Implementation of anisotropic subpixel smoothing in a transfer matrix method program

BRITTANY R. HOARD,[1,3,*] SARUN ATIGANYANUN,[1] AND SANG EON HAN[1,2]

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

Abstract: 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 an 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.

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 is 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, [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. 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 method used to calculate the discretized dielectric function for each pixel, the PHOTON program and its methods suffer from inaccuracy when used to model discontinuous dielectric structures [4].

2. Objectives

The objective of our study is to improve the accuracy of the PHOTON program by implementing 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. 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, 5]. The rate of convergence with the grid resolution can be negatively affected by inaccuracies caused by ϵ interfaces [6].

3. Methods

3.1. Average dielectric tensor

The anisotropic subpixel smoothing scheme developed by [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 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. This average dielectric tensor uses the values $< \epsilon >$ and $< \epsilon^{-1} >^{-1}$, 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 Eq. (1) and Eq. (2):

$$< \epsilon > = \epsilon_1 f_1 + \epsilon_2 f_2 \tag{1}$$

$$< \epsilon^{-1} >^{-1} = (\epsilon_1^{-1} f_1 + \epsilon_2^{-1} f_2)^{-1} \tag{2}$$

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

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

$$\tilde{\epsilon}^{-1} = P < \epsilon^{-1} > + (1 - P) < \epsilon >^{-1} \tag{3}$$

where $P_{ij} = n_i n_j$ is the projection tensor onto the normal and <> 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:

$$\tilde{\epsilon} = \begin{vmatrix} (n_y{}^2 + n_z{}^2) < \epsilon > + n_x{}^2 < \epsilon^{-1} >^{-1} & (< \epsilon^{-1} >^{-1} - < \epsilon >) n_x n_y & (< \epsilon^{-1} >^{-1} - < \epsilon >) n_x n_z \\ (< \epsilon^{-1} >^{-1} - < \epsilon >) n_x n_y & (n_x{}^2 + n_z{}^2) < \epsilon > + n_y{}^2 < \epsilon^{-1} >^{-1} & (< \epsilon^{-1} >^{-1} - < \epsilon >) n_y n_z \\ (< \epsilon^{-1} >^{-1} - < \epsilon >) n_x n_z & (< \epsilon^{-1} >^{-1} - < \epsilon >) n_y n_z & (n_x{}^2 + n_y{}^2) < \epsilon > + n_z{}^2 < \epsilon^{-1} >^{-1} \end{vmatrix} \tag{4}$$

where n_x, n_y, n_z are the surface normals in the x, y, and z directions [7].

3.2. 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, $\mathbf{r}$ is the position vector, $H' = (\frac{i}{c\omega\varepsilon_0})H$, and $\boldsymbol{a}$,$\boldsymbol{b}$,$\boldsymbol{c}$ are the vectors of a cubic pixel in the x, y, z directions, with a,b,c being the lattice spacing in the x, y, 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 Eq. (11) and Eq. (12) into Eq. (5) through Eq. (10), we obtain the following transfer equations:

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

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

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

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

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 Eq. (13) and Eq. (14), and the H fields are subsequently calculated using Eq. (15) and Eq. (16). For backward propagation in the $-z$ direction, H field calculation precedes E field calculation. Eq. (13) and Eq. (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, Eq. (15) and Eq. (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 Eq. (15) and Eq. (16), the E fields at the z level of $\boldsymbol{r}+\boldsymbol{c}$ have already been calculated in forward integration from Eq. (13) and Eq. (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 Eq. (15) and Eq. (16). However, to find E fields in backward integration, Eq. (13) and Eq. (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.

3.3. Computational Experiments

After implementing our anisotropic subpixel smoothing scheme in the PHOTON program, 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 more accurate output than TMM [8]. 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 [9]. However, the complex meshing procedures of FEM render this method computationally demanding, especially for 3D structures. On the other hand, TMM is more efficient but less accurate than FEM due to its simpler meshing procedures and use of effective media approximations [9].

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 cylindrical structure that was composed of a c-Si cylinder 280 nm in diameter within a square-shaped silica substrate measuring 700 nm on a side. We modeled this structure using both the conventional and modified TMM programs with a 90x90 discretization mesh. We calculated the optical transmission over a wavelength range of 0.873 µm to 0.919 µm. We also calculated the transmission for the same structure using the FEM program. The transmission data for all three programs is shown in Figure 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.

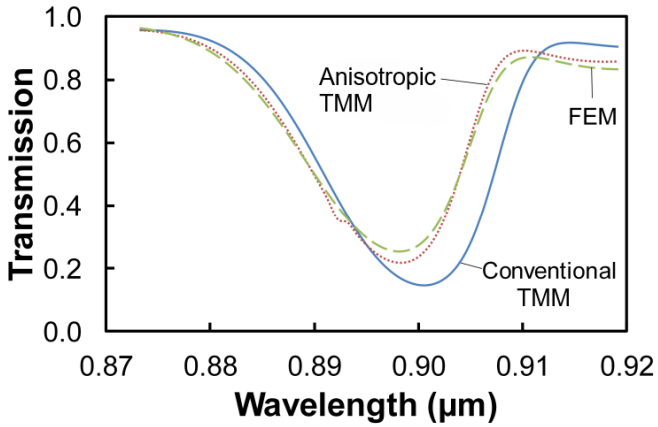

Fig. 1. Optical transmission versus wavelength for the c-Si cylindrical structure. The three methods shown are: TMM using the conventional scheme, TMM using the anisotropic subpixel smoothing scheme, and FEM.

Additionally, we modeled a 2D cylindrical structure composed of an Ag cylinder 200 nm in diameter within a square-shaped silica substrate measuring 500 nm on a side. Like the previously described c-Si structure, we modeled this structure using both the conventional and anisotropic subpixel smoothing TMM programs with a 90x90 discretization mesh. The optical transmission for this structure was calculated over a wavelength range of 0.55 µm to 1.00 µm. The transmission data is shown in Figure 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, Figure 2(b) displays electric field vector plots for all three programs at the two wavelength values indicated in Figure 2(a). The left-hand 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

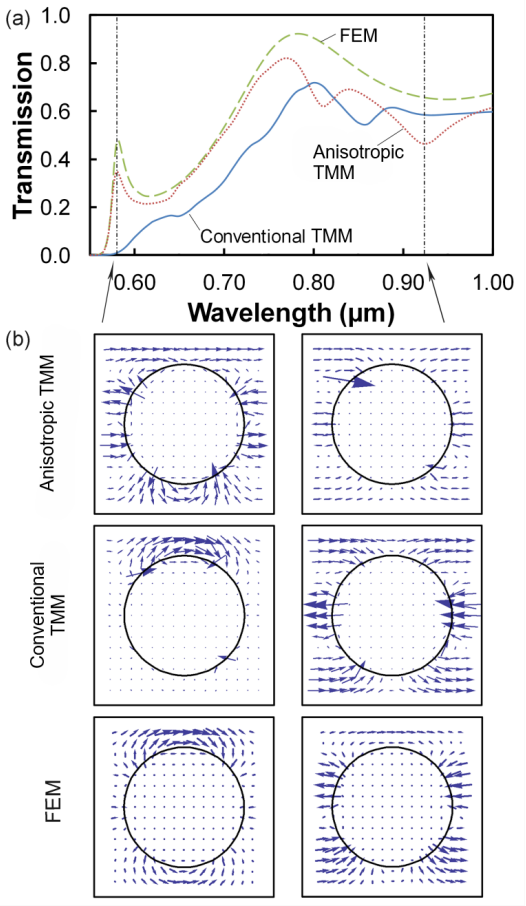

Fig. 2. (a) Optical transmission versus wavelength for the Ag cylindrical structure. (b) Electric field vector plots for the Ag cylindrical structure at the wavelengths indicated in (a) by the dashed vertical lines.

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 Figure 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 regards 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 is much larger than the corresponding vector for the other two programs. The plots shown in Figure 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 Figure 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 the Figure 3 curves where a linear fit is meaningful. The equation that describes a linear relationship between the relative error, ε, and the number of pixels, N, on a log-log plot is shown by Equation (17):

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

where n is the slope and C is a constant value. For the c-Si structure, for which the relative error data is shown in Figure 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 is shown in Figure 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 Figure 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 10-100. The results are displayed in Figure 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 Equation (18) and Equation (19), respectively.

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

$$t_{anisotropic} = CN_{zm}(4N_{xm}N_{ym})^3 + \beta CN_{zm}(2N_{xm}N_{ym})^3 = \left(1 + \frac{\beta}{8}\right) CN_{zm}(4N_{xm}N_{ym})^3$$
$$= \left(1 + \frac{\beta}{8}\right) CN_{zm}(4N_{xm}N_{ym})^3 = \left(1 + \frac{\beta}{8}\right) t_{conventional} \tag{19}$$

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

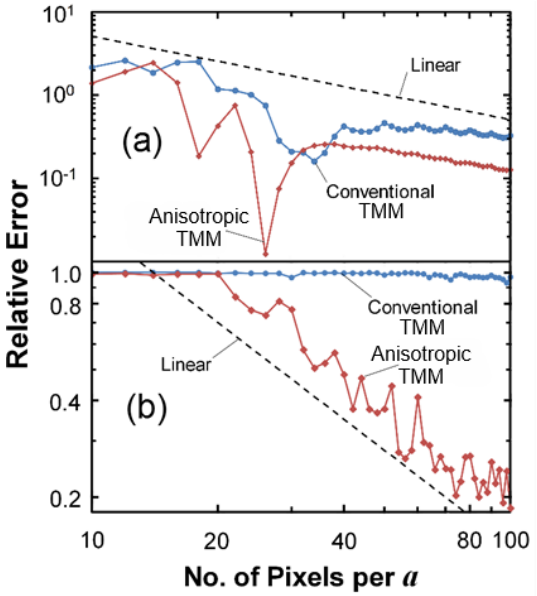

Fig. 3. Relative error compared to FEM results for (a) the c-Si cylindrical structure and (b) the Ag cylindrical structure using the conventional TMM scheme and the anisotropic subpixel smoothing TMM scheme. The relative error is plotted versus the number of pixels per a on a log-log scale. A perfect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

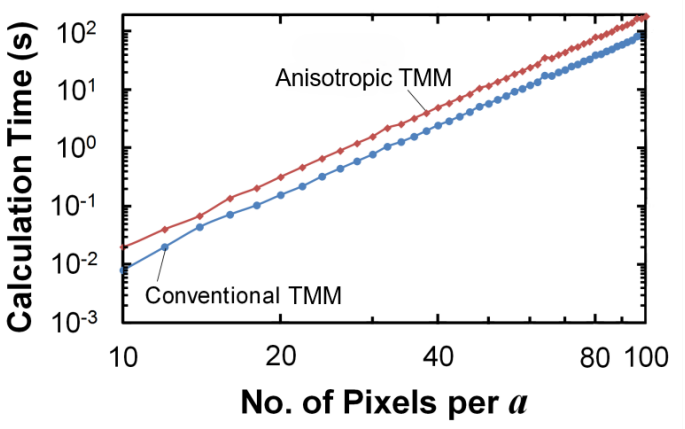

Fig. 4. The calculation time for one wavelength value for the c-Si cylindrical structure using the conventional TMM scheme and the anisotropic subpixel smoothing TMM scheme. The calculation time is plotted versus the number of pixels per a on a log-log scale.

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 2D structures, since $N_{ym} = 1$, and $N_{xm} = N_{zm}$. From performing linear fits on the data in Figure 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, 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, geometries, and numbers of layers, and investigating the irregularities in our rate-of-convergence plots.

4. Back matter

Funding. National Science Foundation CHE-1231046; Oak Ridge Associated Universities Ralph E. Powe Junior Faculty Enhancement Award; School of Engineering at the University of New Mexico.

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 Statement. Data available from the corresponding author upon reasonable request.

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, and A. J. Ward, "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. 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).
6. 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).
7. 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).
8. 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).

9. K. Han and C.-H. Chang, “Numerical modeling of sub-wavelength anti-reflective structures for solar module applications,” Nanomaterials **4**, 87–128 (2014).