Computer Physics Communications 231 (2018) 163–172

Contents lists available at ScienceDirect

Computer Physics Communications

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

MESH: A free electromagnetic solver for far-field and near-field radiative heat transfer for layered periodic structures[☆]

Kaifeng Chen[a,b,*], Bo Zhao[b], Shanhui Fan[b]

[a] Department of Applied Physics, Ginzton Laboratory, Stanford University, Stanford, CA 94305, USA
[b] Department of Electrical Engineering, Stanford University, Stanford, CA 94305, USA

ARTICLE INFO

Article history:
Received 17 November 2017
Received in revised form 9 April 2018
Accepted 27 April 2018
Available online 8 May 2018

Keywords:
Computational electromagnetics
Radiative heat transfer
Rigorous coupled wave analysis
Scattering matrix method
Fourier model method

ABSTRACT

We describe MESH (**M**ultilayer **E**lectromagnetic **S**olver for **H**eat transfer), a free software that combines rigorous coupled wave analysis (RCWA) and scattering matrix formalism to simulate the radiative heat transfer both in the near-field and far-field regimes for layered three-dimensional structures made of planar layers. Each layer can have in-plane one-dimensional or two-dimensional periodicity. In this paper, we provide detailed discussions of the algorithms of MESH, which enables it to be a flexible tool for different types of radiative heat transfer simulations. We also discuss aspects of the codes related to parallelization and user scripting.

Program summary
Program Title: MESH
Program Files doi: http://dx.doi.org/10.17632/zx9v3bf3hf.1
Licensing provisions: GNU General Public License 3 (GPL)
Programming language: C, C++.
External routines: Lua[1], Python[2] and LAPACK and BLAS linear-algebra software[3], and optionally MPI message-passing interface[4]. Armadillo[5] is already contained in MESH.
Nature of problem: Far-field and near-field radiative heat transfer in layered periodic structures.
Solution method: Fourier modal method (rigorous coupled wave analysis) and the scattering matrix method.
[1] R. Ierusalimschy, L.H. de Figueiredo, W.C. Filho, Lua an extensible extension language, Software: Practice and Experience 26 (1996) 635652. http://www.lua.org.
[2] Python Software Foundation. Available at http://www.python.org
[3] MKL: https://software.intel.com/en-us/intel-mkl
[4] T.M. Forum, MPI: A Message Passing Interface, in: Supercomputing 93, Portland, OR, 878883, 1993
[5] Conrad Sanderson and Ryan Curtin. Armadillo: a template-based C++ library for linear algebra. Journal of Open Source Software, Vol. 1, pp. 26, 2016. http://dx.doi.org/10.21105/joss.00026

1. Introduction

Thermal radiation is one of the most fundamental aspects of nature [1]. The ability to control thermal radiation plays a significant role in a number of technological areas including energy conversion [2–7], cooling [8–12], as well as sensing and imaging [13,14]. In seeking to control both far-field and near-field thermal radiation, there are significant recent efforts exploring the use of layered structures with in-plane periodic index contrast. These structures have been extensively used to tailor either the angular

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

[*] Corresponding author at: Department of Applied Physics, Ginzton Laboratory, Stanford University, Stanford, CA 94305, USA.

E-mail address: kfchen@stanford.edu (K. Chen).

https://doi.org/10.1016/j.cpc.2018.04.032
0010-4655/© 2018 Elsevier B.V. All rights reserved.

and/or the spectral characteristics of far-field thermal radiation, as well as to enhance the magnitude of heat transfer in the near-field [15–20].

Fig. 1 shows some exemplary geometries that are common in both theoretical and experimental studies. Fig. 1(a) shows a structure that consists of two multilayer stacks separated by a vacuum gap. Each layer is associated with different temperatures. Fig. 1(b) shows a 1D periodic grating structure that emits thermal radiation to the free space above it. Fig. 1(c) shows the radiative heat transfer between two structures each consisting of a two-dimensionally periodic array of finite-height cylinders. MESH, which stands for **M**ultilayer **E**lectromagnetic **S**olver for **H**eat transfer, is designed to compute the far-field thermal radiation or near-field heat transfer in these geometries.

There already exist a few tools that can solve radiative heat transfer problem between different geometries. All these tools build upon computational electromagnetic techniques that solve Maxwell's equations. For example, boundary element method (BEM) has been used in SCUFF-EM [21–24] and BUFF-EM [25,26]. And finite-difference time-domain (FDTD) method [27] has been used in the NF-RT-FDTD algorithm [28]. MESH builds upon the Fourier Modal Method, which is sometimes referred to as the Rigorous Coupled Wave Analysis (RCWA). This method is particularly suitable for the layered structures shown in Fig. 1, where within each layer the structure is uniform along the vertical direction. For this class of structures, the Fourier Modal Method treats the field propagation along the vertical direction semi-analytically without discretization, and can thus achieve significant reduction in computational time and memory requirement as compared to BEM or FDTD. In addition, MESH uses a scattering matrix (S-matrix) formalism [29,30] which enables a stable propagation of the field along vertical direction without numerical instability.

To simulate radiative heat transfer, one needs to combine computational methods for solving Maxwell's equations, with the formalism of fluctuational electrodynamics. Thermal radiations originate from random current sources. To compute the thermal electromagnetic fields, one needs to integrate the contributions from all these random sources. Most papers on the heat transfer in layered geometries [10,19,31–37] used a scattering approach, where the heat flux is computed by evaluating the trace of certain field correlators. This approach has the advantage that it bypasses the computationally intensive steps of volume integration over the thermal current sources, but is hard to apply to the scenarios of many-body heat transfer, which consist of multiple bodies with different temperatures, such as the case in Fig. 1(a). In contrast, in MESH we directly compute the Green function that can be used to describe the fields from the random thermal sources [38,39], and then analytically perform the spatial integration over the emitting volume. In this paper, we present in detail about an efficient way to perform such volume integration over the random thermal sources, which is a key innovative aspect of MESH in terms of algorithm that has not been previously discussed in the literature.

The paper is organized as follows: Section 2 provides the background in the underlying notation of mathematics, and the /comphy6506 formalism of fluctuational electrodynamics, and derives a trace formula for thermal Poynting flux. Section 3 provides a detailed account of the algorithms that calculate the thermal Poynting flux using the formula as derived in Section 2. Section 4 discusses the geometries supported in MESH. In Section 5 we show an example of the user scripting interface and the corresponding results, and briefly discuss the convergence properties of MESH. Section 6 summarizes the paper.

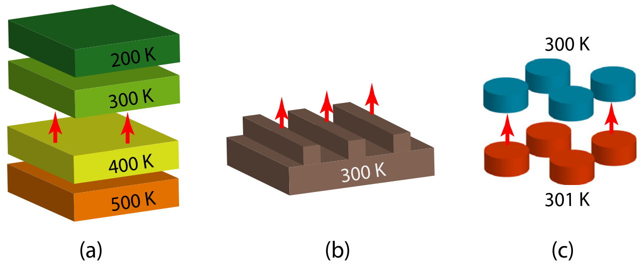

Fig. 1. Some exemplary geometries that can be simulated by MESH: (a) two multilayer stacks separated by a vacuum gap with each layer maintained at different temperatures. (b) a 1D periodic grating structure that emits radiation to the above free space. (c) two structures with each consisting of a two-dimensional periodic array of cylinders.

2. Problem formulation and geometric definitions

2.1. Convention of Fourier transform and stationary random processes

For periodic structures in general, it is convenient to reformulate the Maxwell's equations in Fourier space. In electromagnetism, all physical fields are real. For a real time (t) dependent function $f(t)$, its Fourier representation $f(\omega)$ in the frequency (ω) space is defined by

$$f(t) = \Re \int_0^{+\infty} d\omega\, e^{i\omega t} f(\omega). \tag{1}$$

The Fourier transform from k space k_x (function $f(k_x)$) to real space x (function $f(x)$), is defined as

$$f(x) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} dk_x\, e^{-ik_x x} f(k_x). \tag{2}$$

In thermal radiation calculation, we will be considering stationary random processes. In these processes, the ensemble average of any observable is independent of time. Consider the following ensemble average of the correlation function of two observables $A(t)$ and $B(t)$

$$\langle A(t)B(t)\rangle = \langle \Re\int_0^{\infty} d\omega\, A(\omega)e^{i\omega t} \cdot \Re\int_0^{\infty} d\omega'\, B(\omega')e^{i\omega' t}\rangle. \tag{3}$$

We define

$$\tilde{A}(\omega) = \begin{cases} A(\omega) & \omega \geq 0 \\ A^*(-\omega) & \omega < 0, \end{cases} \tag{4}$$

and similarly for $\tilde{B}(\omega)$. Eq. (4) becomes

$$\begin{aligned} \langle A(t)B(t)\rangle &= \langle \frac{1}{2}\int_{-\infty}^{\infty} d\omega\, \tilde{A}(\omega)e^{i\omega t} \cdot \frac{1}{2}\int_{-\infty}^{\infty} d\omega'\, \tilde{B}(\omega')e^{i\omega' t}\rangle \\ &= \frac{1}{4}\int_{-\infty}^{\infty} d\omega \int_{-\infty}^{\infty} d\omega'\, \langle \tilde{A}(\omega)\tilde{B}^*(\omega')\rangle e^{i(\omega-\omega')t}. \end{aligned} \tag{5}$$

In order to make $\langle A(t)B(t)\rangle$ time-independent, we must have

$$\langle \tilde{A}(\omega)\tilde{B}^*(\omega')\rangle \equiv \langle \tilde{A}\tilde{B}^*\rangle_\omega \delta(\omega-\omega'). \tag{6}$$

One can see that $\langle \tilde{A}\tilde{B}^*\rangle_{-\omega} = \langle \tilde{A}\tilde{B}^*\rangle_\omega^*$. For the rest of the paper, when there is no confusion, we suppress the subscript ω in $\langle \tilde{A}\tilde{B}^*\rangle_\omega$, then

$$\langle A(t)B(t)\rangle = \frac{1}{4}\int_{-\infty}^{\infty} d\omega \langle \tilde{A}\tilde{B}^*\rangle = \frac{1}{2}\int_0^{\infty} d\omega\, \Re\langle AB^*\rangle. \tag{7}$$

The convention of Fourier transformation in Eq. (1) naturally leads to the $e^{i\omega t}$ convention for the steady state Maxwell's equations:

$$\nabla \times \mathbf{H} = i\omega\epsilon\epsilon_0\mathbf{E}, \ \nabla \times \mathbf{E} = -i\omega\mu_0\mathbf{H}. \tag{8}$$

In MESH, we assume the materials are linear and nonmagnetic. These assumptions cover most of the cases considered in the literature.

2.2. Geometry and the formalism of fluctuational electrodynamics

In MESH, we assume that the structure consists of multiple material layers stacked in z-direction. Each layer is periodic in the x-y plane. A few examples of such structures are shown in Fig. 1. The material layers may in addition be surrounded by vacuum layers. Each of the material layers is maintained at a given temperature. And the temperature of different layers may be different.

MESH is developed to treat two scenarios: (1) Thermal radiation (Fig. 1(b)). Here a structure is adjacent to a semi-infinite vacuum region at least on one side. The objective of the calculation is to compute the thermal radiation power to the far-field in the vacuum region. (2) Heat transfer (Fig. 1(a) and (c)). Here the structure consists of at least two material layers separated by a vacuum gap. These two layers are maintained at different temperatures. The objective of the calculation is to compute the power that flows from one layer to the other.

Both scenarios outlined above can be treated in the formalism of fluctuational electrodynamics. In this formalism, the thermal electromagnetic fields are generated by fluctuating current sources, with the magnitude of the current fluctuation given by [40] (note that here we use the convention in Eq. (6) with the subscript ω suppressed)

$$\begin{aligned}\langle \mathbf{J}_\alpha(\mathbf{r}, z)\mathbf{J}^*_\beta(\mathbf{r}', z')\rangle &= \frac{4\epsilon_0\omega}{\pi}\,\frac{\epsilon_{\alpha\beta} - \epsilon^*_{\beta\alpha}}{2i}\,\Theta(\omega, T)\delta(\mathbf{r} - \mathbf{r}')\delta(z - z') \\ &= \frac{4\epsilon_0\omega}{\pi}\,\gamma_{\alpha\beta}\,\Theta(\omega, T)\delta(\mathbf{r} - \mathbf{r}')\delta(z - z'),\end{aligned} \tag{9}$$

where $\mathbf{r}$ and $\mathbf{r}'$ are the spatial vectors in the x-y plane, α and β denote components in a Cartesian coordinate system, $\epsilon_{\alpha\beta}$ is the element of the dielectric function which can be a tensor, $\gamma_{\alpha\beta}$ is the imaginary part of the dielectric tensor, $\delta(\cdot)$ is the Dirac δ-function and $\Theta(\omega, T)$ is the average photon energy in an optical mode at the angular frequency ω and temperature T, expressed as

$$\Theta(\omega, T) = \frac{\hbar\omega}{\exp\left(\frac{\hbar\omega}{k_B T}\right) - 1}. \tag{10}$$

In Eq. (10), $\hbar$ is the reduced Planck's constant, and k_B is the Boltzmann constant.

Eq. (9) can be written more compactly with a tensor notation. We define a tensor

$$\langle \mathbf{J}\mathbf{J}^*\rangle_{\alpha,\mathbf{r};\beta,\mathbf{r}'}(z, z') = \langle \mathbf{J}_\alpha(\mathbf{r}, z)\mathbf{J}^*_\beta(\mathbf{r}', z')\rangle. \tag{11}$$

Then Eq. (9) becomes

$$\langle \mathbf{J}\mathbf{J}^*\rangle(z, z') = \frac{4\epsilon_0\omega}{\pi}\Theta(\omega, T)\delta(z - z')\Gamma, \tag{12}$$

where Γ in the indexed form is

$$\Gamma_{\alpha,\mathbf{r};\beta,\mathbf{r}'} \equiv \gamma_{\alpha\beta}\delta(\mathbf{r} - \mathbf{r}'). \tag{13}$$

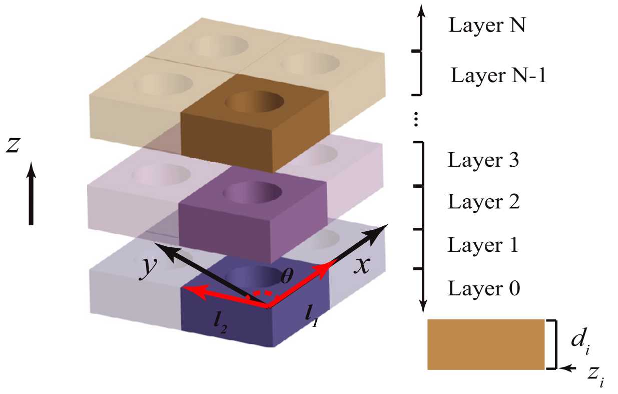

Fig. 2. Geometry setup in MESH with the layers stacked in the order from 0 to N. The lattice vectors are also highlighted in red, with $\mathbf{l}_1$ aligned with x axis and $\mathbf{l}_2$ always in the upper x–y plane. The angle between $\mathbf{l}_1$ and $\mathbf{l}_2$ is θ. The lengths of $\mathbf{l}_1$, $\mathbf{l}_2$ and the value of θ are the input parameters to MESH. For each layer, d_i is the thickness, and z_i is the z-coordinate of the lower edge of the layer. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

For subsequent use, the product of two tensors $\mathcal{A}$ and $\mathcal{B}$ are defined as

$$(\mathcal{A}\mathcal{B})_{\alpha,\mathbf{r};\beta,\mathbf{r}'} = \sum_\mu \int d^2\mathbf{r}''\mathcal{A}_{\alpha,\mathbf{r};\mu,\mathbf{r}''}\mathcal{B}_{\mu,\mathbf{r}'';\beta,\mathbf{r}'}. \tag{14}$$

In all our calculations, the in-plane spatial coordinates are restricted to the single unit cell.

To compute either the transferred power or the radiated power, one will need to compute the ensemble-averaged Poynting vector $\mathbf{S}$ along the z-direction. For this purpose, MESH sets up the geometry illustrated in Fig. 2. The layers in MESH are stacked in order from 0 to N, and each layer is indexed by i with thickness d_i, extending from z_i to $z_i + d_i$. The 0th layer has $z_0 = 0$, and the topmost and bottommost layers (i.e. the 0th and the Nth layers) are assumed to be semi-infinite. The definition of lattice vectors $\mathbf{l}_1$ and $\mathbf{l}_2$ are also illustrated as red arrows in Fig. 2. MESH evaluates the Poynting vector at position z in the probe layer t, from the emission of all the thermal sources in source layers, as is shown in Fig. 3. To compute such Poynting vector, we denote a fluctuating random current source at z' in one of the emitting layers s, and develop the numerical algorithm that efficiently integrates over z' to provide the total contributions of all the sources in the emitting layers.

2.3. Trace formula for the Poynting flux

Using the notation in Section 2.1, the z-component of the spectral Poynting vector $\mathbf{S}$ at angular frequency ω can be expressed as

$$\mathbf{S}_z = \frac{1}{2}\Re\left[\hat{\mathbf{z}} \cdot (\mathbf{E} \times \mathbf{H}^*)\right] = \frac{1}{2}\Re\left[\pi_{z\alpha\beta}\mathbf{E}_\alpha\mathbf{H}^*_\beta\right], \tag{15}$$

where $\hat{\mathbf{z}}$ is the unit vector in z direction, $\mathbf{E}$ and $\mathbf{H}$ are the total electric and magnetic fields, the subscripts α and β denote the corresponding component of the fields, and π is the Levi-Civita symbol. We further define a tensor Π, which in the indexed form is $\Pi_{\mathbf{r},\alpha;\mathbf{r}',\beta} = \pi_{z\alpha\beta}$, and $\langle \mathbf{E}\mathbf{H}^*\rangle_{\mathbf{r},\alpha;\mathbf{r}',\beta} = \langle \mathbf{E}_{\mathbf{r},\alpha}\mathbf{H}^*_{\mathbf{r}',\beta}\rangle$, then the ensemble averaged $\mathbf{S}_z$ averaged over one unit cell can be written as

$$\langle \bar{\mathbf{S}}\rangle_z = \frac{1}{2A}\Re\,\mathrm{Tr}\left[\Pi^\top\langle \mathbf{E}\mathbf{H}^*\rangle\right], \tag{16}$$

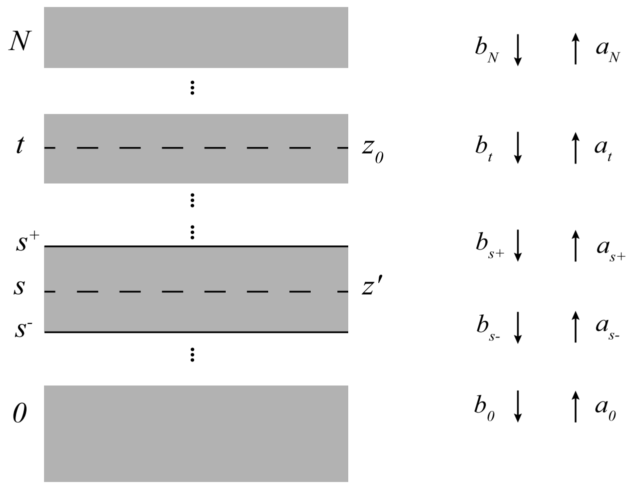

Fig. 3. Left side: illustration of the computational setup, where the emitting layer is index s and a random current source is located at z'. t denotes the layer where Poynting flux is measured, and s^+ and s^- layers, highlighted in solid black lines, are the artificial layers at the topmost and bottommost edges of layer s. These artificial layers are made of the same material as that of layer s but has thickness 0. The Poynting flux is evaluated at position z_0 relative to z_t in the layer indexed t. Right side: the amplitudes of fields in different layers.

where symbol $\top$ stands for the transpose of a tensor, A is the area of the unit cell, and Tr[**O**] denotes the trace of tensor **O**, i.e.

$$\mathrm{Tr}[\mathbf{O}] = \sum_\alpha \int d^2\mathbf{r}\mathbf{O}_{\alpha,\mathbf{r};\alpha,\mathbf{r}}, \tag{17}$$

where α is the polarization and $\mathbf{r} = (x, y)$ is the two-dimensional coordinate vector.

To connect this formula to the current–current correlation function defined in Eq. (12), we define the Green function operators $\mathbf{G}_e$ and $\mathbf{G}_h$ as

$$\mathbf{E} = \mathbf{G}_e\mathbf{J},\ \ \mathbf{H} = \mathbf{G}_h\mathbf{J}. \tag{18}$$

Then the trace in Eq. (16) can further be rewritten in a compact form, i.e.

$$\mathrm{Tr}\left[\Pi^\top \langle \mathbf{EH}^*\rangle\right] = \mathrm{Tr}\left[\Pi^\top \mathbf{G}_e \langle \mathbf{JJ}^*\rangle\, \mathbf{G}_h^\dagger\right]. \tag{19}$$

Thus, the averaged spectral Poynting flux density in one unit cell, due to all the thermal sources in the source layer, can be written as

$$\langle \bar{\mathbf{S}}\rangle_z = \frac{2\epsilon_0}{\pi A}\omega\,\Theta(\omega, T) \int dz'\, \Re\,\mathrm{Tr}\left[\Pi^\top \mathbf{G}_e \Gamma \mathbf{G}_h^\dagger\right]. \tag{20}$$

In MESH, the electromagnetic calculations will be used to evaluate the transmission factor $\Phi(\omega)$ in Eq. (20), defined as:

$$\Phi(\omega) = \frac{2\epsilon_0}{\pi}\omega \int dz'\, \Re\,\mathrm{Tr}\left[\Pi^\top \mathbf{G}_e \Gamma \mathbf{G}_h^\dagger\right]. \tag{21}$$

Here z' is the z-coordinate of the source, and integration is carried out over the source layer. Since all layers are uniform along the z-axis, the dependence of the integrand on z' is only contained in the Green functions. Such integration can be performed analytically using the S-matrix formalism, and will be discussed in detail in Section 3.

3. Evaluation of the transmission factor

In this section, we discuss the evaluation of Eq. (21). We first provide a short review of relevant aspects of the Rigorous Coupled Wave Analysis (RCWA) in Sections 3.1–3.3. In particular, in Section 3.1, we discuss the modal expansion in a single layer. In Section 3.2 we discuss the scattering matrix formalism. In Section 3.3 we discuss the boundary condition associated with the source. Building upon these aspects of RCWA, in Section 3.4 we then give a detailed discussion in the evaluation of Eq. (21), focusing in particular on the integration over the sources.

3.1. Electromagnetic fields in each layer

For the geometry treated in MESH, each layer is uniform along the z-direction with a dielectric function distribution $\epsilon(x, y)$ in the plane. To describe the field in each layer, we consider a dielectric structure that has the same in-plane dielectric distribution $\epsilon(x, y)$, but is infinite in the z direction, and solve for eigenmode field distribution which has the form

$$\phi_n(x, y, z) = \phi_n(x, y)e^{-iq_n z}. \tag{22}$$

Here q_n is the nth eigenvalue with $\Im(q_n) \le 0$, and $\phi_n(x, y)$ denotes the in-plane electric and magnetic field distribution of the corresponding eigenmodes. Such a "forward propagating" eigenmode therefore either propagates or decays along the $+z$ direction. For each of such a mode, there is also a "backward propagating" eigenmode with an eigenvalue $-q_n$, and eigenmode field $\tilde{\phi}_n(x, y)$ can be obtained from $\phi_n(x, y)$ by symmetry considerations, the in-plane electromagnetic fields $e_\parallel$ and $h_\parallel$ in each layer can then be in general written as:

$$\begin{bmatrix} e_\parallel(z) \\ h_\parallel(z) \end{bmatrix} = \mathcal{M} \begin{bmatrix} \mathcal{F}(z) & \mathcal{O} \\ \mathcal{O} & \mathcal{F}(d-z) \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}. \tag{23}$$

Here a and b are column vectors with the elements being the modal amplitudes for forward and backward propagating modes respectively. $\mathcal{F}$ is a diagonal matrix with $\mathcal{F}_{nn} = e^{-iq_n z}$. $\mathcal{O}$ is the zero matrix. Here we focus only on in-plane fields since the Poynting flux as expressed in the trace formula of Eq. (20) only involves in-plane fields. We further denote $\mathcal{F}_l \equiv \mathcal{F}_l(d_l)$ with d_l as the thickness of layer l. The matrix $\mathcal{M} = \begin{bmatrix} \mathcal{M}^e \\ \mathcal{M}^h \end{bmatrix}$ represents the expansion coefficients of the in-plane electromagnetic fields in terms of the modal basis. With an expansion in this form, we have $|\mathcal{F}_{nn}(z)| \le 1$ and $|\mathcal{F}_{nn}(d-z)| \le 1$, which is important for numerical stability. The details for computing the eigenmodes and eigenvalues, and for performing such eigenmode expansion, can be found in Refs. [29,41,42]. For our purpose here, knowing the existence of such an expansion in Eq. (23) is sufficient.

3.2. Scattering matrix

With the eigenmode expansion of Eq. (22), the field in the lth layer is described by the column vector of the modal amplitudes a_l and b_l. In the structures as shown in Fig. 2, the modal amplitudes for the layers l and l' are related by

$$\begin{bmatrix} a_l \\ b_{l'} \end{bmatrix} = \mathcal{S}(l', l) \begin{bmatrix} a_{l'} \\ b_l \end{bmatrix} = \begin{bmatrix} \mathcal{S}_{11} & \mathcal{S}_{12} \\ \mathcal{S}_{21} & \mathcal{S}_{22} \end{bmatrix} \begin{bmatrix} a_{l'} \\ b_l \end{bmatrix}. \tag{24}$$

The details for computing such scattering matrices are standard in various implementations of the RCWA method and can be found in Ref. [29]. For our purpose here, the form of Eq. (24) is sufficient.

3.3. Boundary conditions at the current source

The Green function $\mathbf{G}_e$ and $\mathbf{G}_h$ in Eq. (18) describes the electromagnetic fields generated by an oscillating point dipole source placed at position $(\mathbf{r}', z')$ inside the structure. To compute such Green function, we consider more generally a two-dimensional current distribution

$$\mathbf{J}(\mathbf{r}, z) = \mathbf{J}(\mathbf{r})\delta(z - z'). \tag{25}$$

For a general dielectric tensor ϵ, we denote the inverse of such tensor as

$$\eta = \epsilon^{-1} = \begin{bmatrix} \eta_{xx} & \eta_{xy} & \eta_{xz} \\ \eta_{yx} & \eta_{yy} & \eta_{yz} \\ \eta_{zx} & \eta_{zy} & \eta_{zz} \end{bmatrix}. \tag{26}$$

In this case, following the derivation in Ref. [29], one can obtain a boundary condition at the source z' in real space

$$\begin{pmatrix} e_\parallel \\ h_\parallel \end{pmatrix}_{z'^+} - \begin{pmatrix} e_\parallel \\ h_\parallel \end{pmatrix}_{z'^-} = \begin{pmatrix} \frac{1}{i\omega\epsilon_0}\frac{\partial}{\partial x}\left(\eta_{zx}\mathbf{J}_x + \eta_{zy}\mathbf{J}_y + \eta_{zz}\mathbf{J}_z\right) \\ \frac{1}{i\omega\epsilon_0}\frac{\partial}{\partial y}\left(\eta_{zx}\mathbf{J}_x + \eta_{zy}\mathbf{J}_y + \eta_{zz}\mathbf{J}_z\right) \\ \mathbf{J}_y \\ -\mathbf{J}_x \end{pmatrix}. \tag{27}$$

For the ease of notation, we rewrite Eq. (27) as

$$\begin{pmatrix} e_\parallel \\ h_\parallel \end{pmatrix}_{z'^+} - \begin{pmatrix} e_\parallel \\ h_\parallel \end{pmatrix}_{z'^-} = \mathcal{P}\mathbf{J}, \tag{28}$$

with

$$\mathcal{P} = \begin{pmatrix} \frac{1}{i\omega\epsilon_0}\frac{\partial}{\partial x}\eta_{zx} & \frac{1}{i\omega\epsilon_0}\frac{\partial}{\partial x}\eta_{zy} & \frac{1}{i\omega\epsilon_0}\frac{\partial}{\partial x}\eta_{zz} \\ \frac{1}{i\omega\epsilon_0}\frac{\partial}{\partial y}\eta_{zx} & \frac{1}{i\omega\epsilon_0}\frac{\partial}{\partial y}\eta_{zy} & \frac{1}{i\omega\epsilon_0}\frac{\partial}{\partial y}\eta_{zz} \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}, \tag{29}$$

where here the partial derivatives operate on the product of η and $\mathbf{J}$.

3.4. Volume integration of the current sources

Based on the developments in Sections 3.1–3.3 we now evaluate the transmission factor defined in Eq. (21). We first consider the electromagnetic fields generated by a source at a location z' inside the source layer s. The source layer separates the structure into two disjoint stacks. Among the stack above or below the layer s, the scattering properties between all the layers are already described by the corresponding scattering matrices and are independent of z'. Inside the source layer s, the presence of a current source at z' introduce a discontinuity in the in-plane electromagnetic fields immediately above and below the sources. And moreover, these fields can be related to the fields at the top and bottom boundary of the source layers through the eigenmode expansion in the source layer as described by Section 3.1. Through these relations we can therefore compute the fields for a given source. For the ease of notation, we add two artificial layers s^+ and s^- at top and bottom surfaces of layer s, respectively. These layers are made of the same material as layer s but have zero thickness. In Fig. 3, we schematically represent such artificial layers as thin lines. In addition, all the modal amplitudes required for the following derivation are also labeled in Fig. 3.

Using the S-matrix formalism developed in Section 3.2, we first relate the field amplitudes in layer N and s^+ in Fig. 3 through the following equation

$$\begin{bmatrix} a_N \\ b_{s^+} \end{bmatrix} = \mathcal{S}(s^+, N)\begin{bmatrix} a_{s^+} \\ \mathcal{O} \end{bmatrix} \tag{30}$$

where we impose the outgoing wave boundary condition in layer N by setting $b_N = \mathcal{O}$. From Eq. (30), we obtain the relation between the field amplitudes in layer s^+ as

$$b_{s^+} = \mathcal{S}_{21}(s^+, N)a_{s^+}. \tag{31}$$

Similarly, we connect the field amplitudes in layer s^- with layer 0 as:

$$\begin{bmatrix} a_{s^-} \\ b_0 \end{bmatrix} = \mathcal{S}(s^-, 0)\begin{bmatrix} \mathcal{O} \\ b_{s^-} \end{bmatrix}, \tag{32}$$

where again we impose the outgoing wave boundary condition in layer 0 by setting $a_0 = \mathcal{O}$. From Eq. (32), we have

$$a_{s^-} = \mathcal{S}_{12}(s^-, 0)b_{s^-}. \tag{33}$$

With the modal amplitudes at layers s^+ and s^-, boundary conditions at the source z' using Eqs. (23) and (28) are

$$\mathcal{M}_s\begin{bmatrix} \mathcal{F}_s(z' - d_s)a_{s^+} \\ \mathcal{F}_s(d_s - z')b_{s^+} \end{bmatrix} - \mathcal{M}_s\begin{bmatrix} \mathcal{F}_s(z')a_{s^-} \\ \mathcal{F}_s(-z')b_{s^-} \end{bmatrix} = \mathcal{P}\mathbf{J}. \tag{34}$$

We denote $\mathcal{M}_s^{-1}\mathcal{P}\mathbf{J} = \begin{bmatrix} \mathcal{U}_1 \\ \mathcal{U}_2 \end{bmatrix}$, and rewrite Eq. (34) as two equations:

$$\begin{aligned} \mathcal{F}_s(z' - d_s)a_{s^+} - \mathcal{F}_s(z')a_{s^-} &= \mathcal{U}_1 \\ \mathcal{F}_s(d_s - z')b_{s^+} - \mathcal{F}_s(-z')b_{s^-} &= \mathcal{U}_2 \end{aligned}. \tag{35}$$

Combining Eqs. (31), (33) and Eq. (35), we solve for a_{s^+} as

$$\begin{aligned} a_{s^+} &= \left(\mathbf{I} - \mathcal{F}_s\mathcal{S}_{12}(s^-, 0)\mathcal{F}_s\mathcal{S}_{21}(s^+, N)\right)^{-1}\left[\mathcal{F}_s(d_s - z')\mathcal{U}_1\right. \\ &\quad \left.-\mathcal{F}_s\mathcal{S}_{12}(s^-, 0)\mathcal{F}_s(z')\mathcal{U}_2\right] \\ &= \mathcal{K}_1^{-1}\left[\mathcal{F}_s(d_s - z')\mathcal{U}_1 + \mathcal{K}_2\mathcal{F}_s(z')\mathcal{U}_2\right], \end{aligned} \tag{36}$$

with

$$\mathcal{K}_1 = \mathbf{I} - \mathcal{F}_s\mathcal{S}_{12}(s^-, 0)\mathcal{F}_s\mathcal{S}_{21}(s^+, N),\ \ \mathcal{K}_2 = -\mathcal{F}_s\mathcal{S}_{12}(s^-, 0). \tag{37}$$

Here in Eq. (36) we see that the only z' dependent terms are the matrices $\mathcal{F}$, while the rest of the matrices are independent of z'.

Having computed the modal amplitudes a_s^+ and b_s^+, we then solve the modal amplitudes in the probe layer t. Using the definition of the scattering matrix, we have

$$\begin{bmatrix} a_t \\ b_{s^+} \end{bmatrix} = \mathcal{S}(s^+, t)\begin{bmatrix} a_{s^+} \\ b_t \end{bmatrix}. \tag{38}$$

And moreover, the modal amplitude a_t and b_t can also be related to the modal amplitudes in the Nth layer:

$$\begin{bmatrix} a_N \\ b_t \end{bmatrix} = \mathcal{S}(t, N)\begin{bmatrix} a_t \\ \mathcal{O} \end{bmatrix}, \tag{39}$$

and therefore,

$$b_t = \mathcal{S}_{21}(t, N)a_t. \tag{40}$$

Using Eqs. (38) and (39), we then have:

$$a_t = \mathcal{D}_1 a_{s^+},\ b_t = \mathcal{D}_2 a_{s^+}, \tag{41}$$

with

$$\begin{aligned}\mathcal{D}_1 &= \left(\mathbf{I} - \mathcal{S}_{12}(s^+,t)\mathcal{S}_{21}(t,N)\right)^{-1}\mathcal{S}_{11}(s^+,t),\\ \mathcal{D}_2 &= \mathcal{S}_{21}(t,N)\mathcal{D}_1.\end{aligned} \tag{42}$$

Thus, at position $z = z_0 + z_t$ in the probe layer t, the electric and magnetic fields are evaluated based on Eq. (23) using the corresponding modal amplitudes

$$\begin{aligned}\begin{bmatrix} e_\parallel(z_0+z_t) \\ h_\parallel(z_0+z_t)\end{bmatrix} &= \mathcal{M}_t \begin{bmatrix}\mathcal{F}_t(z_0)a_t \\ \mathcal{F}_t(d_t-z_0)b_t\end{bmatrix} = \mathcal{M}_t\begin{bmatrix}\mathcal{F}_t(z_0)\mathcal{D}_1 a_{s^+} \\ \mathcal{F}_t(d_t-z_0)\mathcal{D}_2 a_{s^+}\end{bmatrix}\\ &= \mathcal{M}_t\begin{bmatrix}\mathcal{F}_t(z_0)\mathcal{D}_1\mathcal{K}_1^{-1}\left[\mathcal{F}_s(d_s-z')\mathcal{U}_1 + \mathcal{K}_2\mathcal{F}_s(z')\mathcal{U}_2\right] \\ \mathcal{F}_t(d_t-z_0)\mathcal{D}_2\mathcal{K}_1^{-1}\left[\mathcal{F}_s(d_s-z')\mathcal{U}_1 + \mathcal{K}_2\mathcal{F}_s(z')\mathcal{U}_2\right]\end{bmatrix}\\ &= \mathcal{M}_t\begin{bmatrix}\mathcal{F}_t(z_0)\mathcal{D}_1 \\ \mathcal{F}_t(d_t-z_0)\mathcal{D}_2\end{bmatrix}\mathcal{K}_1^{-1}\begin{bmatrix}\mathbf{I} & \mathcal{K}_2\end{bmatrix}\\ &\quad\times\begin{bmatrix}\mathcal{F}_s(d_s-z') & \mathcal{O} \\ \mathcal{O} & \mathcal{F}_s(z')\end{bmatrix}\mathcal{M}_s^{-1}\mathcal{P}\mathbf{J}\\ &= \mathcal{R}\mathcal{L}\mathcal{M}_s^{-1}\mathcal{P}\mathbf{J},\end{aligned} \tag{43}$$

where $\mathcal{R} = \begin{bmatrix}\mathcal{R}_e \\ \mathcal{R}_h\end{bmatrix} = \mathcal{M}_t\begin{bmatrix}\mathcal{F}_t(z_0)\mathcal{D}_1 \\ \mathcal{F}_t(d_t-z_0)\mathcal{D}_2\end{bmatrix}\mathcal{K}_1^{-1}\begin{bmatrix}\mathbf{I} & \mathcal{K}_2\end{bmatrix}$ and $\mathcal{L} = \begin{bmatrix}\mathcal{F}_s(d_s-z') & \mathcal{O} \\ \mathcal{O} & \mathcal{F}_s(z')\end{bmatrix}$. Therefore we obtain the matrix representation of the Green functions,

$$\mathbf{G}_e = \mathcal{R}_e\mathcal{L}\mathcal{M}_s^{-1}\mathcal{P},\ \ \mathbf{G}_h = \mathcal{R}_h\mathcal{L}\mathcal{M}_s^{-1}\mathcal{P}. \tag{44}$$

The integrand in Eq. (21) thus reads:

$$\Pi^\top\mathbf{G}_e\Gamma\mathbf{G}_h^\dagger = \Pi^\top\left(\mathcal{R}_e\mathcal{L}\mathcal{M}_s^{-1}\mathcal{P}\right)\Gamma\left(\mathcal{P}^\dagger[\mathcal{M}_s^{-1}]^\dagger\mathcal{L}^\dagger\mathcal{R}_h^\dagger\right), \tag{45}$$

where all the z'-dependent terms are contained in $\mathcal{L}$. To integrate Eq. (45) over z', we denote $\mathcal{Q} = \begin{bmatrix}\mathcal{Q}_{11} & \mathcal{Q}_{12} \\ \mathcal{Q}_{21} & \mathcal{Q}_{22}\end{bmatrix} = \mathcal{M}_s^{-1}\mathcal{P}\Gamma\mathcal{P}^\dagger[\mathcal{M}_s^{-1}]^\dagger$, and focus on the integration for $\mathcal{T} = \mathcal{L}\mathcal{Q}\mathcal{L}^\dagger$, which can be written as

$$\begin{aligned}\mathcal{T} &= \begin{bmatrix}\mathcal{T}_{11} & \mathcal{T}_{12} \\ \mathcal{T}_{21} & \mathcal{T}_{22}\end{bmatrix}\\ &= \begin{bmatrix}\mathcal{F}_s(d_s-z')\mathcal{Q}_{11}\mathcal{F}_s^\dagger(d_s-z') & \mathcal{F}_s(d_s-z')\mathcal{Q}_{12}\mathcal{F}_s^\dagger(z') \\ \mathcal{F}_s(z')\mathcal{Q}_{21}\mathcal{F}_s^\dagger(d_s-z') & \mathcal{F}_s(z')\mathcal{Q}_{22}\mathcal{F}_s^\dagger(z')\end{bmatrix}.\end{aligned} \tag{46}$$

For $\mathcal{T}_{11}$, since $\mathcal{F}_s(d_s-z')$ and $\mathcal{F}_s(z')$ are diagonal, its (m,n)th element is

$$[\mathcal{T}_{11}]_{mn} = e^{-iq_m(d_s-z')}[Q_{11}]_{mn}e^{iq_n^*(d_s-z')}, \tag{47}$$

where $q_i\,(i = m,n)$ is the ith eigenvalue in layer s from RCWA. Hence integrating over z' from 0 to d_s gives

$$\int_0^{d_s} dz'[\mathcal{T}_{11}]_{mn} = [Q_{11}]_{mn}\frac{1-e^{-i(q_m-q_n^*)d_s}}{i(q_m-q_n^*)}. \tag{48}$$

Similarly, for $\mathcal{T}_{12}$, $\mathcal{T}_{21}$ and $\mathcal{T}_{22}$, we have

$$\begin{aligned}\int_0^{d_s} dz'[\mathcal{T}_{12}]_{mn} &= [Q_{12}]_{mn}\frac{e^{iq_n^*d_s} - e^{-iq_m d_s}}{i(q_m+q_n^*)}\\ \int_0^{d_s} dz'[\mathcal{T}_{21}]_{mn} &= [Q_{21}]_{mn}\frac{e^{iq_n^*d_s} - e^{-iq_m d_s}}{i(q_m+q_n^*)}\\ \int_0^{d_s} dz'[\mathcal{T}_{22}]_{mn} &= [Q_{22}]_{mn}\frac{1-e^{-i(q_m-q_n^*)d_s}}{i(q_m-q_n^*)}.\end{aligned} \tag{49}$$

We further define matrix $\mathcal{Y} = \begin{bmatrix}\mathcal{Y}_1 & \mathcal{Y}_2 \\ \mathcal{Y}_2 & \mathcal{Y}_1\end{bmatrix}$ such that

$$\begin{aligned}\mathcal{Y}_{1,(mn)} &= \frac{1-e^{-i(q_m-q_n^*)d_s}}{i(q_m-q_n^*)}\\ \mathcal{Y}_{2,(mn)} &= \frac{e^{iq_n^*d_s} - e^{-iq_m d_s}}{i(q_m+q_n^*)}.\end{aligned} \tag{50}$$

Then the integration of Eq. (45) over z' results in

$$\begin{aligned}\int_0^{d_s}\Pi^\top\mathbf{G}_e\Gamma\mathbf{G}_h^\dagger\,dz' &= \Pi^\top\mathcal{R}_e\left[\mathcal{Q}\circ\mathcal{Y}\right]\mathcal{R}_h^\dagger\\ &= \Pi^\top\mathcal{R}_e\left[(\mathcal{M}_s^{-1}\mathcal{P}\Gamma\mathcal{P}^\dagger[\mathcal{M}_s^{-1}]^\dagger)\circ\mathcal{Y}\right]\mathcal{R}_h^\dagger,\end{aligned} \tag{51}$$

where $\circ$ is Hadamard product. We also note that the results in Eqs. (50) and (51) can also be applied to the general cases where the emitting layer or the probe layer is semi-infinite. MESH implements Eq. (51) to compute the heat transfer.

4. Supported patterns

In MESH, the supported 2D patterns are rectangle, circle, ellipse and polygon. In addition, a layer can contain multiple patterns, and these patterns can be nested but cannot have intersections. In computing the Fourier transform of the dielectric function, the $\epsilon(\mathbf{r})$ is expressed in the form of

$$\epsilon(\mathbf{r}) = \epsilon_b + \sum_j\left(\epsilon_j - \epsilon_{\supset j}\right)I_j(\mathbf{r}), \tag{52}$$

where ϵ_b is the background dielectric constant, the summation is over all the patterns in the layer, ϵ_j is the dielectric constant of shape j, and $\epsilon_{\supset j}$ is the dielectric constant of the shape that immediately contains j, and I_j is the indicator function for the pattern j. Here in MESH, we focus on the case where the zth component of the dielectric function can be decomposed from in-plane components, i.e.

$$\epsilon = \begin{bmatrix}\epsilon_{xx} & \epsilon_{xy} & 0 \\ \epsilon_{yx} & \epsilon_{yy} & 0 \\ 0 & 0 & \epsilon_{zz}\end{bmatrix}, \tag{53}$$

and an efficient algorithm is implemented in order to find the set $\supset j$ for any pattern j. Since even complicated geometries can be decomposed into or approximated by combinations of these 4 supported geometries, MESH is therefore capable to compute almost any geometry pattern.

Eq. (51) is expressed in a form that is independent of basis. In MESH, however, we evaluate Eq. (51) on a plane wave basis with wave vectors $\{\mathbf{k}+\mathbf{G}\}$, where $\mathbf{k}$ is the Bloch wave vector in the first Brillouin zone of the lattice, and the $\mathbf{G}$'s are the reciprocal lattice vectors. With these reciprocal lattice vectors, the Fourier transforms of $\gamma_{\alpha\beta}$ and $\epsilon_{\alpha\beta}$ can be evaluated in closed forms. We also note that for more complicated geometries, usually one needs to use more $\mathbf{G}$'s in order to achieve convergence. In addition, for better numerical conditioning, in the implementation we rescale $\mathbf{E}\epsilon_0 c \to \mathbf{E}$ and $\omega/c \to \omega$.

In terms of the choice of reciprocal lattice vector $\mathbf{G}$, MESH implements both "circular truncation", which include all $\mathbf{G}$ vectors in a circular region centered at $\mathbf{G} = 0$ [41,43], and "parallelogramic truncation", which include $\mathbf{G}$ vectors in a parallelogram around $\mathbf{G} = 0$ [41,43]. MESH only requires the user to input the upper bound for the number of $\mathbf{G}$ and will automatically perform the $\mathbf{G}$ vector selection.

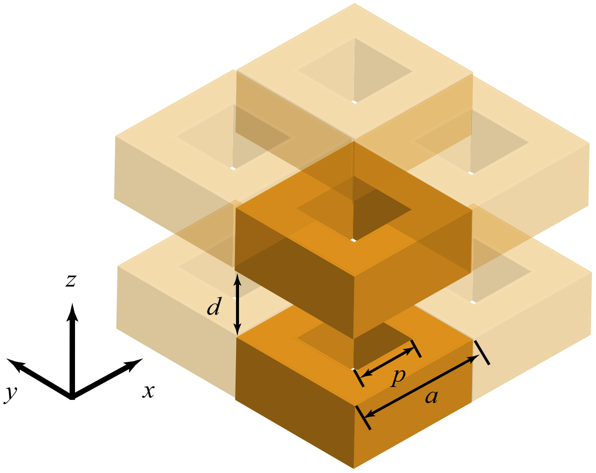

Fig. 4. The unit cell of the structure described by the sample script. Two semi-infinite patterned doped silicon layers are separated by vacuum gap d. The unit cell has width and length a, and the vacuum hole within the layer has width and length p.

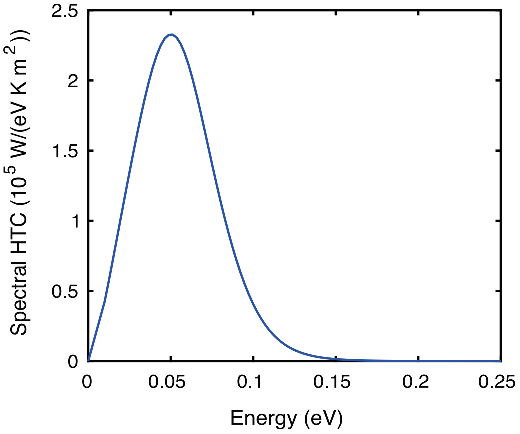

Fig. 5. Spectral heat transfer coefficient (HTC) for the structure shown in Fig. 4, as obtained from the sample script.

5. User scripting and programming interface

MESH is embedded within the Lua and Python scripting languages allowing users to easily construct structures and perform the computation, similar to S^4 [41]. Using Lua and Python as thin wrappers provide not only high-level intuitive syntax for users but also the convenience to use both objective-oriented and functional programming paradigms for programmers. In addition, MESH also provides a Lua wrapper for MPI to facilitate parallelization.

The user manual and a few examples illustrating the usage of MESH are presented in detail at https://kfrancischen.github.io/MESH/. Here we will describe a simple example with its Lua script shown below, which can be used to reproduce the spectral heat transfer coefficient for a structure in Fig. 2(b) in Ref. [37]. In this example, the structure consists of two identical semi-infinite patterned heavily-doped silicon layers (Fig. 4), separated by a $d =$ 20 nm vacuum gap. Each layer consists of a square lattice of air holes in silicon. The periodicities in x and y directions are $a =$ 50 nm, and the filling ratio $f = p^2/a^2$ is 0.98, where p is the size of the air holes. In Ref. [37], the spectral heat transfer coefficient with $f = 0.98$ is obtained using scattering approach, while here we use the approached implemented in MESH to produce the same spectral heat transfer coefficient.

We now go through in detail the Lua script that is used for such calculation. In the beginning set of the codes, we initiate by calling the function `Constants()`, which provides commonly used physics constants. In order to compute the spectral heat transfer coefficient, which is defined in Eq. (1) in [37], one needs to evaluate the derivative of $\Theta(\omega, T)$ over T at a given angular frequency ω. Therefore in the script, we define a function named `thetaDerivative` to compute such derivative. Then we setup the geometry by initiating a `SimulationPattern` object to indicate that we are simulating a structure with two-dimensional in-plane periodicity. In addition to `SimulationPattern`, MESH also provides `SimulationPlanar` and `SimulationGrating` for planar geometry and 1D grating geometry simulations, respectively. The reason to distinguish simulation by the dimensionality of the periodicity is to restrict function usage and to improve readability of the scripts. After initiating a `SimulationPattern` object, we set the lattice vectors of the geometry as well as the number of reciprocal lattice vectors to use.

In the next set of codes in the script, we add materials to the structure. Optical constants for each material need to be put into a text file, where in this case, two files are needed for silicon and vacuum, respectively. For each file, the first column indicates the angular frequency ω in the unit of rad/s, and the second and third columns are the real and imaginary parts of the relative dielectric constants. For example, here are the top five lines of the data file `Si.txt`:

```
1.519268e+13  -1.185134e+02  8.008125e+02
1.887575e+13  -1.167153e+02  6.356555e+02
2.255882e+13  -1.145941e+02  5.230897e+02
2.624189e+13  -1.121841e+02  4.410925e+02
2.992497e+13  -1.095216e+02  3.784913e+02
```

MESH also supports a few different types of anisotropic dielectrics: for the dielectric function in the form of a diagonal matrix, for each line of this text file there are 6 columns for the real and imaginary parts of ϵ_x, ϵ_y, ϵ_z; for the dielectric tensor in the form of Eq. (53), there are 10 columns for the real and imaginary parts of different matrix elements.

The following set of codes in the script add the layers to the system from bottom to the top, and then set the emitting layer which is the source of the radiation, and also the probe layer where the transmission factor in Eq. (21) is evaluated. In MESH, one can set multiple source layers. We also restrict that the transmission factor in Eq. (21) is positive, and under this convention, all source layers have to be below the probe layer. Materials and layers are referred by their names. All layers default to be unpatterned uniform layers unless patterns are specified.

Finally, one sets the number of threads to be used in an OpenMP parallelization and the integration discretization over k_x and k_y, and then perform the integration. Since here the geometry is highly symmetric, we can restrict the integration over $\mathbf{k}$ in the irreducible first Brillouin zone by calling functions with suffix `sym` indicating symmetry. The $\Phi(\omega)$ in Eq. (21) can be retrieved after the integration over k_x and k_y.

In Fig. 5 we show the result obtained from the above script, with in total 440 **G** vectors, we obtain a converged result that exactly

```
-- The curve with f=0.98 in Fig.2 (b) in
-- "Enhancing Near-Field Radiative Heat Transfer with Si-based Metasurfaces",
-- Phys. Rev. Lett. 118, 203901, 2017

--------------------loading predefined constants--------------------
constants = Constants();

------------------defining the derivative of Theta------------------
function thetaDerivative(omega, T)
  local theta = constants.h_bar * omega
    / (math.exp(constants.h_bar * omega/constants.k_B/T) - 1);
  return math.pow(theta, 2) * math.exp(constants.h_bar * omega/constants.k_B/T)
    /constants.k_B / math.pow(T, 2);
end

------------initializing a 2D pattern simulation object-------------
s = SimulationPattern.new();
s:SetLattice(50e-9,   -- length of the lattice vector in x direction
       50e-9,   -- length of the other lattice vector
       90)       -- the angle between two lattice vectors, in degree
s:SetNumOfG(440)       -- number of G vectors

--------------------------adding materials--------------------------
s:AddMaterial("Si", "Si.txt")  -- name and the file containing its dielectric constants
s:AddMaterial("Vacuum", "Vacuum.txt")

----------------------------adding layers---------------------------
f = 0.98
width = math.sqrt(f * 50e-9 * 50e-9)
s:AddLayer("SiBottom", 0, "Si")
s:SetLayerPatternRectangle("SiBottom",       --layer name
       "Vacuum",      -- material for the rectangle hole
       {25e-9, 25e-9}, -- center of the rectangle
       0,             -- rotation angle of the rectangle, in degree
       {width, width}) -- widths of the two sides

s:AddLayer("VacGap", 20e-9, "Vacuum")
s:AddLayerCopy("SiTop", "SiBottom")

----------------setting the source layer and probe layer----------------
s:SetSourceLayer("SiBottom")
s:SetProbeLayer("VacGap")

--------------------setting the integration parameter-------------------
s:SetThread(60)        -- using 60 cores in openmp
s:SetKxIntegralSym(50) -- using 50 points in positive x direction in the first BZ
s:SetKyIntegralSym(50) -- using 50 points in positive y direction in the first BZ
s:InitSimulation()
s:IntegrateKxKy()

-----------------outputing heat transfer coefficient----------------
omega = s:GetOmega()
phi = s:GetPhi()
for i = 1,s:GetNumOfOmega() do
    E = constants.h_bar * omega[i] / constants.q;
    spectrum = thetaDerivative(omega[i], 300) * phi[i] * constants.q / constants.h_bar;
    print(E.."\t"..spectrum/1e5);
end
```

matches the result in Ref. [37]. On the Comet cluster of the San Diego Supercomputing Center (SDSC), by using 600 cores, each of which consists of an Intel Xeon E5-2680v3 processor, the runtime for the MPI version of the above script with in total 100 ω values and 2500 (k_x, k_y) values for each ω is around 6 h.

5.1. Parallelization

Parallelizations with MPI (Message Passing Interface) and OpenMP are fully supported in MESH. In the parallel implementations, the evaluation for different ω, k_x, k_y values are carried out in parallel. The results are then summed to compute the transmission factor. To run MESH on a distributed memory machine, we use MPI. The Lua MPI wrapper within MESH provides same functionalities as the native MPI with similar syntaxes. The rank and size of the MPI world communicator as well as common functions are all available to Lua scripts. An MPI example is available at https://kfrancischen.github.io/MESH/Examples/MPI/. To execute on a shared memory machine, MESH provides OpenMP support — the number of threads can be specified in function `SetNumThread()`.

In principle, one can also utilize the Graphic Processing Unit (GPU) for the parallelization in order to speed up matrix computation, we have not implemented it and will leave it as one of the future work.

5.2. Convergence and computational cost

The accuracy of computing $\Phi(\omega)$ depends on the number of reciprocal lattice vectors **G** (denoted as N) used in the Fourier transform. However, with an increase in N, the memory and time for computing $\Phi(\omega, k_x, k_y)$, which is the integrand of $\Phi(\omega)$, will significantly increase. For example with $N = 1000$, one S-matrix

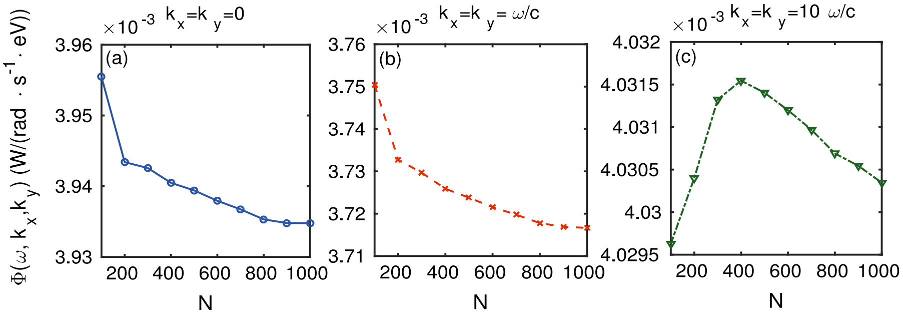

Fig. 6. Convergence test for $\Phi(\omega, k_x, k_y)$ for geometry described in the sample script. The number of **G** varies from 100 to 1000 for three cases: (a) $k_x = k_y = 0$, (b) $k_x = k_y = \frac{\omega}{c}$ and (c) $k_x = k_y = 10\frac{\omega}{c}$ where $\omega = 1.519268 \times 10^{13}$ rad/s.

has size 4000 × 4000 and requires 256 MB for storage. Therefore, in practice one would choose a reasonable value for N that can guarantee convergence without wasting extra computational resources. In MESH one can check the convergence of $\Phi(\omega, k_x, k_y)$ a particular set of (ω, k_x, k_y) as a function of N. The computation cost of MESH also scales almost linearly with the number of layers, since for each extra layer one needs to evaluate one more scattering matrix using the formalism in Eq. (24).

Here we test the convergence of $\Phi(\omega, k_x, k_y)$ for the geometry shown in Fig. 4 with $\omega = 1.519268 \times 10^{13}$ rad/s. At this angular frequency, the relative dielectric constant of silicon is $\epsilon_{\text{Si}} = -118.5134 + 800.8125i$. The results for $(k_x, k_y) = (0, 0), (\omega/c, \omega/c), (10\omega/c, 10\omega/c)$ are shown in Fig. 6. For the three cases considered here, as N changes from 100 to 1000, the changes of the corresponding $\Phi(\omega, k_x, k_y)$ values are within 1%, which indicates a good convergence, even though the inverse Fourier factorization rule for $\epsilon(\mathbf{r})$ [43] has not been adopted.

The convergence of RCWA methods when applying to metallic materials has been discussed in the literature [43–45]. Due to the negativity of dielectric constant of the metal and high contrast between the dielectric function at the interface, the Gibbs phenomenon introduces ringing in the real space reconstruction of the dielectric function, resulting in possibly fictitious interfaces where the dielectric function crosses zero. The implementation of MESH follows the standard procedure of RCWA [29], and hence the speed of convergence and numerical stability depend on the detailed implementation of RCWA. In particular, since RCWA is known to have difficulty for low-loss metals, one should use MESH with caution as well when dealing with low-loss metals. In the example shown in the previous section, even though heavily-doped Si has negative dielectric constants within a certain range of frequencies, MESH can achieve convergence as has been discussed in Section 5.2. In general, however, especially for systems with negative dielectric constants, when using MESH, it is always recommended that the convergence of $\Phi(\omega, k_x, k_y)$ is checked to decide the optimal N value.

6. Concluding remarks

To summarize, in this paper, we have introduced MESH, a flexible open source software for heat flux calculation combining RCWA and fluctuational electromagnetics. Different from the scattering approach that has been widely used in previous literatures, MESH calculates far-field and near-field heat transfer by directly computing the Green functions for the electromagnetic fields resulting from random current sources, which is particularly suitable to treat the scenario of many-body heat transfer. In addition, in MESH we also implemented an efficient algorithm for spatial integration of such random current sources. For large-scale computation of complicated geometries, MESH provides a flexible interface for both OpenMP and MPI parallelization, thus can be easily executed on personal computers as well as computer clusters. For users and programmers, MESH provides a simple scripting interface. MESH can be a useful tool for both theoretical analysis and experimental design of nano-scale devices for the control of far-field thermal radiation and near-field heat transfer.

We have validated the convergence and the correctness of MESH by reproducing existing published results. In addition, MESH is bundled with a few other examples with different structure dimensions and material combinations, and it has already been used in a few works covering different applications of thermal radiation, including thermophotovoltaic system optimization for waste heat recovery [46], and near-field electroluminescent refrigeration device design [47].

Acknowledgment

This work was supported by the DOE “Light-Material Interactions in Energy Conversion” Energy Frontier Research Center under grant DE-SC0001293.

References

[1] S. Fan, Joule 1 (2) (2017) 264–273. http://dx.doi.org/10.1016/j.joule.2017.07.012.
[2] A. Narayanaswamy, G. Chen, Appl. Phys. Lett. 82 (20) (2003) 3544–3546. http://dx.doi.org/10.1063/1.1575936.
[3] M. Laroche, R. Carminati, J.-J. Greffet, J. Appl. Phys. 100 (6) (2006) 063704. http://dx.doi.org/10.1063/1.2234560.
[4] K. Park, S. Basu, W. King, Z. Zhang, J. Quant. Spectrosc. Radiat. Transfer 109 (2) (2008) 305–316, the Fifth International Symposium on Radiative Transfer. http://dx.doi.org/10.1016/j.jqsrt.2007.08.022.
[5] B. Zhao, L. Wang, Y. Shuai, Z.M. Zhang, Int. J. Heat Mass Transfer 67 (2013) 637–645. http://dx.doi.org/10.1016/j.ijheatmasstransfer.2013.08.047.
[6] Z.J. Coppens, W. Li, D.G. Walker, J.G. Valentine, Nano Lett. 13 (3) (2013) 1023–1028 pMID: 23437919. http://dx.doi.org/10.1021/nl304208s.
[7] W. Li, J. Valentine, Nano Lett. 14 (6) (2014) 3510–3514 pMID: 24837991. http://dx.doi.org/10.1021/nl501090w.
[8] K. Chen, P. Santhanam, S. Sandhu, L. Zhu, S. Fan, Phys. Rev. B 91 (2015) 134301. http://dx.doi.org/10.1103/PhysRevB.91.134301.
[9] K. Chen, P. Santhanam, S. Fan, Phys. Rev. Appl. 6 (2016) 024014. http://dx.doi.org/10.1103/PhysRevApplied.6.024014.
[10] X. Liu, Z.M. Zhang, Phys. Rev. Appl. 5 (2016) 034004. http://dx.doi.org/10.1103/PhysRevApplied.5.034004.
[11] W. Li, Y. Shi, K. Chen, L. Zhu, S. Fan, ACS Photonics 4 (4) (2017) 774–782. http://dx.doi.org/10.1021/acsphotonics.7b00089.
[12] W. Li, Y. Shi, K. Chen, L. Zhu, S. Fan, 2017 IEEE 60th International Midwest Symposium on Circuits and Systems, MWSCAS, 2017, pp. 847–850. http://dx.doi.org/10.1109/MWSCAS.2017.8053056.
[13] A. Kittel, W. Müller-Hirsch, J. Parisi, S.-A. Biehs, D. Reddig, M. Holthaus, Phys. Rev. Lett. 95 (2005) 224301. http://dx.doi.org/10.1103/PhysRevLett.95.224301.
[14] E. Saïdi, B. Samson, L. Aigouy, S. Volz, P. Löw, C. Bergaud, M. Mortier, Nanotechnology 20 (11) (2009) 115703. http://stacks.iop.org/0957-4484/20/i=11/a=115703.
[15] D. Polder, M. Van Hove, Phys. Rev. B 4 (1971) 3303–3314. http://dx.doi.org/10.1103/PhysRevB.4.3303.

[16] J.B. Pendry, J. Phys.: Condens. Matter 11 (35) (1999) 6621. http://stacks.iop.org/0953-8984/11/i=35/a=301.

[17] A.I. Volokitin, B.N.J. Persson, Rev. Modern Phys. 79 (2007) 1291–1329. http://dx.doi.org/10.1103/RevModPhys.79.1291.

[18] E. Rousseau, A. Siria, G. Jourdan, S. Volz, F. Comin, J. Chevrier, J.-J. Greffet, Nature Photon. 3 (9) (2009) 514–517. http://dx.doi.org/10.1038/nphoton.2009.144.

[19] R. Guérout, J. Lussange, F.S.S. Rosa, J.-P. Hugonin, D.A.R. Dalvit, J.-J. Greffet, A. Lambrecht, S. Reynaud, Phys. Rev. B 85 (2012) 180301. http://dx.doi.org/10.1103/PhysRevB.85.180301.

[20] R. St-Gelais, L. Zhu, S. Fan, M. Lipson, Nature Nanotechnol. 11 (6) (2016) 515–519. http://dx.doi.org/10.1038/nnano.2016.20. letter.

[21] M.T. Homer Reid, S.G. Johnson, Efficient Computation of Power, Force, and Torque in BEM Scattering Calculations, ArXiv e-prints, arXiv:1307.2966.

[22] http://github.com/homerreid/scuff-EM.

[23] A.W. Rodriguez, M.T.H. Reid, S.G. Johnson, Phys. Rev. B 86 (2012) 220302. http://dx.doi.org/10.1103/PhysRevB.86.220302.

[24] A.W. Rodriguez, M.T.H. Reid, S.G. Johnson, Phys. Rev. B 88 (2013) 054305. http://dx.doi.org/10.1103/PhysRevB.88.054305.

[25] https://github.com/HomerReid/buff-em.

[26] A.G. Polimeridis, M.T.H. Reid, W. Jin, S.G. Johnson, J.K. White, A.W. Rodriguez, Phys. Rev. B 92 (2015) 134202. http://dx.doi.org/10.1103/PhysRevB.92.134202.

[27] C.R. Otey, L. Zhu, S. Sandhu, S. Fan, J. Quant. Spectrosc. Radiat. Transfer 132 (2014) 3–11, special Issue on Micro- and Nano-Scale Radiative Transfer. http://dx.doi.org/10.1016/j.jqsrt.2013.04.017.

[28] A. Didari, M.P. Mengü, J. Quant. Spectrosc. Radiat. Transfer (2017). http://dx.doi.org/10.1016/j.jqsrt.2017.03.010.

[29] D.M. Whittaker, I.S. Culshaw, Phys. Rev. B 60 (1999) 2610–2618. http://dx.doi.org/10.1103/PhysRevB.60.2610.

[30] M. Liscidini, D. Gerace, L.C. Andreani, J.E. Sipe, Phys. Rev. B 77 (2008) 035324. http://dx.doi.org/10.1103/PhysRevB.77.035324.

[31] G. Bimonte, Phys. Rev. A 80 (2009) 042102. http://dx.doi.org/10.1103/PhysRevA.80.042102.

[32] R. Messina, M. Antezza, Phys. Rev. A 84 (2011) 042102. http://dx.doi.org/10.1103/PhysRevA.84.042102.

[33] J. Lussange, R. Guérout, F.S.S. Rosa, J.-J. Greffet, A. Lambrecht, S. Reynaud, Phys. Rev. B 86 (2012) 085432. http://dx.doi.org/10.1103/PhysRevB.86.085432.

[34] J. Dai, S.A. Dyakov, M. Yan, Phys. Rev. B 93 (2016) 155403. http://dx.doi.org/10.1103/PhysRevB.93.155403.

[35] J. Dai, S.A. Dyakov, S.I. Bozhevolnyi, M. Yan, Phys. Rev. B 94 (2016) 125431. http://dx.doi.org/10.1103/PhysRevB.94.125431.

[36] R. Messina, A. Noto, B. Guizal, M. Antezza, Phys. Rev. B 95 (2017) 125404. http://dx.doi.org/10.1103/PhysRevB.95.125404.

[37] V. Fernández-Hurtado, F.J. García-Vidal, S. Fan, J.C. Cuevas, Phys. Rev. Lett. 118 (2017) 203901. http://dx.doi.org/10.1103/PhysRevLett.118.203901.

[38] H. Chalabi, E. Hasman, M.L. Brongersma, Phys. Rev. B 91 (2015) 014302. http://dx.doi.org/10.1103/PhysRevB.91.014302.

[39] H. Chalabi, E. Hasman, M.L. Brongersma, Phys. Rev. B 91 (2015) 174304. http://dx.doi.org/10.1103/PhysRevB.91.174304.

[40] L. Landau, E. Lifshitz, L. Pitaevskii, Course of Theoretical Physics, Vol. 9, Part 2, Pergamon, New York, 1980.

[41] V. Liu, S. Fan, Comput. Phys. Comm. 183 (10) (2012) 2233–2244. http://dx.doi.org/10.1016/j.cpc.2012.04.026.

[42] B. Caballero, A. García-Martín, J.C. Cuevas, Phys. Rev. B 85 (2012) 245103. http://dx.doi.org/10.1103/PhysRevB.85.245103.

[43] L. Li, J. Opt. Soc. Am. A 14 (10) (1997) 2758–2767. http://dx.doi.org/10.1364/JOSAA.14.002758.

[44] N.M. Lyndin, O. Parriaux, A.V. Tishchenko, J. Opt. Soc. Am. A 24 (12) (2007) 3781–3788. http://dx.doi.org/10.1364/JOSAA.24.003781.

[45] L. Li, G. Granet, J. Opt. Soc. Am. A 28 (5) (2011) 738–746. http://dx.doi.org/10.1364/JOSAA.28.000738.

[46] B. Zhao, K. Chen, S. Buddhiraju, G. Bhatt, M. Lipson, S. Fan, Nano Energy 41 (2017) 344–350. http://dx.doi.org/10.1016/j.nanoen.2017.09.054.

[47] K. Chen, T.P. Xiao, P. Santhanam, E. Yablonovitch, S. Fan, J. Appl. Phys. 122 (14) (2017) 143104. http://dx.doi.org/10.1063/1.5007712.