on the right-hand side of the equation the sum over diffraction orders $\boldsymbol{G} = G\hat{\boldsymbol{z}} \in \left\{n\frac{2\pi}{a}\hat{\boldsymbol{z}}|n\in\mathbb{Z}\right\}$. Furthermore, we have $\boldsymbol{k}_\parallel = k_\parallel\hat{\boldsymbol{z}}$. This geometry implies $\cos\theta_{\boldsymbol{k}} = \frac{k_\parallel + G}{k}$.

Similarly, we can transform a sum on a 2D lattice in the x-y-plane to PWs by transforming them with the expansion

$$\sum_{\boldsymbol{R}\in\Lambda_2}\boldsymbol{A}^{(3)}_{lm,\lambda}(\boldsymbol{r}-\boldsymbol{R},k)\mathrm{e}^{\mathrm{i}\boldsymbol{k}_\parallel\boldsymbol{R}} = -\frac{\mathrm{i}N_{lm}2\pi}{Ak^2\mathrm{i}^{l}}\sum_{\boldsymbol{G}\in\Lambda_2^*}\frac{\mathrm{e}^{\mathrm{i}m\varphi_{\boldsymbol{k}}}}{\sqrt{1-\frac{(\boldsymbol{k}_\parallel+\boldsymbol{G})^2}{k^2}}}\left(\tau_{lm}(\theta_{\boldsymbol{k}}) \pm \pi_{lm}(\theta_{\boldsymbol{k}})\right)\boldsymbol{A}_{\boldsymbol{G}+\boldsymbol{k}_\parallel,d,\lambda}(\boldsymbol{r},k)\,, \tag{35}$$

where A is the area of one unit cell and $\cos\theta_{\boldsymbol{k}} = d\frac{\sqrt{k^2-(\boldsymbol{k}_\parallel+\boldsymbol{G})^2}}{k}$. Lastly, there is the transformation of CWs on a 1D lattice along the x-direction into PWs

$$\sum_{\boldsymbol{R}\in\Lambda_1}\boldsymbol{A}_{k_z m,\lambda}(\boldsymbol{r}-\boldsymbol{R},k)\mathrm{e}^{\mathrm{i}\boldsymbol{k}_\parallel\boldsymbol{R}} = \frac{2(-\mathrm{i})^m}{ak}\sum_{\boldsymbol{G}\in\Lambda_1^*}\frac{\mathrm{e}^{\mathrm{i}m\varphi_{\boldsymbol{k}}}}{\sqrt{1-\frac{(\boldsymbol{k}_\parallel+\boldsymbol{G})^2}{k^2}}}\boldsymbol{A}_{\hat{\boldsymbol{k}},\lambda}(\boldsymbol{r},k)\,, \tag{36}$$

with $\boldsymbol{k} = (k_\parallel + G)\hat{\boldsymbol{x}} + d\sqrt{k^2-(k_\parallel+G)^2-k_z^2}\hat{\boldsymbol{y}} + k_z\hat{\boldsymbol{z}}$ Note that in the last case, the periodicity is in the x-z-plane. To permute the labeling of the axes to the more conventional periodicity in the x-y-plane, the formula (Appendix D)

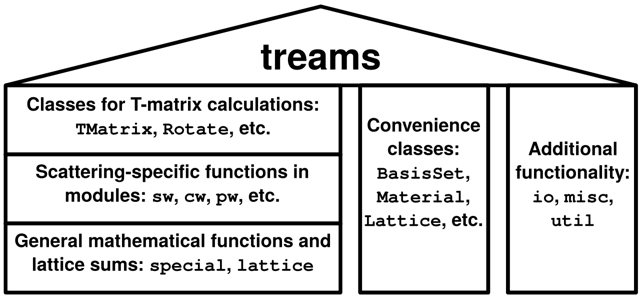
treams
Classes for T-matrix calculations: TMatrix, Rotate, etc.
Scattering-specific functions in modules: sw, cw, pw, etc.
General mathematical functions and lattice sums: special, lattice
Convenience classes: BasisSet, Material, Lattice, etc.
Additional functionality: io, misc, util

Fig. 2. The general structure of *treams*. The T-matrix calculations can be roughly separated into three levels. The focus for the lower levels was speed, and they are therefore implemented in Cython. They are generally available as *numpy* universal functions, which integrates them neatly into the general framework for numerical linear algebra operations. The higher levels are more specifically aimed at scattering calculations and provide more tailored access to the underlying functions. On the highest level, the classes provide more convenience, functionality, and plausibility checks. This functionality is paired with several classes that support creating the necessary definitions of basis set, lattices, and other structures. The program is completed with functions, e.g., to load and store T-matrices.

$$\boldsymbol{A}_{\hat{\boldsymbol{k}}',\lambda}(\boldsymbol{r}',k) = \frac{-k_x k_z + \lambda \mathrm{i} k k_y}{\sqrt{k_x^2 + k_y^2}\sqrt{k_y^2 + k_z^2}} \boldsymbol{A}_{\hat{\boldsymbol{k}},\lambda}(\boldsymbol{r},k) \tag{37}$$

can be used, where we have the transformation from $\boldsymbol{r}' = z\hat{\boldsymbol{x}} + x\hat{\boldsymbol{y}} + y\hat{\boldsymbol{z}}$ and $\boldsymbol{k}' = k_z\hat{\boldsymbol{x}} + k_x\hat{\boldsymbol{y}} + k_y\hat{\boldsymbol{z}}$ to $\boldsymbol{r} = x\hat{\boldsymbol{x}} + y\hat{\boldsymbol{y}} + z\hat{\boldsymbol{z}}$ and $\boldsymbol{k} = k_x\hat{\boldsymbol{x}} + k_y\hat{\boldsymbol{y}} + k_z\hat{\boldsymbol{z}}$.

Other common transformations are the rotations by Euler angles or translations along some direction. For SWs, rotations are represented by the Wigner-D matrix elements [42]. For PWs and CWs, currently, only rotations about the z-axis are implemented. Translations along a vector $\boldsymbol{r}$ for PWs only include a phase factor $\mathrm{e}^{\mathrm{i}\boldsymbol{k}\boldsymbol{r}}$. For CWs and SWs, the translation in the local basis is just block-diagonal matrices containing the translation coefficients $\boldsymbol{C}^{(1)}(\boldsymbol{r})$. The last transformation to be explicitly mentioned here is the change of the polarization between helicity and parity basis, which is given by Eq. (7).

3. Program overview

The program *treams* is a Python package that provides functionality to perform electromagnetic scattering calculations using the T-matrix method. It aims to provide an accessible framework for fast calculations using the basis sets defined in the previous section and operations on them in combination with implementations of exponentially fast converging lattice sums. First, we will introduce the general structure of *treams*, where on the one hand, we have fast and flexible functions for multi-scattering calculations, and on the other hand, a more high-level description of the T-matrices and possible operations on them that provide more convenience and plausibility checks of the arguments in functions. We describe the implemented operators and, eventually, the implemented classes for T-matrices and S-matrices.

3.1. General structure

A broad overview of *treams* is shown in Fig. 2. One of the major parts of the program is, of course, directly related to electromagnetic scattering calculations. These form the first column supporting *treams*. The second column is related to convenience classes that hold descriptions, e.g., of the lattices or materials. Lastly, there are modules with additional functionality to support, for example, loading or storing T-matrices into a file format based on the HDF5 format or the interface to *numpy* [43] arrays, which are one of the standard ways to store matrices and vectors in computer memory.

The first column, containing the electromagnetic scattering modules, classes, and functions, can be further separated into three levels. At the base, there are the implementations of mathematical functions that are not available from widely used packages such as *scipy* [44]: these include, for example, the different types of waves and their translation coefficients and associated functions such as the Wigner-3j symbols [45]. In analogy to the submodule in *scipy*, these functions are in a module called `special` (from special mathematical functions). The other part of this low-level implementation is the lattice sums and associated integrals. These basic methods are optimized for speed. Therefore, they are implemented in Cython, which gets converted to C-code and compiled, increasing the speed of the computations. These functions are then exposed in Python as *numpy* universal functions. This makes those functions behave similarly to the functions of *numpy* itself, such as the exponential function. For example, they are vectorized and broadcast when called with arrays as arguments. Furthermore, the implemented functions are also pure functions. Since those functions are available as independent modules, they can easily be reused in other programs. Moreover, a Cython-interface to those functions is provided via `special.cython_special` and `lattice.cython_lattice`, such that they can be reused in other compiled code.

The next higher level contains the modules `pw`, `cw`, `sw`, and `coeff`. They provide mathematical functions with a dedicated purpose, e.g., implementing the basis change coefficients with all prefactors. These functions are still provided as part of compiled universal functions with the same procedure as explained for the basic functions. This makes the separation of those two levels arbitrary to some point. However, by providing a coarse categorization by the basis set involved, they are intended to help find the relevant functions quickly. The module `coeff` offers functions to compute scattering coefficients, namely the Fresnel coefficients for planar interfaces and multilayers with chirality, the Mie coefficients for chiral spheres also generalized for spheres with additional spherical shells, and the scattering coefficients for infinitely long cylinders in the CW basis.

The high-level classes provide an alternative interface to these functions through several classes. They interact closely with the functionality of classes listed in the other columns. A detailed unified modeling language (UML) style diagram of the relevant classes is shown in Fig. 3. Central to the calculations is the class `PhysicsArray`, which behaves like *numpy* arrays by using interfaces provided by *numpy* and deriving from its class `numpy.lib.mixins.NDArrayOperatorMixin`. The parent class `AnnotatedArray` implements a general way to store information, such as the underlying basis set, in each dimension of the array. In a `PhysicsArray`, these annotations are restricted to a defined set of physical parameters,

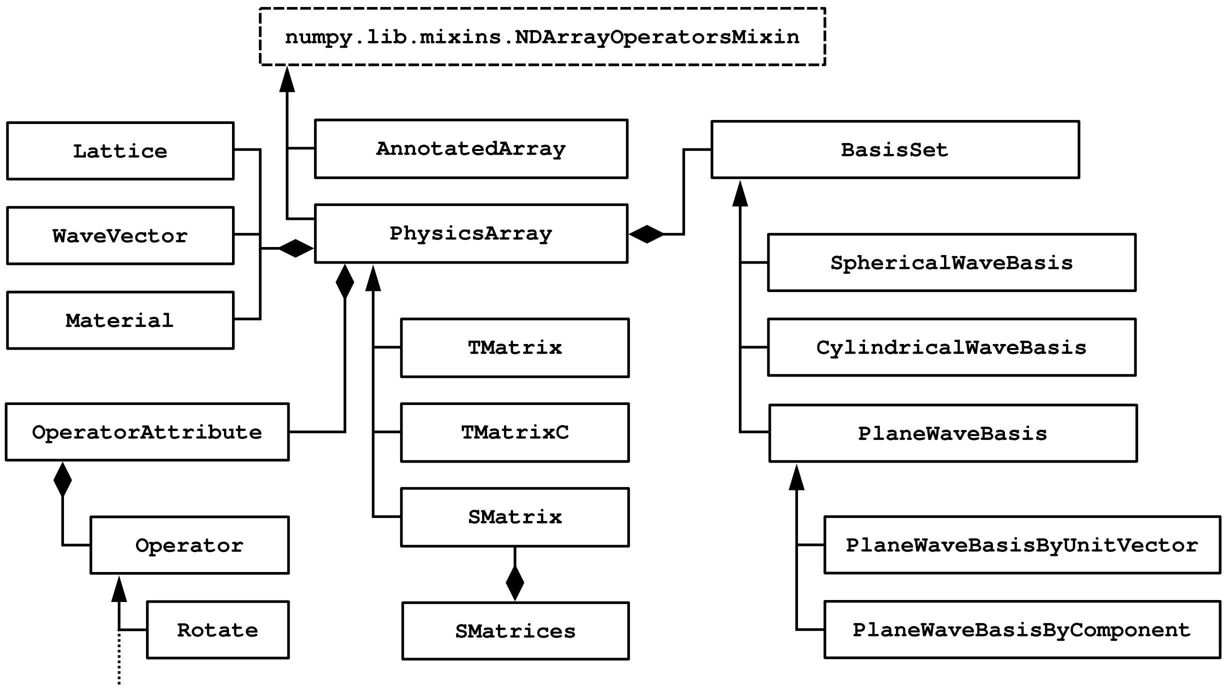

Fig. 3. Major classes of *treams*. The central column shows structures that behave mostly like *numpy* arrays and adds additional functionality to it. The central class for most of the functionality is `PhysicsArray`. Instances of that class keep track of physically relevant information, such as the wave number, the basis set, the embedding material, etc. Some of these objects are implemented as classes themselves. Especially the class `BasisSet` is the parent to all available types of basis sets implemented. A range of operators is wrapped as attributes and composed to the class.

Table 1
Available Operators.

Operator name	Description
`Rotate`	Rotate field by Euler angles
`Translate`	Translate field by a vector
`Expand`	Expand field in another basis/at other positions
`ExpandLattice`	Expand field in another basis/at other positions assuming a periodic lattice
`Permute`	Permute the axes of PWs
`ChangePoltype`	Switch between helicity and parity basis
`EField`, `HField`, …	Evaluate the field at specified positions

which can then be exposed as attributes. Thus, a `PhysicsArray` holds, e.g., wave number, lattice, wave vector, material, or basis set information in a composition. Compositions are shown with the diamond symbol.

Another part composed of a `PhysicsArray` is called `OperatorAttribute`. They perform certain transformations on the values in the array and will be treated in Section 3.2. Finally, on the right side of Fig. 3, we see the different types of basis sets: they obviously hold the SWs, the CWs, and the PWs. The latter is further separated into two possible descriptions. First, a PW basis can be described by a set of unit vectors $\hat{\boldsymbol{k}}$. Second, the wave can be described by two components and additionally the sign of the remaining third component, which is a convenient representation in the presence of a lattice and diffraction orders, as mentioned earlier. The classes for T-matrices and S-matrices then derive from the `PhysicsArray`. They place even tighter restrictions on the annotations, e.g., the basis set has to be of a certain type. Most operations, like rotations, are then automatically implemented by the operators of the parent class `PhysicsArray`. Only specific functions, e.g., to solve the multi-scattering problem in Eq. (19) or to compute scattering and extinction cross-sections, are explicitly implemented for the T-matrices. We will discuss the operators and the classes `PhysicsArray`, `TMatrix`, and `SMatrix` now in more detail.

3.2. Operators

The main idea underlying the class Operators and its subclasses is transferring the concept of abstract operators and their concrete representations from the mathematical domain to an implementation in Python. For example, if we want to apply a rotation by $\frac{\pi}{3}$ around the z-axis to a matrix $\mathbf{M}$, we can write this as

$$\mathbf{R}\left(\frac{\pi}{3}\right)\mathbf{M}\mathbf{R}^{-1}\left(\frac{\pi}{3}\right), \tag{38}$$

with $\mathbf{R}(\frac{\pi}{3})$ as an abstract rotation operator that depends on one variable. Depending on the basis in which we interpret the coefficients of $\mathbf{M}$, this abstract operator has different representations. If it is in the SW basis, these are Wigner-D matrix elements. With *treams*, it is possible to write this as

Table 2
Attributes of `PhysicsArray`.

Description	Attribute	Type
Vacuum wave number	`k0`	`float`
Embedding material	`material`	`Material`
Basis set	`basis`	`BasisSet`
Polarization type	`poltype`	`str`
Mode type (depending on basis set)	`modetype`	`str`
Lattice	`lattice`	`Lattice`
Wave Vector (along the lattice dimensions)	`kpar`	`WaveVector`

Listing 1: Minimal example for the use of `Operator`

```
>>> import numpy as np
>>> import treams
>>> class Foo:
>>>     def __init__(self, arr, basis):
...         self.arr = arr
...         self.basis = basis
...     def __array__(self):
...         return self.arr
...
>>> swb = treams.SphericalWaveBasis.default(1)
>>> mat = Foo(np.eye(6), swb)
>>> rot = treams.Rotate(np.pi)
>>> rot @ mat @ rot.inv
PhysicsArray(
...
```

where we first import *numpy* and *treams*and define a custom class with a minimal interface. Then, we create the matrix and the rotation operator, and, finally, the last command is an almost one-to-one correspondence of Eq. (38). *treams* is only invoked three times, first to create a basis, second to create the rotation operator, and implicitly when using the @-symbol. Most of the lines are used to set up a custom class; of course, *treams* comes with more elaborate implementations, but to emphasize the fact that the operator only depends on the presence of the attribute `basis`, we use the custom class here. This approach is chosen to facilitate custom class implementations by users if they need special functions and, thereby, improve the reusability of different parts of *treams*.

There are several other operators implemented, which are listed in Table 1. Besides rotations, there are translations along a vector $\boldsymbol{r}$, the expansion in another basis which is separated in expansions with and without periodic boundaries. The operator `Expand` includes the expansion within the same basis set but at different positions, i.e., $\mathbf{C}^{(1)}$ and $\mathbf{C}^{(3)}$, but also the expansions as defined in Eqs. (31) to (33). The expansions with periodic boundary conditions create the matrices $\tilde{\mathbf{C}}^{(3)}$ when used within the same type of basis or the expansions of periodic waves in other basis sets, as in Eqs. (34) to (36). Additionally, the change between helicity and parity modes and the permutation of the lattice axes are implemented. The primary purpose of the latter is conveniently switching between the PW expansions of the scattering at a 2D lattice SWs in the x-y-plane and the expansion of the scattered waves at a lattice of CWs in the x-z-plane. We also implement operators to evaluate the electric, magnetic, or Riemann-Silberstein fields.

The operators can also be wrapped by the class `OperatorAttribute` and composed to a class to allow an alternative interface to the functionality by calling, e.g., in the example above, `mat.rotate(np.pi / 3)` in the last line of Listing 1 to achieve the same result.

3.3. *Physics-aware arrays, T-matrices, and S-matrices*

In this section, we will motivate and highlight some use cases of the class `PhysicsArray`. The `PhysicsArray` objects behave much like *numpy* arrays and can, when necessary, always be converted to them. However, they keep track of a set of relevant parameters attached to them during the computation. This set, also listed in Table 2, currently contains the wave number, the material, the basis set, the polarization type, the mode type, and, for periodic structures, the underlying lattice and wave vector defining the phase relation between lattice sites. With polarization type, we refer to the helicity and parity basis. The mode type is relevant for the CWs and SWs, where it can be either regular or singular, and for the PWs if they are defined by two components to distinguish the direction ↑ or ↓ for the remaining third component. A set of these is attached to each dimension of the `PhysicsArray` and then compared during computations: most binary operations, such as addition or elementwise multiplications, compare the same dimensions; special operations, most importantly matrix multiplications, compare the last dimension of the first operand with the first dimension of the last operand. In simple cases, a single value can be attached to all dimensions, as shown here

Listing 2: Preservation of general attributes in `PhysicsArray`

```
>>> import treams
>>> treams.PhysicsArray([[0, 1], [2, 3]], k0=1.2)
PhysicsArray(
    [[0, 1],
     [2, 3]],
    k0=1.2,
)
>>> treams.PhysicsArray([[0, 1], [2, 3]], k0=1.2) + [4, 5]
PhysicsArray(
    [[ 4,  6],
     [ 6, 8]],
    k0=1.2,
)
```

where the wave number is kept track of throughout the computation. However, it's also possible to attach one value per dimension. Here, two different wave numbers are attached to the two dimensions of the array:

Listing 3: Preservation of attributes in `PhysicsArray` per dimension

```
>>> import treams
>>> treams.PhysicsArray([[0, 1], [2, 3]], k0=(1.2, 3.4))
PhysicsArray(
    [[0, 1],
     [2, 3]],
    k0=(1.2, 3.4),
)
>>> treams.PhysicsArray([[0, 1], [2, 3]], k0=(1.2, 3.4)) + [4, 5]
PhysicsArray(
    [[ 4,  6],
     [ 6, 8]],
    k0=(1.2, 3.4),
)
>>> treams.PhysicsArray([[0, 1], [2, 3]], k0=(1.2, 3.4)) @ [4, 5]
PhysicsArray(
    [ 5, 23],
    k0=1.2,
)
>>> treams.PhysicsArray([0, 1], k0=1.2) + treams.PhysicsArray([2, 3], k0=3.4)
AnnotationWarning: at index 0: overwriting key 'k0'
PhysicsArray(
    [2, 4],
    k0=1.2,
)
```

It can be seen that the addition keeps both values in the result. For the matrix-vector multiplication, the result is a vector where the wave number corresponds to the value of the first matrix dimension. The last command shows that if the parameters are not the same, a warning is issued to notify the user of potential errors.

These parameters are also exposed as attributes of the class. In total, they are sufficient to get a concrete representation of all operators described in the previous section. So, although the implementation of the class `Operator` and the associated `OperatorAttribute` is agnostic to the object which provides the parameters to get a concrete representation, the class `PhysicsArray` obviously is well suited to work with the implemented operators seamlessly.

Turning to the T-matrices now, we can build up on the class `PhysicsArray`, which already provides many useful operations. In fact, the class `TMatrix` derives from that class with additional restrictions on the parameters, for example, the basis set must be a `SphericalWaveBasis`. Also, it is supplemented with a range of additional methods to construct the T-matrix, e.g., from a single sphere by `TMatrix.sphere` or from a cluster of particles, and to evaluate its properties, e.g., the scattering cross-section or duality breaking. The function `TMatrix.sphere` can calculate the T-matrix for spheres and multiple layers of concentric spherical shells. The material parameters of both, the scatterers and the embedding medium, can contain a non-zero chirality. If the embedding medium is chiral, SWs of well-defined helicity are required. If the shape of the scatterer does not have a spherical symmetry, its T-matrix has to be provided from an external source, which is demonstrated in Section 4.2. *treams* supports external T-matrices in parity and helicity basis, and, thus, the scatterers and embedding medium can be chiral as long as the external tool also supports them.

In the very short example, we create the incident plane wave and the T-matrix of a core-shell particle, where the core is chiral.

Listing 4: T-matrix of a core-shell particle and

```
>>> import numpy as np
>>> import treams
>>> inc = treams.plane_wave([np.sin(np.pi / 4), 0, np.cos(np.pi / 4)], [0, 1, 0])
>>> mat_core = treams.Material(epsilon=4, kappa=0.1)
>>> mat_shell = treams.Material(epsilon=2 + 0.1j)
>>> tm = treams.TMatrix.sphere(
    lmax=6,
    k0=2 * np.pi,
    radii=[.4, .5],
    materials=[mat_core, mat_shell, 1]
)
>>> sca = (tm @ treams.Expand(basis=inc.basis).inv) @ inc
```

The scattered field is then calculated in the last line by

$$\boldsymbol{p}_{\mathrm{vsw}} = (\mathbf{T}_{\mathrm{vsw}}\mathbf{E}^{-1}_{\mathrm{vpw,vsw}})\boldsymbol{a}_{\mathrm{vpw}}\,, \tag{39}$$

Table 3
Covered examples in the online documentation.

Section	Example name	Description
T-matrices	Spheres	T-matrix of spheres and their cross-section
	Cluster	Global and local T-matrices of a finite cluster of particles
	One-dimensional arrays	Chains of particles and their interaction
	Two-dimensional arrays	Metasurface of spheres and their interaction
	Three-dimensional arrays	Mode calculation in a crystal
Cylindrical T-matrices	Cylinders	Cross-width of infinitely long cylinders
	Transition from T-matrices	Cylindrical T-matrix from a chain of spheres
	Cluster	Interaction of multiple cylindrical T-matrices
	One-dimensional arrays	A one-dimensional grating of cylindrical T-matrices
	Two-dimensional arrays	Eigenmodes in a two-dimensional array
S-Matrices	Slabs	Transmittance and reflectance of a chiral slab
	From T-matrices	Convert an array of T-matrices to an S-matrix
	From cylindrical T-matrices	Convert an array of cylindrical T-matrices to an S-matrix
	Band structure	Band structure calculation in periodic media

where we added subscripts to indicate the type of basis. The matrix $\mathbf{E}^{-1}_{\text{vpw,vsw}}$ contains the expansion coefficients of Eq. (31). Note that the SW basis or the wave number is not explicitly specified but taken from the T-matrix object `tm`. If we continued in the example, we could now evaluate the electric field at specific points with `sca.efield`. Similar to the T-matrices for SWs, the class `TMatrixC` implements functions for T-matrices in the `CylindricalWaveBasis`. The T-matrix can be solved analytically for cylinders, possibly with multiple concentric cylindrical shells, which is implemented in *treams*as `TMatrixC.cylinder`. The primary usage of the class is similar to the one shown in Listing 4. Similar to the function `TMatrix.sphere`, the function `TMatrixC.cylinder` can accommodate chiral material parameters for the scatterer and the embedding medium. The T-matrices in the CW basis for scatterers with shapes that have no cylindrical symmetry have to be computed externally.

The S-matrix in the PW basis has a somewhat special role. A single `SMatrix` holds the coefficients for incoming waves propagating in one particular direction scattered into outgoing waves of one direction. However, as seen in Eq. (29), a full description needs four S-matrices that cover all combinations of directions. Thus, an instance of the class `SMatrices` holds four instances of `SMatrix` with matching parameters.

All the different classes shown here have many more methods implemented. However, we refer to the online documentation (https://tfp-photonics.github.io/treams) for detailed information on all functionalities. Especially, it also includes a range of simple examples with full scripts, as listed in Table 3. These examples are automatically rerun with each new version of *treams* to ensure they stay up-to-date.

4. Examples

To demonstrate some capabilities of *treams*, we present in the following a range of examples in the following. We start by demonstrating the different lattice sums and how they complement each other. Then, we show a more practical example of finding bound states in the continuum in metasurfaces.

4.1. Complex unit cells in 1D and 2D lattices

This section contains two examples: a 2D array of a unit cell consisting of two different spheres and a combination of spheres and cylinders in another array. These systems are shown in Fig. 4. Each panel shows one electric field component in three cross-sections through the domain.

In panel (a), the simulated system consists of gold spheres with radii 225 nm and 195 nm in an array with 900 nm lattice pitch. They are positioned in the unit cell with a relative shift of $\boldsymbol{r} = (240\,\text{nm}, 300\,\text{nm}, 360\,\text{nm})$ from the center of the bigger to the center of the smaller sphere. The lattice is illuminated with a TE polarized plane wave and oblique incidence at the wavelength 600 nm. One electric field component is shown in the sections through the computational domain. The shown values are spliced into different parts. Between the spheres, only the SW expansion is valid, and lattice sums translating the scattered fields from all particles to each probed point are necessary. This requires highly efficient summation methods. The region above and below the array are computed with PWs. The absence of visible discontinuities, where the splicing occurs, shows how well these different expansions fit together. Another line, where the expansions are spliced, is at the boundary between the four shown elementary unit cells. Furthermore, there, these domains are seamlessly spliced.

Panel (b) shows a slightly more complicated system that includes SWs, CWs, and PWs. It consists of gold spheres and cylinders that are shifted by $\boldsymbol{r} = (300\,\text{nm}, 360\,\text{nm}, 240\,\text{nm})$ to each other. First, the spheres are coupled together as a chain along the z-axis and transformed from a SW basis to CWs. Then a complex unit cell containing the chain and additionally the cylinder is formed using the CWs. Again, we splice four unit cells and also different parts depending on the convergence domains of the different sums. Above and below the spheres, only the SW expansion is valid. Between the cylinders and the spheres, the CW expansion is used. The PW expansion is used for smaller and larger values of y according to Fig. 1(c).

4.2. Quasi-bound states in the continuum

This example is concerned with finding quasi-BICs by introducing asymmetry in two-dimensional arrangements of particles. The complete script to reproduce the data using *treams* is provided in Appendix G. The T-matrix is provided in the supplementary materials together with all necessary files to reproduce it. The unit cell of the lattice is rectangular with side length $a_x = 500\,\text{nm}$ and $a_y = 1000\,\text{nm}$ and contains two identical particles, each described by a T-matrix of maximal multipole order 6. The T-matrix is computed using the software JCMsuite [47] and its built-in functionality to evaluate the multipole decomposition of the scattered field in a finite element method simulation. We verify the numerical convergence of the finite element method simulation by running it additionally with a coarser mesh yielding, essentially, the same result for the computed T-matrices. Furthermore, we confirm the use of a sufficient number of multipole orders by comparing it with the result of multipole order 5. The results of these convergence studies are presented in Appendix F. The particles are ellipsoids with short radius $2^{-\frac{1}{4}} \times 210\,\text{nm}$ and long radius $2^{\frac{1}{4}} \times 210\,\text{nm}$. Thus, the aspect ratio of these prolate ellipsoids is $\sqrt{2}$. The relative permittivity is set to 12.25, mimicking silicon in the wavelength range between 1000 nm

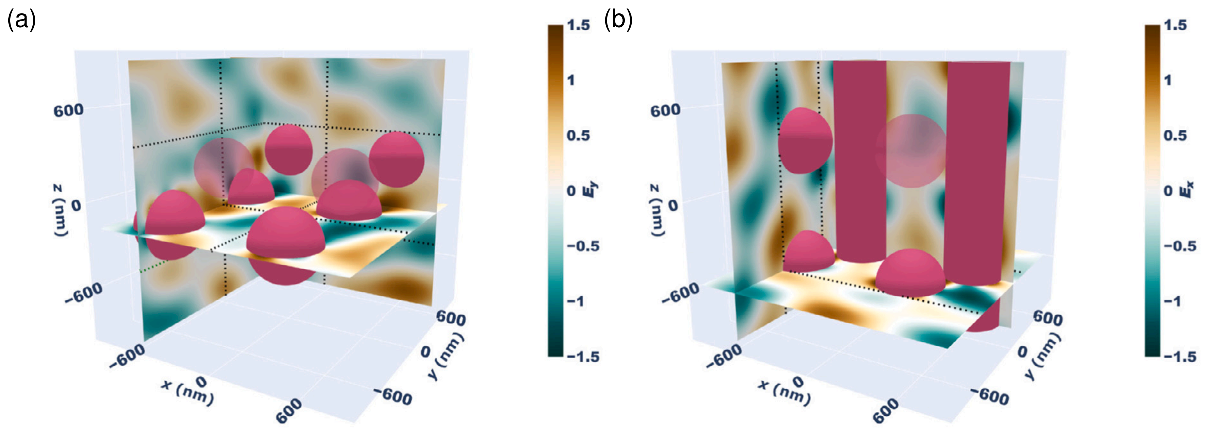

Fig. 4. The electric field in two different periodic arrangements. The scatterers are shown in red, and one electric field component is shown in three cross-sections through the 3D domain. Scatterers outside of these cross-sections are shown semi-transparent. Panel (a) shows four unit cells of a 2D lattice of spheres in the x-y-plane. The unit cell of the square lattice with a pitch of 900 nm consists of two spheres with radii 225 nm and 195 nm. The vector $\boldsymbol{r} = (240\,\text{nm}, 300\,\text{nm}, 360\,\text{nm})$ points from the center of the bigger sphere to the center of the smaller sphere. Both spheres are modeled with a relative permittivity $\epsilon = -9.4 + 1.5\text{i}$ at the frequency 600 THz corresponding to the properties of gold [46]. The embedding medium is vacuum. The distance between the spheres in the z-direction is 360 nm. Therefore, they cannot be coupled by using an intermediate PW expansion. They must be treated as a whole with a SW expansion in a lattice using a complex unit cell. To create the electric field plot, different domains of validity are spliced. The fields above and below the bounding planes for the whole unit cell are computed in the PW expansion. The fields between these planes – the region between the spheres – have to be computed by calculating lattice sums of SWs. The repeated calculation of these lattice sums, with and without shifts perpendicular to the lattice plane, requires efficient implementations. Panel (b) shows another example of the electric field calculation. Four unit cells of a 2D lattice in the x-z-plane are shown. The lattice pitch is 900 nm. The sphere has the radius 225 nm and the cylinder has the radius 195 nm. Both objects are modeled as gold like in panel (a). This calculation is done by combining all three basis sets. First, the coupling within 1D chains of spheres along the z-axis is calculated, and the result is expressed as a cylindrical T-matrix. Then, a complex unit cell of the sphere chain and the cylinder is formed, and their coupling is calculated for the periodic arrangement in the x-direction. Due to the different regions of validity, this step-wise procedure is necessary. The electric field is now spliced from PW expansions outside the bounding rectangular region. We can use the CW expansion between the solid cylinder and the circumscribing cylinder for the chain of spheres. Between the spheres themselves, only the SW expansion is valid. The absence of sharp features at the bounding boxes of the different domains indicates that the methods complement each other well when using sufficiently many expansion orders. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

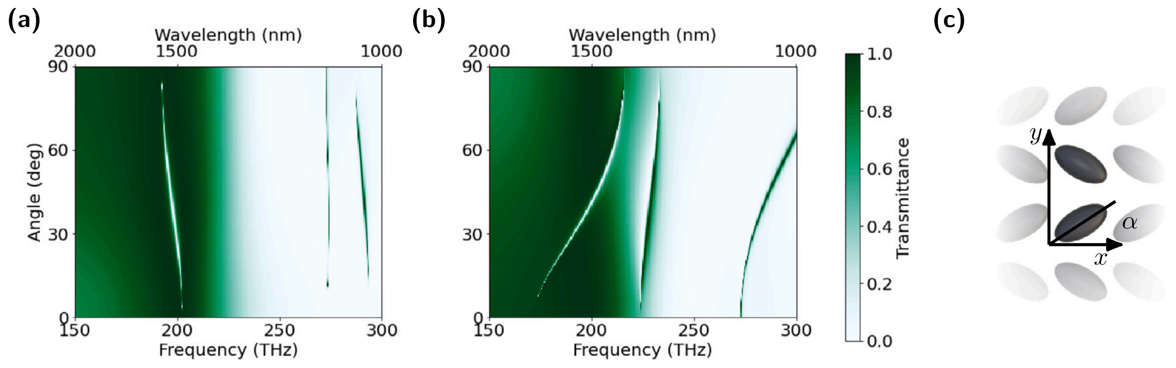

Fig. 5. Quasi-BICs in a two-dimensional array of ellipsoids. Each unit cell contains two ellipsoids, which are rotated by an angle α to the x-axis. When the angle α is not zero or ninety degrees, the BIC in the lattice becomes a quasi-BIC and is thus visible in the transmission spectrum. Panels (a) and (b) show the result for normal incidence and linearly polarized light along the x- or y-axis, respectively. Panel (c) shows a unit cell of the lattice.

and 2000 nm. The unit cell is shown in Fig. 5(c) together with the coordinate axes and the definition of the angle α for the asymmetry. The panels (a) and (b) of Fig. 5 show the transmittance for linearly x- and y-polarized light. In the case of zero and ninety degrees, the BICs are not visible due to the symmetry of the lattice, but with increasing angle α, they become quasi-BICs and appear as fine lines in the transmission spectrum. At 1500 nm and 38° angle, the two quasi-BIC lines for the two polarizations, shown in panels (a) and (b), are crossing.

5. Conclusion

We presented the program *treams* that provides a comprehensive framework to perform scattering calculations for a wide range of scenarios. Among the main features are the possibilities to perform T-matrix calculation in periodic systems with complex unit cells, i.e., unit cells with multiple objects described in a local basis, and the inclusion of chiral material parameters. Chiral materials are fully supported: They can be used in the scatterers as well as the embedding medium. The required lattice sums are implemented using exponentially fast converging series which we derived previously [13]. Besides the case of periodic boundaries, standard T-matrix calculations for finite systems can also be done with typical operations such as rotations and translations included. Besides T-matrices using SWs, also CWs are implemented. Scattering in stratified media is implemented using an S-matrix description for PWs. The functionality is provided in two ways: a low-level interface that provides maximum speed and flexibility and a high-level interface that provides more convenient access to these functions and additional plausibility and error checks. Furthermore, the T-matrices can be stored and loaded in an HDF5 file format. The package *treams* is open-source with permissive licensing. Moreover, it is accompanied by an extensive online documentation, including many examples and a large set of tests. By using continuous integration, the documentation, doctests included in the documentation, and tests are automatically run. Building pre-compiled packages for all three major operating systems, Windows, MacOS, and Linux, is also part of the automation pipeline and are available from the official Python Package Index.

There are multiple future avenues for the development of *treams*. Regarding speed, making use of parallelization and GPU computation are apparent improvements. Adding automatic differentiation would enhance the capabilities of the program regarding possible optimization schemes.

We showed several examples where *treams* can be applied: solving multi-scattering problems with arbitrary periodic boundary conditions with complex unit cells and finding quasi-BICs by breaking symmetries in a lattice. Furthermore, the methods described and implemented were used in various other publications, using several of its unique capabilities, such as the inclusion of chiral matter or the efficient stacking of S-matrices: Those include the design of cavities for enhanced sensing of chiral molecules [48–50], multi-scale simulation of molecular arrays in cavities [51,52], T-matrix based homogenization [53], the analysis of flat bands in moiré structures [54] and analytical models for metasurfaces [55].

CRediT authorship contribution statement

Dominik Beutel: Conceptualization, Investigation, Methodology, Software, Visualization, Writing – original draft. **Ivan Fernandez-Corbaton:** Conceptualization, Methodology, Supervision, Writing – review & editing. **Carsten Rockstuhl:** Conceptualization, Methodology, Supervision, Writing – review & editing.

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.

Data availability

Data will be made available on request.

Acknowledgements

D.B. and C.R. acknowledge support by the Deutsche Forschungsgemeinschaft (German Research Foundation) under Germany's Excellence Strategy via the Excellence Cluster 3D Matter Made to Order (Grant No. EXC – 2082/1 – 390761711) and from the Carl Zeiss Foundation via CZF-Focus@HEiKA. The authors are grateful to the company JCMwave for their free provision of the FEM Maxwell solver JCMsuite. The authors would like to thank Stefan Mühlig, Martin Fruhnert, and Achim Groner that developed earlier versions of similar codes.

Appendix A. Cylindrical and spherical coordinates

We use the convention

$$x = \rho\cos\varphi = r\sin\theta\cos\varphi \tag{A.1a}$$

$$y = \rho\sin\varphi = r\sin\theta\sin\varphi \tag{A.1b}$$

$$z = r\cos\theta \tag{A.1c}$$

for the definition of the cylindrical coordinates ρ, φ, and z and of the spherical coordinates r, θ, and φ. Note that the definition of φ coincides in these coordinate sets. The unit vectors are defined by

$$\hat{\boldsymbol{\rho}} = \cos\varphi\hat{\boldsymbol{x}} + \sin\varphi\hat{\boldsymbol{y}} \tag{A.2a}$$

$$\hat{\boldsymbol{r}} = \sin\theta\cos\varphi\hat{\boldsymbol{x}} + \sin\theta\sin\varphi\hat{\boldsymbol{y}} + \cos\theta\hat{\boldsymbol{z}} \tag{A.2b}$$

$$\hat{\boldsymbol{\theta}} = \cos\theta\cos\varphi\hat{\boldsymbol{x}} + \cos\theta\sin\varphi\hat{\boldsymbol{y}} - \sin\theta\hat{\boldsymbol{z}} \tag{A.2c}$$

$$\hat{\boldsymbol{\varphi}} = -\sin\varphi\hat{\boldsymbol{x}} + \cos\varphi\hat{\boldsymbol{y}}\,. \tag{A.2d}$$

If these coordinates and unit vectors are used for wave vector $\boldsymbol{k}$ instead of the real space coordinates $\boldsymbol{r}$, we use it as a subscript, for example, $\hat{\boldsymbol{\varphi}}_{\boldsymbol{k}}$. The exception is the radial vector $\hat{\boldsymbol{k}}$.

Appendix B. Spherical harmonics and associated Legendre polynomials

We define the spherical harmonics by

$$Y_{lm}(\theta,\varphi) = \sqrt{\frac{2l+1}{4\pi}\frac{(l-m)!}{(l+m)!}}P_l^m(\cos\theta)\mathrm{e}^{\mathrm{i}m\varphi}\,, \tag{B.1}$$

where the associated Legendre polynomials are given by

$$P_l^m(x) = \frac{(-1)^m}{2^l l!}\left(1-x^2\right)^{\frac{m}{2}}\frac{\mathrm{d}^{l+m}}{\mathrm{d}x^{l+m}}(x^2-1)^l \tag{B.2}$$

for $l \in \mathbb{N}_0$ and $m \in \{-l, -l+1, \dots, l\}$.

Appendix C. Translation coefficients for vector spherical and cylindrical waves

The translation coefficients for SWs [30–32]

$$A^{(n)}_{l'm'lm}(\boldsymbol{r},k) = (-1)^m \mathrm{i}^{l'-l}\sqrt{\pi \frac{2l'+1}{l'(l'+1)}\frac{2l+1}{l(l+1)}}$$
$$\cdot \sum_p \mathrm{i}^p \sqrt{2p+1}\, z^{(n)}_p(kr) Y_{p,m-m'}(\theta,\varphi) \begin{pmatrix} l & l' & p \\ m & -m' & -m+m' \end{pmatrix}\begin{pmatrix} l & l' & p \\ 0 & 0 & 0 \end{pmatrix}\left[l(l+1)+l'(l'+1)-p(p+1)\right] \tag{C.1a}$$

$$B^{(n)}_{l'm'lm}(k\boldsymbol{r}) = (-1)^m \mathrm{i}^{l'-l}\sqrt{\pi \frac{2l'+1}{l'(l'+1)}\frac{2l+1}{l(l+1)}}$$
$$\cdot \sum_p \mathrm{i}^p \sqrt{2p+1}\, z^{(n)}_p(kr) Y_{p,m-m'}(\theta,\varphi) \begin{pmatrix} l & l' & p \\ m & -m' & -m+m' \end{pmatrix}\begin{pmatrix} l & l' & p-1 \\ 0 & 0 & 0 \end{pmatrix}\sqrt{\left[(l+l'+1)^2-p^2\right]\left[p^2-(l-l')^2\right]}\,, \tag{C.1b}$$

where the sum index p runs over all values where the Wigner 3j-symbols are non-zero, are used to expand fields in the same type, namely incident in incident fields and scattered in scattered fields, using

$$\boldsymbol{A}^{(n)}_{lm,\pm}(\boldsymbol{r},k) = \sum_{l=1}^{\infty}\sum_{m=-l}^{l}\left(A^{(1)}_{l'm',lm}(\boldsymbol{r}-\boldsymbol{r}',k) \pm B^{(1)}_{l'm',lm}(\boldsymbol{r}-\boldsymbol{r}',k)\right)\boldsymbol{A}^{(n)}_{l'm',\pm}(\boldsymbol{r}',k) \tag{C.2}$$

and to expand scattered fields in incident fields by

$$\boldsymbol{A}^{(3)}_{lm,\pm}(\boldsymbol{r},k) = \sum_{l=1}^{\infty}\sum_{m=-l}^{l}\left(A^{(3)}_{l'm',lm}(\boldsymbol{r}-\boldsymbol{r}',k) \pm B^{(3)}_{l'm',lm}(\boldsymbol{r}-\boldsymbol{r}',k)\right)\boldsymbol{A}^{(1)}_{l'm',\pm}(\boldsymbol{r}',k)\,. \tag{C.3}$$

With a similar structure but simpler expressions, we have

$$\boldsymbol{A}^{(n)}_{k_z m,\pm}(\boldsymbol{r},k) = \sum_{m=-\infty}^{\infty} J_{m-m'}\left(\sqrt{k^2-k_z^2}\rho_{\boldsymbol{r}-\boldsymbol{r}'}\right)\mathrm{e}^{\mathrm{i}(m-m')\varphi_{\boldsymbol{r}-\boldsymbol{r}'}+\mathrm{i}k_z(z-z')}\boldsymbol{A}^{(n)}_{k_z m',\pm}(\boldsymbol{r}',k) \tag{C.4}$$

and

$$\boldsymbol{A}^{(3)}_{k_z m,\pm}(\boldsymbol{r},k) = \sum_{m=-\infty}^{\infty} H^{(1)}_{m-m'}\left(\sqrt{k^2-k_z^2}\rho_{\boldsymbol{r}-\boldsymbol{r}'}\right)\mathrm{e}^{\mathrm{i}(m-m')\varphi_{\boldsymbol{r}-\boldsymbol{r}'}+\mathrm{i}k_z(z-z')}\boldsymbol{A}^{(1)}_{k_z m',\pm}(\boldsymbol{r}',k) \tag{C.5}$$

for CWs [15].

Appendix D. Permutation of Cartesian axes

Due to the different alignment of the arrays for the different basis sets, it is necessary to permute the labels of the Cartesian axes. As an example, the 1D lattice of CWs is in the x-z-plane, but the 2D lattice of SWs is conventionally placed in the x-y-plane. To couple such systems, it is necessary to permute the corresponding basis sets. Also, the definition of the S-matrices is mostly suited to x-y-periodicity. Therefore, we need the transformation from a coordinate system with x, y, and z to a system with $x = y'$, $y = z'$, and $z = x'$. Thus, the wave vector is $\boldsymbol{k} = k_x\hat{\boldsymbol{x}} + k_y\hat{\boldsymbol{y}} + k_z\hat{\boldsymbol{z}} = k_z\hat{\boldsymbol{x}}' + k_x\hat{\boldsymbol{y}}' + k_y\hat{\boldsymbol{z}}'$ in these systems. This change leaves the scalar product $\boldsymbol{k}\boldsymbol{r}$ and, so, the exponential of the PWs invariant. However, the polarization vectors $\boldsymbol{m}_{\boldsymbol{k}}$ and $\boldsymbol{n}_{\boldsymbol{k}}$ of the PWs $\boldsymbol{M}_{\boldsymbol{k}}$ and $\boldsymbol{N}_{\boldsymbol{k}}$ change their definition.

In the original system, we have

$$\boldsymbol{m}_{\boldsymbol{k}} = \mathrm{i}\frac{k_y\hat{\boldsymbol{x}} - k_x\hat{\boldsymbol{y}}}{k_{xy}} \tag{D.1a}$$

$$\boldsymbol{n}_{\boldsymbol{k}} = \frac{-k_xk_z\hat{\boldsymbol{x}} - k_yk_z\hat{\boldsymbol{y}} + k_{xy}^2\hat{\boldsymbol{z}}}{kk_{xy}}\,, \tag{D.1b}$$

with $k_{xy} = \sqrt{k_x^2+k_y^2}$, whereas the same definition leads to

$$\boldsymbol{m}'_{\boldsymbol{k}} = \mathrm{i}\frac{k_x\hat{\boldsymbol{x}}' - k_z\hat{\boldsymbol{y}}'}{k_{xz}} = \mathrm{i}\frac{k_x\hat{\boldsymbol{z}} - k_z\hat{\boldsymbol{x}}}{k_{xz}} \tag{D.2a}$$

$$\boldsymbol{n}'_{\boldsymbol{k}} = \frac{-k_zk_y\hat{\boldsymbol{x}}' - k_xk_y\hat{\boldsymbol{y}}' + k_{xz}^2\hat{\boldsymbol{z}}'}{kk_{xz}} = \frac{-k_zk_y\hat{\boldsymbol{z}} - k_xk_y\hat{\boldsymbol{x}} + k_{xz}^2\hat{\boldsymbol{y}}}{kk_{xz}} \tag{D.2b}$$

in the new coordinate system with k_{xz} defined analogous to k_{xy}. Now, we can calculate the projection of the polarizations $\boldsymbol{m}_{\boldsymbol{k}}$ and $\boldsymbol{n}_{\boldsymbol{k}}$ onto the new polarizations $\boldsymbol{m}'_{\boldsymbol{k}}$ and $\boldsymbol{m}'_{\boldsymbol{k}}$. Remembering the minus sign, when projecting onto $\boldsymbol{m}'_{\boldsymbol{k}}$ we obtain

$$-\boldsymbol{m}'_{\boldsymbol{k}}\boldsymbol{m}_{\boldsymbol{k}} = -\frac{k_yk_z}{k_{xy}k_{xz}} \tag{D.3a}$$

$$-\boldsymbol{m}'_{\boldsymbol{k}}\boldsymbol{n}_{\boldsymbol{k}} = -\mathrm{i}\frac{k_xk_{xy}^2 + k_xk_z^2}{kk_{xy}k_{xz}} = -\mathrm{i}\frac{k_xk}{k_{xy}k_{xz}} \tag{D.3b}$$

$$\boldsymbol{n}'_{\boldsymbol{k}}\boldsymbol{m}_{\boldsymbol{k}} = -\mathrm{i}\frac{k_x k_y^2 + k_x k_{xz}^2}{k k_{xy} k_{xz}} = -\mathrm{i}\frac{k_x k}{k_{xy} k_{xz}} \tag{D.3c}$$

$$\boldsymbol{n}'_{\boldsymbol{k}}\boldsymbol{n}_{\boldsymbol{k}} = \frac{-k_z k_y k_{xy}^2 + k_z k_y k_x^2 - k_z k_y k_{xz}^2}{k^2 k_{xy} k_{xz}} = -\frac{k_y k_z}{k_{xy} k_{xz}}\,. \tag{D.3d}$$

So, the expansion coefficients of the PWs in the original system get transformed by multiplying with the matrix

$$\frac{1}{k_{xy}k_{xz}}\begin{pmatrix} -k_y k_z & -\mathrm{i}k k_x \\ -\mathrm{i}k k_x & -k_y k_z \end{pmatrix}\,. \tag{D.4}$$

In the special case that $k_x = k_y = 0$, the matrix becomes

$$\begin{pmatrix} 0 & -\mathrm{i} \\ -\mathrm{i} & 0 \end{pmatrix} \tag{D.5}$$

and for $k_x = k_z = 0$ it is

$$\begin{pmatrix} -\frac{k_y}{k} & 0 \\ 0 & -\frac{k_y}{k} \end{pmatrix}\,. \tag{D.6}$$

In the helicity basis, the two helicities are not mixed, but the expansion coefficient values have to be adjusted by multiplying with $\frac{-k_y k_z \mp \mathrm{i}k k_x}{k_{xy}k_{xz}}$ for helicity ± 1. The matrix for the inverse transformation $x = z''$, $y = x''$, and $z = y''$

$$\frac{1}{k_{xy}k_{yz}}\begin{pmatrix} -k_x k_z & \mathrm{i}k k_y \\ \mathrm{i}k k_y & -k_x k_z \end{pmatrix} \tag{D.7}$$

can be obtained by applying the forward transformation twice with a correct relabelling of the entries.

Appendix E. Expansions of lattice sums in other basis sets

To find expansions of periodic arrangements of scattered CWs of SWs in other basis sets, we use the integral expressions [41]

$$\boldsymbol{M}^{(3)}_{lm}(\boldsymbol{r},k) = \frac{1}{2\pi\mathrm{i}^l}\iint\limits_{\mathbb{R}^2}\frac{\mathrm{d}k_x\mathrm{d}k_y}{k^2\gamma_{xy}}\boldsymbol{X}_{lm}(\theta_{\boldsymbol{k}},\varphi_{\boldsymbol{k}})\mathrm{e}^{\mathrm{i}(k_x x + k_y y + k_z z)} = \frac{1}{2\pi\mathrm{i}^l}\iint\limits_{\mathbb{R}^2}\frac{\mathrm{d}k_x\mathrm{d}k_z}{k^2\gamma_{xz}}\boldsymbol{X}_{lm}(\theta_{\boldsymbol{k}},\varphi_{\boldsymbol{k}})\mathrm{e}^{\mathrm{i}(k_x x + k_y y + k_z z)} \tag{E.1a}$$

$$\boldsymbol{N}^{(3)}_{lm}(\boldsymbol{r},k) = \frac{1}{2\pi\mathrm{i}^{l-1}}\iint\limits_{\mathbb{R}^2}\frac{\mathrm{d}k_x\mathrm{d}k_y}{k^2\gamma_{xy}}\hat{\boldsymbol{k}}\times\boldsymbol{X}_{lm}(\theta_{\boldsymbol{k}},\varphi_{\boldsymbol{k}})\mathrm{e}^{\mathrm{i}(k_x x + k_y y + k_z z)} = \frac{1}{2\pi\mathrm{i}^{l-1}}\iint\limits_{\mathbb{R}^2}\frac{\mathrm{d}k_x\mathrm{d}k_z}{k^2\gamma_{xz}}\hat{\boldsymbol{k}}\times\boldsymbol{X}_{lm}(\theta_{\boldsymbol{k}},\varphi_{\boldsymbol{k}})\mathrm{e}^{\mathrm{i}(k_x x + k_y y + k_z z)}\,, \tag{E.1b}$$

where $\gamma_{ij} = \sqrt{1 - \frac{k_i^2 + k_j^2}{k^2}}$ for the SWs. There is a branch cut in the expansions at the $z = 0$ plane for the first integral and $y = 0$ plane for the second integral representation. The unspecified component of the integrals, k_z in the first one and k_y in the second one, is defined such that the PW propagates or decays when moving away from the branch cut. Thus, it is $k_z = \pm k\gamma_{xy}$ for $z \gtrless 0$ in the first case and analogous for k_y in the second case. For the CWs, we use [40]

$$\boldsymbol{M}^{(3)}_{k_z,m}(\boldsymbol{r},k) = \frac{1}{\pi\mathrm{i}^{m+1}}\int_{-\infty}^{\infty}\frac{\mathrm{d}k_x}{k\gamma_{xz}}\hat{\boldsymbol{\varphi}}_{\boldsymbol{k}}\mathrm{e}^{\mathrm{i}(k_x x + k_y y + k_z z + m\varphi_{\boldsymbol{k}})} \tag{E.2a}$$

$$\boldsymbol{N}^{(3)}_{k_z,m}(\boldsymbol{r},k) = -\frac{1}{\pi\mathrm{i}^{m}}\int_{-\infty}^{\infty}\frac{\mathrm{d}k_x}{k\gamma_{xz}}\hat{\boldsymbol{\theta}}_{\boldsymbol{k}}\mathrm{e}^{\mathrm{i}(k_x x + k_y y + k_z z + m\varphi_{\boldsymbol{k}})}\,, \tag{E.2b}$$

with $k_y = \pm k\gamma_{xz}$ for $z \gtrless 0$. We will use Poisson's formula

$$\sum_{\boldsymbol{R}\in\Lambda_d}\mathrm{e}^{\mathrm{i}\boldsymbol{k}\boldsymbol{R}} = \frac{(2\pi)^d}{V_d}\sum_{\boldsymbol{G}\in\Lambda_d^*}\delta^{(d)}(\boldsymbol{k}-\boldsymbol{G}) \tag{E.3}$$

to evaluate lattice sums over these solutions. V_d is the generalized d-dimensional volume of a unit cell; in one dimension, it is equal to the lattice pitch a, and in two dimensions, it corresponds to the unit cell area A. This transformation is useful since the resulting delta distributions simplify the integrals.

Starting with SWs on a 2D lattice in the x-y-plane, we thereby get

$$\sum_{\boldsymbol{R}\in\Lambda_2}\boldsymbol{M}^{(3)}_{lm}(\boldsymbol{r}-\boldsymbol{R},k)\mathrm{e}^{\mathrm{i}\boldsymbol{k}_\parallel\boldsymbol{R}} = \frac{2\pi N_{lm}}{Ak^2\mathrm{i}^l}\sum_{\boldsymbol{G}\in\Lambda_2^*}\left(\mathrm{i}\pi_{lm}(\theta_{\boldsymbol{k}})\hat{\boldsymbol{\theta}}_{\boldsymbol{k}} - \tau_{lm}(\theta_{\boldsymbol{k}})\hat{\boldsymbol{\varphi}}_{\boldsymbol{k}}\right)\frac{\mathrm{e}^{\mathrm{i}(\boldsymbol{k}\boldsymbol{r}+m\varphi_{\boldsymbol{k}})}}{\sqrt{1-\frac{(\boldsymbol{k}_\parallel+\boldsymbol{G})^2}{k^2}}} \tag{E.4}$$

from combining Eqs. (E.1) and (E.3), where $\boldsymbol{k} = \boldsymbol{k}_\parallel + \boldsymbol{G} \pm \sqrt{k^2 - (\boldsymbol{k}_\parallel + \boldsymbol{G})^2}\hat{\boldsymbol{z}}$ for $z \gtrless 0$ on the right-hand side. We can directly identify the PWs now. Similarly, we can derive the relations for the TM modes. In total, the result can be written as

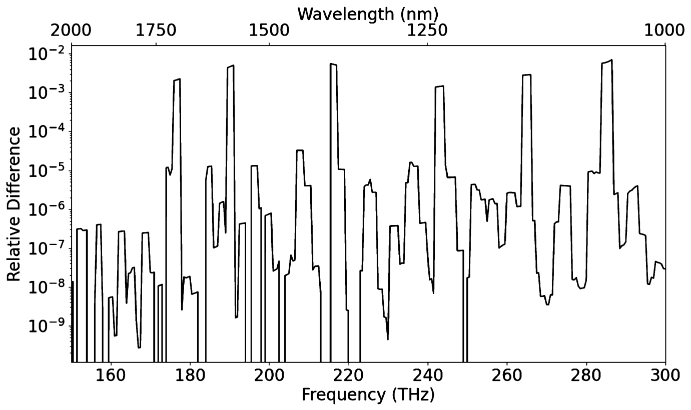

Fig. F.6. Difference in the T-matrices computed with the finite element method and different mesh element sizes.

$$\sum_{\boldsymbol{R}\in\Lambda_2}\begin{pmatrix}\boldsymbol{M}^{(3)}_{lm}(\boldsymbol{r}-\boldsymbol{R},k)\\ \boldsymbol{N}^{(3)}_{lm}(\boldsymbol{r}-\boldsymbol{R},k)\end{pmatrix}\mathrm{e}^{\mathrm{i}\boldsymbol{k}_\parallel \boldsymbol{R}} = -\frac{2\pi\mathrm{i}N_{lm}}{Ak^2\mathrm{i}^l}\sum_{\boldsymbol{G}\in\Lambda_2^*}\frac{\mathrm{e}^{\mathrm{i}m\varphi_{\boldsymbol{k}}}}{\sqrt{1-\frac{(\boldsymbol{k}_\parallel+\boldsymbol{G})^2}{k^2}}}\begin{pmatrix}\tau_{lm}(\theta_{\boldsymbol{k}}) & \pi_{lm}(\theta_{\boldsymbol{k}})\\ \pi_{lm}(\theta_{\boldsymbol{k}}) & \tau_{lm}(\theta_{\boldsymbol{k}})\end{pmatrix}\begin{pmatrix}\boldsymbol{M}_{\boldsymbol{k}_\parallel+\boldsymbol{G},\mathrm{sign}(z)}(\boldsymbol{r},k)\\ \boldsymbol{N}_{\boldsymbol{k}_\parallel+\boldsymbol{G},\mathrm{sign}(z)}(\boldsymbol{r},k)\end{pmatrix} \tag{E.5}$$

and after a change to helicity basis we get Eq. (35).

For SWs on a 1D lattice along the z-axis, we use the other form of the integral representation and get

$$\sum_{\boldsymbol{R}\in\Lambda_1}\boldsymbol{M}^{(3)}_{lm}(\boldsymbol{r}-\boldsymbol{R},k)\mathrm{e}^{\mathrm{i}\boldsymbol{k}_\parallel \boldsymbol{R}} = \frac{N_{lm}}{ak\mathrm{i}^l}\sum_{G\in\Lambda_1^*}\int_{-\infty}^{\infty}\frac{\mathrm{d}k_x\mathrm{e}^{\mathrm{i}(\boldsymbol{kr}+m\varphi_{\boldsymbol{k}})}}{k\sqrt{1-\frac{k_x^2+(k_\parallel+G)^2}{k^2}}}\left(\mathrm{i}\pi_{lm}(\theta_{\boldsymbol{k}})\hat{\boldsymbol{\theta}}_{\boldsymbol{k}}-\tau_{lm}(\theta_{\boldsymbol{k}})\hat{\boldsymbol{\varphi}}_{\boldsymbol{k}}\right), \tag{E.6}$$

where we used that $\boldsymbol{G}=G\hat{\boldsymbol{z}}$ and $\boldsymbol{k}_\parallel = k_\parallel\hat{\boldsymbol{z}}$. The wave vector is $\boldsymbol{k}=k_x\hat{\boldsymbol{x}}\pm\sqrt{k^2-k_x^2-(k_\parallel+G)^2}+(k_\parallel+G)\hat{\boldsymbol{z}}$ for $y\gtrless 0$. Here, we can identify the integral representation of the CWs in Eq. (E.2). Again, an equivalent procedure for the TM mode leads to the total result

$$\sum_{\boldsymbol{R}\in\Lambda_1}\begin{pmatrix}\boldsymbol{M}^{(3)}_{lm}(\boldsymbol{r}-\boldsymbol{R},k)\\ \boldsymbol{N}^{(3)}_{lm}(\boldsymbol{r}-\boldsymbol{R},k)\end{pmatrix}\mathrm{e}^{\mathrm{i}\boldsymbol{k}_\parallel \boldsymbol{R}} = -\frac{\mathrm{i}\pi N_{lm}}{ak\mathrm{i}^{l-m}}\sum_{G\in\Lambda_1^*}\begin{pmatrix}\tau_{lm}(\theta_{\boldsymbol{k}}) & \pi_{lm}(\theta_{\boldsymbol{k}})\\ \pi_{lm}(\theta_{\boldsymbol{k}}) & \tau_{lm}(\theta_{\boldsymbol{k}})\end{pmatrix}\begin{pmatrix}\boldsymbol{M}^{(3)}_{k_\parallel+G,m}(\boldsymbol{r},k)\\ \boldsymbol{N}^{(3)}_{k_\parallel+G,m}(\boldsymbol{r},k)\end{pmatrix}, \tag{E.7}$$

which can then be converted to Eq. (34).

Finally, the CWs on a 1D lattice along the x-axis can be rewritten as

$$\sum_{\boldsymbol{R}\in\Lambda_1}\boldsymbol{M}_{k_z,m}(\boldsymbol{r}-\boldsymbol{R},k)\mathrm{e}^{\mathrm{i}\boldsymbol{k}_\parallel \boldsymbol{R}} = \frac{2}{ak\mathrm{i}^{m+1}}\sum_{G\in\Lambda_1^*}\frac{\mathrm{e}^{\mathrm{i}(\boldsymbol{kr}+m\varphi_{\boldsymbol{k}})}}{\sqrt{1-\frac{(k_\parallel+G)^2+k_z^2}{k^2}}}\hat{\boldsymbol{\varphi}}_{\boldsymbol{k}}, \tag{E.8}$$

where we use $\boldsymbol{k}_\parallel = k_\parallel\hat{\boldsymbol{x}}$, $\boldsymbol{G}=G\hat{\boldsymbol{x}}$, and $\boldsymbol{k}=(k_\parallel+G)\hat{\boldsymbol{x}}\pm\sqrt{k^2-(k_\parallel+G)^2-k_z^2}\hat{\boldsymbol{y}}+k_z\hat{\boldsymbol{z}}$ for $y\gtrless 0$ and in total we obtain

$$\sum_{\boldsymbol{R}\in\Lambda_1}\begin{pmatrix}\boldsymbol{M}^{(3)}_{k_z,m}(\boldsymbol{r}-\boldsymbol{R},k)\\ \boldsymbol{N}^{(3)}_{k_z,m}(\boldsymbol{r}-\boldsymbol{R},k)\end{pmatrix}\mathrm{e}^{\mathrm{i}\boldsymbol{k}_\parallel \boldsymbol{R}} = \frac{2}{ak\mathrm{i}^{m}}\sum_{G\in\Lambda_1^*}\frac{1}{\sqrt{1-\frac{(k_\parallel+G)^2+k_z^2}{k^2}}}\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}\boldsymbol{M}_{\hat{\boldsymbol{k}}}(\boldsymbol{r},k)\\ \boldsymbol{N}_{\hat{\boldsymbol{k}}}(\boldsymbol{r},k)\end{pmatrix}, \tag{E.9}$$

which is Eq. (36) in helicity basis.

Appendix F. Convergence study for the quasi-BIC example

To assess the convergence of the T-matrices calculated with the finite element method, we perform two tests. First, we assess the convergence of the finite element calculation by comparing meshing with different maximal side lengths: A converged result for the electric field and its decomposition into SWs should be independent of the maximal size of mesh elements. Second, we verify the sufficient convergence of the multipole expansion by removing the highest-order multipoles from the expansion and checking the differences in the obtained results.

In addition to the T-matrix calculated for the example of Section 4.2, we repeat the same calculation with a coarser mesh. To compare the T-matrices, we calculate $|\mathbf{T}-\mathbf{T}_{\text{coarse}}|$, where we use the Hilbert-Schmidt matrix norm. The result is shown in Fig. F.6. We find a very good agreement between the T-matrices with a maximal difference of approximately 10^{-3}. For most frequencies, the difference is in the order of 10^{-7}, which corresponds roughly to the number of significant digits of single-precision calculations.

To verify a sufficient number of included modes, we rerun the calculation of Fig. 5 where we take the T-matrix with a maximal multipole order reduced by one, i.e., Fig. F.7 shows the result with $l_{\max}=5$ instead of $l_{\max}=5$. A direct comparison of these results is shown in Fig. F.8. We find that, especially for the smaller frequencies, there is no difference between the two results. At larger frequencies, we find a difference, which, however, can be attributed to slight shifts of the narrow resonances.

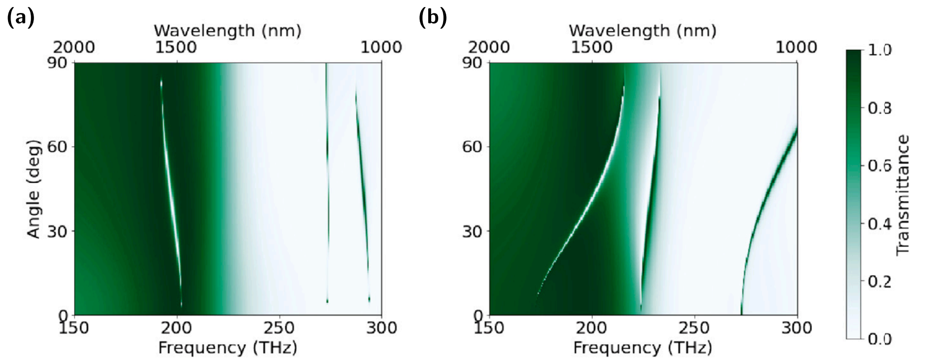

Fig. F.7. The result for the same study as shown in Fig. 5 but with the maximal multipole order set to 5. We find the same result as in Fig. 5. Fig. F.8 shows the absolute value of the difference between both results.

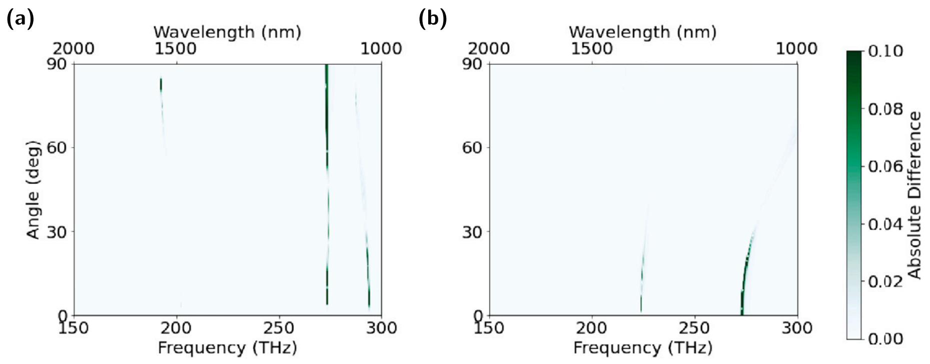

Fig. F.8. Absolute value of the difference of the transmittance calculated with maximal multipole orders 5 and 6. The panels (a) and (b) correspond to panels (a) and (b) in Figs. 5 and F.7. We find, in general, a very good agreement of the results, especially for the result in the lower frequencies, where the quasi-BIC appears. For larger frequencies, we find differences. However, they can be attributed to slight shifts in the position of the resonances.

Appendix G. Script for the quasi-BIC example

The following script can be used to recalculate the data for Fig. 5

Listing 5: Commented quasi-BIC calculation

```python
import pickle

import joblib

import numpy as np
import treams
import treams.io


def run_xy(tm, lat, positions, theta, phi):
    """Calculate transmittance for x- and y-polarization.

    For two objects described by the same T-matrix, calculate the transmittance of x-
    and y-polarized plane waves, when the objects are arranged in a lattice and have
    the given positions in their unit cell. The objects are rotated by theta and then
    plus and minus phi, respectively.

    Args:
        tm (treams.TMatrix): T-matrix.
        lat (Sequence): Lattice definition.
        positions (Sequence): Positions in the unit cell.
        theta (float): Rotation about the y-axis.
        phi (float): Rotation about the z-axis.

    Returns:
        Tuple
    """
    # Create lattice from definition
    lattice = treams.Lattice(lat)
```

```
30
31     # Build cluster T-matrix
32     cluster = treams.TMatrix.cluster(
33         [tm.rotate(phi, theta, 0), tm.rotate(-phi, theta, 0)], np.array(positions)
34     )
35     # Calculate the interaction in the lattice for normal incidence
36     kpar = [0, 0]
37     cluster = cluster.latticeinteraction.solve(lattice, kpar)
38     pwb = treams.PlaneWaveBasisByComp.default([0, 0])
39     pwx = treams.plane_wave(
40         kpar,
41         [1, 0, 0],
42         basis=pwb,
43         k0=tm.k0,
44         material=1,
45         poltype="parity",
46         modetype="up",
47     )
48     pwy = treams.plane_wave(
49         kpar,
50         [0, 1, 0],
51         basis=pwb,
52         k0=tm.k0,
53         material=1,
54         poltype="parity",
55         modetype="up",
56     )
57     # Calculate S-matrix from the array
58     sm = treams.SMatrices.from_array(cluster, pwb)
59     return sm.tr(pwx)[0], sm.tr(pwy)[0]
60
61
62 def wrapper(tm):
63     """Take a T-matrix and run a sweep over angles."""
64     theta = np.pi / 2
65     angles = np.linspace(0, np.pi / 2, 200)
66     lattice = [500, 1000]
67     positions = [[0, -250, 0], [0, 250, 0]]
68     return [run_xy(tm, lattice, positions, theta=theta, phi=phi) for phi in angles]
69
70
71 tms = treams.io.load_hdf5("ellipsoid.h5")
72 # We use joblib for parallel execution
73 res = joblib.Parallel(n_jobs=8)([joblib.delayed(wrapper)(tm) for tm in tms])
74
75 # Store in pickle file
76 with open("qbics.pickle", "w") as fobj:
77     pickle.dump(res, fobj)
```

The necessary T-matrix file in the HDF format is available in the supplementary materials. Furthermore, we also provide scripts and the necessary files to recalculate the T-matrix with JCMsuite or COMSOL in the supplementary materials. For JCMsuite, the necessary configuration files are used by calling `ellipsoid_jcm.py`, which generates the HDF5 file directly. For the COMSOL calculation, the java file needs to be compiled with `comsol compile ellipsoid_comsol.java` and then run with `comsol batch ellipsoid_comsol.class`. The resulting coefficients are then exported as a text file, that can be converted to the suitable T-matrix format with the script `ellipsoid_comsol.py`. The calculation with COMSOL returns the T-matrix in the helicity format, such that the listing above needs to be changed accordingly in lines 45 and 54.

References

[1] P.C. Waterman, Matrix formulation of electromagnetic scattering, Proc. IEEE 53 (8) (1965) 805–812, https://doi.org/10.1109/proc.1965.4058.

[2] N. Stefanou, V. Yannopapas, A. Modinos, MULTEM 2: a new version of the program for transmission and band-structure calculations of photonic crystals, Comput. Phys. Commun. 132 (1–2) (2000) 189–196, https://doi.org/10.1016/s0010-4655(00)00131-4.

[3] D.W. Mackowski, M.I. Mishchenko, A multiple sphere T-matrix Fortran code for use on parallel computer clusters, J. Quant. Spectrosc. Radiat. Transf. 112 (13) (2011) 2182–2192, https://doi.org/10.1016/j.jqsrt.2011.02.019.

[4] W.R.C. Somerville, B. Auguié, E.C. Le Ru, Smarties: user-friendly codes for fast and accurate calculations of light scattering by spheroids, J. Quant. Spectrosc. Radiat. Transf. 174 (2016) 39–55, https://doi.org/10.1016/j.jqsrt.2016.01.005.

[5] J. Markkanen, A.J. Yuffa, Fast superposition T-matrix solution for clusters with arbitrarily-shaped constituent particles, J. Quant. Spectrosc. Radiat. Transf. 189 (2017) 181–188, https://doi.org/10.1016/j.jqsrt.2016.11.004.

[6] A. Egel, Y. Eremin, T. Wriedt, D. Theobald, U. Lemmer, G. Gomard, Extending the applicability of the T-matrix method to light scattering by flat particles on a substrate via truncation of Sommerfeld integrals, J. Quant. Spectrosc. Radiat. Transf. 202 (2017) 279–285, https://doi.org/10.1016/j.jqsrt.2017.08.016.

[7] M. Nečada, P. Törmä, Multiple-scattering T-matrix simulations for nanophotonics: symmetries and periodic lattices, Commun. Comput. Phys. 30 (2) (2021) 357–395, https://doi.org/10.4208/cicp.OA-2020-0136, arXiv:2006.12968.

[8] D. Schebarchov, A. Fazel-Najafabadi, E.C. Le Ru, B. Auguié, Multiple scattering of light in nanoparticle assemblies: user guide for the terms program, J. Quant. Spectrosc. Radiat. Transf. 284 (2022) 108131, https://doi.org/10.1016/j.jqsrt.2022.108131.

[9] T. Wriedt, J. Hellmers, New scattering information portal for the light-scattering community, J. Quant. Spectrosc. Radiat. Transf. 109 (8) (2008) 1536–1542, https://doi.org/10.1016/j.jqsrt.2007.11.008.

[10] J. Hellmers, T. Wriedt, New approaches for a light scattering Internet information portal and categorization schemes for light scattering software, J. Quant. Spectrosc. Radiat. Transf. 110 (14) (2009) 1511–1517, https://doi.org/10.1016/j.jqsrt.2009.01.023.

[11] C.M. Linton, Lattice sums for the Helmholtz equation, SIAM Rev. 52 (4) (2010) 630–674, https://doi.org/10.1137/09075130X.

[12] E. Popov (Ed.), Gratings: Theory and Numeric Applications, second revisited edition, 2nd edition, Institut Fresnel, AMU, 2014.

[13] D. Beutel, I. Fernandez-Corbaton, C. Rockstuhl, Unified lattice sums accommodating multiple sublattices for solutions of the Helmholtz equation in two and three dimensions, Phys. Rev. A 107 (1) (2023) 013508, https://doi.org/10.1103/PhysRevA.107.013508.

[14] P.P. Ewald, Die Berechnung optischer und elektrostatischer Gitterpotentiale, Ann. Phys. 369 (3) (1921) 253–287, https://doi.org/10.1002/andp.19213690304.

[15] H. Huang, L. Tsang, A. Colliander, S.H. Yueh, Propagation of waves in randomly distributed cylinders using three-dimensional vector cylindrical wave expansions in Foldy–Lax equations, IEEE J. Multiscale Multiphys. Comput. Techn. 4 (2019) 214–226, https://doi.org/10.1109/JMMCT.2019.2948022.

[16] N. Stefanou, V. Yannopapas, A. Modinos, Heterostructures of photonic crystals: frequency bands and transmission coefficients, Comput. Phys. Commun. 113 (1) (1998) 49–77, https://doi.org/10.1016/s0010-4655(98)00060-5.

[17] M. Folk, G. Heber, Q. Koziol, E. Pourmal, D. Robinson, An overview of the HDF5 technology suite and its applications, in: Proceedings of the EDBT/ICDT 2011 Workshop on Array Databases, AD '11, Association for Computing Machinery, New York, NY, USA, 2011, pp. 36–47.

[18] R.N.S. Suryadharma, M. Fruhnert, I. Fernandez-Corbaton, C. Rockstuhl, Studying plasmonic resonance modes of hierarchical self-assembled meta-atoms based on their transfer matrix, Phys. Rev. B 96 (4) (2017) 045406, https://doi.org/10.1103/PhysRevB.96.045406.

[19] G. Kristensson, Scattering of Electromagnetic Waves by Obstacles, SciTech Publishing Inc, 2016.

[20] L. Silberstein, Elektromagnetische Grundgleichungen in bivektorieller Behandlung, Ann. Phys. 327 (3) (1907) 579–586, https://doi.org/10.1002/andp.19073270313.

[21] A. Lakhtakia, Beltrami Fields in Chiral Media, World Scientific, Singapore, River Edge, NJ, 1994.

[22] P.M. Morse, H. Feshbach, Methods of Theoretical Physics, McGraw-Hill, New York, 1953.

[23] C.F. Bohren, D.R. Huffman, Absorption and Scattering of Light by Small Particles, Wiley-VCH, 1998.

[24] A. Moroz, A recursive transfer-matrix solution for a dipole radiating inside and outside a stratified sphere, Ann. Phys. 315 (2) (2005) 352–418, https://doi.org/10.1016/j.aop.2004.07.002.

[25] Q.-C. Shang, Z.-S. Wu, T. Qu, Z.-J. Li, L. Bai, Scattering from a multilayered chiral sphere using an iterative method, J. Quant. Spectrosc. Radiat. Transf. 173 (2016) 72–82, https://doi.org/10.1016/j.jqsrt.2015.12.030.

[26] A. Doicu, T. Wriedt, Extended boundary condition method with multipole sources located in the complex plane, Opt. Commun. 139 (1) (1997) 85–91, https://doi.org/10.1016/S0030-4018(97)00113-2.

[27] D.W. Mackowski, Discrete dipole moment method for calculation of the T matrix for nonspherical particles, JOSA A 19 (5) (2002) 881–893, https://doi.org/10.1364/JOSAA.19.000881.

[28] M. Fruhnert, I. Fernandez-Corbaton, V. Yannopapas, C. Rockstuhl, Computing the T-matrix of a scattering object with multiple plane wave illuminations, Beilstein J. Nanotechnol. 8 (2017) 614–626, https://doi.org/10.3762/bjnano.8.66.

[29] G. Demésy, J.-C. Auger, B. Stout, Scattering matrix of arbitrarily shaped objects: combining finite elements and vector partial waves, JOSA A 35 (8) (2018) 1401–1409, https://doi.org/10.1364/JOSAA.35.001401.

[30] O.R. Cruzan, Translational addition theorems for spherical vector wave functions, Q. Appl. Math. 20 (1) (1962) 33–40, https://doi.org/10.1090/qam/132851.

[31] S. Stein, Addition theorems for spherical wave functions, Q. Appl. Math. 19 (1) (1961) 15–24, https://doi.org/10.1090/qam/120407.

[32] L. Tsang, J.A. Kong, R.T. Shin, Theory of Microwave Remote Sensing, Wiley Series in Remote Sensing and Image Processing, Wiley-Interscience, 1985.

[33] A. Moroz, Quasi-periodic Green's functions of the Helmholtz and Laplace equations, J. Phys. A, Math. Gen. 39 (36) (2006) 11247–11282, https://doi.org/10.1088/0305-4470/39/36/009.

[34] K. Kambe, Theory of low-energy electron diffraction (II. Cellular method for complex monolayers and multilayers), Z. Naturforsch. A 23 (9) (1968) 1280–1294, https://doi.org/10.1515/zna-1968-0908.

[35] V. Eyert, The Augmented Spherical Wave Method: A Comprehensive Treatment, second edition, Lecture Notes in Physics, vol. 849, Springer, Heidelberg, 2012.

[36] B. Peterson, S. Ström, T-matrix for electromagnetic scattering from an arbitrary number of scatterers and representations of E(3), Phys. Rev. D 8 (10) (1973) 3661–3678, https://doi.org/10.1103/PhysRevD.8.3661.

[37] J.B. Pendry, Low Energy Electron Diffraction: The Theory and Its Application to Determination of Surface Structure (Techniques of Physics), Academic Press, 1974.

[38] M.I. Mishchenko, L.D. Travis, D.W. Mackowski, T-matrix computations of light scattering by nonspherical particles: a review, J. Quant. Spectrosc. Radiat. Transf. 55 (5) (1996) 535–575, https://doi.org/10.1016/0022-4073(96)00002-7.

[39] F. Frezza, F. Mangini, N. Tedeschi, Introduction to electromagnetic scattering: tutorial, JOSA A 35 (1) (2018) 163–173, https://doi.org/10.1364/JOSAA.35.000163.

[40] G. Han, Y. Han, H. Zhang, Relations between cylindrical and spherical vector wavefunctions, J. Opt. A, Pure Appl. Opt. 10 (1) (2007) 015006, https://doi.org/10.1088/1464-4258/10/01/015006.

[41] R.C. Wittmann, Spherical wave operators and the translation formulas, IEEE Trans. Antennas Propag. 36 (8) (1988) 1078–1087, https://doi.org/10.1109/8.7220.

[42] H. Dachsel, Fast and accurate determination of the Wigner rotation matrices in the fast multipole method, J. Chem. Phys. 124 (14) (2006) 144115, https://doi.org/10.1063/1.2194548.

[43] C.R. Harris, K.J. Millman, S.J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N.J. Smith, R. Kern, M. Picus, S. Hoyer, M.H. van Kerkwijk, M. Brett, A. Haldane, J.F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, T.E. Oliphant, Array programming with NumPy, Nature 585 (7825) (2020) 357–362, https://doi.org/10.1038/s41586-020-2649-2.

[44] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S.J. van der Walt, M. Brett, J. Wilson, K.J. Millman, N. Mayorov, A.R.J. Nelson, E. Jones, R. Kern, E. Larson, C.J. Carey, İ. Polat, Y. Feng, E.W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E.A. Quintero, C.R. Harris, A.M. Archibald, A.H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0: fundamental algorithms for scientific computing in Python, Nat. Methods 17 (3) (2020) 261–272, https://doi.org/10.1038/s41592-019-0686-2.

[45] Y.-l. Xu, Efficient evaluation of vector translation coefficients in multiparticle light-scattering theories, J. Comput. Phys. 139 (1) (1998) 137–165, https://doi.org/10.1006/jcph.1997.5867.

[46] P.B. Johnson, R.W. Christy, Optical constants of the Noble metals, Phys. Rev. B 6 (12) (1972) 4370–4379, https://doi.org/10.1103/PhysRevB.6.4370.

[47] S. Burger, L. Zschiedrich, J. Pomplun, F. Schmidt, JCMsuite: an adaptive FEM solver for precise simulations in nano-optics, in: Integrated Photonics and Nanophotonics Research and Applications, Optical Society of America, 2008, Paper ITuE4, ITuE4, https://doi.org/10.1364/IPNRA.2008.ITuE4, 2008.

[48] J. Feis, D. Beutel, J. Köpfler, X. Garcia-Santiago, C. Rockstuhl, M. Wegener, I. Fernandez-Corbaton, Helicity-preserving optical cavity modes for enhanced sensing of chiral molecules, Phys. Rev. Lett. 124 (3) (2020) 033201, https://doi.org/10.1103/PhysRevLett.124.033201.

[49] P. Scott, X. Garcia-Santiago, D. Beutel, C. Rockstuhl, M. Wegener, I. Fernandez-Corbaton, On enhanced sensing of chiral molecules in optical cavities, Appl. Phys. Rev. 7 (4) (2020) 041413, https://doi.org/10.1063/5.0025006.

[50] D. Beutel, A. Groner, C. Rockstuhl, C. Rockstuhl, I. Fernandez-Corbaton, Efficient simulation of biperiodic, layered structures based on the T-matrix method, J. Opt. Soc. Am. A 38 (6) (2021) 1782–1791, https://doi.org/10.1364/JOSAB.419645.

[51] B. Zerulla, M. Krstić, D. Beutel, C. Holzer, C. Wöll, C. Rockstuhl, I. Fernandez-Corbaton, A multi-scale approach for modeling the optical response of molecular materials inside cavities, Adv. Mater. 34 (21) (2022) 2200350, https://doi.org/10.1002/adma.202200350.

[52] B. Zerulla, C. Li, D. Beutel, S. Oßwald, C. Holzer, J. Bürck, S. Bräse, C. Wöll, I. Fernandez-Corbaton, L. Heinke, C. Rockstuhl, M. Krstić, Exploring functional photonic devices made from a chiral metal–organic framework material by a multiscale computational method, Adv. Funct. Mater. (2023) 2301093, https://doi.org/10.1002/adfm.202301093.

[53] B. Zerulla, R. Venkitakrishnan, D. Beutel, M. Krstić, C. Holzer, C. Rockstuhl, I. Fernandez-Corbaton, A T-matrix based approach to homogenize artificial materials, Adv. Opt. Mater. 11 (3) (2023) 2201564, https://doi.org/10.1002/adom.202201564.

[54] D. Dams, D. Beutel, X. Garcia-Santiago, C. Rockstuhl, R. Alaee, Moiré flat bands in strongly coupled atomic arrays, Opt. Mater. Express 13 (7) (2023) 2003–2019, https://doi.org/10.1364/OME.486789.

[55] A. Rahimzadegan, T.D. Karamanos, R. Alaee, A.G. Lamprianidis, D. Beutel, R.W. Boyd, C. Rockstuhl, A comprehensive multipolar theory for periodic metasurfaces, Adv. Opt. Mater. (2022) 2102059, https://doi.org/10.1002/adom.202102059.