



Norwegian University of
Science and Technology

Engineered Surfaces for Redirection of Light

Øystein Walle

Master of Science in Physics and Mathematics

Submission date: July 2011

Supervisor: Ingve Simonsen, IFY

Acknowledgements

This master thesis was written in the spring of 2011 as the final work of the Physics and Mathematics engineering programme at the Norwegian University of Science and Technology in Trondheim. My supervisor was Ingve Simonsen who deserves my sincere thanks for his help. He also made this thesis possible in writing the simulation program `Maxwell11D` which I have made use of greatly.

I want to thank my fellow students Ole Martin Brende and Magnus Nord for invaluable feedback on this text. Additionally, I thank professor Morten Kildemo for feedback on my project thesis, the lessons of which I hope I have put to use here.

Abstract

It is of interest to construct windows that can spread the transmitted light in a specified manner. The Kirchhoff approximation in the geometrical optical limit in combination with a chosen general form of the window surfaces yields the profile for the window surfaces, letting us specify how the light should be spread.

The probability distribution function for the slopes of a window surface consisting of joined line segments was implemented in the simulation software `Maxwell1D`. Simulations show the feasibility of such windows. However they do not respond well when subject to light incident with another angle of incidence than the angle in mind.

By using Snell's law to compensate for using a simpler system, the time needed for the simulations can be greatly reduced while simultaneously obtaining a higher accuracy in the results.

Contents

1	Introduction	1
1.1	Overview	1
1.2	Theory	4
1.2.1	Maxwell's equations	4
1.2.2	Constitutive relations	5
1.2.3	The wave equation	5
1.2.4	Boundary conditions	7
1.2.5	Polarization and the Fresnel equations	7
2	Randomly rough surfaces	9
2.1	Overview of the system	9
2.2	Mean Differential Transmission Coefficient	10
2.3	The surfaces	13
2.4	The rejection method	19
3	Results	23
3.1	Preliminary work	23
3.2	Simulation overview	25
3.3	Results	26
3.3.1	Comparison with the results of Maradudin et al. [1]	26
3.3.2	Non-normal incidence	27
3.3.3	Results from initial window simulations.	28
3.3.4	Dependency on wavelength and polarization	29
3.3.5	Taking a shortcut	31
3.3.6	Non-ideal angles of incidence	36
4	Conclusion	39
4.1	Summary	39

Contents

4.2 Further work	39
A Mathematical relations and definitions	41
B Source code excerpts	43
B.1 Excerpts from utility programs	43
B.2 Excerpts from Maxwell11D	47

1.1 Overview

The reflection of light is a topic introduced early in science classes. At an introductory level a perfectly flat surface is assumed. Upon making this assumption one will find that the angle the incident light makes with the surface must equal the angle of the reflected light. Furthermore, the angle of transmitted light can be calculated by Snell's law [2]. Figure 1.1 shows this common assumption. In reality this is not so simple.

When you shine your flashlight on an object you expect it to appear illuminated. This is the case for most objects in real life. If the surface of the object were perfectly flat, however, you would only see that illumination by looking directly into the reflected light. By looking at it from any other angle you wouldn't see the illumination. In fact, a large amount of the light does come back towards you which makes the object appear

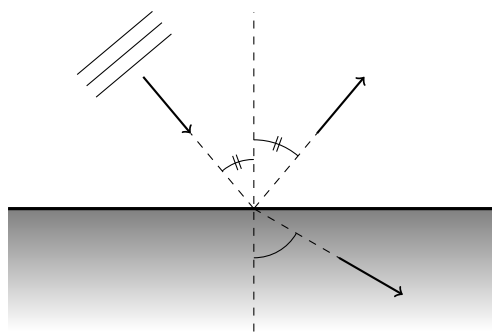


Figure 1.1: Reflection from a flat surface. The incoming wave fronts on the upper left are propagating in the direction of the arrow. As indicated, the angles relative to the surface normal of the incident and reflected light are equal. The angle of the transmitted light may be different.

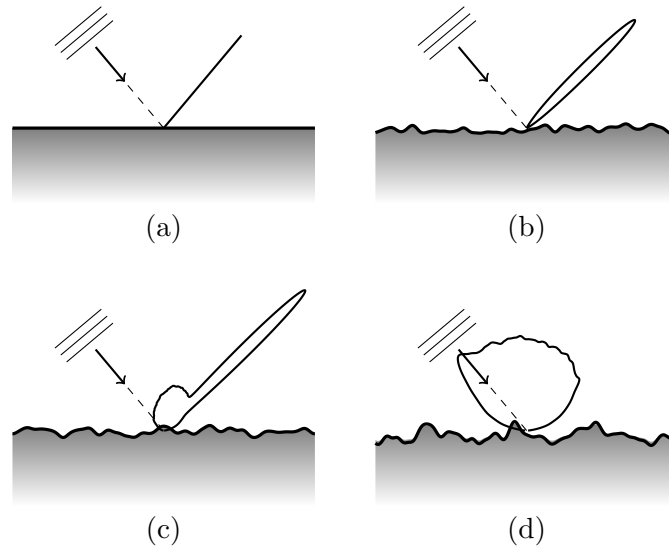


Figure 1.2: When the surface is perfectly flat, shown in (a), we have the same situation as in Fig. 1.1. When the surface is rough the light is no longer reflected in one direction but is spread in many directions with varying intensity. As the roughness increases (from (b) to (d)) the light becomes less and less directed. The plumes shown here exemplifies the reflection from a small area on the surface.

brighter. This is because the surface of the object is rough.

In reality *all* surfaces are rough at some scale. For example, by touch it is simple to assert that the surface of sandpaper is rough, in contrast to regular printing paper. By looking at the printing paper under a microscope, however, one will notice that the surface of the paper is indeed rough. When the surface of an object is rough the light reflected by the object is sent in all possible directions with varying intensity. Figure 1.2 qualitatively shows the effect roughness will have on the reflection of light from a surface. The rougher the surface is, the more spread out the reflected light becomes. The light that is transmitted through a transparent object is also affected similarly.

Optical elements that manipulate light has many uses. A common example is the lenses found in any camera or projector. Such lenses can be used to make corrections for e.g. chromatic aberrations. The practical focus in this thesis will be windows that can redirect light.

It is well known that ambient lighting can affect the human body in several ways. For example, our metabolism is affected if we are exposed to high amounts of electric lighting. This can lead to accelerated weight gain, because our cardiac rhythm is disturbed [3]. Research also shows that we *prefer* to work in lighting that follows the daylight cycle instead of a constant level of light [4, 5].

Seasonal Affective Disorder is a mood disorder that affects otherwise mentally healthy people. They can experience depressive symptoms if they are exposed to too little or too much sunlight. Usually this happens during summers or winters but it can also happen when one's workplace is poorly lit [4, 5]. A high level of luminance is often

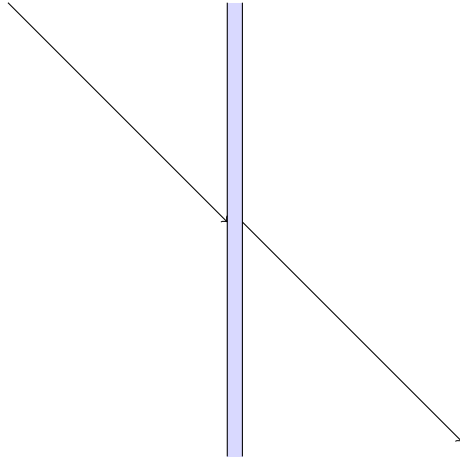


Figure 1.3: A regular window. The light is incident on the glass and slightly changes direction upon transmission into the glass. On exiting the glass the same happens but in the opposite direction and the light continues on in the same direction as before being transmitted through the glass.

desirable, which is something normal daylight can easily provide.

Humans' tolerance to light also differs depending on the type of light. We tolerate a higher level of luminance if the light is natural. We also tolerate a *lower* level of luminance given that the light is natural [4, 5].

From the above it is clear that there are both the possibility of and good reason to investigate the concept of designing surfaces and windows in order to redirect incoming light in a favourable manner. By design one can for instance make the surface of a window so that part of the daylight from the outside is directed towards the ceiling of the room inside. This may reduce the need for artificial lighting indoors, improving indoor lighting conditions and possibly also saving energy.

In this thesis investigate the concept of windows whose behaviour we can control is investigated. By choosing how the light should be transmitted, it is possible to calculate how the surface should be constructed. Figure 1.3 illustrates a typical window. For all practical purposes the light passes straight through it.

Consider the window in Fig. 1.4. The light transmitted through it is redirected to the ceiling. This will distribute the ambient light in a favourable manner compared to regular windows. In addition, the ceiling inside could be painted with a special paint that aids in distributing the light further.

In my project thesis [6] I investigated the effect of roughness on the reflected light, similarly to what is depicted in Fig. 1.1. In this thesis I will focus instead on the light that is transmitted through an object with a rough surface. Instead of a system with two media, as in Fig. 1.1, the system considered here consists of three media. Furthermore, in this thesis the problem is investigated from "the other side." Instead of focusing on how the light is affected by changing the surface, we rather decide how the light should be affected, and work backwards to construct the surface.

Studies investigating engineered surfaces have been performed before to some extent.

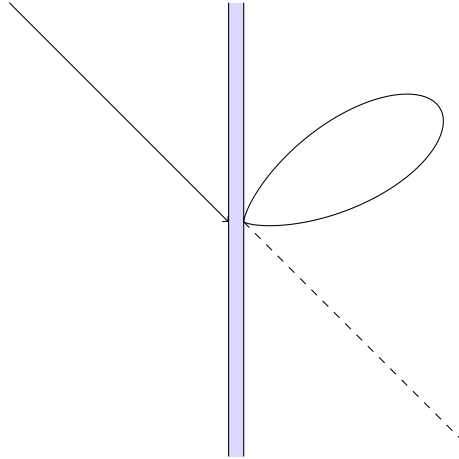


Figure 1.4: By designing the window we can make it reflect the light onto e.g. the ceiling instead of passing straight through like in Fig. 1.3. This will distribute ambient lighting efficiently in the room. Here the light is redirected upward and spread unevenly.

Leskova et al. [7] considers a surface of consecutive grooves designed to uniformly reflect normally incident light within a range of the angle of reflection. Maradudin et al. [1] continued by looking at the transmission of light through a similar surface. Their results will be compared with the results achieved here later.

The practical problem of windows in buildings has also received attention. One advanced study was done by Lim and Kim [8] who did a thorough computer simulation on the use of shading devices on an apartment building in an attempt to improve the indoor lighting conditions. This is a very complicated solution. A simpler solution would be to make windows redirect the light. A pursuit of such a design for windows was made in this thesis in the form of numerical simulations solving the equations governing the physical phenomena involved. The next section provides an overview of the parts of the electromagnetic theory needed.

The numerical simulations will rely on generation of randomly rough surfaces. These surfaces will play the role of the window surfaces. Chapter 2 provides a review of such randomly rough surfaces and the method of generating a random surface is discussed.

The numerical simulations and the results thereof are discussed in Chapter 3. A summary and a short discussion on future work is found in Chapter 4.

1.2 Theory

1.2.1 Maxwell's equations

The topic of this project falls entirely under the scope of classical electromagnetism. The fundamental equations in this field are Maxwell's equations. They only hold in vacuum and for a collection of point charges. In matter there can typically be *many* charges which complicate the solutions. By introducing quantities representing the point

charges collective behaviour, Maxwell's equations can be written as [2]:

$$\nabla \cdot \mathbf{D} = \rho, \quad (1.1a)$$

$$\nabla \cdot \mathbf{B} = 0, \quad (1.1b)$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}, \quad (1.1c)$$

$$\nabla \times \mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t}. \quad (1.1d)$$

The vector fields \mathbf{E} and \mathbf{H} are called the *electric field* and the *magnetic field* respectively, and these are the fields that combined give the *electromagnetic field*. The vector fields \mathbf{D} and \mathbf{B} are called the *electric displacement* and the *magnetic induction* respectively. The fields \mathbf{D} and \mathbf{H} are the fields introduced to represent the collective behaviour of the point charges. The quantities ρ and \mathbf{J} are the free charge density and the free charge current density respectively.

1.2.2 Constitutive relations

Since the fields \mathbf{D} and \mathbf{H} were introduced to deal with the collective behaviour of many point charges there are relationships between \mathbf{D} and \mathbf{E} , and between \mathbf{H} and \mathbf{B} . These are the *constitutive relations* [2]:

$$\mathbf{D} = \epsilon \mathbf{E} = \epsilon_r \epsilon_0 \mathbf{E} \quad (1.2a)$$

$$\mathbf{B} = \mu \mathbf{H} = \mu_r \mu_0 \mathbf{H} \quad (1.2b)$$

Here ϵ_0 and μ_0 are called the *vacuum permittivity* and *vacuum permeability* respectively. The quantity ϵ_0 is found experimentally while μ_0 is defined. They are

$$\epsilon_0 \approx 8.854 \times 10^{-12} \text{ F m}^{-1}$$

and

$$\mu_0 \equiv 4\pi \times 10^{-7} \text{ H m}^{-1}.$$

The factors ϵ_r and μ_r are dimensionless quantities which are the ratios between the permittivities and permeabilities in the current medium and in vacuum. In the most general case, these are tensors which can depend on any number of physical attributes. In this thesis, however, they are scalars and may depend on the frequency of the incoming light.

1.2.3 The wave equation

All waves obey what is called the wave equation. It can be written

$$\frac{\partial^2 A}{\partial t^2} = v^2 \frac{\partial^2 A}{\partial x^2},$$

1 Introduction

in the one-dimensional case. The quantity A is the amplitude of the wave. What this represents physically depends on the type of wave. For e.g. sound waves it is the difference in air pressure from the ambient pressure. For electromagnetic waves it is the strength of the electric or magnetic field. The quantity v is the propagation speed of the wave. For sound waves this is typically 343 m s^{-1} while for electromagnetic waves it is the speed of light. A similar equation for the electromagnetic field can be derived with the help of Eq. (1.1).

In media where there are no free charges and no free current, Eq. (1.1a) reduces to $\nabla \cdot \mathbf{D} = 0$. This also means that we can simply substitute \mathbf{D} with \mathbf{E}/ϵ and \mathbf{H} with $\mu\mathbf{B}$. This gives $\nabla \cdot \mathbf{E} = 0$ and Eq. (1.1d) becomes $\nabla \times \mathbf{B} = \mu\epsilon \frac{\partial \mathbf{E}}{\partial t}$. From Eq. (1.1a) we know that applying the vector identity in Eq. (A.1) on the electric field will yield $\nabla \times \nabla \times \mathbf{E} = \nabla^2 \mathbf{E}$. Further, by Eq. (1.1c):

$$\begin{aligned} \nabla \times \nabla \times \mathbf{E} &= \nabla \times \left(-\frac{\partial \mathbf{B}}{\partial t} \right) \\ &= -\nabla \times \mathbf{B} \\ &= -\epsilon\mu \frac{\partial^2 \mathbf{E}}{\partial t^2} \end{aligned}$$

By then combining the results we arrive at

$$\nabla^2 \mathbf{E} = \epsilon\mu \frac{\partial^2 \mathbf{E}}{\partial t^2} \quad (1.3)$$

which is the wave equation for the electric field. This shows that the electric field can propagate as a wave. Its speed is $1/\sqrt{\epsilon\mu}$ or $c/\sqrt{\epsilon_r\mu_r}$ where c is the speed of light in vacuum. The equivalent equation for the magnetic induction can be found by applying the same vector identity used above to Eq. (1.1d).

One possible solution to Eq. (1.3) is

$$\mathbf{E} = \mathbf{E}_0 \exp(i\mathbf{k} \cdot \mathbf{r} - i\omega t),$$

which is a plane wave with wave vector \mathbf{k} and angular frequency ω . The wave vector gives the direction the wave is propagating in as well as its wavelength $\lambda = |\mathbf{k}|/2\pi$. If and only if

$$|\mathbf{k}|^2 = \epsilon\mu \frac{\omega^2}{c^2} \quad (1.4)$$

is fulfilled, is the solution proposed above a proper solution. This is called the *dispersion relation* for a wave in a homogeneous medium. Since $\nabla \cdot \mathbf{E} = 0$ it is a requirement that $\mathbf{r} \times \mathbf{E}_0$. This means that the electromagnetic wave is a *transverse* wave. Furthermore it can be shown that

$$\mathbf{B}_0 = \frac{k}{\omega} (\mathbf{r} \times \mathbf{E}_0)$$

which shows that \mathbf{E} and \mathbf{B} are perpendicular to each other and that they must be in phase. When considering scattering from a surface it is convenient to express \mathbf{k} as a sum of two vectors where one is perpendicular to the surface and the other is parallel to it,

$$\mathbf{k} = \mathbf{k}_\perp + \mathbf{k}_\parallel.$$

They are therefore also perpendicular to each other. By aligning the chosen coordinate system to these vectors we write $k^2 = k_{\perp}^2 + k_{\parallel}^2$. Inserting this into the dispersion relation yields

$$k_{\perp}^2 + k_{\parallel}^2 = \epsilon\mu \frac{\omega^2}{c^2}.$$

It is possible for either of the two components to grow larger than $|\mathbf{k}|$ itself. In that case the other component will be imaginary resulting in an *evanescent* wave in that direction, meaning that the wave will decrease in amplitude exponentially as it propagates through the medium. Here, k_{\perp} may become imaginary whereas k_{\parallel} stays real.

1.2.4 Boundary conditions

When an electromagnetic wave arrives at a boundary between two media it is subject to *boundary conditions*. From Maxwell's equations certain relations can be derived which impose requirements on the electromagnetic field at the boundary. For this purpose the integral form of Maxwell's equations is useful [2] and they read:

$$\oint_{\mathcal{S}} \mathbf{E} \cdot d\mathbf{A} = Q, \quad (1.5a)$$

$$\oint_{\mathcal{S}} \mathbf{B} \cdot d\mathbf{A} = 0, \quad (1.5b)$$

$$\oint_{\partial\mathcal{S}} \mathbf{E} \cdot d\mathbf{l} = -\frac{d}{dt} \int_{\mathcal{S}} \mathbf{B} \cdot d\mathbf{A}, \quad (1.5c)$$

$$\oint_{\partial\mathcal{S}} \mathbf{H} \cdot d\mathbf{l} = I + \frac{d}{dt} \int_{\mathcal{S}} \mathbf{D} \cdot d\mathbf{A}. \quad (1.5d)$$

In the first two equations \mathcal{S} is any *closed* surface and Q is the charge enclosed by it. In the last two equations \mathcal{S} is any surface (not necessarily closed) with circumference $\partial\mathcal{S}$ and I is the free current passing through $\partial\mathcal{S}$. Griffiths [2] shows that

$$D_1^{\perp} - D_2^{\perp} = \sigma_f \quad (1.6a)$$

$$B_1^{\perp} - B_2^{\perp} = 0 \quad (1.6b)$$

$$\mathbf{E}_1^{\parallel} - \mathbf{E}_2^{\parallel} = 0 \quad (1.6c)$$

$$\mathbf{H}_1^{\parallel} - \mathbf{H}_2^{\parallel} = \mathbf{J}_f \times \hat{\mathbf{n}} \quad (1.6d)$$

where σ_f is the charge density, \mathbf{J}_f is the current density and $\hat{\mathbf{n}}$ is a unit vector pointing from medium 2 to medium 1. We see in Eqs. (1.6b) and (1.6c) that \mathbf{E}^{\parallel} and B_1^{\perp} are continuous at the boundary, while \mathbf{H}_1^{\parallel} and D_1^{\perp} are discontinuous as shown in Eqs. (1.6a) and (1.6d).

1.2.5 Polarization and the Fresnel equations

Polarization refers to the orientation of the electric and magnetic fields relative to the plane of incidence, which is the plane spanned of the normal vector of the interface and

1 Introduction

the wave vector of the incident light. In all the figures shown in the previous section the plane of incidence is therefore the plane of the paper.

The polarization of light can take several forms. In this thesis we will concern ourselves with two. One is when electric field is parallel to the plane of incidence and the magnetic field is normal on the plane of incidence. This is called p-polarization. The other polarization is when the electric field is normal on the plane of incidence and the magnetic field is parallel to the plane of incidence. The light is then s-polarized.

Throughout the symbol ν will indicate the polarization of the light discussed at that point. We write either $\nu = \text{p}$ or $\nu = \text{s}$. When ν itself appears as a subscript it indicates that it holds for both forms of polarization.

The *index of refraction* for a medium is defined as

$$n = \sqrt{\epsilon_r \mu_r}.$$

When light is incident on an interface between two media with different indices of refraction, the reflection and transmission coefficients give the ratios between the intensities of the reflected and transmitted light, and the intensity of the incident light. They are usually denoted by R and T respectively, and can be calculated by the Fresnel equations [2]. If the incoming light is p-polarized the reflection coefficient is given by

$$R_p = \left(\frac{n_1 \cos \theta_t - n_2 \cos \theta_0}{n_1 \cos \theta_t + n_2 \cos \theta_0} \right)^2 \quad (1.7a)$$

and if the light is s-polarized it is given by

$$R_s = \left(\frac{n_1 \cos \theta_0 - n_2 \cos \theta_t}{n_1 \cos \theta_0 + n_2 \cos \theta_t} \right)^2. \quad (1.7b)$$

In both cases the transmission coefficient T is simply calculated by

$$T_\nu = 1 - R_\nu. \quad (1.7c)$$

In the equations above n_1 and n_2 are the refractive indices for media 1 and 2.

2.1 Overview of the system

The type of system we are working with in this project is presented in Fig. 2.1. There are three media in this system, each with its own relative permittivity which are labeled $\epsilon_{r,1}$, $\epsilon_{r,2}$ and $\epsilon_{r,3}$. The interface between media 1 and 2 is a flat surface, parallel to the x_1 -axis while the interface between media 2 and 3 is rough. The latter interface is described by the function $\zeta(x_1)$ so that the height of a point on the interface is given by $x_3 = \zeta(x_1)$. This function is called the *surface profile function*. Section 2.3 goes into more detail.

The upper interface is illuminated by a plane wave with the x_1x_3 -plane being its plane of incidence. Its wave vector is \mathbf{k} . The transmitted light has the wave vector \mathbf{q} . The x_1 -component of the wave vectors \mathbf{k} and \mathbf{q} are in the following referred to as k and q and they are

$$k = \frac{\omega}{c} \sqrt{\epsilon_1} \sin \theta_0, \quad (2.1)$$

$$q = \frac{\omega}{c} \sqrt{\epsilon_3} \sin \theta_t. \quad (2.2)$$

For the most part the system in Fig. 2.1 will approximate a window. Medium 1 will represent the outside of the windows since this is where the light source is located. This means that the inside of the window is rough and the outside is smooth.

Most windows are vertically oriented, meaning that Fig. 2.1 shows a rotated view. Note then, that a negative angle of incidence θ_0 corresponds to incident light from above, e.g. from the sun. Most of the simulations presented here will feature a negative angle of incidence.

We approximate air and glass by choosing $\epsilon_{r,1} = \epsilon_{r,3} = 1.01$ and $\epsilon_{r,2} = 2.31$. Furthermore we assume that all three media are completely non-magnetic, i.e. that $\mu_{r,1} = \mu_{r,2} = \mu_{r,3} = 1$.

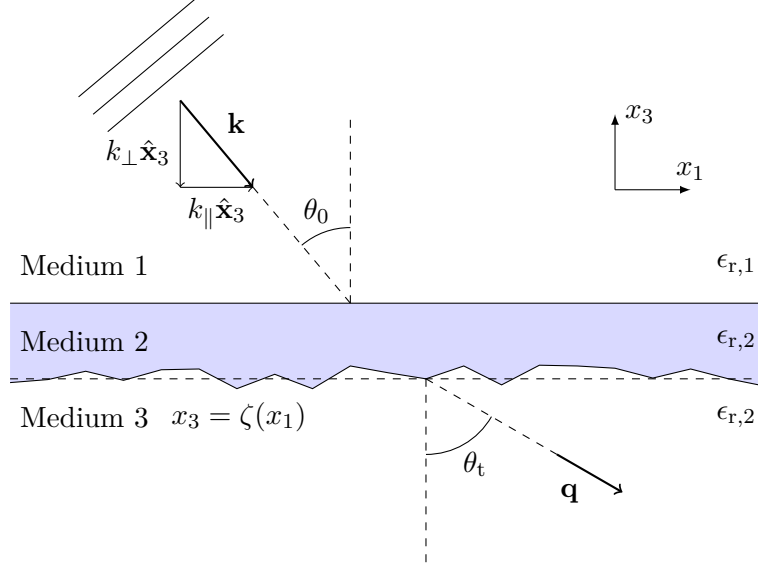


Figure 2.1: An overview of the system considered in this thesis. The incident plane is the x_1x_3 -plane. A flat surface is parallel to the x_1 -axis, forming the interface between media 1 and 2. Below it, a rough surface described by $x_3 = \zeta(x_1)$ forms the interface between media 2 and 3. The incoming wave vector is \mathbf{k} with components \mathbf{k}_\perp and \mathbf{k}_\parallel as indicated. The wave vector of the reflected light is \mathbf{q} and is decomposed similarly. The x_1 -component are commonly called k and q . The angle of transmission is measured relative to the x_1 -axis.

In the notation used above the relative permittivities have carried the “r” subscript for consistency with the notation used in Section 1.2. Going forward, the relative permittivities will appear often. For convenience they will from now on be referenced without the additional subscript, so that ϵ_i is the relative permittivity of medium i .

2.2 Mean Differential Transmission Coefficient

The goal is to design windows that will improve indoor lighting conditions. This means that the window must be able to transmit light unlike normal windows. Controlling how light is transmitted through an object essentially means controlling its transmission coefficient.

The Mean Differential Reflection and Transmission Coefficients (MDRC and MDTC) give the average intensity of the reflected and transmitted light, respectively, as a function of the angle of incidence and the angle of reflection/transmission. The averaging is done over an ensemble of statistically identical systems. These are the properties we want to control. The MDTC, is defined as [9]:

$$\left\langle \frac{\partial T_\nu}{\partial \theta_t} \right\rangle (q|k) = \frac{1}{L_1} \frac{\epsilon_3}{\sqrt{\epsilon_1}} \frac{\kappa_\nu}{\mu_\nu} \frac{\omega}{2\pi c} \frac{\cos^2 \theta_t}{\cos \theta_0} \langle |T_\nu(q|k)|^2 \rangle. \quad (2.3)$$

The parameters κ_ν and μ_ν depend on the polarization, as indicated by their subscripts. Later on a third factor, λ_ν , will also be present. For p-polarized light, $\nu = p$, we have $k_p = \epsilon_1$, $\lambda_p = \epsilon_2$ and $\mu_p = \epsilon_3$. When the incident light is s-polarised we have $k_s = \lambda_s = \mu_s = 1$.

The length of the system in the x_1 -direction appears as L_1 and $T_\nu(q|k)$ is the transmission coefficient of light whose wave vector's x_1 -component is k and transmitted light where the x_1 -component is q . The angle brackets $\langle \cdot \rangle$ denote averaging over an ensemble of surfaces.

By specifying the shape of the mean differential transmission coefficient we are in effect specifying the properties of the light that is transmitted. We can e.g. decide that the direction of the light should be shifted 10° upon passing through the system or that all the transmitted light should be normal to the window no matter the angle of incidence. This is specified by providing a function of the angle of transmission describing the desired shape of the MDTC.

If the system being investigated does not absorb energy, i.e., $\text{Im}\epsilon = 0$, then all the incident light is either reflected or transmitted. If this is the case then the relative reflection and the transmission energies should sum up to 1:

$$\int_{-\pi/2}^{\pi/2} \left\langle \frac{\partial R_\nu}{\partial \theta_s} \right\rangle d\theta_s + \int_{-\pi/2}^{\pi/2} \left\langle \frac{\partial T_\nu}{\partial \theta_t} \right\rangle d\theta_t = 1,$$

where θ_s is the angle of reflection and θ_t is the angle of transmission. This is used to check the validity of the simulations. The value obtained by evaluating the left-hand side will be referred to as the *unitarity* of the simulation. A value closer to 1 suggests more accurate results.

As can be seen in Fig. 2.1, θ_0 is the angle of incidence and θ_t is the angle the transmitted light forms with the normal of the surface. The surfaces discussed are constructed with a specific angle of incidence in mind which will be referred to as the incident angle of design and denoted $\theta_{0,d}$. That is, the surfaces are supposed to yield the wanted MDTC if $\theta_0 = \theta_{0,d}$. The last angle is the transmission angle of design, $\theta_{t,d}$. This is a parameter to the MDTCs. The interpretation of this angle depends on the specific MDTC.

One possible shape for the MDTC is

$$\left\langle \frac{\partial T_\nu}{\partial \theta_t} \right\rangle(q|k) = C \cos^n(\theta_t - \theta_{t,d}) \quad (2.4)$$

where $\theta_{t,d}$ is a chosen value and n is a positive integer. The transmission angle of design here specifies the angle of transmission at which the MDTC has a maximum. This function in fact corresponds to the drop shape seen in Fig. 1.4 when $n = 9$ and $\theta_t = 30^\circ$. This specific MDTC is one possible choice. Which MDTC is used pertains little to the overall objective of this thesis, but I will use this one with $n = 9$ in order to compare my results to that of Winjum [10].

The MDTC above will ideally redirect *all* of the light toward the ceiling thus rendering the window impossible to see through. An alternative is to diffuse a part of the

2 Randomly rough surfaces

light while the rest passes through as if the window were perfectly normal:

$$\left\langle \frac{\partial T_\nu}{\partial \theta_t} \right\rangle (q|k) = C_1 \cos^n(\theta_t - \theta_{t,d}) + C_2 \delta(\theta_t - \theta_{0,d}) \quad (2.5)$$

The function $\delta(\theta)$ is the Dirac delta function and contributes only when the angle of incidence equals the angle of transmission, mimicking a normal window. When these angles are equal the other term is very small. The factors C_1 and C_2 can be set to adjust how the total intensity is spread, for example 40 % of the light can be redirected and 60 % can pass straight through. This MDTC is not very interesting, however. It is probably easier to let only part of the proposed window have redirection properties. This also solves the issue of not being able to see through the window. This MDTC will therefore not be discussed further, but mentioned to illustrate the possibility.

Another MDTC which will be considered is

$$\left\langle \frac{\partial T_\nu}{\partial \theta_t} \right\rangle (q|k) = C_1 u(|\theta_d| - \theta_t). \quad (2.6)$$

where $u(\theta)$ is the Heaviside step function. This gives constant transmission from $\theta_t = -\theta_{t,d}$ to $\theta_t = \theta_{t,d}$ and zero transmission outside this interval.

Figure 2.2 shows the MDTCs displayed in Eqs. (2.4) and (2.6) for $\theta_{t,d} = 40^\circ$ and $\theta_{t,d} = 7^\circ$, respectively. The former MDTC looks very different in a plot like this compared to Fig. 1.4 which gives a more intuitive picture of the same function.

We still need to tie our choice of MDTC to the description of the window. This is done by the Kirchhoff approximation. Recall that Section 2.1 introduced the surface profile function $\zeta(x_1)$. Assuming that the rough surface of the window is described by that function, then by the Kirchhoff Approximation and the method of stationary phase [11]:

$$\begin{aligned} \langle |T_\nu(q|k)|^2 \rangle &= |T_\nu^{(0)}(q|k)|^2 \\ &\times \int_{-\infty}^{\infty} dx_1 \int_{-\infty}^{\infty} dx'_1 e^{-i(q-k)(x_1-x'_1)} \langle e^{i[\alpha_3(q)-\alpha_2(k)][\zeta(x_1)-\zeta(x'_1)]} \rangle. \end{aligned}$$

so that the MDRC becomes

$$\begin{aligned} \left\langle \frac{\partial T_\nu}{\partial \theta_t} \right\rangle (q|k) &= \frac{1}{L_1} \frac{\epsilon_3}{\sqrt{\epsilon_1}} \frac{\kappa_\nu}{\mu_\nu} \frac{\omega}{2\pi c} \frac{\cos^2 \theta_t}{\cos \theta_0} |T_\nu^{(0)}(q|k)|^2 \\ &\times \int_{-\infty}^{\infty} dx_1 \int_{-\infty}^{\infty} dx'_1 e^{-i(q-k)(x_1-x'_1)} \langle e^{i[\alpha_3(q)-\alpha_2(k)][\zeta(x_1)-\zeta(x'_1)]} \rangle. \end{aligned} \quad (2.7)$$

The definition of the factor $T_\nu^{(0)}(q|k)$ is found in Eq. (2.8.57) in *Designer Surfaces* [11]. Recall that the x_3 -component of the wave vectors are not necessarily real. In the equations above the quantities α_i are defined by

$$\alpha_i(k) = \left[\epsilon_i \left(\frac{\omega}{c} \right)^2 - k_{\parallel}^2 \right]^{\frac{1}{2}}, \quad \text{Re } \alpha_i(k) \geq 0, \quad \text{Im } \alpha_i(k) \geq 0, \quad (2.8)$$

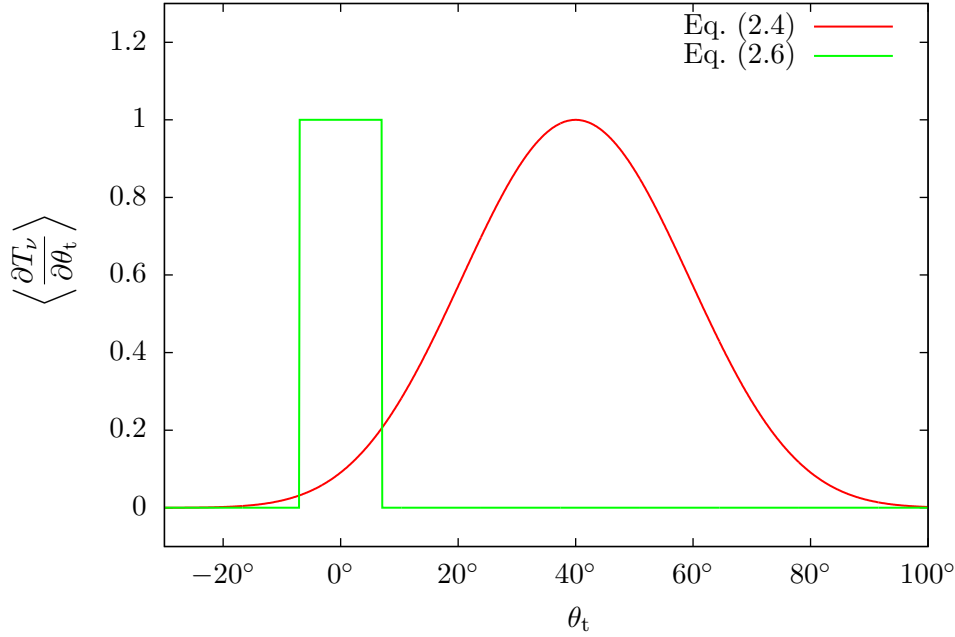


Figure 2.2: Mean differential transmission coefficient as functions of θ_t . The two MDTCs presented are $\langle \partial T_\nu / \partial \theta_t \rangle = \cos^9(\theta_t - 40^\circ)$ and $\langle \partial T_\nu / \partial \theta_t \rangle = u(7^\circ - |\theta_t|)$ corresponding to Eqs. (2.4) and (2.6) for $\theta_{0,d} = 40^\circ$ and $\theta_{0,d} = 7^\circ$ respectively.

and express the x_3 components of the wave vector in medium number i . Equation (2.9) is given as an integral over x_1 which makes it impossible to invert. Let $x'_1 = x_1 + u$ so that x_1 and x'_1 are two points separated by the distance u . The difference in value of the surface profile function can then be approximated by $\zeta(x_1) - \zeta(x'_1) \approx -u\zeta'(x_1)$. Inserting this gives [10, 11]

$$\begin{aligned} \left\langle \frac{\partial T_\nu}{\partial \theta_t} \right\rangle(q|k) &= \frac{1}{L_1} \frac{\epsilon_3}{\sqrt{\epsilon_1}} \frac{\kappa_\nu}{\mu_\nu} \frac{\omega}{2\pi c} \frac{\cos^2 \theta_t}{\cos \theta_0} |T_\nu^{(0)}(q|k)|^2 \\ &\times \int_{-\infty}^{\infty} dx_1 \int_{-\infty}^{\infty} du e^{-i(q-k)u} \left\langle e^{-i[\alpha_3(q) - \alpha_2(k)]u\zeta'(x_1)} \right\rangle. \end{aligned} \quad (2.9)$$

We now have the MDTC as a function of (among other things) the surface profile function.

2.3 The surfaces

There are many ways to construct randomly rough surfaces. One possible method is the method used in my project thesis [6] where the height of any point above the mean height is drawn from a Gaussian distribution and filtered through a desired power spectrum. Another way is constructing the surface as a series of trapezoidal grooves. This was done by Maradudin et al. [1] whose results will be compared with results achieved here.

2 Randomly rough surfaces

The surfaces we concern ourselves consists of several joined line segments connected so that the surfaces become continuous. Figure 2.3 illustrates such a surface. A line segment is called a *facet* whose length is d in the x_1 -direction. The length d will from here on be called the *facet size*. The surface profile function can then be written as

$$\zeta(x_1) = a_n x_1 + b_n, \quad dn < x_1 < d(n+1). \quad (2.10)$$

where a_n is the slope of line segment number n and b_n is that line segment's intercept with the x_3 -axis. The slopes a_n will be drawn from a probability distribution and the intercepts b_n are chosen afterwards so that the surface becomes continuous at the junctions between successive line segments. The probability function for the a_n is what we want to find.

As we saw in Section 2.2 the MDTC is dependent on the function describing the rough surfaces. For simplicity, let I be the product of the two integrals in Eq. (2.9). Inserting Eq. (2.10) gives

$$\begin{aligned} I &= \int_{-\infty}^{\infty} dx_1 \int_{-\infty}^{\infty} du e^{-i(q-k)u} \langle e^{-i[\alpha_3(q)-\alpha_2(k)]u\zeta'(x_1)} \rangle \\ &= \int_{-\infty}^{\infty} du e^{-i(q-k)u} \sum_{n=-N_s}^{N_s-1} \int_{dn}^{d(n+1)} dx_1 \langle e^{-i[\alpha_3(q)-\alpha_2(k)]ua_n} \rangle. \end{aligned}$$

where N_s is the number of facets the surface consists of. The slopes a_n are as mentioned drawn from a distribution which, we will call $f(\gamma)$. Replacing a_n with its distribution gives

$$I = \int_{-\infty}^{\infty} du e^{-i(q-k)u} \sum_{n=-N_s}^{N_s-1} \int_{dn}^{d(n+1)} dx_1 \int_{-\infty}^{\infty} d\gamma f(\gamma) e^{-i[\alpha_3(q)-\alpha_2(k)]u\gamma}. \quad (2.11)$$

Integration over x_1 is trivial and gives the length of the surface L_1 . Integration over u gives a delta function since u only occurs in exponential function whose argument is imaginary. We are then left with

$$\begin{aligned} I &= 2N_s d \int_{-\infty}^{\infty} d\gamma f(\gamma) 2\pi \delta(q-k - [\alpha_3(q) - \alpha_2(k)]\gamma) \\ &= \frac{4\pi L_1}{|\alpha_3(q) - \alpha_2(k)|} f\left(\frac{q-k}{\alpha_3(q) - \alpha_2(k)}\right). \end{aligned} \quad (2.12)$$

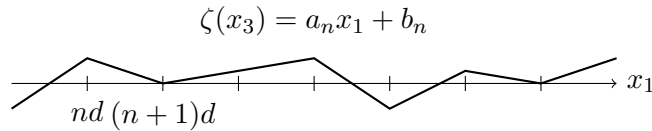


Figure 2.3: Illustration showing how the surfaces are constructed. The surfaces consist of a number of joined line segments, each of which has a slope a_n drawn from a probability distribution function and an x_3 -intercept b_n . The line segments in the x_1 -direction are all equal and denoted d .

Combining with result with Eq. (2.9) gives

$$\begin{aligned} \left\langle \frac{\partial T_\nu}{\partial \theta_t} \right\rangle (q|k) &= \frac{\epsilon_3}{\sqrt{\epsilon_1}} \frac{\kappa_\nu \cos^2 \theta_t}{\mu_\nu \cos \theta_{0,d}} \frac{|T_\nu^{(0)}(q|k)|^2}{\left| \sqrt{\epsilon_3} \cos \theta_t - \sqrt{\epsilon_2 - \epsilon_1 \sin^2 \theta_{0,d}} \right|} \\ &\times f \left(\frac{\sqrt{\epsilon_3} \sin \theta_t - \sqrt{\epsilon_1} \sin \theta_{0,d}}{\sqrt{\epsilon_3} \cos \theta_t - \sqrt{\epsilon_2 - \epsilon_1 \sin^2 \theta_{0,d}}} \right). \end{aligned} \quad (2.13)$$

A long and complicated derivation, done by Winjum [10], gives after inverting:

$$\begin{aligned} f(\gamma) &= \frac{(\lambda_\nu \sqrt{\epsilon_1} \cos \theta_{0,d} + \kappa_\nu r)^2}{\lambda_\nu \sqrt{\epsilon_1} \cos \theta_{0,d}} \frac{\mu_\nu}{\kappa_\nu} \left| \frac{1}{1 + \gamma^2} \left[\gamma(\gamma r - s) + \sqrt{(1 + \gamma^2)\epsilon_3 - (\gamma r - s)^2} \right] - r \right|^3 \\ &\times \frac{\left| \left(\epsilon_2 - \frac{\lambda_\nu}{\mu_\nu} \left(\frac{\lambda_\nu}{\mu_\nu} - 1 \right) \frac{1}{1 + \gamma^2} \left[\gamma(\gamma r - s) + \sqrt{(1 + \gamma^2)\epsilon_3 - (\gamma r - s)^2} \right] \right)^2 \right.}{\left. \left\{ \frac{1}{1 + \gamma^2} \left[\gamma(\gamma r - s) + \sqrt{(1 + \gamma^2)\epsilon_3 - (\gamma r - s)^2} \right] - \frac{1}{2}(\epsilon_2 + \epsilon_3) \right\} - \frac{1}{4}(\epsilon_2 - \epsilon_3)^2 \right|^2} \\ &\times \left\langle \frac{\partial T_\nu}{\partial \theta_t} \right\rangle (\gamma). \end{aligned} \quad (2.14)$$

The variable γ is defined by

$$\gamma = \frac{\sqrt{\epsilon_3} \sin \theta_t - \sqrt{\epsilon_1} \sin \theta_{0,d}}{\sqrt{\epsilon_3} \cos \theta_t - \sqrt{\epsilon_2 - \epsilon_1 \sin^2 \theta_{0,d}}}.$$

In addition, r and s are defined by

$$r = \sqrt{\epsilon_2 - \epsilon_1 \sin^2 \theta_{0,d}}$$

and

$$s = \sqrt{\epsilon_1} \sin \theta_{0,d}.$$

In the end we have an expression for the probability distribution of the slopes, partly defined by the MDTC. We can then “plug in” our desired MDTC to give the needed probability density function, after writing the MDTC in terms of γ instead of θ_t . For example, Eq. (2.4) becomes

$$\begin{aligned} \left\langle \frac{\partial T_\nu}{\partial \theta_t} \right\rangle (q|k) &= C \left(\frac{\cos \theta_{t,d}}{\sqrt{\epsilon_1}(1 + \gamma^2)} \left[\gamma(\gamma r - s) + \sqrt{(1 + \gamma^2)\epsilon_3 - (\gamma r - s)^2} \right] \right. \\ &\left. + \frac{\sin \theta_{t,d}}{\sqrt{\epsilon_1}(1 + \gamma^2)} \left[-(\gamma r - s) + \gamma \sqrt{(1 + \gamma^2)\epsilon_3 - (\gamma r - s)^2} \right] \right)^9, \end{aligned}$$

since [10]

$$\begin{aligned} \cos \theta_t &= \frac{1}{\sqrt{\epsilon_3}(1 + \gamma^2)} \left[\gamma(\gamma r - s) + \sqrt{(1 + \gamma^2)\epsilon_3 - (\gamma r - s)^2} \right], \\ \sin \theta_t &= \frac{1}{\sqrt{\epsilon_3}(1 + \gamma^2)} \left[-(\gamma r - s) + \gamma \sqrt{(1 + \gamma^2)\epsilon_3 - (\gamma r - s)^2} \right] \end{aligned}$$

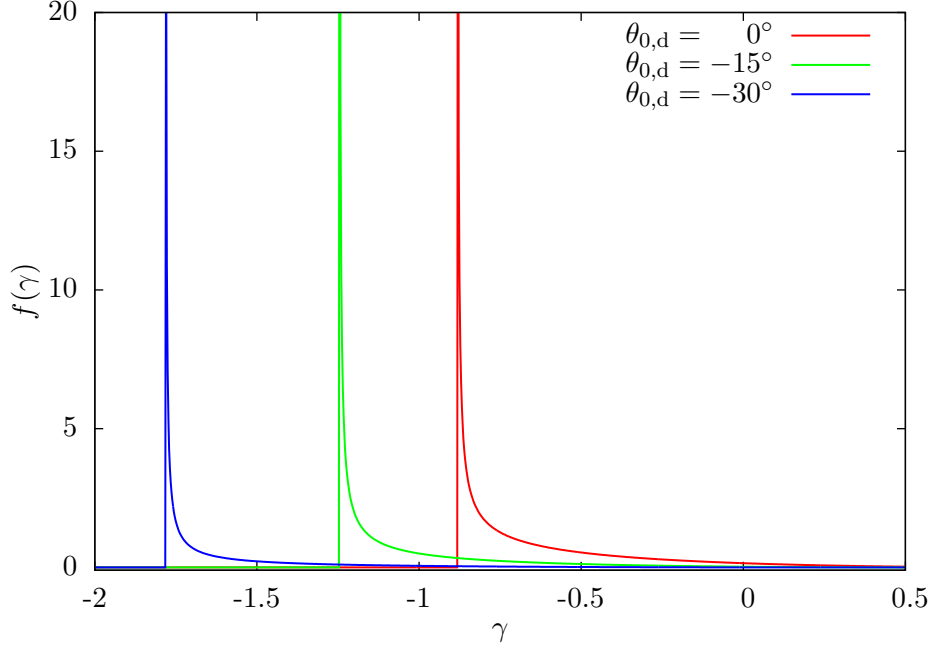


Figure 2.4: The probability distribution $f(\gamma)$ shown in Equation (2.14) plotted for the most studied system in this thesis characterized by $\epsilon_1 = \epsilon_3 = 1.01$ and $\epsilon_2 = 2.31$. Furthermore the polarization was p, the angle of design $\theta_{t,d} = 30^\circ$ and $\theta_{0,d}$ as indicated in the legend.

The constants in Eq. (2.4) and Eq. (2.6) have been arbitrarily chosen to be 1 throughout. It is important to note that because of this the expression for $f(\gamma)$ in Eq. (2.14) is generally *not* normalized. Because the random numbers used are selected by the rejection method (see Section 2.4) this does not matter. As long as the auxiliary function's area and domain is larger, the rejection method will work. In all the plots shown here, a normalized $f(\gamma)$ is displayed, i.e., what is plotted is actually $f(\gamma)/A$ where A is the area under $f(\gamma)$.

How the probability distribution function $f(\gamma)$ appears depends on which MDTC is used. Figure 2.4 shows Eq. (2.14) for the system we will study the most in this thesis. Media 1 and 3 are air, and medium 2 is glass, i.e. $\epsilon_1 = \epsilon_3 = 1.01$ and $\epsilon_2 = 2.31$. The angle of incidence $\theta_{0,d}$ is varied from 0° to 30° . The MDTC used was Eq. (2.4) with the parameter $\theta_{t,d}$ set to 0° .

There is a very noticeable shift in $f(\gamma)$ when $\theta_{0,d}$ is changed. When $\theta_{t,d}$ is held constant and $\theta_{0,d}$ is changed, however, there are very minute differences. In Fig. 2.5 $f(\gamma)$ is plotted for $\theta_{t,d} = 15^\circ$ and $\theta_{t,d} = 30^\circ$, along with the difference between them.

The difference itself appears to be very similar to $f(\gamma)$ except of course for the region where it is negative. It will be interesting to see if such a small difference can actually affect the transmission properties of the surface to the degree $f(\gamma)$ it is supposed to.

Figure 2.6 shows the same probability distribution function, but where Eq. (2.6) is used as the MDTC. The relative permittivities are the same and $\theta_{0,d} = 0^\circ$. Un-

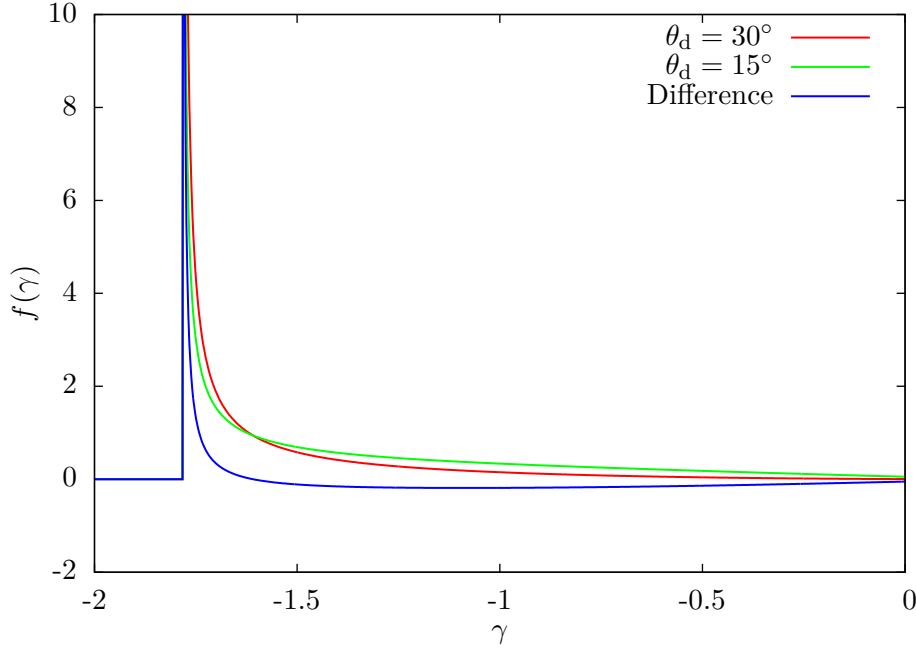


Figure 2.5: The probability distribution $f(\gamma)$ shown in Equation (2.14) when $\theta_{0,d} = 15^\circ$ and $\theta_{0,d} = 30^\circ$. In addition the difference is plotted. The two distributions are very similar, and the difference between them resembles them as well.

surprisingly the probability density function (pdf) for the segment slopes is completely symmetrical. This is reasonable since the actual MDTC the surface should produce is symmetrical. Furthermore the pdf is centered on $\gamma = 0$. This is also understandable since the desired MDTC is centered on the normal to the surface, i.e. it is centered on $\theta_t = 0^\circ$. The pdf also has the general shape of the desired MDTC itself, but with a slight dip on the top.

Drawing random numbers from $f(\gamma)$ yields the slopes of the line segments. If the mean of the distribution is zero then we expect the constructed interface to be flat on average. Figure 2.7 shows a system when the MDTC shown in Eq. (2.6) was used to draw 20 such numbers. The lower line in the figure is the interface produced by connecting 20 line segments (Eq. (2.10)) and has been shifted downwards by five micrometres. The upper lines is the flat interface, as described in Section 2.1.

The probability distributions shown in Fig. 2.4 clearly have a mean value unequal to zero. In this case the mean slope of the generated interface must also be unequal to zero. Figure 2.8 is similar to Fig. 2.7 but where the MDTC has been specified to be Eq. (2.4) with $\theta_{t,d} = 15^\circ$. This gives a mean slope to the lower interface. The lower interface has been offset by $20\ \mu\text{m}$ downwards. Eq. (2.10) itself has no facilities for adjusting the mean value of the surface profile function $\zeta(x_1)$ directly. The surfaces are therefore offset manually.

2 Randomly rough surfaces

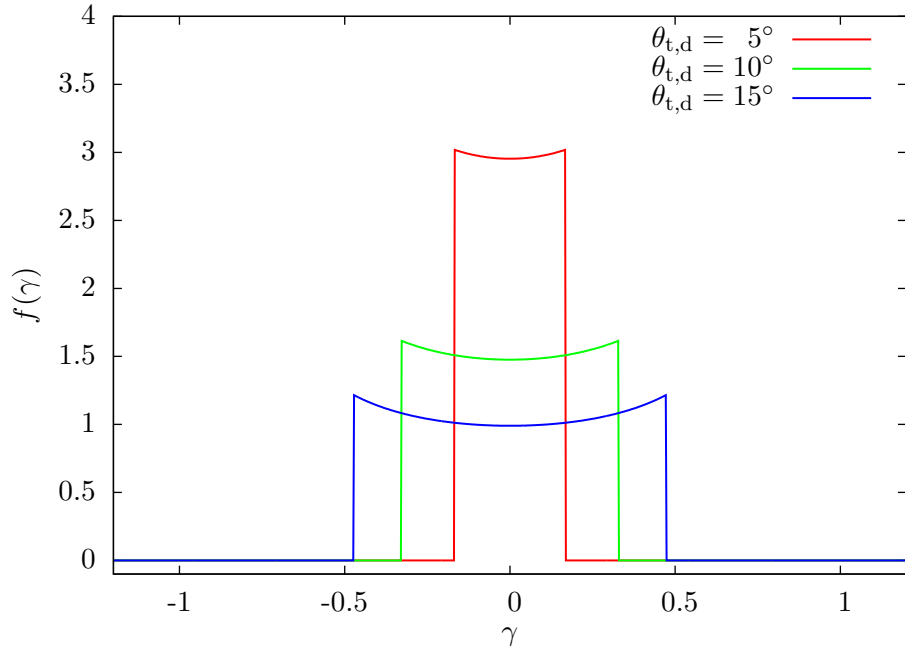


Figure 2.6: The probability distribution $f(\gamma)$ shown in Equation (2.14) plotted with the following parameters: $\epsilon_1 = \epsilon_3 = 1.01, \epsilon_2 = 2.31, \nu = p, \theta_{0,d} = 0^\circ$ and $\theta_{t,d}$ as indicated. Note that all three distributions are centered on $\gamma = 0$ and are symmetric.

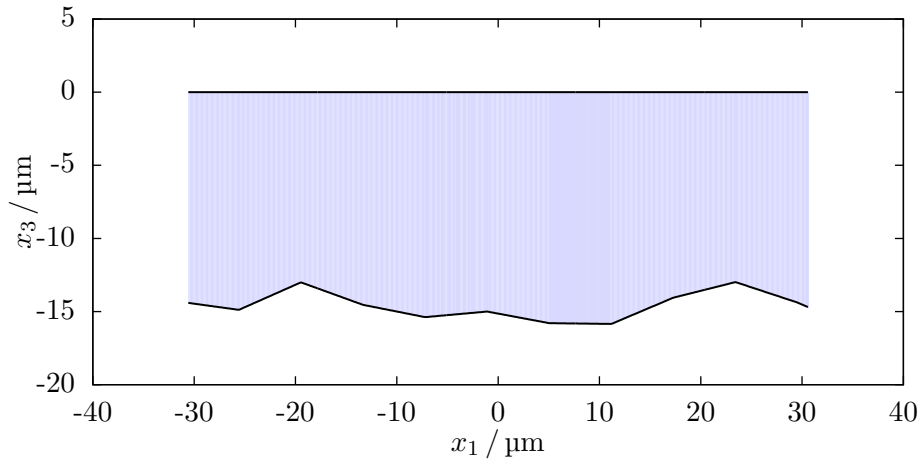


Figure 2.7: This is one realization of a system generated by the simulation software. The upper interface is completely flat. The lower interface is randomly generated by the method outlined in Section 2.2, using Eqs. (2.6) and (2.14). The lower surface has been offset downwards by $5 \mu\text{m}$. It consists of 20 facets, each 10λ long, and is designed to transmit normally incident light evenly between $\theta_t = -7^\circ$ and $\theta_t = 7^\circ$.

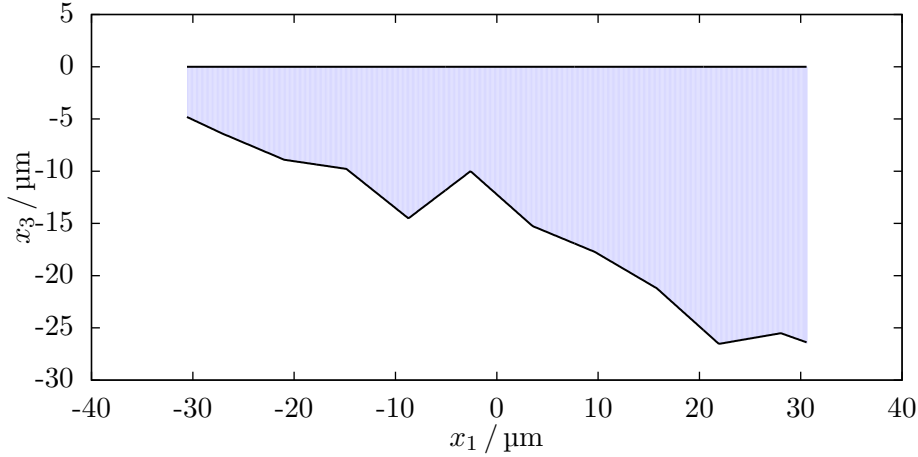


Figure 2.8: Similar to Fig. 2.7 but where the MDTC shown in Eq. (2.4) has been used. The angles used are $\theta_{0,d} = 0^\circ$ and $\theta_{t,d} = 15^\circ$. The lower surface has been offset downwards by $20 \mu\text{m}$.

2.4 The rejection method

The *rejection method* is one of several algorithms used to draw random numbers that follow from a specified probability distribution. This method is the one employed by the simulation software used in this thesis. In the following $p(x)$ denotes a probability distribution so that $p(x) dx$ is the probability of x being between x and $x + dx$.

The advantage of the rejection method is that we do not need to know the cumulative distribution function accurately, nor its inverse. This allows for sampling of probability distribution functions that have very complicated shapes. To provide this advantage the rejection method uses an *auxiliary* function $g(x)$ which is a simpler function whose cumulative inverse we *do* know. The function $g(x)$ should be defined and larger than $p(x)$ everywhere $p(x)$ is defined. The ratio $g(x)/p(x)$ is then necessarily always larger than one. Let the ratio's largest value be called c , that is, $c = \max(g(x)/p(x))$.

The algorithm for drawing a random number from the probability density function $p(x)$ is as follows: Generate two random numbers. One number is from $U(0, 1)$, which is the uniform distribution between zero and one. Call that number u . The other number is drawn from $g(x)$ by any method and is called x' . If $uc \frac{g(x')}{p(x')} \leq 1$ then accept x' as a random number drawn from $p(x)$. If not, draw new numbers u and x' and repeat the above. The method is summarized in Algorithm 1.

An interpretation of the rejection method is that we randomly draw a point in the area under $g(x)$. If that point happens to be within the area of $p(x)$ as well we accept the x -coordinate as correctly drawn from $p(x)$. If not, we draw a new number under $g(x)$.

It is easier to understand the interpretation above if we choose $g(x)$ as a constant, e.g. $1.1 \times \max(p(x))$. Drawing a number from $g(x)$ will be the same as drawing a number from $U(a, b)$ where a and b are the ends of g 's domain. The ratio between the

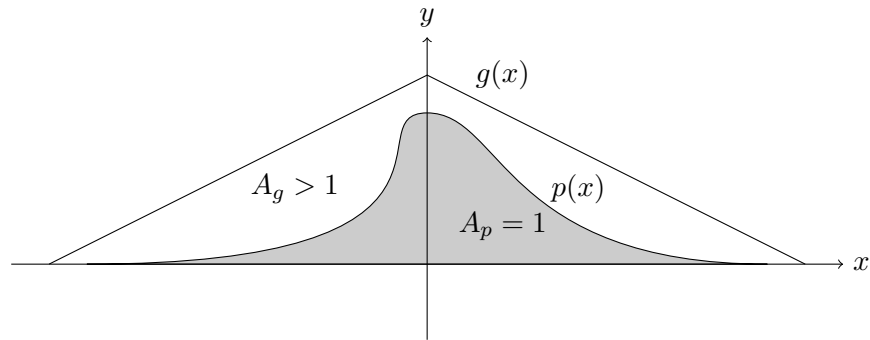


Figure 2.9: An overview of the rejection method. The function $p(x)$ is a probability distribution whose integral is difficult to describe analytically. The function $g(x)$ is a simpler function whose domain is a superset of that of $p(x)$ and is larger than $p(x)$ everywhere $p(x)$ is defined. This means that the area under $g(x)$ is larger than that under $p(x)$.

Algorithm 1: The rejection method

Input: None

Output: A random number x drawn from $p(x)$

repeat

$u = U \sim U(0, 1);$

$x' = X' \sim g(x);$

until $uc \frac{g(x')}{p(x')} \leq 1 ;$

$x \leftarrow x';$

distributions' maximum values c will be 1.1. The test $ucg(x')/p(x') \leq 1$ becomes

$$uc \frac{g}{p(x')} \leq 1,$$

$$uc \leq \frac{p(x')}{g}.$$

Understandably, the efficiency of this method is highly dependent on the choice of the auxiliary function $g(x)$. The closer to $p(x)$ the auxiliary function is, the fewer numbers have to be rejected on average before successfully drawing a random number.

Interestingly, the area under the actual probability density function does not have to be equal to 1, either. Whether a number is accepted or not depends on the ratio between the probability distribution function and the auxiliary function, not on the areas under them.

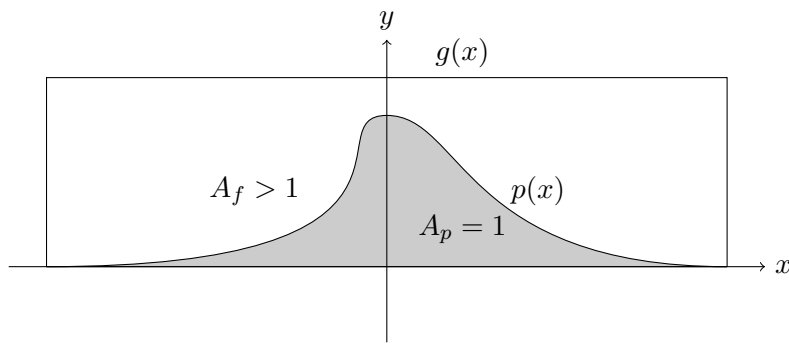


Figure 2.10: Similar to Fig. 2.9 but here $g(x)$ is constant and slightly larger than the largest value of $p(x)$.

3.1 Preliminary work

In order to familiarize myself with the probability density function Eq. (2.14) and the rejection method (Section 2.4) I wrote a small program in Python to plot this function and some MDRCs. This program also uses the rejection method to draw random numbers. It is called `rejection.py` and is available in the accompanying zip archive.

Winjum [10] gives separate expressions for the general $f(\gamma)$, i.e., Eq. (2.14) and an equivalent distribution for the specific case where the angle of incidence θ_0 is equal to zero. The program `rejection.py` includes both these expressions and I have confirmed that Eq. (2.14) overlaps with the special case when $\theta_0 = 0^\circ$.

The plot in Fig. 3.1 shows $f(\gamma)$ for the angles $\theta_{0,d} = -30^\circ$ and $\theta_{t,d} = 30^\circ$. The auxiliary function used was $g(\gamma) \equiv 1.1 \max f(\gamma)$. This is a very inefficient choice for the auxiliary function given the shape of $f(\gamma)$. But drawing these 1 000 000 numbers took approximately 30 s which is dwarfed by the time the actual simulations spend.

The tool `Maxwell11D` to perform the simulations. This is a piece of software written in Fortran 90 by my supervisor Ingve Simonsen. It takes a text file as input consisting of various parameters and generates a `NetCDF` data file containing the simulation results.

The distribution given by Eqs. (2.4) and (2.14) were not initially supported by `Maxwell11D`. After receiving the source code from my supervisor I added the support for this function and the wanted MDRCs to the source code and recompiled the program. The Python program I wrote was of great help in checking that everything was working correctly since it was much easier to handle and debug. After confirming that the Python program produced the correct results, the mathematical portion of the code was easily adjusted and incorporated into the Fortran 90 source code.

I wrote several small utility scripts in Bash and Python to aid in batch generation of configuration files, running `Maxwell11D` in parallel and producing flat files for plotting

3 Results

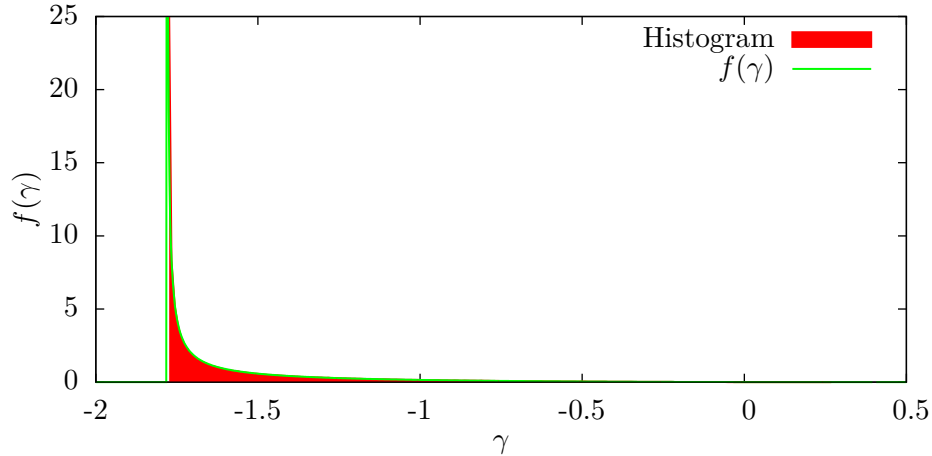


Figure 3.1: The green curve is the probability distribution $f(\gamma)$ given in Eq. (2.14) with the MDTC given by Eq. (2.4) for the following parameters: $\epsilon_1 = \epsilon_3 = 1.01, \epsilon_2 = 2.31, \nu = p, \theta_{0,d} = -30^\circ$ and $\theta_{0,d} = 30^\circ$. The histogram was generated from 1 000 000 randomly drawn numbers by the rejection method and is shown in red.

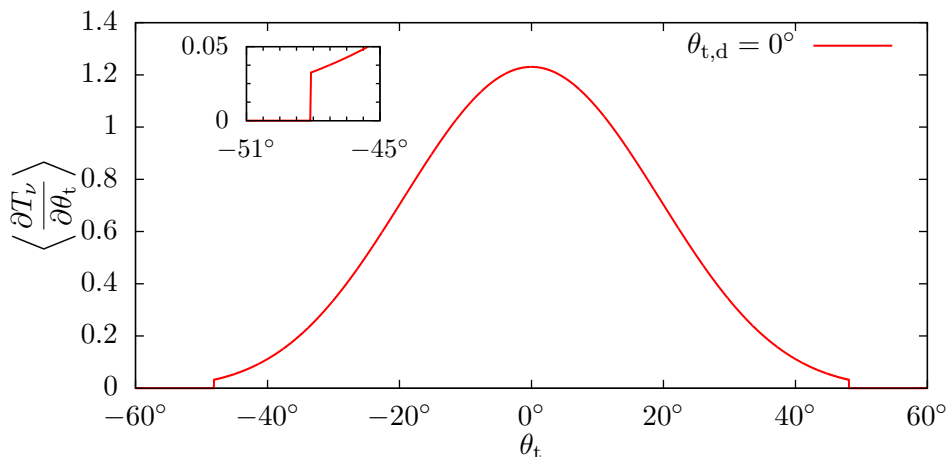


Figure 3.2: The Kirchhoff approximation shown in Eq. (A.2) plotted with the following parameters: $\epsilon_1 = \epsilon_3 = 1.01, \epsilon_2 = 2.31$ and $\theta_{0,d} = \theta_{t,d} = 0^\circ$. The MDTC used was Eq. (2.4). Notice the cutoffs at each end of the curves. The inset shows a close-up view of the MDTC dropping to zero on the left. The cutoff on the right side is entirely equivalent to the one on the left.

via `Maxwell11D_Plot`. These are also available in the accompanying zip archive.

The Kirchhoff approximation can give an expectation of the results of a simulation. Its full expression is given in Eq. (A.2). Figure 3.2 shows the approximation for most common system in this thesis, where $\epsilon_1 = \epsilon_3 = 1.01$ and $\epsilon_2 = 2.31$. The approximation is the same for both polarizations.

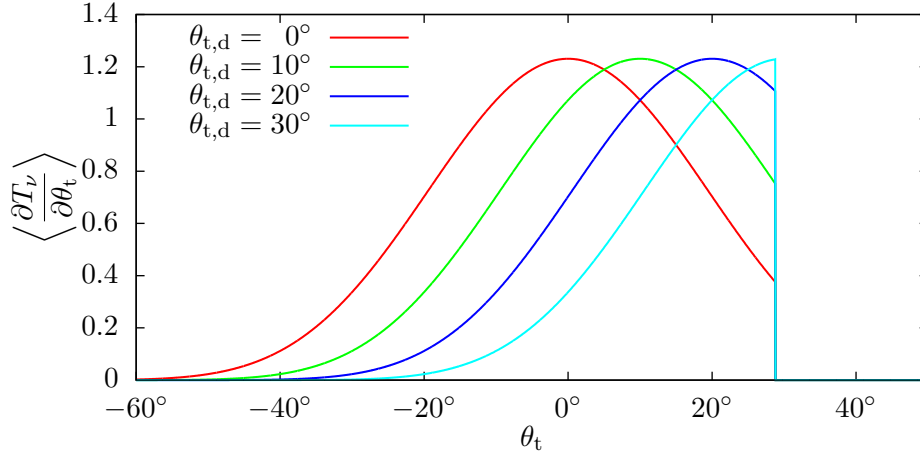


Figure 3.3: The Kirchhoff approximation plotted for the same parameters as in Fig. 3.2 but for different angles of design $\theta_{t,d}$. The cutoffs remain the same for all angles. The angle of incidence $\theta_{0,d}$ used here is -30° . Note also that the upper cutoff has shifted to the left compared to the upper cutoff in Fig. 3.2. This is because the incident angle of design was reduced to -30° .

The cutoffs depend on the angle of incidence and is symmetrically placed around it. For all angles of incidence the cutoffs are approximately $\theta_{0,d} \pm 48.34^\circ$. Figure 3.3 shows the Kirchhoff approximation for various $\theta_{t,d}$. The cutoffs remain the same and for high $\theta_{t,d}$ the resulting approximation becomes quite different from a $\cos^9 \theta$ function.

Since the Kirchhoff approximation is a rather crude approximation I don't expect to see such sharp cutoffs in practice.

3.2 Simulation overview

It is computationally easier to calculate the average scattering of many small surfaces compared to fewer but larger surfaces. The number of surfaces used affects the results greatly. In the following I have mostly used either 5000 or 10 000 surfaces, all of which are statistically identical to each other within one simulation. Each surface was for the most part 5 facets long. The facet size d mentioned in Section 2.3 has mostly been chosen to be 10λ . Some simulations were run with $d = 20\lambda$.

The surfaces are discretized. It would be tempting to suggest just a few points per facet since they are perfectly straight lines. But to capture the physics properly the resolution of the surface discretization must be significantly less than one wavelength. In most of the simulations performed the horizontal length between two adjacent discretization points was $\lambda/5$. Some simulations were run with a discretization of $\lambda/10$ but this significantly increased the time spent on one simulations while only yielding negligible improvements in accuracy.

Some simulations were also run with an iterative solver (BCGStab [12].) For the

simulations run in this thesis that method was very volatile as to whether it converged or not. While it did vastly improve the average speed of the simulations, just a slight shift in a parameter might hinder convergence. Based on how long most simulations typically took the iterative solver remained mostly unused.

Finally, the number of simulations actually run is much higher than what will be presented in the following sections. Many simulations were run with errors, or as duplicates of other simulations with a different seed number to the random number generator the program uses.

The results are presented as plots of $\langle \partial T_\nu / \partial \theta_t \rangle$ as a function of θ_t . The values of other parameters accompany each plot. Also indicated is the approximation in the geometrical optical limit of the Kirchhoff approximation the design of each surface was based on.

3.3 Results

3.3.1 Comparison with the results of Maradudin et al. [1]

In 2001 Maradudin et al. [1] published an article showing the results of a simulation where surfaces were designed to transmit light uniformly within a certain range of the angle of transmission. The MDTC used was Eq. (2.6) where $\theta_{t,d} = 7^\circ$. In that article the surfaces were generated by another method than the one described in Section 2.3. Instead of a series of connected line segments the surfaces were a series of trapezoidal grooves. The grooves were equal in height and characterized by the slope of their walls. Consequently the expression for $f(\gamma)$ is very different. The theory behind the derivation of $f(\gamma)$, however, remains the same.

Figure 3.4 shows results obtained by Maradudin et al. [1] for a system where the lower interface (i.e. between media 2 and 3) was generated in the manner outlined above. The system is different from what we have been discussing in other features, as well. The upper medium was characterized by a relative permittivity of 2.25. The second medium was a photoresist where $\epsilon_2 = 2.69$ while the last medium was vacuum, i.e. $\epsilon_3 = 1$. This system was designed to transmit normally incident light uniformly between $\theta_t = -7^\circ$ and $\theta_t = 7^\circ$ and achieved this very well. The calculated MDTC closely follows the approximation (labelled “Geo. Optics” in the figure.)

A simulation was run which was virtually identical to that shown in Fig. 3.4 except the surfaces were generated by the method outlined in Section 2.3 with a facet size of 10λ . The results are shown in Fig. 3.5 and are very similar. This is interesting since, as mentioned, the surfaces in these two simulations are generated using different methods. Since both methods use statistical representations that are derived from the wanted MDTC, ideally the results should be exactly the same. That the results achieved here agree to this extent with Maradudin et al. [1] suggests that the implementation of Eqs. (2.6) and (2.14) into `Maxwell1D` works properly.

Recall that the unitarity can be used a measure of the accuracy of the numerical simulations. The unitarity achieved in this simulation was 0.999, which is very good.

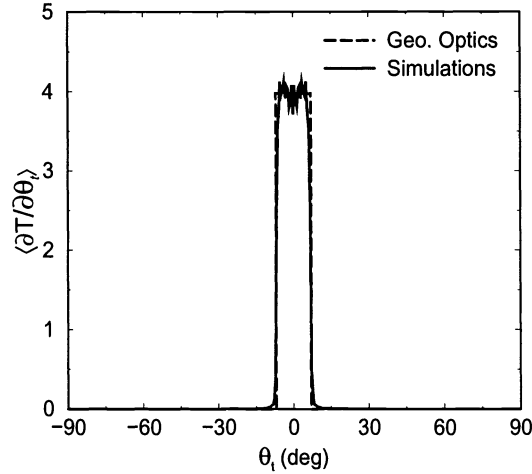


Figure 3.4: Results obtained by Maradudin et al. [1]. The plot shows the measured MDTC as a function of θ_t . An ensemble of 2972 surfaces were used. The incident light was s-polarized and its wavelength was $\lambda = 612.7$ nm. The system was characterized by $\epsilon_1 = 2.25$, $\epsilon_2 = 2.69$ and $\epsilon_3 = 1$. The number of sampling points was $N = 1000$, ($\delta x \approx \lambda/10$), The mean thickness of the film was $H = 5$ μm . $\theta_{t,d} = 7^\circ$. The length of each surfaces was $L = 60.6$ $\mu\text{m} = 98.9\lambda$.

3.3.2 Non-normal incidence

The simulation above—and several other simulation presented here—use surfaces that were designed for a specific angle of incidence and used light that was incident at that specific angle. Out of curiosity a simulation was run to see what would happen if a surface were designed as above, but that light incident on it from angles that deviate from the incident angle of design $\theta_{t,d}$. Figure 3.6 shows the results of a simulation where light was incident on the surface at $\theta_0 = \pm 10^\circ$ while at the same time $\theta_{0,d} = 0^\circ$.

First of all, the symmetry in the deviance is not surprising. Since the surface was created to symmetrically spread light around $\theta = 0^\circ$ when $\theta_0 = 0^\circ$ the sign in the deviance in the angle of incidence should not influence the result.

The unitarity achieved in these simulations were 0.976 and 0.979 for $\theta_0 = -10^\circ$ and 10° respectively. These are slightly lower than the unitarity achieved in the previous simulation, but still more than acceptable.

As expected the surface does not transmit the incident light as well as it does when the light was incident normal to the surface. The midpoints of the MDTCs are moved slightly farther away from zero than the actual θ_0 . The MDTCs are however still characterized by sharp edges on either side of their approximate axes of symmetry, although not as sharp. Also, the widths of the MDTCs are approximately 20° which they ideally should be.

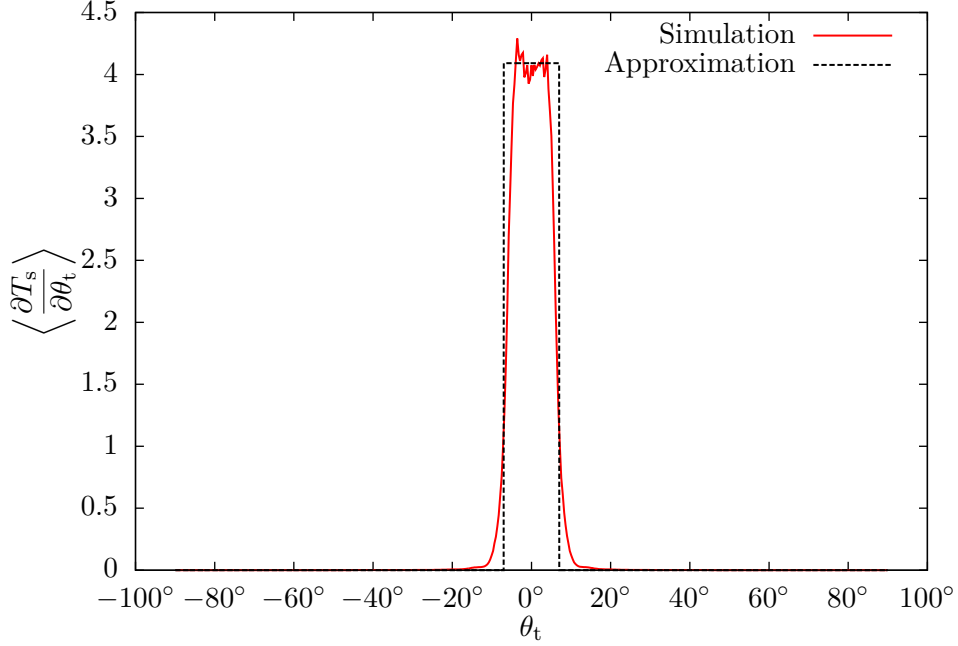


Figure 3.5: The results of a simulation of almost the exact same system as used by Maradudin et al. [1] which is shown in Fig. 3.4. The relative permittivities, mean thickness and sampling interval are all the same. The length of the system was approximately the same; here it was 10 facets, i.e. 100λ . In this simulation the surface was characterized by a facet size of 10λ . The Kirchhoff approximation of the MDTC is shown alongside.

3.3.3 Results from initial window simulations.

The first simulations performed using Eq. (2.4) as the desired MDTC all finished without errors but the results were very bad. Some of the unitarities achieved were as low as 0.15.

Normally, the simulation software can offset the surfaces by a specified amount. For example, the surfaces used in Section 3.3.1 had a vertical offset of $-5\ \mu\text{m}$ to achieve the mean thickness of the second medium in the system. The simulation software implements this by adding the specified offset to each coordinate in the discretization of the surface without checking if the two interfaces would cross each other, potentially creating an unphysical system. The surfaces produced using Eq. (2.4) are very slanted and this leads to the two interfaces crossing. After correcting this by specifying a much higher offset the results quickly became much better.

Figure 3.7 shows the results obtained for 5000 surfaces each 10 facets long. The facet size was 10λ and the MDTC used was Eq. (2.4). The sampling interval was $\lambda/10$. In contrast to the simulations in Section 3.3.1 the system here consisted of glass with air on both sides, meaning that $\epsilon_1 = \epsilon_3 = 1.01$ and $\epsilon_2 = 2.31$. Furthermore, all the angles $\theta_0 = \theta_{0,d} = \theta_{t,d} = 0^\circ$. The unitarity achieved here was 0.997 which is very good. The shape of the MDTC very much resembles a $\cos^9\theta$ function, although it appears more

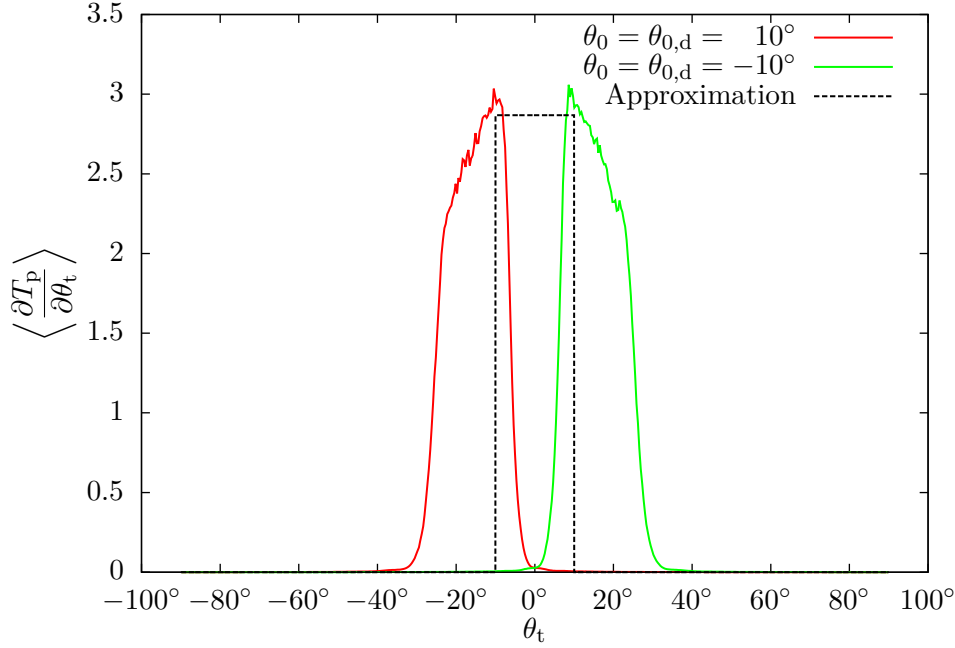


Figure 3.6: The results of two simulations almost identical to the simulation shown in Fig. 3.5. The length of the system, the relative permittivities, mean thickness and sampling interval are the same. In this simulation the surface was characterized by a facet size of 10λ and the number of surfaces used was 5000. The angles of incidence were *not* 0° but rather $\pm 10^\circ$ as indicated in the legend.

jagged. Overall the result is very good.

The results where $\theta_{t,d} \neq 0^\circ$ are unfortunately not as good. Figure 3.8 shows the same situation as above but $\theta_{t,d} = 15^\circ$ and θ_0 varying between 0° and 30° as indicated in the legend of the plot. The unitarities achieved here were 0.95 and 0.64 for $\theta_0 = \theta_{0,d} = 0^\circ$ and $\theta_0 = \theta_{0,d} = -15^\circ$, respectively. The former result is ample but when θ_0 is reduced the results worsen quickly.

3.3.4 Dependency on wavelength and polarization

The ultimate goal is to have a fully functional window that can direct sunlight. The electromagnetic radiation the sun emits does not consist of light of only one wavelength. It is rather a whole spectrum and receives contributions from all wavelengths that give visible light. Furthermore the intensity of the visible light is not uniform.

The He-Ne laser is commonly used in experiments. This laser emits light with a wavelength of 612.7 nm. In Section 3.3.1 the wavelength emitted by the light source in the simulations is the same. By choosing the wavelength used in simulations to a value that is readily realizable it becomes much easier to compare results between numerical simulation and physical experiments. For simplicity I continued using this wavelength

3 Results

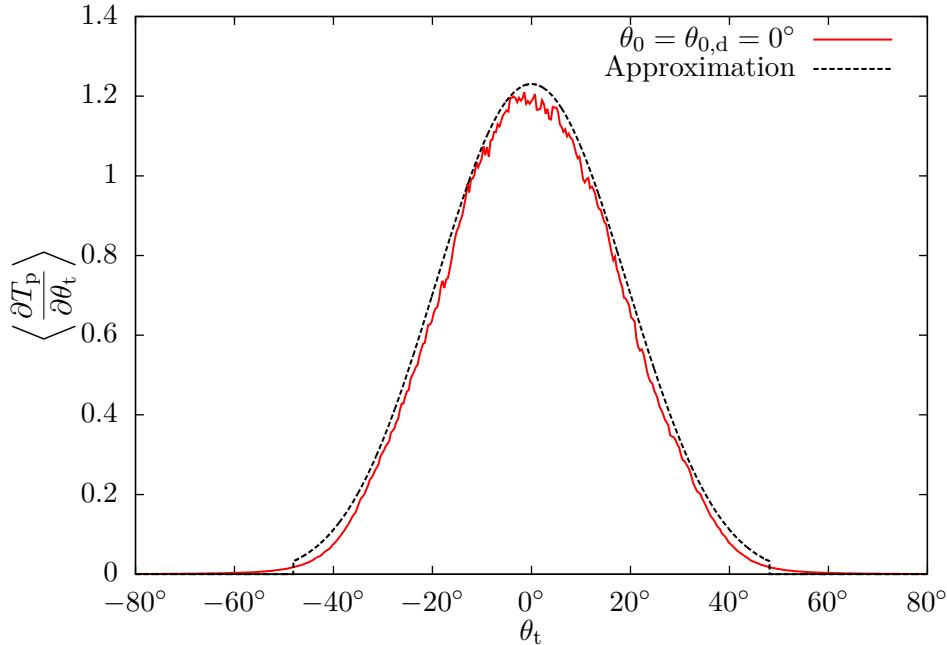


Figure 3.7: Results obtained for $N = 5000$ surfaces. Each surface was 10 facets long. The facet size was 10λ and the MDTC used was Eq. (2.4). The sampling interval was $\lambda/10$. In contrast to the simulations in Section 3.3.1 the system here consisted of glass with air on both sides, meaning that $\epsilon_1 = \epsilon_3 = 1.01$ and $\epsilon_2 = 2.31$. Furthermore, $\theta_0 = \theta_{0,d} = \theta_{t,d} = 0^\circ$. The unitarity was 0.997.

in further simulations. This is very important. If the window did react differently depending on wavelength, the lighting indoors would be intrusive. Splitting the light in a manner reminiscent of a prism be disruptive.

For a window with the properties pursued here to function as intended, the effect of changing the wavelength should ideally be zero. A small effect, however, is of little importance. Because the facet size is much larger than the wavelength ($d \gg \lambda$) we expect changing the wavelength to have very little effect. This is because diffraction would be negligible. To investigate this I ran simulations for various wavelengths evenly sampled from the visible electromagnetic spectrum. Figure 3.10 shows the results from a simulation with the modest angles $\theta_0 = \theta_{0,d} = -15^\circ$ and $\theta_{t,d} = 15^\circ$. Although the results were not very good in terms of unitarity, they are all very similar to each other, suggesting that the window can perform as intended in all situations.

The polarization used should also have very little effect. Simulations were run here the incident light was polarized differently than what was expected by the system. This showed no significant difference compared with the other simulations.

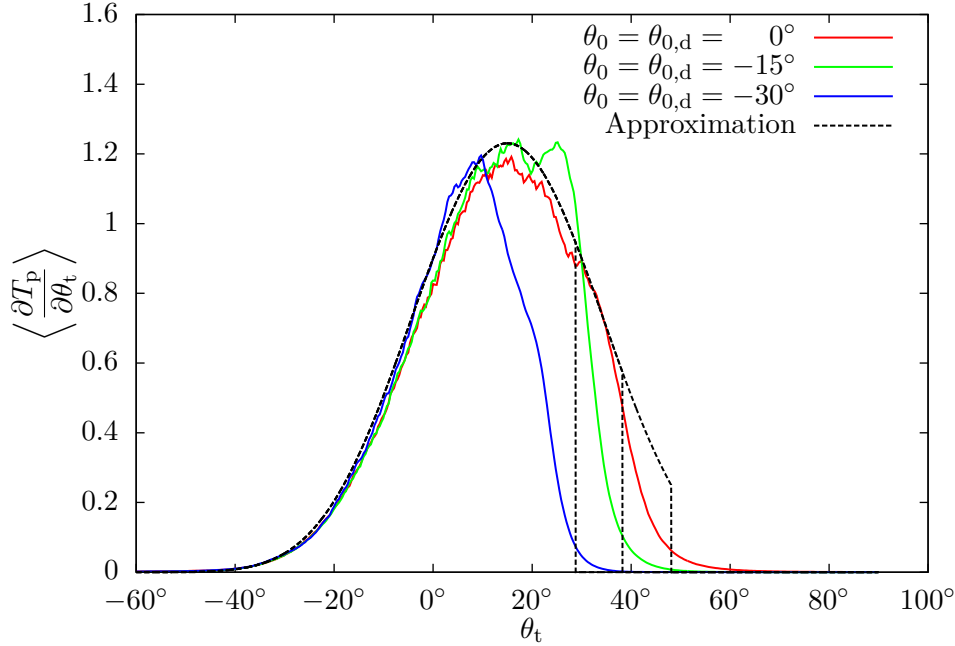


Figure 3.8: Results obtained for the same system as in Fig. 3.7 where $\theta_{t,d} = 15^\circ$ but for $\theta_0 = \theta_{0,d}$ as indicated in the legend. When $\theta_0 = \theta_{0,d} = 0^\circ$ the results are respectable with a nice but jagged peak centered on $\theta_t = 15^\circ$. The unitarity achieved was 0.93. When θ_0 is reduced to -15° the results are worse and the unitarity reduces to 0.88. The last result where $\theta_0 = -30^\circ$ the unitarity drops to 0.67.

3.3.5 Taking a shortcut

Most of the simulations presented so far have yielded unitarities that are significantly lower than one. This is not necessarily detrimental. One possible explanation for the lower unitarities is that some of the light that is reflected by the second interface escapes out through the sides of the simulated system, and is therefore not “counted”.

Consider for example Fig. 2.8 in the sense that the extent of the system in the x_3 -direction is significant compared to the extent in the x_1 -direction. The system ends up being very tall. Since the second interface is slanted, some of the light reflected by that interface will reach one of the ends of the system before reaching the first interface. That light will therefore not contribute in the calculation of the reflection coefficient. This will reduce the unitarity.

The first interface of the system discussed is flat. Our shortcut will therefore be simply to use Snell’s law to calculate the effective angle of incidence on the second interfaces. In Fig. 3.11 the intermediate angles ϕ_0 and ϕ_t are shown.

The angle ϕ_t is the angle of transmission produced by the flat interface and the angle ϕ_0 is the angle of incidence “observed” by the second interface. We see that $\phi_t = \phi_0$ because they are alternate interior angles. The equivalent angle of incidence ϕ_0 can then

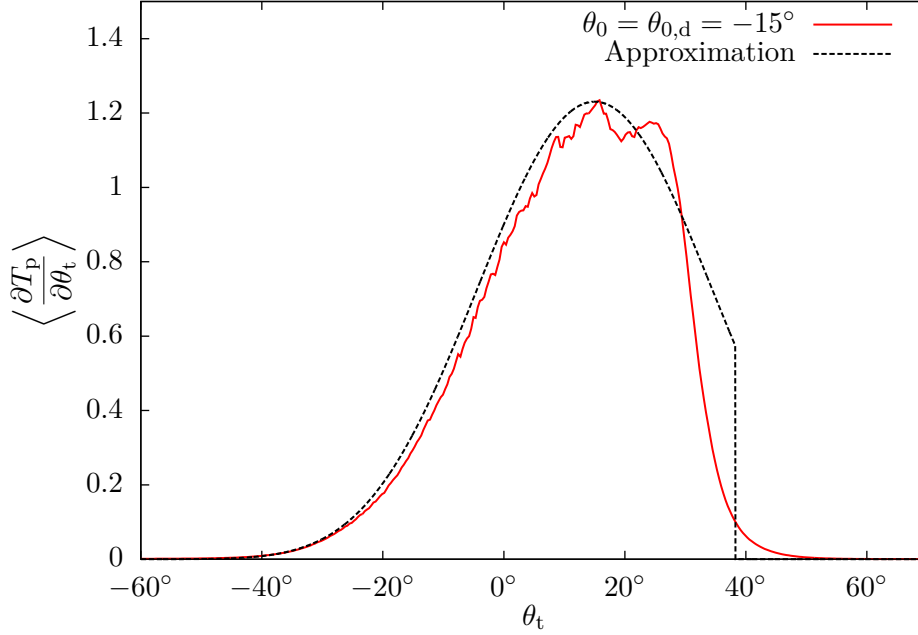


Figure 3.9: Results for a similar simulation to that in Fig. 3.7 but where $\theta_0 = \theta_{0,d} = -15^\circ$. The unitarity was 0.865. As is evident from the plot the result for this simulation is not as good. Around $\theta_t = 0^\circ$ the actual results closely follow the prediction of the Kirchhoff approximation, but after reaching the peak at $\theta_t = 15^\circ$ there are large discrepancies.

be calculated by Snell's law:

$$n_1 \sin \theta_0 = n_2 \sin \phi_t. \quad (3.1)$$

By running simulations on the system shown in Fig. 3.12 we should be able to approximate the system in Fig. 2.1 while gaining better accuracy.

Some of the physical interaction will be completely ignored when simulating a simplified system in the manner above. For example, some of the light will be reflected off the first interface in the original system. In the simplified system this interface is simply not there. Therefore a simulation run on the simplified system will not be able to analyse this. This can be corrected for by multiplying the MDTC with the transmission coefficient as given by the Fresnel equations from Section 1.2.5.

The second interface is of course present in the simplified system and reflection caused by this surface should be correctly modelled. Some of the light that is reflected from this interface will also be reflected the upper interface. This reflection cannot be taken into account by the simplified system.

The results of a simulation equivalent to the simulation in the comparison in Section 3.3.1 but using the simplified system are shown Fig. 3.13. Since $\theta_0 = 0^\circ$ then $\phi_0 = 0^\circ$ as well. The second comparison presented here is between two otherwise identical simulations where $\theta_0 = \theta_{0,d} = 0^\circ$ and Eq. (2.4) was used. Again, since $\theta_0 = 0^\circ$ then $\phi_0 = 0^\circ$. The results are shown in Fig. 3.14. The results are overall very similar. The

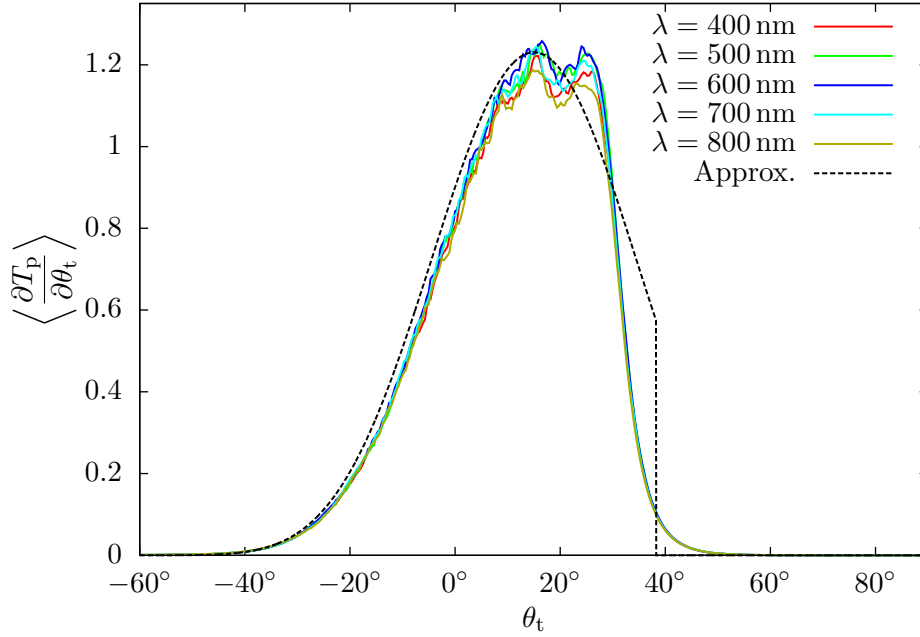


Figure 3.10: Results for a simulation with $\theta_0 = \theta_{0,d} = -15^\circ$ and $\theta_{t,d} = 15^\circ$. The number of surfaces were 10 000 in all five simulations. The unitarities were all approximately 0.86. As in Fig. 3.9 we see that for lower θ_t the simulations perform admirably, but that they become inaccurate after the peak. However, the results are almost identical for all wavelengths.

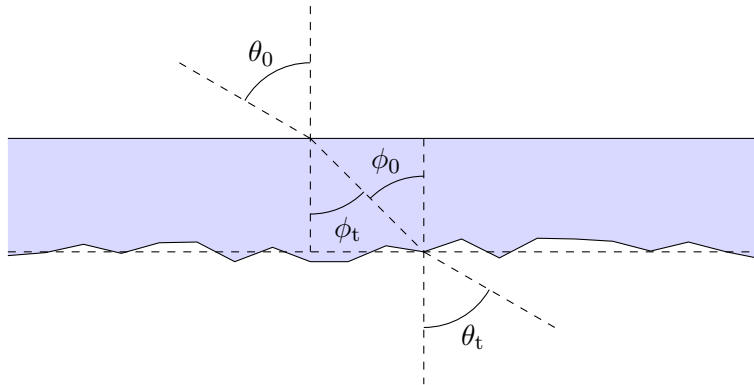


Figure 3.11: A simplified version of Fig. 2.1 with the intermediate angles ϕ_0 and ϕ_t indicated.

MDTC from the second simulation is visibly taller than the first. This is due to the increased unitarity: 1.03417 compared to 0.996679. The unitarity in the first simulation is therefore better in principle, since it is closer to one. A unitarity of 1.03 is nevertheless very good and similar to most of the results I achieved in my thesis [6]. On the outer regions of the MDTC, where it tapers off, the two results are almost identical while simultaneously noticeably different from the approximation.

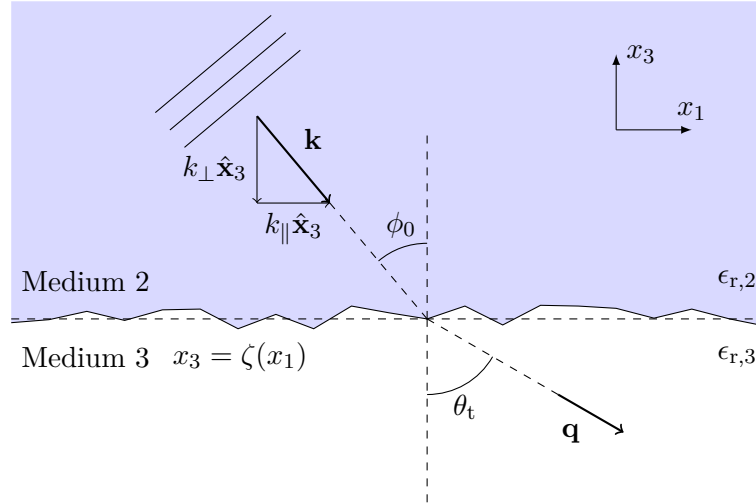


Figure 3.12: The simplified system. Choosing the correct ϕ_0 should make this system approximate the original system.

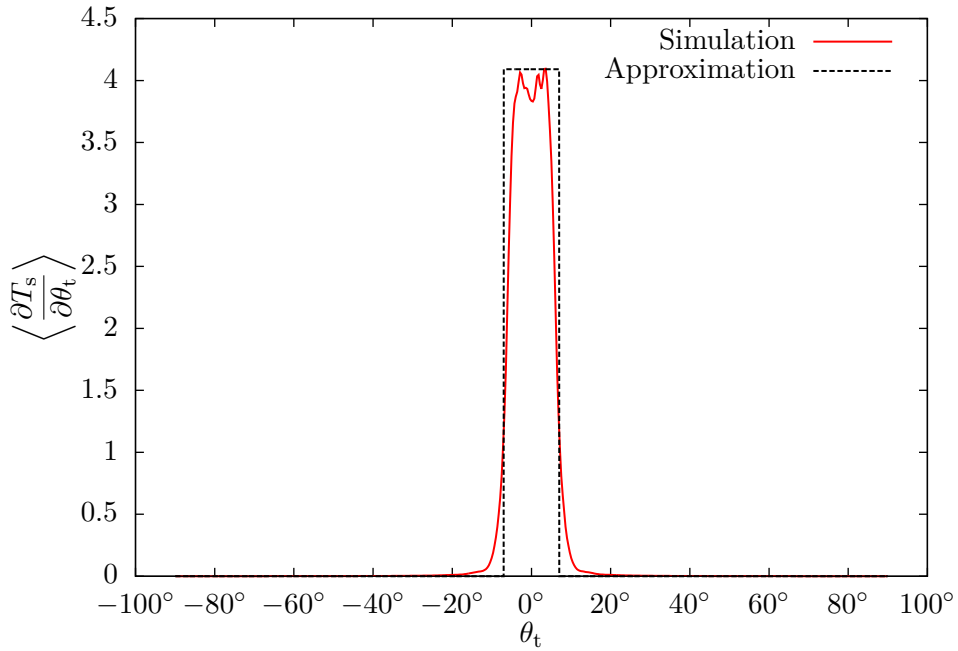


Figure 3.13: The results of a simulation similar to that in Section 3.3.1 which attempted to replicate the results achieved by Maradudin et al. [1]. Also shown is the Kirchhoff approximation. The results were very good and the unitarity was 1.0003. The number of surfaces was 5000. Despite the very good unitarity the simulation for the simplified system took only 30min to complete while the original simulation took approximately 4 hours.

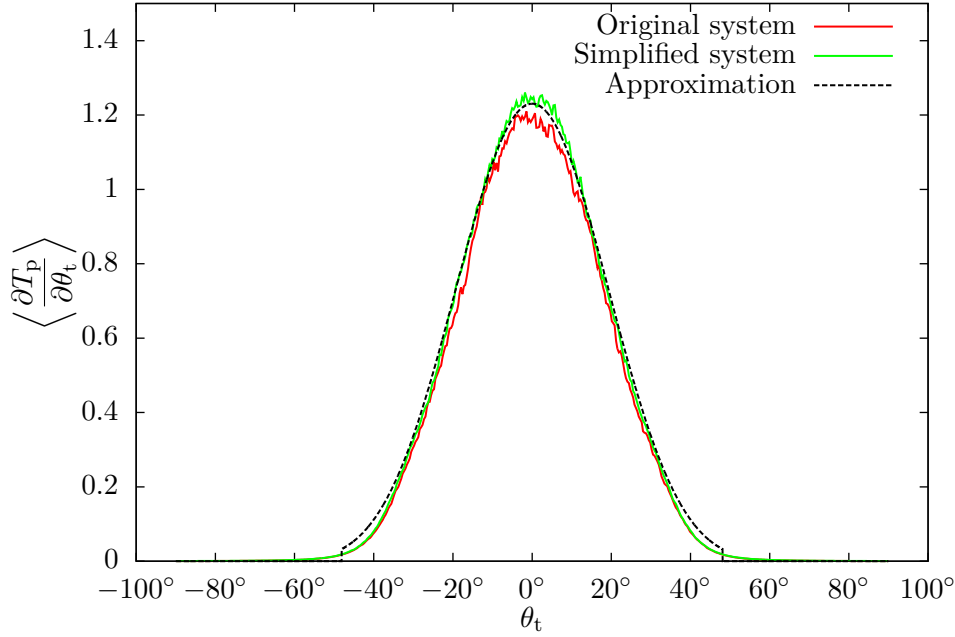


Figure 3.14: Comparison with the simulation first shown in Fig. 3.7 and the equivalent using the simplified system. The unitarity of the former was 0.996679. The latter simulation achieved a unitarity of 1.03417 which explains why the corresponding curve is slightly taller.

θ_0	ϕ_0
0°	0°
-10°	-6.59°
-15°	-9.85°
-20°	-13.07°
-30°	-19.31°

Table 3.1: Values for θ_0 used in this project and the corresponding values for ϕ_0 as given by Eq. (3.1). The relative permittivities were $\epsilon_1 = 1.01$ and $\epsilon_2 = 2.31$ which are the same as used previously.

For non-normal angles of incidence θ_0 the equivalent angle of incidence ϕ_0 is not zero. Table 3.1 lists the corresponding values used here.

Figure 3.15 shows the results from a simulation done with the simplified system where $\theta_0 = \theta_{0,d} = -15^\circ$. In the original simulation an unitarity of 0.89 was obtained whereas the unitarity was 1.01 when the simplified system was used. Because of this the measured MDTC for the simplified system is significantly higher than that of the original system.

As mentioned earlier the reflection from the first interfaces is not taken into account. This can be corrected for by calculating the transmission coefficients using Eqs. (1.7a) and (1.7c). The transmission coefficient for the system above where

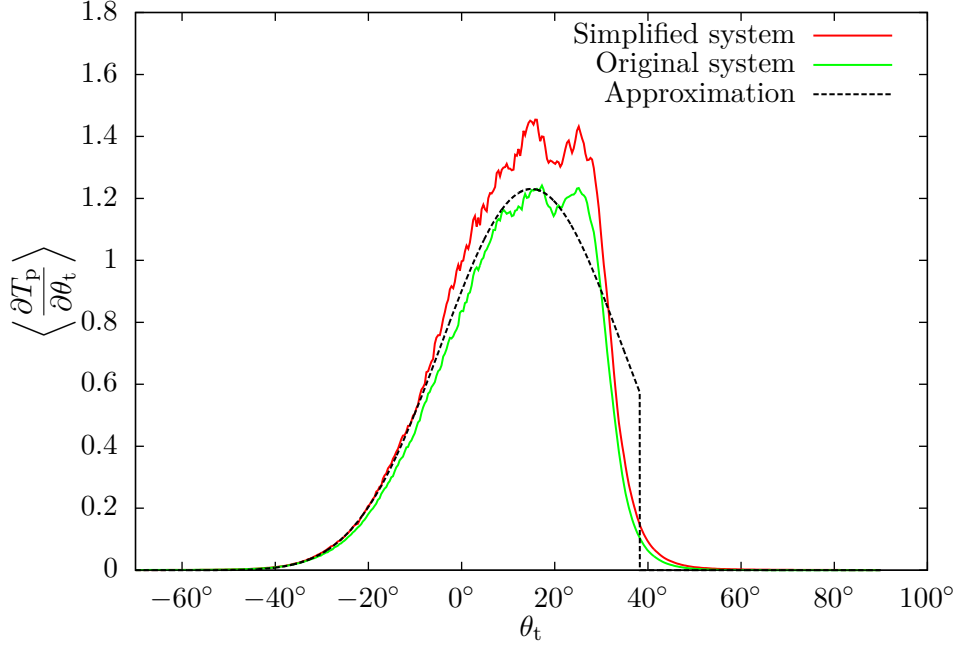


Figure 3.15: Comparison between results for the measured MDTC for the original and the simplified system when $\theta_0 = \theta_{0,d} = -15^\circ$ and $\theta_{t,d} = 15^\circ$. The results for the original system are the same as those shown in Fig. 3.7. The unitarities were 0.89 for the original system and 1.01 for the simplified system. This explains why the results for the simplified system are much higher. In addition the light that would otherwise be reflected upon hitting the second medium is not taken into account by the simplified system.

$\theta_0 = -15^\circ$ and $\phi_t = -9.815^\circ$ and the light is p-polarized is approximately 0.96. This explains the discrepancy between the results somewhat but the calculated MDTC for the simplified system is still significantly higher.

3.3.6 Non-ideal angles of incidence

In Section 3.3.4 we studied how the windows would handle light of different wavelengths and polarizations. It is necessary that they handle this since the light emitted by the sun is a spectrum containing all visible wavelengths. This is one example of non-ideal conditions.

As the sun moves across the sky the angle the incident light makes with the window necessarily changes. The windows should therefore be able to accommodate light incident on them when the angle of incidence θ_0 is different incident angle of design $\theta_{0,d}$. This was tested in Section 3.3.1 for the flat system used by [1]. There it was found that the flat surface handled non-normal incidence relatively well, albeit displaced from the expected angle of transmission. The overall shapes of the measured MDTCs were roughly similar.

The results from similar experiments for the MDTC shown in Eq. (2.4) are not

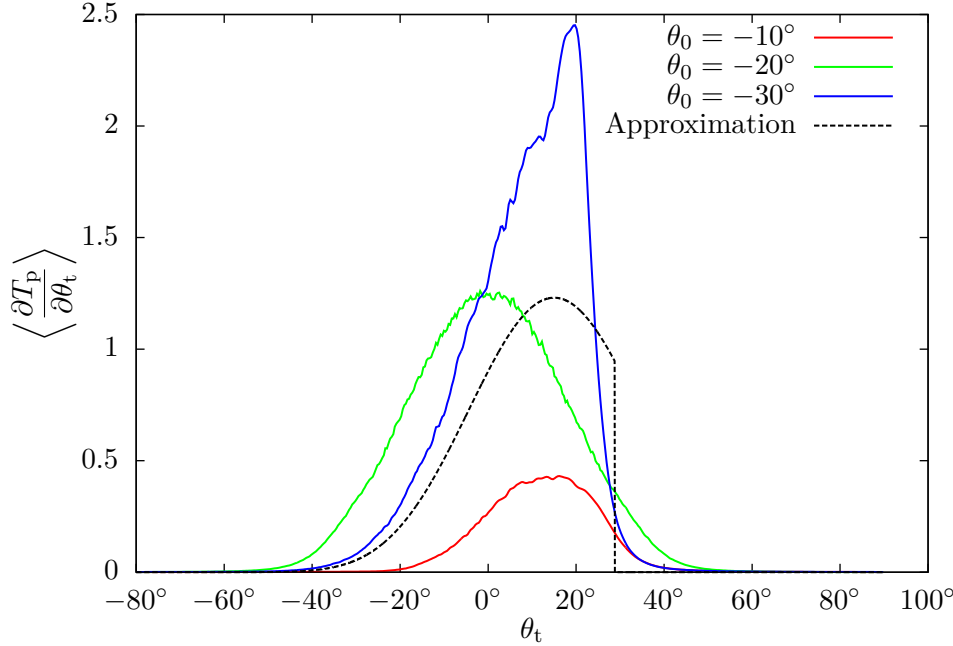


Figure 3.16: The measured MDTCs when $\theta_{t,d} = 15^\circ$, $\theta_{0,d} = -30^\circ$ and the actual angle of incidence θ_0 as indicated in the legend. The unitarities were 0.724957, 1.19627, and 1.30241 for angles of incidence -10° , -20° and -30° , respectively. Even in the case where $\theta_0 = \theta_{0,d}$ the unitarity is not good.

as good. Figure 3.16 shows the results for a simulation where $\theta_{t,d}$ is fixed at 15° , the surface is designed for an incident angle of design $\theta_{0,d}$ of -30° and the actual angle of incidence θ_0 varies from 0° to 30° as indicated in the legend. The unitarity when $\theta_0 = -10^\circ$ was 0.724957. When $\theta_0 = -20^\circ$ the unitarity was 1.19627 and finally when the angle of incidence was -30° a unitarity of 1.30241 was recorded. The latter is notable because it was actually the angle of incidence the surface was designed for. This type of unexpectedly poor results are also presented results from other simulations as well. When the angles involved become high enough the unitarity drifts away from 1 in both directions.

The initial suspect was the simplified system. To investigate this similar simulations to the ones presented above were run using the original system. In this case as well the unitarities were not good either. As noted in Section 3.3.3 the unitarities were much lower, probably caused by some of the light escaping out of the system and therefore not “counted”. However, despite the unitarities, the results between corresponding simulations for the original and simplified systems match very well. Figure 3.17 displays results for both the original system and the simplified system. The unitarities for the original system are consistently much lower than 1

That the results from the two different types of simulations show similar trends despite the bad unitarities may suggest that the qualitative results of the simulations

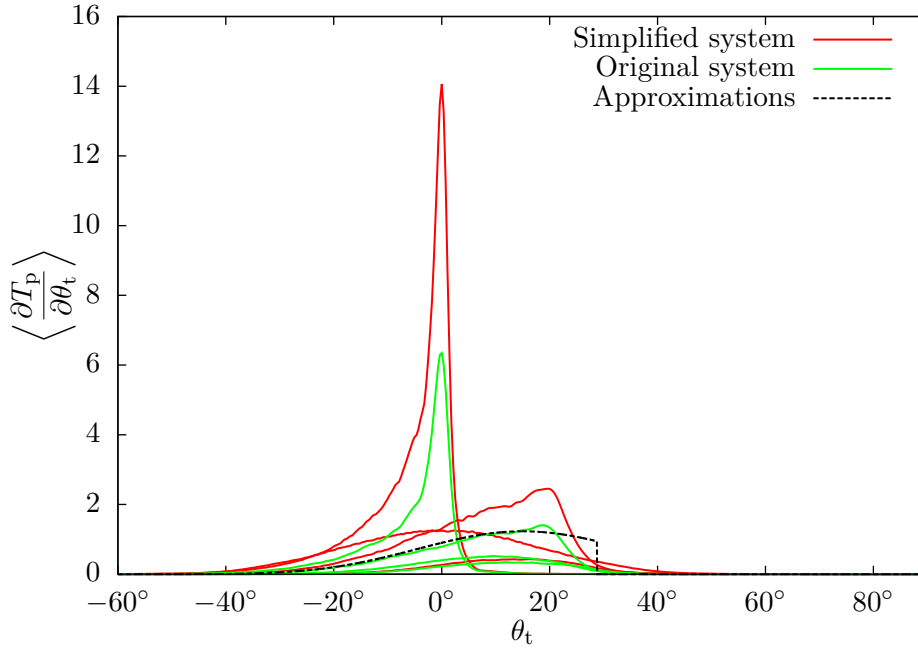


Figure 3.17: The plot shows the same results as in Fig. 3.16 compared with the results of similar results where the original system was used. The effect of the unitarities are spotted as the curves for the simplified system are in all cases much larger than the equivalent curves for the original system. Still, corresponding curves unmistakably possess the same characteristics, such as overall shape and for which values of θ_t the MDTCs reach their maxima. This may suggest that qualitatively the results can be trusted.

are correct. If that is the case then the surfaces do not handle non-ideal angles of incidence very well, potentially making it impossible to use objects with these properties as windows.

4.1 Summary

Windows can be constructed with the purpose of spreading the light transmitted through the window in order to improve indoor lighting conditions and reduce the need for artificial lighting. By using the Kirchhoff approximation in the geometrical optical limit in combination with a chosen general form of the window surfaces a expression for the statistical properties was found, given a desired profile for the transmitted light.

In this thesis the probability distribution function for the slopes of a window surface consisting of joined line segments was implemented in the simulation software `Maxwell11D`. Several simulations have been run exploring the feasibility of such windows. It has been demonstrated that creating a window with a desired profile of the transmitted light is possible. Such a window will perform very well for different wavelengths of the incident light and for varying polarization. For angles of incidence that deviate from the expected angle of incidence, however, the windows are not able to redirect the light in the manner specified beforehand.

By using Snell's law to compensate for simplifying the original system under consideration, the time needed for the simulations can be greatly reduced while simultaneously obtaining a higher accuracy in the results. This is very useful since more time-consuming simulations are needed to investigate this further.

4.2 Further work

In this thesis two profiles for the mean differential transmission coefficient have been considered. In practice, it may well be that other MDTC are better suited for the purposes in mind. The Kirchhoff approximation does not assume anything in particular about

4 Conclusion

the shape of the desired MDTC. Running simulations with other MDTCs is therefore straightforward and the Fortran 90 source code is easily extendible.

When the rough surface of the window has a mean slope one runs in to aesthetic problems. One may end up in a situation where the thickness of the window at the bottom is for example 1 cm while at the top it is several tens of centimetres. A possible solution to this problem is to let the window consist of several stripes with individual slopes. Then the thickness is reset several times going from top to bottom. Another way would be to let only the uppermost part of the window have the ability to redirect the light. This would also solve the problem of not being able to see out the window.

The simulations performed in this thesis were strictly one-dimensional. In possible real-life applications it may be interesting to redirect the light in two dimensions to e.g. focus the drop shape seen in Fig. 1.4 towards the centre of the ceiling of an indoor space. This may lead to unwanted effects as the sun moves across the sky and needs to be studied. The surfaces must be described by a function $x_3 = \zeta(x_1, x_2)$ and the expression for the mean differential transmission coefficient reworked.

Mathematical relations and definitions

The following is the vector identity used in the derivation of the wave equations for the electromagnetic field:

$$\nabla \times \nabla \times \mathbf{A} = \nabla(\nabla \cdot \mathbf{A}) - \nabla^2 \mathbf{A}. \quad (\text{A.1})$$

The approximations used in the presentation of the results are calculated from the following large expression [10]:

$$\begin{aligned} \left\langle \frac{\partial T_\nu}{\partial \theta_t} \right\rangle (q|k) &= \frac{4 |A_\nu|^2}{\sqrt{\epsilon_1} \cos \theta_{0,d}} \frac{\kappa_\nu}{\mu_\nu} \frac{1}{\left| \sqrt{\epsilon_3} \cos \theta_t - \sqrt{\epsilon_2 - \epsilon_1 \sin^2 \theta_{0,d}} \right|^3} \\ &\times \frac{\left| [(P + Q) - \frac{1}{2}(\epsilon_2 + \epsilon_3)]^2 - \frac{1}{3}(\epsilon_2 - \epsilon_3)^2 \right|^2}{\left| \left(\epsilon_2 - \frac{\lambda_\nu}{\mu_\nu} \epsilon_3 \right) + \left(\frac{\lambda_\nu}{\mu_\nu} - 1 \right) (P + Q) \right|^2} \\ &\times f \left(\frac{\sqrt{\epsilon_3} \sin \theta_t - \sqrt{\epsilon_1} \sin \theta_{0,d}}{\sqrt{\epsilon_3} \cos \theta_t - \sqrt{\epsilon_2 - \epsilon_1 \sin^2 \theta_{0,d}}} \right). \end{aligned} \quad (\text{A.2})$$

In this expression two helper quantities P and Q appear. They are defined as

$$P = \sqrt{\epsilon_3(\epsilon_2 - \epsilon_1 \sin^2 \theta_{0,d})} \cos \theta_t \quad (\text{A.3})$$

and

$$Q = \sqrt{\epsilon_1 \epsilon_3} \sin \theta_{0,d} \sin \theta_t.$$

The quantity A_ν is given by

$$A_\nu = \frac{2\lambda_\nu \alpha_1(k)}{\lambda_\nu \alpha_1(k) + \kappa_\nu \alpha_2(k)} \exp[i\alpha_2(k)d].$$

A Mathematical relations and definitions

Furthermore, Eq. (A.2) is only non-zero if

$$\text{sgn}(\sqrt{\epsilon_3} \cos \theta_t - \sqrt{\epsilon_2 - \epsilon_1 \sin^2 \theta_{0,d}}) = \text{sgn}(\epsilon_3 - P - Q).$$

See Winjum [10] for more details regarding this and the Kirchhoff approximation.

B.1 Excerpts from utility programs

The following Python program was used to run the simulations. It uses the modules `multiprocessing` and `subprocess` from the Python standard library to run several instances of `Maxwell11D` at once. The program takes a list of configuration files for `Maxwell11D` as input and runs the simulation once for each of these. If many configuration files are specified on the command line, the program starts at most as many simulation as half of the amount of CPU cores on the system. As soon as one simulation is completed, another will start until there are no more configuration files in the queue.

In addition, the program writes a short log file. It logs the actual command that was run, what date it was started and how long the simulation took, measured in wall-clock time. The simulation software itself measures CPU time used.

Lastly, it sets the niceness of its own process to 19. This niceness is inherited by all processes that is started by it, so all the simulations are run with a niceness of 19 as well. The simulation will use as much CPU as possible, but only if there are no other processes with a higher priority present. This means that two near identical simulations run on separate computers or at different times can use very different amounts of time, depending on the availability of the computer. In practice the computers were “free” most of the time, and in any case the recording of wall-clock time was mostly for my own benefit.

Listing B.1: “Wrapper” for `Maxwell11D` capable of starting and logging several simulation instances simultaneously.

```
#!/usr/bin/python

from sys import argv as sifFiles
import time, os, datetime
import subprocess as subp
```

B Source code excerpts

```
import multiprocessing as multip
from socket import gethostname

def maxwell_process(sifFile):
    cmd = list(maxwell)
    cmd[cmd.index("sifFile")] = sifFile
    cmd[cmd.index("ncFile")] = sifFile.replace("sif", "nc")
    logg = open(sifFile.replace("sif", "log"), "w")
    logg.write("Kommando:_" + "_".join(cmd) + "\n")
    logg.write("Dato:_" + time.strftime("%d/%m/%Y_%X") + "\n")
    print "Starter_simulering_for", sifFile, "paa", hostname
    tid = time.time()
    process = subp.Popen(cmd)
    process.wait()
    tid = time.time() - tid
    logg.write("Ekte_tid:_" + str(datetime.timedelta(seconds=tid)))
    logg.close()
    print "Ferdig_med", sifFile

sifFiles.pop(0) # Fjerner navnet paa programmet selv
os.nice(19)

#print sifFiles
for sifFile in sifFiles:
    if not ".sif" in sifFile:
        print "Attempted_to_start_with_a_non-sif_file!"
        exit()

hostname = gethostname().split(".")[0].upper()
maxwell = [ "./Maxwell1D_x64", "-p", "sifFile", "-o", "ncFile" ]
cpu_count=multip.cpu_count()/2
pool = multip.Pool(processes=cpu_count)
print "Starting_pool"
result = pool.map_async(maxwell_process, sifFiles)
print "Pool_started"
pool.close()
pool.join()
print "Done"
```

The following Python program was used for batch generation of configuration files. A template configuration file—itself not a valid configuration file—is read. Placeholder text in that file is then replaced by the desired values and is written to a new file. The desired values are specified in the code of the program itself. Since Python programs do not have to be compiled before before running them, specifying values inside the program is in principle just as easy as specifying them in an additional configuration file.

All the desired values can be specified either as a single value or as a Python-type list of values. In the latter case one configuration file is written for each of the values in the list. If more than one such list is present among the desired values, one configuration file is written for all possible combinations of the values in the lists.

In all cases a separate random seed is provided for all the files.

All the configuration files are named according to the following scheme. A prefix is given as an argument on the command line. If any of the desired values are given as lists, the name and value is specified the name of the configuration file. Lastly, an optional postfix is appended if present.

Listing B.2: Python script for batch generation of configuration files.

```
#!/usr/bin/env python
# coding=utf-8

from sys import argv
from random import randint
from types import ListType as List
from itertools import product

def filename(config, keylist):
    filnavn = prefix
    for key in keylist:
        filnavn += "-" + key + "_" + str(config[key])
    filnavn += postfix + ".sif"
    return filnavn

def dictinlist(settings):
    settingsList = []
    keylist = []
    for k, v in settings.iteritems():
        if type(v) == List:
            keylist.append(k)
    args = [ settings[k] for k in keylist ]
    prod = product(*args)
    while True:
        try:
            tmp2 = prod.next()
            tmp = dict(settings)
            for i, j in enumerate(tmp2):
                tmp[keylist[i]] = j
            settingsList.append(tmp)
        except StopIteration:
            break
    return (settingsList, keylist)

def makesif(configs):
    keylist = configs[1]
    configs = configs[0]
    for config in configs:
        sifstringre = sifstring
        for k, v in config.iteritems():
            sifstringre = sifstringre.replace(string+'seed',
                str(randint(0, 1e10)))
            sifstringre = sifstringre.replace(string+'S',
                str(config['wavel']/config['S']))
            sifstringre = sifstringre.replace(string+'F',
```

B Source code excerpts

```

                                str(config['wavel']*config['F']))
sifstringre = sifstringre.replace(string+'L',
                                str(config['wavel']*config['F']*config['L']
sifstringre = sifstringre.replace(string+k, str(v))
sifFile = filename(config,keylist)
print sifFile
sif = open(sifFile, "w")
sif.write(sifstringre)
sif.close()

string="placeholder"

prefix=""
postfix=""
if len(argv) == 1:
    exit()
if len(argv) >= 2:
    prefix = argv[1]
if len(argv) >=3:
    postfix = argv[2]

wavelength = 0.6127
t00 = -40
config = {
'wavel' : wavelength,
'yoff' : 30.0,
'N' : 10000,
'e1' : 1.01,
'e2' : 2.31,
'e3' : 1.01,
'pol' : 'p',
't0d' : t00,
'ttd' : -t00,
'dTdt' : 1,
'seed' : randint(0,1e10),
'solver' : '!',
'S' : 5,
'F' : 10,
'L' : 10,
't0' : t00
}

zif = open("ossem.zif", "r")
sifstring = zif.read()
configs = dictinlist(config)
print "Producing", len(configs[0]), "files..."
makesif(configs)
print "Done"
```

Maxwell1D itself is capable of accepting lists of parameters as input for some of the parameters, such as the angle of incidence θ_0 and the wavelength λ . The convenience of this script, however, made it desirable to generate separate configuration files for each

simulation. I then used another script to extract information about the parameters and the results of the simulation and gathered it in a spreadsheet.

The rest of the additional programs used are available in the accompanying zip archive.

B.2 Excerpts from Maxwell11D

The bulk of the code added to Maxwell11D by me resides in a file over 700 lines long. Therefore, only some snippets of that file is shown here.

Listing B.3 shows the implementation of Eq. (2.14). Most of the input parameters are stored in an object called LDP and are retrieved from there.

I noticed that in Eqs. (2.4), (2.6) and (2.14) there are many factors that are repeated, such as $\gamma r - s$ and $1 + \gamma^2$. Therefore I centralized parts of the calculations by writing a separate function calculating some of the common factors. In Listings B.3 to B.5 these are retrieved by calling that function, which is called `calc_common()`.

Listing B.3: Implementation of Eq. (2.14) in Maxwell11D.

```

Function Prob(g) result (res)
  Implicit none
  real(wp) :: g, res, t0, td, tt, eps(0:2)
  real(wp) :: r, grms, rpgs, eng2, frac, sqrtSign
  real(wp) :: prefactor, factor1, factor2over, factor2under, resdTdt
  real(wp) :: k, l, u, storSak
  character :: pol
  t0 = LDP%theta0_design
  eps = LDP%eps
  pol = LDP%pol
  r = LDP%r
  call calc_common(g)
  grms = LDP%grms
  rpgs = LDP%rpgs
  eng2 = LDP%eng2
  frac = LDP%frac
  sqrtSign = LDP%sqrtSign
  if (sqrtSign < 0._wp) then
    res = 0._wp
  return
endif
  sqrtSign=sqrt(sqrtSign)
  storSak = grms**2+rpgs*sqrtSign

  if (pol == 'p') then
    k = eps(0)
    l = eps(1)
    u = eps(2)
  elseif (pol == 's') then
    k = 1._wp
    l = 1._wp
    u = 1._wp

```

B Source code excerpts

```
endif

prefactor = (1*sqrt(eps(0))*cos(t0)+k*r)**2*u/(16._wp*1**2*sqrt(eps(0))*co
factor1 = abs(frac*(g*grms+sqrtSign)-r)**3
factor2over = ((eps(1)-1*eps(2)/u)+(1/u-1._wp)*frac*storSak)**2
factor2under = ((frac*storSak-0.5_wp*(eps(1)+eps(2)))**2-0.25_wp*(eps(1)-e
if (LDP%dTdt == 1) then
    resdTdt = cosinedTdt(g)
else
    resdTdt = rectdTdt(g)
endif

res = prefactor*factor1*factor2over/factor2under*resdTdt
End Function Prob
```

Note that the function is told which MDTC to use by setting a separate variable. This variable is set in the configuration file, which may in turn is set by the corresponding value to the dictionary key `dTdt` in Listing B.2.

Listings B.4 and B.5 show the implementations of Eqs. (2.6) and (2.6).

Listing B.4: Implementation of Eq. (2.4) in Maxwell11D.

```
Function cosinedTdt(g) result(res)
    Implicit none
    integer :: n = 9
    real(wp) :: C = 1._wp
    real(wp) :: g, res, t0, td, tt, eps(0:2)
    real(wp) :: grms, rpgs, eng2, frac, sqrtSign
    td = LDP%thetad_design
    t0 = LDP%theta0_design
    eps = LDP%eps
    call calc_common(g)
    grms = LDP%grms
    rpgs = LDP%rpgs
    eng2 = LDP%eng2
    frac = LDP%frac
    sqrtSign = LDP%sqrtSign
    sqrtSign = sqrt(sqrtSign)

    res = C*(cos(td)/(sqrt(eps(2))*eng2)*(g*grms+sqrtSign)+
        sin(td)/(sqrt(eps(2))*eng2)*(-grms+g*sqrtSign))**n
End Function cosinedTdt
```

Listing B.5: Implementation of Eq. (2.6) in Maxwell11D.

```
Function rectdTdt(g) result(res)
    Implicit none
    real(wp) :: g, res, t0, td, tt, eps(0:2)
    real(wp) :: grms, rpgs, eng2, frac, sqrtSign
    eps = LDP%eps
    td = LDP%thetad_design
    call calc_common(g)
    grms = LDP%grms
```



```

eng2 = LDP%eng2
sqrtSign = sqrt(LDP%sqrtSign)
tt = (-grms+g*sqrtSign)/(sqrt(eps(2))*eng2)
tt = abs(asin(tt))
if (tt < td) then
    res = 1._wp
else
    res = 0._wp
endif
End Function rectdTdt

```

There was a function in Maxwell11D initially that produced random numbers via the rejection method. This was however tied heavily to the existing implementations of the probability density function, so a replacement was written.

Listing B.6 displays my version the rejection method function. It does exactly what is shown in Algorithm 1. It accepts one input parameter, which is an array it fills with random numbers drawn with the rejection method. The auxiliary function $g(x)$ is chosen to be constant and equal to 1.1 times the maximum value of $f(\gamma)$.

Listing B.6: Implementation of Algorithm 1 in Maxwell11D

```

Subroutine Random_deviate(RD)
    implicit none
    real(wp) :: RD(:)
    real(wp) :: x(2), a, b, c, maximum
    integer :: i,N
    N = size(RD)
    ! Get random y values
    a = -LDP%gamma_max
    b = LDP%gamma_max
    c = 1.1_wp
    maximum = c*characteristics(0)
    do i=1,N
        !write(*,*) i
        call random_number(x)
        x(2) = a + (b-a)*x(2)
        do while (x(1)*c > prob(x(2))/maximum)
            call random_number(x)
            x(2) = a + (b-a)*x(2)
        enddo
        RD(i) = x(2)
    enddo
    Return
End Subroutine Random_deviate

```

References

- [1] Alexei A. Maradudin, Eugenio R. Mendez, Tamara A. Leskova, and Ingve Simonsen. Band-limited uniform diffuser in transmission ii: a multilayer system. volume 4447, pages 130–139. SPIE, 2001. doi: 10.1117/12.446727.
- [2] D.J. Griffiths. *Introduction to Electrodynamics*. Pearson Benjamin Cummings, 2008.
- [3] I.N Karatsoreos, S Bhagat, E.B Bloss, J.H Morrison, and B.S. McEwen. Disruption of circadian clocks has ramifications for metabolism, brain, and behavior. *Proceedings of the National Academy of Sciences of the United States of America*, 108(4): 1657–1662, 2011. cited By (since 1996) 0.
- [4] Mohamed Boubekri. *Daylighting, Architecture and Health: Building Design Strategies*. Architectural Press, 2008. ISBN 9780750667241.
- [5] S. H. A. Begemann, G. J. van den Beld, and A. D. Tenner. Daylight, artificial light and people in an office environment, overview of visual and biological responses. *International Journal of Industrial Ergonomics*, 20(3):231 – 239, 1997. ISSN 0169-8141. doi: DOI:10.1016/S0169-8141(96)00053-4.
- [6] Ø. Walle. Electromagnetic scattering from randomly rough surfaces. *Norwegian University of Science and Technology*, 2010. Project thesis.
- [7] T.A Leskova, A.A Maradudin, E.R Méndez, and A.V. Shchegrov. The design and fabrication of one-dimensional random surfaces with specified scattering properties. *Fizika Tverdogo Tela*, 41(5):918–924, 1999. cited By (since 1996) 1.
- [8] H.S. Lim and G. Kim. Predicted performance of shading devices for healthy visual environment. *Indoor and Built Environment*, 19(4):486–496, 2010. cited By (since 1996) 1.

References

- [9] I. Simonsen. Optics of surface disordered systems. *The European Physical Journal - Special Topics*, 181:1–103, 2010. ISSN 1951-6355.
- [10] I. Winjum. Designer surfaces. *Norwegian University of Science and Technology*, 2010. Project thesis.
- [11] A.A. Maradudin, E.R. Méndez, and T.A. Leskova. *Designer surfaces*. Amsterdam Oxford : Elsevier, 2008.
- [12] H. A. van der Vorst. Bi-cgstab: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 13(2):631–644, 1992. doi: 10.1137/0913035.