Project

Impact of Broadband Antireflection Coating Design on Solar Power Production

ELG 3106 - Electromagnetic Engineering

Fall 2023

School of Electrical Engineering and Computer Science

University of Ottawa

Tristan Ruel 300272156

Course Coordinator: Dr. Henry Schriemer

December 4th, 2023

Table of Contents

Index

Executive Summary

This project presents a comprehensive analysis of anti-reflection coatings for solar cells, exploring both double-layer and triple-layer configurations. Employing the Transfer Matrix Method (TMM) and MATLAB for simulations, this study examines the efficiency of these anti-reflection coatings in converting solar energy under the assumption of 100% conversion efficiency across all wavelengths. The focus is on the wavelength-dependent transmissivity of the anti-reflection coatings and the quest to understand whether traditional broadband anti-reflection coatings achieve optimal power production.

Key insights were gained in two parts. First, the analysis of double-layer coatings using the TMM revealed that optimizing anti-reflection coatings for a central wavelength of 650 nm, despite achieving a refractive index of around 2.62, does not yield maximum power across the solar spectrum. Graphical analysis confirmed these findings, showing limited reflectivity reduction at this central wavelength.

The second part extended this analysis to triple-layer coatings. Here, the optimal refractive index for the middle layer was approximately 2.36 for minimized reflection at the central wavelength. However, this optimization too was wavelength-specific and did not cover the full solar spectrum.

A crucial finding of this study is the importance of optimizing anti-reflection coatings across the entire solar spectrum rather than focusing on a single wavelength. The research demonstrated that changes in refractive indices or layer thicknesses could significantly enhance total power transmission, a concept validated through MATLAB-based optimization algorithms.

However, the project faced limitations due to its theoretical and computational nature, not accounting for real-world variables such as atmospheric effects and temperature variations as well as utilizing two very unrealistic assumptions, 100% of light being converted into electrical energy, and the light is perfectly incident to the solar cell. Furthermore, the practical implementation and testing of these anti-reflection coatings, along with considerations of their material properties, remain areas for future exploration.

This study emphasized that while optimizing anti-reflection coatings for a specific central wavelength can reduce reflectivity, it is not sufficient for maximizing solar power production.

Introduction

The relentless pursuit of energy sustainability has thrust solar power technology into new light. A critical component in enhancing the efficiency of these solar cells is the optimization of anti-reflective coatings. These coatings are instrumental in reducing light reflection and increasing the absorption of incident light, thereby boosting the overall power output of solar cells. In this context, our report delves into the analysis of both double and triple-layer anti-reflective coatings, focusing on identifying the refractive index configurations that optimize power transfer across the solar spectrum.

In a world increasingly reliant on renewable energy sources, such findings in solar cell technology are not just beneficial but essential. The ability of these coatings to maximize light absorption directly correlates to the effectiveness and viability of solar energy as a sustainable energy solution. Utilising the Transfer Matrix Method (TMM) and MATLAB simulations, this study aims to unravel the complexities of these coatings and their impact on solar energy conversion efficiency.

This report is structured into two significant parts: the first part scrutinises double-layer coatings and triple-layer, evaluating their effectiveness in light absorption and power conversion at a central wavelength of 650 nm their performance across a broader wavelength spectrum. The central theme of this study is to challenge the traditional notion that optimizing anti-reflection coatings at a specific central wavelength is sufficient for maximizing solar power production. We propose that considering the entire solar spectrum is vital for the optimization of these coatings.

Our findings reveal that while optimizing the refractive index of anti-reflection coatings at a central wavelength does reduce reflectivity, it falls short in harnessing the maximum potential of solar power production. This revelation underscores the need for a broader spectrum approach in designing anti-reflection coatings. Despite the theoretical and computational nature of our study, which does not account for real-world variables such as atmospheric effects and inefficiencies, the insights gained provide a substantial foundation for future research. This report presents a comprehensive analysis of these coatings, challenging conventional methods, and advocating for a spectrum-wide approach to maximize solar power production.

Theory

The wavelength distribution of light normally incident on the system is given by an idealised solar spectral irradiance, known as “blackbody” irradiance which neglects the filtering effects of the earth’s atmosphere is given by:

$$I(\lambda) = \frac{6.16\times10^{15}}{\lambda^{5}(e^{\frac{2484}{\lambda}} - 1)}$$

Where I has units of power per unit area per unit wavelength ($W/m^2/nm$) and an area of $1m^2$ is assumed. Additionally, it is assumed that regardless of wavelength λ, the solar cell transforms all sunlight incident into electricity at 100% efficiency. Hence, since the antireflection coating has a transmissivity of $T(\lambda)$, the electrical power production is given by:

$$P = \int_{\lambda_1}^{\lambda_2} T(\lambda)I(\lambda)d\lambda$$

Where the range of the solar spectrum, 200 nm to 2200 nm, defines the limits of integration λ_1 and λ_2.

This study aims to show that the seemingly straightforward method for designing broadband antireflection coatings might not always result in optimal power production, and to explore the reasons behind this. The strategy involves the numerical integration of the above equation, and employing the Transfer Matrix Method (TMM) to simulate the wavelength-dependent transmissivity of the planar multilayer stack forming the antireflection coating on a solar cell in MATLAB.

**The above section was created using the works of Dr. Henry Schriemer (Schriemer, 2023)*

The Transfer Matrix Method

Figure 1.0 below displays an arbitrary multilayer structure, marked with the electric field components at every interface. Here, the incident wave is depicted as travelling to the right. Layer identification is done through subscripts, while the '+' and '−' signs are used to identify forward and backward waves, respectively. Additionally, a prime notation is utilised to differentiate waves on the right side of an interface from those on the left.

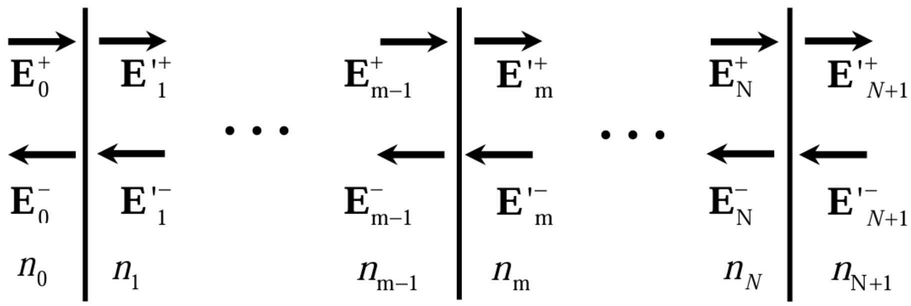

Figure 1.0 - The Transfer Matrix Method Illustrated by an Arbitrary Multilayer

The multilayer structure consists of N layers, not including the unbounded medium on the left, which in this instance is air (n_0 = 1), and to the right, silicon in our case, along with N+1 interfaces. The electric field vectors E on both sides of an interface are governed by boundary conditions, enabling a clear depiction of the m^{th} interface with a 2x2 matrix. The relationship between the components of the field is given by:

$$\begin{pmatrix} E_{m-1}^{+} \\ E_{m-1}^{-} \end{pmatrix} = Q_{m-1,\,m} \begin{pmatrix} E'^{+}_{m} \\ E'^{-}_{m} \end{pmatrix}$$

Where the dynamical matrix is given by:

$$Q_{m-1,\,m} = \frac{1}{\tau_{m-1,\,m}} \begin{bmatrix} 1 & \Gamma_{m-1,\,m} \\ \Gamma_{m-1,\,m} & 1 \end{bmatrix}$$

Which is defined in terms of the usual reflection and transmission coefficients:

$$\Gamma_{m-1,m} = \frac{n_{m-1} - n_m}{n_{m-1} + n_m} \; and \; \tau_{m-1,m} = \frac{2n_{m-1}}{n_{m-1} + n_m}$$

Additionally, the field components on the left and right-hand sides of the *m*th layer are related by the propagation matrix:

$$\begin{pmatrix} E'^{+}_{m} \\ E'^{-}_{m} \end{pmatrix} = P_m \begin{pmatrix} E^{+}_{m} \\ E^{-}_{m} \end{pmatrix}$$

Where,

$$P_m = \begin{bmatrix} \exp{(j\delta_m)} & 0 \\ 0 & \exp{(-j\delta_m)} \end{bmatrix}$$

Where the phase thickness is $\delta_m = \frac{2\pi}{\lambda} n_m d_m$ for the m^{th} layer of physical thickness d_m and λ is the wavelength in free-space. The previous transformations are reapplied for N layers and N+1 interfaces, culminating in a product of (N+1) 2x2 matrices. This product effectively links the total field in the left-hand unbounded medium with the total field in the right-hand unbounded medium.

$$\begin{pmatrix} E^{+}_{0} \\ E^{-}_{0} \end{pmatrix} = T \begin{pmatrix} E'^{+}_{N+1} \\ E'^{-}_{N+1} \end{pmatrix}$$

Where T is the system transfer matrix:

$$T = \begin{bmatrix} T_{1,1} & T_{1,2} \\ T_{2,1} & T_{2,2} \end{bmatrix} = Q_{0,1} \prod_{m=1}^{N} P_m Q_{m,m+1}$$

Where the reflection and transmission coefficients are:

$$\Gamma = \frac{E_r}{E_i} = \frac{T_{2,1}}{T_{1,1}} \; and \; \tau = \frac{E_t}{E_i} = \frac{1}{T_{1,1}}$$

For a lossless, nonmagnetic, medium, we have:

$$|\Gamma|^2 + |\tau|^2 \left(\frac{n_{N+1}}{n_0} \right) = 1$$

We define the reflectivity and transmissivity as:

$$R = |\Gamma|^2 \; and \; T = |\tau|^2 \left(\frac{n_{n+1}}{n_0} \right)$$

**The above section used the Hypatia Create software to format the equations*
***The above section was created using the works of Dr. Henry Schriemer (Schriemer, 2023)*

Part I: The Double-Layer Coating

A. Calculating Center Wavelength Reflectivity and Power Transmission Without Anti-Reflective Layer.

Knowing that the refractive index of the cell is given by n_{cell} = 3.5 and the refractive index of air is given by n_0 = 1, we can calculate the power transmitted for a center wavelength of λ_C = 650 nm as follows:

$$\Gamma = \frac{n_{cell} - n_0}{n_{cell} + n_0}$$

$$\Gamma = \frac{3.5-1}{3.5+1} = 0.5555... \approx 0.56$$

$$R = |\Gamma|^2 = 0.3086 \approx 0.31$$

$$P = \frac{(1-0.3086)(6.16\times10^{15})}{(650)^5(e^{\frac{2484}{650}}-1)} = 0.8216 \approx 0.82\frac{W}{m^2}$$

B. Double-Layer Coating TTM Approach.

In the following analysis, we will employ the Transfer Matrix Method (TMM) for an analytical calculation of both transmissivity, Γ, and reflectivity, τ, across the multilayer system depicted in Figure 1.1, utilising their respective refractive indices.

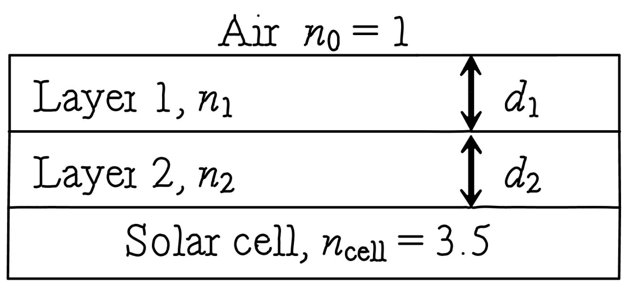

Figure 1.1 - Schematic of the Double-Layer Anti-Reflective Coating System

As observed in Figure 1.2, which illustrates light wave interactions with the system, we recognize that within this two-layer setup, two propagating matrices P and three dynamic matrices Q are expected. this leads us to the formulation of the transfer matrix T, as detailed below:

$$T = Q_{0,1}P_1Q_{1,2}P_2Q_{2,3}$$

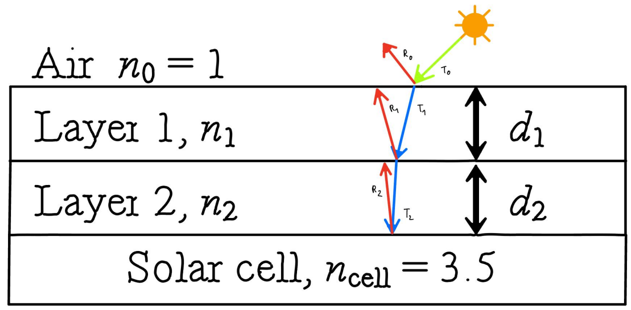

Figure 1.2 - Schematic of the Interaction of Light in Double-Layer Anti-Reflective Coating System

Hence, the transmissivity, Γ, and reflectivity, τ, coefficients across the double-layer system can be calculated using the refractive indices, as seen below:

$$\Gamma_{0,1} = \frac{n_0 - n_1}{n_0 + n_1} \quad \tau_{0,1} = \frac{2n_0}{n_0 + n_1}$$

$$\Gamma_{1,2} = \frac{n_1 - n_2}{n_1 + n_2} \quad \tau_{1,2} = \frac{2n_1}{n_1 + n_2}$$

$$\Gamma_{2,3} = \frac{n_2 - n_3}{n_2 + n_3} \quad \tau_{2,3} = \frac{2n_2}{n_2 + n_3}$$

The transmissivity, Γ, and reflectivity, τ, coefficients for the dynamic matrices Q can be calculated by:

$$Q_{0,1} = \frac{1}{\tau_{0,1}} \begin{bmatrix} 1 & \Gamma_{0,1} \\ \Gamma_{0,1} & 1 \end{bmatrix}$$

$$Q_{1,2} = \frac{1}{\tau_{1,2}} \begin{bmatrix} 1 & \Gamma_{1,2} \\ \Gamma_{1,2} & 1 \end{bmatrix}$$

$$Q_{2,3} = \frac{1}{\tau_{2,3}} \begin{bmatrix} 1 & \Gamma_{2,3} \\ \Gamma_{2,3} & 1 \end{bmatrix}$$

Additionally, the phase thickness δ_m is given by:

$$\delta_m = \frac{2\pi n_m d_m}{\lambda} \text{ where } d_m = \frac{\lambda_c}{4n_m}$$

The thickness of each layer of the anti-reflective coating is given by d_m. It is stated that $\delta_m = \frac{\pi}{2}$ at the center wavelength $\lambda_C = 650$ nm.[1]

Hence, two propagation matrices P are calculated using:

$$P_m = \begin{bmatrix} \exp(j\delta_m) & 0 \\ 0 & \exp(-j\delta_m) \end{bmatrix}$$

P_1 and P_2 are given by the following matrices:

$$P_1 = \begin{bmatrix} \exp\left(j\delta_1\right) & 0 \\ 0 & \exp\left(-j\delta_1\right) \end{bmatrix}$$

$$P_2 = \begin{bmatrix} \exp\left(j\delta_2\right) & 0 \\ 0 & \exp\left(-j\delta_2\right) \end{bmatrix}$$

We can implement these formulas into the transfer matrix T.

**The above section used the Hypatia Create software to format the equations*

[1] H. Schriemer, "ELG3106 2023 Project - ARC notes" University of Ottawa, Ottawa, Ontario, Canada, Accessed: Nov. 25, 2023

C. Analytical Approach of Double-Layer Anti-Reflection Coating at Central Wavelength.

Taking the phase thickness δ_m at the central wavelength λ_C of 650 nm, we have:

$$\delta_m = \frac{2\pi n_m}{\lambda_c} \times \frac{\lambda_c}{4n_m} = \frac{\pi}{2}$$

We now calculate matrices P_1 and P_2:

$$P_1 = \begin{bmatrix} \exp\left(j\frac{\pi}{2}\right) & 0 \\ 0 & \exp\left(-j\frac{\pi}{2}\right) \end{bmatrix} = j\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$P_2 = \begin{bmatrix} \exp\left(j\frac{\pi}{2}\right) & 0 \\ 0 & \exp\left(-j\frac{\pi}{2}\right) \end{bmatrix} = j\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

The dynamic matrices Q can be calculated by:

$$Q_{0,1} = \frac{1}{\tau_{0,1}}\begin{bmatrix} 1 & \Gamma_{0,1} \\ \Gamma_{0,1} & 1 \end{bmatrix}$$

$$Q_{1,2} = \frac{1}{\tau_{1,2}}\begin{bmatrix} 1 & \Gamma_{1,2} \\ \Gamma_{1,2} & 1 \end{bmatrix}$$

$$Q_{2,3} = \frac{1}{\tau_{2,3}}\begin{bmatrix} 1 & \Gamma_{2,3} \\ \Gamma_{2,3} & 1 \end{bmatrix}$$

We now calculate the transfer matrix T:

$$T = Q_{0,1} P_1 Q_{1,2} P_2 Q_{2,3}$$

We now implement P_1, P_2, $Q_{0,1}$, $Q_{1,2}$, and $Q_{2,3}$ into the transfer matrix T:

$$T = \frac{j^2}{\tau_{0,1}\tau_{1,2}\tau_{2,3}} \begin{bmatrix} 1 & \Gamma_{0,1} \\ \Gamma_{0,1} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & \Gamma_{1,2} \\ \Gamma_{1,2} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & \Gamma_{2,3} \\ \Gamma_{2,3} & 1 \end{bmatrix}$$

$$T = \begin{bmatrix} 1 - \Gamma_{0,1}\Gamma_{1,2} & \Gamma_{0,1} - \Gamma_{1,2} \\ \Gamma_{0,1} - \Gamma_{1,2} & 1 - \Gamma_{0,1}\Gamma_{1,2} \end{bmatrix} \begin{bmatrix} 1 & \Gamma_{2,3} \\ \Gamma_{2,3} & 1 \end{bmatrix}$$

$$T = \begin{bmatrix} 1 - \Gamma_{0,1}\Gamma_{1,2} + (\Gamma_{0,1} - \Gamma_{1,2})\Gamma_{2,3} & (1 - \Gamma_{0,1}\Gamma_{1,2})\Gamma_{2,3} + \Gamma_{0,1} - \Gamma_{1,2} \\ \Gamma_{0,1} - \Gamma_{1,2} + (1 - \Gamma_{0,1}\Gamma_{1,2})\Gamma_{2,3} & (\Gamma_{0,1} - \Gamma_{1,2})\Gamma_{2,3} + 1 - \Gamma_{0,1}\Gamma_{1,2} \end{bmatrix}$$

We now calculate the reflection coefficient R:

$$R = |\Gamma|^2 = \left|\frac{T_{2,1}}{T_{1,1}}\right|^2 = \left|\frac{\Gamma_{0,1} - \Gamma_{1,2} + (1 - \Gamma_{0,1}\Gamma_{1,2})\Gamma_{2,3}}{1 - \Gamma_{0,1}\Gamma_{1,2} + (\Gamma_{0,1} - \Gamma_{1,2})\Gamma_{2,3}}\right|^2$$

In order to find n_2, we extrapolate from the following:

$$T = \frac{T_{2,1}}{T_{1,1}}$$

Where:

$$T_{1,1} = 1 - \Gamma_{0,1}\Gamma_{1,2} + \left(\Gamma_{0,1} - \Gamma_{1,2}\right)\Gamma_{2,3}$$
$$T_{2,1} = \Gamma_{0,1} - \Gamma_{1,2} + \left(1 - \Gamma_{0,1}\Gamma_{1,2}\right)\Gamma_{2,3}$$

We can derive the following relationship:

$$n_0 n_2^2 = n_{cell} n_1^2$$

$$n_2 = \sqrt{\frac{n_{cell} n_1^2}{n_0}}$$

For $n_0 = 1$:

$$n_2 = \sqrt{n_{cell} n_1^2}$$

**The above section used the Hypatia Create software to format the equations*

D. Minimising Reflectivity for Double-Layer at Central Wavelength.

In order to find the relationship between refractive indices to minimize the reflectivity, $\Gamma = 0$, at the central wavelength λ_C of 650 nm, we extrapolate from the following:

$$\Gamma = \frac{T_{2,1}}{T_{1,1}} = 0$$

Where:

$$T_{2,1} = \Gamma_{0,1} - \Gamma_{1,2} + \left(1 - \Gamma_{0,1}\Gamma_{1,2}\right)\Gamma_{2,3} = 0$$

We can now use the previous relationship:

$$n_0 n_2^2 = n_{cell} n_1^2$$

$$n_0 = 1, n_1 = 1.4, n_{cell} = 3.5$$

$$n_2 = \sqrt{\frac{n_{cell} n_1^2}{n_0}} = 1.4\sqrt{3.5} = 2.6191 \approx 2.62$$

Hence, we calculated that the n_2 value to minimize reflectivity for a double-layer coating at central wavelength λ_C of 650 nm is approximately 2.62. We can verify this calculation using MATLAB, the output of the code named "calculate_n_2_for_2_layer" is found in Figure 1.3 below.

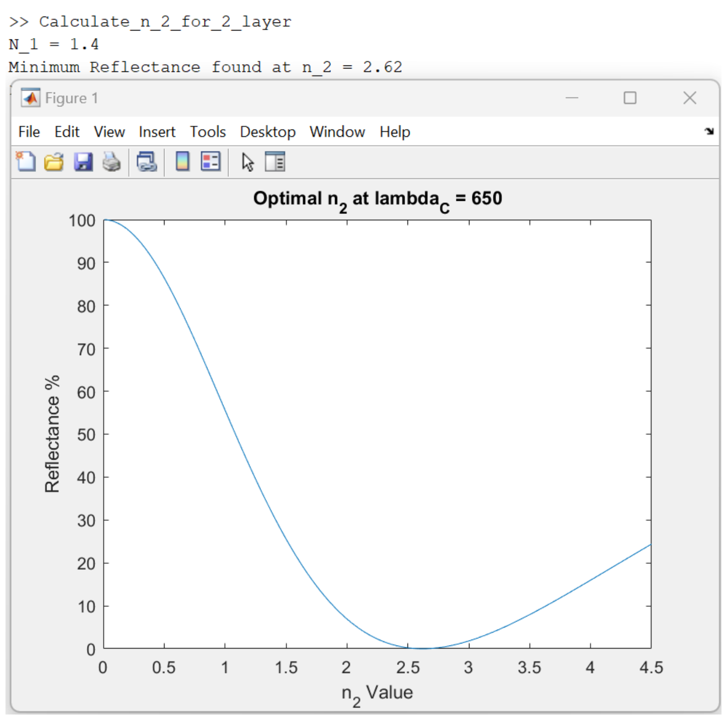

Figure 1.3 - Verification of Optimal n_2 at Central Wavelength

Part II: The Double-Layer Coating (Cont.)

A. Graphing Total Reflectivity Spectrum Over Wavelength Range 400-1400 nm Using TMM-Derived Refractive Index.

Figure 2.0 below displays the reflectivity versus the wavelength over the spectrum of 400 nm to 1400 nm. We used the n_2 value previously calculated, $n_2 = 2.62$. We can notice that at the center wavelength, we have minimum reflectivity at the center wavelength of 650 nm, which further validates our analytical calculation done previously. These results were produced with the "Part_2_a_and_b_400_1400.m" and "Part_2_a_and_b_200_2200.m" codes. For interest, we added the graph of reflectivity versus wavelength for 200 nm to 2200 nm as well.

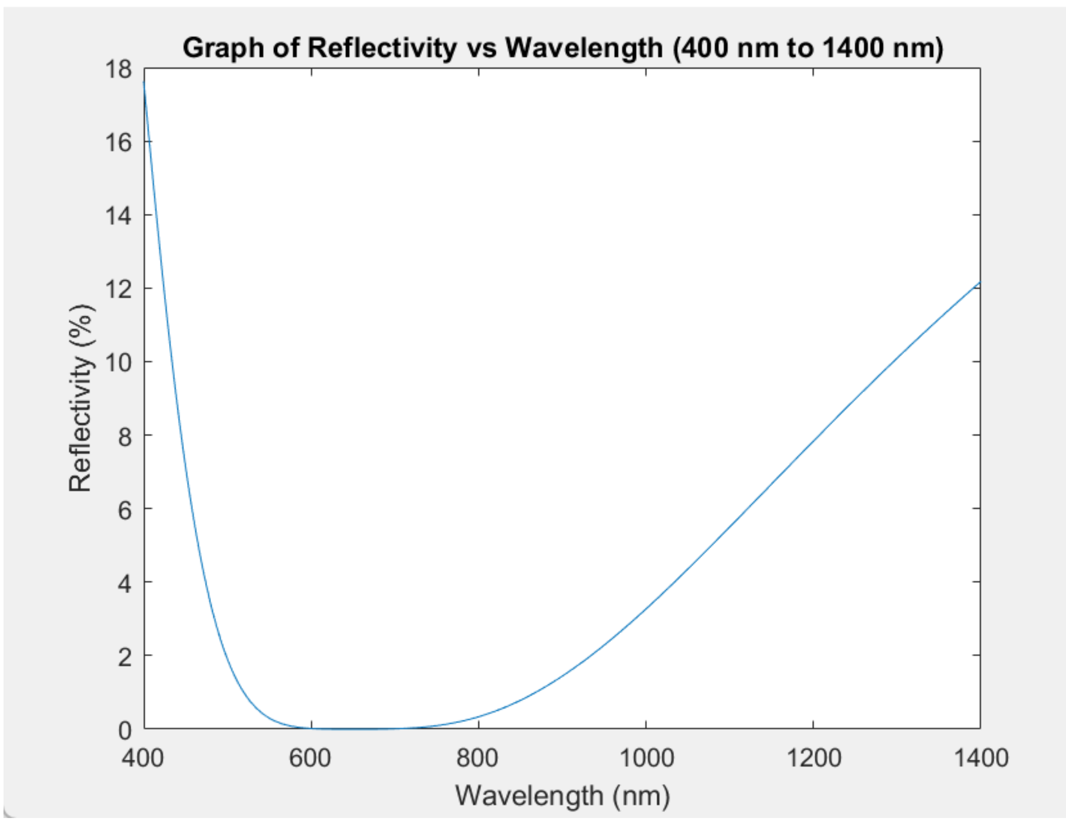

Figure 2.0 - Graph of Reflectivity versus Wavelength (400 nm to 1400 nm)

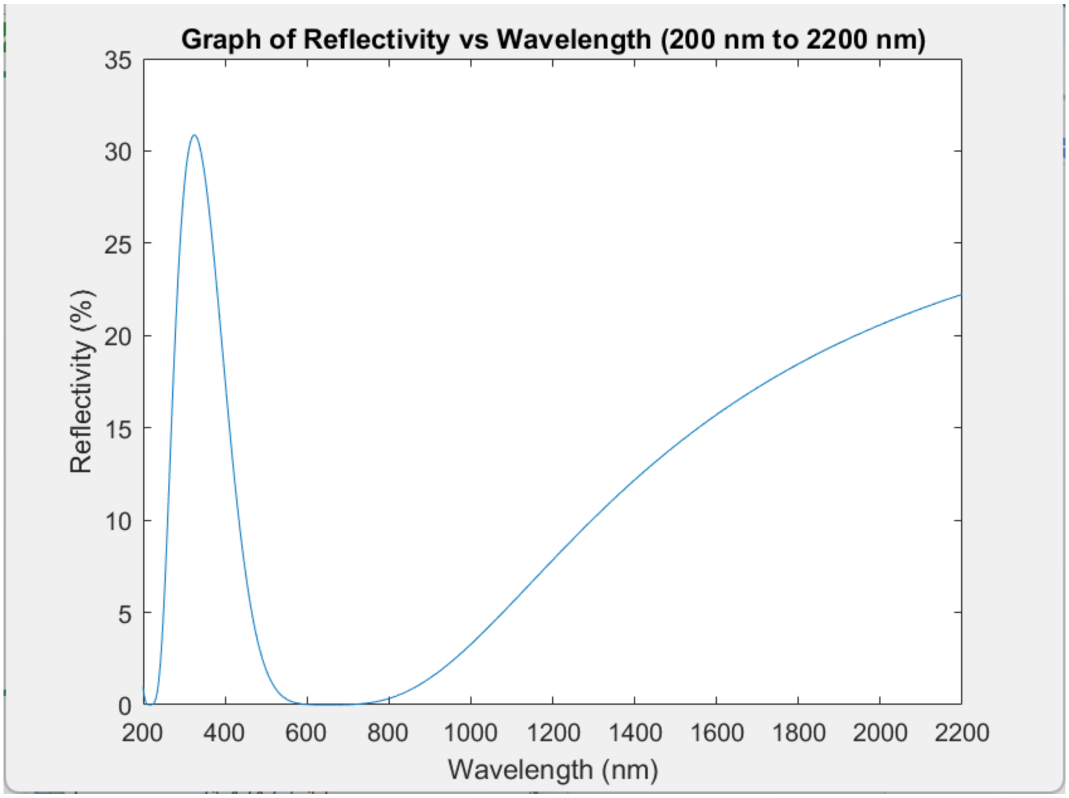

Figure 2.1 - Graph of Reflectivity vs Wavelength (200 nm to 2200 nm)

B. Total Power Transmitted Into Solar Cell For λ_1 = 200 nm to λ_2 = 2200 nm, And From λ_1 = 400 nm to λ_2 = 1400 nm.

Next, we calculated the total power for λ_1 = 200 nm to λ_2 = 2200 nm, and from λ_1 = 400 nm to λ_2 = 1400 nm using the following equation:

$$P = \int_{\lambda_1}^{\lambda_2} T(\lambda) I(\lambda) d\lambda$$

For λ_1 = 200 nm to λ_2 = 2200 nm, the power was calculated to be 929.5915 watts, as seen in Figure 2.2. These results were produced with the “Part_2_a_and_b_200_2200.m” and “Part_2_b.m” codes.

```
>> Part_2_a_and_b_200_2200
Total Power in Watts = 929.591599
```

Figure 2.2 - Total Power Calculation Output for 200 nm to 2200 nm

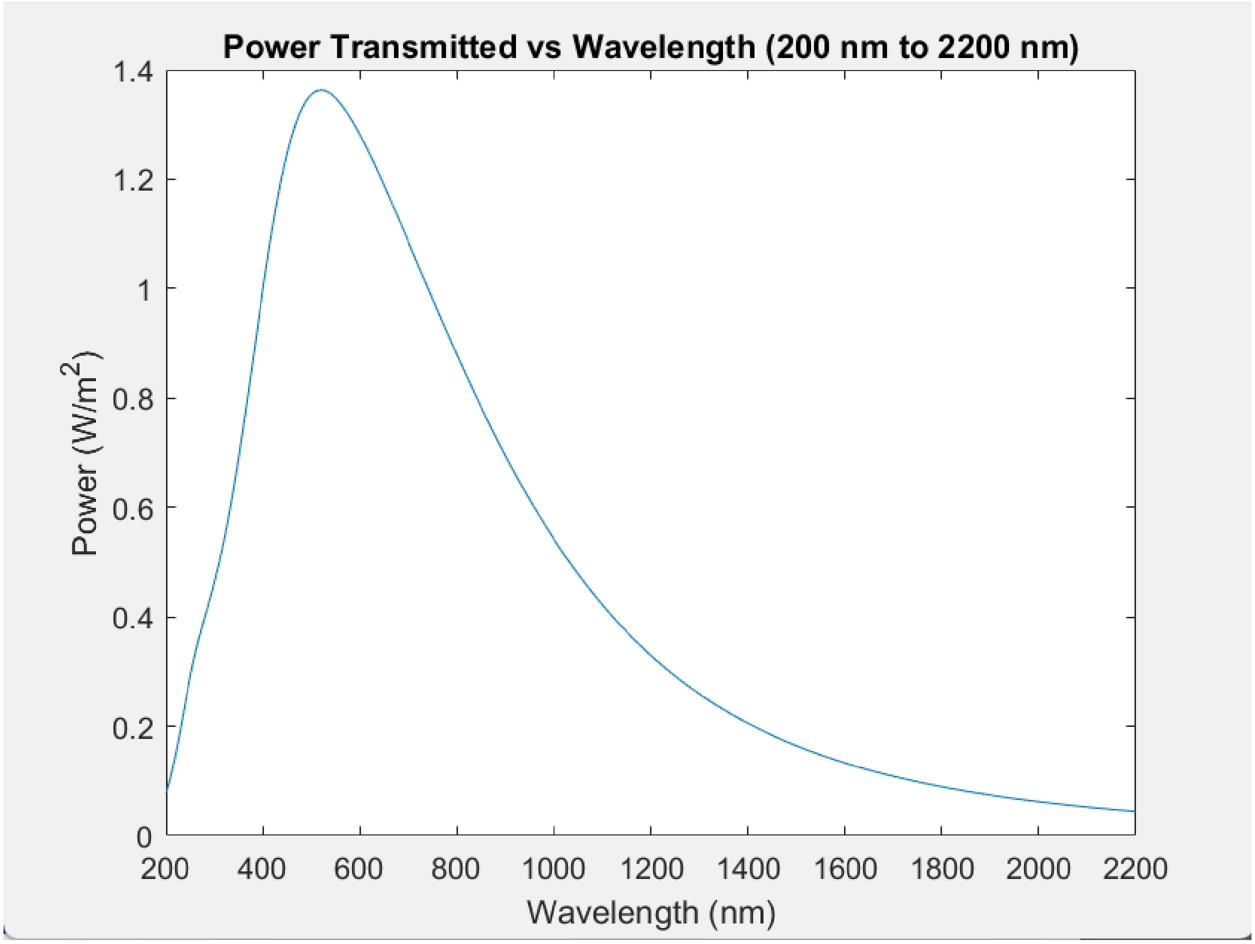

Figure 2.3 - Graph of Power Transmitted versus Wavelength (200 nm to 2200 nm)

For λ_1 = 400 nm to λ_2 = 1400 nm, the power was calculated to be 751.202 watts, as seen in Figure 2.3. These results were produced with the "Part_2_a_and_b_400_1400.m" and "Part_2_b.m" codes.

```
>> Part_2_a_and_b_400_1400
Total Power in Watts = 751.202068
```

Figure 2.4 - Total Power Calculation Output for 400 nm to 1400 nm

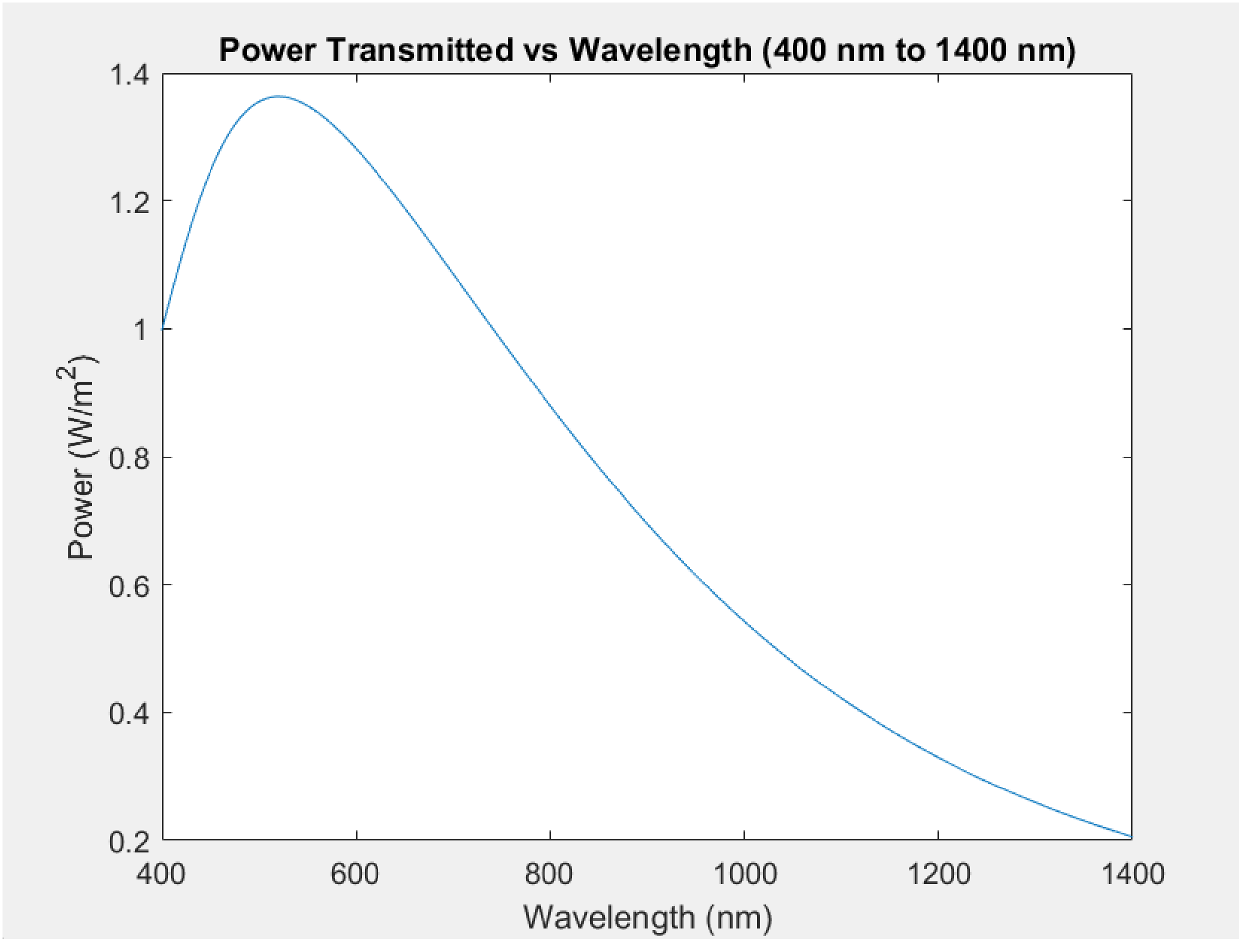

Figure 2.5 - Graph of Power Transmitted versus Wavelength (400 nm to 1400 nm)

C. Maximising Total Power Transmitted into Solar Cells for Both Wavelength Ranges by Optimizing n_2.

Yes, we can increase the total power by changing the values of the refractive indices or layer thicknesses. Here, we utilize an optimization algorithm to find the values of n_1 and n_2 that will yield the most power.

In the first case, we ran the optimization algorithm for λ_1 = 400 nm to λ_2 = 1400 nm, the power was calculated to be 758.888 watts for n_1 = 1.45 and n_2 = 2.4, as seen in Figure 2.4. These results were produced with the "Part_2_c_400_1400.m" code.

```
>> Part_2_c_400_1400
 Optimal n_1 = 1.45
 Optimal n_2 = 2.40
 Total Power in Watts = 758.8876
```

Figure 2.6 - Optimal n_1 and n_2 Values for Maximum Total Power (400 nm - 1400 nm)

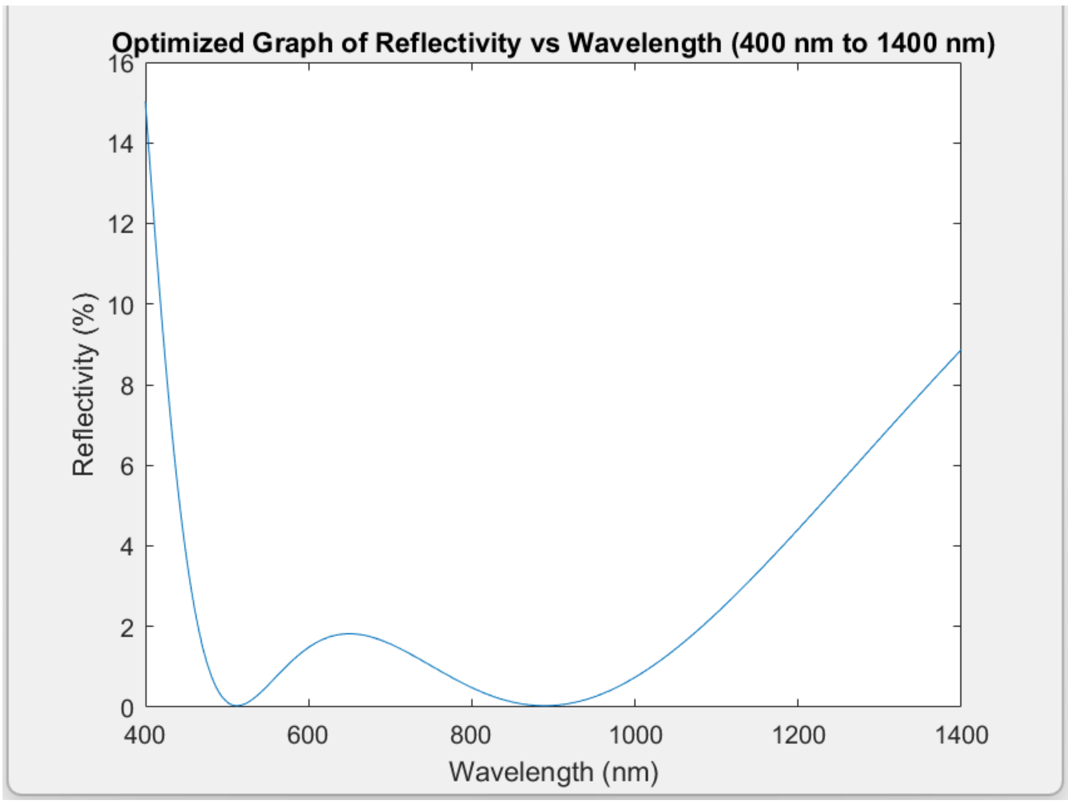

Figure 2.7 - Graph of Reflectivity vs Wavelength (400 nm to 1400 nm) Using Optimized n_1 and n_2 Values

In the second case, we ran the optimization algorithm for λ_1 = 200 nm to λ_2 = 2200 nm, where the total power was calculated to be 941.417 watts for n_1 = 1.45 and n_2 = 2.38, as seen in Figure 2.6. These results were produced with the “Part_2_c_200_2200.m” code.

```
>> Part_2_c_200_2200
 Optimal n_1 = 1.47
 Optimal n_2 = 2.38
 Total Power in Watts = 941.4174
```

Figure 2.8 - Optimal n_1 and n_2 Values for Maximum Total Power (200 nm - 2200 nm)

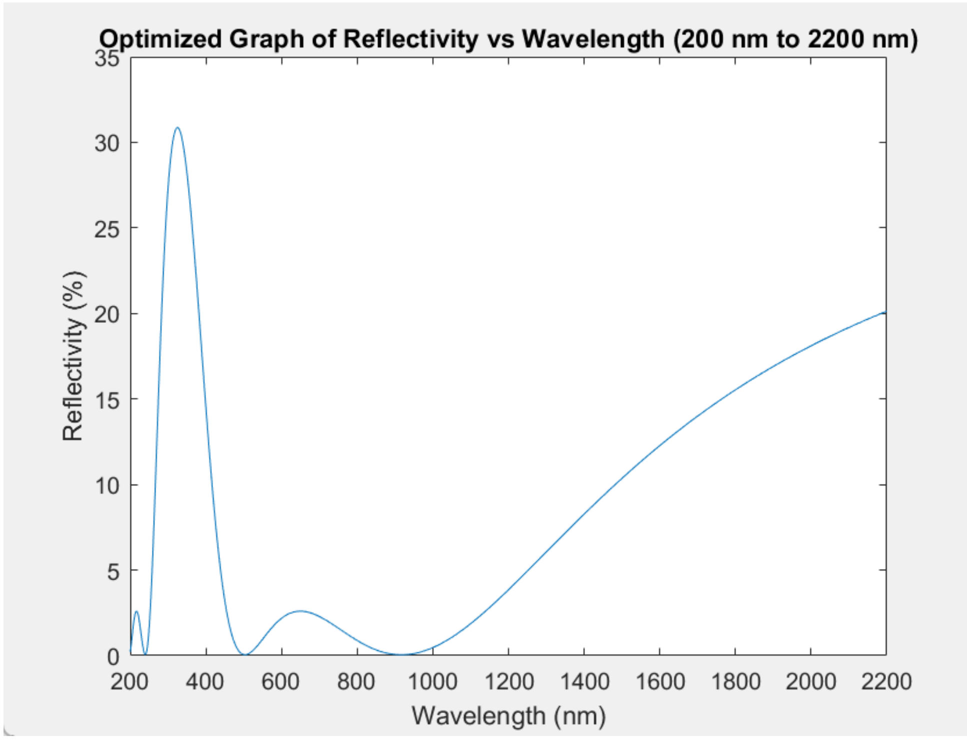

Figure 2.9 - Graph of Reflectivity vs Wavelength (200 nm to 2200 nm) Using Optimized n_1 and n_2 Values

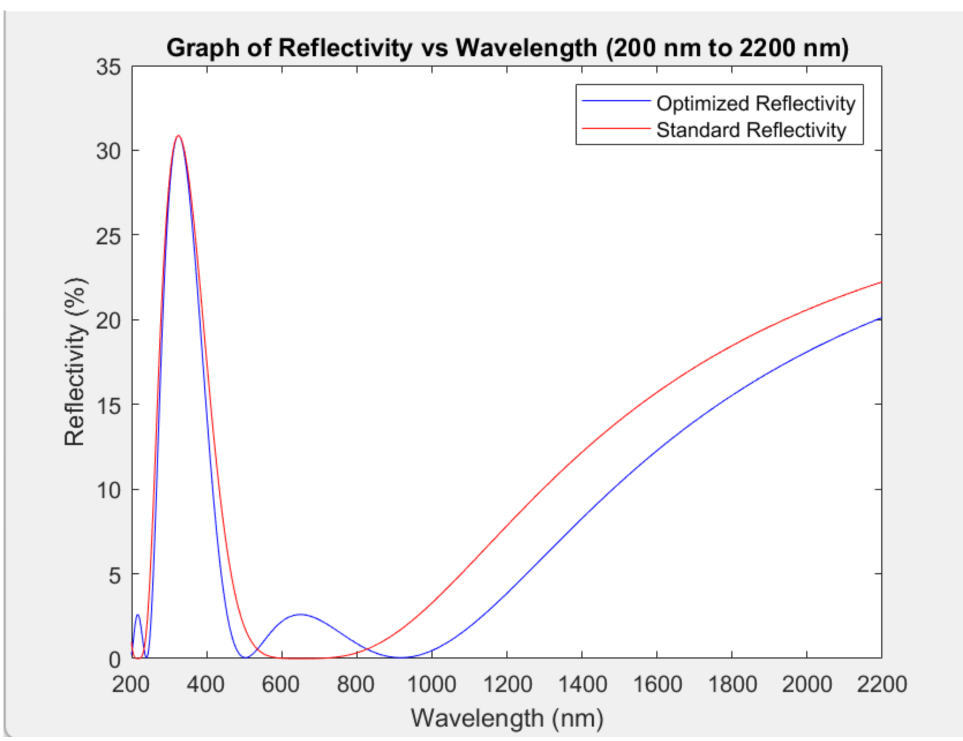

Figure 2.10 - Graph of Optimized and Standard Reflectivity vs Wavelength (200 nm to 2200 nm)

When comparing Figures 2.4 and 2.6, we do notice a slight increase in the total power, which proves that our optimization was successful. More interestingly though, when comparing Figures 2.7 to 2.9, in Figure 2.10, we notice that the reflectivity at central wavelength λ_C of 650 nm has significantly increased! This is caused by the fact that the emission spectra of the sun is very vast, hence, maximising power production at the central wavelength does not yield an accurate result. Therefore, in Figure 2.10, we observe a lower reflectivity across the entire spectrum for the optimized reflectivity graph versus the graph of standard reflectivity which translates to a higher power production. Figure 2.10 was created using the “overlap.m” code.

Part III: The Triple-Layer Coating

A. Analytical Calculation of Reflectivity at Central Wavelength.

In order to calculate the reflectivity at the central wavelength λ_C, we will utilize the transfer matrix method (TMM). Figure 3.0 describes the system that will be studied using the transfer matrix method:

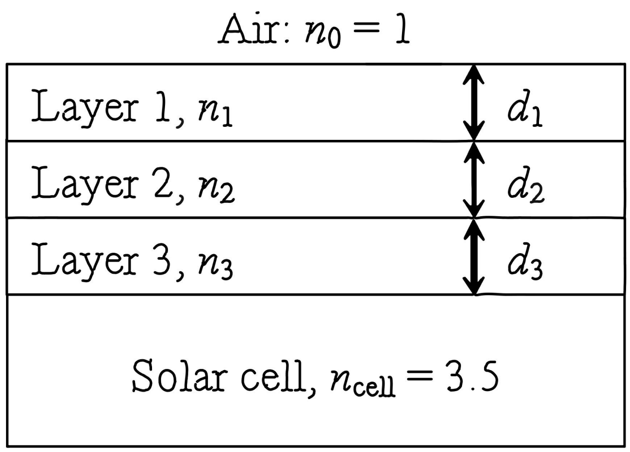

Figure 3.0 - Schematic of the Triple-Layer Anti-Reflective Coating System

While Figure 3.1 details light interactions within this system. Analysis of Figure 3.1 reveals eight reflection and transmission coefficients requiring calculation.

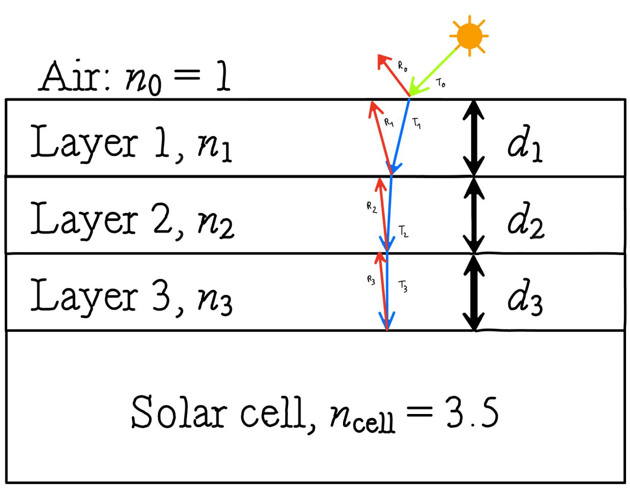

Figure 3.1 - Schematic of the Interaction of Light in Triple-Layer Anti-Reflective Coating System

The transfer matrix T is given by the multiplication of four dynamical matrices, Q, and three propagation matrices, P, given by:

$$T = Q_{0,1} P_1 Q_{1,2} P_2 Q_{2,3} P_3 Q_{3,4}$$

$$T = \frac{j^3}{\tau_{0,1}\tau_{1,2}\tau_{2,3}} \begin{bmatrix} 1 & \Gamma_{0,1} \\ \Gamma_{0,1} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & \Gamma_{1,2} \\ \Gamma_{1,2} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & \Gamma_{2,3} \\ \Gamma_{2,3} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & \Gamma_{3,4} \\ \Gamma_{3,4} & 1 \end{bmatrix}$$

$$T = \begin{bmatrix} 1 - \Gamma_{0,1}\Gamma_{1,2} & \Gamma_{0,1} - \Gamma_{1,2} \\ \Gamma_{0,1} - \Gamma_{1,2} & 1 - \Gamma_{0,1}\Gamma_{1,2} \end{bmatrix} \begin{bmatrix} 1 & \Gamma_{2,3} \\ \Gamma_{2,3} & 1 \end{bmatrix}$$

$$T = \begin{bmatrix} 1 - \Gamma_{0,1}\Gamma_{1,2} + (\Gamma_{0,1} - \Gamma_{1,2})\Gamma_{2,3} & (\Gamma_{0,1} - \Gamma_{1,2})\Gamma_{2,3} + \Gamma_{0,1} - \Gamma_{1,2} \\ \Gamma_{0,1} - \Gamma_{1,2} + (1 - \Gamma_{0,1}\Gamma_{1,2})\Gamma_{2,3} & (\Gamma_{0,1} - \Gamma_{1,2})\Gamma_{2,3} + 1 - \Gamma_{0,1}\Gamma_{1,2} \end{bmatrix}$$

$$T_{21} = (\Gamma_{0,1} - \Gamma_{1,2})(1 - \Gamma_{2,3}\Gamma_{3,4}) + (1 - \Gamma_{0,1}\Gamma_{1,2})(\Gamma_{2,3} - \Gamma_{3,4})$$

$$T_{11} = (\Gamma_{0,1}\Gamma_{1,2})(1 - \Gamma_{2,3}\Gamma_{3,4}) + (\Gamma_{0,1} - \Gamma_{1,2})(\Gamma_{2,3} - \Gamma_{3,4})$$

Where, $\Gamma = 0$ for minimized reflection:

$$\Gamma = \frac{T_{21}}{T_{11}} = 0$$

$$T_{21} = (\Gamma_{0,1} - \Gamma_{1,2})(1 - \Gamma_{2,3}\Gamma_{3,4}) + (1 - \Gamma_{0,1}\Gamma_{1,2})(\Gamma_{2,3} - \Gamma_{3,4}) = 0$$

Calculating our reflection coefficient for each layer:

$$\Gamma_{0,1} = \frac{n_0 - n_1}{n_0 + n_1}; \Gamma_{1,2} = \frac{n_1 - n_2}{n_1 + n_2}$$

$$\Gamma_{2,3} = \frac{n_2 - n_3}{n_2 + n_3}; \Gamma_{3,4} = \frac{n_3 - n_4}{n_3 + n_4}$$

Substituting into T_{21}:

$$= \Gamma_{0,1} - \Gamma_{0,1}\Gamma_{2,3}\Gamma_{3,4} - \Gamma_{1,2} + \Gamma_{1,2}\Gamma_{2,3}\Gamma_{3,4} + \Gamma_{2,3} - \Gamma_{3,4} - \Gamma_{0,1}\Gamma_{1,2}\Gamma_{2,3} + \Gamma_{0,1}\Gamma_{1,2}\Gamma_{3,4}$$

$$= \left(\frac{n_0 - n_1}{n_1 + n_0}\right) - \left(\frac{n_0 - n_1}{n_1 + n_0}\right)\left(\frac{n_2 - n_3}{n_3 + n_2}\right)\left(\frac{n_3 - n_4}{n_3 + n_4}\right) - \left(\frac{n_1 - n_2}{n_2 + n_1}\right) + \left(\frac{n_1 - n_2}{n_2 + n_1}\right)\left(\frac{n_2 - n_3}{n_3 + n_2}\right)\left(\frac{n_3 - n_4}{n_3 + n_4}\right) + \left(\frac{n_2 - n_3}{n_3 + n_2}\right)$$

$$- \left(\frac{n_3 - n_4}{n_3 + n_4}\right) - \left(\frac{n_0 - n_1}{n_1 + n_0}\right)\left(\frac{n_1 - n_2}{n_2 + n_1}\right)\left(\frac{n_3 - n_4}{n_3 + n_4}\right)$$

$$\left(n_0 - n_1\right)\left(n_1 + n_2\right)\left(n_2 + n_3\right)\left(n_3 + n_4\right) - \left(n_0 - n_1\right)\left(n_1 + n_2\right)\left(n_2 - n_3\right)\left(n_3 - n_4\right) -$$

$$\left(n_0 + n_1\right)\left(n_1 - n_2\right)\left(n_2 - n_3\right)\left(n_3 - n_4\right) + \left(n_0 + n_1\right)\left(n_1 + n_2\right)\left(n_2 - n_3\right)\left(n_3 + n_4\right) -$$

$$\left(n_0 - n_1\right)\left(n_1 - n_2\right)\left(n_2 - n_3\right)\left(n_3 + n_4\right) + \left(n_0 - n_1\right)\left(n_1 - n_2\right)\left(n_2 + n_3\right)\left(n_3 - n_4\right) = 0$$

The above can be simplified to:

$$4\left(n_0 n_2 - n_1^2\right)\left(n_2 n_4 - n_3^2\right) + 4\left(n_0 n_2 + n_1^2\right)\left(n_2 n_4 - n_3^2\right)$$

$$n_0 n_4 n_2^2 + n_0 n_2 n_3^2 - n_2 n_4 n_1^2 - n_1^2 n_3^2 + n_0 n_4 n_2^2 - n_0 n_2 n_3^2 + n_2 n_4 n_1^2 - n_1^2 n_3^2 = 0$$

$$2 n_0 n_4 n_2^2 - 2 n_1^2 n_3^2 = 0$$

To give an equation to find the values of n_2 for which the reflectivity is zero: $n_2 = \frac{n_1 n_3}{\sqrt{n_0 n_4}}$

B. Determining Refractive Index Relationship to Minimize Central Wavelength Reflectivity.

Using the above derivation, we can calculate the value of n_2 for minimum reflectivity for given values of n_1 and n_3.

$$n_0 = 1;\ n_1 = 1.4$$

$$n_3 = 3.15;\ n_4 = 3.5$$

$$n_2 = \frac{n_1 n_3}{\sqrt{n_0 n_4}} = \frac{1.4*3.15}{\sqrt{1*3.5}} = 2.3572... \approx 2.36$$

For a triple-layer coating, the optimal value of n_2 that achieves minimum reflection (Γ=0) at the central wavelength λ_C is approximately 2.36. This is confirmed in the following section.

C. Optimizing n_2 Value to Minimize Reflectivity with n_1 = 1.4 and n_3 = 3.15.

For this section, we will verify our previous results using MATLAB. In Figure 3.3, we graph the reflectivity of the system versus varying n_2 values at the central wavelength λ_C. We can graphically observe a minima at 2.36, this is verified in Figure 3.2 which displays the optimal value of n_2 for the respective n_1 and n_3 values. These results were produced with the "calculate_n_2_for_3_layer.m" code.

```
>> Calculate_n_2_for_3_layer
n_1 = 1.4
Minimum Reflectivity found at n_2 = 2.36
n_3 = 3.15
```

Figure 3.2 - Optimal n_2 Values for Given n_1 and n_3

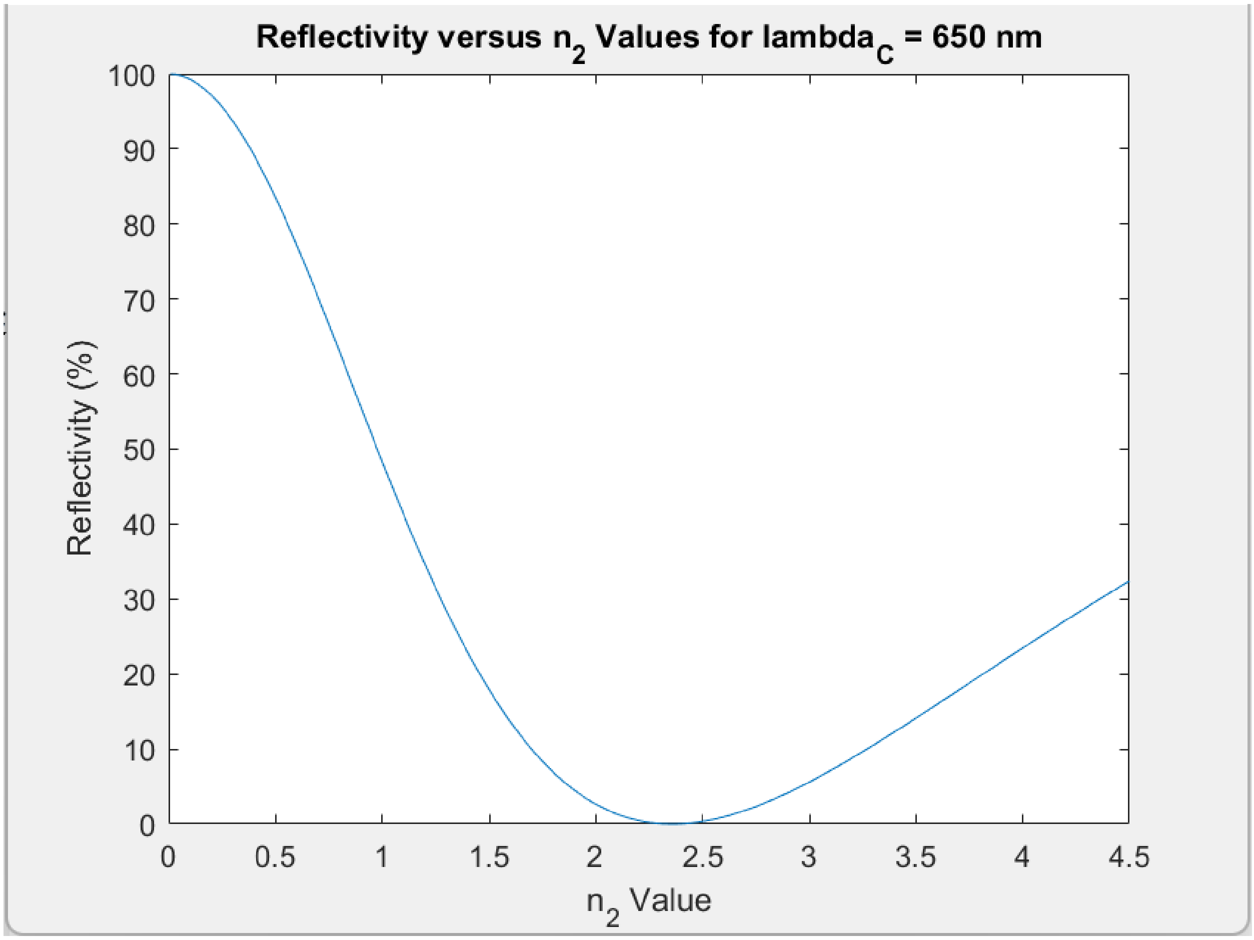

Figure 3.3 - Reflectivity versus n_2 Values for λ_C = 650 nm

Part IV: The Triple-Layer Coating (Cont.)

A. Calculating Power Transmitted at Different Refractive Indices.

In order to compute the power transmitted for different refractive indices of n_2, we will extrapolate from the previous algorithm to plot two curves, one for transmitted power for wavelengths of 200 nm to 2200 nm and a second for wavelengths from 400 nm to 1400 nm. These results were produced with the “part_4_a_b.m” code and are graphically represented in Figure 4.0.

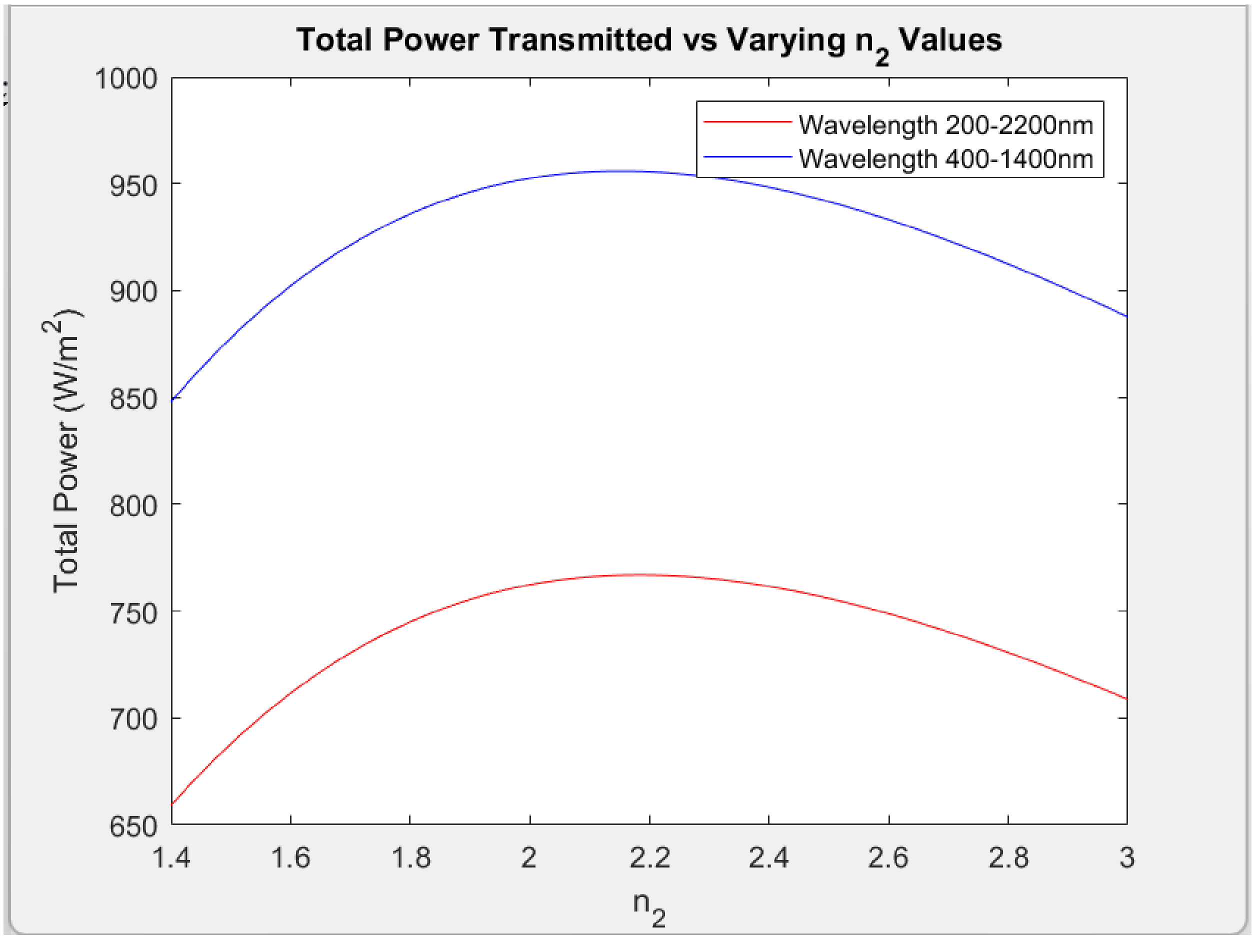

Figure 4.0 - Total Power Transmitted versus Varying n_2 Values

B. Value of n_2 for Maximum Power Transmitted.

Continuing from the previous task, we utilize the same algorithm and obtain the values of n_2 for which we have a maximum of total power transmission. Utilising MATLAB, we found that the optimal values of n_2 for $n_1 = 1.4$ and $n_3 = 3.15$, are $n_2 = 2.18$ for the spectrum of 400 nm to 1400 nm and $n_2 = 2.15$ for the spectrum of 200 nm to 2200 nm. These results were produced as well with the “part_4_a_b.m” code and are analytically represented in Figure 4.1.

```
>> part_4_a_b
 Maximum power transmitted in Watts (Wavelength 400-1400nm) = 766.95W
 n_2 value = 2.18
 Maximum power transmitted in Watts (Wavelength 200-2200nm) = 955.95W
 n_2 value = 2.15
```

Figure 4.1 - Maximum Power Transmitted for Varying n_2 Values

C. Comparison with Analytic Approach.

The values found for n_2 for the maximum power transmitted are not consistent with our analytic approach. This is due to a few factors, the first, is that when performing the analytic approach, we optimized the anti-reflective layer for the central wavelength λ_C of 650 nm. While a significant amount of power is transmitted when optimizing at this wavelength, it does not account for the entire spectrum. Additionally, the MATLAB optimization program allows us to calculate significantly more parameters for the power calculation.

D. Optimization of n_1, n_2, and n_3 for Maximum Power Transmission.

In this final section, we try to optimise the values of n_1, n_2, and n_3 for maximum power transmission across the entire spectrum from 200 nm to 2200 nm and from 400 nm to 1400 nm. This code uses a classic "brute-force" algorithm to compute 10,000 data points over 1000 iterations in order to optimize the values of n_1, n_2, and n_3. There are two versions of this code, "best_power.m" and "best_power_parallel_processing.m".

When running "best_power.m" on my computer, due to the extensive computational load it required, it took over 15 minutes to calculate the data. However, MATLAB was only using one core of my CPU, hence, "best_power_parallel_processing.m" utilizes the parallel processing tool-box from MATLAB to allow my code to utilize the 24 cores available on my device, greatly increasing the computation speeds and allowing for further accuracy! However, due to the selective requirements of the parallel processing toolbox, both codes are attached to this submission. The results from both codes are the same, except the code utilising the parallel processing functionality is significantly more time efficient and more representative of a code that would be used in industry.

```
Iteration: 987
Iteration: 988
Iteration: 989
Iteration: 990
Iteration: 991
Iteration: 992
Iteration: 993
Iteration: 994
Iteration: 995
Iteration: 996
Iteration: 997
Iteration: 998
Iteration: 999
Iteration: 1000

Optimal n_1 = 1.2675
Optimal n_2 = 1.8775
Optimal n_3 = 2.7775
Total Power Production (400 nm to 1400nm) = 770.8036 Watts
```

Figure 4.2 - Optimized n_1, n_2, and n_3 Values for 400 nm to 1400 nm

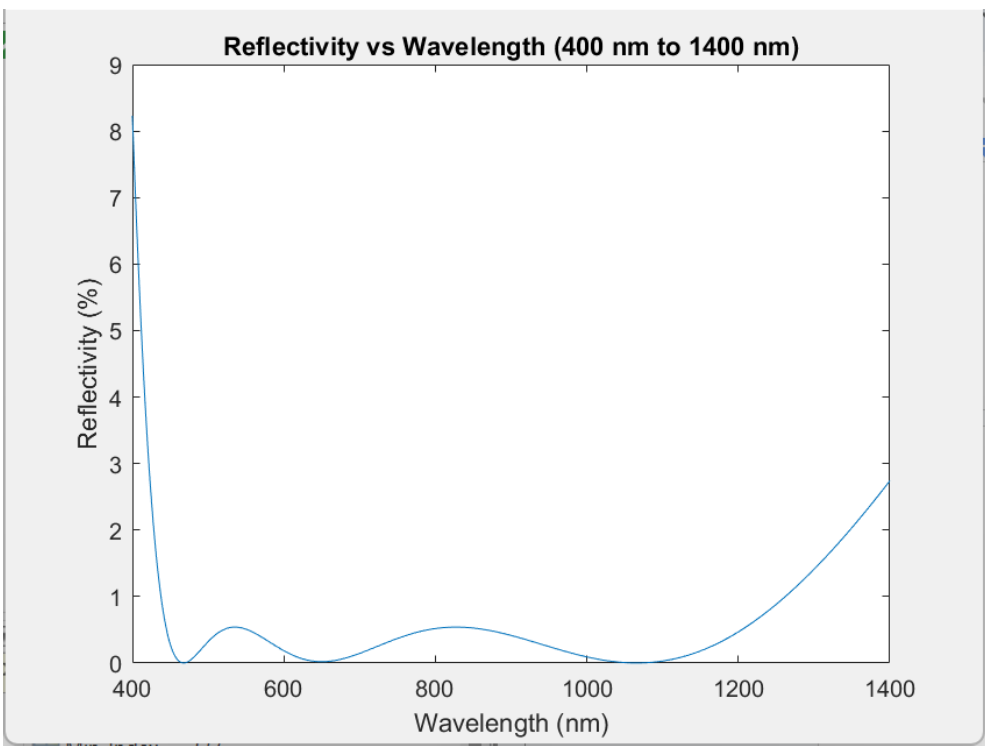

Figure 4.3 - Reflectivity versus Wavelength (400 nm to 1400 nm)

```
Iteration:  985
Iteration:  986
Iteration:  987
Iteration:  988
Iteration:  989
Iteration:  990
Iteration:  991
Iteration:  992
Iteration:  993
Iteration:  994
Iteration:  995
Iteration:  996
Iteration:  997
Iteration:  998
Iteration:  999
Iteration:  1000

Optimal n_1 = 1.3075
Optimal n_2 = 1.8700
Optimal n_3 = 2.6800
Total Power Production (200 nm to 2200nm) = 964.6009 Watts
```

Figure 4.4 - Optimized n_1, n_2, and n_3 Values for 200 nm to 2200 nm

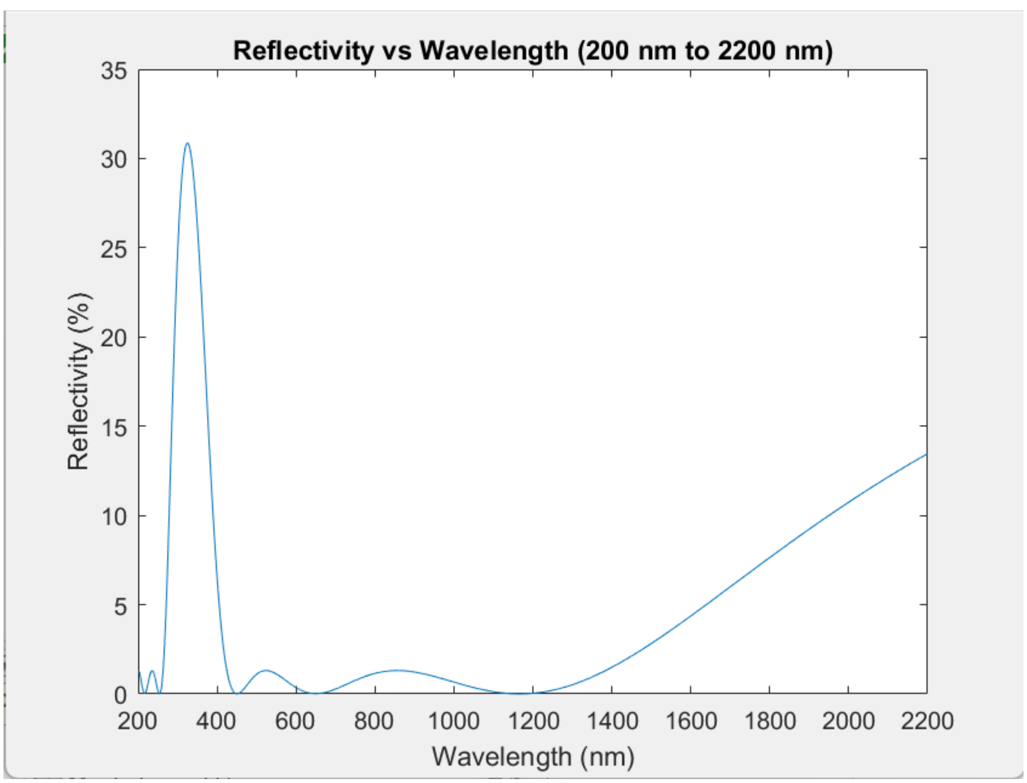

Figure 4.5 - Reflectivity versus Wavelength (200 nm to 2200 nm)

Conclusion

The comprehensive exploration into the design of anti-reflective coatings for solar cells, covering both double-layer and triple-layer configurations, has culminated in significant revelations about their efficacy in solar power optimization. Our study, utilised the Transfer Matrix Method (TMM) and MATLAB simulations, has illuminated the limitations of focusing solely on a central wavelength of 650 nm in the design of these coatings.

This project's application of TMM has been instrumental in understanding solar cell behavior under varying coating scenarios, ranging from the absence of an anti-reflective coating to more complex double-layer and triple-layer coatings. Our analyses began with a simplified model focusing on a single interface between air and the semiconductor, which confirmed that approximately 30% of incident light is reflected. This foundational understanding was crucial as we transitioned to more intricate analyses involving multi-layer coatings.

Further findings indicate that while optimizing anti-reflective coatings at the specific wavelength of 650 nm can indeed decrease reflectivity, it falls short in the overall enhancement of solar power production. The double-layer coatings, even when optimized for this wavelength, failed to maximize power across the broader solar spectrum, with a similar trend observed in triple-layer coatings. This pivotal insight illuminates the crucial need to extend the focus beyond a single wavelength and embrace a spectrum-wide approach when designing anti-reflective coatings.

The power transmission analysis further solidified this notion, showcasing a substantial uptick in efficiency with a broader spectrum optimization strategy. Such a methodology, as opposed to a narrow wavelength-centric approach, holds the key to unlocking the full potential of solar power production. Although our investigation was primarily theoretical and computational, grappling with idealized conditions and not delving into practical testing or material properties, its implications are extremely important.

This project accentuates the disparity between theoretical models and real-world applications with regards to the omission of environmental variables and inherent solar cell inefficiencies. It underscores the necessity of a balanced approach between theoretical predictions with empirical realities.

In essence, this project demonstrates that when desgining anti-reflective coatings for solar cells, one must encompass the entire solar spectrum, rather than fixating on a single wavelength, as it can significantly improve solar power production.

With that, future studies and simulations would greatly benefit from more accurate results by utilising nonidealised solar spectral irradiance, solar cell efficiency calculations, absorption parameters from the anti-reflective coating, atmospheric effects, temperature variations and accounting for all transmission angles of light into the solar cell.

In conclusion, this project lays down a foundational framework for future explorations aimed at bridging the gap between theoretical optimization and practical applicability. The transition from basic scenarios to complex multi-layer coatings represents a mission towards optimizing the system for enhanced energy conversion, laying a foundational framework for future explorations. As solar energy continues to be a critical element in humanity's pursuit of sustainable and renewable energy solutions, the comprehensive analyses conducted in this study create a basis for future advancements in solar cell technology.

References

- H. Schriemer, "ELG 3106 2023 Project" University of Ottawa, Ottawa, Ontario, Canada, Accessed: Nov. 23, 2023
- H. Schriemer, "Schriemer - The fundamentals of technical report writing" University of Ottawa, Ottawa, Ontario, Canada, Accessed: Nov. 23, 2023
- H. Schriemer, "ELG3106 2023 Project - ARC notes" University of Ottawa, Ottawa, Ontario, Canada, Accessed: Nov. 25, 2023
- H. Schriemer, "ELG3106 2023 Project - Finite Slab notes" University of Ottawa, Ottawa, Ontario, Canada, Accessed: Nov. 25, 2023

Appendix

Flow Charts

Calculate_n_2_for_2_layer.m

Initialize Parameters:

- Set n_0 to 1, n_1 to 1.4, n_3 and n_cell to 3.5, and lambda_C to 650.
- Calculate r01, t01, and Q01 using n_0 and n_1.
- Calculate r3S, t3S, and Q3S using n_3 and n_cell.
- Set Delta to π/2 and compute matrix P.

Create n_2 Range and Initialize Reflectance Storage:

- Create a range n_2_range from 0 to 4.5 with steps of 0.01.
- Initialize an array Store_Reflectance to store reflectance values, sized to match n_2_range.

Loop Through n_2 Values:

- For each n_2 in n_2_range:
    - Compute r12, t12, Q12 using n_1 and n_2.
    - Compute r23, t23, Q23 using n_2 and n_3.
    - Calculate the transfer matrix T as Q01 * P * Q12 * P * Q23 * P * Q3S.
    - Extract the reflectance coefficient Gamma from matrix T.
    - Compute reflectance and store it in Store_Reflectance.

Find Minimum Reflectance and Corresponding n_2 Value:

- Find the index of the minimum value in Store_Reflectance (denoted as Min_Index).
- Determine the corresponding n_2 value from n_2_range using Min_Index (denoted as min_n_2).

Plot and Output Results:

- Plot n_2_range against Store_Reflectance multiplied by 100.
- Label the graph with the appropriate title, x-label, and y-label.
- Output the results, indicating n_1 and the minimum reflectance found at n_2 (min_n_2).

Part_2_a_and_b_400_1400.m

Initialize Constants and Variables:

- Set complex unit j and refractive indices n_0, n_1, n_2, n_3.
- Define central wavelength Lambda_C and range from Lambda_Start (400) to Lambda_End (1400).

Calculate Coefficients for Layers:

- Compute reflection and transmission coefficients (r01, t01, r12, t12, r2S, t2S) using the refractive indices.

Create Q-Matrices:

- Formulate Q matrices (Q01, Q12, Q2S) for interfaces using the calculated coefficients.

Set up Wavelength Range and Delta:

- Create an array Lambda_Range for the specified wavelength range.
- Calculate Deltas array for phase change using Lambda_C and Lambda_Range.

Initialize Arrays for Storing Results:

- Create zero arrays Reflectance and Power matching the size of Lambda_Range.

Loop Over Wavelengths:

- For each wavelength (Lambda) in Lambda_Range:
  - Construct P matrix (P_Matrix) for current wavelength.
  - Calculate total transfer matrix T by multiplying the Q matrices and P matrices.
  - Compute reflectance Gamma from the transfer matrix.
  - Store the square of the absolute value of Gamma in Reflectance.
  - Calculate transmitted power (Trans) and irradiance (IRRAD) for the wavelength.
  - Compute and store power using Trans and IRRAD.

Plot Results:

- Plot Lambda_Range against Reflectance scaled to percentages.
- Set graph title, labels, and x-axis limits.

Output Total Power:

- Calculate the total power by summing the Power array.
- Print the total power

Part_2_a_and_b_200_2200.m

Initialize Constants and Variables:

- Set complex unit j and refractive indices n_0, n_1, n_2, n_3.
- Define central wavelength Lambda_C and range from Lambda_Start (200) to Lambda_End (2200).

Calculate Coefficients for Layers:

- Compute reflection and transmission coefficients (r01, t01, r12, t12, r2S, t2S) using the refractive indices.

Create Q-Matrices:

- Formulate Q matrices (Q01, Q12, Q2S) for interfaces using the calculated coefficients.

Set up Wavelength Range and Delta:

- Create an array Lambda_Range for the specified wavelength range.
- Calculate Deltas array for phase change using Lambda_C and Lambda_Range.

Initialize Arrays for Storing Results:

- Create zero arrays Reflectance and Power matching the size of Lambda_Range.

Loop Over Wavelengths:

- For each wavelength (Lambda) in Lambda_Range:
    - Construct P matrix (P_Matrix) for current wavelength.
    - Calculate total transfer matrix T by multiplying the Q matrices and P matrices.
    - Compute reflectance Gamma from the transfer matrix.
    - Store the square of the absolute value of Gamma in Reflectance.
    - Calculate transmitted power (Trans) and irradiance (IRRAD) for the wavelength.
    - Compute and store power using Trans and IRRAD.

Plot Results:

- Plot Lambda_Range against Reflectance scaled to percentages.
- Set graph title, labels, and x-axis limits.

Output Total Power:

- Calculate the total power by summing the Power array.
- Print the total power

Part_2_b.m

Initialization:

- Set refractive indices n_0, n_1, n_2, n_3.
- Calculate interface coefficients g01, g12, g23 and transmission coefficients t01, t12, t23.
- Define central wavelength lambda_c and create a wavelength range from 200 nm to 2199 nm.

Calculate Q-Matrices:

- Compute Q matrices q01, q12, q23 for the interfaces between different media.

Initialize Power Array:

- Create an array Power initialized to zero, with the same length as the wavelength range.

Loop Over Wavelengths:

- Iterate over each wavelength value:
  - For each wavelength, calculate L (related to wavelength).
  - Compute phase change delta_m using lambda_c and L.
  - Construct the P matrix p1 for the phase change.
  - Calculate the total transfer matrix T as the product of Q matrices and P matrices.
  - Compute reflectance g from the transfer matrix.
  - Calculate and store power using the formula involving g, L, and known constants.

Create Sub-Wavelength Range:

- Define a sub-range of wavelengths from 400 nm to 1399 nm.

Plot Results:

- Plot wavelength against Power for the range 400 nm to 1400 nm (Figure 1).
- Plot wavelength against Power for the full range 200 nm to 2200 nm (Figure 2).
- Set appropriate graph titles, labels, and axis limits for each plot.

Output Total Power:

- Calculate and print the total power for the 400 nm to 1400 nm range.
- Calculate and print the total power for the entire 200 nm to 2200 nm range.

Part_2_c_400_1400.m

Initialization:

- Set complex unit j, refractive indices n_0 and n_3, central wavelength central, and define the wavelength range Lambda_Range from 400 to 1400 nm.
- Initialize constants IRRAD_Const and Exp_Const.
- Define start and end values for n_1 and n_2, and set Step_Size and Max_Iteration.

Prepare for Iterative Search:

- Calculate the number of steps (numN1 and numN2) based on the range and step size for n_1 and n_2.
- Initialize arrays to store n_1, n_2, and total power for each combination.

Iterative Optimization Loop:

- For each iteration up to Max_Iteration:
    - Initialize index idx.
    - Nested loops over n_1 and n_2 ranges with increments of Step_Size:
        - Initialize zero arrays Best_Reflec and Store_PWR for each lambda.
        - Calculate reflection and transmission coefficients at interfaces (r01, r12, r2S, t01, t12, t2S).
        - Construct Q matrices (Q01, Q12, Q2S).
        - Loop over each wavelength in Lambda_Range:

        - Compute Delta and matrix P.
        - Calculate the transfer matrix T.
        - Compute reflectance Gamma and transmittance Tau.
        - Calculate and store power and reflectance for each wavelength.
      - Sum the power over all wavelengths and store in Store_Total_Power.
      - Increment index idx.
    - Find the best power and corresponding indices for n_1 and n_2.
    - Update n_1 and n_2 ranges and Step_Size for the next iteration.

Plot Results:

- Plot reflectivity vs. wavelength for the range of 200 to 2200 nm.
- Set the graph title and axis labels.

Output Results:

- Print the optimal values of n_1 and n_2.
- Print the total power for the best combination of n_1 and n_2.

Part_2_c_200_2200.m

Initialization:

- Set complex unit j, refractive indices n_0 and n_3, central wavelength central, and define the wavelength range Lambda_Range from 200 to 2200 nm.
- Initialize constants IRRAD_Const and Exp_Const.
- Define start and end values for n_1 and n_2, and set Step_Size and Max_Iteration.

Prepare for Iterative Search:

- Calculate the number of steps (numN1 and numN2) based on the range and step size for n_1 and n_2.
- Initialize arrays to store n_1, n_2, and total power for each combination.

Iterative Optimization Loop:

- For each iteration up to Max_Iteration:

        - Initialize index idx.
        - Nested loops over n_1 and n_2 ranges with increments of Step_Size:
            - Initialize zero arrays Best_Reflec and Store_PWR for each lambda.
            - Calculate reflection and transmission coefficients at interfaces (r01, r12, r2S, t01, t12, t2S).
            - Construct Q matrices (Q01, Q12, Q2S).
            - Loop over each wavelength in Lambda_Range:
                - Compute Delta and matrix P.
                - Calculate the transfer matrix T.
                - Compute reflectance Gamma and transmittance Tau.
                - Calculate and store power and reflectance for each wavelength.
            - Sum the power over all wavelengths and store in Store_Total_Power.
            - Increment index idx.
        - Find the best power and corresponding indices for n_1 and n_2.
        - Update n_1 and n_2 ranges and Step_Size for the next iteration.

Plot Results:

- Plot reflectivity vs. wavelength for the range of 200 to 2200 nm.
- Set the graph title and axis labels.

Output Results:

- Print the optimal values of n_1 and n_2.
- Print the total power for the best combination of n_1 and n_2.

Overlap.m

Initialization:

- Define initial parameters: refractive indices (n_0, n_3), central wavelength (central), irradiance constant (IRRAD_Const), and exponential constant (Exp_Const).
- Create Lambda_Range from 200 nm to 2200 nm.

Set Optimization Parameters:

- Initialize ranges for n_1 and n_2, step size (Step_Size), and maximum iterations (Max_Iteration).
- Calculate the number of steps (numN1, numN2) and initialize arrays to store n_1, n_2, and total power.

Optimization Loop for Best Reflectivity:

- For each iteration up to Max_Iteration:
    - Iterate through possible values of n_1 and n_2.
    - For each combination of n_1 and n_2:
        - Calculate reflection and transmission coefficients, and construct Q matrices (Q01, Q12, Q2S).
        - Loop through each wavelength in Lambda_Range:
            - Compute phase change Delta and matrix P.
            - Calculate transfer matrix T.
            - Compute reflectance Gamma, transmittance Tau, and power.
            - Store calculated power.
        - Sum and store total power for each combination.
    - Identify the combination with the best power, update the range of n_1 and n_2, and reduce Step_Size.

Plot Optimized Reflectivity:

- Plot Lambda_Range against the best reflectivity.
- Print the optimal values of n_1, n_2, and total power.

Standard Reflectivity Calculation:

- Set specific values for n_1 and n_2.
- Calculate reflection and transmission coefficients for fixed n_1 and n_2.
- Loop through Lambda_Range:
    - Calculate the transfer matrix T and reflectance Gamma.
    - Compute transmittance, irradiance, and power.
    - Store the computed reflectance and power.

Plot Standard Reflectivity:

- Add a plot for standard reflectivity to the existing graph.
- Set the graph title, labels, and legend.
- Print the total power for the standard reflectivity.

Finalize Plots:

- Display the combined plot showing both optimized and standard reflectivity.

Calculate_n_2_for_3_layer.m

Set Initial Parameters:

- Define refractive indices: n_0 = 1, n_1 = 1.4, n_3 = 3.15, n_cell = 3.5.
- Set central wavelength: lambda_C = 650.

Calculate Interface Coefficients:

- Compute reflection and transmission coefficients: r01, t01, r3S, t3S.
- Construct Q matrices for interfaces: Q01 and Q3S.

Prepare Phase Change Matrix:

- Set Delta to pi/2.
- Define matrix P for phase changes.

Initialize Reflectance Storage:

- Create n_2_range from 0 to 4.5 with a step size of 0.01.
- Initialize an array Store_Reflectance to store reflectance values.

Loop Over n_2 Values:

- For each value n_2 in n_2_range:
    - Compute reflection and transmission coefficients r12, t12, r23, t23.
    - Form Q matrices Q12 and Q23.
    - Calculate total transfer matrix T using Q01, P, Q12, Q23, P, and Q3S.
    - Determine reflectance Gamma from matrix T.
    - Store squared absolute value of Gamma in Store_Reflectance.

Identify Minimum Reflectance:

- Find the index of the minimum value in Store_Reflectance.
- Determine the corresponding n_2 value (min_n_2) from n_2_range.

Plot Reflectance vs n_2:

- Plot n_2_range against Store_Reflectance scaled to percentages.
- Set graph title, x-label, and y-label.

Output Results:

- Print n_1, the minimum reflectivity value, and corresponding n_2 (min_n_2), and n_3.

Part_4_a_b.m

Set Initial Parameters:

- Define refractive indices n_0, n_1, n_3, n_4, and set Center wavelength.
- Calculate reflection and transmission coefficients r01, r3S, t01, t3S.
- Create Q matrices Q01 and Q3S for interfaces.
- Define range for n_2 using linspace from 1.4 to 3 with num_N_2 steps.

Compute Total Power for Different Wavelength Ranges:

- Call computeTotalPower function for two wavelength ranges: 400-1400 nm and 200-2200 nm.
- Store the results in Store_Total_Power_1 and Store_Total_Power_2.

Plot Total Power vs n_2 Values:

- Plot Store_n_2 against Store_Total_Power_1 and Store_Total_Power_2.
- Set graph title, labels, and add a legend.

Find Maximum Power and Corresponding n_2 Value:

- For each power array, find the maximum power and its corresponding index.
- Print the maximum power and corresponding n_2 value for each wavelength range.

Function: computeTotalPower:

- Inputs: Lambda_Start, Lambda_End, Q01, Q3S, refractive indices, Center wavelength, numN2, Store_n_2.
- Create an array Lambda_Array from Lambda_Start to Lambda_End.
- Calculate Delta_Array and IRRAD_Array for each wavelength.

- Loop over numN2:
  - For each N2 in Store_n_2, calculate intermediate reflection/transmission coefficients and Q matrices (Q12, Q23).
  - Compute matrix P for phase changes.
  - Loop over Lambda_Array:
    - Construct P1 matrix for each wavelength.
    - Calculate the transfer matrix T for each wavelength.
    - Compute transmittance Tau and transmitted power Trans.
  - Calculate total power by summing Store_PWR over all wavelengths.
  - Store total power in Store_Total_Power.

Best_power.m

Initialization:

- Set initial refractive indices n_0, n_1, n_3, n_4 and central wavelength Lambda_C.
- Define the range of wavelengths from Lambda_Start to Lambda_End.
- Initialize step size and maximum iterations for the optimization process.
- Pre-calculate the maximum size for storage arrays based on the range and step size.

Create Storage Arrays:

- Initialize arrays Store_n_1, Store_n_2, Store_n_3, Store_Total_Power, and Best_Powers to store refractive indices, total power, and best powers found in each iteration.

Optimization Loop:

- Iterate Max_Iteration times to find the optimal refractive indices (n_1, n_2, n_3) that maximise total power:
  - In each iteration:
    - Loop through possible values of n_1, n_2, and n_3 in the defined range with increments of Step_Size.
    - For each combination of n_1, n_2, n_3:

- Initialize Store_PWR and Reflectance arrays.
- Loop over each wavelength in the range:
  - Calculate reflection and transmission coefficients at each interface.
  - Form Q matrices and phase change matrix P.
  - Compute the transfer matrix T.
  - Calculate reflectance Gamma and transmittance Tau.
  - Compute and store power for each wavelength in Store_PWR.
- Store the refractive indices and total power in the respective arrays.
- Update the index for the next combination.

Find Optimal Combination:

- After each iteration, find the combination of n_1, n_2, n_3 that resulted in the maximum total power.
- Update the start and end values of n_1, n_2, n_3, and reduce the step size for the next iteration.

Plot Results:

- Plot reflectivity vs. wavelength for the range 400 nm to 1400 nm.
- Set title, x-label, and y-label for the plot.

Output Results:

- Print the optimal values of n_1, n_2, n_3, and the maximum total power achieved.

Best_power_parallel_processing.m

This code uses the MATLAB Parallel Processing toolbox, if you'd like to compile it, ensure that you set the correct # of cores available for MATLAB and that you have the Parallel Processing toolbox installed.

Initialization:

- Set initial refractive indices n_0, n_4, and central wavelength Lambda_C.

- Define wavelength range from Lambda_Start to Lambda_End.
- Initialize step size Step_Size and maximum iterations Max_Iteration.
- Initialize ranges for n_1, n_2, n_3.
- Start parallel computing pool if not already running.

Iteration Loop:

- Iterate from 1 to Max_Iteration:
  - Print current iteration number.
  - Calculate the size of the refractive index arrays based on current ranges and step size.
  - Initialize temporary storage arrays for total power and refractive indices.

Parallelized Computation for Refractive Indices:

- For each combination of n_1, n_2, n_3 within their respective ranges:
  - Calculate refractive indices using ind2sub function.
  - Initialize an array Store_PWR to store power values for each wavelength.
  - Loop through wavelengths from Lambda_Start to Lambda_End:
    - Calculate reflection and transmission coefficients for each interface.
    - Form Q matrices and phase change matrix P.
    - Compute the transfer matrix T.
    - Calculate transmittance Tau and irradiance IRRAD.
    - Compute and store power in Store_PWR.
  - Store the calculated refractive indices and the sum of Store_PWR in temporary arrays.

Find Best Combination for Maximum Power:

- After each iteration, find the combination that resulted in the highest total power.
- Update the start and end values of n_1, n_2, n_3 based on the best combination found.
- Reduce the step size for the next iteration.

Plot Results:

- Plot reflectance vs. wavelength.
- Set graph title, x-label, and y-label.

Output Results:

- Print optimal values for n_1, n_2, n_3, and the maximum power achieved.

Figures

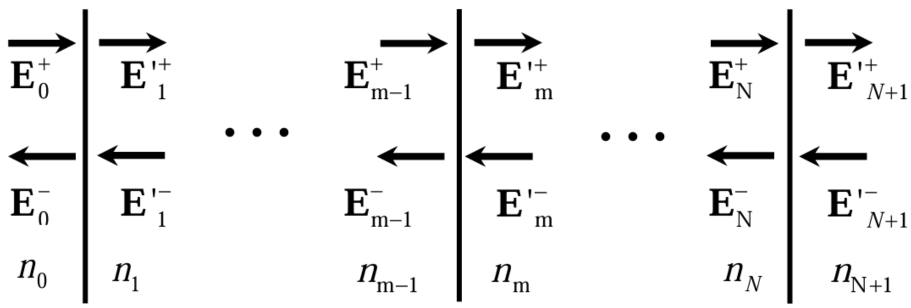

Figure 1.0 - The Transfer Matrix Method Illustrated by an Arbitrary Multilayer

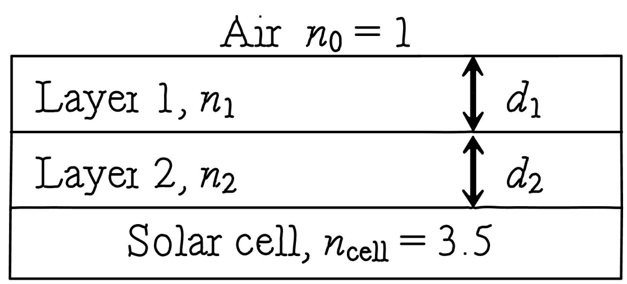

Figure 1.1 - Schematic of the Double-Layer Anti-Reflective Coating System

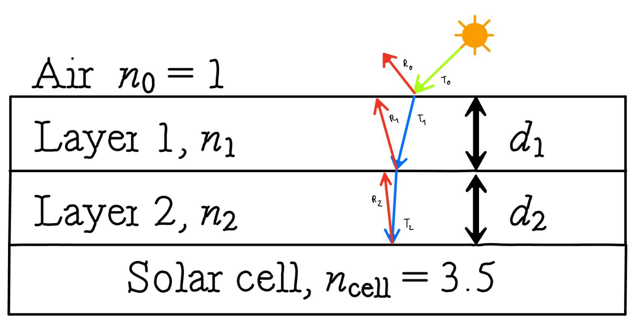

Figure 1.2 - Schematic of the Interaction of Light in Double-Layer Anti-Reflective Coating System

```
>> Calculate_n_2_for_2_layer
N_1 = 1.4
Minimum Reflectance found at n_2 = 2.62
```

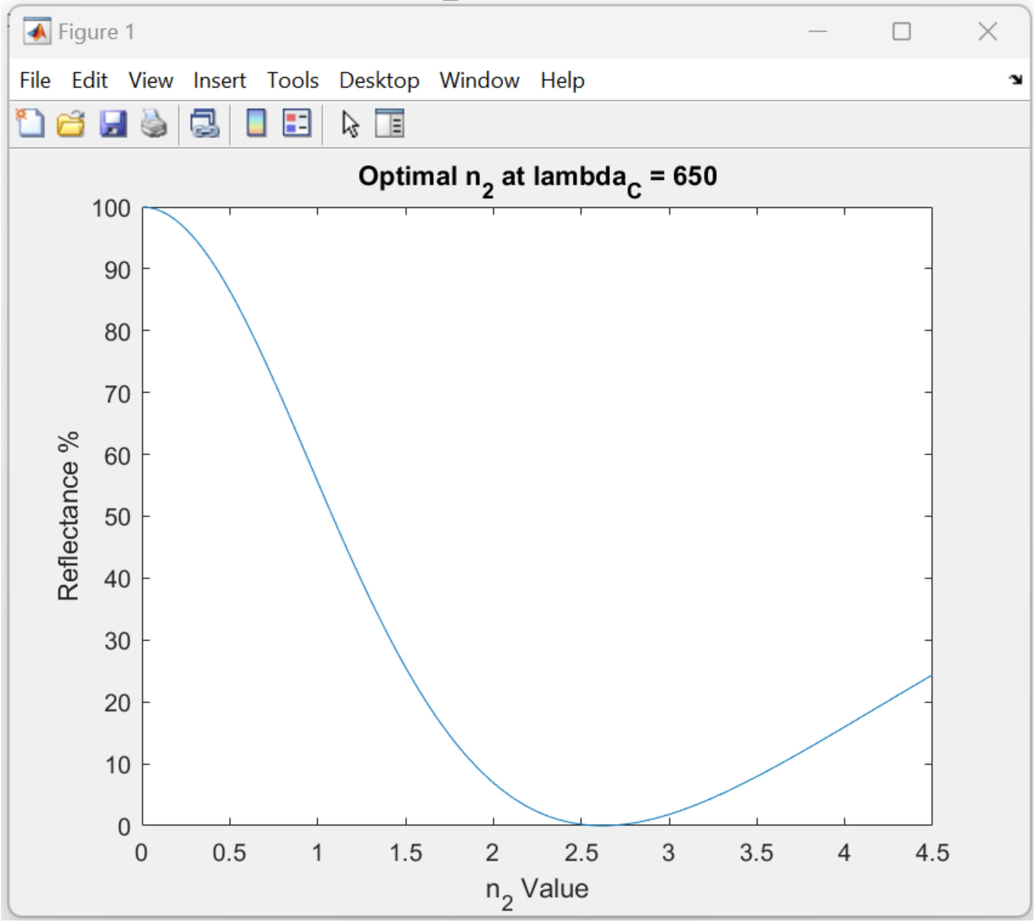

Figure 1.3 - Verification of Optimal n_2 at Central Wavelength

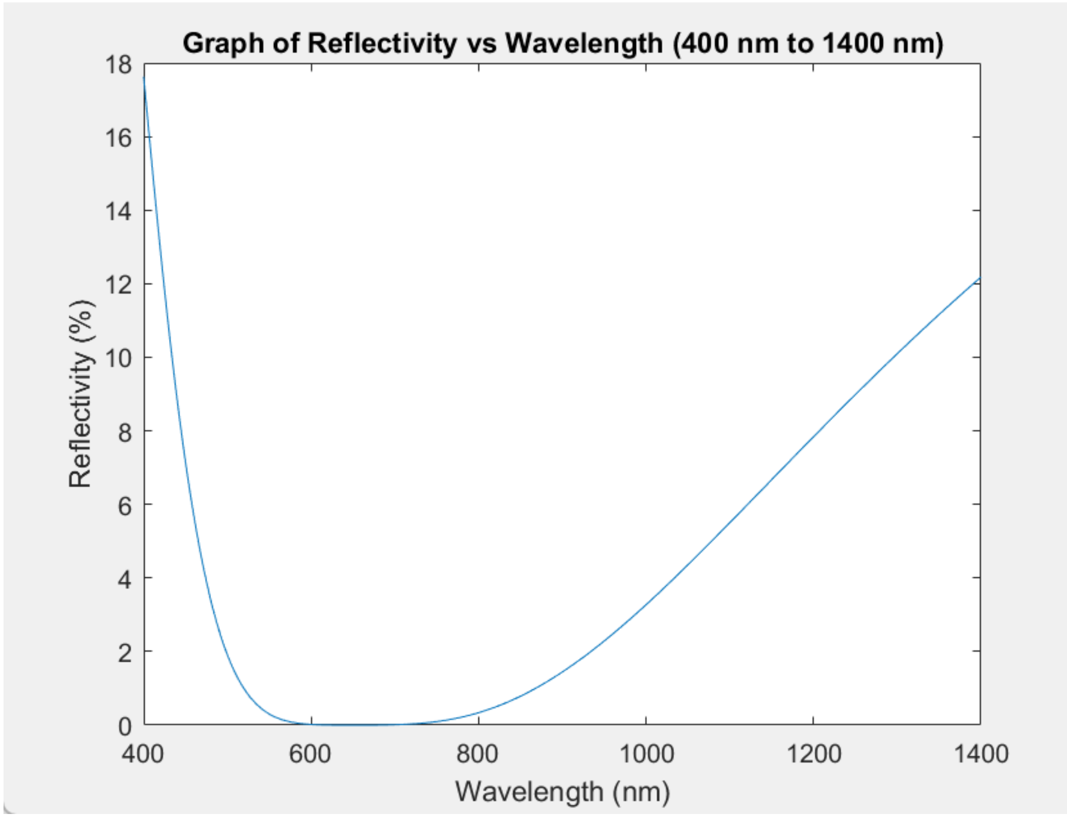

Figure 2.0 - Graph of Reflectivity versus Wavelength (400 nm to 1400 nm)

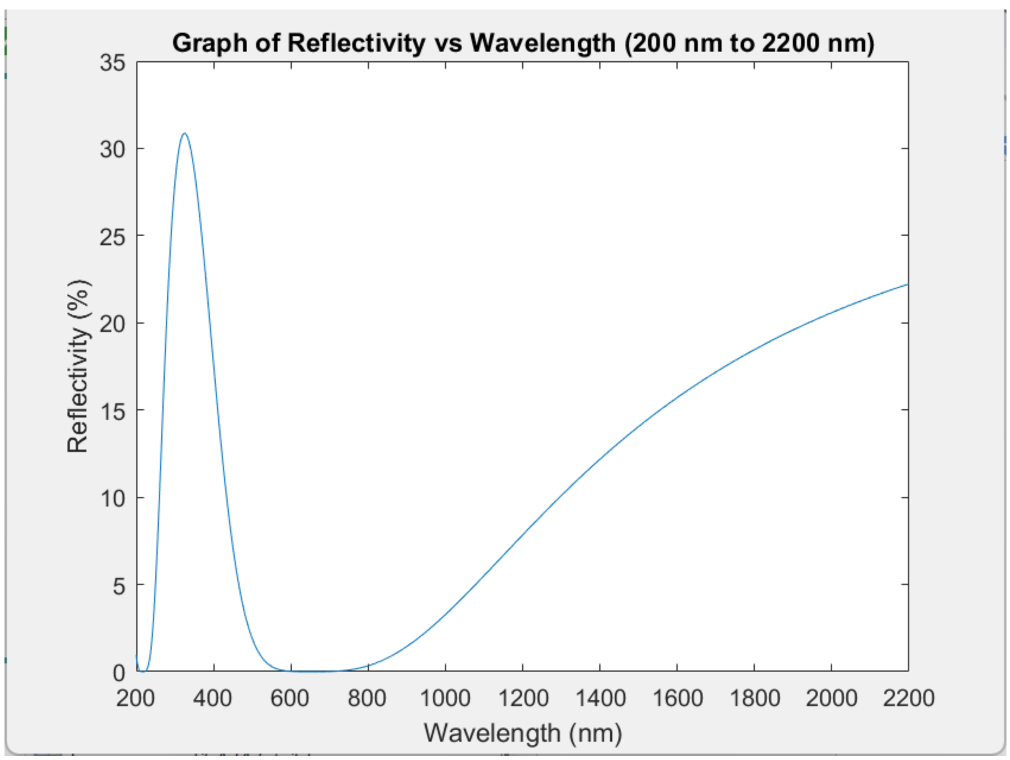

Figure 2.1 - Graph of Reflectivity vs Wavelength (200 nm to 2200 nm)

```
>> Part_2_a_and_b_200_2200
Total Power in Watts = 929.591599
```

Figure 2.2 - Total Power Calculation Output for 200 nm to 2200 nm

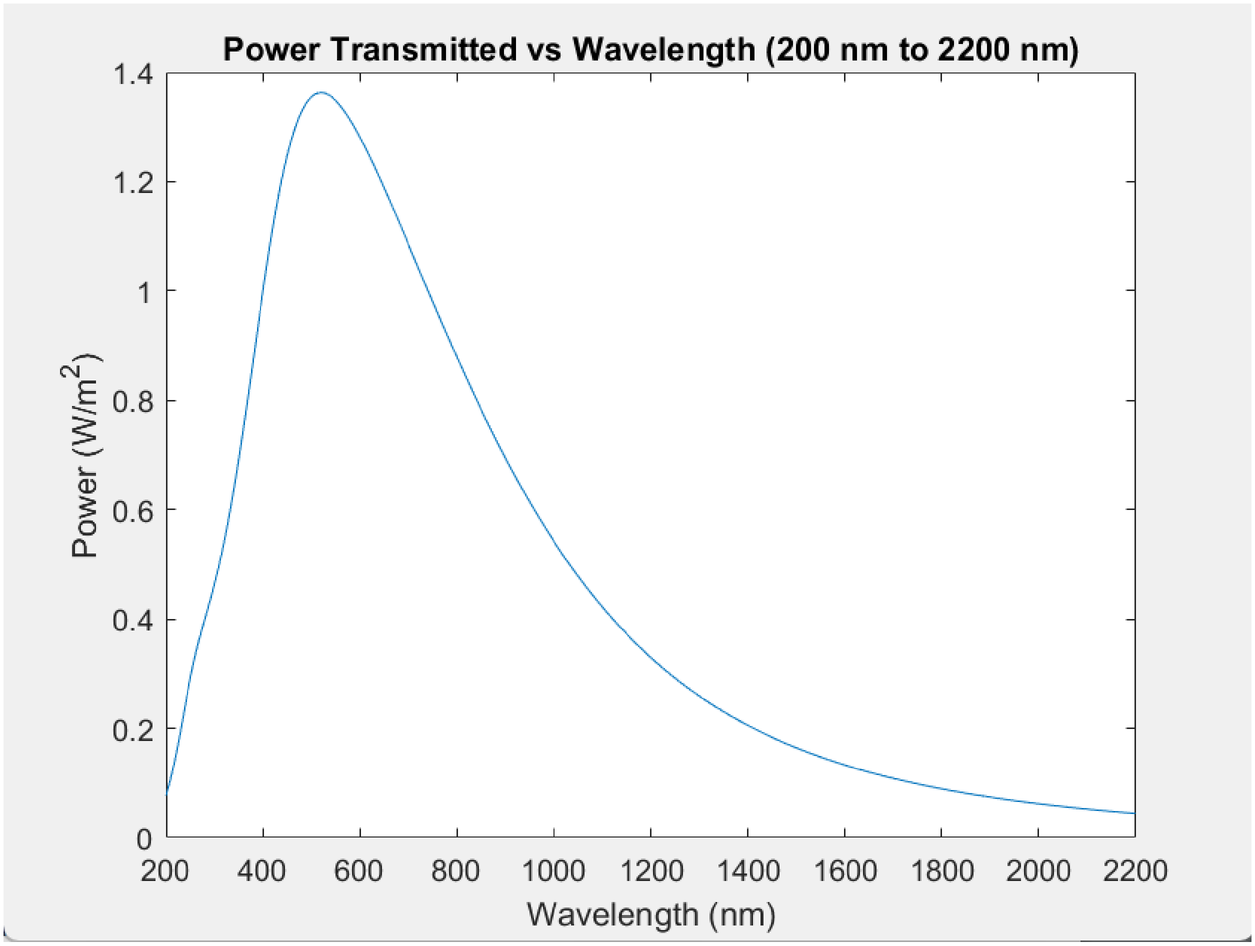

Figure 2.3 - Graph of Power Transmitted versus Wavelength (200 nm to 2200 nm)

```
>> Part_2_a_and_b_400_1400
Total Power in Watts = 751.202068
```

Figure 2.4 - Total Power Calculation Output for 400 nm to 1400 nm

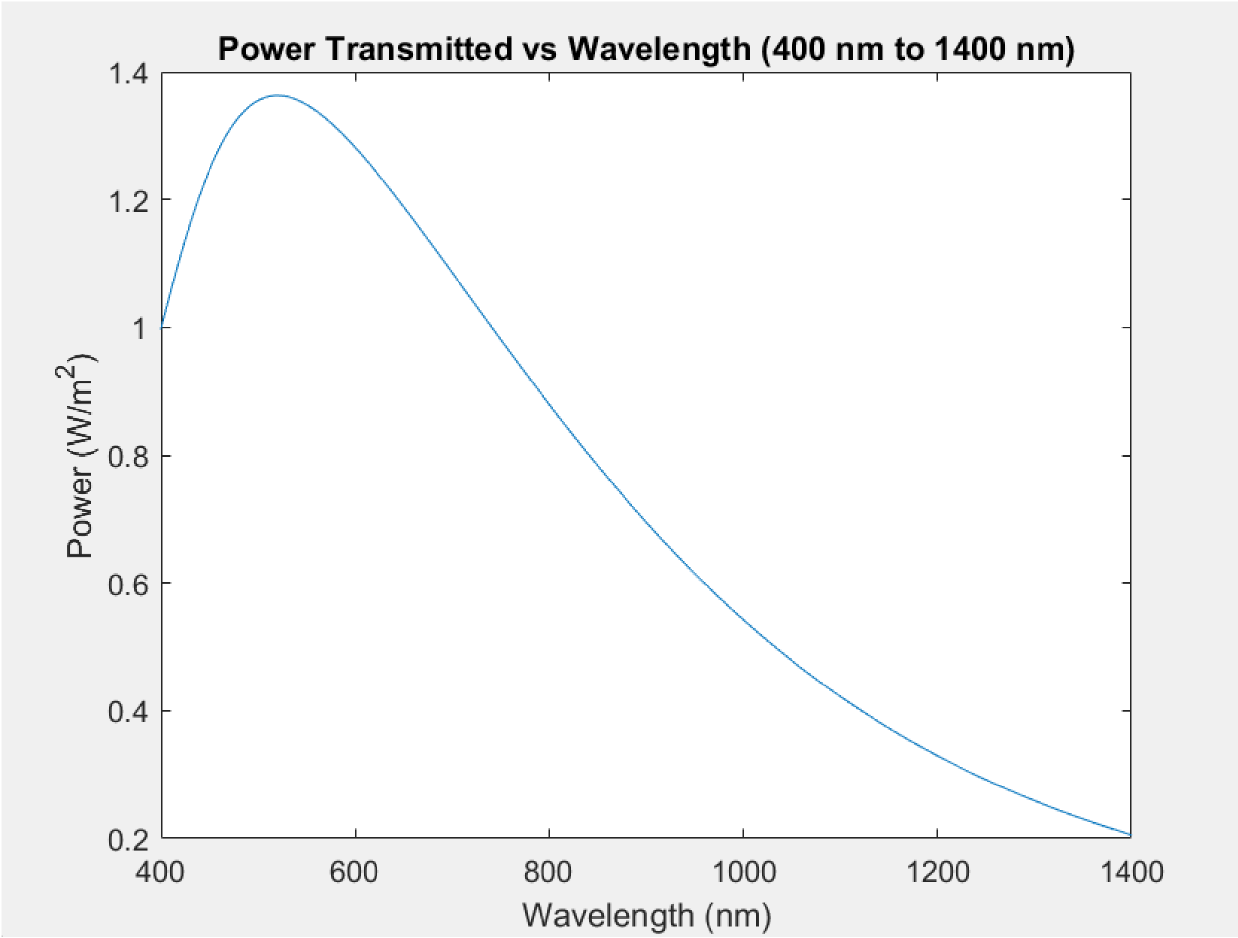

Figure 2.5 - Graph of Power Transmitted versus Wavelength (400 nm to 1400 nm)

```
>> Part_2_c_400_1400
 Optimal n_1 = 1.45
 Optimal n_2 = 2.40
 Total Power in Watts = 758.8876
```

Figure 2.6 - Optimal n_1 and n_2 Values for Maximum Total Power (400 nm - 1400 nm)

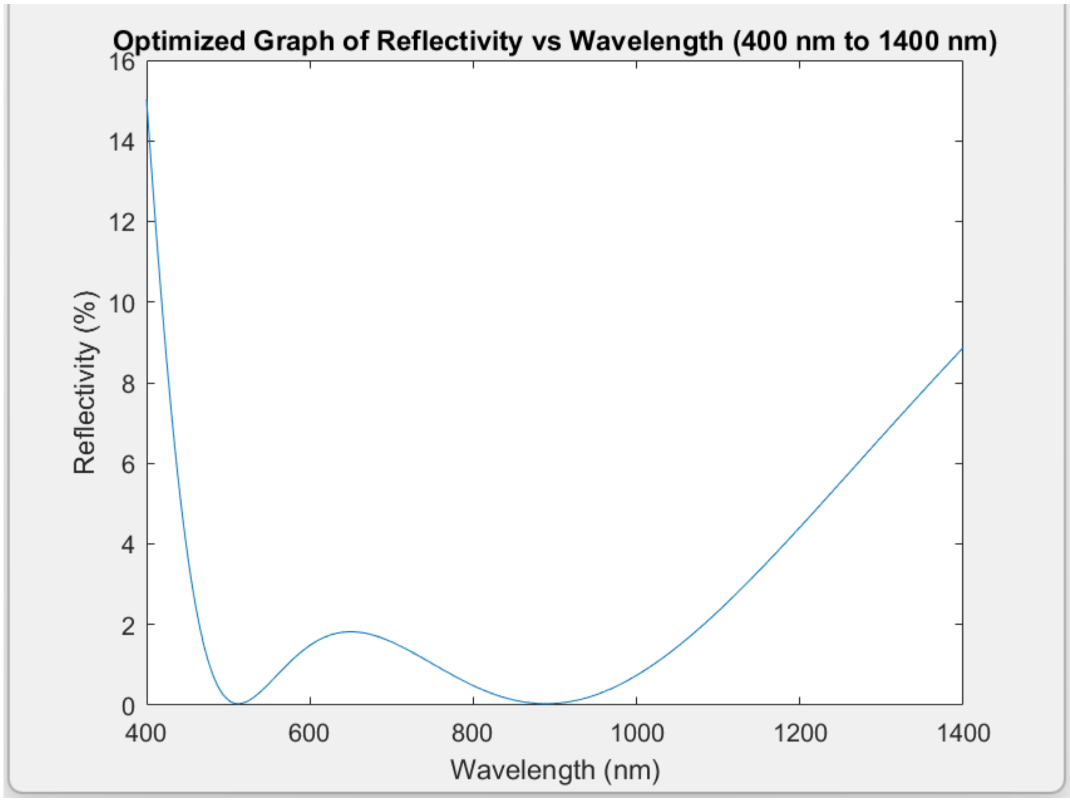

Figure 2.7 - Graph of Reflectivity vs Wavelength (400 nm to 1400 nm) Using Optimized n_1 and n_2 Values

```
>> Part_2_c_200_2200
 Optimal n_1 = 1.47
 Optimal n_2 = 2.38
 Total Power in Watts = 941.4174
```

Figure 2.8 - Optimal n_1 and n_2 Values for Maximum Total Power (200 nm - 2200 nm)

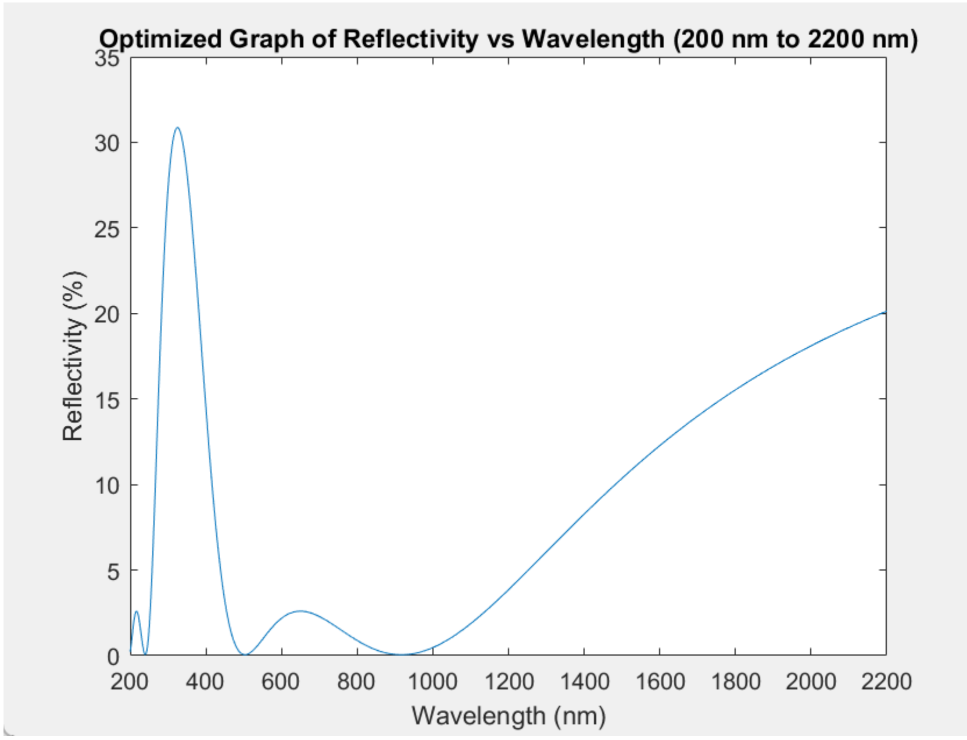

Figure 2.9 - Graph of Reflectivity vs Wavelength (200 nm to 2200 nm) Using Optimized n_1 and n_2 Values

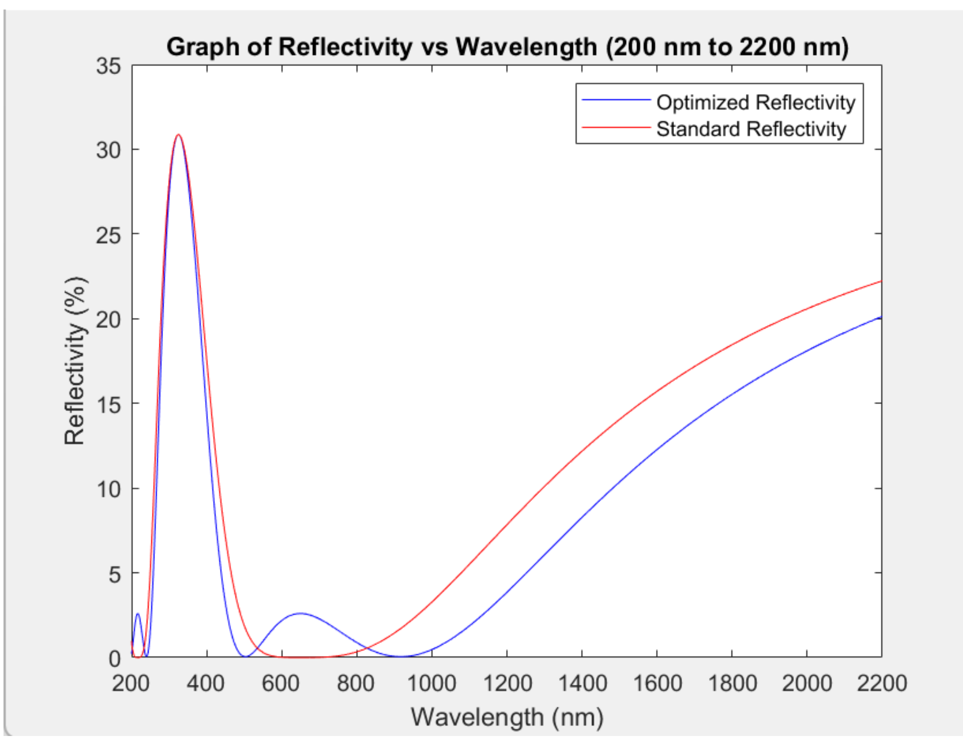

Figure 2.10 - Graph of Optimized and Standard Reflectivity vs Wavelength (200 nm to 2200 nm)

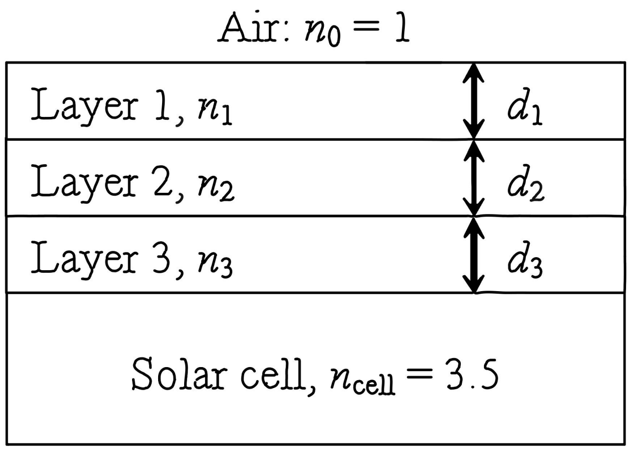

Figure 3.0 - Schematic of the Triple-Layer Anti-Reflective Coating System

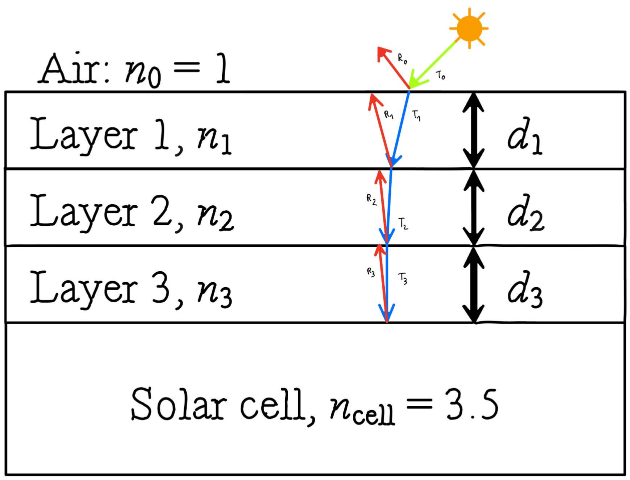

Figure 3.1 - Schematic of the Interaction of Light in Triple-Layer Anti-Reflective Coating System

```
>> Calculate_n_2_for_3_layer
n_1 = 1.4
Minimum Reflectivity found at n_2 = 2.36
n_3 = 3.15
```

Figure 3.2 - Optimal n_2 Values for Given n_1 and n_3

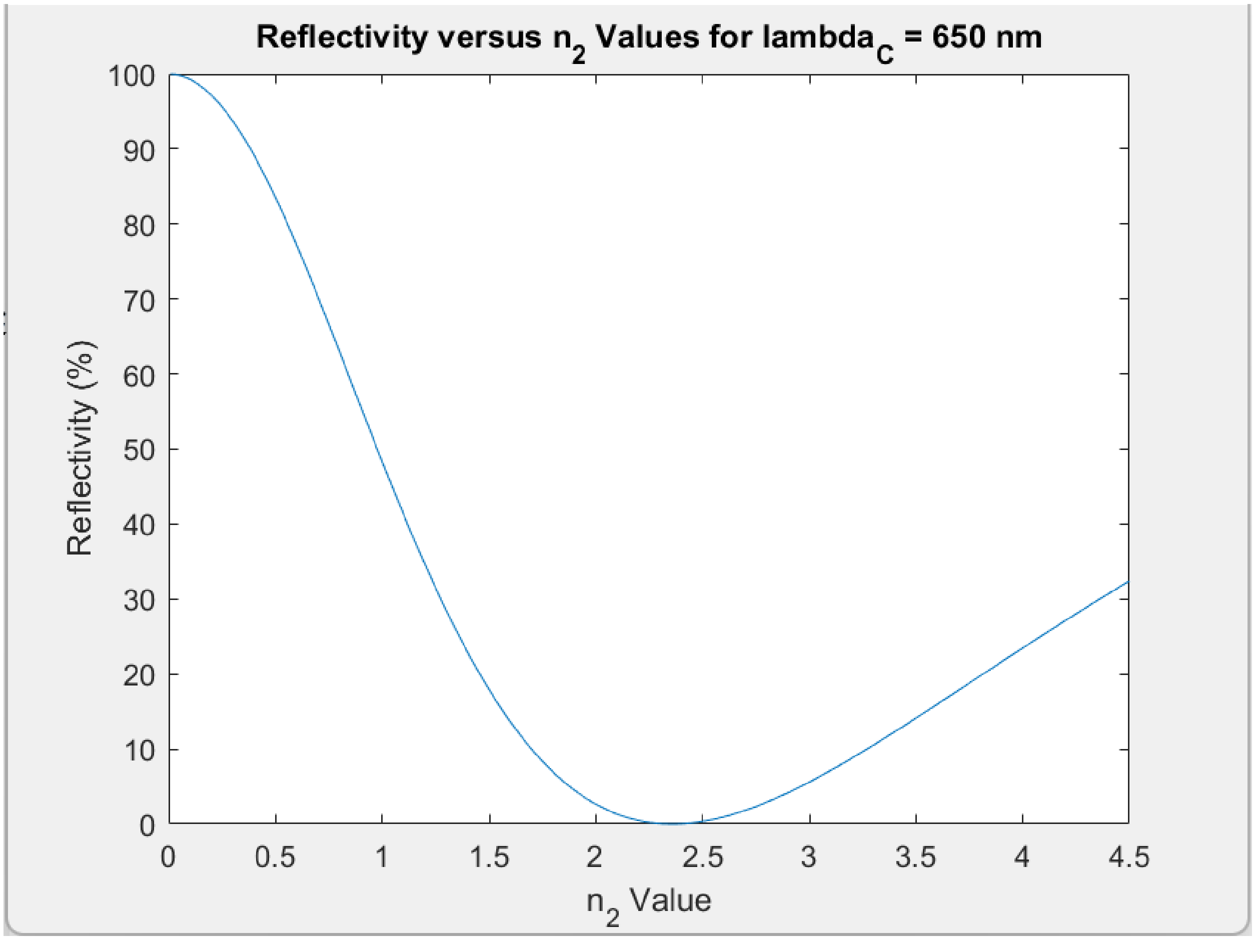

Figure 3.3 - Reflectivity versus n_2 Values for λ_C = 650 nm

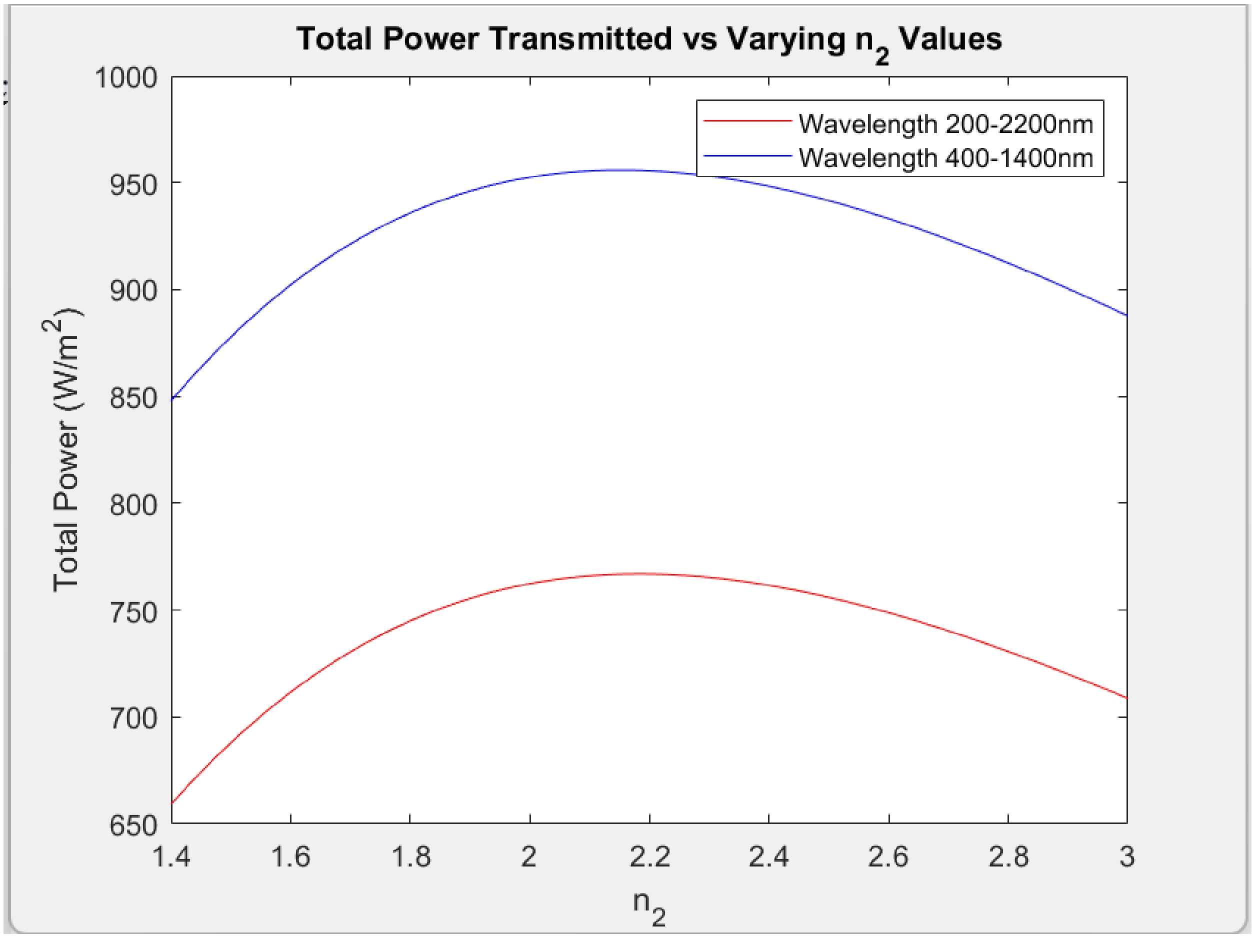

Figure 4.0 - Total Power Transmitted versus Varying n_2 Values

```
>> part_4_a_b
 Maximum power transmitted in Watts (Wavelength 400-1400nm) = 766.95W
 n_2 value = 2.18
 Maximum power transmitted in Watts (Wavelength 200-2200nm) = 955.95W
 n_2 value = 2.15
```

Figure 4.1 - Maximum Power Transmitted for Varying n_2 Values

```
Iteration: 987
Iteration: 988
Iteration: 989
Iteration: 990
Iteration: 991
Iteration: 992
Iteration: 993
Iteration: 994
Iteration: 995
Iteration: 996
Iteration: 997
Iteration: 998
Iteration: 999
Iteration: 1000

Optimal n_1 = 1.2675
Optimal n_2 = 1.8775
Optimal n_3 = 2.7775
Total Power Production (400 nm to 1400nm) = 770.8036 Watts
```

Figure 4.2 - Optimized n_1, n_2, and n_3 Values for 400 nm to 1400 nm

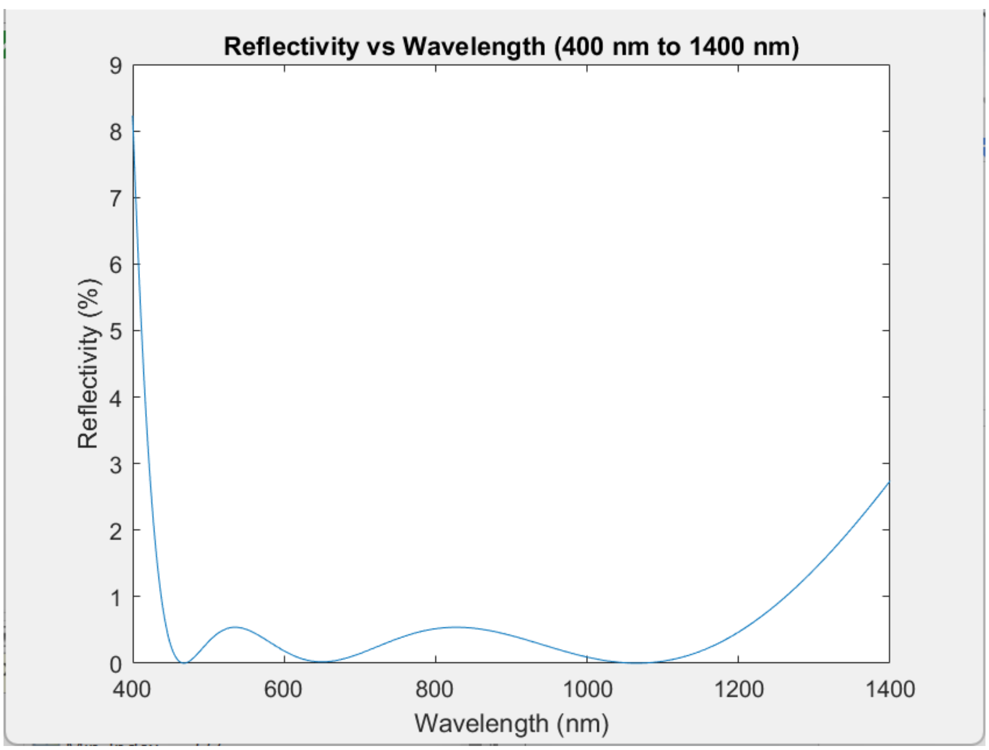

Figure 4.3 - Reflectivity versus Wavelength (400 nm to 1400 nm)

```
Iteration: 985
Iteration: 986
Iteration: 987
Iteration: 988
Iteration: 989
Iteration: 990
Iteration: 991
Iteration: 992
Iteration: 993
Iteration: 994
Iteration: 995
Iteration: 996
Iteration: 997
Iteration: 998
Iteration: 999
Iteration: 1000

Optimal n_1 = 1.3075
Optimal n_2 = 1.8700
Optimal n_3 = 2.6800
Total Power Production (200 nm to 2200nm) = 964.6009 Watts
```

Figure 4.4 - Optimized n_1, n_2, and n_3 Values for 200 nm to 2200 nm

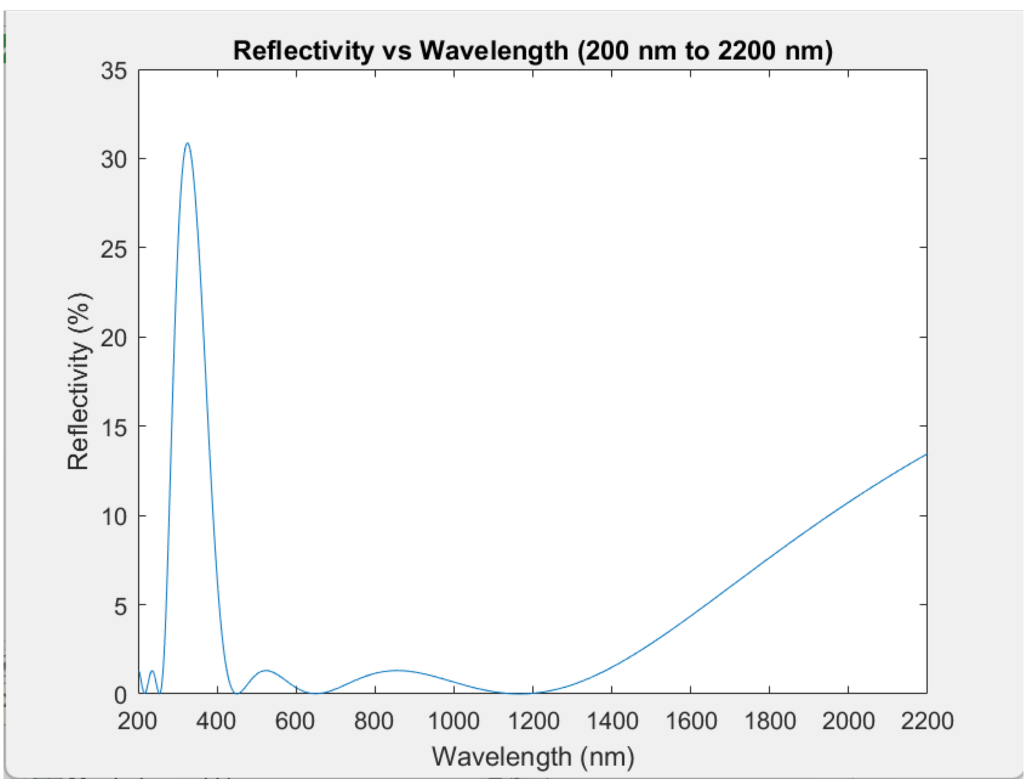

Figure 4.5 - Reflectivity versus Wavelength (200 nm to 2200 nm)

Code Appendix

The following code was used:

Calculate_n_2_for_2_layer.m

```
n_0 = 1;
n_1 = 1.4;
n_3 = 3.5;
n_cell = 3.5;
lambda_C = 650;
r01 = (n_0 - n_1)/(n_0 + n_1);
t01 = 2*n_0/(n_0 + n_1);
Q01 = (1/t01)*([1 r01; r01 1]);
r3S = (n_3 - n_cell)/(n_3 + n_cell);
t3S = 2*n_3/(n_3 + n_cell);
Q3S = (1/t3S)*([1 r3S; r3S 1]);
Delta = pi/2;
P = [exp(1j*Delta) 0; 0 exp(-1j*Delta)];
n_2_range = 0:0.01:4.5;
Store_Reflectance = zeros(1, length(n_2_range));
for i = 1:length(n_2_range)
  n_2 = n_2_range(i);
  r12 = (n_1 - n_2)/(n_1 + n_2);
  t12 = 2*n_1/(n_1 + n_2);
  Q12 = (1/t12)*([1 r12; r12 1]);
  r23 = (n_2 - n_3)/(n_2 + n_3);
  t23 = 2*n_2/(n_2 + n_3);
  Q23 = (1/t23)*([1 r23; r23 1]);
  T = Q01*P*Q12*P*Q23*P*Q3S;
  Gamma = T(2,1)/T(1,1);
  Store_Reflectance(i) = abs(Gamma)^2;
end
[~, Min_Index] = min(Store_Reflectance);
min_n_2 = n_2_range(Min_Index);
plot(n_2_range, Store_Reflectance * 100);
title('Optimal n_2 at lambda_C = 650');
xlabel('n_2 Value');
ylabel('Reflectance %');
fprintf('N_1 = 1.4\nMinimum Reflectance found at n_2 = %.2f\n', min_n_2);
```

Part_2_a_and_b_400_1400.m

```
j = 1j;
n_0 = 1;
n_1 = 1.4;
n_2 = 2.62;
n_3 = 3.5;
Lambda_C = 650;
Lambda_Start = 400;
Lambda_End = 1400;
Lambda_Range = Lambda_Start:Lambda_End;
r01 = (n_0 - n_1)/(n_0 + n_1);
r12 = (n_1 - n_2)/(n_1 + n_2);
r2S = (n_2 - n_3)/(n_2 + n_3);
t01 = 2*n_0/(n_0 + n_1);
t12 = 2*n_1/(n_1 + n_2);
t2S = 2*n_2/(n_2 + n_3);
Q01 = (1/t01)*([1 r01; r01 1]);
Q12 = (1/t12)*([1 r12; r12 1]);
Q2S = (1/t2S)*([1 r2S; r2S 1]);
Deltas = (pi/2)*(Lambda_C./Lambda_Range);
P1 = [exp(j*Deltas); exp(-j*Deltas)];
Reflectance = zeros(size(Lambda_Range));
Power = zeros(size(Lambda_Range));
for i = 1:length(Lambda_Range)
  Lambda = Lambda_Range(i);
  P_Matrix = [P1(1, i) 0; 0 P1(2, i)];

  T = Q01*P_Matrix*Q12*P_Matrix*Q2S;
  Gamma = T(2,1)/T(1,1);

  Reflectance(i) = abs(Gamma)^2;
  Trans = abs(1/T(1,1))^2/(n_0/n_3);
  IRRAD = (6.16*10^15)/((Lambda^5)*(exp(2484/Lambda)-1));
  Power(i) = Trans * IRRAD;
end
plot(Lambda_Range, Reflectance*100);
title('Graph of Reflectivity vs Wavelength (400 nm to 1400 nm)');
xlabel('Wavelength (nm)');
ylabel('Reflectivity (%)');
xlim([Lambda_Start,Lambda_End]);
fprintf('Total Power in Watts = %f\n', sum(Power));
```

Part_2_a_and_b_200_2200.m

```
j = 1j;
n_0 = 1;
n_1 = 1.4;
n_2 = 2.62;
n_3 = 3.5;
Lambda_C = 650;
Lambda_Start = 200;
Lambda_End = 2200;
Lambda_Range = Lambda_Start:Lambda_End;
r01 = (n_0 - n_1)/(n_0 + n_1);
r12 = (n_1 - n_2)/(n_1 + n_2);
r2S = (n_2 - n_3)/(n_2 + n_3);
t01 = 2*n_0/(n_0 + n_1);
t12 = 2*n_1/(n_1 + n_2);
t2S = 2*n_2/(n_2 + n_3);
Q01 = (1/t01)*([1 r01; r01 1]);
Q12 = (1/t12)*([1 r12; r12 1]);
Q2S = (1/t2S)*([1 r2S; r2S 1]);
Deltas = (pi/2)*(Lambda_C./Lambda_Range);
P1 = [exp(j*Deltas); exp(-j*Deltas)];
Reflectance = zeros(size(Lambda_Range));
Power = zeros(size(Lambda_Range));
for i = 1:length(Lambda_Range)
  Lambda = Lambda_Range(i);
  P_Matrix = [P1(1, i) 0; 0 P1(2, i)];

  T = Q01*P_Matrix*Q12*P_Matrix*Q2S;
  Gamma = T(2,1)/T(1,1);

  Reflectance(i) = abs(Gamma)^2;
  Trans = abs(1/T(1,1))^2/(n_0/n_3);
  IRRAD = (6.16*10^15)/((Lambda^5)*(exp(2484/Lambda)-1));
  Power(i) = Trans * IRRAD;
end
plot(Lambda_Range, Reflectance*100);
title('Graph of Reflectivity vs Wavelength (200 nm to 2200 nm)');
xlabel('Wavelength (nm)');
ylabel('Reflectivity (%)');
xlim([Lambda_Start,Lambda_End]);
fprintf('Total Power in Watts = %f\n', sum(Power));
```

Part_2_b.m

```
n_0 = 1;
n_1 = 1.4;
n_2 = 2.6192;
n_3 = 3.5;
g01 = (n_0-n_1)/(n_0+n_1);
g12 = (n_1-n_2)/(n_1+n_2);
g23 = (n_2-n_3)/(n_2+n_3);
t01 = 2*n_0/(n_0+n_1);
t12 = 2*n_1/(n_1+n_2);
t23 = 2*n_2/(n_2+n_3);
lambda_c = 650;
wavelength = 200:1:2199;
L_values = 200:1:2399;
q01 = (1/t01)*[1 g01; g01 1];
q12 = (1/t12)*[1 g12; g12 1];
q23 = (1/t23)*[1 g23; g23 1];
Power = zeros(1, length(wavelength));
for k = 1:length(wavelength)
  L = L_values(k);
  delta_m = (pi/2) * (lambda_c / L);
  p1 = [exp(1i*delta_m) 0; 0 exp(-1i*delta_m)];
  T = q01 * p1 * q12 * p1 * q23;
  g = T(2,1) / T(1,1);
  r = abs(g)^2;
  pow = (((1-r)*(6.16*10^15)) / ((L^5)*(exp(2484/L)-1)));
  Power(k) = pow;
end
wavelength1 = 400:1:1399;
figure(1);
plot(wavelength, Power);
title('Power Transmitted vs Wavelength (400 nm to 1400 nm)');
xlabel('Wavelength (nm)');
ylabel('Power (W/m^2)');
xlim([400,1400]);
figure(2);
plot(wavelength, Power);
title('Power Transmitted vs Wavelength (200 nm to 2200 nm)');
xlabel('Wavelength (nm)');
ylabel('Power (W/m^2)');
xlim([200,2200]);
fprintf('Total Power in Watts (400 nm to 1400 nm) = %f\n', sum(Power(231:1399)));
fprintf('Total Power in Watts (200 nm 2200 nm) = %f\n', sum(Power));
```

Part_2_c_400_1400.m

```
j = 1j;
n_0 = 1;
n_3 = 3.5;
central = 650;
Lambda_Range = 400:1400;
numLambdas = length(Lambda_Range);
IRRAD_Const = 6.16 * 10^15;
Exp_Const = 2484;
n_1_Start = 1;
n_1_End = 3;
n_2_Start = 1;
n_2_End = 3;
Step_Size = 0.4;
Max_Iteration = 5;
numN1 = ceil((n_1_End - n_1_Start) / Step_Size) + 1;
numN2 = ceil((n_2_End - n_2_Start) / Step_Size) + 1;
Store_n_1 = zeros(1, numN1 * numN2);
Store_n_2 = zeros(1, numN1 * numN2);
Store_Total_Power = zeros(1, numN1 * numN2);
for Iteration = 0:Max_Iteration
  idx = 1;
  for n_1 = n_1_Start:Step_Size:n_1_End
    for n_2 = n_2_Start:Step_Size:n_2_End
      Best_Reflec = zeros(1, numLambdas);
      Store_PWR = zeros(1, numLambdas);
      r01 = (n_0 - n_1)/(n_0 + n_1);
      r12 = (n_1 - n_2)/(n_1 + n_2);
      r2S = (n_2 - n_3)/(n_2 + n_3);
      t01 = 2 * n_0 / (n_0 + n_1);
      t12 = 2 * n_1 / (n_1 + n_2);
      t2S = 2 * n_2 / (n_2 + n_3);
      Q01 = (1/t01) * [1 r01; r01 1];
      Q12 = (1/t12) * [1 r12; r12 1];
      Q2S = (1/t2S) * [1 r2S; r2S 1];
      for i = 1:numLambdas
        Lambda = Lambda_Range(i);
        Delta = (pi/2) * (central / Lambda);
        P = [exp(j * Delta) 0; 0 exp(-j * Delta)];
        T = Q01 * P * Q12 * P * Q2S;
        Gamma = T(2,1) / T(1,1);
        Tau = 1 / T(1,1);
        Trans = (abs(Tau)^2) / (n_0 / n_3);
        Reflectance = abs(Gamma)^2;
        IRRAD = IRRAD_Const / ((Lambda^5) * (exp(Exp_Const / Lambda) - 1));
        Power = Trans * IRRAD;
        Store_PWR(i) = Power;
        Best_Reflec(i) = Reflectance;
```

```
            end
            PowerSum = sum(Store_PWR);
            Store_n_1(idx) = n_1;
            Store_n_2(idx) = n_2;
            Store_Total_Power(idx) = PowerSum;
            idx = idx + 1;
        end
    end
    [Best_Power, Pos] = max(Store_Total_Power);
    Best_n_1 = Store_n_1(Pos);
    Best_n_2 = Store_n_2(Pos);
    if Iteration < 5
        n_1_Start = max(Best_n_1 - Step_Size * 2, 1);
        n_1_End = min(Best_n_1 + Step_Size * 2, n_1_End);
        n_2_Start = max(Best_n_2 - Step_Size * 2, 1);
        n_2_End = min(Best_n_2 + Step_Size * 2, n_2_End);
        Step_Size = max(Step_Size / 2, 0.01);
    end
end
figure(1)
plot(Lambda_Range, Best_Reflec * 100);
title('Optimized Graph of Reflectivity vs Wavelength (400 nm to 1400 nm)');
xlabel('Wavelength (nm)');
ylabel('Reflectivity (%)');
xlim([400, 1400]);
fprintf(' Optimal n_1 = %.2f', Best_n_1);
fprintf('\n Optimal n_2 = %.2f', Best_n_2);
fprintf('\n Total Power in Watts = %.4f\n', Best_Power);
```

Part_2_c_200_2200.m

```
j = 1j;
n_0 = 1;
n_3 = 3.5;
central = 650;
Lambda_Range = 200:2200;
numLambdas = length(Lambda_Range);
IRRAD_Const = 6.16 * 10^15;
Exp_Const = 2484;
n_1_Start = 1;
n_1_End = 3;
n_2_Start = 1;
n_2_End = 3;
Step_Size = 0.4;
Max_Iteration = 5;
numN1 = ceil((n_1_End - n_1_Start) / Step_Size) + 1;
numN2 = ceil((n_2_End - n_2_Start) / Step_Size) + 1;
Store_n_1 = zeros(1, numN1 * numN2);
Store_n_2 = zeros(1, numN1 * numN2);
Store_Total_Power = zeros(1, numN1 * numN2);
for Iteration = 0:Max_Iteration
  idx = 1;
  for n_1 = n_1_Start:Step_Size:n_1_End
    for n_2 = n_2_Start:Step_Size:n_2_End
      Best_Reflec = zeros(1, numLambdas);
      Store_PWR = zeros(1, numLambdas);
      r01 = (n_0 - n_1)/(n_0 + n_1);
      r12 = (n_1 - n_2)/(n_1 + n_2);
      r2S = (n_2 - n_3)/(n_2 + n_3);
      t01 = 2 * n_0 / (n_0 + n_1);
      t12 = 2 * n_1 / (n_1 + n_2);
      t2S = 2 * n_2 / (n_2 + n_3);
      Q01 = (1/t01) * [1 r01; r01 1];
      Q12 = (1/t12) * [1 r12; r12 1];
      Q2S = (1/t2S) * [1 r2S; r2S 1];
      for i = 1:numLambdas
        Lambda = Lambda_Range(i);
        Delta = (pi/2) * (central / Lambda);
        P = [exp(j * Delta) 0; 0 exp(-j * Delta)];
        T = Q01 * P * Q12 * P * Q2S;
        Gamma = T(2,1) / T(1,1);
        Tau = 1 / T(1,1);
        Trans = (abs(Tau)^2) / (n_0 / n_3);
        Reflectance = abs(Gamma)^2;
        IRRAD = IRRAD_Const / ((Lambda^5) * (exp(Exp_Const / Lambda) - 1));
        Power = Trans * IRRAD;
        Store_PWR(i) = Power;
        Best_Reflec(i) = Reflectance;
```

```
            end
            PowerSum = sum(Store_PWR);
            Store_n_1(idx) = n_1;
            Store_n_2(idx) = n_2;
            Store_Total_Power(idx) = PowerSum;
            idx = idx + 1;
        end
    end
    [Best_Power, Pos] = max(Store_Total_Power);
    Best_n_1 = Store_n_1(Pos);
    Best_n_2 = Store_n_2(Pos);
    if Iteration < 5
        n_1_Start = max(Best_n_1 - Step_Size * 2, 1);
        n_1_End = min(Best_n_1 + Step_Size * 2, n_1_End);
        n_2_Start = max(Best_n_2 - Step_Size * 2, 1);
        n_2_End = min(Best_n_2 + Step_Size * 2, n_2_End);
        Step_Size = max(Step_Size / 2, 0.01);
    end
end
figure(1)
plot(Lambda_Range, Best_Reflec * 100);
title('Optimized Graph of Reflectivity vs Wavelength (200 nm to 2200 nm)');
xlabel('Wavelength (nm)');
ylabel('Reflectivity (%)');
xlim([200, 2200]);
fprintf(' Optimal n_1 = %.2f', Best_n_1);
fprintf('\n Optimal n_2 = %.2f', Best_n_2);
fprintf('\n Total Power in Watts = %.4f\n', Best_Power);
```

Overlap.m

```
j = 1j;
n_0 = 1;
n_3 = 3.5;
central = 650;
Lambda_Range = 200:2200;
numLambdas = length(Lambda_Range);
IRRAD_Const = 6.16 * 10^15;
Exp_Const = 2484;
n_1_Start = 1;
n_1_End = 3;
n_2_Start = 1;
n_2_End = 3;
Step_Size = 0.4;
Max_Iteration = 5;
numN1 = ceil((n_1_End - n_1_Start) / Step_Size) + 1;
numN2 = ceil((n_2_End - n_2_Start) / Step_Size) + 1;
Store_n_1 = zeros(1, numN1 * numN2);
Store_n_2 = zeros(1, numN1 * numN2);
Store_Total_Power = zeros(1, numN1 * numN2);
for Iteration = 0:Max_Iteration
  idx = 1;
  for n_1 = n_1_Start:Step_Size:n_1_End
    for n_2 = n_2_Start:Step_Size:n_2_End
      Best_Reflec = zeros(1, numLambdas);
      Store_PWR = zeros(1, numLambdas);
      r01 = (n_0 - n_1)/(n_0 + n_1);
      r12 = (n_1 - n_2)/(n_1 + n_2);
      r2S = (n_2 - n_3)/(n_2 + n_3);
      t01 = 2 * n_0 / (n_0 + n_1);
      t12 = 2 * n_1 / (n_1 + n_2);
      t2S = 2 * n_2 / (n_2 + n_3);
      Q01 = (1/t01) * [1 r01; r01 1];
      Q12 = (1/t12) * [1 r12; r12 1];
      Q2S = (1/t2S) * [1 r2S; r2S 1];
      for i = 1:numLambdas
        Lambda = Lambda_Range(i);
        Delta = (pi/2) * (central / Lambda);
        P = [exp(j * Delta) 0; 0 exp(-j * Delta)];
        T = Q01 * P * Q12 * P * Q2S;
        Gamma = T(2,1) / T(1,1);
        Tau = 1 / T(1,1);
        Trans = (abs(Tau)^2) / (n_0 / n_3);
        Reflectance = abs(Gamma)^2;
        IRRAD = IRRAD_Const / ((Lambda^5) * (exp(Exp_Const / Lambda) - 1));
        Power = Trans * IRRAD;
        Store_PWR(i) = Power;
        Best_Reflec(i) = Reflectance;
```

```
            end
            PowerSum = sum(Store_PWR);
            Store_n_1(idx) = n_1;
            Store_n_2(idx) = n_2;
            Store_Total_Power(idx) = PowerSum;
            idx = idx + 1;
        end
    end
    [Best_Power, Pos] = max(Store_Total_Power);
    Best_n_1 = Store_n_1(Pos);
    Best_n_2 = Store_n_2(Pos);
    if Iteration < 5
        n_1_Start = max(Best_n_1 - Step_Size * 2, 1);
        n_1_End = min(Best_n_1 + Step_Size * 2, n_1_End);
        n_2_Start = max(Best_n_2 - Step_Size * 2, 1);
        n_2_End = min(Best_n_2 + Step_Size * 2, n_2_End);
        Step_Size = max(Step_Size / 2, 0.01);
    end
end
figure(1)
plot(Lambda_Range, Best_Reflec * 100, 'b-');
hold on;
fprintf(' Optimal n_1 = %.2f', Best_n_1);
fprintf('\n Optimal n_2 = %.2f', Best_n_2);
fprintf('\n Total Power in Watts = %.4f\n', Best_Power);
j = 1j;
n_0 = 1;
n_1 = 1.4;
n_2 = 2.62;
n_3 = 3.5;
Lambda_C = 650;
Lambda_Start = 200;
Lambda_End = 2200;
Lambda_Range = Lambda_Start:Lambda_End;
r01 = (n_0 - n_1)/(n_0 + n_1);
r12 = (n_1 - n_2)/(n_1 + n_2);
r2S = (n_2 - n_3)/(n_2 + n_3);
t01 = 2*n_0/(n_0 + n_1);
t12 = 2*n_1/(n_1 + n_2);
t2S = 2*n_2/(n_2 + n_3);
Q01 = (1/t01)*([1 r01; r01 1]);
Q12 = (1/t12)*([1 r12; r12 1]);
Q2S = (1/t2S)*([1 r2S; r2S 1]);
Deltas = (pi/2)*(Lambda_C./Lambda_Range);
P1 = [exp(j*Deltas); exp(-j*Deltas)];
Reflectance = zeros(size(Lambda_Range));
Power = zeros(size(Lambda_Range));
for i = 1:length(Lambda_Range)
    Lambda = Lambda_Range(i);
```

```
  P_Matrix = [P1(1, i) 0; 0 P1(2, i)];

  T = Q01*P_Matrix*Q12*P_Matrix*Q2S;
  Gamma = T(2,1)/T(1,1);

  Reflectance(i) = abs(Gamma)^2;
  Trans = abs(1/T(1,1))^2/(n_0/n_3);
  IRRAD = (6.16*10^15)/((Lambda^5)*(exp(2484/Lambda)-1));
  Power(i) = Trans * IRRAD;
end
plot(Lambda_Range, Reflectance * 100, 'r-');
title('Graph of Reflectivity vs Wavelength (200 nm to 2200 nm)');
xlabel('Wavelength (nm)');
ylabel('Reflectivity (%)');
xlim([Lambda_Start,Lambda_End]);
legend('Optimized Reflectivity', 'Standard Reflectivity');
hold off;
fprintf(' Total Power in Watts = %f\n', sum(Power));
```

Calculate_n_2_for_3_layer.m

```
n_0 = 1;
n_1 = 1.4;
n_3 = 3.15;
n_cell = 3.5;
lambda_C = 650;
r01 = (n_0 - n_1)/(n_0 + n_1);
t01 = 2*n_0/(n_0 + n_1);
Q01 = (1/t01)*([1 r01; r01 1]);
r3S = (n_3 - n_cell)/(n_3 + n_cell);
t3S = 2*n_3/(n_3 + n_cell);
Q3S = (1/t3S)*([1 r3S; r3S 1]);
Delta = pi/2;
P = [exp(1j*Delta) 0; 0 exp(-1j*Delta)];
n_2_range = 0:0.01:4.5;
Store_Reflectance = zeros(1, length(n_2_range));
for i = 1:length(n_2_range)
  n_2 = n_2_range(i);
  r12 = (n_1 - n_2)/(n_1 + n_2);
  t12 = 2*n_1/(n_1 + n_2);
  Q12 = (1/t12)*([1 r12; r12 1]);
  r23 = (n_2 - n_3)/(n_2 + n_3);
  t23 = 2*n_2/(n_2 + n_3);
  Q23 = (1/t23)*([1 r23; r23 1]);
  T = Q01*P*Q12*P*Q23*P*Q3S;
  Gamma = T(2,1)/T(1,1);
  Store_Reflectance(i) = abs(Gamma)^2;
end
[~, Min_Index] = min(Store_Reflectance);
min_n_2 = n_2_range(Min_Index);
plot(n_2_range, Store_Reflectance * 100);
title('Reflectivity versus n_2 Values for lambda_C = 650 nm');
xlabel('n_2 Value');
ylabel('Reflectivity (%)');
fprintf('n_1 = 1.4\nMinimum Reflectivity found at n_2 = %.2f\nn_3 = 3.15\n', min_n_2);
```

Part_4_a_b.m

```
j = 1j;
n_0 = 1;
n_1 = 1.4;
n_3 = 3.15;
n_4 = 3.5;
Center = 650;
num_N_2 = 300;
r01 = (n_0 - n_1) / (n_0 + n_1);
r3S = (n_3 - n_4) / (n_3 + n_4);
t01 = 2 * n_0 / (n_0 + n_1);
t3S = 2 * n_3 / (n_3 + n_4);
Q01 = (1 / t01) * [1 r01; r01 1];
Q3S = (1 / t3S) * [1 r3S; r3S 1];
Store_n_2 = linspace(1.4, 3, num_N_2);
Store_Total_Power_1 = computeTotalPower(400, 1400, Q01, Q3S, n_0, n_1, n_3, n_4, Center, num_N_2,
Store_n_2);
Store_Total_Power_2 = computeTotalPower(200, 2200, Q01, Q3S, n_0, n_1, n_3, n_4, Center, num_N_2,
Store_n_2);
plot(Store_n_2, Store_Total_Power_1, 'r', Store_n_2, Store_Total_Power_2, 'b');
title('Total Power Transmitted vs Varying n_2 Values');
xlabel('n_2');
ylabel('Total Power (W/m^2)');
legend('Wavelength 200-2200nm', 'Wavelength 400-1400nm');
[max_Power_1, maxY_1] = max(Store_Total_Power_1);
fprintf(' Maximum power transmitted in Watts (Wavelength 400-1400nm) = %.2fW', max_Power_1);
fprintf('\n n_2 value = %.2f', Store_n_2(maxY_1));
[max_Power_2, maxY_2] = max(Store_Total_Power_2);
fprintf('\n Maximum power transmitted in Watts (Wavelength 200-2200nm) = %.2fW', max_Power_2);
fprintf('\n n_2 value = %.2f \n', Store_n_2(maxY_2));
hold on;
function Store_Total_Power = computeTotalPower(Lambda_Start, Lambda_End, Q01, Q3S, n_0, n_1, n_3, n_4,
Center, numN2, Store_n_2)
  Lambda_Array = Lambda_Start:Lambda_End;
  Delta_Array = (pi / 2) * (Center ./ Lambda_Array);
  IRRAD_Array = 6.16 * 10^15 ./ (Lambda_Array .^ 5 .* (exp(2484 ./ Lambda_Array) - 1));
  Store_Total_Power = zeros(1, numN2);
  for idx = 1:numN2
    N2 = Store_n_2(idx);
    r12 = (n_1 - N2) / (n_1 + N2);
    r23 = (N2 - n_3) / (N2 + n_3);
    t12 = 2 * n_1 / (n_1 + N2);
    t23 = 2 * N2 / (N2 + n_3);
    Q12 = (1 / t12) * [1 r12; r12 1];
    Q23 = (1 / t23) * [1 r23; r23 1];
    P = [exp(j * Delta_Array); exp(-j * Delta_Array)];
    T = zeros(2, length(Lambda_Array));
    for i = 1:length(Lambda_Array)
```

```
        P1 = [P(1, i) 0; 0 P(2, i)];
        T(:, i) = Q01 * P1 * Q12 * P1 * Q23 * P1 * Q3S * [1; 0];
    end
    Tau = 1 ./ T(1, :);
    Trans = (abs(Tau) .^ 2) * (n_4 / n_0);
    Store_PWR = Trans .* IRRAD_Array;
    Store_Total_Power(idx) = sum(Store_PWR);
  end
end
```

Best_power.m

```
n_0 = 1;
n_4 = 3.5;
Lambda_C = 650;
Lambda_Start = 400;
Lambda_End = 1400;
Step_Size = 0.1;
Max_Iteration = 1000;
Max_Size = ceil((3 - 1) / 0.4 + 1) ^ 3;
Store_n_1 = zeros(1, Max_Size);
Store_n_2 = zeros(1, Max_Size);
Store_n_3 = zeros(1, Max_Size);
Store_Total_Power = zeros(1, Max_Size);
Best_Powers = zeros(1, Max_Iteration);
Best_Power=0;
n_1_Start = 1; n_1_End = 3;
n_2_Start = 1; n_2_End = 3;
n_3_Start = 1; n_3_End = 3;
index = 1;
for Iteration = 1:Max_Iteration
  fprintf('Iteration: %d\n', Iteration);
  for n_1 = n_1_Start:Step_Size:n_1_End
     for n_2 = n_2_Start:Step_Size:n_2_End
        for n_3 = n_3_Start:Step_Size:n_3_End
           Store_PWR = zeros(1, Lambda_End - Lambda_Start + 1);
           Reflectance = zeros(1, Lambda_End - Lambda_Start + 1);
           for Lambda = Lambda_Start:Lambda_End
              r01 = (n_0 - n_1)/(n_0 + n_1);
              r12 = (n_1 - n_2)/(n_1 + n_2);
              r23 = (n_2 - n_3)/(n_2 + n_3);
              r3S = (n_3 - n_4)/(n_3 + n_4);
              t01 = 2 * n_0 / (n_0 + n_1);
              t12 = 2 * n_1 / (n_1 + n_2);
              t23 = 2 * n_2 / (n_2 + n_3);
              t3S = 2 * n_3 / (n_3 + n_4);
              Q01 = (1 / t01) * [1 r01; r01 1];
              Q12 = (1 / t12) * [1 r12; r12 1];
              Q23 = (1 / t23) * [1 r23; r23 1];
              Q3S = (1 / t3S) * [1 r3S; r3S 1];
              Delta = (pi / 2) * (Lambda_C / Lambda);
              P1 = [exp(1i * Delta) 0; 0 exp(-1i * Delta)];
              P2 = P1;
              P3 = P1;
              T = Q01 * P1 * Q12 * P2 * Q23 * P3 * Q3S;
              Gamma = T(2, 1) / T(1, 1);
              Tau = 1 / T(1, 1);
              Reflectance(Lambda - Lambda_Start + 1) = abs(Gamma)^2;
              Trans = abs(Tau)^2 / (n_0 / n_4);
```

```
            IRRAD = (6.16 * 10^15) / (Lambda^5 * (exp(2484 / Lambda) - 1));
            Power = Trans * IRRAD;
            Store_PWR(Lambda - Lambda_Start + 1) = Power;
            Best_Powers(Iteration) = Best_Power;
          end
          Store_n_1(index) = n_1;
          Store_n_2(index) = n_2;
          Store_n_3(index) = n_3;
          Store_Total_Power(index) = sum(Store_PWR);
          index = index + 1;
        end
      end
    end
    [Best_Power, Pos] = max(Store_Total_Power);
    b_n_1 = Store_n_1(Pos); b_n_2 = Store_n_2(Pos); b_n_3 = Store_n_3(Pos);
    n_1_Start = max(1, b_n_1 - Step_Size * 2);
    n_1_End = min(3, b_n_1 + Step_Size * 2);
    n_2_Start = max(1, b_n_2 - Step_Size * 2);
    n_2_End = min(3, b_n_2 + Step_Size * 2);
    n_3_Start = max(1, b_n_3 - Step_Size * 2);
    n_3_End = min(3, b_n_3 + Step_Size * 2);
    Step_Size = max(0.01, Step_Size / 2);
end
plot(Lambda_Start:Lambda_End, Reflectance * 100);
title('Reflectivity vs Wavelength (400 nm to 1400 nm)');
xlabel('Wavelength (nm)');
ylabel('Reflectivity (%)');
xlim([Lambda_Start, Lambda_End]);
fprintf('\nOptimal n_1 = %.4f\nOptimal n_2 = %.4f\nOptimal n_3 = %.4f\nTotal Power Production (400 nm to 1400nm) = %.4f Watts\n', b_n_1, b_n_2, b_n_3, Best_Power);
```

Best_power_parallel_processing.m

```
n_0 = 1;
n_4 = 3.5;
Lambda_C = 650;
Lambda_Start = 400;
Lambda_End = 1400;
Step_Size = 0.1;
Max_Iteration = 1000;
n_1_Start = 1; n_1_End = 3;
n_2_Start = 1; n_2_End = 3;
n_3_Start = 1; n_3_End = 3;
if isempty(gcp('nocreate'))
  parpool(24);
end
for Iteration = 1:Max_Iteration
  fprintf('Iteration: %d\n', Iteration);
  n_1_Size = ceil((n_1_End - n_1_Start) / Step_Size + 1);
  n_2_Size = ceil((n_2_End - n_2_Start) / Step_Size + 1);
  n_3_Size = ceil((n_3_End - n_3_Start) / Step_Size + 1);
  Max_Size = n_1_Size * n_2_Size * n_3_Size;
  Temp_Store_Total_Power = zeros(1, Max_Size);
  Temp_Store_n_1 = zeros(1, Max_Size);
  Temp_Store_n_2 = zeros(1, Max_Size);
  Temp_Store_n_3 = zeros(1, Max_Size);
  parfor idx = 1:Max_Size
    [i1, i2, i3] = ind2sub([n_1_Size, n_2_Size, n_3_Size], idx);

    n_1 = n_1_Start + (i1 - 1) * Step_Size;
    n_2 = n_2_Start + (i2 - 1) * Step_Size;
    n_3 = n_3_Start + (i3 - 1) * Step_Size;
    Store_PWR = zeros(1, Lambda_End - Lambda_Start + 1);
    for Lambda = Lambda_Start:Lambda_End
      r01 = (n_0 - n_1)/(n_0 + n_1);
      r12 = (n_1 - n_2)/(n_1 + n_2);
      r23 = (n_2 - n_3)/(n_2 + n_3);
      r3S = (n_3 - n_4)/(n_3 + n_4);
      t01 = 2 * n_0 / (n_0 + n_1);
      t12 = 2 * n_1 / (n_1 + n_2);
      t23 = 2 * n_2 / (n_2 + n_3);
      t3S = 2 * n_3 / (n_3 + n_4);
      Q01 = (1 / t01) * [1 r01; r01 1];
      Q12 = (1 / t12) * [1 r12; r12 1];
      Q23 = (1 / t23) * [1 r23; r23 1];
      Q3S = (1 / t3S) * [1 r3S; r3S 1];
      Delta = (pi / 2) * (Lambda_C / Lambda);
      P1 = [exp(1i * Delta) 0; 0 exp(-1i * Delta)];
      P2 = P1;
      P3 = P1;
```

```
        T = Q01 * P1 * Q12 * P2 * Q23 * P3 * Q3S;
        Gamma = T(2, 1) / T(1, 1);
        Tau = 1 / T(1, 1);
        Trans = abs(Tau)^2 / (n_0 / n_4);
        IRRAD = (6.16 * 10^15) / (Lambda^5 * (exp(2484 / Lambda) - 1));
        Power = Trans * IRRAD;
        Store_PWR(Lambda - Lambda_Start + 1) = Power;
      end
      Temp_Store_n_1(idx) = n_1;
      Temp_Store_n_2(idx) = n_2;
      Temp_Store_n_3(idx) = n_3;
      Temp_Store_Total_Power(idx) = sum(Store_PWR);
    end
    [Best_Power, Pos] = max(Temp_Store_Total_Power);
    Best_Powers(Iteration) = Best_Power;
    b_n_1 = Temp_Store_n_1(Pos);
    b_n_2 = Temp_Store_n_2(Pos);
    b_n_3 = Temp_Store_n_3(Pos);
    n_1_Start = max(1, b_n_1 - Step_Size);
    n_1_End = min(3, b_n_1 + Step_Size);
    n_2_Start = max(1, b_n_2 - Step_Size);
    n_2_End = min(3, b_n_2 + Step_Size);
    n_3_Start = max(1, b_n_3 - Step_Size);
    n_3_End = min(3, b_n_3 + Step_Size);
    Step_Size = max(0.01, Step_Size / 2);
end
figure;
plot(Lambda_Start:Lambda_End, Reflectance * 100);
title('Reflectance vs Wavelength');
xlabel('Wavelength (nm)');
ylabel('Reflectance (%)');
fprintf('\nOptimal N1 = %.4f\nOptimal N2 = %.4f\nOptimal N3 = %.4f\nTotal Power = %.4f Watts\n', b_n_1,
b_n_2, b_n_3, Best_Power);
```