Journal of the Optical Society of America B
OPTICAL PHYSICS

Implementation of anisotropic subpixel smoothing in a transfer matrix method program

Brittany R. Hoard,[1,2,*] Sarun Atiganyanun,[1] and Sang Eon Han[1,3]

[1]The University of New Mexico Nanoscience and Microsystems Engineering Program, University of New Mexico, MSC01 1150, 1 University of New Mexico, Albuquerque, New Mexico 87131, USA
[2]Currently at School of Technology, Western Governors University, 4001 S 700 East, #700, Salt Lake City, Utah 84107, USA
[3]Department of Chemical and Biological Engineering, University of New Mexico, MSC01 1120, 1 University of New Mexico, Albuquerque, New Mexico 87131, USA
*hoardbrittany@gmail.com

Received 29 January 2026; revised 13 May 2026; accepted 14 May 2026; posted 20 May 2026; published 18 June 2026

The transfer matrix method (TMM) is a technique that uses Maxwell's equations and a series of matrix operations to calculate the transmission, reflection, and absorptance of electromagnetic waves propagating through a layered structure. We implement, a novel, to our knowledge, anisotropic subpixel smoothing scheme within a TMM program to improve the accuracy of the output. In this study, we use this modified TMM program to calculate the transmission data of two structures: a c-Si 2D cylinder within a silica substrate and an Ag 2D cylinder within a silica substrate. We observe a significant increase in accuracy for both structures for most, but not all, optical wavelengths. In addition, we find a substantial improvement of the rate of convergence over that of the conventional scheme for both structures, and a modest increase in calculation time.

https://doi.org/10.1364/JOSAB.592032

1. INTRODUCTION

The transfer matrix method (TMM) is a formalism used to calculate the transmission and reflection characteristics of electromagnetic waves propagating through a structure of material, or "slab" [1]. The slab may or may not be spatially homogeneous; it may consist of multiple layers of different materials. Although the Fresnel equations can be used to calculate the transmission and reflection of an EM wave that is incident to a single interface between two optical mediums, the interactions of the EM wave with the mediums are considerably more complex when multiple interfaces are involved, since there are multiple partial transmissions and partial reflections. A major strength of the TMM is its ability to handle this more complex scenario involving multiple interfaces. With the TMM, Maxwell's equations and their boundary conditions are applied to the calculation of the electric field across the boundaries between the different optical mediums in the structure. One layer of the structure is represented by a matrix, with a multilayered structure being represented by the product of each layer matrix. Initially, the electric field is calculated at one side of the structure, and a succession of matrix operations is used to calculate the field at the other side of the structure [2,3]. Once the field is fully calculated, the matrix is transformed into the transmission and reflection coefficients, which comprise the final output of the TMM calculation.

Computer programs that utilize the TMM are effective at calculating the transmission and reflection characteristics of layered periodic structures. Such programs are also useful for obtaining photonic dispersion surfaces because they calculate a set of allowed wave vectors at a fixed frequency. Moreover, the TMM is advantageous when the structures involve dielectric functions that exhibit a strong dependence on frequency. Because the TMM is a frequency-domain method, the dispersion of the dielectric function does not need to be modeled by an approximate function as in the finite-difference time-domain (FDTD) method.

In order to efficiently calculate the transmission and reflection coefficients as well as the photonic dispersion surfaces of a complex metallic or dielectric structure, Bell *et al.* [2] created a program titled "PHOTON" in the FORTRAN 77 language that utilizes the TMM to model EM waves incident on the structure. In this program, Maxwell's equations are discretized using a cubic discretization mesh to allow the integration of the electric and magnetic fields through the unit cell of a user-specified structure, and the equations are discretized on a uniform grid. It should be noted that PHOTON does not utilize perfectly matched layers. The PHOTON program diagonalizes the real-space transfer matrix to enable the calculation of the band structure and transforms the transfer matrix into a plane wave basis in order to calculate the transmission and reflection coefficients. However, due to the limitations of the cubic discretization mesh and the simple subpixel smoothing method

used to calculate a scalar discretized dielectric function for each pixel, the PHOTON program and its methods suffer from inaccuracy when used to model discontinuous dielectric structures [4]. It should be noted that an analytic treatment of the equations described in Section 3.B in this paper can mitigate this inaccuracy, as demonstrated in [5,6]. However, the PHOTON program used in this study is a numerical implementation rather than an analytical one.

The importance and impact of our study lie in its addressing of the inadequacies of the scalar subpixel smoothing method currently employed by the PHOTON program, and the novel implementation of an anisotropic subpixel smoothing method in said program. Although we have chosen to implement this method in the PHOTON program, the method could potentially be implemented in software packages other than PHOTON.

2. OBJECTIVES

The objective of our study is to improve the accuracy of the PHOTON program by updating its subpixel smoothing method. To this end, we perform a novel implementation of an anisotropic subpixel smoothing scheme that calculates a more suitable dielectric function ϵ for pixels that contain interfaces between different materials. This smoothing scheme was initially developed by [4] to improve the accuracy of finite-difference time-domain (FDTD) methods and was employed by [7] in their development of a vectorial algorithm to compute the definite-frequency eigenstates of Maxwell's equations in arbitrary periodic dielectric structures. It should be noted that, in the context of this study, the term "subpixel" is taken to mean "within a pixel" and refers to the calculations that are performed within each pixel. The goal of [4] in developing this scheme was to solve two problems that arise with a discretized dielectric function ϵ:

1. A uniform discretization mesh makes it difficult to model structures with a complex geometry, and
2. The discretized ϵ is not an accurate representation of the actual dielectric function at interfaces between different values of ϵ [4].

Problem 1 can be mitigated by any smoothing scheme that assigns an effective ϵ to each pixel, which can vary continuously by geometry. In fact, a commonly used and simple smoothing scheme addresses this problem by calculating the volumetric average of ϵ within each pixel; the smoothed ϵ remains a scalar value. However, the perturbed geometry that results from this simple smoothing method may increase the error rather than decrease it. In addition, this simple smoothing scheme does not adequately address Problem 2, as the accuracy of the resulting ϵ is questionable, especially for structures in which the dielectric contrast between the materials is high, as in metal–dielectric interfaces [4,8]. The rate of convergence with the grid resolution can be negatively affected by inaccuracies caused by ϵ interfaces [9].

3. METHODS

A. Average Dielectric Tensor

The anisotropic subpixel smoothing scheme developed by Farjadpour *et al.* [4] is based on effective medium theory and has zero first-order error in Maxwell's equations. In this scheme, for a pixel containing an interface of two isotropic materials, an anisotropic average of ϵ, which is in the form of an average dielectric tensor, is calculated according to the orientation of the interface [4]. This average dielectric tensor uses the values $\langle\epsilon\rangle$ and $\langle\epsilon^{-1}\rangle^{-1}$, which are the volumetric averages of ϵ for the electric fields parallel and perpendicular to the interface, respectively. These volumetric averages are calculated according to Eqs. (1) and (2):

$$\langle\epsilon\rangle = \epsilon_1 f_1 + \epsilon_2 f_2, \tag{1}$$

$$\langle\epsilon^{-1}\rangle^{-1} = (\epsilon_1^{-1} f_1 + \epsilon_2^{-1} f_2)^{-1}, \tag{2}$$

where ϵ_1 and ϵ_2 are the dielectric constants for the first and second materials, respectively, and f_1 and f_2 are the filling fractions for the first and second materials within the pixel, respectively.

Once we have the volumetric averages $\langle\epsilon\rangle$ and $\langle\epsilon^{-1}\rangle^{-1}$ for the pixel, we can calculate the inverse dielectric tensor for the pixel according to Eq. (3):

$$\tilde{\epsilon}^{-1} = P\langle\epsilon^{-1}\rangle + (1-P)\langle\epsilon\rangle^{-1}, \tag{3}$$

where $P_{ij} = n_i n_j$ is the projection tensor onto the normal, and $\langle\rangle$ indicates an average over the pixel.

In order to implement this anisotropic subpixel smoothing scheme into the PHOTON program, we modified the program so that the smoothed ϵ at each pixel is a tensor rather than a scalar and is consistent with Eq. (3). We obtained the following dielectric tensor $\tilde{\epsilon}$ for each pixel:

$$\begin{vmatrix} (n_y{}^2 + n_z{}^2)\langle\epsilon\rangle + n_x{}^2\langle\epsilon^{-1}\rangle^{-1} & (\langle\epsilon^{-1}\rangle^{-1} - \langle\epsilon\rangle)n_x n_y & (\langle\epsilon^{-1}\rangle^{-1} - \langle\epsilon\rangle)n_x n_z \\ (\langle\epsilon^{-1}\rangle^{-1} - \langle\epsilon\rangle)n_x n_y & (n_x{}^2 + n_z{}^2)\langle\epsilon\rangle + n_y{}^2\langle\epsilon^{-1}\rangle^{-1} & (\langle\epsilon^{-1}\rangle^{-1} - \langle\epsilon\rangle)n_y n_z \\ (\langle\epsilon^{-1}\rangle^{-1} - \langle\epsilon\rangle)n_x n_z & (\langle\epsilon^{-1}\rangle^{-1} - \langle\epsilon\rangle)n_y n_z & (n_x{}^2 + n_y{}^2)\langle\epsilon\rangle + n_z{}^2\langle\epsilon^{-1}\rangle^{-1} \end{vmatrix}, \tag{4}$$

where n_x, n_y, and n_z are the surface normals in the x, y, and z directions [10].

B. E and H Fields

The E and H field equations within the updated PHOTON program also needed to be modified to take into account that the dielectric constant at each pixel is a tensor, not a scalar. To obtain these equations, we begin with the discrete forms of Maxwell's equations on a cubic lattice:

$$\frac{1}{c}[E_y(\boldsymbol{r}) - E_y(\boldsymbol{r}+\boldsymbol{c})] - \frac{1}{b}[E_z(\boldsymbol{r}) - E_z(\boldsymbol{r}+\boldsymbol{b})] = c\mu\frac{\omega^2}{{c_0}^2}H_x'(\boldsymbol{r}), \tag{5}$$

$$\frac{1}{a}[E_z(\boldsymbol{r}) - E_z(\boldsymbol{r}+\boldsymbol{a})] - \frac{1}{c}[E_x(\boldsymbol{r}) - E_x(\boldsymbol{r}+\boldsymbol{c})] = c\mu\frac{\omega^2}{{c_0}^2}H'_y(\boldsymbol{r}), \tag{6}$$

$$\frac{1}{b}[E_x(\boldsymbol{r}) - E_x(\boldsymbol{r}+\boldsymbol{b})] - \frac{1}{a}[E_y(\boldsymbol{r}) - E_y(\boldsymbol{r}+\boldsymbol{a})] = c\mu\frac{\omega^2}{{c_0}^2}H'_z(\boldsymbol{r}), \tag{7}$$

$$-[H'_y(\boldsymbol{r}) - H'_y(\boldsymbol{r}-\boldsymbol{c})] + \frac{c}{b}[H'_z(\boldsymbol{r}) - H'_z(\boldsymbol{r}-\boldsymbol{b})] = \epsilon_{xx}(\boldsymbol{r})E_x(\boldsymbol{r}) + \epsilon_{xy}(\boldsymbol{r})E_y(\boldsymbol{r}) + \epsilon_{xz}(\boldsymbol{r})E_z(\boldsymbol{r}), \tag{8}$$

$$-\frac{c}{a}[H'_z(\boldsymbol{r}) - H'_z(\boldsymbol{r}-\boldsymbol{a})] + [H'_x(\boldsymbol{r}) - H'_x(\boldsymbol{r}-\boldsymbol{c})] = \epsilon_{yx}(\boldsymbol{r})E_x(\boldsymbol{r}) + \epsilon_{yy}(\boldsymbol{r})E_y(\boldsymbol{r}) + \epsilon_{yz}(\boldsymbol{r})E_z(\boldsymbol{r}), \tag{9}$$

$$-\frac{c}{b}[H'_x(\boldsymbol{r}) - H'_x(\boldsymbol{r}-\boldsymbol{b})] + \frac{c}{a}[H'_y(\boldsymbol{r}) - H'_y(\boldsymbol{r}-\boldsymbol{a})] = \epsilon_{zx}(\boldsymbol{r})E_x(\boldsymbol{r}) + \epsilon_{zy}(\boldsymbol{r})E_y(\boldsymbol{r}) + \epsilon_{zz}(\boldsymbol{r})E_z(\boldsymbol{r}), \tag{10}$$

where ω is the angular frequency, c_0 is the speed of light, μ is the magnetic permeability, ϵ_{ij} is the i, j component of the dielectric tensor, $\boldsymbol{r}$ is the position vector, $H' = (\frac{i}{c\omega\varepsilon_0})H$, and $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$ are the vectors of a cubic pixel in the x, y, and z directions, with a, b, and c being the lattice spacing in the x, y, and z directions [3].

From Eq. (5) through Eq. (10), we derive expressions for $E_z(\boldsymbol{r})$ and $H'_z(\boldsymbol{r})$:

$$E_z(\boldsymbol{r}) = -\frac{\epsilon_{zx}(\boldsymbol{r})}{\epsilon_{zz}(\boldsymbol{r})}E_x(\boldsymbol{r}) - \frac{\epsilon_{zy}(\boldsymbol{r})}{\epsilon_{zz}(\boldsymbol{r})}E_y(\boldsymbol{r}) - \frac{1}{\epsilon_{zz}(\boldsymbol{r})}\frac{c}{b}[H'_x(\boldsymbol{r}) - H'_x(\boldsymbol{r}-\boldsymbol{b})] + \frac{1}{\epsilon_{zz}(\boldsymbol{r})}\frac{c}{a}[H'_y(\boldsymbol{r}) - H'_y(\boldsymbol{r}-\boldsymbol{a})], \tag{11}$$

$$H'_z(\boldsymbol{r}) = \frac{{c_0}^2}{c\mu\omega^2}\left[\frac{E_x(\boldsymbol{r}) - E_x(\boldsymbol{r}+\boldsymbol{b})}{b} - \frac{E_y(\boldsymbol{r}) - E_y(\boldsymbol{r}+\boldsymbol{a})}{a}\right]. \tag{12}$$

By substituting Eqs. (11) and (12) into Eq. (5) through Eq. (10), we obtain the following transfer equations:

$$\begin{aligned} E_x(\boldsymbol{r}+\boldsymbol{c}) = {} & E_x(\boldsymbol{r}) + \frac{c^2\omega^2}{{c_0}^2}\mu H'_y(\boldsymbol{r}) + \frac{c^2}{a\epsilon_{zz}(\boldsymbol{r})}\left[\frac{\epsilon_{zx}(\boldsymbol{r})E_x(\boldsymbol{r}) + \epsilon_{zy}(\boldsymbol{r})E_y(\boldsymbol{r})}{c} + \frac{H'_y(\boldsymbol{r}-\boldsymbol{a}) - H'_y(\boldsymbol{r})}{a} - \frac{H'_x(\boldsymbol{r}-\boldsymbol{b}) - H'_x(\boldsymbol{r})}{b}\right] \\ & - \frac{c^2}{a\epsilon_{zz}(\boldsymbol{r}+\boldsymbol{a})}\left[\frac{\epsilon_{zx}(\boldsymbol{r}+\boldsymbol{a})E_x(\boldsymbol{r}+\boldsymbol{a}) + \epsilon_{zy}(\boldsymbol{r}+\boldsymbol{a})E_y(\boldsymbol{r}+\boldsymbol{a})}{c} + \frac{H'_y(\boldsymbol{r}) - H'_y(\boldsymbol{r}+\boldsymbol{a})}{a}\right] \\ & + \frac{c^2}{a\epsilon_{zz}(\boldsymbol{r}+\boldsymbol{a})}\left[\frac{H'_x(\boldsymbol{r}+\boldsymbol{a}-\boldsymbol{b}) + H'_x(\boldsymbol{r}+\boldsymbol{a})}{b}\right], \end{aligned} \tag{13}$$

$$\begin{aligned} E_y(\boldsymbol{r}+\boldsymbol{c}) = {} & E_y(\boldsymbol{r}) - \frac{c^2\omega^2}{{c_0}^2}\mu H'_x(\boldsymbol{r}) + \frac{c^2}{b\epsilon_{zz}(\boldsymbol{r})}\left[\frac{\epsilon_{zx}(\boldsymbol{r})E_x(\boldsymbol{r}) + \epsilon_{zy}(\boldsymbol{r})E_y(\boldsymbol{r})}{c} + \frac{H'_y(\boldsymbol{r}-\boldsymbol{a}) - H'_y(\boldsymbol{r})}{a} - \frac{H'_x(\boldsymbol{r}-\boldsymbol{b}) - H'_x(\boldsymbol{r})}{b}\right] \\ & - \frac{c^2}{b\epsilon_{zz}(\boldsymbol{r}+\boldsymbol{b})}\left[\frac{\epsilon_{zx}(\boldsymbol{r}+\boldsymbol{b})E_x(\boldsymbol{r}+\boldsymbol{b}) + \epsilon_{zy}(\boldsymbol{r}+\boldsymbol{b})E_y(\boldsymbol{r}+\boldsymbol{b})}{c} + \frac{H'_y(\boldsymbol{r}-\boldsymbol{a}+\boldsymbol{b}) - H'_y(\boldsymbol{r}+\boldsymbol{b})}{a}\right] \\ & + \frac{c^2}{b\epsilon_{zz}(\boldsymbol{r}+\boldsymbol{b})}\left[\frac{H'_x(\boldsymbol{r}) + H'_x(\boldsymbol{r}+\boldsymbol{b})}{b}\right], \end{aligned} \tag{14}$$

$$\begin{aligned} & H'_y(\boldsymbol{r}) - \left[\epsilon_{xx}(\boldsymbol{r}+\boldsymbol{c}) - \frac{\epsilon_{xz}(\boldsymbol{r}+\boldsymbol{c})\epsilon_{zx}(\boldsymbol{r}+\boldsymbol{c})}{\epsilon_{zz}(\boldsymbol{r}+\boldsymbol{c})}\right]E_x(\boldsymbol{r}+\boldsymbol{c}) - \left[\epsilon_{xy}(\boldsymbol{r}+\boldsymbol{c}) - \frac{\epsilon_{yz}(\boldsymbol{r}+\boldsymbol{c})\epsilon_{zy}(\boldsymbol{r}+\boldsymbol{c})}{\epsilon_{zz}(\boldsymbol{r}+\boldsymbol{c})}\right]E_y(\boldsymbol{r}+\boldsymbol{c}) \\ & + \frac{{c_0}^2}{b\omega^2\mu}\left[\frac{E_x(\boldsymbol{r}+\boldsymbol{c}) - E_x(\boldsymbol{r}+\boldsymbol{b}+\boldsymbol{c}) - E_x(\boldsymbol{r}-\boldsymbol{b}+\boldsymbol{c}) + E_x(\boldsymbol{r}+\boldsymbol{c})}{b}\right] \\ & - \frac{{c_0}^2}{b\omega^2\mu}\left[\frac{E_y(\boldsymbol{r}+\boldsymbol{c}) - E_y(\boldsymbol{r}+\boldsymbol{a}+\boldsymbol{c}) - E_y(\boldsymbol{r}-\boldsymbol{b}+\boldsymbol{c}) + E_y(\boldsymbol{r}+\boldsymbol{a}-\boldsymbol{b}+\boldsymbol{c})}{a}\right] \\ & = \left[1 + \frac{c\epsilon_{xz}(\boldsymbol{r}+\boldsymbol{c})}{a\epsilon_{zz}(\boldsymbol{r}+\boldsymbol{c})}\right]H'_y(\boldsymbol{r}+\boldsymbol{c}) - \frac{c\epsilon_{xz}(\boldsymbol{r}+\boldsymbol{c})}{a\epsilon_{zz}(\boldsymbol{r}+\boldsymbol{c})}H'_y(\boldsymbol{r}-\boldsymbol{a}+\boldsymbol{c}) - \frac{c\epsilon_{xz}(\boldsymbol{r}+\boldsymbol{c})}{b\epsilon_{zz}(\boldsymbol{r}+\boldsymbol{c})}H'_x(\boldsymbol{r}+\boldsymbol{c}) + \frac{c\epsilon_{xz}(\boldsymbol{r}+\boldsymbol{c})}{b\epsilon_{zz}(\boldsymbol{r}+\boldsymbol{c})}H'_x(\boldsymbol{r}-\boldsymbol{b}+\boldsymbol{c}), \end{aligned} \tag{15}$$

$$H_x'(\boldsymbol{r}) + \left[\epsilon_{yx}(\boldsymbol{r}+\boldsymbol{c}) - \frac{\epsilon_{yz}(\boldsymbol{r}+\boldsymbol{c})\epsilon_{zx}(\boldsymbol{r}+\boldsymbol{c})}{\epsilon_{zz}(\boldsymbol{r}+\boldsymbol{c})}\right] E_x(\boldsymbol{r}+\boldsymbol{c}) + \left[\epsilon_{yy}(\boldsymbol{r}+\boldsymbol{c}) - \frac{\epsilon_{yz}(\boldsymbol{r}+\boldsymbol{c})\epsilon_{zy}(\boldsymbol{r}+\boldsymbol{c})}{\epsilon_{zz}(\boldsymbol{r}+\boldsymbol{c})}\right] E_y(\boldsymbol{r}+\boldsymbol{c})$$
$$+ \frac{{c_0}^2}{a\omega^2\mu}\left[\frac{E_x(\boldsymbol{r}+\boldsymbol{c}) - E_x(\boldsymbol{r}+\boldsymbol{b}+\boldsymbol{c}) - E_x(\boldsymbol{r}-\boldsymbol{a}+\boldsymbol{c}) + E_x(\boldsymbol{r}-\boldsymbol{a}+\boldsymbol{b}+\boldsymbol{c})}{b}\right]$$
$$- \frac{{c_0}^2}{a\omega^2\mu}\left[\frac{E_y(\boldsymbol{r}+\boldsymbol{c}) - E_y(\boldsymbol{r}+\boldsymbol{a}+\boldsymbol{c}) - E_y(\boldsymbol{r}-\boldsymbol{a}+\boldsymbol{c}) + E_y(\boldsymbol{r}+\boldsymbol{c})}{a}\right]$$
$$= \left[1 + \frac{c\epsilon_{yz}(\boldsymbol{r}+\boldsymbol{c})}{b\epsilon_{zz}(\boldsymbol{r}+\boldsymbol{c})}\right] H_x'(\boldsymbol{r}+\boldsymbol{c}) - \frac{c\epsilon_{yz}(\boldsymbol{r}+\boldsymbol{c})}{b\epsilon_{zz}(\boldsymbol{r}+\boldsymbol{c})} H_x'(\boldsymbol{r}-\boldsymbol{b}+\boldsymbol{c}) - \frac{c\epsilon_{yz}(\boldsymbol{r}+\boldsymbol{c})}{a\epsilon_{zz}(\boldsymbol{r}+\boldsymbol{c})} H_y'(\boldsymbol{r}+\boldsymbol{c}) + \frac{c\epsilon_{yz}(\boldsymbol{r}+\boldsymbol{c})}{a\epsilon_{zz}(\boldsymbol{r}+\boldsymbol{c})} H_y'(\boldsymbol{r}-\boldsymbol{a}+\boldsymbol{c}). \tag{16}$$

We used Eq. (13) through Eq. (16) in our modified PHOTON program to implement the anisotropic subpixel smoothing scheme. It should be noted that these equations are applicable to nonmagnetic structures only.

When the program propagates the fields forward in the $+z$ direction, the E fields are first calculated using Eqs. (13) and (14), and the H fields are subsequently calculated using Eqs. (15) and (16). For backward propagation in the $-z$ direction, H field calculation precedes E field calculation. Equations (13) and (14) show that, for forward integration, the E fields at the z level of $\boldsymbol{r}+\boldsymbol{c}$ can be directly obtained from the E fields at the z level of $\boldsymbol{r}$. However, Eqs. (15) and (16) show that the H fields at the z level of $\boldsymbol{r}+\boldsymbol{c}$ cannot be directly obtained. Rather, for forward integration, the H fields can be obtained by solving a matrix problem with $H_x'(\boldsymbol{r}+\boldsymbol{c})$, $H_x'(\boldsymbol{r}-\boldsymbol{b}+\boldsymbol{c})$, $H_y'(\boldsymbol{r}+\boldsymbol{c})$, and $H_y'(\boldsymbol{r}-\boldsymbol{a}+\boldsymbol{c})$ as unknowns. This matrix problem occurs because ϵ is a tensor. Note that, in Eqs. (15) and (16), the E fields at the z level of $\boldsymbol{r}+\boldsymbol{c}$ have already been calculated in forward integration from Eqs. (13) and (14) and are not unknown. If the off-diagonal elements of the ϵ tensor are zero, the H fields at the $\boldsymbol{r}+\boldsymbol{c}$ plane are directly obtained without solving a matrix problem. For backward integration, the H fields at a z plane are directly obtained from the fields at the $\boldsymbol{r}+\boldsymbol{c}$ plane using Eqs. (15) and (16). However, to find E fields in backward integration, Eqs. (13) and (14) become a matrix problem with $E_x(\boldsymbol{r})$, $E_y(\boldsymbol{r})$, $E_x(\boldsymbol{r}+\boldsymbol{a})$, and $E_y(\boldsymbol{r}+\boldsymbol{a})$ as unknowns for Eq. (13) and with $E_x(\boldsymbol{r})$, $E_y(\boldsymbol{r})$, $E_x(\boldsymbol{r}+\boldsymbol{b})$, and $E_y(\boldsymbol{r}+\boldsymbol{b})$ as unknowns for Eq. (14). Because a matrix problem is required to solve the transfer equations, the calculation time for the modified PHOTON program is greater than that of the unmodified PHOTON program for the same structure and mesh size, as will be discussed later.

C. Computational Experiments

After implementing our anisotropic subpixel smoothing scheme in the PHOTON program, we performed a set of computational experiments. The computing environment used for these experiments was a local workstation running Microsoft Windows 11 with an 11th Gen Intel Core i7-1165G7 @ 2.80 GHz processor and 8 GB of RAM. The TMM code for each experiment was compiled using the Intel FORTRAN compiler version 2025.3.2.

First, we calculated the transmission data for two 2D structures using the modified PHOTON program and compared it to the transmission data calculated using the unmodified PHOTON program (which uses simple volumetric averaging to calculate a scalar ϵ value for each pixel). We also calculated the transmission data for these two structures using COMSOL, a finite-element method (FEM) program, for comparison with our PHOTON data. The reason why we used FEM output as a benchmark for our study is that we know from previous research carried out in our lab and from existing literature that FEM tends to produce a more accurate output than TMM [11]. Like our TMM program, COMSOL can handle dielectric averaging [12]. One of the advantages of FEM is the flexibility of its meshing; for example, it allows for conformal meshing at interfaces between different materials, which removes the need for effective approximations of ϵ and enhances the accuracy of the output [13]. However, the complex meshing procedures of FEM render this method computationally demanding, especially for 3D structures, which results in a substantially higher computational time needed to complete a simulation compared to TMM. On the other hand, TMM is more efficient but less accurate than FEM due to its simpler meshing procedures and use of effective medium approximations [13].

The main goal of our study was to determine if our modified TMM method (i.e., our modified PHOTON program) demonstrated an improvement over the conventional TMM method (i.e., the unmodified PHOTON program) by producing output that was more closely aligned with the FEM output for dielectric and metallic structures. We also set out to determine the rate of convergence of both PHOTON programs and to determine the difference in calculation time between these two programs.

The first structure that we studied was a 2D structure composed of a c-Si cylinder 280 nm in diameter within a square-shaped silica substrate measuring 700 nm on the x and z sides (the structure is not infinite in the z dimension). We situated the structure within the x−z plane, rather than the x−y plane, so that we could test the accuracy of our program for a structure with materials that vary along the z axis, which is the axis along which light travels in our simulation. This substrate was embedded in vacuum and was 1 nm thick. We modeled this structure using both the conventional and modified TMM programs with a 90 × 90 discretization mesh. (The anisotropic subpixel smoothing TMM code for this structure is the file `cylinder_aniso_Si.f`, and the original TMM code

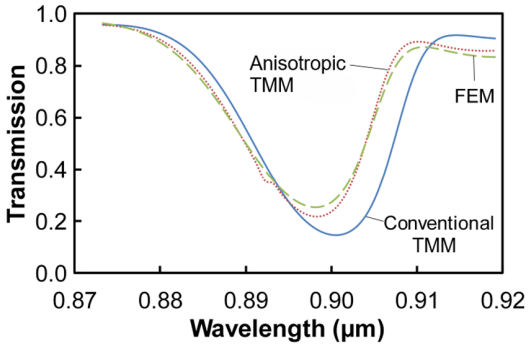

Fig. 1. Optical transmission versus wavelength for the c-Si cylindrical structure. The three methods shown are TMM using the conventional scheme, TMM using the anisotropic subpixel smoothing scheme, and FEM.

for this structure is the file `cylinder_original_Si.f`. The relevant Si dielectric values are also provided in the file `Si.txt`. All of these files can be found in Dataset 1, Ref. [14], Dataset 2, Ref. [15], Code 1, Ref. [16], Code 2, Ref. [17], Code 3, Ref. [18], and Code 4, Ref. [19], as well as in the GitHub repository [20]). We calculated the optical transmission over a wavelength range of 0.873–0.919 μm. We also calculated the transmission for the same structure using the FEM program. The transmission data for all three programs are shown in Fig. 1. We observe that, compared to the curve for the unmodified TMM program, the curve for the modified TMM program is significantly more like the FEM curve, suggesting that anisotropic subpixel smoothing improves the accuracy of the TMM calculation for this dielectric structure.

Additionally, we modeled a 2D structure composed of an Ag cylinder 200 nm in diameter within a square-shaped silica substrate measuring 500 nm on the x and z sides (the structure is not infinite in the z dimension). Similarly to the first structure, we situated this structure within the x−z plane, rather than the x−y plane. This substrate was embedded in vacuum and was 1 nm thick. Like the previously described c-Si structure, we modeled this structure using both the conventional and anisotropic subpixel smoothing TMM programs with a $90 * 90$ discretization mesh. (The anisotropic subpixel smoothing TMM code for this structure is the file named `cylinder_aniso_Ag.f`, and the original TMM code for this structure is the file `cylinder_original_Ag.f`. The relevant Ag dielectric values are also provided in the file `Ag.txt`. All of these files can be found in Dataset 1, Ref. [14], Dataset 2, Ref. [15], Code 1, Ref. [16], Code 2, Ref. [17], Code 3, Ref. [18], and Code 4, Ref. [19], as well as in the GitHub repository [20]). The optical transmission for this structure was calculated over a wavelength range of 0.55–1.00 μm. The transmission data are shown in Fig. 2(a). It can be observed from this figure that although the curve for the modified TMM program is similar to the FEM curve at lower wavelengths, it deviates from the FEM curve significantly at higher wavelengths, even being less accurate than the unmodified TMM program at some wavelengths.

To gain a better understanding of this phenomenon, Fig. 2(b) displays electric field vector plots for all three programs at the two wavelength values indicated in Fig. 2(a). The left-hand

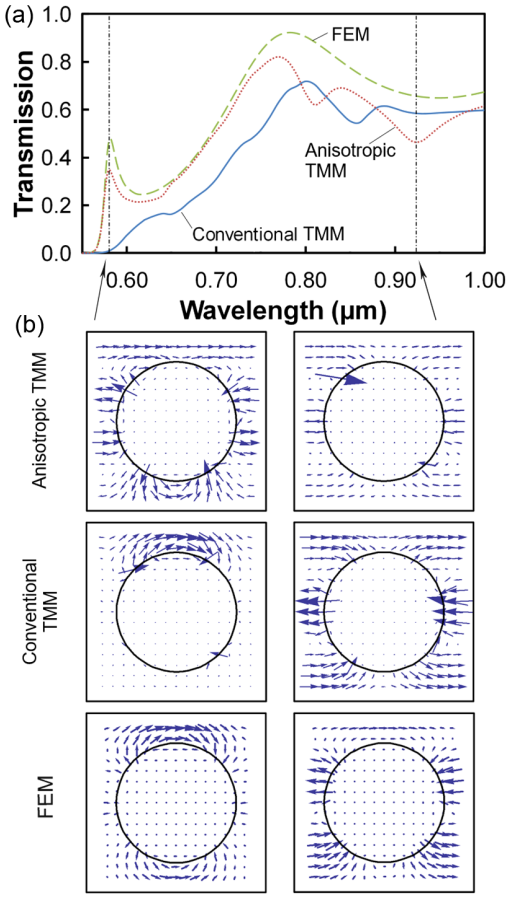

Fig. 2. (a) Optical transmission versus wavelength for the Ag cylindrical structure. (b) Electric field vector plots for the Ag cylindrical structure when a normally incident plane electromagnetic wave corresponding to mode (0, 0) hits the structure from the +z direction at the wavelengths indicated in (a) by the dashed vertical lines.

set of vector plots shows the electric field at the resonance peak located at 0.58 μm. This resonance peak is notable because the unmodified TMM program does not capture this peak, whereas the modified TMM program captures it, indicating that the anisotropic subpixel smoothing scheme is significantly more accurate than the conventional smoothing scheme at this wavelength. In addition, at the bottom of the structure, the electric field for the unmodified TMM program is much smaller in magnitude than the electric fields for the FEM model and the modified TMM program, which further demonstrates the modified TMM program's greater accuracy when compared with the unmodified TMM program.

On the other hand, the right-hand set of vector plots in Fig. 2(b), which show the electric field at 0.92 μm, show a large deviation in the anisotropic subpixel smoothing curve from the FEM curve, making it even less accurate than the conventional TMM program at this wavelength. The electric field plot for the modified TMM program is significantly different from the other two plots with regard to the magnitude of the vectors; in the plot for the modified TMM program, most of the electric field vectors are smaller than the corresponding vectors in the plots for the FEM and unmodified TMM programs, except for a single vector located at the upper-left edge of the cylinder, which

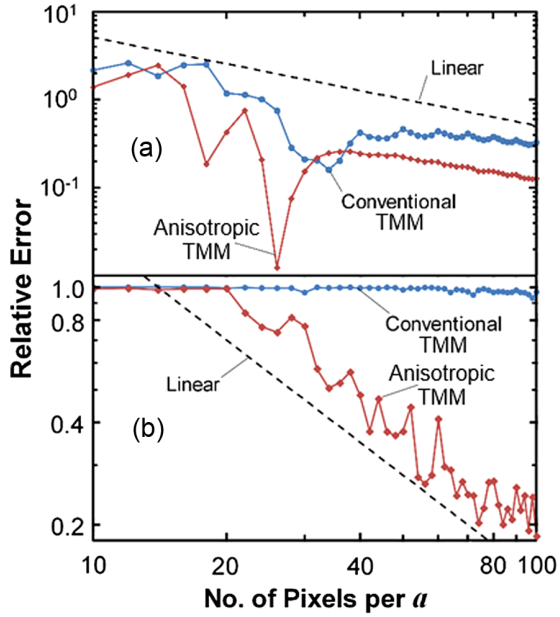

Fig. 3. Relative error compared to FEM results for (a) the c-Si cylindrical structure and (b) the Ag cylindrical structure using the conventional TMM scheme and the anisotropic subpixel smoothing TMM scheme. The relative error is plotted versus the number of pixels per a on a log-log scale. A perfect linear curve is included for comparison.

is much larger than the corresponding vector for the other two programs. The plots shown in Fig. 2(b) demonstrate that for metallic structures, the anisotropic subpixel smoothing TMM scheme is not necessarily more accurate than the conventional TMM scheme for all wavelengths.

Next, we analyzed the rate of convergence of the relative error of the transmission data with the TMM discretization mesh size (in other words, the number of pixels per a). We varied the number of pixels per a from 10 to 100 and calculated the relative error for the conventional and anisotropic TMM subpixel smoothing schemes at each mesh size. The error was calculated relative to the FEM value. The optical frequency was kept constant at 1.38 eV (approximately 0.90 μm) for the c-Si structure; a constant wavelength of 0.58 μm was used for the Ag structure. Figure 3(a) shows the relative error versus mesh size plot for the c-Si structure, and Fig. 3(b) shows the same plot for the Ag structure. A logarithmic scale is used for the x and y axes, and a linear curve is included for comparison.

Linear fits were performed on the areas of Fig. 3 curves where a linear fit is meaningful. The equation that describes a linear relationship between the relative error, R, and the number of pixels, N, on a log-log plot is shown by Eq. (17):

$$\ln R = -n \ln N + C, \tag{17}$$

where n is the slope and C is a constant value. For the c-Si structure, for which the relative error data are shown in Fig. 3(a), a linear fit was performed over the range $N = 50 - 100$; we found that for the conventional TMM curve, $n = 0.46$, and for the anisotropic subpixel smoothing TMM curve, $n = 0.83$. For the Ag structure, for which the relative error data are shown in Fig. 3(b), a linear fit was performed over the range $N = 22 - 100$; for the conventional TMM curve, $n = 0.03$, and for the anisotropic subpixel smoothing TMM curve, $n = 0.995$. We observe from these numbers that for both structures, the anisotropic subpixel smoothing scheme has a significantly higher rate of convergence than the conventional scheme, and that this difference is much more pronounced for the Ag structure than for the c-Si structure.

Error analysis indicates that $n = 1$ when the boundary condition does not cause any error, so n will always be less than or equal to 1 for TMM calculations. For structures in which the rate of convergence for the conventional scheme is close to 1, the anisotropic subpixel smoothing scheme will improve the rate of convergence to a lesser degree, since there is less improvement that can possibly be had. We also observe that for both structures, although there is a downward trend in relative error as the number of pixels increases, the anisotropic subpixel smoothing curve is not monotonically decreasing but rather has peaks and troughs, which are especially pronounced for the c-Si structure [see Fig. 3(a)]. The rate of convergence for the c-Si structure suggests an optimal number of pixels per a of 26 when using the anisotropic subpixel smoothing scheme. The irregularities in the rate-of-convergence curves may be caused by variations in accuracy resulting from the way the geometry of a particular structure is discretized for certain mesh sizes.

Finally, we measured the calculation time for both the modified and unmodified TMM programs. The time taken to calculate the transmission of light of a single wavelength through the c-Si cylindrical structure was measured for each number of pixels per a from 10 to 100. The results are displayed in Fig. 4 on a log-log plot. From an analysis of the code of the TMM programs, we expect that the time data for the conventional TMM program and the anisotropic subpixel smoothing TMM program should be consistent with Eqs. (18) and (19), respectively:

$$t_{conventional} = C N_{zm} (4 N_{xm} N_{ym})^3, \tag{18}$$

$$\begin{aligned} t_{anisotropic} &= C N_{zm} (4 N_{xm} N_{ym})^3 + \beta C N_{zm} (2 N_{xm} N_{ym})^3 \\ &= \left(1 + \frac{\beta}{8}\right) C N_{zm} (4 N_{xm} N_{ym})^3 \\ &= \left(1 + \frac{\beta}{8}\right) C N_{zm} (4 N_{xm} N_{ym})^3 = \left(1 + \frac{\beta}{8}\right) t_{conventional}. \end{aligned} \tag{19}$$

In these equations, t is the calculation time, C and β are constant values relating to the calculation time per pixel, and N_{xm}, N_{ym}, and N_{zm} are the number of pixels along the x, y, and z axes, respectively. For our 2D structures, $N_{ym} = 1$ and $N_{xm} = N_{zm}$. In our TMM programs, the number of pixels along the z axis is given by the expression IZMAX $*$ NZMAX; we set IZMAX $= 1$ for all cases to avoid numerical instability, so that $N_{zm} =$ NZMAX. By choosing IZMAX $= 1$, we minimized the increase in time it takes for the anisotropic subpixel smoothing TMM program to calculate transmission data compared to the conventional TMM program.

From our analysis, we expect that the calculation time for both TMM programs should be proportional to $N_{xm}{}^4$ for our

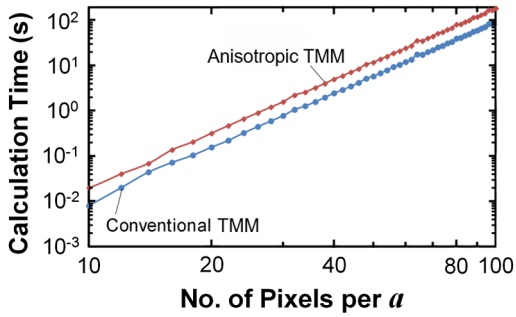

Fig. 4. Calculation time for one wavelength value for the c-Si cylindrical structure using the conventional TMM scheme and the anisotropic subpixel smoothing TMM scheme. The calculation time is plotted versus the number of pixels per a on a log-log scale.

2D structures, since $N_{ym} = 1$ and $N_{xm} = N_{zm}$. From performing linear fits on the data in Fig. 4, we find that the actual calculation time is proportional to $N_{xm}^{3.96}$ for both programs. The calculation time is larger for the anisotropic subpixel smoothing program than for the conventional TMM program by a factor of 2, so that $\beta = 8$. This particular value is a result of our use of certain compiler options. To avoid stack overflow, we used the -heap-arrays option when compiling our TMM programs with the Intel FORTRAN compiler; this option allocates all arrays on the heap instead of the stack, which increases the calculation time. Without using this compiler option, β would be smaller.

In conclusion, the implementation of the anisotropic subpixel smoothing scheme within the TMM program described in [2] has been shown to improve the accuracy of the program's output for a c-Si structure and, in general, for an Ag structure. However, caution must be exercised when using this anisotropic smoothing scheme for metallic structures, as its accuracy is worse than that of the conventional TMM scheme for some optical wavelengths. On the other hand, the use of the anisotropic subpixel smoothing TMM program significantly improved the rate of convergence for both the c-Si and Ag structures, and the increase in computational cost (i.e., calculation time) was reasonable. The anisotropic subpixel smoothing TMM program could be used to increase the accuracy of optical calculations performed on dielectric structures with discontinuities, such as nanostructures for solar photovoltaic applications. Future work could entail improving the overall accuracy of our modified TMM program for metallic structures, examining the accuracy of our program for other structures with various materials and geometries, comparing the results of our TMM program to those of FDTD programs that implement subpixel smoothing (such as MEEP [21]) and investigating the irregularities in our rate-of-convergence plots. Another idea for future work is to implement and test a subpixel smoothing method for structures containing anisotropic materials, such as the method described in [22] for FDTD simulations.

Funding. University of New Mexico; Oak Ridge Associated Universities.

Acknowledgment. Thanks to the School of Engineering at the University of New Mexico for providing the computing resources needed to perform this study.

Disclosures. The authors declare no conflicts of interest.

Data availability. Data available from the corresponding author upon reasonable request. See also Dataset 1, Ref. [14]; Dataset 2, Ref. [15]; Code 1, Ref. [16]; Code 2, Ref. [17]; Code 3, Ref. [18]; Code 4, Ref. [19]; and Ref. [20].

REFERENCES

1. T. G. Mackay and A. Lakhtakia, *The Transfer-Matrix Method in Electromagnetics and Optics* (Springer Nature, 2022).
2. P. M. Bell, J. B. Pendry, L. M. Moreno, *et al.*, "A program for calculating photonic band structures and transmission coefficients of complex structures," Comput. Phys. Commun. **85**, 306–322 (1995).
3. A. J. Ward, "Transfer matrices, photonic bands and related quantities," Ph.D. thesis (University of London, 1996).
4. A. Farjadpour, D. Roundy, A. Rodriguez, *et al.*, "Improving accuracy by subpixel smoothing in the finite-difference time domain," Opt. Lett. **31**, 2972–2974 (2006).
5. O.-Z. Lipan and A. D. Sabata, "Bound states in the continuum via singular transfer matrices," Phys. Rev. A **111**, 033516 (2025).
6. O.-Z. Lipan and A. D. Sabata, "Closed-form analytical solution for the transfer matrix based on Pendry-MacKinnon discrete Maxwell's equations," Opt. Express **33**, 3777–3817 (2025).
7. S. G. Johnson and J. D. Joannopoulos, "Block-iterative frequency-domain methods for Maxwell's equations in a planewave basis," Opt. Express **8**, 173–190 (2001).
8. S. G. Johnson, M. Ibanescu, M. A. Skorobogatiy, *et al.*, "Perturbation theory for Maxwell's equations with shifting material boundaries," Phys. Rev. E **65**, 066611 (2002).
9. A. Ditkowski, K. Dridi, and J. S. Hesthaven, "Convergent Cartesian grid methods for Maxwell's equations in complex geometries," J. Comput. Phys. **170**, 39–80 (2001).
10. R. D. Meade, A. M. Rappe, K. D. Brommer, *et al.*, "Accurate theoretical analysis of photonic band-gap materials," Phys. Rev. B **55**, 15942 (1997).
11. I. Haddouche and L. Cherbi, "Comparison of finite element and transfer matrix methods for numerical investigation of surface plasmon waveguides," Opt. Commun. **382**, 132–137 (2017).
12. "Homogenization of material properties," (COMSOL Learning Center, 2026). Accessed: 2026-04-19.
13. K. Han and C.-H. Chang, "Numerical modeling of sub-wavelength anti-reflective structures for solar module applications," Nanomaterials **4**, 87–128 (2014).
14. B. R. Hoard, S. Atiganayun, and S. E. Han, "Ag.txt," figshare (2026), https://doi.org/10.6084/m9.figshare.31190764.
15. B. R. Hoard, S. Atiganayun, and S. E. Han, "Si.txt," figshare (2026), https://doi.org/10.6084/m9.figshare.31190767.
16. B. R. Hoard, S. Atiganayun, and S. E. Han, "cylinder_original_Ag.f," figshare (2026), https://doi.org/10.6084/m9.figshare.31190779.
17. B. R. Hoard, S. Atiganayun, and S. E. Han, " cylinder_aniso_Si.f," Version 2025.3.2, figshare (2026), https://doi.org/10.6084/m9.figshare.31190785.
18. B. R. Hoard, S. Atiganayun, and S. E. Han, "cylinder_aniso_Ag.f," Version 2025.3.2, figshare (2026), https://doi.org/10.6084/m9.figshare.31190782.
19. B. R. Hoard, S. Atiganayun, and S. E. Han, "cylinder_original_Si.f," Version 2025.3.2, figshare (2026), https://doi.org/10.6084/m9.figshare.31190788.
20. B. R. Hoard, "Anisotropic-subpixel," GitHub (2026), https://github.com/hoardbrittany-source/anisotropic-subpixel/.
21. A. Oskooi, D. Roundy, M. Ibanescu, *et al.*, "MEEP: a flexible free-software package for electromagnetic simulations by the FDTD method," Comput. Phys. Commun. **181**, 687–702 (2010).
22. A. F. Oskooi, C. Kottke, and S. G. Johnson, "Accurate finite-difference time-domain simulation of anisotropic media by subpixel smoothing," Opt. Lett. **34**, 2778–2780 (2009).