

The Transfer Matrix Method in Linear Dielectrics

Claudio Barros

University at Buffalo, Amherst, New York 14260

(Dated: December 18, 2020)

1. Introduction

The Transfer Matrix Method (TMM) provides us with a powerful tool for calculating the propagation of light, or for that matter any wave-like phenomenon, through a series of optical elements. Most startling of all, the apparent complex nature of light propagation can be greatly simplified by casting these optical elements into matrices that transform a given initial beam of light into a final state based solely on the properties of the matrices. Now, these optical elements can represent a wide range of contraptions such as wave-plates, lenses, potentials, or films, which is the main subject of this paper. More specifically, we shall focus on the propagation of an electromagnetic wave through a series of dielectric thin-films of different thicknesses, and with different indices of refraction. Our starting point, as always, will be the Maxwell Equations coupled with their boundary conditions, after which it will be straightforward to arrive at expressions for our transfer matrices. The main purpose of this method is to eventually produce reflection and transmission plots for different layers of materials, and to model systems such as anti-reflecting coatings, among others. Next, we will use these tools to develop a program written in Python to verify our results and see how they compare with similar experiments performed in real-life.

2. Electromagnetic Plane-Waves in Dielectric Media

Let us begin with Maxwell's equations in their most general form:

$$\nabla \cdot \mathbf{D} = \rho_f \tag{2.1}$$

$$\nabla \cdot \mathbf{B} = 0 \tag{2.2}$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \tag{2.3}$$

$$\nabla \times \mathbf{H} = \frac{\partial \mathbf{D}}{\partial t} + \mathbf{J}_f \tag{2.4}$$

where ρ_f and \mathbf{J}_f are the free charge and current densities, respectively. In the next section we shall closely follow much of the conventions used in [1] chapter 7. In the absence of any sources, these two quantities vanish; since we are only considering wave propagation through dielectric media, we shall explicitly omit them for the remainder of our calculations. Furthermore, for isotropic linear media, we have the relations:

$$\mathbf{D} = \epsilon \mathbf{E} \quad (2.5)$$

$$\mathbf{H} = \frac{1}{\mu} \mathbf{B} \quad (2.6)$$

where $\epsilon = \epsilon_0(1 + \chi_e)$ and $\mu = \mu_0(1 + \chi_m)$ are the permittivity and permeability of the material. Next, let us assume an ansatz solution with a harmonic time dependence $e^{-i\omega t}$ and substitute this along with our previous assumptions into Eq.(2.3) and Eq.(2.4) to obtain:

$$\nabla \times \mathbf{E} - i\omega \mathbf{B} = 0 \quad (2.7)$$

$$\nabla \times \mathbf{B} + i\omega \mu \epsilon \mathbf{E} = 0 \quad (2.8)$$

To obtain the Helmholtz wave equation, we simply take the divergence of these two equations and combine them to eliminate either \mathbf{E} or \mathbf{B} . The result becomes:

$$(\nabla^2 + \mu \epsilon \omega^2) \mathbf{E} = 0 \quad (2.9)$$

$$(\nabla^2 + \mu \epsilon \omega^2) \mathbf{B} = 0 \quad (2.10)$$

As usual, we assume plane-wave solutions of the form $e^{i(k\mathbf{n}\cdot\mathbf{r}-\omega t)}$ for some wave-vector $\mathbf{k} = k\mathbf{n}$, where \mathbf{n} is a unit vector pointing in the direction of propagation. Substituting this plane-wave into Eq.(2.9) results in the condition:

$$k^2 = \mu \epsilon \omega^2 \quad (2.11)$$

which is known as the dispersion relation. From the definition of the index of refraction $n = c/v$ (n is not to be confused with the unit vector \mathbf{n}), we can derive an expression for the index of refraction that is entirely dependent on the properties of the material through which our wave is propagating:

$$n = \sqrt{\frac{\mu \epsilon}{\mu_0 \epsilon_0}} \quad (2.12)$$

With all this in mind, the plane-wave solutions to Eq.(2.9) and Eq.(2.10) yield:

$$\mathbf{E}(\mathbf{r}, t) = \mathbf{E}_0 e^{i(k\mathbf{n}\cdot\mathbf{r}-\omega t)} \quad (2.13)$$

$$\mathbf{B}(\mathbf{r}, t) = \mathbf{B}_0 e^{i(k\mathbf{n}\cdot\mathbf{r}-\omega t)} \quad (2.14)$$

Here the amplitudes \mathbf{E}_0 and \mathbf{B}_0 can be, in general complex; the real, physical fields are given by taking the real component of the above equations. One matter should be clarified at this

point. We note that $\mathbf{n} = \mathbf{n}_R + i\mathbf{n}_I$, which demonstrates that \mathbf{n} need not be a real vector. When this is taken into account, we obtain the additional term $e^{-k\mathbf{n}_I \cdot \mathbf{r}}$ in our plane-wave solutions, clearly exhibiting exponential growth or decay. Alternatively, this outcome can also be achieved by defining a complex index of refraction $n = \alpha + i\beta$ instead. In either case, these two situations are related to each other through their dispersion relation. For this paper we shall explicitly assume a complex index of refraction with a positive imaginary coefficient β , otherwise known as the extinction coefficient. From the additional term $e^{-\beta\mathbf{k} \cdot \mathbf{r}}$ it becomes apparent that this condition excludes exponential growth (i.e. stimulated emission) from consideration.

Two important consequences of the Maxwell equations allow us to simplify matters even further. From Gauss' laws for magnetic and electric fields, we see that:

$$\mathbf{n} \cdot \mathbf{E}_0 = \mathbf{n} \cdot \mathbf{B}_0 = 0 \quad (2.15)$$

which simply states that electromagnetic waves are transverse in nature. Next, we can conclude from Faraday's law that the magnetic field can be formulated as:

$$\mathbf{B}_0 = \sqrt{\mu\epsilon}\mathbf{n} \times \mathbf{E}_0 \quad (2.16)$$

Put together, these two equations demonstrate that the magnetic and electric fields are in phase and orthogonal to one another. It is for this reason that we can ignore \mathbf{B} and focus solely on \mathbf{E} for our calculations; Eq.(2.16) provides the connection between the two.

3. Reflection and Transmission at Dielectric Interfaces

Now that we have defined some of the more trivial aspects of our problem, let us derive the Fresnel equations for the situation at hand. Figure 1 demonstrates the geometry we shall be working with; a wave with wave-vector \mathbf{k} is incident on a surface located at $z = 0$ at an angle θ_i , which gives rise to a reflected wave \mathbf{k}'' at an angle θ_r and a refracted wave \mathbf{k}' at an angle θ_r . Furthermore, we shall restrict our attention to two special cases of linearly polarized light. In the first of these, the electric field oscillates in a plane that is parallel to the plane of incidence, defined as the plane formed by all three wave rays (i.e. the

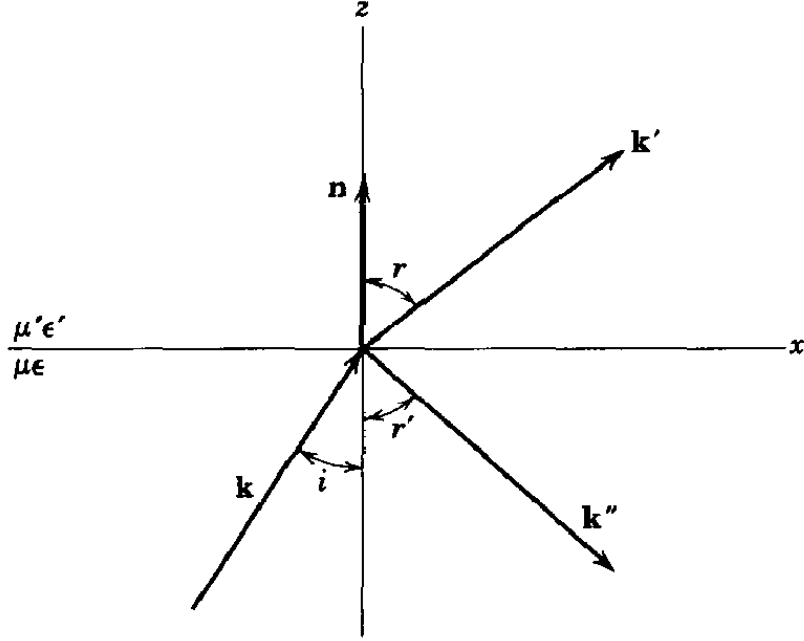


FIG. 1. Propagation of a wave into a medium with a different index of refraction, which gives rise to reflected and refracted waves. The y-axis points into the page for this geometry [1].

z-x plane in the figure above). Conversely, the magnetic field oscillates in a plane that is orthogonal to the plane of incidence, as can be seen from Eq.(2.16); such a wave is known as a Transverse Magnetic (TM) wave, or p-wave. When the electric field oscillates in a plane that is orthogonal to the plane of incidence, this wave is called a Transverse Electric (TE) wave, or s-wave. Note that we are intentionally leaving out more complicated behavior such as elliptically polarized light for the sake of brevity.

Using the superposition principle, we can imagine each of the two regions $\mu\epsilon$ and $\mu'\epsilon'$ as being composed of a combination of forward-moving and backward-moving waves. In the $\mu\epsilon$ region we have the following two waves:

$$\mathbf{E}(\mathbf{r}, t) = \mathbf{E}_0 e^{i(\mathbf{k} \cdot \mathbf{r} - \omega t)} \quad (3.1)$$

$$\mathbf{E}''(\mathbf{r}, t) = \mathbf{E}_0'' e^{i(\mathbf{k}'' \cdot \mathbf{r} - \omega t)} \quad (3.2)$$

whereas in the $\mu'\epsilon'$ region we have:

$$\mathbf{E}'(\mathbf{r}, t) = \mathbf{E}_0' e^{i(\mathbf{k}' \cdot \mathbf{r} - \omega t)} \quad (3.3)$$

Joining all three equations at the interface then gives:

$$\mathbf{E}_0 e^{i(\mathbf{k} \cdot \mathbf{r} - \omega t)} + \mathbf{E}_0'' e^{i(\mathbf{k}'' \cdot \mathbf{r} - \omega t)} = \mathbf{E}_0' e^{i(\mathbf{k}' \cdot \mathbf{r} - \omega t)} \quad (3.4)$$

Since the frequency is decided at the source, it is the same for all three waves. This leads to the three relations:

$$|\mathbf{k}| = |\mathbf{k}''| = \omega \sqrt{\mu \epsilon} \quad (3.5)$$

$$|\mathbf{k}'| = \omega \sqrt{\mu' \epsilon'} \quad (3.6)$$

From Eq.(3.4) and from the fact that the boundary conditions must hold for all \mathbf{r} and t , we see that the arguments of the exponents must all be equal at $z = 0$. Notice how this happens because the time-dependence is confined to a $e^{-i\omega t}$ harmonic. And so, at $z = 0$ we have the following conditions:

$$\mathbf{k} \cdot \mathbf{r} = \mathbf{k}'' \cdot \mathbf{r} = \mathbf{k}' \cdot \mathbf{r} \quad (3.7)$$

This implies:

$$|k||r| \cos \theta_i = |k''||r| \cos \theta_{r'} \quad \rightarrow \quad \theta_i = \theta_{r'} \quad (3.8)$$

We have thus recovered the law of reflection, demonstrating the fact that all three wave rays lay in the plane of incidence. We may now eliminate the y-components of the scalar products given above and keep only the k_x components so that $k_x = k_x'' = k_x'$. This gives:

$$k \sin \theta_i = k'' \sin \theta_{r'} = k' \sin \theta_r \quad (3.9)$$

so that:

$$k \sin \theta_i = k' \sin \theta_r \quad (3.10)$$

which is a re-statement of Snell's law, as can be seen by using Eq.(2.12), Eq.(3.5), and Eq.(3.6). Together, the laws of reflection and refraction dictate the kinematic properties of wave propagation. The *dynamic* properties are dictated by the interface, which we will now examine. The standard boundary conditions of the Maxwell equations state:

$$[\epsilon(\mathbf{E}_0 + \mathbf{E}_0'') - \epsilon' \mathbf{E}_0'] \cdot \mathbf{n} = 0 \quad (3.11)$$

$$[\mathbf{k} \times \mathbf{E}_0 + \mathbf{k}'' \times \mathbf{E}_0'' - \mathbf{k}' \times \mathbf{E}_0'] \cdot \mathbf{n} = 0 \quad (3.12)$$

$$(\mathbf{E}_0 + \mathbf{E}_0'' - \mathbf{E}_0') \times \mathbf{n} = 0 \quad (3.13)$$

$$\left[\frac{1}{\mu}(\mathbf{k} \times \mathbf{E}_0 + \mathbf{k}'' \times \mathbf{E}_0'') - \frac{1}{\mu'}(\mathbf{k}' \times \mathbf{E}_0') \right] \times \mathbf{n} = 0 \quad (3.14)$$

Notice how we have eliminated \mathbf{B} through Eq.(2.16). The first two of these equations state that the components of the electric and magnetic fields that are orthogonal to the interface experience a discontinuity and continuity at the interface, respectively; the latter two state that the parallel components of the electric and magnetic fields are continuous and discontinuous at the interface instead. Lets begin applying these boundary conditions to the case of a TM wave. The first condition yields:

$$\epsilon(E_0 \sin \theta_i + E_0'' \sin \theta_{r'}) = \epsilon' E_0' \sin \theta_r \quad (3.15)$$

Using Snell's law and the law of reflection then gives:

$$\sqrt{\frac{\mu' \epsilon'}{\mu \epsilon}}(E_0 + E_0'') = \frac{\epsilon'}{\epsilon} E_0' \quad (3.16)$$

This can be rearranged to obtain:

$$\sqrt{\frac{\epsilon}{\mu}}(E_0 + E_0'') = \frac{\epsilon'}{\mu'} E_0' \quad (3.17)$$

The second boundary condition gives nothing since there are no components of the magnetic fields that are perpendicular to the interface. The third boundary condition imposes restrictions on the parallel components of the electric field and so:

$$-E_0 \cos \theta_i + E_0'' \cos \theta_{r'} = -E_0' \cos \theta_r \quad (3.18)$$

Once again, the law of reflection simplifies this to:

$$(E_0 - E_0'') \cos \theta_i = E_0' \cos \theta_r \quad (3.19)$$

Lastly, the fourth boundary condition merely reproduces Eq.(3.17). We now proceed to solve Eq.(3.17) and Eq.(3.19) for the ratios E_0''/E_0 and E_0'/E_0 , otherwise known as the reflection and transmission coefficients [2]:

$$r \equiv \frac{E_0''}{E_0} = \frac{\frac{\mu}{\mu'} n'^2 \cos \theta_i - n \sqrt{n'^2 - n^2 \sin^2 \theta_i}}{\frac{\mu}{\mu'} n'^2 \cos \theta_i + n \sqrt{n'^2 - n^2 \sin^2 \theta_i}} \quad (3.20)$$

$$t \equiv \frac{E_0'}{E_0} = \frac{2nn' \cos \theta_i}{\frac{\mu}{\mu'} n'^2 \cos \theta_i + n \sqrt{n'^2 - n^2 \sin^2 \theta_i}} \quad (3.21)$$

In this paper we will only consider non-magnetic materials for which $\mu = \mu_0$ which immediately yields $\mu/\mu' = 1$. Furthermore, the relation $n \sqrt{n'^2 - n^2 \sin^2 \theta_i} = nn' \cos \theta_r$ allows us to simplify these expressions to:

$$r_p = \frac{n_f \cos \theta_i - n_i \cos \theta_f}{n_f \cos \theta_i + n_i \cos \theta_f} \quad (3.22)$$

$$t_p = \frac{2n_i \cos \theta_i}{n_f \cos \theta_i + n_i \cos \theta_f} \quad (3.23)$$

In these expression we have re-labeled $n, n' \rightarrow n_i, n_f$ and $\theta_i, \theta_r \rightarrow \theta_i, \theta_f$ for clarity. The p subscript denotes a p-wave (i.e. a TM wave). Similar calculations for an s-wave give the following relations:

$$r_s = \frac{n_i \cos \theta_i - n_f \cos \theta_f}{n_i \cos \theta_i + n_f \cos \theta_f} \quad (3.24)$$

$$t_s = \frac{2n_i \cos \theta_i}{n_i \cos \theta_i + n_f \cos \theta_f} \quad (3.25)$$

Equations (3.22 - 3.25) constitute the core of our theory, and we shall use them to derive the behavior of light as it propagates through multiple interfaces.

4. The Transfer Matrix Method

Having derived the Fresnel equations, our final step is to generalize them for the case where multiple dielectric interfaces are present. We imagine a scenario similar to Figure 1 but with N layers of different thicknesses d_j and indices of refraction n_j stacked along the z -axis so that they lay parallel to the y - x plane. This means that the dielectric layers can be organized as follows:

$$\begin{aligned} n(z) = & n_0, & z < z_0 \\ & n_1, & z_0 < z < z_1 \\ & n_2, & z_1 < z < z_2 \\ & \vdots \\ & n_N, & z_{N-1} < z \end{aligned} \quad (4.1)$$

From this geometry, it is easy to see that the z -component of \mathbf{k} in each layer becomes $k_{zj} = (2\pi n_j / \lambda_0) \cos \theta_j$ since, as before, the wave rays are confined to the plane of incidence. Let us now consider the total electric field of a TE (s-wave) within each individual layer, as s-waves tend to satisfy simpler boundary conditions. The superposition principle allows us to once again decompose $\mathbf{E}(\mathbf{r}, t)$ into different components; one moving forward in the

positive z-direction, and another moving backwards in the -z-direction [3][4]:

$$\begin{aligned}
E(\mathbf{r}, t) &= E_F e^{i(\mathbf{k}_F \cdot \mathbf{r} - \omega t)} + E_B e^{i(\mathbf{k}_B \cdot \mathbf{r} - \omega t)} \\
&= E_F e^{i(k_x x + k_{zj} z - \omega t)} + E_B e^{i(k_x x - k_{zj} z - \omega t)} \\
&= E_F e^{ik_{zj} z} + E_B e^{-ik_{zj} z}
\end{aligned} \tag{4.2}$$

In the last line we absorbed $e^{ik_x x}$ into the complex amplitudes and dropped the time dependence for clarity. We can recast this equation into a more illuminating form if we consider how the state of the light beam changes as it interacts with each layer j :

$$\begin{aligned}
E'_F &= E_F e^{ik_{zj} z} \\
E'_B &= E_B e^{-ik_{zj} z}
\end{aligned} \tag{4.3}$$

Interestingly enough, this resembles a two-state matrix equation of the form:

$$E' = T_j E \tag{4.4}$$

for some matrix T_j . Thus, as a wave propagates within a material of thickness d_j , the corresponding propagation matrix can be obtained through inspection of Eqs.(4.3):

$$T_j = \begin{pmatrix} \exp ik_{zj} d_j & 0 \\ 0 & \exp -ik_{zj} d_j \end{pmatrix} = \begin{pmatrix} \exp i\Phi_j & 0 \\ 0 & \exp -i\Phi_j \end{pmatrix} \tag{4.5}$$

where we have defined the phase angle as $\Phi_j = k_{zj} d_j$. As helpful as the T_j matrix is, it does not provide the entire picture. We must now investigate how the state of a light wave changes as it passes *through* an interface. The difference can be subtle, so the reader is advised to give it some thought. As was the case for wave reflection and refraction, boundary conditions will once again dictate the dynamics of our system. Since we are dealing with s-waves, the relevant boundary conditions are those for tangential \mathbf{E} and \mathbf{B} , which we already acquired in the previous section:

$$\begin{aligned}
E_{Fi} + E_{Bi} &= E_{Fj} + E_{Bj} \\
n_i \cos \theta_i (E_{Fi} - E_{Bi}) &= n_j \cos \theta_j (E_{Fj} - E_{Bj})
\end{aligned} \tag{4.6}$$

The difference in the indices between the left-hand and right-hand sides explicitly demonstrates the propagation of light through an interface. In matrix form these equations read:

$$\begin{pmatrix} 1 & 1 \\ n_i \cos \theta_i & -n_i \cos \theta_i \end{pmatrix} \begin{pmatrix} E_{Fi} \\ E_{Bi} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ n_j \cos \theta_j & -n_j \cos \theta_j \end{pmatrix} \begin{pmatrix} E_{Fj} \\ E_{Bj} \end{pmatrix} \tag{4.7}$$

Rearranging this expression yields:

$$\begin{pmatrix} E_{Fj} \\ E_{Bj} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ n_j \cos \theta_j & -n_j \cos \theta_j \end{pmatrix}^{-1} \begin{pmatrix} 1 & 1 \\ n_i \cos \theta_i & -n_i \cos \theta_i \end{pmatrix} \begin{pmatrix} E_{Fi} \\ E_{Bi} \end{pmatrix}$$

$$\begin{pmatrix} E_{Fj} \\ E_{Bj} \end{pmatrix} = T_j^{-1} T_i \begin{pmatrix} E_{Fi} \\ E_{Bi} \end{pmatrix} = T_{ji} \begin{pmatrix} E_{Fi} \\ E_{Bi} \end{pmatrix} \quad (4.8)$$

The transfer matrix T_{ji} is now trivial to find. The end result is:

$$T_{ji} = \frac{1}{t_{ji}} \begin{pmatrix} 1 & r_{ji} \\ r_{ji} & 1 \end{pmatrix} \quad (4.9)$$

Evidently, the elements of T_{ji} are nothing but the Fresnel coefficients we calculated in the previous section. Combining this interface matrix with the layer-propagation matrix Eq.(4.5) gives the *total* transfer matrix of our system T . For an arbitrary amount of layers, T is obtained through ordinary matrix multiplication of each individual element:

$$T = T_{N(N-1)} T_{N-1} \dots T_{32} T_2 T_{21} T_1 T_{10} \quad (4.10)$$

The order of these matrices is to be noted, as this configuration gives the total output from some initial input.

Now that we have all the tools required, we can finally make progress and perform some trial computations.

5. Numerical Implementations of the TMM in Python

To put all the tools we developed in the previous sections, we carry out a computation in Python that showcases some of the more significant aspects of the TMM. The following code was adapted from Byrnes module on the TMM [5]. We start by defining a function that distinguishes whether a given wave is traveling forward or backwards in a given medium of complex reflectivity. To achieve this, we perform two **assert** statements that ensure that both parts of the index of refraction n are positive. As we discussed before, a negative extinction coefficient would lead to exponential growth which we are avoiding. Now, the wave-number in a given layer $k_{zj} = (2\pi n_j / \lambda_0) \cos \theta_j$ generally points in the direction of propagation in isotropic media, so we can use it to further refine the forward direction

criteria. We thus create an **if** conditional that returns true if the extinction coefficient is above a flotation point value and false otherwise. Since decay points if the forward direction, a positive extinction coefficient is a good metric for this problem:

```
def theta_forward(n, theta):

    assert n.real >= 0
    assert n.imag >= 0

    n_cos_theta = n * np.cos(theta)

    if np.absolute(n_cos_theta.imag) > 100 * EPSILON:
        value = (n_cos_theta.imag > 0)
    else:
        value = (n_cos_theta.real > 0)

return value
```

We then create a simple function that returns a refracted angle from an incident angle based on Snell's Law Eq.(3.10) and uses the previous function for forward propagation. If the angle fails to point in the forward direction, this function returns a backwards angle which is simply $(\pi - \theta)$:

```
def snells_law(n_1, n_2, theta_1):

    theta_2 = arcsin(n_1 * np.sin(theta_1) / n_2)

    if theta_forward(n_2, theta_2):
        return theta_2
    else:
        return np.pi - theta_2
```

Moving onward, the next function defines a list of refracted angles within the dielectric structure based on the incident angle on the first layer. The **if not** conditionals assure that the wave propagates in the forward direction for the first layer **n_list[0]** and for the last layer **n_list[-1]** as well:

```

def snells_law_list(n_list , theta_incident):

    theta_refracted_list = arcsin(n_list[0]
                                   * np.sin(theta_incident) / n_list)

    if not theta_forward(n_list[0], theta_refracted_list[0]):
        theta_refracted_list[0] = np.pi - theta_refracted_list[0]

    if not theta_forward(n_list[-1], theta_refracted_list[-1]):
        theta_refracted_list[-1] = np.pi - theta_refracted_list[-1]

    return theta_refracted_list

```

Now comes the part where we define the reflection amplitudes for s-waves and p-waves directly from the Fresnel equations Eqs.(3.22 - 3.25). The **if** loops take care of both situations:

```

def reflection_amplitude(polarization , n_i , n_f , theta_i , theta_f):

    if polarization == 's':
        return ((n_i * np.cos(theta_i) - n_f * np.cos(theta_f)) /
                (n_i * np.cos(theta_i) + n_f * np.cos(theta_f)))

    elif polarization == 'p':
        return ((n_f * np.cos(theta_i) - n_i * np.cos(theta_f)) /
                (n_f * np.cos(theta_i) + n_i * np.cos(theta_f)))

    else:
        raise ValueError("Polarization must be 's' or 'p'")

def amplitude_r(polarization , n_i , n_f , theta_i , theta_f):

    r = reflection_amplitude(polarization , n_i , n_f , theta_i , theta_f)

    return reflected_power(r)

```

Similarly, the transmission amplitudes can be computed for s-waves and p-waves as well. The **amplitude_r** and **amplitude_t** functions serve to return values that can be fed into the computation of the reflected and transmitted power:

```

def transmission_amplitude(polarization , n_i , n_f , theta_i , theta_f):

```

```

    if polarization == 's':
        return ((2 * n_i * np.cos(theta_i)) /
                (n_i * np.cos(theta_i) + n_f * np.cos(theta_f)))
    elif polarization == 'p':
        return ((2 * n_i * np.cos(theta_i)) /
                (n_f * np.cos(theta_i) + n_i * np.cos(theta_f)))
    else:
        raise ValueError("Polarization must be 's' or 'p'")

def amplitude_t(polarization, n_i, n_f, theta_i, theta_f):

    t = transmission_amplitude(polarization, n_i, n_f, theta_i, theta_f)
    return transmitted_power(polarization, t, n_i, n_f, theta_i, theta_f)

```

It is now straightforward to define functions that return the **reflected_power** and the **transmitted_power** functions based on Eqs.(78798686664343):

```

def reflected_power(r):

    return np.absolute(r)**2

def transmitted_power(polarization, t, n_i, n_f, theta_i, theta_f):

    if polarization == 's':
        return np.absolute(t**2) * (((n_f * np.cos(theta_f)).real)
                                     / (n_i * np.cos(theta_i)).real)
    elif polarization == 'p':
        return np.absolute(t**2)
               * (((n_f * np.conj(np.cos(theta_f))).real)
                 / (n_i * np.conj(np.cos(theta_i))).real)
    else:

```

```
raise ValueError("Polarization must be 's' or 'p'")
```

Lastly we define a 2 x 2 **matrix_array** function which we will use as a template for our transfer matrices:

```
def matrix_array(a, b, c, d, dtype=float):

    template_array = np.empty((2,2), dtype=dtype)

    template_array[0,0] = a
    template_array[0,1] = b
    template_array[1,0] = c
    template_array[1,1] = d

    return template_array
```

At long last, we can piece all the these functions together to build our TMM algorithm. Lets discuss some of its important features. The first block of code defines a series of tests meant to guide a user's inputs. For example, the first **if** conditional safeguards against the input of more than one incident wavelength or angle, while the second **if** conditional ensures that the dimensionality of the **n_list** and **d_list** arrays does not exceed one. Furthermore, it also returns an error if we have more indices of refraction values than dielectric layers and vice-versa. Next come a series of **assert** statements that verify that the first layer **d_list[0]** and the second layer **d_list[-1]** are of infinite thickness, and that our incident angle/index of refraction describe physical values:

```
def transfer_matrix_method(polarization, n_list, d_list, theta_incident,
                           wavelength):

    n_list = np.array(n_list)

    d_list = np.array(d_list, dtype=float)

    if ((hasattr(wavelength, 'size') and wavelength.size > 1)
        or (hasattr(theta_incident, 'size') and theta_incident.size > 1)):
        raise ValueError("Only 1 wavelength.....")
```

```

if ((n_list.ndim != 1)
      or (d_list.ndim != 1)
      or (n_list.size != d_list.size)):
raise ValueError("Problem with n_list or d_list!")

assert d_list[0] == d_list[-1] == np.inf
assert np.absolute((n_list[0]*np.sin(theta_incident)).imag)<100*EPSILON
assert theta_forward(n_list[0], theta_incident)

```

Once all of these checks have passed, we move on to the implementation of the TMM. The first four lines recast our relevant parameters as lists with values for each layer. Afterwards, we create arrays for our reflection and transmission amplitudes **r_list** and **t_list** and loop then through all layers. Finally, we create two matrix lists **Transfer1_list** and **Transfer2_list** which are precisely the T_j and T_{ji} matrices we defined in Eq.(4.5) and Eq.(4.9):

```

num_layers = n_list.size

theta_list = snells_law_list(n_list, theta_incident)

k_list = 2 * np.pi * n_list * np.cos(theta_list) / wavelength

phase_angle = k_list * d_list

r_list = np.zeros((num_layers, num_layers), dtype=complex)
t_list = np.zeros((num_layers, num_layers), dtype=complex)
for i in range(num_layers-1):
    r_list[i,i+1] = reflection_amplitude(polarization, n_list[i],
                                         n_list[i+1], theta_list[i], theta_list[i+1])
    t_list[i,i+1] = transmission_amplitude(polarization, n_list[i],
                                           n_list[i+1], theta_list[i], theta_list[i+1])

Transfer1_list = np.zeros((num_layers, 2, 2), dtype=complex)

```

```

for i in range(1, num_layers-1):
    Transfer1_list[i] = (1/t_list[i,i+1]) * np.dot(
        matrix_array(np.exp(-1j * phase_angle[i]),
            0, 0, np.exp(1j * phase_angle[i]), dtype=complex),
        matrix_array(1, r_list[i,i+1], r_list[i,i+1],
            1, dtype=complex))

Transfer2_list = matrix_array(1, 0, 0, 1, dtype=complex)
for i in range(1, num_layers-1):
    Transfer2_list = np.dot(Transfer2_list, Transfer1_list[i])

Transfer2_list = np.dot(matrix_array(1, r_list[0,1], r_list[0,1], 1,
    dtype=complex)/t_list[0,1], Transfer2_list)

```

Finally, we construct the reflection and transmission amplitudes/powers from our matrices and then end the algorithm by returning values for all variables. An extra function that is included is the **unpolarized** function which, as its name states, returns the reflected/transmitted powers for unpolarized light by averaging the s-waves with the p-waves:

```

r = Transfer2_list[1,0]/Transfer2_list[0,0]
t = 1/Transfer2_list[0,0]

R = reflected_power(r)
T = transmitted_power(polarization, t,
    n_list[0], n_list[-1], theta_incident, theta_list[-1])

return {'r': r, 't': t, 'R': R, 'T': T,
    'k_list': k_list, 'theta_list': theta_list,
    'polarization': polarization, 'n_list': n_list, 'd_list': d_list,
    'theta_incident': theta_incident, 'wavelength': wavelength}

def unpolarized(n_list, d_list, theta_incident, wavelength):

```



```

s_data = transfer_matrix_method('s', n_list, d_list,
    theta_incident, wavelength)

p_data = transfer_matrix_method('p', n_list, d_list,
    theta_incident, wavelength)

R = (s_data['R'] + p_data['R']) / 2.
T = (s_data['T'] + p_data['T']) / 2.

return {'R': R, 'T': T}

```

6. Conclusion

The transfer matrix method outlined in this paper is a deceptively simple yet versatile instrument that has proven to be indispensable in a diverse array of scientific fields. Although we focused on a narrow range of its applications, we nevertheless arrived at conclusions that seemed to be acquired almost out of sheer luck. Indeed it is quite surprising that this method works at all, given the complicated interactions that take place between light and matter. Please refer to the accompanying Jupyter Notebook for a more detailed overview of some of the computations that the TMM can perform.

-
- [1] J. Jackson, *Classical Electrodynamics* (John Wiley & Sons, Inc., 1999).
 - [2] E. Hecht, *Optics* (Pearson, 2002).
 - [3] M. Born, and E. Wolf, *Principles of Optics* (Pergamon Press, 1986).
 - [4] M. Claudia Troparevsky, Adrian S. Sabau, Andrew R. Lupini, and Zhenyu Zhang, "Transfer-matrix formalism for the calculation of optical response in multilayer systems: from coherent to incoherent interference," *Opt. Express* 18, 24715-24721 (2010)
 - [5] <https://pypi.org/project/tmm/>