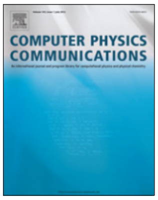
Computer Physics Communications 183 (2012) 2233–2244

Contents lists available at SciVerse ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc



S^4: A free electromagnetic solver for layered periodic structures[☆]

Victor Liu [*], Shanhui Fan

Ginzton Laboratory, Department of Electrical Engineering, Stanford University, Stanford, CA 94305, USA

ARTICLE INFO

Article history:
Received 29 November 2011
Received in revised form
15 March 2012
Accepted 25 April 2012
Available online 19 May 2012

Keywords:
Computational electromagnetics
Maxwell solver
Fourier modal method
Rigorous coupled wave analysis
Scattering matrix method

ABSTRACT

We describe S^4, a free implementation of the Fourier modal method (FMM), which has also been commonly referred to as rigorous coupled wave analysis (RCWA), for simulating electromagnetic propagation through 3D structures with 2D periodicity. We detail design aspects that allow S^4 to be a flexible platform for these types of simulations. In particular, we highlight the ability to select different FMM formulations, user scripting, and extensibility of program capabilities for eigenmode computations.

Program summary

Program title: S4
Catalogue identifier: AEMO_v1_0.
Program summary URL: http://cpc.cs.qub.ac.uk/summaries/AEMO_v1_0..html
Program obtainable from: CPC Program Library, Queen's University, Belfast, N. Ireland
Licensing provisions: GNU General Public License, version 2
No. of lines in distributed program, including test data, etc.: 56910
No. of bytes in distributed program, including test data, etc.: 433883
Distribution format:
Programming language: C, C++.
Computer: Any computer with a Unix-like environment and a C++ compiler. Developed on 2.3 GHz AMD Phenom 9600.
Operating system: Any Unix-like environment; developed under MinGW32 on Windows 7.
Has the code been vectorized or parallelized?: Yes. Parallelized using MPI.
RAM: Problem dependent (linearly proportional to number of layers and quadratic in number of Fourier components). A single layer calculation with approximately 100 Fourier components uses approximately 10 MB.
Classification: 10. Electrostatics and Electromagnetics.
External routines: Lua [1] and optionally exploits additional free software packages: FFTW [2], CHOLMOD [3], MPI message-passing interface [4], LAPACK and BLAS linear-algebra software [5], and Kiss FFT [6].
Nature of problem:
Time-harmonic electromagnetism in layered bi-periodic structures.
Solution method:
The Fourier modal method (rigorous coupled wave analysis) and the scattering matrix method.
Running time:
Problem dependent and highly dependent on quality of the BLAS implementation (linearly proportional to number of layers and cubic in number of Fourier components). A single layer calculation with approximately 100 Fourier components takes 4 s on the development machine using the reference BLAS.

References

[1] R. Ierusalimschy, L.H. de Figueiredo, W.C. Filho, Lua — an extensible extension language, Software: Practice and Experience 26 (1996) 635–652. http://www.lua.org.
[2] FFTW, http://www.fftw.org.
[3] Y. Chen, T.A. Davis, W.W. Hager, and S. Rajamanickam, Algorithm 887: CHOLMOD, supernodal sparse Cholesky factorization and update/downdate, ACM Trans. Math. Software, Vol. 35, No. 3, 2009. http://www.cise.ufl.edu/research/sparse/cholmod.

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

[*] Correspondence to: Ginzton Laboratory, Nano Building, Mail Code 4088, 348 Via Pueblo Mall, Stanford, CA 94305-4088, USA. Tel.: +1 408 482 6910.
E-mail addresses: vkl@stanford.edu, victorliu@alumni.stanford.edu (V. Liu), shanhui@stanford.edu (S. Fan).

0010-4655/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.cpc.2012.04.026

[4] T.M. Forum, MPI: A Message Passing Interface, in: Supercomputing 93, Portland, OR, 878883, 1993.
[5] LAPACK, http://www.netlib.org/lapack.
[6] Kiss FFT, http://kissfft.sourceforge.net.

1. Introduction

The study of layered photonic structures is of great practical importance due to planar fabrication technology. An important subset of layered structures is those with in-plane periodicity, representing a large class of interesting devices and systems from the standpoint of both engineering and fundamental physics. Fig. 1 shows an example of such a structure which is periodic in the *xy*-plane, and piecewise constant in the *z*-direction. We present here the Stanford Stratified Structure Solver (S^4), a frequency-domain computational electromagnetics tool to simulate the interaction of light with such structures.

There are a great variety of general-purpose tools available for computational electromagnetism (CEM), such as the finite-difference time-domain (FDTD) method [1], finite-difference frequency-domain (FDFD) method [2], and the finite element method (FEM) [3], that can be adapted to simulate three-dimensional structures with 2D periodicity. However, for the specific class of periodic structures composed of layers invariant in the direction normal to the periodicity, the Fourier modal method (FMM) [4], also called rigorous coupled wave analysis (RCWA) [5–9] or the scattering matrix method (SMM) [10–12], is particularly suitable due to its Fourier basis representation. S^4 is an implementation of the FMM for computing modal expansions within layers, combined with a scattering matrix (*S*-matrix) algorithm [10,13] to join together layers for solving electromagnetic fields throughout a three-dimensional structure.

The basic idea behind the FMM is to expand the electromagnetic fields within each layer into eigenmodes with an exponential dependence in the normal direction. In the case of the FMM, the fields and eigenmodes are represented using a Fourier basis in the plane of periodicity, with each Fourier component coupled via the dielectric distribution within a layer. The modal expansion coefficients are then related at layer interfaces to satisfy field continuity conditions in the Fourier basis. Alternatively, one may take the viewpoint that the scattering matrix of propagation through a layer is diagonalized in the modal basis, and so knowledge of the complete set of eigenmodes in each layer allows construction of the total scattering matrix by combining the scattering matrices of adjacent layers. We should note that the use of RCWA in the study of gratings is closely related to two other methods: the differential method [14], and the *C*-method [15]. Both of these methods are better suited for non-lamellar gratings or structures which are not inherently planar.

The earliest works using coupled wave analysis studied grating diffraction problems for holograms, mostly involving a sinusoidally varying dielectric function in a single layer [5,6,16]. These core principles were summarized by Moharam and Gaylord detailing the original RCWA formulation [7], and subsequently reformulated into a numerically stable method [9]. The extension to crossed gratings and general two-dimensionally periodic structures occurred later with improved computational power and numerical algorithms [8,10].

It was noted for some time that RCWA and related modal methods suffer from very slow convergence (with respect to the number of Fourier series terms used) when analyzing metallic gratings under TM polarization [17]. It was further noted that a simple reformulation of the dielectric coupling matrix could result in vastly superior convergence [18,19]. In 1996, Li published his seminal explanation for this drastic difference in convergence rate [20]. At the heart of the issue are the different boundary conditions for the normal and tangential electric field components at an interface, requiring different "Fourier factorization" rules. Li's reformulation of the FMM using proper Fourier factorization rules [4] resulted in improved convergence for staircase-discretized structures, but proved to be computationally demanding and inadequate for modeling large classes of structures with smooth dielectric inclusions. The recent work on the fast Fourier factorization (FFF) methods [21–23] has enabled more efficient application of the correct Fourier factorization rules, at the cost of requiring knowledge of a local polarization basis at all points within the unit cell. The FMM using FFF has been applied using custom tailored basis fields [24–26] as well as automatically generated fields [27] and has been extremely successful at substantially improving the accuracy of the original methods.

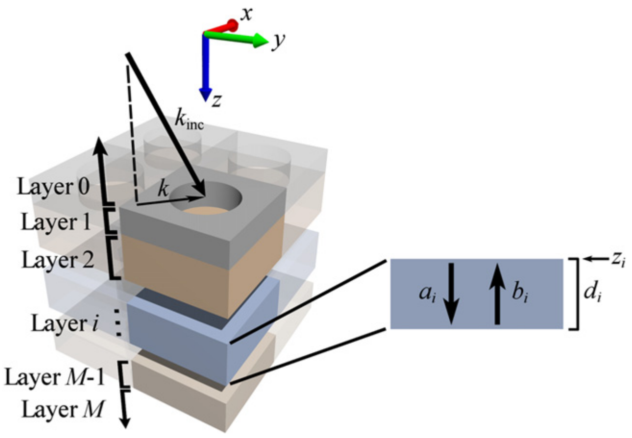


Fig. 1. Left: schematic of the problem formulation for a stack of layers. A single square unit cell is highlighted among the four shown. The incident radiation is assumed to be a plane wave with wavevector $\mathbf{k}_{\text{inc}}$, and its projection into the *xy*-plane is $\mathbf{k}$. Right: schematic for the cross-section of layer *i*. The arrows indicate the propagation directions of forward and backward layer modes.

Our development of S^4 includes all formulations known to the authors with the notable exception of recent work on using adaptive spatial resolution (ASR) [28–30]. Although ASR can be used for structures with significantly different patterning in different layers [31], its implementation is complicated and requires projection of fields onto a common basis at layer interfaces, which may cause a loss in accuracy and may also violate energy conservation. The use of ASR falls beyond the scope of the unified derivation we present in this paper and so we opted to keep S^4 a relatively simple and general FMM solver. The architecture of S^4 enables rapid experimentation and inclusion of new extensions to the FMM, and it would not be difficult to extend it with ASR techniques.

This paper is organized as follows: Section 2 describes the problem geometry and coordinates. Section 3 formulates the eigenmode problem for the plane wave Maxwell's equations within a single layer. Section 4 explains in more detail the various formulations mentioned above and summarizes the implementations in S^4. In Sections 5–7 the various solution capabilities of S^4 are described, as well as the user interface. In Section 8, we briefly comment on the convergence properties of the various formulations. In Section 9 we remark on the use of the FMM for metallic structures. Finally we summarize the capabilities of S^4 in Section 10.

2. Geometric definitions

To facilitate the discussion in this paper, the coordinate system is oriented such that the z-axis is normal to the layers of the structure, and the structure is assumed to be periodic in the xy-plane with primitive lattice vectors $\mathbf{l}_1$ and $\mathbf{l}_2$. We will lump the transverse coordinates in the xy-plane into a vector $\mathbf{r}$. Each layer is indexed by i with thickness d_i and extending from $z = z_i$ to $z = z_i + d_i$, with layer 1 extending from $z = z_1 = 0$ to $z = d_1$, layer 2 extending from $z = z_2 = d_1$ to $z = d_1 + d_2$, and so on. The infinite half-space $z < 0$ in front of the structure is denoted layer 0, and the infinite half-space behind the structure is denoted layer M. Each layer is assumed invariant in the z-direction; that is, in layer i, the spatial dielectric distribution $\epsilon(x, y, z)$ is constant for fixed x and y, and $z_i < z < z_i + d_i$.

To determine the reciprocal lattice of the Fourier domain, we first define a real space primitive lattice vector matrix whose columns are $\mathbf{l}_1$ and $\mathbf{l}_2$:

$$L_r = \begin{bmatrix} \mathbf{l}_1 & \mathbf{l}_2 \end{bmatrix} = \begin{bmatrix} l_{1x} & l_{2x} \\ l_{1y} & l_{2y} \end{bmatrix}. \tag{1}$$

Then the reciprocal lattice is defined by the columns of

$$L_k = 2\pi L_r^{-\mathsf{T}} \tag{2}$$

where $^{-\mathsf{T}}$ denotes the transpose of the inverse matrix.

Typical problems require solving for transmission, reflection, or absorption spectra of structures. In these cases, incident radiation from layer 0 is assumed to be a plane wave propagating in the positive z direction towards layer 1. The incident wavevector is $\mathbf{k}_{\text{inc}}$ with in-plane component $\mathbf{k}$. There are also instances in which an embedded source within the structure is required. In these instances, the source is a current source in the form of a spatial harmonic $\mathbf{j}e^{i\mathbf{k}\cdot\mathbf{r}}\delta(z - z_i)$ specified between layers $i - 1$ and i.

3. Maxwell's equations in a layer

3.1. Units and conventions

We will adopt a derivation and notation similar to those used in [10]. The starting point is Maxwell's equations in time-harmonic form, assuming an $\exp(-i\omega t)$ time dependence:

$$\nabla \times \mathbf{H} = -i\omega\epsilon_0\epsilon\mathbf{E} \qquad \nabla \times \mathbf{E} = i\omega\mu_0\mathbf{H}. \tag{3}$$

For simplicity we assumed that materials are linear and nonmagnetic. These assumptions are satisfied for most calculations for nanophotonics. From here on after, we will use Lorentz–Heaviside units, so that the speed of light c and vacuum impedance $Z_0 = \sqrt{\mu_0/\epsilon_0}$ are both unity (making c, μ_0, and ϵ_0 drop out). These units are effectively the same as starting with SI units and scaling $\sqrt{\mu_0\epsilon_0}\omega_{\text{SI}} \rightarrow \omega$ and $\sqrt{\mu_0/\epsilon_0}\mathbf{H}_{\text{SI}} \rightarrow \mathbf{H}$ and $\mathbf{E}_{\text{SI}} \rightarrow \mathbf{E}$. This change of units brings the electric and magnetic fields onto the same scale and the temporal and spatial frequency scales onto the same scale, providing better numerical conditioning in the implementation, and simplifying notation. In these new units, Maxwell's equations become

$$\nabla \times \mathbf{H} = -i\omega\epsilon\mathbf{E} \qquad \nabla \times \mathbf{E} = i\omega\mathbf{H}. \tag{4}$$

3.2. Fourier transforms

The next step is to take the spatial Fourier transform in the xy-plane. Because of the periodicity and separability of the z-axis, the fields must have the form

$$\mathbf{H}(\mathbf{r}, z) = \sum_{\mathbf{G}} \mathbf{H}_{\mathbf{G}}(z)e^{i(\mathbf{k}+\mathbf{G})\cdot\mathbf{r}} \tag{5}$$

where $\mathbf{k}$ is the in-plane component of the k-vector set by the excitation (for example, by the angle of incidence for an incident plane wave) and $\mathbf{G}$ is a reciprocal lattice vector ($L_k^{-1}\mathbf{G} \in \mathbb{Z}^2$). An analogous equation holds for $\mathbf{E}$. The truncation to a finite set of $\mathbf{G}$ for a simulation is, in principle, the only "discretization" required for the FMM. There have been numerous discussions concerning the best way to choose a set of $\mathbf{G}$ [4]. For general lattices, the best choice is to use "circular truncation" where all the $\mathbf{G}$ vectors within a circular region around the origin in reciprocal space are used. In S^4, the user only specifies an upper bound for the number of $\mathbf{G}$, and the circular region is chosen automatically. The exact number of reciprocal lattice vectors that is used is determined based on symmetry considerations (e.g. if $\mathbf{G}$ is included, then $-\mathbf{G}$ should also be included). Assuming that a fixed set of $\mathbf{G}$ has been chosen as well as an ordering (the same for all layers), denote by $\mathbf{h}(z)$ the vector $[\mathbf{H}_{\mathbf{G}_1}(z), \mathbf{H}_{\mathbf{G}_2}(z), \ldots]^{\mathsf{T}}$, and similarly for $\mathbf{e}(z)$.

The Fourier transform of the in-plane dielectric function is

$$\epsilon_{\mathbf{G}} = \frac{1}{|L_r|}\int_{\text{cell}} \epsilon(\mathbf{r})e^{-i\mathbf{G}\cdot\mathbf{r}}d\mathbf{r} \tag{6}$$

where the integral is over one unit cell of the lattice. In general ϵ can be a tensor [23,32], but we will assume the z-axis is separable for simplicity; i.e. that it is of the form

$$\epsilon = \begin{bmatrix} \epsilon_{xx} & \epsilon_{xy} & 0 \\ \epsilon_{yx} & \epsilon_{yy} & 0 \\ 0 & 0 & \epsilon_z \end{bmatrix}. \tag{7}$$

In this case, each component can be Fourier transformed separately, and we obtain five sets of coefficients: $\hat{\epsilon}_{\mathbf{G},xx}$, $\hat{\epsilon}_{\mathbf{G},xy}$, $\hat{\epsilon}_{\mathbf{G},yx}$, $\hat{\epsilon}_{\mathbf{G},yy}$, and $\hat{\epsilon}_{\mathbf{G},z}$. Using the same ordering of $\mathbf{G}$ as for $\mathbf{h}(z)$, we can form the block Toeplitz matrix $\hat{\epsilon}_{xx}$ whose (m, n)-th element is defined by

$$\hat{\epsilon}_{xx,mn} = \epsilon_{(\mathbf{G}_m-\mathbf{G}_n),xx}. \tag{8}$$

That is, the (m, n) entry of $\hat{\epsilon}_{xx}$ is the Fourier coefficient corresponding to the reciprocal lattice vector $\mathbf{G}_m - \mathbf{G}_n$. The matrices $\hat{\epsilon}_{xy}$, $\hat{\epsilon}_{yx}$, $\hat{\epsilon}_{yy}$, and $\hat{\epsilon}_z$ are defined analogously. Generally, we will use the hat symbol ($\hat{}$) to refer to square matrix operators acting in $\mathbf{G}$ space.

Using these definitions, we can Fourier transform Maxwell's equations for each field component:

$$i\hat{k}_y h_z(z) - h_y'(z) = -i\omega d_x(z) \tag{9}$$

$$h_x'(z) - i\hat{k}_x h_z(z) = -i\omega d_y(z) \tag{10}$$

$$i\hat{k}_x h_y(z) - i\hat{k}_y h_x(z) = -i\omega\hat{\epsilon}_z e_z(z) \tag{11}$$

$$i\hat{k}_y e_z(z) - e_y'(z) = i\omega h_x(z) \tag{12}$$

$$e_x'(z) - i\hat{k}_x e_z(z) = i\omega h_y(z) \tag{13}$$

$$i\hat{k}_x e_y(z) - i\hat{k}_y e_x(z) = i\omega h_z(z) \tag{14}$$

where primes denote differentiation with respect to z, and $\hat{k}_x$ is a diagonal matrix with entries $(k_x + G_{1x}, k_x + G_{2x}, \ldots)$ and analogously for $\hat{k}_y$. The last two equations contain d_x and d_y which are the Fourier coefficients of the displacement field $\mathbf{D}$. To obtain a closed set of equations, we will need to relate the displacement field d_x and d_y, to the electric field e_x and e_y. This turns out to be subtle due to the need to apply the proper Fourier factorization rules, taking into account discontinuities in both ϵ and E. For now we assume that there exists a matrix $\mathcal{E}$ such that

$$\begin{bmatrix} -d_y(z) \\ d_x(z) \end{bmatrix} = \mathcal{E}\begin{bmatrix} -e_y(z) \\ e_x(z) \end{bmatrix}. \tag{15}$$

As we will see in Section 4, different formulations compute $\mathcal{E}$ in different ways, but they all take on the above form.

Eliminating the z components using Eq. (11), Eqs. (12) and (13) become

$$-\hat{k}_y\hat{\epsilon}_z^{-1}\hat{k}_x h_y(z) + \hat{k}_y\hat{\epsilon}_z^{-1}\hat{k}_y h_x(z) + i\omega e'_y(z) = \omega^2 h_x(z) \tag{16}$$

$$-i\omega e'_x(z) + \hat{k}_x\hat{\epsilon}_z^{-1}\hat{k}_x h_y(z) - \hat{k}_x\epsilon_z^{-1}\hat{k}_y h_x(z) = \omega^2 h_y(z) \tag{17}$$

or in matrix form

$$(\omega^2 I - \mathcal{K})\begin{bmatrix} h_x(z) \\ h_y(z) \end{bmatrix} = -i\omega \begin{bmatrix} -e'_y(z) \\ e'_x(z) \end{bmatrix} \tag{18}$$

$$\mathcal{K} = \begin{bmatrix} \hat{k}_y\hat{\epsilon}_z^{-1}\hat{k}_y & -\hat{k}_y\epsilon_z^{-1}\hat{k}_x \\ -\hat{k}_x\hat{\epsilon}_z^{-1}\hat{k}_y & \hat{k}_x\epsilon_z^{-1}\hat{k}_x \end{bmatrix} \tag{19}$$

where I is the identity matrix of the proper dimensions. Similarly, eliminating the z components using Eq. (14), Eqs. (9) and (10) become

$$i\omega h'_x(z) + \hat{k}_x\hat{k}_x e_y(z) - \hat{k}_x\hat{k}_y e_x(z) = \omega^2 d_y(z) \tag{20}$$

$$-\hat{k}_y\hat{k}_x e_y(z) + \hat{k}_y\hat{k}_y e_x(z) - i\omega h'_y(z) = \omega^2 d_x(z) \tag{21}$$

which can be written as

$$\left(\omega^2\mathcal{E} - K\right)\begin{bmatrix} -e_y(z) \\ e_x(z) \end{bmatrix} = -i\omega \begin{bmatrix} h'_x(z) \\ h'_y(z) \end{bmatrix} \tag{22}$$

$$K = \begin{bmatrix} \hat{k}_x\hat{k}_x & \hat{k}_x\hat{k}_y \\ \hat{k}_y\hat{k}_x & \hat{k}_y\hat{k}_y \end{bmatrix}. \tag{23}$$

Therefore the original Maxwell's equations are reduced to Eqs. (18) and (22).

3.3. Layer eigenmodes

The basic idea behind the FMM is to expand the fields within a layer into eigenmodes which have a simple $\exp(iqz)$ dependence for some complex number q. We assume the form of the magnetic field eigenmode is

$$\mathbf{H}(z) = \sum_{\mathbf{G}}\Bigg[\phi_{\mathbf{G},x}\mathbf{x} + \phi_{\mathbf{G},y}\mathbf{y} - \frac{(k_x + G_x)\phi_{\mathbf{G},x} + (k_y + G_y)\phi_{\mathbf{G},y}}{q}\mathbf{z}\Bigg] e^{i(\mathbf{k}+\mathbf{G})\cdot\mathbf{r}+iqz} \tag{24}$$

where $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ are the Cartesian unit vectors and $\phi_{\mathbf{G},x}$ and $\phi_{\mathbf{G},y}$ are expansion coefficients. These expansion coefficients may be written as vectors $\phi_x = [\phi_{\mathbf{G}_1,x}, \phi_{\mathbf{G}_2,x}, \ldots]^{\mathsf{T}}$, and analogously for ϕ_y. We then have

$$h(z) = \left[\phi_x\mathbf{x} + \phi_y\mathbf{y} - q^{-1}(\hat{k}_x\phi_x + \hat{k}_y\phi_y)\mathbf{z}\right] e^{iqz} \tag{25}$$

where $h(z)$ is a column vector whose elements correspond to $\mathbf{G}$ vectors. With this, and using Eqs. (18) and (22), Maxwell's equations above become

$$(\omega^2 I - \mathcal{K})\begin{bmatrix} \phi_x \\ \phi_y \end{bmatrix} = \omega q \begin{bmatrix} -e_y \\ e_x \end{bmatrix} \tag{26}$$

$$\omega q \begin{bmatrix} \phi_x \\ \phi_y \end{bmatrix} = \left(\omega^2\mathcal{E} - K\right)\begin{bmatrix} -e_y \\ e_x \end{bmatrix} \tag{27}$$

where we have dropped the z dependence on e_x and e_y to represent a fixed mode with $\exp(iqz)$ variation. Eliminating the electric field and using the fact that $\mathcal{K}K = 0$,

$$\left(\mathcal{E}(\omega^2 - \mathcal{K}) - K\right)\Phi = \Phi \mathrm{q}^2 \qquad \Phi = \begin{bmatrix} \phi_x \\ \phi_y \end{bmatrix} \tag{28}$$

where q^2 is the diagonal matrix whose diagonal elements are the eigenvalues q_n^2. The columns of the square matrix Φ are $[\phi_{x,n}, \phi_{y,n}]^{\mathsf{T}}$, the Fourier coefficients of the eigenmodes.

The solution of the eigenvalue equation forms the bulk of the computing time and is solved using Lapack [33]. One can see directly that the size of the eigenvalue problem scales as N, the number of $\mathbf{G}$. Thus the storage requirements for an entire simulation scale as $O\left(MN^2\right)$ where M is the number of layers. This can be reduced if certain intermediate layer-dependent matrices are not needed for output computations, making the required storage space proportional to the number of layers for which fields must be computed [13]. However, S^4 does not make this optimization in order to permit a more flexible programming interface. The solution to the eigenvalue problems dominates the computation time, and using standard variants of the QR algorithm requires on the order of $O(N^3)$ operations [34]. Thus the total run time of a single simulation scales as $O\left(MN^3\right)$.

3.4. Field recovery

Using the eigenmode basis determined by solving Eq. (28), the transverse magnetic field in layer i may be represented as

$$\begin{bmatrix} h_x(z) \\ h_y(z) \end{bmatrix} = \sum_n \begin{bmatrix} \phi_{x,n} \\ \phi_{y,n} \end{bmatrix}\left(a_n e^{iq_n z} + b_n e^{iq_n(d_i - z)}\right) \tag{29}$$

where n indexes the eigenmodes, a_n is the coefficient of a forward propagating wave (towards positive z) at $z = z_i$, and b_n is the coefficient of a backward propagating wave at $z = z_i + d_i$. For $q = \pm\sqrt{q^2}$ there are two choices depending on the sign chosen. For numerical stability, the sign is chosen such that $\operatorname{Im} q \geq 0$ so that the coefficients are the maximum amplitudes within a layer. Let us now define a diagonal matrix operator $f(z)$ with entries

$$f(z)_{nn} = e^{iq_n z} \tag{30}$$

which represents the modal phase accumulation operator. Let us also define transverse field component vectors in the Fourier basis:

$$h_t(z) = [h_x(z), h_y(z)]^{\mathsf{T}} \tag{31}$$

$$e_t(z) = [-e_y(z), e_x(z)]^{\mathsf{T}}, \tag{32}$$

as well as mode amplitude vectors for forward and backward waves:

$$a = [a_1, a_2, \ldots]^{\mathsf{T}} \tag{33}$$

$$b = [b_1, b_2, \ldots]^{\mathsf{T}}. \tag{34}$$

With these definitions, we may relate the mode amplitudes to the physical fields:

$$\begin{bmatrix} e_t(z) \\ h_t(z) \end{bmatrix} = \begin{bmatrix} \left(\omega^2 I - \mathcal{K}\right)\Phi\mathrm{q}^{-1} & -\left(\omega^2 I - \mathcal{K}\right)\Phi\mathrm{q}^{-1} \\ \Phi & \Phi \end{bmatrix} \times \begin{bmatrix} f(z)a \\ f(d - z)b \end{bmatrix}. \tag{35}$$

Note that the mode decomposition in Eq. (35) does not rely on any orthogonality relations; to decompose fields into mode amplitudes, the inverse of the matrix in Eq. (35) must be used. The z components of the fields may be recovered using Eqs. (11) and (14).

The next stages of solution involve forming the layer S-matrix, and constructing the total S-matrix from the S-matrix of each layer. The details of each of these steps will not be repeated here since our implementation follows [10] quite closely. One particular optimization to note is that often many layers have identical dielectric profiles. S^4 allows the user to duplicate layers to avoid redundant modal calculations.

4. FMM formulations

The preceding section related the Fourier components of the displacement field d_x and d_y to those of the electric field e_x and e_y via a matrix $\mathcal{E}$ without describing how it is computed. In fact, all the various reformulations of RCWA and the FMM mentioned in the introduction simply reduce to different definitions of $\mathcal{E}$.

There are a number of reasons to implement a variety of FMM formulations in a single package. Having a variety of formulations is beneficial because certain formulations may be more applicable or lead to better results for certain kinds of problems. Here we provide an overview and a final summary of the various formulations of the FMM that are implemented in S^4.

4.1. Closed-form Fourier transform

For many problems of interest, the dielectric constant profile in each layer can be specified in a non-discretized way using simple geometric shapes, within which the dielectric constant is assumed constant. Almost all engineered structures fall into this class of piecewise-constant dielectric functions, and other structures may be well approximated using piecewise-constant functions. The important idea here is that the shapes are specified exactly, and not initially discretized into pixels (they are specified by "vector" paths instead of "bitmaps"). For later use, we define the indicator function of a shape to have value 1 inside the shape and 0 otherwise. Each of the allowed shapes admits a closed form Fourier transform of their respective indicator functions: rectangle, circle, ellipse, and any simple closed (not necessarily convex) polygon [35].

Although only a few shapes can be Fourier transformed using efficient closed form equations, they admit a great deal of flexibility in specifying most structures of interest, and the inclusion of polygons allows the approximation of arbitrary structures to any desired precision. Note that in this model of specifying the dielectric constant, shapes cannot intersect each other. Shapes must either be completely contained within another shape, or their interiors must be disjoint. With these considerations, we formally express the dielectric constant as

$$\epsilon(\mathbf{r}) = \epsilon_b + \sum_j \left(\epsilon_j - \epsilon_{\supset j}\right) I_j(\mathbf{r}) \tag{36}$$

where the sum is over all shapes, ϵ_b is the background dielectric constant (in the absence of any patterning), ϵ_j is the dielectric constant within the j-th shape, and I_j is the indicator function for the j-th shape. As shown in Fig. 2, $\epsilon_{\supset j}$ is the dielectric constant in the shape that immediately contains shape j. The Fourier transform of Eq. (36) then becomes a simple sum over the scaled Fourier transforms of the indicator functions.

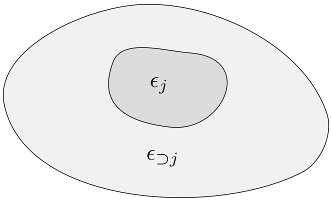


Fig. 2. Illustration of the shape containment constraint when specifying dielectric constants within a layer. Shapes must be entirely contained within another shape and cannot intersect.

The most basic FMM formulation in S^4 uses

$$\mathcal{E} = \begin{bmatrix} \hat{\epsilon}_{xx} & \hat{\epsilon}_{xy} \\ \hat{\epsilon}_{yx} & \hat{\epsilon}_{yy} \end{bmatrix} \tag{37}$$

where $\hat{\epsilon}$ is calculated using Eq. (8) using closed form transforms. This formulation is identical to the classical formulations [7,10] in the scalar case, and extends the birefringent model in [11]. Note that this formulation does not use proper Fourier factorization rules, as will be discussed in Section 4.4, and can suffer poor convergence (with respect to the number of $\mathbf{G}$ used) if the solution electric field has significant components normal to in-plane material interfaces.

4.2. Fast Fourier transform

A simple alternative to using closed form Fourier transforms as described in the previous section is to discretize the dielectric constant onto a grid and apply the fast Fourier transform (FFT) to approximate the Fourier series coefficients of ϵ. With the availability of closed form Fourier transforms, there is usually no advantage to this approach since it is inferior in both accuracy and speed if simple staircasing is used on material boundaries. However, this method is useful for structure optimization since it is often simpler to parameterize the dielectric function on a grid rather than as a composition of a set of shape outlines. When the geometry is specified using shapes, the discretization method in S^4 calculates exact pixel overlap areas with each shape to avoid staircasing effects.

4.3. Subpixel anisotropic averaging

Closely related to the FFT discretization is the method using subpixel averaging of the dielectric function within each discretization pixel to compute an anisotropic permittivity tensor. This technique was originally derived from the perturbation theory of dielectric interfaces and has been applied with great success to discretized electromagnetic simulations [36]. Subpixel averaging is widely used in the time-domain code Meep [37] as well as the mode-solver code MPB [38]. We have implemented the subpixel anisotropic averaging in 2D for each layer. The Fourier series coefficients of the in-plane dielectric function of each layer are then obtained using a fast Fourier transform. To our knowledge, the application of subpixel averaging to the FMM has not been reported in the literature and we will discuss the convergence behavior of this method in Section 8.

4.4. Vector-field-based formulations

The Fourier factorization rules presented by Li in [20] dictate that the Fourier coefficients of the displacement and electric field components parallel to a material interface should be related by Laurent's rule:

$$d_\parallel = \hat{\epsilon}\, e_\parallel \tag{38}$$

where $\hat{\epsilon}$ is the matrix of Fourier components of ϵ given by Eq. (8) and repeated below in Eq. (41). In contrast, the components normal to an interface should be related by the inverse rule:

$$e_\perp = \hat{\eta}\, d_\perp \tag{39}$$

where $\hat{\eta}$ is defined similarly to $\hat{\epsilon}$ except that in Eq. (6), the spatial distribution of $\eta = \epsilon^{-1}$ is Fourier transformed.

In the reformulation of the FMM by Li [4], the expression obtained for $\mathcal{E}$ using the proper Fourier factorization rules involved multiple 1D Fourier transforms of inverses of Toeplitz matrices, making its implementation both difficult and time-consuming. Inherent to his formulation is an effective staircasing effect that arises from taking 1D transforms in the principal lattice directions. Methods were soon developed that alleviate these difficulties by using a normal vector field defined over the entire unit cell, dubbed fast Fourier factorization (FFF) [23]. We briefly outline the details of computing $\mathcal{E}$ under these approaches since S^4 implements three different FFF methods.

We will assume scalar ϵ for this section for simplicity (for tensorial ϵ, S^4 currently does not perform proper Fourier factorization). Original formulations of the FMM did not perform the proper Fourier factorization in computing d_x and d_y. Most formulations [7,10] used Laurent's rule directly:

$$\begin{bmatrix} -d_y \\ d_x \end{bmatrix} = \begin{bmatrix} \hat{\epsilon} & \\ & \hat{\epsilon} \end{bmatrix} \begin{bmatrix} -e_y \\ e_x \end{bmatrix}. \tag{40}$$

Recall that $\hat{\epsilon}$ is defined as the Fourier convolution matrix with matrix elements defined by

$$\hat{\epsilon}_{\mathbf{GG}'} = \frac{1}{|L_r|} \int_{\text{cell}} \epsilon(\mathbf{r}) e^{-i(\mathbf{G}-\mathbf{G}')\cdot\mathbf{r}} d\mathbf{r}. \tag{41}$$

We will adopt this meaning for the hat (ˆ) symbol for the remainder of this section.

In order to efficiently simulate 1D metallic gratings in TM polarization, Refs. [18,19] used the inverse rule instead:

$$e_x = \widehat{\epsilon^{-1}} d_x \tag{42}$$

where $\widehat{\epsilon^{-1}}$ denotes the Fourier transform of ϵ^{-1} (that is, first taking the pointwise inverse in real space, then taking the Fourier transform). The correct application of these rules to a structure with 1D periodicity along the x direction would require

$$\begin{bmatrix} -d_y \\ d_x \end{bmatrix} = \begin{bmatrix} \hat{\epsilon} & \\ & \left(\widehat{\epsilon^{-1}}\right)^{-1} \end{bmatrix} \begin{bmatrix} -e_y \\ e_x \end{bmatrix}. \tag{43}$$

It is straightforward to implement the proper rules for structures with only 1D periodicity, but this fails for general 2D periodicity since the two spatial dimensions are no longer separable.

A proper Fourier factorization requires decomposing the **E** field into components normal and tangential to material interfaces. Let us assume that we have available a smooth (possibly complex) vector field $\mathbf{t} = [t_x, t_y]^{\mathrm{T}}$ which is periodic and tangent to all material interfaces within the unit cell. As in Ref. [23], we can then perform a coordinate transformation from x and y components of the fields into a coordinate system locally defined by the **t** vector. At material interfaces, the tangential and normal components are given by

$$\begin{bmatrix} E_t \\ E_n \end{bmatrix} = \begin{bmatrix} t_x & -t_y^* \\ t_y & t_x^* \end{bmatrix}^{-1} \begin{bmatrix} E_x \\ E_y \end{bmatrix} \tag{44}$$

where * denotes complex conjugation (so this applies to a general complex **t** field). Thus, to relate the D and E fields in real space,

$$\begin{bmatrix} -D_y \\ D_x \end{bmatrix} = \begin{bmatrix} t_y & t_x^* \\ -t_x & t_y^* \end{bmatrix} \begin{bmatrix} \epsilon & \\ & \eta^{-1} \end{bmatrix} \begin{bmatrix} t_y & t_x^* \\ -t_x & t_y^* \end{bmatrix}^{-1} \begin{bmatrix} -E_y \\ E_x \end{bmatrix} \tag{45}$$

$$= T \begin{bmatrix} \epsilon & \\ & \eta^{-1} \end{bmatrix} T^{-1} \begin{bmatrix} -E_y \\ E_x \end{bmatrix} \tag{46}$$

where in real space $\epsilon = \eta^{-1}$. This distinction is important when applying the inverse rule when Fourier transforming these equations.

There are many ways to implement Eq. (46) in Fourier space and we outline two below. As the first example, one can define

$$\mathcal{E}_1 = \widehat{T} \begin{bmatrix} \hat{\epsilon} & \\ & \hat{\eta}^{-1} \end{bmatrix} \widehat{T^{-1}} \tag{47}$$

where $\widehat{T^{-1}}$ indicates the inverse should be taken in the spatial domain and then Fourier transformed (it is understood that the Fourier transforms for $\widehat{T}$ and $\widehat{T^{-1}}$ are applied separately to each of the four blocks of these matrices). Note that here $\hat{\eta}^{-1}$ refers to the inverse of the matrix of Fourier coefficients of the ϵ^{-1} distribution, whereas $\hat{\epsilon}$ is composed of Fourier coefficients of the ϵ distribution.

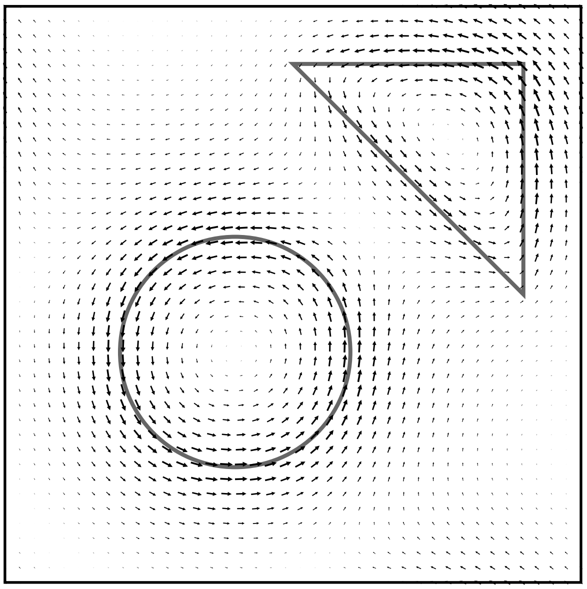


Fig. 3. An example **t** vector field generated automatically for a square unit cell with two inclusions: one circle and one triangle. Note the excellent field alignment with the boundaries of the inclusions.

Alternatively, we can wait to take the Fourier transform and instead simplify Eq. (45) to

$$\begin{bmatrix} -D_y \\ D_x \end{bmatrix} = \frac{1}{|t_x|^2 + |t_y|^2} \begin{bmatrix} |t_x|^2 \epsilon + |t_y|^2 \eta^{-1} & t_x^* t_y \left(\eta^{-1} - \epsilon\right) \\ t_x t_y^* \left(\eta^{-1} - \epsilon\right) & |t_x|^2 \eta^{-1} + |t_y|^2 \epsilon \end{bmatrix} \times \begin{bmatrix} -E_y \\ E_x \end{bmatrix}. \tag{48}$$

Denoting $\Delta = \epsilon - \eta^{-1}$,

$$\begin{bmatrix} -D_y \\ D_x \end{bmatrix} = \left\{ \begin{bmatrix} \epsilon & \\ & \epsilon \end{bmatrix} - \frac{1}{|t_x|^2 + |t_y|^2} \begin{bmatrix} \Delta & \\ & \Delta \end{bmatrix} \begin{bmatrix} |t_y|^2 & t_x^* t_y \\ t_x t_y^* & |t_x|^2 \end{bmatrix} \right\} \times \begin{bmatrix} -E_y \\ E_x \end{bmatrix} \tag{49}$$

$$= \left\{ \begin{bmatrix} \epsilon & \\ & \epsilon \end{bmatrix} - \begin{bmatrix} \Delta & \\ & \Delta \end{bmatrix} \begin{bmatrix} P_{yy} & P_{yx} \\ P_{xy} & P_{xx} \end{bmatrix} \right\} \begin{bmatrix} -E_y \\ E_x \end{bmatrix}. \tag{50}$$

Fourier transforming, we obtain

$$\mathcal{E}_2 = \begin{bmatrix} \hat{\epsilon} & \\ & \hat{\epsilon} \end{bmatrix} - \begin{bmatrix} \hat{\Delta} & \\ & \hat{\Delta} \end{bmatrix} \begin{bmatrix} \widehat{P_{yy}} & \widehat{P_{yx}} \\ \widehat{P_{xy}} & \widehat{P_{xx}} \end{bmatrix}, \tag{51}$$

which can be seen as a correction upon the naive application of Laurent's rule.

Both $\mathcal{E}_1$ and $\mathcal{E}_2$ have been used in the literature. In implementing either approach, one needs to generate a **t** field. A real **t** field that is smooth and locally tangential to the edge of closed curves must have a null somewhere in the interior. As we can see from Fig. 3, for a circular inclusion, for example, the field vanishes near the center of the circle. Thus there is an additional complication in choosing the **t** field because the denominator of Eq. (48) vanishes. Below we summarize various approaches that we have implemented in S^4.

4.4.1. Vector field generation

The $\mathbf{t}$ field generation in S^4 is fully automatic. There have been previous studies employing automatic vector field generation using electrostatic potentials or conformal mapping [25] and numerical gradient interpolation [27]. The conformal mapping and gradient interpolation techniques can produce discontinuities and tend to generate vector fields of low quality. In contrast, S^4 uses a very simple and rigorous approach by minimizing an energy functional while enforcing tangent vector constraints in a least squares sense [39]. This approach is capable of generating high quality conforming fields for very complex patterns. Fig. 3 shows an example of a vector field generated completely automatically using this method. Note that these fields are generated on a mesh over the unit cell and the discretely sampled field is Fourier transformed using an FFT. All the vector fields that are required by the different formulations in S^4 are generated by this method.

4.5. Summary of implemented algorithms

S^4 implements six different FMM formulations, which are listed below with shorthand names that will be used in later sections.

1. *Closed*: Eq. (37) is applied directly to the closed-form Fourier transforms of the layer shape pattern. This corresponds to the earliest FMM formulations.
2. *FFT*: The Fourier transform of the layer shape pattern is computed with an FFT and Eq. (37) is used to compute the $\mathcal{E}$ matrix.
3. *Subpixel*: The anisotropic average of the dielectric function is first computed over a discretized unit cell, and then the FFT is used to apply Eq. (37).
4. *Normal*: The vector field $\mathbf{t}$ is generated on the discretized unit cell using the method in Ref. [39], and normalized to unit length at each discretization point. Eq. (51) is applied to the FFT of the layer shape pattern sampled on the same discretization grid. This formulation corresponds to Ref. [25]. At locations where the $\mathbf{t}$ field vanishes, the normalization is undefined, leading to discontinuities in the field and so the components of $\hat{P}$ in Eq. (51) suffer from slow convergence.
5. *Jones*: A tangential vector field $\mathbf{t}$ is generated on the discretized unit cell and used to compute a Jones vector field $\mathbf{J}$ using the formula:
$$\mathbf{J} = \frac{e^{i\theta}}{|\mathbf{t}|}\begin{bmatrix}\mathbf{t} & \mathbf{t}^{\perp}\end{bmatrix}\begin{bmatrix}\cos\varphi & i\sin\varphi\end{bmatrix} \tag{52}$$
where $\mathbf{t}$ is uniformly scaled to have unit maximum length, $\mathbf{t}^{\perp} = [-t_y, t_x]^{\mathsf{T}}$, $\theta = \angle\mathbf{t}$ (the angle of the vector with respect to the x-axis), and $\varphi = \frac{\pi}{8}(1+\cos\pi\,|\mathbf{t}|)$ is a scalar field defined over the entire unit cell. Eq. (47) is then applied to the FFT of the layer shape pattern using $\mathbf{J}$ instead of $\mathbf{t}$. This is a generalization of the method described in Ref. [26].
6. *Pol*: The vector field $\mathbf{t}$ is generated on the discretized unit cell and uniformly scaled so the maximum length is unity. Eq. (51) is applied to the FFT of the layer shape pattern sampled on the same discretization grid. This method is original to S^4.

5. Output computations

The S-matrix algorithm allows calculation of mode amplitudes within any layer at any z-offset within a layer. To obtain physically meaningful fields, Eq. (35) is used to recover the four transverse field components. This amounts to performing an inverse Fourier transform on the transverse field components. Eqs. (11) and (14) can then be used to recover the normal components.

While in principle the fields at an arbitrary point in space can be calculated this way, and quantities such as the power flux can be numerically integrated pointwise, for certain common computations it is vastly more efficient to perform the calculations in Fourier space.

5.1. Power flux

By far the most commonly used program output is the power flux through a plane parallel to the layers. This is used for transmission, reflection, and absorption calculations, as well as diffraction if it is computed separately for each $\mathbf{G}$ component. The time averaged Poynting vector is defined by

$$\mathbf{S} = \frac{1}{2}\mathrm{Re}\left(\mathbf{E}\times\mathbf{H}^*\right). \tag{53}$$

Converting the right hand side to the Fourier basis, the z-component of the time-averaged power flux through a unit cell is

$$2S_z = \begin{bmatrix} e_t(z) \\ h_t(z) \end{bmatrix}^{\dagger}\begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix}\begin{bmatrix} e_t(z) \\ h_t(z) \end{bmatrix} \tag{54}$$

where e_t and h_t are defined in Eqs. (31) and (32) and the dagger superscript (†) indicates the conjugate transpose. We understand that the fields are the transverse fields and at the same z-coordinate and we suppress it in the notation. We now decompose this into forward and backward components:

$$2S_z^{\mathrm{forw}} = \begin{bmatrix} e \\ h \end{bmatrix}_{\mathrm{total}}^{\dagger}\begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix}\begin{bmatrix} e \\ h \end{bmatrix}_{\mathrm{forw}} \tag{55}$$

$$2S_z^{\mathrm{back}} = \begin{bmatrix} e \\ h \end{bmatrix}_{\mathrm{total}}^{\dagger}\begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix}\begin{bmatrix} e \\ h \end{bmatrix}_{\mathrm{back}} \tag{56}$$

where the forward and backward components of e and h are obtained by using only a or b, respectively, in Eq. (35) and setting the other to zero, whereas the total subscript means both a and b are generally nonzero. Expanding,

$$2S_z^{\mathrm{forw}} = e_{\mathrm{total}}^{\dagger}h_{\mathrm{forw}} + h_{\mathrm{total}}^{\dagger}e_{\mathrm{forw}} \tag{57}$$

$$2S_z^{\mathrm{back}} = e_{\mathrm{total}}^{\dagger}h_{\mathrm{back}} + h_{\mathrm{total}}^{\dagger}e_{\mathrm{back}}. \tag{58}$$

Using the block elements of the matrix in Eq. (35),

$$\begin{aligned} 2S_z^{\mathrm{forw}} &= \left[B\Phi(\omega \mathrm{q})^{-1}(a-b)\right]^{\dagger}[\Phi a] \\ &\quad + [\Phi(a+b)]^{\dagger}\left[B\Phi(\omega \mathrm{q})^{-1}a\right] \end{aligned} \tag{59}$$

$$\begin{aligned} 2S_z^{\mathrm{back}} &= \left[B\Phi(\omega \mathrm{q})^{-1}(a-b)\right]^{\dagger}[\Phi b] \\ &\quad + [\Phi(a+b)]^{\dagger}\left[-B\Phi(\omega \mathrm{q})^{-1}b\right] \end{aligned} \tag{60}$$

where $B = \omega^2 I - \mathcal{K}$. Letting $A = B\Phi(\omega\mathrm{q})^{-1}$ and simplifying,

$$2S_z^{\mathrm{forw}} = \left(a^{\dagger}A^{\dagger}\Phi a + a^{\dagger}\Phi^{\dagger}Aa\right) + \left(b^{\dagger}\Phi^{\dagger}Aa - b^{\dagger}A^{\dagger}\Phi a\right) \tag{61}$$

$$2S_z^{\mathrm{back}} = -\left(b^{\dagger}A^{\dagger}\Phi b + b^{\dagger}\Phi^{\dagger}Ab\right) - \left(a^{\dagger}\Phi^{\dagger}Ab - a^{\dagger}A^{\dagger}\Phi b\right). \tag{62}$$

Computationally, we precompute the intermediates $\alpha = \Phi^{\dagger}Aa$ and $\beta = \Phi^{\dagger}Ab$,

$$2S_z^{\mathrm{forw}} = \left(\alpha^{\dagger}a + a^{\dagger}\alpha\right) + \left(b^{\dagger}\alpha - \beta^{\dagger}a\right) \tag{63}$$

$$2S_z^{\mathrm{back}} = -\left(\beta^{\dagger}b + b^{\dagger}\beta\right) + \left(b^{\dagger}\alpha - \beta^{\dagger}a\right)^{*}. \tag{64}$$

We similarly define the power flux of an individual diffraction order (indexed by i) as

$$\begin{aligned} 2S_i^{\mathrm{forw}} &= \left[(Aa)_i^*(\Phi a)_i + (\Phi a)_i^*(Aa)_i\right] \\ &\quad + \left[(\Phi b)_i^*(Aa)_i - (Ab)_i^*(\Phi a)_i\right] \end{aligned} \tag{65}$$

$$\begin{aligned} 2S_i^{\mathrm{back}} &= -\left[(Ab)_i^*(\Phi b)_i + (\Phi b)_i^*(Ab)_i\right] \\ &\quad + \left[(\Phi b)_i^*(Aa)_i - (Ab)_i^*(\Phi a)_i\right]^{*}. \end{aligned} \tag{66}$$

We can then precompute $\alpha_h = \Phi a$, $\beta_h = \Phi b$, $\alpha_e = Aa$, and $\beta_e = Ab$, resulting in

$$\begin{aligned} 2S_i^{\text{forw}} &= \left[(\alpha_e)_i^*(\alpha_h)_i + (\alpha_h)_i^*(\alpha_e)_i\right] \\ &\quad + \left[(\beta_h)_i^*(\alpha_e)_i - (\beta_e)_i^*(\alpha_h)_i\right] \end{aligned} \tag{67}$$

$$\begin{aligned} 2S_i^{\text{back}} &= -\left[(\beta_e)_i^*(\beta_h)_i + (\beta_h)_i^*(\beta_e)_i\right] \\ &\quad + \left[(\beta_h)_i^*(\alpha_e)_i - (\beta_e)_i^*(\alpha_h)_i\right]^* . \end{aligned} \tag{68}$$

5.2. Stress tensor

The computation of optical forces between photonic crystal slabs has recently received interest. The integral of the stress tensor over a surface parallel to layer surfaces over the unit cell can be computed efficiently in the Fourier basis. The components of the time averaged stress tensor are defined by

$$\langle T_{ab}\rangle = \frac{1}{2}\mathrm{Re}\left[E_a D_b^* + H_a B_b^* - \frac{1}{2}\delta_{ab}\sum_c \left(E_c D_c^* + H_c B_c^*\right)\right] \tag{69}$$

where the subscripts a, b, and c can each be x, y, and z. Using the Fourier representation of the fields, as in Eq. (5), the integral over a unit cell becomes

$$\begin{aligned} \int_{\text{cell}} \langle T_{ab}\rangle\, dS_z &= \frac{A}{2}\mathrm{Re}\sum_{\mathbf{G}}\Bigg[e_{\mathbf{G},a} d^*_{\mathbf{G},z} + h_{\mathbf{G},a} b^*_{\mathbf{G},z} \\ &\quad - \frac{1}{2}\delta_{az}\sum_c \left(e_{\mathbf{G},c} d^*_{\mathbf{G},c} + h_{\mathbf{G},c} b^*_{\mathbf{G},c}\right)\Bigg] \end{aligned} \tag{70}$$

where A is the area of the unit cell and dS_z is the differential area normal to the z-axis.

5.3. Energy density

The energy density within a layer can also be computed in closed form. For the i-th layer, the energy density is

$$U = \int_{z_i}^{z_i+d_i}\int_{\text{cell}} (\mathbf{E}^*\cdot\mathbf{D} + \mathbf{H}^*\cdot\mathbf{B})\, d^2\mathbf{r}\, dz. \tag{71}$$

Suppose we have a field vector in the Fourier basis: $F(z) = [-e_y, e_x, h_x, h_y, e_z, -h_z]^{\mathsf{T}}$. Using Eqs. (11), (14) and (35), this can be written as

$$F(z) = C\begin{bmatrix} f(z) & \\ & f(d-z)\end{bmatrix}\begin{bmatrix} a \\ b\end{bmatrix} \tag{72}$$

where C is a matrix, $f(z)$ was defined in Eq. (30), and a and b are mode amplitude vectors defined in Eqs. (33) and (34). All the transverse dependence of the modes is of the form $\exp[i(\mathbf{k}+\mathbf{G})\cdot\mathbf{r}]$, so by orthogonality,

$$U = A\int_{z_i}^{z_i+d_i} F^\dagger \Lambda F\, dz \tag{73}$$

where A is the area of the unit cell and Λ is a block diagonal matrix $\Lambda = \mathrm{diag}(\mathcal{E}, I, I, \hat{\epsilon}, I)$. Let

$$Q(z) = \begin{bmatrix} f^\dagger(z) & \\ & f^\dagger(d-z)\end{bmatrix} C^\dagger \Lambda C \begin{bmatrix} f(z) & \\ & f(d-z)\end{bmatrix}. \tag{74}$$

Then the problem is to evaluate

$$U = A\int_{z_i}^{z_i+d_i}\begin{bmatrix} a \\ b\end{bmatrix}^\dagger Q(z)\begin{bmatrix} a \\ b\end{bmatrix} dz. \tag{75}$$

Each matrix element of the quadratic form in the integrand can be integrated separately in closed form. Similarly, expressions for the electric or magnetic energy density can also be obtained with modified forms of Λ. The ability to evaluate these densities in closed form allows rapid evaluation of quantities such as effective mode volume, which is useful for optical cavity design and optimization [40].

6. Mode solving

In addition to solving for the fields in response to sources, S^4 can also solve for modes of a structure (the homogeneous problem). This capability is of great interest for the engineering of LED emission [41] and exploring spontaneous emission in periodic structures [42]. Mode solving computations do not assume any sources. The S-matrix algorithm allows the efficient determination of the S-matrix of the entire structure for constant ω and $\mathbf{k}$. Using this basic capability, the dispersion relation of guided and leaky modes can be obtained using a root-finding algorithm which seeks the singularities of the S-matrix determinant [41].

7. User scripting and programming interface

Like many other popular CEM tools, S^4 is embedded within a scripting language environment allowing users to easily use the solver as a basic unit in a frequency scan or design loop. The scripting language Lua [43] was chosen for its ease of use for both user and programmer, featuring an intuitive syntax and easily accommodating both objective and functional programming paradigms. The use of Lua also means that all of the user-exposed functionality is available in a low level programming interface. In fact, each user-callable Lua function is a thin wrapper around a C++ API function. This allows S^4 to be integrated within a higher level package very straightforwardly.

The user manual and scripting interface to S^4 are described in detail at http://fan.group.stanford.edu/S4. Included with the software are numerous examples of calculations appearing in publications, which cover almost all of the available functionality. We will describe a simple script, listed in Fig. 4, to illustrate the basic simulation flow. We will focus on computing the transmission spectrum of a photonic crystal slab (in fact, Fig. 12 of [44], originally computed by the FDTD method) whose structure is shown in Fig. 5. The structure is composed of a silicon slab ($\epsilon = 12$) with thickness $0.5a$ surrounded by air, with a square lattice of air holes with radius $0.2a$, where a is the lattice constant.

First, we obtain a new `Simulation` object, which forms the basis of all user interaction with S^4. The reason for interacting through an object instead of setting simulation parameters globally is to enable efficient multi-threaded parallelization, a common use-case for small to medium sized simulations. We first set the lattice vectors which define the periodicity in the xy-plane, the number of Fourier components to use ($N = |\mathbf{G}|$), and settings to choose a particular FMM formulation.

Next, we define the materials used in the simulation and the structure layers. The reason for applying names to material constants instead of supplying numbers directly is to simplify calculations involving frequency dependent materials. By referring to materials by name, only material definitions need to be changed at different frequencies, rather than having to modifying the structure description. The layers of the structure are specified one after another in the direction of increasing z coordinate. All layers default to unpatterned uniform layers unless patterns are specified.

The final step of defining a simulation is specifying the excitation condition. In the example, a plane wave is normally incident on the structure towards the positive z direction. Up to this point, no computations have been performed; any expensive computations are triggered by user requests for outputs that require them.

```
-- Bottom pane of Fig. 12 in: S. Fan and J. D. Joannopoulos,
-- "Analysis of guided resonances in photonic crystal slabs",
-- Phys. Rev. B, Vol. 65, 235112.

S = S4.NewSimulation()
S:SetLattice({1,0}, {0,1})           -- lattice vectors (1,0) and (0,1)
S:SetNumG(100)
S:UsePolarizationDecomposition()

---------------------------- Add materials ----------------------------
S:AddMaterial("Silicon", {12,0})     -- real and imaginary parts
S:AddMaterial("Air", {1,0})

----------------------------- Add Layers ------------------------------
S:AddLayer(’AirAbove’,               -- layer name
           0,                        -- thickness
           ’Air’)                    -- material

S:AddLayer(’Slab’, 0.5, ’Silicon’)
S:SetLayerPatternCircle(’Slab’,      -- layer to modify
                        ’Air’,       -- material inside hole
                        {0,0},       -- center of circle
                        0.2)         -- radius

S:AddLayerCopy(’AirBelow’,           -- layer name
               0,                    -- thickness
               ’AirAbove’)           -- layer to copy

--------------------------- Set Excitation ----------------------------
S:SetExcitationPlanewave(
    {0,0}, -- incidence angles (phi in [0, 180], theta in [0, 360])
    {1,0}, -- s-polarization amplitude and phase [0, 360] deg
    {0,0}) -- p-polarization amplitude and phase

-- Compute spectrum from f = 0.25 to 0.6 in steps of 0.001 (2*pi*c/a)
for freq = 0.25, 0.6, 0.001 do
    S:SetFrequency(freq)
    -- Get incident and reflected powers in the first layer at 0 offset
    forward, backward = S:GetPoyntingFlux(’AirAbove’, 0)
    -- Get transmitted powers in the last layer at 0 offset
    forward = S:GetPoyntingFlux(’AirBelow’, 0)
    -- Output a table of: freq, transmitted, reflected powers
    print(freq, forward, backward)
end
```

Fig. 4. Example S^4 input script for reproducing Fig. 12 of [44]. The structure is shown in Fig. 5.

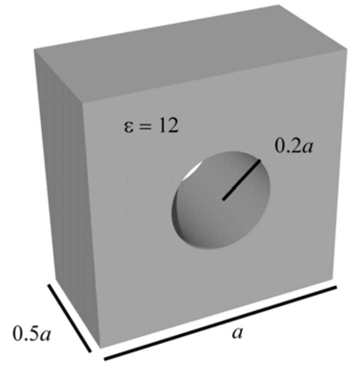


Fig. 5. One unit cell of a photonic crystal slab of silicon ($\epsilon = 12$) consisting of a square lattice of air holes with lattice constant a, hole radius $0.2a$, and thickness $0.5a$. An example simulation script involving this structure is listed in Fig. 4, and a small portion of the transmission spectrum is shown in Fig. 7.

To obtain a transmission spectrum for the example structure, we must perform the simulation at many frequencies in an interval. We loop over the frequency range of interest setting the frequency of the simulation and retrieving the transmitted and reflected powers. Functions that compute outputs in a particular layer cache the solutions within a layer to avoid duplication of work. For example, if field amplitudes within the air layers were requested at the end of the loop body, they would not require re-solving the system.

In the example shown, there is never any implicit spatial discretization of the structure. Throughout S^4, discretizations are always avoided unless the user explicitly requests a discretized shape (such as polygons) or discretized formulation (such as using FFTs). Therefore, smooth parameter variations generally always produce smoothly varying results. This is especially important in structure optimization where very small geometric changes must produce correspondingly small changes to spectral features.

7.1. Parallelization

Parallelization within S^4 is specified explicitly using the few parallelization primitives provided. Any computation performed for a single structure at a single frequency is not inherently

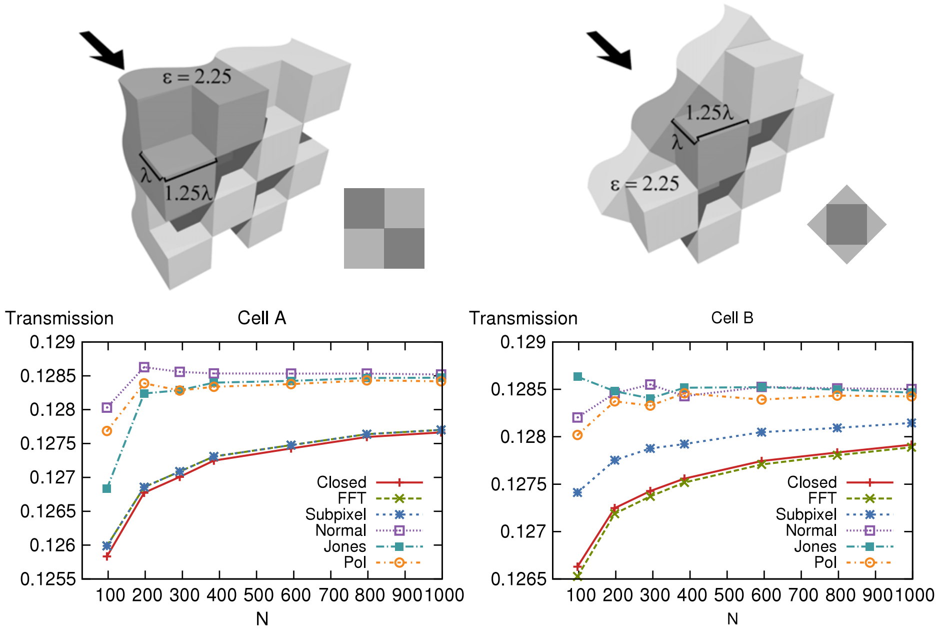


Fig. 6. Convergence of the transmitted power in the $(0, -1)$-th diffraction order with respect to the Fourier truncation order for the checkerboard grating for two different unit cells (shown above). The unit cell is shaded darker in the 3D structures and a planar schematic of each unit cell is shown in the insets. The direction of incident radiation is shown with an arrow and the structure is infinite in extent into the page (as indicated by the wavy back boundary).

parallelized. However, for almost all applications, a frequency spectrum is usually required, and the computations at different frequencies are completely independent of each other. Thus, on distributed memory machines, each node can independently compute disjoint portions of frequency spectra using only knowledge of the node index. S^4 supports the Message Passing Interface (MPI) and the rank and size of the MPI world communicator are available as global variables to scripts.

Parallelism is also possible on shared memory machines by explicitly requesting solutions for multiple `Simulation` objects to be performed simultaneously in different execution threads. S^4 provides basic functionality to clone `Simulation` objects and to compute layer solutions for a set of `Simulation` objects to achieve thread-level parallelism.

8. Convergence and computational cost for dielectric structures

The accuracy of FMM is directly related to N (the number of $\mathbf{G}$ used in the simulation) and the convergence of the Fourier series representations of the layer dielectric functions. As the number of Fourier components used (N) increases, the dielectric distribution within layers is represented with greater accuracy. However, the space and time requirements of the eigenvalue problem described in Section 3.3 make simulations involving large N prohibitively expensive (for example, for $N = 1000$, storing the S-matrix alone would require 256 MB of memory). As seen in Section 4, much work has focused on improving the convergence of the FMM with respect to N so that fewer Fourier components may be used. Typically used values of N are in the low to mid-hundreds as suggested by the convergence plots.

We begin by analyzing a dielectric structure described in [4], composed of a checkerboard grating for two different choices of the unit cell. The grating is made of $\epsilon = 2.25$ material in the incident halfspace ($z < 0$), and vacuum on the other side. The width and depth of the squares of the grating are 1.25λ and 1λ, respectively, where λ is the free space wavelength of light. A plane wave is normally incident on the grating and polarized parallel to the edges of the squares. Fig. 6 shows details of the structure as well as the convergence of the transmitted power in the $(0, -1)$-th diffraction order for the formulations whose names were given in Section 4. It is clear that the vector field based formulations achieve significantly faster convergence than the other methods. The Pol method shows slightly faster and more stable convergence behavior than the Jones and Normal formulations. It is interesting to note that for Cell A, where the material interfaces are parallel to the coordinate directions, the subpixel averaging method performs almost identically to the FFT method since the pixels do not straddle any interfaces. However, for Cell B, the interfaces do not align with the pixel boundaries and the Subpixel formulation shows a clear advantage over using only the FFT.

At $N = 1000$, the slow converging formulations are still far from convergence while at the same time producing relative error on the order of 1%. Rapid convergence is extremely important for the FMM since it may be impractical or impossible to use larger N.

In order to investigate convergence properties more relevant to nanophotonics, we analyze the convergence of the resonance frequencies of a photonic crystal slab described in [44] and shown in Fig. 5. A portion of the transmission spectrum is shown in Fig. 7 which shows three total transmission peaks. The convergence of the frequencies at which these features are observed for increasing truncation orders is shown in Fig. 8. Once again, the vector field based methods generally converge more rapidly than the other methods. Among them, the Pol method described in Section 4.4 appears to have the most rapid and stable convergence. Surprisingly, the Subpixel formulation performs very poorly for two of the resonance frequencies. This may be caused by the fact that the subpixel averaging modifies the average dielectric constant over the cell, which would cause an overall shift in resonance frequencies. For these numerical experiments, the grid resolution was set to eight times the maximum reciprocal lattice extent (in terms of integer orders).

9. Using the FMM for metals

Recent interest in nano-plasmonics has led to many applications of the FMM for studying structures containing metals. The

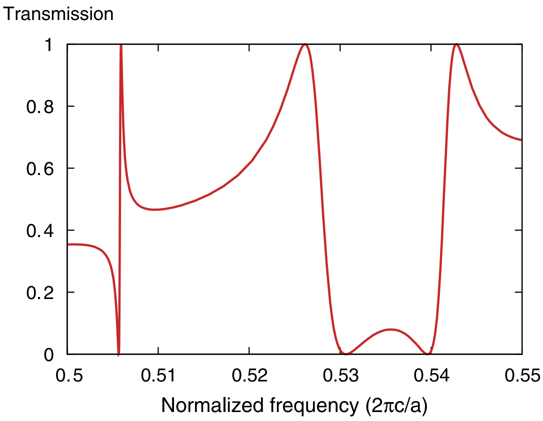


Fig. 7. Transmission spectrum of the photonic crystal slab shown in Fig. 5 exhibiting three total transmission peaks. The convergence behavior of the frequencies at which these peaks occur is shown in Fig. 8.

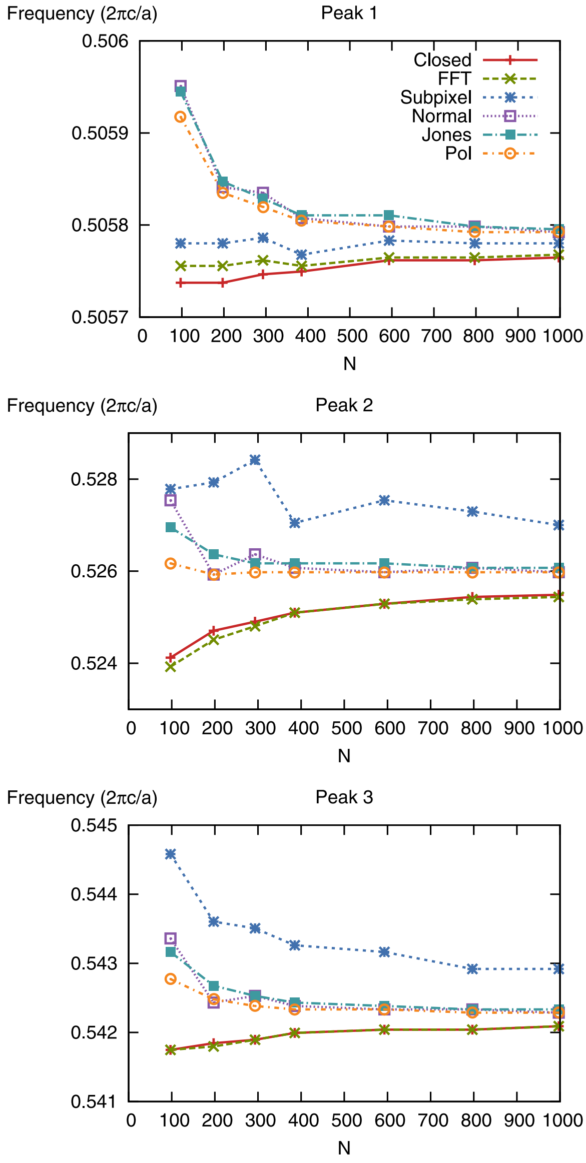


Fig. 8. Convergence of the three total transmission peak frequencies in Fig. 7 for the photonic crystal slab structure shown in Fig. 5.

dielectric constant of metals in the plasmonic regime can be very negative ($\epsilon = -100$ is not unusual) compared to typical positive dielectric values on the order of 1–10. The presence of metals leads to poor performance of the FMM, and a number of works have explored the fundamental reasons for these problems as well as proposing possible solutions [45–47]. We will outline several difficulties encountered when using metals in the FMM.

The first and most obvious difficulty is that in the presence of metals, the dielectric contrast at interfaces is much greater. Discontinuities in the dielectric function lead to very slow convergence of the Fourier series representation due to the Gibbs phenomenon [48]. The Fourier series coefficients of a discontinuous function decay as $1/f$ where f is the spatial frequency. Since the coefficients are generally proportional to the dielectric contrast, to achieve approximately the same level of accuracy with 2D periodicity, $\sqrt{N}$ must scale in proportion to the largest dielectric contrast in a structure.

Secondly, the Gibbs phenomenon, due to finite truncation of a Fourier series, introduces ringing in the real space reconstruction of a function. In the context of the FMM, this means that at metal–dielectric interfaces, it is possible to introduce additional fictitious interfaces where the dielectric function crosses zero, leading to spurious modes [49]. Note that the amplitude of the Gibbs ringing does not decrease by increasing the number of Fourier components, so this problem is independent of N [50].

Finally, at metal–dielectric corners and edges, the electric field can become singular due to charge buildup [51]. The Fourier representation of fields simply cannot capture this singular behavior, leading to poor convergence with respect to N. In fact it has been shown that there are geometries for which the FMM will *never* converge [52].

However, the applicability and proper formulation of the FMM for metallic structures to resolve the above mentioned problems is still an open question. The design of S^4 facilitates experimentation and development of new formulations of the FMM, and can serve as a platform for further work in new application domains.

10. Concluding remarks

The availability of an open FMM code will be of great importance in the growing field of nanophotonics and metamaterials. We have described S^4, a flexible open-source framework for performing simulations using the FMM. Common calculations, such as computing transmission and reflection coefficients of grating structures, are specified straightforwardly using a simple programming interface. More sophisticated calculations, such as mode solving, are enabled through the flexible scripting interface. The interface has been designed to hide almost all of the numerical details without sacrificing efficiency, while at the same time allowing access to all of the algorithmic flexibility. A unique aspect of our implementation is the ease of integrating and comparing new variants of the FMM. S^4 already includes six different FMM formulations, among them state-of-the-art techniques that use the proper Fourier factorization rules to achieve superior convergence compared to the classical formulations.

We have validated S^4 against existing published FMM results, as shown in the convergence analysis, as well as against published results produced by other simulation methods such as FDTD and FDFD, such as the example shown in Fig. 4. The code is bundled with a number of these validation test cases as well as didactic examples. In addition, S^4 has already been used in the investigation of a diverse set of phenomena, including optical forces [53], biosensing [54], optical isolation [55], and light-trapping in solar cells [56].

Acknowledgments

This work is supported in part by the United States Air Force Office of Scientific Research (USAFOSR) grant FA9550-09-1-0704, and the National Science Foundation (NSF) grant DMS-0968809. Victor Liu is supported by a Stanford Graduate Fellowship. We thank Wonseok Shin and the reviewers for a very thorough and careful proofreading of the manuscript.

References

[1] A. Taflove, S. Hagness, Computational Electrodynamics: The Finite-Difference Time-Domain Method, in: Artech House Antennas and Propagation Library, Artech House, 2005.

[2] G. Veronis, R.W. Dutton, S. Fan, Method for sensitivity analysis of photonic crystal devices, Opt. Lett. 29 (2004) 2288–2290.

[3] J. Jin, The Finite Element Method in Electromagnetics, A Wiley-Interscience Publication, Wiley, 2002.

[4] L. Li, New formulation of the Fourier modal method for crossed surface-relief gratings, J. Opt. Soc. Amer. A 14 (1997) 2758–2767.

[5] H. Kogelnik, Coupled wave theory for thick hologram gratings, Bell Syst. Tech. J. 48 (1969) 2909–2947.

[6] F.G. Kaspar, Diffraction by thick, periodically stratified gratings with complex dielectric constant, J. Opt. Soc. Amer. 63 (1973) 37–45.

[7] M.G. Moharam, T.K. Gaylord, Rigorous coupled-wave analysis of planar-grating diffraction, J. Opt. Soc. Amer. 71 (1981) 811–818.

[8] R. Bräuer, O. Bryngdahl, Electromagnetic diffraction analysis of two-dimensional gratings, Opt. Commun. 100 (1993) 1–5.

[9] M.G. Moharam, D.A. Pommet, E.B. Grann, T.K. Gaylord, Stable implementation of the rigorous coupled-wave analysis for surface-relief gratings: enhanced transmittance matrix approach, J. Opt. Soc. Amer. A 12 (1995) 1077–1086.

[10] D.M. Whittaker, I.S. Culshaw, Scattering-matrix treatment of patterned multilayer photonic structures, Phys. Rev. B 60 (1999) 2610–2618.

[11] M. Liscidini, D. Gerace, L.C. Andreani, J.E. Sipe, Scattering-matrix analysis of periodically patterned multilayers with asymmetric unit cells and birefringent media, Phys. Rev. B 77 (2008) 035324.

[12] S.G. Tikhodeev, A.L. Yablonskii, E.A. Muljarov, N.A. Gippius, T. Ishihara, Quasiguided modes and optical properties of photonic crystal slabs, Phys. Rev. B 66 (2002) 045102.

[13] L. Li, Formulation and comparison of two recursive matrix algorithms for modeling layered diffraction gratings, J. Opt. Soc. Amer. A 13 (1996) 1024–1035.

[14] F. Montiel, M. Nevière, Differential theory of gratings: extension to deep gratings of arbitrary profile and permittivity through the R-matrix propagation algorithm, J. Opt. Soc. Amer. A 11 (1994) 3241–3250.

[15] J. Chandezon, M.T. Dupuis, G. Cornet, D. Maystre, Multicoated gratings: a differential formalism applicable in the entire optical region, J. Opt. Soc. Amer. 72 (1982) 839–846.

[16] K. Knop, Rigorous diffraction theory for transmission phase gratings with deep rectangular grooves, J. Opt. Soc. Amer. 68 (1978) 1206–1210.

[17] L. Li, C.W. Haggans, Convergence of the coupled-wave method for metallic lamellar diffraction gratings, J. Opt. Soc. Amer. A 10 (1993) 1184–1189.

[18] P. Lalanne, G.M. Morris, Highly improved convergence of the coupled-wave method for TM polarization, J. Opt. Soc. Amer. A 13 (1996) 779–784.

[19] G. Granet, B. Guizal, Efficient implementation of the coupled-wave method for metallic lamellar gratings in TM polarization, J. Opt. Soc. Amer. A 13 (1996) 1019–1023.

[20] L. Li, Use of Fourier series in the analysis of discontinuous periodic structures, J. Opt. Soc. Amer. A 13 (1996) 1870–1876.

[21] E. Popov, M. Nevière, Grating theory: new equations in Fourier space leading to fast converging results for TM polarization, J. Opt. Soc. Amer. A 17 (2000) 1773–1784.

[22] B. Chernov, M. Nevière, E. Popov, Fast Fourier factorization method applied to modal analysis of slanted lamellar diffraction gratings in conical mountings, Opt. Commun. 194 (2001) 289–297.

[23] E. Popov, M. Nevière, Maxwell equations in Fourier space: fast-converging formulation for diffraction by arbitrary shaped, periodic, anisotropic media, J. Opt. Soc. Amer. A 18 (2001) 2886–2894.

[24] P. Boyer, E. Popov, M. Nevière, G. Tayeb, Diffraction theory in TM polarization: application of the fast Fourier factorization method to cylindrical devices with arbitrary cross section, J. Opt. Soc. Amer. A 21 (2004) 2146–2153.

[25] T. Schuster, J. Ruoff, N. Kerwien, S. Rafler, W. Osten, Normal vector method for convergence improvement using the RCWA for crossed gratings, J. Opt. Soc. Amer. A 24 (2007) 2880–2890.

[26] R. Antos, Fourier factorization with complex polarization bases in modeling optics of discontinuous bi-periodic structures, Opt. Express 17 (2009) 7269–7274.

[27] P. Götz, T. Schuster, K. Frenner, S. Rafler, W. Osten, Normal vector method for the RCWA with automated vector field generation, Opt. Express 16 (2008) 17295–17301.

[28] G. Granet, Reformulation of the lamellar grating problem through the concept of adaptive spatial resolution, J. Opt. Soc. Amer. A 16 (1999) 2510–2516.

[29] T. Weiss, G. Granet, N.A. Gippius, S.G. Tikhodeev, H. Giessen, Matched coordinates and adaptive spatial resolution in the Fourier modal method, Opt. Express 17 (2009) 8051–8061.

[30] S. Essig, K. Busch, Generation of adaptive coordinates and their use in the Fourier modal method, Opt. Express 18 (2010) 23258–23274.

[31] T. Vallius, M. Honkanen, Reformulation of the Fourier modal method with adaptive spatial resolution: application to multilevel profiles, Opt. Express 10 (2002) 24–34.

[32] L. Li, Fourier modal method for crossed anisotropic gratings with arbitrary permittivity and permeability tensors, J. Opt. A: Pure Appl. Opt. 5 (2003) 345.

[33] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, D. Sorensen, LAPACK Users' Guide, third ed., Society for Industrial and Applied Mathematics, Philadelphia, PA, 1999.

[34] G. Golub, C.F. Van Loan, Matrix Computations, third ed., Johns Hopkins University Press, Baltimore, 1996.

[35] K. McInturff, P. Simon, The Fourier transform of linearly varying functions with polygonal support, IEEE Trans. Antennas Propag. 39 (1991) 1441–1443.

[36] C. Kottke, A. Farjadpour, S.G. Johnson, Perturbation theory for anisotropic dielectric interfaces, and application to subpixel smoothing of discretized numerical methods, Phys. Rev. E 77 (2008) 036611.

[37] A.F. Oskooi, D. Roundy, M. Ibanescu, P. Bermel, J.D. Joannopoulos, S.G. Johnson, MEEP: a flexible free-software package for electromagnetic simulations by the FDTD method, Comput. Phys. Comm. 181 (2010) 687–702.

[38] S.G. Johnson, J.D. Joannopoulos, Block-iterative frequency-domain methods for Maxwell's equations in a planewave basis, Opt. Express 8 (2001) 173–190.

[39] M. Fisher, P. Schröder, M. Desbrun, H. Hoppe, Design of tangent vector fields, ACM Trans. Graph. 26 (2007).

[40] K. Vahala, Optical Microcavities, World Scientific, Singapore, 2004.

[41] H. Taniyama, M. Notomi, *S*-matrix calculation of radiation characteristics from dipole oscillation in two-dimensional photonic crystal slabs, J. Appl. Phys. 103 (2008) 083115.

[42] S. Fan, P.R. Villeneuve, J.D. Joannopoulos, E.F. Schubert, High extraction efficiency of spontaneous emission from slabs of photonic crystals, Phys. Rev. Lett. 78 (1997) 3294–3297.

[43] R. Ierusalimschy, L.H. de Figueiredo, W.C. Filho, Lua—an extensible extension language, Softw.-Pract. Exp. 26 (1996) 635–652.

[44] S. Fan, J.D. Joannopoulos, Analysis of guided resonances in photonic crystal slabs, Phys. Rev. B 65 (2002) 235112.

[45] E. Popov, B. Chernov, M. Nevière, N. Bonod, Differential theory: application to highly conducting gratings, J. Opt. Soc. Amer. A 21 (2004) 199–206.

[46] K.M. Gundu, A. Mafi, Reliable computation of scattering from metallic binary gratings using Fourier-based modal methods, J. Opt. Soc. Amer. A 27 (2010) 1694–1700.

[47] K.M. Gundu, A. Mafi, Constrained least squares Fourier modal method for computing scattering from metallic binary gratings, J. Opt. Soc. Amer. A 27 (2010) 2375–2380.

[48] H.S. Carslaw, Introduction to the Theory of Fourier's Series and Integrals, third ed., Dover Publications Inc., New York, 1930.

[49] N.M. Lyndin, O. Parriaux, A.V. Tishchenko, Modal analysis and suppression of the Fourier modal method instabilities in highly conductive gratings, J. Opt. Soc. Amer. A 24 (2007) 3781–3788.

[50] D. Gottlieb, C.-W. Shu, On the Gibbs phenomenon and its resolution, SIAM Rev. 39 (1997) 644–668.

[51] J. Meixner, The behavior of electromagnetic fields at edges, IEEE Trans. Antennas Propag. 20 (1972) 442–446.

[52] L. Li, G. Granet, Field singularities at lossless metal–dielectric right-angle edges and their ramifications to the numerical modeling of gratings, J. Opt. Soc. Amer. A 28 (2011) 738–746.

[53] V. Liu, M. Povinelli, S. Fan, Resonance-enhanced optical forces between coupled photonic crystal slabs, Opt. Express 17 (2009) 21897–21909.

[54] M.E. Beheiry, V. Liu, S. Fan, O. Levi, Sensitivity enhancement in photonic crystal slab biosensors, Opt. Express 18 (2010) 22702–22714.

[55] K. Fang, Z. Yu, V. Liu, S. Fan, Ultracompact nonreciprocal optical isolator based on guided resonance in a magneto-optical photonic crystal slab, Opt. Lett. 36 (2011) 4254–4256.

[56] Z. Yu, A. Raman, S. Fan, Fundamental limit of nanophotonic light trapping in solar cells, Proc. Natl. Acad. Sci. (2010).