# Integrating coherent and incoherent models: RayFlare

Phoebe Pearce (pmp31@cam.ac.uk)

April 2021

## 1 Introduction

In order to produce electricity, a solar cell must first absorb photons, and thus maximizing how much sunlight is absorbed by a solar cell is unsurprisingly an important research area for improving solar cell efficiency. Light-trapping strategies range from relatively simple, for example the use of reflective materials such as metals at the rear surface of a cell, to highly-engineered nanophotonic structures to maximize broadband absorption of sunlight in very thin layers. Strategies to maximize absorption fall broadly into two categories: minimizing front-surface reflection, and making sure light which is able to enter the cell remains trapped until it is absorbed in the photovoltaic layers (some structures, such a pyramids on an Si cell, may perform both of these functions).

The optical design of solar cells is extremely important for all types of device, whether they are III-V-based, (multi)-crystalline or amorphous Si, thin-film devices, organic materials or otherwise. There are many strategies for increasing the path length inside the cell to increase absorption, e.g. causing light to travel at a more oblique angle and/or have multiple passes due to internal reflection. Assuming light enters perpendicularly to the cell interface with no front surface reflection and makes a single pass through the cell, and is not reflected at the back surface, the total absorption in the layer will be given simply by the Beer-Lambert law:

$$A_{BL} = 1 - e^{-\alpha d} \tag{1}$$

A theoretically simple enhancement which significantly boosts absorption in a cell is to assume perfect rear reflection, so that the light makes a double pass through the cell (again, assuming normal incidence):

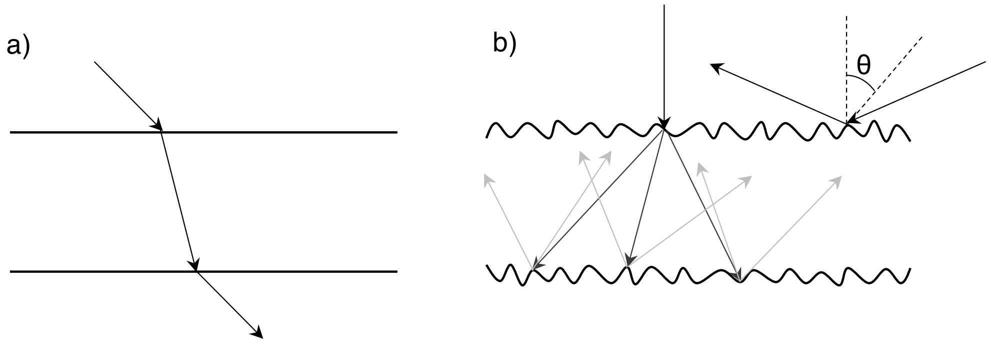

Figure 1: Optical sheets with a) planar surface and b) a randomizing surface texture with acceptance angle $\theta$.

$$A_{DP} = 1 - e^{-2\alpha d} \tag{2}$$

However, absorption can be improved further by scattering the light into more oblique angles inside the cell, increasing the path length. A theoretical upper limit to absorption enhancement can be derived using a thermodynamic approach; according to Minano [1], in this limit the average path length of rays entering into a cell is given by:

$$\bar{l}_{th} \leq \frac{4n^2}{\sin^2(\theta)} \tag{3}$$

With the upper limit occurring when all ray paths in the cell are illuminated at maximum brightness [2]. The angle $\theta$ is the "angular acceptance" of the cell: only incident light within a cone of $\theta$ from the surface normal can enter the cell (see Fig. 1). In general, for a planar surface being illuminated isotropically, light may enter from any $\theta$ between 0 and 90°, and the expression for the limit reduces to:

$$\bar{l}_{Lam} = 4n^2 \tag{4}$$

Yablonovitch and Cody [3] showed that the mean path length for ideal Lambertian scattering is identical to equation 4; a Lambertian scatterer is an ideal diffuse reflector, having the same radiance (radiant flux per unit projected unit solid angle) regardless of the viewing angle. This increase in path length leads to an absorption given by:

$$A_{\text{YC}} = \frac{4n^2\alpha d}{1 + 4n^2\alpha d} \tag{5}$$

In fact, although this value can be computed at any wavelength with corresponding values of $n$ and $\alpha$, it is only physically meaningful in the limit of low absorption ($\alpha \to 0$). Green [4] generalized the Lambertian limit over all wavelengths, giving:

$$A_{\text{Green}} = \frac{1 - e^{-4\alpha d}}{1 - \left[1 - \frac{1}{n^2}\right] e^{-4\alpha d}} \tag{6}$$

As expected, $A_{\text{Green}}$ reduces to $A_{\text{YC}}$ in the limit $\alpha d \ll 1$, so that $e^{-4\alpha d} \approx 1 - 4\alpha d$.

Practically, increased path lengths inside cells can be achieved in various ways, including through diffraction gratings [2, 5], random or periodic scattering structures [6], reflectors [3], and plasmonic structures involving metals [7, 8]. Such structures can be placed on the front and/or rear of the cell, or even between junctions of a multi-junction cell [5] depending on the device design. Highly efficient Si cells generally have random or regular pyramidal textures to reduce front-surface reflectivity, and increase path length inside the cell by redirecting light into more oblique angles and increasing internal reflection [9, 10, 11]. The Green/Lambertian limit applies to the maximum possible amount of broadband path enhancement without wavelength restriction. However, in a solar cell, we generally only require light-trapping over a limited wavelength range, generally at energies just above the bandgap where absorption is weakest, and thus it is possible to exceed the Green limit over a limited wavelength range [12, 2, 13, 14, 15] by using periodic scattering structures.

Optical simulations are key in developing, analysing and optimizing novel concepts for light-trapping architectures. Depending on the device structure and light-trapping strategies used, very different modelling methods may be appropriate. Some of these methods include:

1. The Beer-Lambert law (eq. 1), describing absorption in a bulk material without scattering or additional front surface or interface reflections; it is often used in conjunction with other methods or to provide initial estimates of the absorption.

2. Ray-tracing. A geometrical optics method in which light is treated as a collection of rays which can reflect or refract at material interfaces. These methods are appropriate when describing the interaction of light with material layers and features with characteristic sizes much larger than the wavelength, a regime in which wave-optical effects such as diffraction and interference can be neglected.

3. Transfer-matrix method (TMM) [16]. A wave-optical method which deals correctly with the reflection of light at multiple interfaces, and resulting interference. This method can only describe planar layers; any structure such as a grating or nanoparticles can only be described using an effective medium approximation (EMA) which treats the structure as a one-dimensional layer with an effective refractive index.

4. Rigorous coupled-wave analysis [17, 18]. A wave-optical Maxwell solver also referred to as the Fourier modal method (FMM). RCWA is a method for solving Maxwell's equations in layer stacks which may contain structures such as gratings.

5. Guided mode analysis [19, 20]. By treating the solar cell as a waveguide, in which certain guided modes can be excited through diffraction by a periodic structure (depending on the geometry and period of the grating). However, the effectiveness of diffraction into each possible mode will depend on the precise specifications of the grating, and thus this method can only predict where resonant features may occur; thus it can be used to inform the development of structures alongside e.g. RCWA.

Separate implementations for each of these methods exist, but for certain light-trapping architectures which combine optical features over multiple length scales, it may not be physically appropriate or computationally efficient to use a single method for the full structure. For example, for a silicon cell with a thickness on the order of 100 µm with a rear-side grating with a period of 1 µm [21] it is possible to use a Maxwell solver such as RCWA, or alternative Finite Element (FEM) or Finite Difference Time Domain (FDTD) methods, to describe the behaviour of the full structure, but this will be extremely inefficient computationally due to the very large difference in feature sizes between the bulk thickness and the grating. Another technologically relevant example is the implementation of perovskite on Si tandem cells; the thin perovskite layers will need to be treated with e.g. TMM, while the silicon cell will generally be more suited to geometric optics (ray-tracing) methods if it has a pyramidal texture [22]. These problems have led several groups to independently develop matrix methods which can combine different optical methods across interfaces; implementations include GenPro4, developed by Santbergen et al. at TU Delft [23], the OPTOS formalism developed by Tucher, Eisenlohr et al. at Fraunhofer ISE [11, 24] and by Li et al. at UNSW Sydney [25]. For e.g. thin-film on Si tandem cells, methods combining ray-tracing for the pyramidal textures with reflection, transmission and absorption probabilities calculated through TMM have also been developed both in conjunction with general matrix methods, as for GenPro4 [23] or separately, including the CROWM simulator by Lipovšek et al. [26] (University of Ljubljana). RayFlare [27] combines several optical simulation methods into one open-source Python package.

Accurate and efficient optical modelling of different types of structures (planar vs. structured layers, random vs. periodic structures, nano vs. micro-scale or larger) requires the use of different computational methods. Although in principle most problems can be solved by the most sophisticated type of solver available (in the case of the types of optical simulations we are considering, a full Maxwell solver), this is often not computationally efficient and can be less insightful than using a method with appropriate simplifications for the structure being considered. For instance, modelling micron-sized pyramids with a Maxwell solver such as RCWA or FDTD (finite-difference time-domain), or using RCWA to calculate the effect of a distributed Bragg reflector (DBR) made of planar layers, is possible and should yield correct results but is both computationally slow compared to a more physically appropriate method (e.g. ray-tracing and TMM, respectively, for the examples given), and does not generally lend additional insight into the physical processes that choosing a suitable simpler model would not. Especially if the problem being considered has an optimization aspect, requiring the simulation of many different structures, it is important to choose a modelling method which can accurately describe the relevant structure but is not unnecessarily computationally intensive. The development of RayFlare began due to a desire to integrate the various optical modelling codes developed and adapted for different types of problems over the course of the PhD into a single user-friendly Python package, which can be used to solve a wide range of optical problems by implementing ray-tracing, TMM, RCWA, simple Beer-Lambert law absorption, analytic models for surface scattering (e.g. perfect mirrors or Lambertian scattering) and an angular matrix formalism [24, 11]. The angular matrix formalism allows coupling of surfaces treated with the same or different optical methods across a thick, incoherent layer. By using Solcore's [28] material system and database, and ability to define new materials easily it thus becomes possible to model the wavelength-dependent optical behaviour of a wide variety of structures including structures involving planar layers of any thickness, structures with diffraction gratings, or structures with larger-scale textures such as pyramids. Combining e.g. a structure with pyramids on one side and a diffraction grating on the other is possible through the matrix framework, and comparison with idealized cases such as a perfect mirror or Lambertian scattering is easy and fast.

The matrix framework used is similar to OPTOS [11, 24]. It divides the optical structure into three parts: the front surface, the bulk, and the back surface[1]. The front and back interfaces may be made up of several layers, or be textured, as long as there is a way to calculate the reflection and transmission for different angles of incidence, as well as (if relevant) the absorption in each layer of the interface. The matrix method itself is straightforward, and essentially amounts to matrix multiplication. For both interfaces, using any suitable optical method (e.g. TMM, ray-tracing, RCWA, or analytical expressions), a matrix must be constructed which describes how light incident on the interfaces is redistributed into other angles, or absorbed. In two dimensions, with a non-absorbing interface, this matrix will relate some polar angle $\theta_{in}$ to one or more $\theta_{out}$, with the matrix elements describing what fraction of incident light is directed into each outgoing angle. This can be expanded to three dimensions by including an azimuthal angle $\phi$ [11]. The angular matrix framework and structures of the matrices in 2D, specifically using an integrated ray-tracing/TMM approach, are illustrated in Fig. 2. The ray-tracer takes the probabilities of reflection and transmission from an angle- and wavelength-dependent TMM calculation, to account for the multi-layer structure of the front surface of a realistic solar cell. The OPTOS framework has been expanded since its initial publication to include absorption in surface layers [30]; this ability was also developed independently as part of RayFlare over the course of the PhD approximately simultaneously [27]. While the existing matrix frameworks discussed above provide methods for combining different surface textures, there is no existing open-source modelling package which combines the angular matrix method with the actual generation of the matrices describing the redistribution at each interface. OPTOS provides only the matrix multiplication for

[1] The formalism could also be expanded to a stack with multiple interfaces and several 'bulk' materials, though this has not yet been implemented.

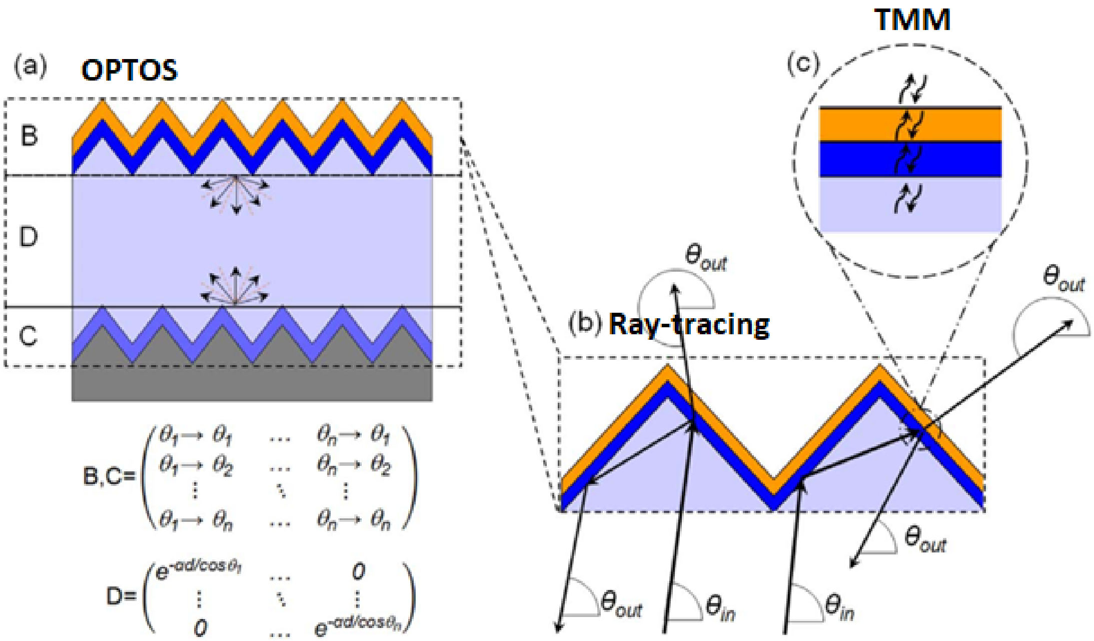
(a) OPTOS
B
D
C
B,C=
$\theta_1 \to \theta_1$ ... $\theta_n \to \theta_1$
$\theta_1 \to \theta_2$ ... $\theta_n \to \theta_2$
$\theta_1 \to \theta_n$ ... $\theta_n \to \theta_n$
D=
$e^{-\alpha d/\cos\theta_1}$ ... 0
0 ... $e^{-\alpha d/\cos\theta_n}$
TMM
(c)
(b) Ray-tracing
$\theta_{out}$
$\theta_{in}$

Figure 2: Schematic of the angular matrix framework, specifically when used with the integrated ray-tracing/TMM method, showing a) the function of the matrices B, C and D; b) how the matrix B could be calculated through ray-tracing; and c) how the TMM can be used with the ray-tracer to calculate reflection, absorption and transmission for a multi-layer surface. From [29].

calculating absorption in the bulk layers, but the matrices have to be computed and stored in the correct format by the user using external methods, while GenPro4 [23] is a proprietary software which combines the matrix approach with the generation of matrices but does not include an RCWA solver or other method for calculating diffraction from wavelength-scale periodic structures. The aim of RayFlare is to provide a free, open-source solution integrating both the generation of the matrices and the matrix multiplication to generate useful results, as well as tools for visualizing the outputs and matrices generated.

This section describes the work undertaken so far in developing the integrated optical modelling package RayFlare. The new ray-tracing implementation, developed specifically for RayFlare, is discussed first, followed by the description of the matrix method and validation of the various methods contained within RayFlare.

## 2 Ray-tracing

The ray-tracer used was written in Python3 specifically for use with RayFlare. The basis of the ray-tracer is the definition of a surface in terms of the Delaunay triangulation of a set of $(x, y, z)$ points which define a surface, and the intersection of rays (directed lines) with those triangular planes. The surface could be very simple, such as regular V-grooves or pyramids (see Fig. 6a), or a more complicated surface based on e.g. an AFM scan of a real surface structure.

The equations for a point on a line **l** (i.e. a ray) and a triangular plane **p** (i.e. one of the planes defining the surface) can be expressed in parametric form as:

$$\begin{aligned}\mathbf{l} &= \mathbf{l_a} + \mathbf{d}t \\ \mathbf{p} &= \mathbf{p}_0 + \mathbf{p}_{01}u + \mathbf{p}_{02}v\end{aligned} \tag{7}$$

where $\mathbf{d}$ is a vector pointing in the ray's direction of travel. The meaning of the symbols relating to the line-plane intersection is shown in Fig. 3a. Setting the RHS of both equations in eq. 7 equal to one another (which occurs at the intersection of the line $\mathbf{l}$ and the plane $\mathbf{p}$) and rearranging, the following matrix equation can be found:

$$[\mathbf{l}_a - \mathbf{p}_0] = \begin{bmatrix} -\mathbf{d} & \mathbf{p}_{01} & \mathbf{p}_{02} \end{bmatrix} \begin{bmatrix} t \\ u \\ v \end{bmatrix} \tag{8}$$

Note that since $\mathbf{d}$, $\mathbf{p}_{01}$ and $\mathbf{p}_{02}$ are vectors, the first term on the RHS of this equation is a 3 by 3 matrix.

This matrix equation can be solved in the standard way, by inverting the matrix, which after rearranging and multiplying out gives:

$$\begin{aligned} t =& \frac{(\mathbf{p}_{01} \times \mathbf{p}_{02}) \cdot (\mathbf{r}_a - \mathbf{p}_0)}{-\mathbf{d} \cdot (\mathbf{p}_{01} \times \mathbf{p}_{02})} \\ u =& \frac{(\mathbf{p}_{02} \times -\mathbf{d}) \cdot (\mathbf{r}_a - \mathbf{p}_0)}{-\mathbf{d} \cdot (\mathbf{p}_{01} \times \mathbf{p}_{02})} \\ v =& \frac{(-\mathbf{d} \times \mathbf{p}_{01}) \cdot (\mathbf{r}_a - \mathbf{p}_0)}{-\mathbf{d} \cdot (\mathbf{p}_{01} \times \mathbf{p}_{02})} \end{aligned} \tag{9}$$

For the intersection of a triangular surface with corners at $\mathbf{p}_0$, $\mathbf{p}_1$ and $\mathbf{p}_2$, with a forward-travelling ray, the following must be satisfied:

- $t > 0$, since the ray is travelling forward; $t < 0$ means the intersection point lies behind the point of origin of the ray
- $u > 0$ and $v > 0$ for the point to lie inside the triangle
- $u + v \leq 1$ for the point to lie inside the triangle

This calculation is straightforward to carry out for all the triangles which define a surface; however, looping through each surface and checking whether the ray intersects with it is computationally inefficient. For a complicated surface, such as one defined by an AFM scan, there may be hundreds or thousands of individual triangles; even for a single square-based pyramid, there are four surfaces to check. Thus, vectorizing this calculation so that for any given line, all surfaces can be checked at once using matrix operations will lead to much faster code. This can be done by defining matrices $P_0$, $P_1$ and $P_2$ which contain the $p_0$, $p_1$, and $p_2$ points of all the surfaces in the triangulation, and matrix versions of $\mathbf{d}$ and $\mathbf{r}_a$:

$$P_0 = \begin{bmatrix} \mathbf{p}_{0,I} \\ \mathbf{p}_{0,II} \\ \vdots \\ \mathbf{p}_{0,N} \end{bmatrix} \quad R_a = \begin{bmatrix} \mathbf{r}_a \\ \mathbf{r}_a \\ \vdots \\ \mathbf{r}_a \end{bmatrix} \quad D = \begin{bmatrix} \mathbf{d} \\ \mathbf{d} \\ \vdots \\ \mathbf{d} \end{bmatrix} \tag{10}$$

where in the first equation, $\mathbf{p}_{0,I}$ refers to the $p_0$ coordinates of the first triangle in the surface triangulation, $\mathbf{p}_{0,II}$ refers to the second triangle, etc.

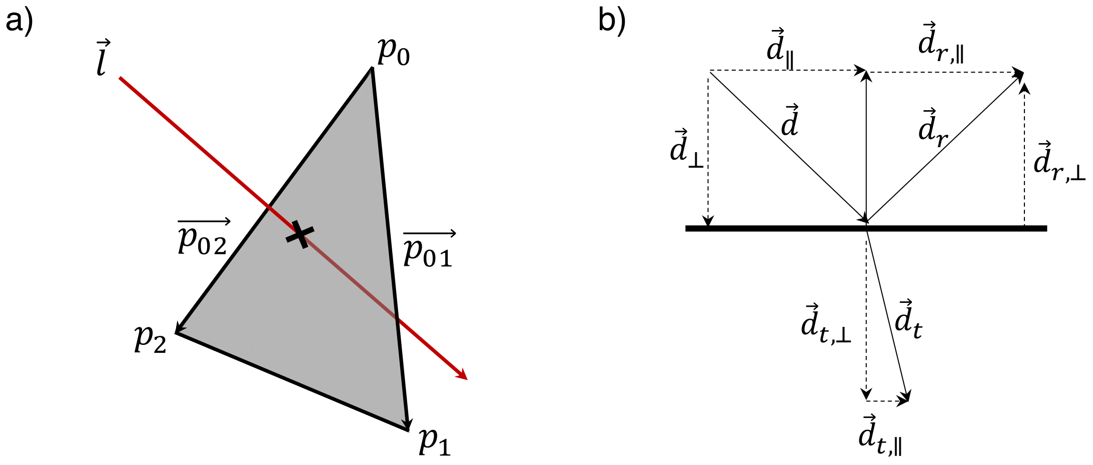

Figure 3: a) Intersection of a line with a finite, triangular plane. b) Reflection and transmission of a ray through an interface (in this case, from low to high refractive index).

Eq. 9 has some vectors which appear multiple times. Matrix forms of these values can be computed once, then used in the calculation of matrix forms of $t$, $u$ and $v$. The following quantities are defined:

$$C = \begin{bmatrix} (\mathbf{p}_{1,I} - \mathbf{p}_{0,I}) \times (\mathbf{p}_{2,I} - \mathbf{p}_{0,I}) \\ (\mathbf{p}_{1,II} - \mathbf{p}_{0,II}) \times (\mathbf{p}_{2,II} - \mathbf{p}_{0,II}) \\ \vdots \\ (\mathbf{p}_{1,N} - \mathbf{p}_{0,N}) \times (\mathbf{p}_{2,N} - \mathbf{p}_{0,N}) \end{bmatrix} = P_{01} \times P_{02} \tag{11}$$

$C$ is a matrix with dimensions $(n_{\text{triangles}}, 3)$ and contains the cross products of $\mathbf{p}_{01}$ and $\mathbf{p}_{02}$ for each of the $N$ triangles in the surface (one cross product per row). The '$\times$' symbol in the final expression is used to mean the row-wise cross product rather than the more standard cross or outer product. Note that because $C$ depends exclusively on the surface itself and not on the incident ray, it can be computed before ray-tracing begins and stored for use in each call of the check_intersect function (see Listing 1). The vector version of the denominator factor common to the expressions of $t$, $u$ and $v$ is:

$$P_i = \frac{1}{\sum_j (-D \odot C)_{ij}} \tag{12}$$

$P_i$ is a vector where the number of elements is equal to the number of triangles which define the surface. The '$\odot$' symbol is used to mean element-wise multiplication of the values in the matrix. The sum over the second matrix dimension $j$ achieves the dot product per column (i.e. summing over the three dimensions after multiplying element-wise). This factor can be recognized as the denominator common to the expressions for $t$, $u$ and $v$ in eq. 9.

Finally, the matrix version of the $(\mathbf{r}_a - \mathbf{p}_0)$ factor common to $t$, $u$ and $v$ is defined simply as $X = R_a - P_0$.

Combining all this, and considering the relevant cross products and dot products, the vector versions of $t$, $u$ and $v$ can be written as:

$$\begin{aligned} &t_i = P_i \textstyle\sum_j (C \odot X)_{ij} \\ &u_i = P_i \textstyle\sum_j \left[(P_{02} \times -D) \odot X\right]_{ij} \\ &v_i = P_i \textstyle\sum_j \left[(-D \times P_{01}) \odot X\right]_{ij} \end{aligned} \tag{13}$$

Note that throughout this section the use of indexing subscripts is *not* Einstein notation (summation of repeated indices is not implied – summation is explicitly denoted), but refers explicitly to which elements of the vectors should be multiplied together to give the $i$th element of $t$, $u$, and $v$. Computationally, all elements of $t$, $u$ and $v$ are calculated simultaneously using array multiplication in Python. Once $t$, $u$ and $v$ are computed, the conditions for intersection given above are checked simultaneously using logical 'and' statements to determine whether there is an intersection with each surface, resulting in a vector of 'True' and 'False' values with length equal to the number of triangles. Depending on the incident ray and surface, there may be more than one 'True' entry in the vector (the ray travelling in the forward direction can strike more than one triangle in the surface if allowed to continue travelling indefinitely); thus, if there are multiple intersections, the intersection with the minimum value of $t$ is found, corresponding to the closest triangle. The location of the intersection is then given simply by $\mathbf{x} = \mathbf{r_a} + t_{min}\mathbf{d}$. The surface normal and thus the local angle of incidence between $\mathbf{d}$ and the surface normal are also calculated, since this is needed to calculate the reflection and transmission probabilities using either the Fresnel equations directly, or the TMM if a multi-layer surface is being considered.

The check_intersect function (Listing 1) implements the mathematical procedure as outlined, and forms the core of the ray-tracing algorithm. In order to treat a real structure, multiple auxiliary functions are needed, which define the triangulated surfaces, materials and implement scanning across the surface for a large number of rays. The reflection and refraction of light in three dimensions, with the correct wavelength-dependent refractive index of the material on each side of the surface, must also be implemented:

$$\begin{aligned} &\vec{d_r} = \vec{d} - 2\vec{d_\perp} = \vec{d} - 2\left(\vec{d} \cdot \vec{N}\right)\vec{N} \\ &\vec{d}_{t,\parallel} = \tfrac{n_0}{n_1}\vec{d_\parallel} = \tfrac{n_0}{n_1}\left(\vec{d} - \left(\vec{d} \cdot \vec{N}\right)\vec{N}\right) \\ &\vec{d}_{t,\perp} = -\sqrt{1 - \left|\vec{d}_{t,\parallel}\right|^2}\,\vec{N} \\ &\vec{d_t} = \vec{d_\parallel} + \vec{d_\perp} \end{aligned} \tag{14}$$

which is shown schematically in Fig. 3b. $\vec{d}$ is the unit direction vector of the incoming ray, while $\vec{d_r}$ and $\vec{d_t}$ are the directions of the reflected and transmitted ray, respectively. $\vec{N}$ is the normal to the plane, pointing into the half-plane from which the ray $\vec{d}$ is entering. During reflection of the ray, the component of the ray perpendicular to the surface is flipped, while in transmission the component of the ray parallel to the surface obeys Snell's law ($n_0 \sin\theta_0 = n_1 \sin\theta_1$, or in vector notation $n_0\vec{d_\parallel} = n_1\vec{d}_{t,\parallel}$). The parallel ($\parallel$) and perpendicular ($\perp$) components are as indicated in Fig. 3b.

This describes a single interaction of the ray with a specific point of intersection with one triangular plane in the surface texture, but each ray must be tracked through the surface since in general a ray may interact with the same surface multiple times before being reflected or transmitted into the material above or below. Intersections with the surface texture inside one unit cell are checked until no further intersections are found; the ray is then translated back into the unit cell at the point where it would enter the next unit cell of the structure, assuming it is periodic in the $x$ and $y$ directions. This procedure is repeated until the ray is fully below (lower than the minimum point on the surface) or above (higher than the highest point on the surface) the triangulated surface. A diagram of a ray interacting with a square-based pyramid texture,

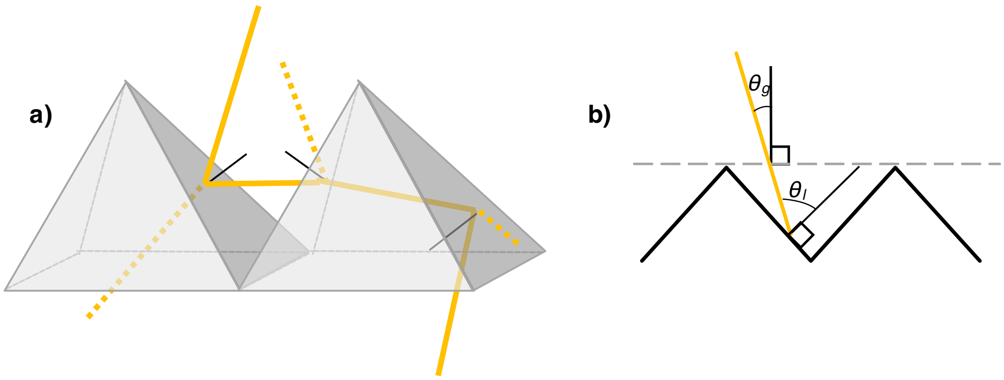

Figure 4: a) Schematic of three-dimensional ray-tracing for pyramids. b) Illustration of the meaning of the local angle of incidence $\theta_l$ and the global angle of incidence $\theta_g$.

interacting with multiple interfaces, is shown in Fig. 4a. In order to accurately capture the behaviour of a surface, the unit cell defining the surface must be scanned, since a ray striking e.g. the top of a pyramid will behave differently to a ray which strikes the structure near the base. The reflection and transmission probabilities can be calculated directly from the optical constants of the interface materials using the Fresnel equations, or calculated through TMM (see Section 2.2).

Code Listing 1: The check_intersection function which forms the basis of RayFlare's ray-tracing algorithm.

```
def check_intersect(r_a, d, tri):

    D = np.tile(-d, (tri.size, 1))
    pref = 1 / np.sum(D * tri.crossP, axis=1)
    corner = r_a - tri.P_0s
    t = pref * np.sum(tri.crossP * corner, axis=1)
    u = pref * np.sum(np.cross(tri.P_2s - tri.P_0s, D) * corner, axis=1)
    v = pref * np.sum(np.cross(D, tri.P_1s - tri.P_0s) * corner, axis=1)

    which_intersect = (u + v <= 1) & (np.all(np.vstack((u, v)) >= -1e-10, axis=0)) & (t > 0)

    if sum(which_intersect) > 0:

        t = t[which_intersect]
        P0 = tri.P_0s[which_intersect]
        P1 = tri.P_1s[which_intersect]
        P2 = tri.P_2s[which_intersect]
        ind = np.argmin(t)
        t = min(t)

        intersn = r_a + t * d
        N = np.cross(P1[ind] - P0[ind], P2[ind] - P0[ind])
        N = N / np.linalg.norm(N)

        theta = atan(np.linalg.norm(np.cross(N, -d))/np.dot(N, -d))  # in radians, angle relative
             to plane
        return [intersn, theta, N]
    else:
        return False
```

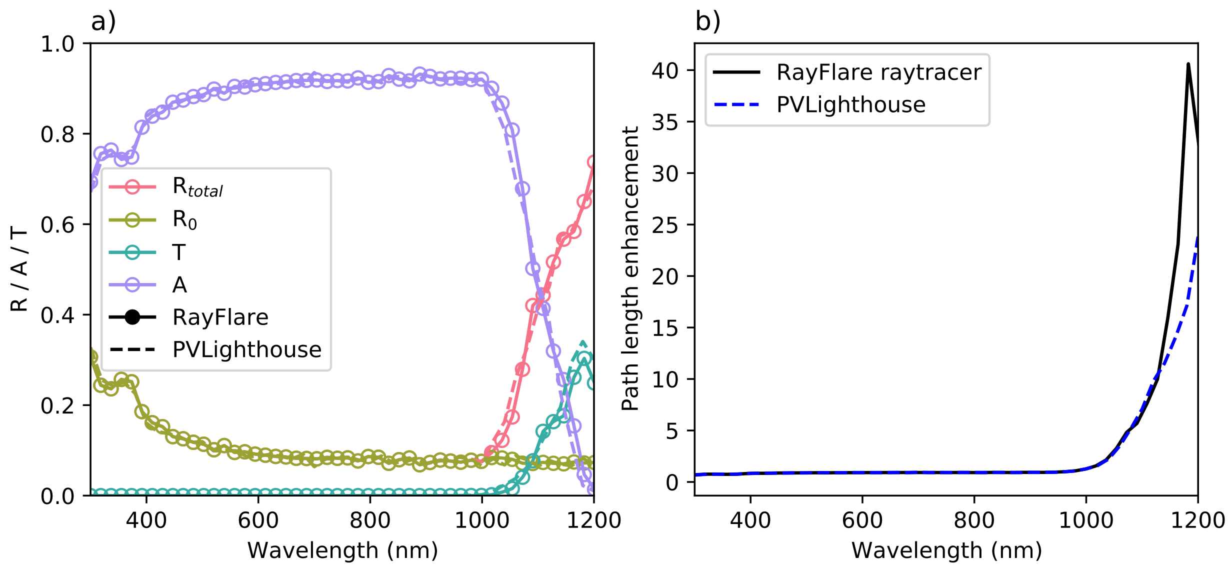

Figure 5: a) Comparison between the calculated reflection (total reflection and front surface reflection, $R_0$), transmission and absorption for a c-Si wafer, calculated using the widely-used PVLighthouse wafer ray-tracer [31] and the new Python3-based ray-tracer built for RayFlare. The structure considered is 300 µm of c-Si [32] with regular inverted pyramids with an elevation angle of 55° and base width 2 µm on the front surface. The rear surface is planar, and the surrounding medium is air. b) Comparison of the calculated path length enhancement using both methods.

Note that once an intersection has been found, before continuing to see if there are other intersections with the surface, the ray-tracer advances the ray by an infinitesimally small amount ($\mathbf{d} \times 10^{-9}$), otherwise the same intersection would immediately be found again. It is possible to use the ray-tracer in conjunction with the matrix method (Section 2.3) to calculate the angular redistribution of light at one interface, or to track the rays throughout the whole structure, interacting with the front and back surfaces until the ray is absorbed, transmitted, or reflected.

## 2.1 Model validation

To check if the ray-tracing algorithm is performing as expected, its results were compared to the widely-used PVLighthouse wafer ray-tracer [31], which is available as an online tool. Note that the matrix framework discussed below was not used in these simulations; ray-tracing was used to track the full path of rays through the structure. The same structures are defined in RayFlare and the PVLighthouse ray-tracer, consisting of a 300 µm thick Si wafer (optical constants in both cases were taken from [32]). The front surface of the wafer is textured with regular inverted pyramids with an opening angle of 55° while the rear of the cell is planar. The results in terms of total absorption, transmission, front surface reflection $R_0$, and total reflection calculated through both methods are shown in Fig. 5, showing very close agreement between both ray-tracing implementations. The total reflection includes the front surface reflection, i.e. light which is reflected during the first interaction of the incident light with the wafer, and escape reflection caused by rays which escape when light hits the front surface of the cell from the inside.

In addition, the results for the path length enhancement in an Si substrate patterned with random pyramids and regular pyramids depending on the substrate thickness are shown in Fig. 6. In both the

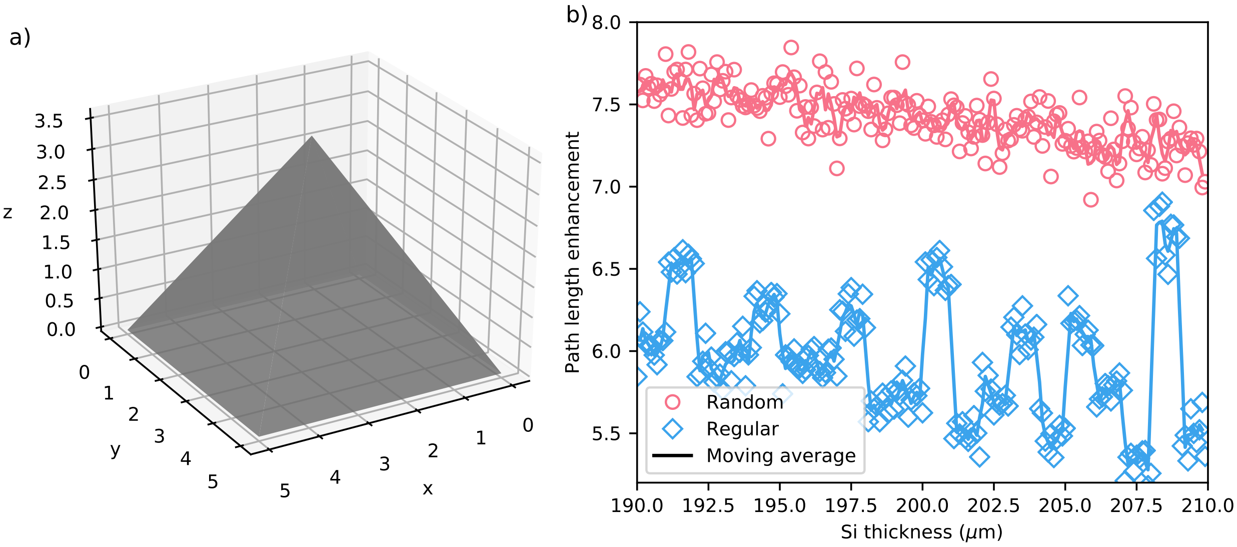

Figure 6: a) Example of a Delaunay triangulation surface object used for ray-tracing both regular and random pyramids. b) Comparison between the path length enhancement at 1100 nm compared to a single pass through a 200 µm Si substrate for structures with random (pink circles) and regular pyramids (blue diamonds) on the front surface, and a planar rear surface. The open symbols show the calculated data while the line shows a three-point moving average.

regular and random cases, the opening angle of the pyramids is 55°, the typical opening angle for chemically etched pyramids in Si. The size of the pyramids was set to 5 µm. The front surface is patterned with upright pyramids while the rear surface is planar. The triangulated surface used for both regular and random pyramids is shown in Fig. 6a; for the regular pyramids, the overall position in the unit cell is tracked while the ray makes passes inside the cell, preserving the location information of the ray. To simulate the effect of random pyramids, the same unit cell is used but the position of the ray is randomized before each interaction with the front or rear surface (the direction vector is kept at the value calculated during the previous surface interaction). The path length enhancement, depending on the Si thickness, at incident wavelength 1100 nm (close to the bandgap of Si) is shown in Fig. 6b. Fig. 6b shows that the path length enhancement for the regular pyramids is significantly lower than for the random pyramids. This is due to increased escape reflection through the front surface due to the relationship between the location on a pyramid at which the light enters the cell and the corresponding point on a pyramid which it will hit after traversing the cell. For truly random pyramids, the ray should be equally likely to encounter any $(x, y)$ position of the unit cell with no correlation to its previous interaction with the surface, while for regular pyramids, there is a clear geometric relationship between the face of the pyramid the ray strikes while entering the cell and the face it will strike after traversing the cell and reflecting from the planar back surface, as discussed in detail in [9]. This effect can enhance or reduce the light-trapping; if a ray strikes the equivalent face of a pyramid through which it was coupled into the structure, it will be totally internally reflected back into the cell, while if it strikes the opposite face, it is likely to couple straight out of the cell (the reflection probability below the critical angle at an Si/air interface is around 30%). The results here are consistent with the predictions in [9]: random pyramids on the front surface only, with a planar back surface, consistently perform better than regular pyramids in a square grid, and are not very sensitive to the substrate thickness except a general trend of increasing absorption as the thickness increases. Note that while the total absorption increases, the path length enhancement – the ratio of the fraction of light absorbed in the structure with light-trapping to the fraction of light absorbed in a single pass of a planar cell, calculated through the Beer-Lambert law – actually decreases, as shown in Fig. 6b. As the cell gets thicker, less absorption enhancement can be gained from the light-trapping structures, as the light has an increasing probability of being absorbed in a single pass. The performance of regular pyramids is extremely sensitive to the thickness of the substrate, shown by the sharp oscillations in Fig. 6b. For regular pyramids, the predicted interval between substrate thicknesses at which the path length enhancement reaches a local maximum is predicted to be [9]:

$$\Delta W' = (d/2)\tan(\theta_1 + \theta_2) \tag{15}$$

where $\theta_1 = \cot^{-1}(\sqrt{2}) \approx 0.615$ and $\theta_2 = \sin^{-1}\left[(\cos\theta_1)/n_3\right] \approx 0.233$, giving $\Delta W' = 2.83$ µm for $d = 5$ µm, matching the peak spacing observed in Fig. 6b.

The fact that regular pyramids consistently perform worse than random pyramids seems somewhat counter-intuitive, since random pyramids must represent some average over many different regular structures, and thus there must be regular structures which outperform a random structure for any wafer thickness and pyramid size. However, as shown in [9], these optimal regular structures are not structures where the pyramids are on a regular square grid, but rather regular structures where the overall grid is periodically staggered.

To further illustrate the effectiveness of randomizing the ray direction before rays return to a surface with similar features, Fig 7 compares the absorption in Si at long wavelengths for multiple structures with either regular V-grooves, random pyramids, or regular pyramids on the front surface and V-grooves, a planar rear surface, a perfect mirror or a perfect Lambertian reflector (see Section 2.7) on the rear surface. In each case, the feature size (pyramid base width or spacing between groove maxima) is 5 µm and the elevation angle of the features is 55°, with a substrate thickness of 100 µm. This means the front surface reflection for normal incidence in all cases is very similar (around 10% for the wavelength range 800-1200 nm), as Fig. 7 demonstrates. However, the transmission through the back surface and escape reflection vary significantly depending on the combination of front and rear surfaces. A structure with V-grooves on both sides performs significantly better when the V-grooves on the two surfaces are perpendicular; in the crossed case, most rays reaching the back surface cannot couple straight out by striking an equivalent surface to the one they entered through, reducing transmission, and the direction of the rays is changed before they re-encounter the front surface, thus also reducing escape reflection compared to the parallel case. Similarly, random front surface pyramids significantly outperform regular pyramids; in this case, the difference is mainly due to a reduction in escape reflection, since transmission through the planar back surface is similar. Random pyramids or a Lambertian scattering back surface lead to more absorption, with the highest absorption at long wavelengths achieved in the structure with random front pyramids and a Lambertian rear surface.

The results of the ray-tracing results are stochastic, as demonstrated by the fluctuations visible in the results shown here; a summary of the convergence of the ray-tracing for different numbers of rays and scanning points on the surface is shown in Appendix B.

## 2.2 Ray-tracing with integrated TMM

RayFlare can calculate the probability of reflection and transmission in the ray-tracer directly using the Fresnel equations; however, the probabilities can also be calculated using TMM if there are multiple surface layers, one or more of which may be absorbing. To be a useful approximation, the thickness of the surface

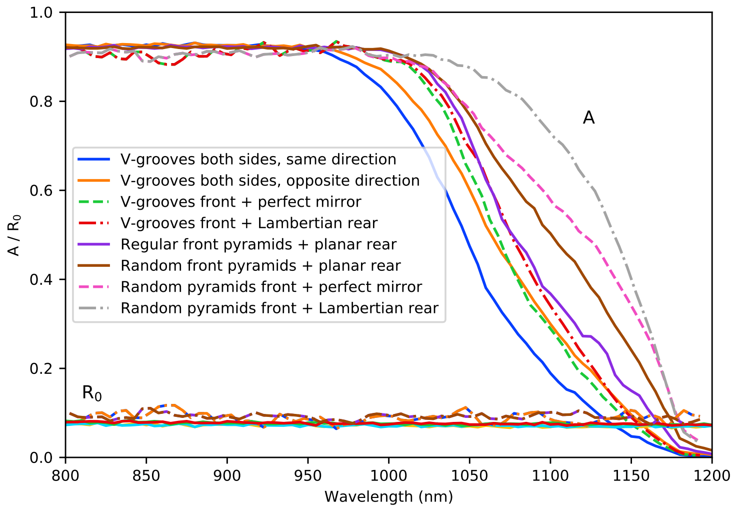

Figure 7: Comparison of the bulk absorption and front surface reflection $R_0$ for different structures with increasingly Lambertian behaviour.

textures should be thinner than the lateral dimensions of the surface texture (Fig. 2).

Computationally, it was found to be much faster to use the TMM to pre-compute lookup-tables with reflection, transmission and absorption probabilities which can be used as-needed by the ray-tracer, rather than doing individual TMM calculations each time those probabilities are needed. The TMM capabilities of RayFlare are based on the existing methods implemented in Solcore, which are vectorized over wavelengths, and thus pre-computing the probabilities for a large number of incidence angles and the desired wavelength values is relatively fast (taking a few seconds or minutes on a normal computer depending on the structure) compared to computing individual probabilities as-needed, especially for a large number of rays. The downside is that these large arrays must be stored, as discussed in Section 2.8.

## 2.3 Matrix framework for multi-scale optical calculations

As discussed in the introduction, it is not always appropriate to treat an optical structure with a single optical modelling method, for example because of diverging length scales (e.g. a diffraction grating with a period of hundreds of nanometres in a cell with a thickness of hundreds of microns). This is where the angular matrix method comes in; by calculating matrices which describe how incoming light is redistributed into outgoing directions by one interface of a structure (or absorbed in interface layers), optical methods can be combined across a single structure. The power fraction in each angular bin, $P(\theta_i, \phi_j)$, at any point within the simulation is represented as a vector $\vec{v}$:

$$\vec{v} = \begin{pmatrix} P(\theta_1, \phi_1) \\ P(\theta_1, \phi_2) \\ \vdots \\ P(\theta_2, \phi_1) \\ \vdots \\ P(\theta_m, \phi_n) \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{16}$$

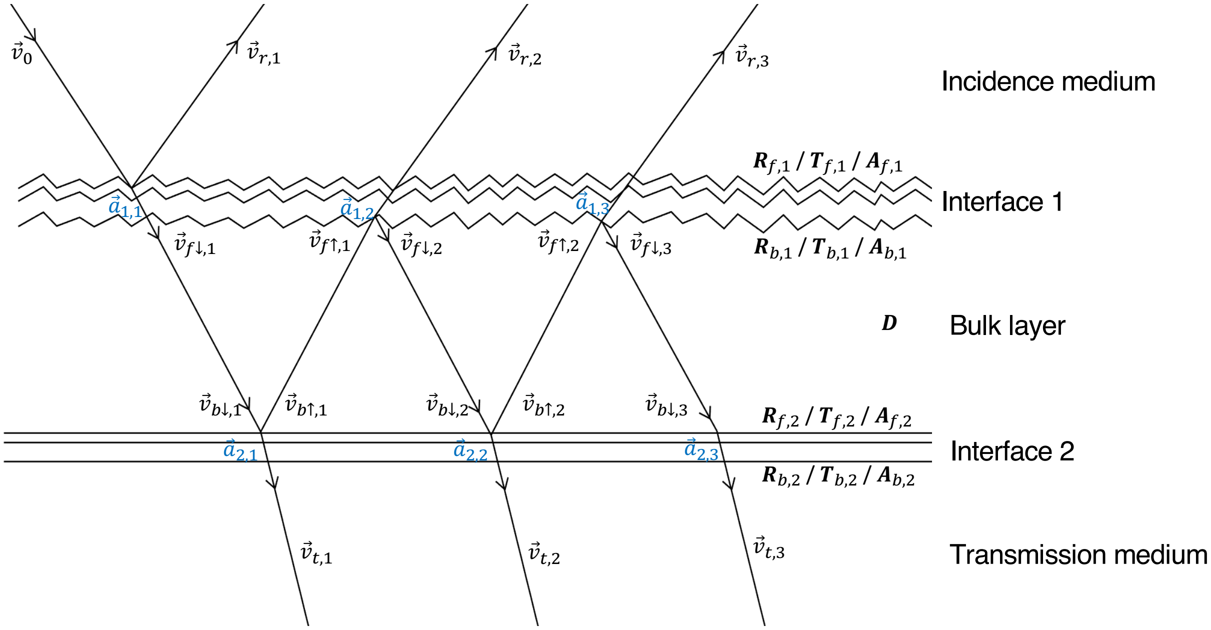
$\vec{v}_0$
$\vec{v}_{r,1}$
$\vec{v}_{r,2}$
$\vec{v}_{r,3}$
Incidence medium
$\boldsymbol{R}_{f,1}$ / $\boldsymbol{T}_{f,1}$ / $\boldsymbol{A}_{f,1}$
Interface 1
$\vec{a}_{1,1}$
$\vec{a}_{1,2}$
$\vec{a}_{1,3}$
$\vec{v}_{f\downarrow,1}$
$\vec{v}_{f\uparrow,1}$
$\vec{v}_{f\downarrow,2}$
$\vec{v}_{f\uparrow,2}$
$\vec{v}_{f\downarrow,3}$
$\boldsymbol{R}_{b,1}$ / $\boldsymbol{T}_{b,1}$ / $\boldsymbol{A}_{b,1}$
$\boldsymbol{D}$
Bulk layer
$\vec{v}_{b\downarrow,1}$
$\vec{v}_{b\uparrow,1}$
$\vec{v}_{b\downarrow,2}$
$\vec{v}_{b\uparrow,2}$
$\vec{v}_{b\downarrow,3}$
$\boldsymbol{R}_{f,2}$ / $\boldsymbol{T}_{f,2}$ / $\boldsymbol{A}_{f,2}$
Interface 2
$\vec{a}_{2,1}$
$\vec{a}_{2,2}$
$\vec{a}_{2,3}$
$\boldsymbol{R}_{b,2}$ / $\boldsymbol{T}_{b,2}$ / $\boldsymbol{A}_{b,2}$
$\vec{v}_{t,1}$
$\vec{v}_{t,2}$
$\vec{v}_{t,3}$
Transmission medium

Figure 8: Schematic showing the labelling conventions used in RayFlare.

The length of the vector is $l$, the total number of angle bins (combinations of $\theta$ and $\phi$). The specific example for $\vec{v}$ shown on the right-hand side of eq. 16 is for light incident from $\theta = 0°$ which has not yet interacted with any texture. Often, this is the form $\vec{v_0}$, the vector representing the incident light, will take. The discretization of $\theta$ and $\phi$ used here is the one proposed for the three-dimensional implementation of OPTOS [30]. The polar angle bins have equal $\sin(\theta)$ spacing; this means the $\mathbf{k}$ vector of light projected onto the surface planes has uniform spacing [24]. The azimuthal angle spacing is equidistant, and the number of azimuthal angle bins for any polar angle bin is given by $N_{\text{azimuth}} = \lceil c_{\text{azimuth}} \cdot r_{\text{polar}} \rceil$, where $r$ is the index of the polar angle bin (starting from $r = 1$ for the $\theta = 0$ bin), $\lceil \rceil$ denotes rounding up the nearest integer, and $c_{azimuth}$ is a number between 0 and 1 which describes how fine the discretization should be. This discretization method means that the size of the angle bins when projected onto the plane in which the surfaces lie is approximately equal for all bins. The value of $c_{azimuth}$ used for all results presented here is 1/4, demonstrated to give accurate results while reducing computation time relative to using $c_{azimuth} = 1$ [11].

Fig. 8 shows an outline of how the propagation of light across a thick (compared to the wavelength of light) layer with front and rear textures is described by RayFlare. The $\vec{v}$ vectors, with length $l$, describe light propagating in the incidence or transmission medium, or within the structure, while the $\vec{a}$ vectors track the light absorbed in the interface layers. The $\vec{v_r}$ and $\vec{v_t}$ vectors track light which travels into the semi-infinite incidence and transmission media, respectively. Note that there is a difference between the meaning of the $\vec{v_f}/\vec{v_b}$ vectors inside the structure and the $\vec{v_r}/\vec{v_t}$ vectors outside the structure; the former track the total fraction of incident intensity left in the light propagating inside the structure over the course of the simulation, while the latter track how much power escapes from the structure at each interaction with the interface, and thus need to be summed over to give total reflection or transmission in each angular bin. The $\vec{a}$ vectors have length equal to the number of layers in the corresponding interface.

The $\mathbf{R}$ and $\mathbf{T}$ matrices describe the redistribution of light into other angular channels at each interface, either transmitted through the interface or reflected back into the same half-plane from which the light is incident. The $R$ and $T$ labels clarify whether the matrix describes reflection or transmission through an interface, but could describe light incident from either the front or back of the texture; the subscripted '$f$' or '$b$' labels are used to distinguish incidence from the front and back respectively ('front' meaning the side of any interface closest to the incidence medium), while the subscripted number describes which interface the matrices describe. For instance, matrix $\mathbf{T_{b,1}}$ describes the angular redistribution when light incident from the inside of the structure onto the back of the front interface is transmitted into the incidence medium. For the 3D case, including azimuthal angle discretization, the matrices $\mathbf{R}$ and $\mathbf{T}$ take the form:

$$\mathbf{R}, \mathbf{T} = \begin{pmatrix} p(\{\theta_1, \phi_1\} \rightarrow \{\theta_1, \phi_1\}) & p(\{\theta_1, \phi_2\} \rightarrow \{\theta_1, \phi_1\}) & \dots & p(\{\theta_n, \phi_m\} \rightarrow \{\theta_1, \phi_1\}) \\ p(\{\theta_1, \phi_1\} \rightarrow \{\theta_1, \phi_2\}) & p(\{\theta_1, \phi_2\} \rightarrow \{\theta_1, \phi_2\}) & \dots & p(\{\theta_n, \phi_m\} \rightarrow \{\theta_1, \phi_2\}) \\ \vdots & \ddots & & \vdots \\ & & & \\ p(\{\theta_1, \phi_1\} \rightarrow \{\theta_n, \phi_m\}) & p(\{\theta_1, \phi_2\} \rightarrow \{\theta_n, \phi_m\}) & \dots & p(\{\theta_n, \phi_m\} \rightarrow \{\theta_n, \phi_m\}) \end{pmatrix} \tag{17}$$

where $p(\{\theta_i, \phi_j\} \rightarrow \{\theta_k, \phi_l\})$ is the probability that light incident from a direction in the $\{\theta_i, \phi_j\}$ bin is scattered into a direction in the $\{\theta_k, \phi_l\}$ bin.

The A matrices describing absorption take the form:

$$\mathbf{A} = \begin{pmatrix} p\left(\{\theta_1, \phi_1\} \rightarrow A_1\right) & p\left(\{\theta_1, \phi_2\} \rightarrow A_1\right) & \cdots & p\left(\{\theta_m, \phi_n\} \rightarrow A_1\right) \\ p\left(\{\theta_1, \phi_1\} \rightarrow A_2\right) & p\left(\{\theta_1, \phi_2\} \rightarrow A_2\right) & \cdots & p\left(\{\theta_m, \phi_n\} \rightarrow A_2\right) \\ \vdots & \vdots & \ddots & \vdots \\ p\left(\{\theta_1, \phi_1\} \rightarrow A_k\right) & p\left(\{\theta_1, \phi_2\} \rightarrow A_k\right) & \cdots & p\left(\{\theta_m, \phi_n\} \rightarrow A_k\right) \end{pmatrix} \tag{18}$$

Where $p\left(\{\theta_1, \phi_1\} \rightarrow A_1\right)$ is the probability that light incident from a direction that falls in the $\{\theta_1, \phi_1\}$ bin is absorbed in layer 1 of the interface. The dimensions of the $\mathbf{A_{f/b,i}}$ matrix depend on which interface it is describing; the number of columns is $l$, as for the $\mathbf{R}$ and $\mathbf{T}$ matrices, while the number of rows is equal to the number of layers in the interface with label $i$ (thus the number of rows is interface-dependent, because not every interface has the same number of surface layers). In this labelling convention, and the code, 'layer 1' of an interface is always the layer at the front (closest to the incidence medium), and not necessarily the first layer the light encounters; light incident from inside the structure on the front surface would encounter the $k$th layer first.

Note that while Fig. 8 shows the light travelling through the structure as discrete rays to avoid confusion, light is generally scattered into multiple directions at each interface; the $\vec{v}$ vectors do not represent light travelling in a single direction, but keep track of the fraction of the incident intensity in each angular bin.

The definition of the angles $\theta$ and $\phi$ is somewhat ambiguous; Fig. 9 shows how light interacting with an interface with incident angles $\{\theta_1, \phi_1\}$ is scattered into outgoing angles $\{\theta_2, \phi_2\}$. $\theta$ is the polar angle from the positive $z$-direction, while $\phi$ is the azimuthal angle counter-clockwise from the $x$-axis (when viewed from the $z > 0$ half-plane). This coordinate system fully describes three-dimensional space with ranges $0 \leq \theta < \pi$ and $0 \leq \phi < 2\pi$. Keeping the same definition of $\theta$ and $\phi$, the scattered ray travelling away from the first interaction in the $\{\theta_2, \phi_2\}$ direction will then be incident on a subsequent surface with polar angle $\theta_2' = \pi - \theta_2$ and azimuthal angle $\phi_2' = \pi + \phi_2$. The same coordinate system, with the $z$-axis pointing towards or into the incidence medium of the structure, is used for all interfaces. The transformation $\theta \rightarrow \pi - \theta$ and $\phi \rightarrow \pi + \phi$ is applied to angles describing 'outgoing' directions to convert them into 'incoming' directions after each

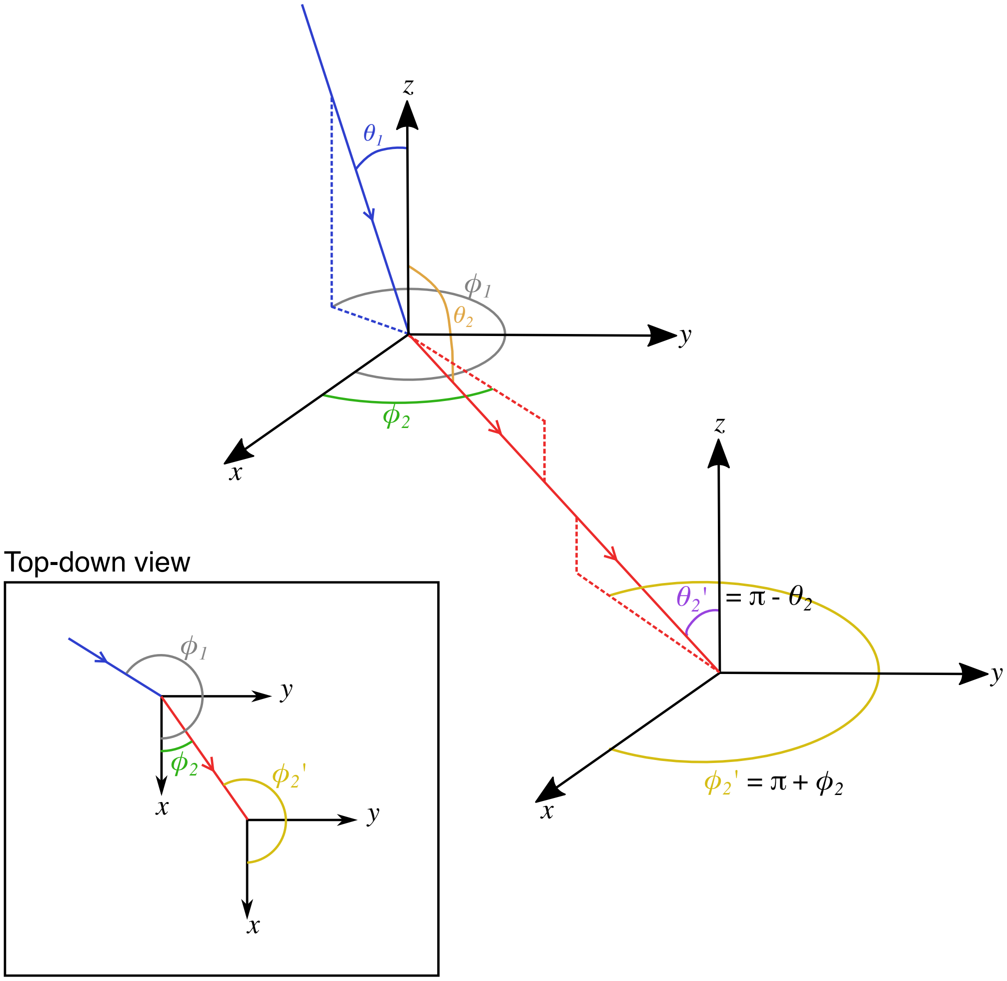

Figure 9: Schematic of the angle conventions used in RayFlare, and how they are transformed for incoming and outgoing light paths.

interaction with a surface (Fig. 9).[2]

Considering these definitions, it can be deduced that the **R** matrices describe light incident with some value of $\theta$ scattered so the outgoing direction (prior to a transformation into an ‘incoming’ vector) is in the same half-plane (so if $\theta < \pi/2$ before the interaction, it remains so after the interaction with the surface, and vice versa). The **T** matrices describe light where the value of $\theta$ is changed so the outgoing ray is travelling into the other half-plane. The attenuation of the intensity in the bulk, which in these simulations are assumed to be thick enough that interference effects within the bulk can be ignored, is described by matrix **D** using the Beer-Lambert absorption law:

[2] Applying this transformation often does not affect the results, as surfaces with mirror symmetry along the $x$ and $y$ axes, such as regular square-based pyramids or a square diffraction grating, are not affected by the transformations $\theta \rightarrow \pi - \theta$ and $\phi \rightarrow \pi + \phi$. However, implementing this transformation ensures the framework can be used for surfaces which do not exhibit such symmetry.

$$D = \begin{bmatrix} e^{-\alpha d/|\cos\theta_1|} & 0 & \dots & 0 \\ 0 & \ddots & & 0 \\ 0 & \vdots & \vdots & 0 \\ 0 & \dots & 0 & e^{-\alpha d/|\cos\theta_m|} \end{bmatrix}$$

The absolute value sign ensures that it does not matter whether the **D** matrix is applied before or after the transformation from outgoing to incoming direction, since $\cos(\pi - x) = -\cos(x)$. There may be multiple identical diagonal entries for each $\theta$, depending on how many corresponding $\phi$ channels there are. The matrix **D** has the same dimensions as **R** and **T**. Note that the $\mathbf{R_{b,1}}$ and $\mathbf{R_{f,2}}$ matrices are equivalent to the **B** and **C** matrices, respectively, in the OPTOS formalism [24]; the labelling of the **D** matrix in Fig. 8 is unchanged from OPTOS.

After each surface interaction or pass through the bulk of the structure, the power remaining as a fraction of the incident power can be calculated by adding up all the entries in $\vec{v}$ inside the structure:

$$P = \sum_{m=1}^{l} \vec{v}_m \tag{19}$$

Note that the sum here is over all the angular bins, of which there are $l$. The initial total power sums to one: $\sum_{m=1}^{l} (\vec{v_0})_m = 1$. With reference to Fig. 8 for the vector and matrix labelling, the absorption, reflection and transmission can be calculated iteratively as follows:

$$\begin{gathered} \text{First interaction with front surface:} \\ \begin{aligned} \vec{v}_{f\downarrow,1} &= \mathbf{T_{f,1}}\vec{v}_0 \\ \vec{v}_{r,1} &= \mathbf{R_{f,1}}\vec{v}_0 \\ \vec{a}_{1,1} &= \mathbf{A_{f,1}}\vec{v}_0 \end{aligned} \end{gathered} \tag{20}$$

$$\begin{gathered} \text{Iterative calculation:} \\ \begin{aligned} \vec{v}_{b\downarrow,i} &= \mathbf{D}\ \vec{v}_{f\downarrow,i} \\ \vec{a}_{2,i} &= \mathbf{A_{f,2}}\ \vec{v}_{b\downarrow,i} \\ \vec{v}_{t,i} &= \mathbf{T_{f,2}}\ \vec{v}_{b\downarrow,i} \\ \vec{v}_{b\uparrow,i} &= \mathbf{R_{f,2}}\ \vec{v}_{b\downarrow,i} \\ \vec{v}_{f\uparrow,i} &= \mathbf{D}\ v_{b\uparrow,i} \\ \vec{a}_{1,i+1} &= \mathbf{A_{b,1}}\ \vec{v}_{f\uparrow,i} \\ \vec{v}_{r,i+1} &= \mathbf{T_{b,1}}\ \vec{v}_{f\uparrow,i} \\ v_{f\downarrow,i+1} &= \mathbf{R_{b,1}}\ v_{f\uparrow,i} \\ &\vdots \end{aligned} \end{gathered} \tag{21}$$

This iterative calculation can be repeated until the power remaining in the vector $\vec{v}_{f/b}$ describing light travelling inside the structure is below a certain threshold defined by the user, i.e. all the light has been

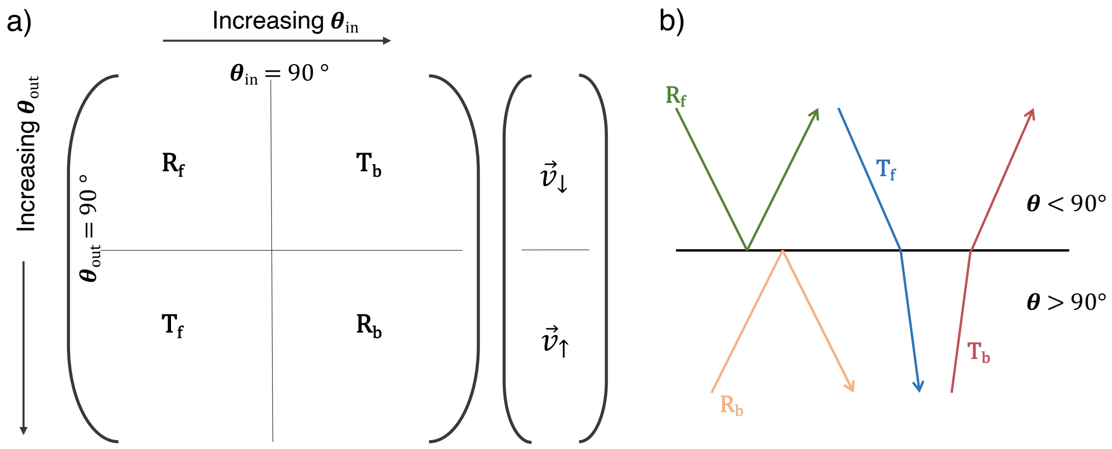
a)
Increasing $\boldsymbol{\theta}_{\text{in}}$
$\boldsymbol{\theta}_{\text{in}} = 90\,°$
Increasing $\boldsymbol{\theta}_{\text{out}}$
$\boldsymbol{\theta}_{\text{out}} = 90\,°$
$\mathbf{R_f}$
$\mathbf{T_b}$
$\mathbf{T_f}$
$\mathbf{R_b}$
$\vec{v}_\downarrow$
$\vec{v}_\uparrow$
b)
$\mathrm{R_f}$
$\mathrm{T_f}$
$\boldsymbol{\theta} < 90°$
$\boldsymbol{\theta} > 90°$
$\mathrm{T_b}$
$\mathrm{R_b}$

Figure 10: Schematic of a) the layout of the redistribution matrices for each surface and the angular vector $\vec{v}$ and b) the corresponding processes at the interface.

absorbed, transmitted, or reflected. At this point, all the relevant information will be stored in the $\vec{v}$ and $\vec{a}$ vectors. Although other values may be of interest, some of the most commonly-used quantities can be calculated as:

Calculating absorption in the bulk:

$$\begin{aligned}
A_{\downarrow,i} &= \sum_l \vec{v}_{f\downarrow,i} - \sum_l \vec{v}_{b\downarrow,i} \\
A_{\uparrow,i} &= \sum_l \vec{v}_{b\uparrow.i} - \sum_l \vec{v}_{f\uparrow,i} \\
A_{total} &= \sum_i (A_{\downarrow,i} + A_{\uparrow,i})
\end{aligned} \tag{22}$$

Calculating absorption per layer in the interfaces:

$$\begin{aligned}
\vec{a}_1 &= \sum_i \vec{a}_{1,i} \\
\vec{a}_2 &= \sum_i \vec{a}_{2,i}
\end{aligned}$$

Calculating reflection and transmission per angular bin:

$$\begin{aligned}
\mathbf{R}(\theta,\phi) &= \sum_i \vec{v}_{r,i} \\
\mathbf{T}(\theta,\phi) &= \sum_i \vec{v}_{t,i}
\end{aligned} \tag{23}$$

Calculating total reflection and transmission:

$$\begin{aligned}
R_{\text{total}} &= \sum_l \mathbf{R}\,(\theta,\phi) \\
T_{\text{total}} &= \sum_l \mathbf{T}\,(\theta,\phi)
\end{aligned}$$

The relationship of the different matrices $\mathbf{R_f}$, $\mathbf{T_b}$, $\mathbf{T_f}$ and $\mathbf{R_b}$ to values of $\theta$ for the ray before and after interacting with a surface are shown in Fig. 10. $\mathbf{R_f}$ and $\mathbf{T_f}$ affect only downward-travelling rays ($\theta_{in} < \pi/2$), while $\mathbf{R_b}$ and $\mathbf{T_b}$ affect only upward travelling rays ($\theta_{in} > \pi/2$). The $\mathbf{R}$ matrices will redistribute light into the same half-plane while the $\mathbf{T}$ matrices will redistribute light into the other half-plane. Computationally, this means that if full matrices and vectors with $0 \leq \theta \leq \pi$ are used, three-quarters of the matrices and half of the vectors $\vec{v}$ will, by definition, be empty during any given matrix multiplication. Thus, only the relevant non-zero parts of the matrices and vectors are actually multiplied. The matrices $\mathbf{R_{b,2}}$, $\mathbf{T_{b,2}}$ and $\mathbf{A_{b,2}}$, corresponding to incidence from the back on the rear surface of the cell, are not automatically computed, since it is assumed no light will enter the structure from the semi-infinite transmission medium.

The iterative nature of the calculations means that in addition to the values shown above, more complex metrics can be tracked, e.g. how much of the incident power is absorbed on each pass of the bulk; the effect of direct reflection $R_0$ (due to the first interaction of the incident light with the front surface of the cell) and escape reflection (due to light escaping into the incidence medium during subsequent interactions with the front surface); or how much light is absorbed in the front interface layers due to light incident from outside vs. inside the cell. Note that the dependence on wavelength of each of the matrices has not been made explicit so far to simplify the description of the model, but in fact the iterative calculation is carried out simultaneously for all wavelengths using three-dimensional arrays with dimensions wavelength, index of the incoming angles, and index of the outgoing angles or absorbing layers (the Python package xarray is used to deal with the high-dimensional arrays).

A key advantage of the matrix approach to optical modelling problems is that although changing either of the surfaces means that the relevant $\mathbf{R}$, $\mathbf{T}$ and $\mathbf{A}$ matrices must be recalculated using an appropriate method, repeating the simulation with a different bulk thickness requires very little additional simulation time as only matrix $\mathbf{D}$ has to be recalculated, which is computationally trivial. If only one of the surfaces is changed, only the matrices for that surface need to be recalculated, rather than for the whole structure. This is especially useful when comparing realistic structures with idealized textures (Section 2.7); comparing the performance of a realistic back reflector or scatterer with that of a perfect mirror does not require re-computing the behaviour of the whole structure, and is very fast because these matrices are trivial to calculate.

## 2.4 Absorption in individual layers

Including absorption in the surface textures, in addition to bulk absorption and reflection/transmission at the interfaces is a natural extension of the angular matrix framework; instead of light being redistributed into other angles, it can be redistributed into a vector which describes absorption in each layer. This extension was also developed for the OPTOS method [30], but was conceptualized and developed independently for the RayFlare framework.

The ray-tracing method, whether used with the Fresnel equations or a lookup-table with values calculated through TMM, is a Monte Carlo method and therefore inherently stochastic; rays can be reflected, absorbed, or transmitted with some probability, depending on the structure being considered, the wavelength, and the point of incidence on the surface. If lookup-tables for an interface with multiple layers are used, the total probability of the light being absorbed anywhere in the stack is found by adding up the fraction of incident light absorbed in each layer. It is not necessary to choose which layer the absorption takes place in stochastically, since these probabilities are calculated exactly from the TMM; the choice of reflection, transmission or absorption is made stochastically, but the distribution of absorbed photons between the layers can be done exactly based on the relative probabilities. Thus, if a ray is absorbed, RayFlare checks in the lookup-table what the probability of absorption per layer is (for the correct wavelength, side of approach, polarization and local incidence angle) and stores this information. At the end of the simulation, a matrix can then be generated relating the global incidence angle in terms of $\theta$ and $\phi$ bins to the probability of absorption in each layer; the intensity of absorbed rays is divided between the layers exactly based on the relative probability of absorption in that layer. However, because the absorption probabilities per layer are determined analytically using TMM while the fraction of rays reflected and transmitted are calculated stochastically, the situation can arise that (e.g. for a two-layer stack) $R + T + A_{\text{layer }1} + A_{\text{layer }2} \neq 1$. Thus, when generating the final matrices at the end of the simulation for each wavelength, the total number of rays which are not reflected or transmitted (i.e. must therefore have been absorbed) is stored, so that the fraction of absorbed rays $A$ is known for each global incidence angle. This is then used to scale the absorption fractions per layer, so that $\sum_{i=1}^{n_{\text{layers}}} A_i \equiv A$.

## 2.5 Absorption profiles

When using the combined TMM/ray-tracing approach, it is possible to calculate absorption profiles within the surface layers by considering the local incidence angle and using existing TMM methods to generate absorption profiles. The absorption profile at a depth $z$ in a coherent layer in a TMM calculation can be expressed analytically as [16]:

$$a(z) = A_1 e^{2z\,\text{Im}(k_z)} + A_2 e^{-2z\,\text{Im}(k_z)} + A_3 e^{2iz\,\text{Re}(k_z)} + A_3^* e^{-2iz\,\text{Re}(k_z)} \tag{24}$$

where $k_z = 2\pi n \cos\theta / \lambda_{vac}$ is the $z$-component of the wavevector $\vec{k}$. Eq. 24 can be found by calculating $-d(\mathbf{P} \cdot \hat{\mathbf{z}})/dz$, the negative of the derivative with respect to the $z$ position of the $z$-component of the time-averaged Poynting vector $\mathbf{P} = \frac{1}{2}\,\text{Re}\,[(\mathbf{E}^* \times \mathbf{H})]$. Physically, $A_1$ describes the intensity of the backward-travelling wave in the layer, $A_2$ describes the intensity of the forward-travelling wave, and $A_3$ describes interference between these waves. $A_1$, $A_2$ and $A_3$ depend on the layer structure under consideration and the wavelength. Note that if we consider a structure in which interference can be ignored, the expression for $a(z)$ at normal incidence simplifies to $a(z) = A_2 e^{-2z(2\pi\kappa)/\lambda_{vac}} = A_2 e^{-\alpha z}$, which is the negative of the first derivative with respect to $z$ of the Beer-Lambert law ($I = I_0 e^{-\alpha z}$ with $A_2 = \alpha I_0$), as expected.

When the TMM lookup-tables are generated, the relevant parameters ($A_1$, $A_2$, $A_3$, and $k_z$) used to calculate the absorption profile for light incident on the layer stack from a specific (local) angle are stored along with the reflection, transmission, and absorption probabilities. $A_1$ and $A_2$ are, by definition, real, while $k_z$ and $A_3$ are generally complex numbers for absorbing layers. During the ray-tracing process, the local incidence angle for each ray which gets absorbed must be stored if the absorption profile is to be calculated, so the correct values of the absorption profile parameters can be used (the difference between the overall angle of incidence on the surface and the local incidence angle is shown in Fig. 4b).

There is an added complication related to the issue discussed above for calculating absorption per layer in the combined TMM/ray-tracing approach; since the absorption fraction per layer has to be scaled to ensure that $R + A + T = 1$ for each global incidence angle, it follows that $A_1$, $A_2$ and $A_3$ must also be scaled accordingly. In addition, the data in the R, A and T redistribution matrices is in terms of global incidence angle, while the absorption profile depends on the local incidence angle. Thus, it is necessary to know the relationship between the global incidence angle and the local incidence angle for rays which were absorbed. Therefore, similar to the scaling to account for previous absorption, we must also scale the absorption profiles to account for how much of a contribution each local angle makes. For regular textures like pyramids, clearly a global incidence angle will relate to only a small number of local incidence angles.

Although the intensity incident on the surface at each interaction can only be determined during the full matrix calculation, the relationship between the global and local incidence angle and thus the scaling required for $A_1$, $A_2$ and $A_3$ is known after ray-tracing, before the matrix calculation takes place. During the ray-tracing procedure, the relationship between the global incidence angle and the local incidence angle for rays which are absorbed is recorded in a matrix, with rows describing the local incidence angle bins and columns describing the global incident angle bin (in terms of both $\theta$ and $\phi$ as discussed above). Thus the dimension of this matrix is $(n_{theta\ bins}, n_{angle\ bins})$. The global/local angle matrix is used to scale $A_1$, $A_2$ and $A_3$ for each global incidence angle. Thus, we end up with a very large multi-dimensional matrix storing the values of $A_1$, $A_2$, $A_3$ and $k_z$ for each layer and for each local angle per global incidence angle. This can then be used to generate a depth-dependent profile by using equation 24 for each local angle with a non-zero contribution, and then adding all the local angle contributions at each wavelength. The final result is a matrix with the different $z$ positions along the rows, and the different global incidence angles along the columns.

Finally, this matrix is scaled so that the integrated total of all these contributions is consistent with the R and T probabilities. This final matrix can be used just like the R, T and A matrices; when multiplying a $\vec{v}$ vector, this gives the resulting absorption profile from that interaction with the surface. At the end of the matrix multiplication, all these contributions can be added up to come up with an overall absorption profile. Care must be taken to take into account how absorption profiles generated for front incidence and rear incidence are added together, since the light encounters the structure from the opposite direction, so the absorption profiles for rear incidence must be flipped.

## 2.6 Absorbing incidence media

In absorbing incidence media, the meaning of the incidence angle $\theta$ becomes somewhat ambiguous. This can be understood by considering the wavevector $\vec{k}$ of a plane wave. For a general plane wave (assuming without loss of generality that we are in the $x - z$ plane – see Fig. 9 for the angle convention):

$$\vec{k} = \frac{2\pi(n + i\kappa)}{\lambda}(\sin\tilde{\theta}, 0, \cos\tilde{\theta}) \tag{25}$$

Where the complex refractive index $\tilde{n}$ is expressed explicitly in terms of its real and imaginary parts. If $\kappa = 0$ (non-absorbing medium), the meaning of the angle $\theta$ is clear; it is the angle from the normal at which the light is travelling. But if we have $\kappa \neq 0$, this meaning becomes less clear. Specifically, $\tilde{n}\sin\tilde{\theta}$ will not be a real number if $\theta$ is a real number; this implies that there is dissipation of power in the direction parallel to the incident medium, since the $x$-component of the wavevector is not real, and therefore this cannot be an infinite plane wave. So $\tilde{n}\sin\tilde{\theta}$ must be real for a plane wave, and if $\tilde{n}$ is complex, this means $\sin\tilde{\theta}$ must be complex, and thus $\tilde{\theta}$ must be complex. This is equivalent to saying Snell's law ($n_0 \sin\theta_0 = n_1 \sin\theta_1$) holds even for interfaces where one or both of the media are absorbing. Although it is possible to deduce a relationship between the direction from which power is incident, a useful definition within the RayFlare matrix framework, and the complex angle $\tilde{\theta}$ (see Appendix A), this has not currently been implemented in RayFlare; instead, the much simpler assumption that $\kappa = 0$ for all incidence media has been made. This means that for light incident from inside the bulk structure, $\kappa = 0$ is set to zero. The justification for this simplification is as follows:

- To be treated correctly within the RayFlare framework, the 'bulk' layer must be thick relative to the wavelength.
- If the bulk is thick relative to the wavelength and has high $\kappa$, then a large fraction of the light entering the bulk will be absorbed before ever encountering the back surface; so while the high incidence-angle entries[3] in the redistribution matrix for the rear surface, and back of the front surface, may not be accurate, this will not affect the overall results because a negligible fraction of the light encounters those surfaces at the problematic wavelengths.
- If the bulk has low $\kappa$, meaning the light may make many passes through the bulk before being absorbed and the high incidence-angle entries in the matrix become relevant (e.g. in an indirect semiconductor or glass), the angle at which power is travelling and the angle $\theta$ will be very similar since $\kappa$ is close to zero.

Further work is needed to assess how much these assumptions affect modelling results, and whether implementing a more rigorous treatment of the complex angles as outlined in Appendix A is beneficial.

## 2.7 Ideal cases

To explore fundamental limits to light trapping or compare the performance of a device design with theory, it is useful to include a convenient way of generating matrices for several cases where the optical behaviour can be calculated analytically. Two cases used frequently in investigations of the performance of light-trapping structures have so far been included in RayFlare: a perfect mirror and an ideal Lambertian scatterer. The perfect mirror case is straightforward: light hitting the interface at a polar angle $\theta$ and azimuthal angle $\phi$ is scattered into outgoing direction $\{\theta, \phi + \pi\}$. 100% reflectivity at all wavelengths is assumed, so the redistribution matrix is identical for all wavelengths, with a single entry equal to '1' per row and column of the matrix, with the location of the '1' entry describing the change in angle.

A perfectly Lambertian scatterer is an ideal diffuse reflector, having the same radiance (radiant flux per unit projected solid angle) regardless of the viewing angle. As the viewing angle increases from the normal, the projected area of the emitting surface appears to decrease proportional to $\cos\theta$, and thus the radiant flux should also decrease in order to maintain the same radiant flux. The power reflected into a polar angle bin with angle $\theta$ and width $d\theta$ can be written as:

$$dP \propto \cos(\theta) d\theta$$

The azimuthal dependence, $\phi$, has been ignored; it is clear from the symmetry of the coordinate system and the requirement of an ideal diffuse reflector that the power should be divided evenly between the $\phi$ bins, which are of equal size for any given $\theta$. The total power (summing over $\phi$ bins), as a fraction of incoming power, scattered into the bin at $\theta$ with a width of $d\theta$ is thus given by $\cos\theta d\theta$, with the final matrix normalized so that each column (corresponding to each incoming direction) sums to 1.

## 2.8 Storing information

For many structures commonly used in solar cells, such as pyramids or gratings, light will be scattered into specific directions according to its incidence angle, rather than being scattered across all possible outgoing

[3] The discrepancy between the angle from which power can be taken to be incident and the (real part of) the complex angle $\theta$ increases at higher angles of incidence, and with higher values of $\kappa$, as shown in Fig. 22.

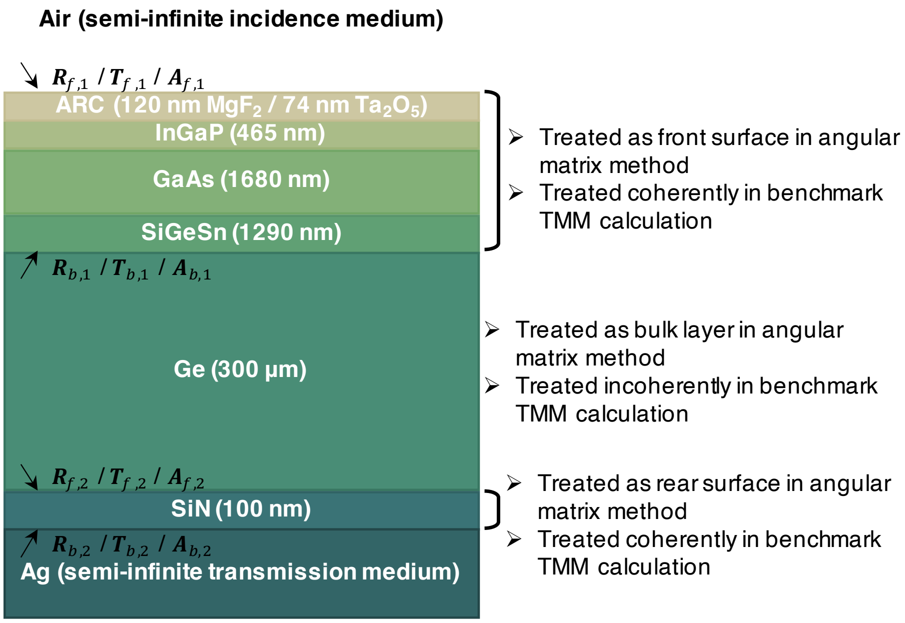
Air (semi-infinite incidence medium)
$R_{f,1}$ / $T_{f,1}$ / $A_{f,1}$
ARC (120 nm $MgF_2$ / 74 nm $Ta_2O_5$)
InGaP (465 nm)
GaAs (1680 nm)
SiGeSn (1290 nm)
➢ Treated as front surface in angular matrix method
➢ Treated coherently in benchmark TMM calculation
$R_{b,1}$ / $T_{b,1}$ / $A_{b,1}$
Ge (300 μm)
➢ Treated as bulk layer in angular matrix method
➢ Treated incoherently in benchmark TMM calculation
$R_{f,2}$ / $T_{f,2}$ / $A_{f,2}$
SiN (100 nm)
➢ Treated as rear surface in angular matrix method
➢ Treated coherently in benchmark TMM calculation
$R_{b,2}$ / $T_{b,2}$ / $A_{b,2}$
Ag (semi-infinite transmission medium)

Figure 11: The four-junction solar cell optical structure used for validation of the RayFlare angular matrix model. The treatment of the layers in both RayFlare and the TMM reference calculation, done using Solcore's existing TMM implementation, is indicated. The labels $R_{f,1}$, $T_{f,1}$ etc. refer to the labels used in Fig. 8 for the angular matrices.

directions (although this is not the case for randomly scattering structures or perfect Lambertian scattering). This means that the redistribution matrices are often extremely sparse, i.e. the vast majority of matrix entries is equal to zero; thus the matrices are stored in a sparse matrix format using the Python package sparse. A sparse matrix format stores a list of locations at which the matrix *is not* equal to zero, and the values at those indices, rather than storing each matrix entry explicitly. This significantly reduces the storage space required since typically a large number of matrices are generated during a simulation over a range of wavelengths.

In the case of perfect Lambertian scattering, where all the matrix elements describing reflection are non-zero and the scattering is assumed to be the same at all wavelengths, only a single matrix is stored as opposed to one matrix per wavelength to save storage space. The largest files which need to be stored are those for the TMM lookup-tables (Section 2.2), and those storing the parameters for calculating absorption profiles (Section 2.5); these multi-dimensional matrices are stored in NetCDF format using the xarray package.

## 2.9 Matrix method validation

In order to confirm if the matrix multiplication and methods for populating the redistribution matrices are working as expected, a planar multi-layer structure with a thick, incoherent layer can be considered. Such a structure can be modelled using the TMM method from Solcore directly, using the ability to treat layers incoherently, or by using the angular matrix framework with redistribution matrices calculated through TMM, ray-tracing in combination with a TMM lookup-table, or RCWA. As discussed previously, using ray-tracing or RCWA to calculate reflection, absorption and transmission through planar layers does not offer additional insight beyond that from a TMM calculation, but choosing a structure which can be modelled using all these methods allows a simple check of the angular binning method used in each, and the matrix

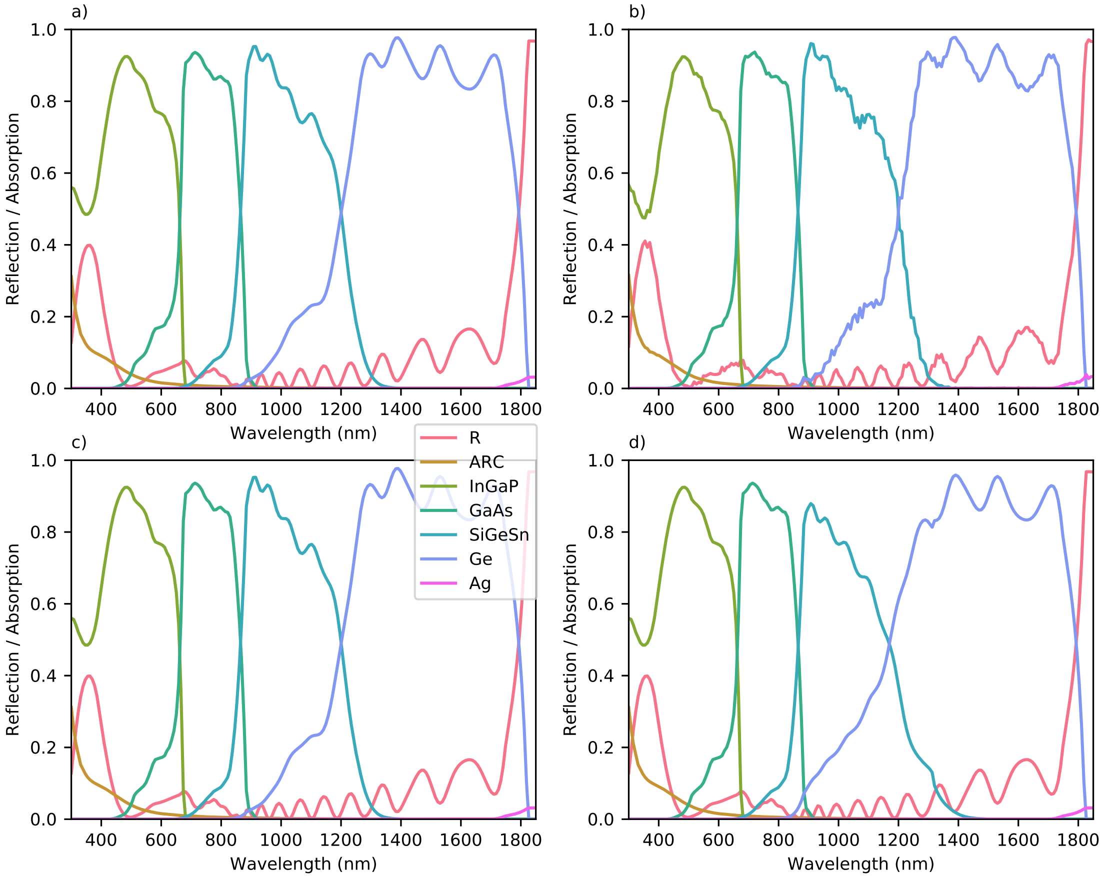

Figure 12: Calculated reflection, absorption per layer and transmission in a 4J solar cell-like optical structure for normally-incident unpolarized light, calculated using the RayFlare matrix multiplication framework and a) a TMM model, b) a ray-tracing model with TMM lookup-tables and c) an RCWA model to populate the redistribution matrices. d) shows the expected result, calculated using Solcore's TMM model (separate from RayFlare).

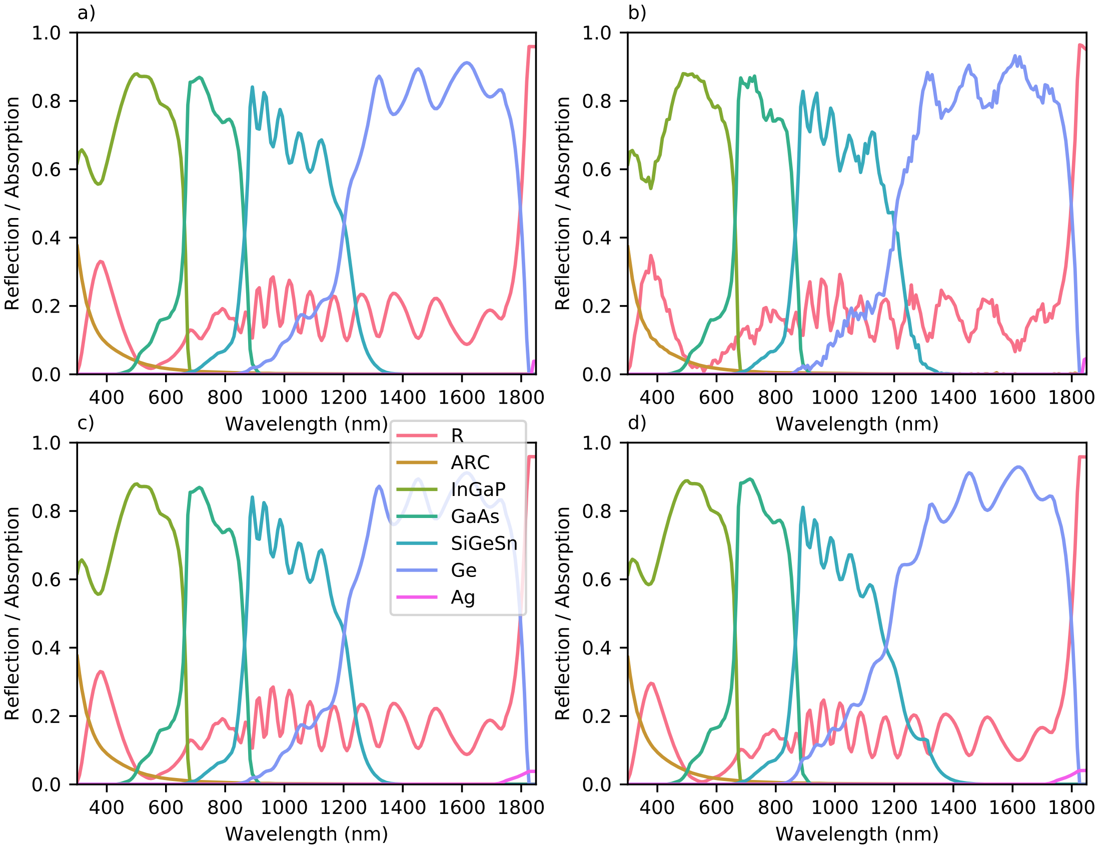

Figure 13: Calculated reflection, absorption per layer and transmission in a 4J solar cell-like optical structure with *p*-polarized light incident at 70° from the surface normal, calculated using the RayFlare matrix multiplication framework and a) a TMM model, b) a ray-tracing model with TMM lookup-tables and c) an RCWA model to populate the redistribution matrices. d) shows the expected result, calculated using Solcore's TMM model (separate from RayFlare).

multiplication itself.

The structure considered for the angular matrix model validation, and validation of the ray-tracing, TMM and RCWA methods as they are implemented to generate the redistribution matrices, is shown in Fig. 11. This structure is based on the optimized layer structure for a 4J solar cell with a SiGeSn subcell. Using Solcore's TMM method, this structure is simply a Solcore layer stack with the thick Ge layer treated incoherently. In the RayFlare matrix method, this Ge layer is the bulk medium described by the matrix $\mathbf{D}$ and the thinner layers ($MgF_2$, $Ta_2O_5$, GaInP, GaAs and SiGeSn) form the front interface (this is also shown in Fig. 11). When the TMM and RCWA methods are specified in RayFlare, the reflection, absorption and transmission of these three layers is calculated by treating them as a coherent stack with an infinite transmission medium made of Ge, and the matrix method is used to couple this behaviour to the back surface. If ray-tracing is specified, a lookup-table is first generated using TMM. Clearly, there is much redundancy between these methods, but the aim is to demonstrate the correct functioning of each method individually, and the connections between them. The optical behaviour of this structure was calculated at several angles of incidence, and for unpolarized, *s* and *p*-polarized light, using the four methods listed above. The comparison for normally incident, unpolarized light is shown in Fig. 12, showing that the results match

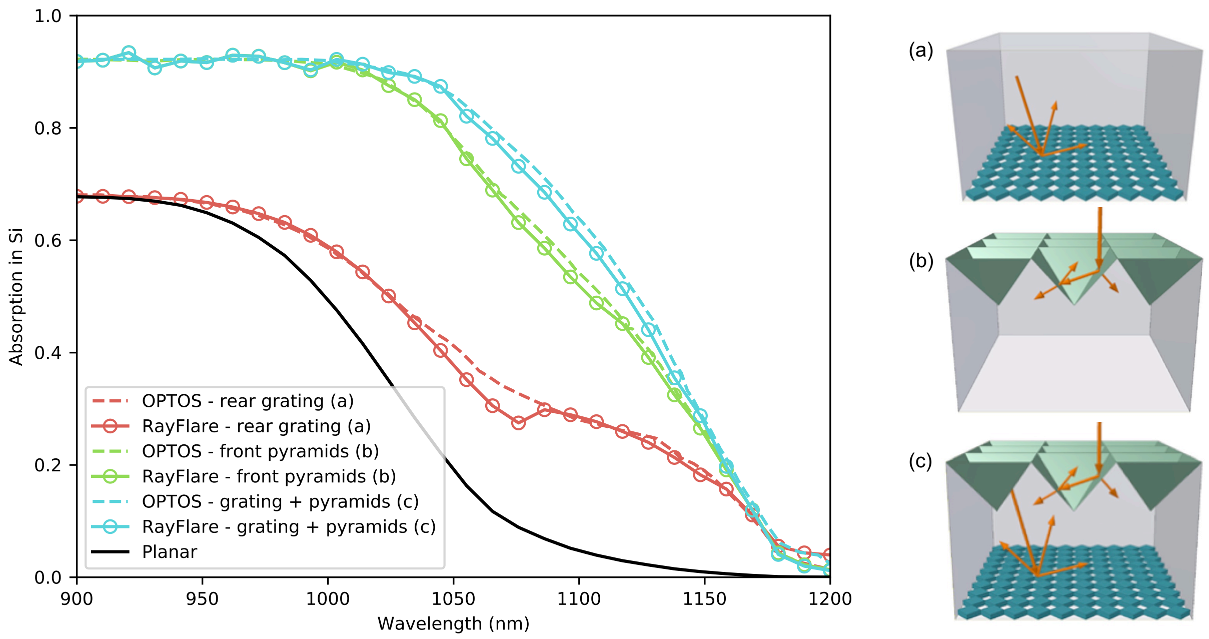

Figure 14: Comparison of absorption in 200 µm of bulk Si calculated using RayFlare (solid lines with open circles) and OPTOS (dashed lines) for (a) a structure with a planar front surface and a crossed diffraction grating at the rear (red lines), (b) a structure with inverted pyramids on the front of the structure and a planar rear surface (green lines) and (c) an Si layer with both structures, inverted pyramids on the front surface and a diffraction grating on the rear surface (blue lines). Schematics of the structures, reproduced from [11], are shown on the right.

extremely closely; the only deviation observed are the noisy oscillations in the result calculated through ray-tracing, due to the stochastic nature of this method. Results for $p$-polarized light incident at a glancing angle of 70° from the normal is shown in Fig. 13. Results for $s$-polarization and unpolarized light at non-zero incidence angles were also checked extensively to make sure the different methods were consistent.

To validate the performance of RayFlare for more complex structures, Fig. 14 compares the output of RayFlare with the results of OPTOS as reported in [11]. Three structures are considered, each with the bulk layer consisting of 200 µm of c-Si; (a) a structure with a planar front and crossed diffraction grating with period 1000 nm on the rear, (b) a structure with a planar rear surface and regular inverted pyramids (opening angle 55°) on the front and (c) a structure with pyramids on the front and a grating on the rear. The final structure uses the matrices for the front surface calculated for structure (a) and the matrix for the rear calculated for structure (b), so no additional time-consuming simulations are needed. There is a close match between the simulations in all cases. The dip in absorption in structure (a) around 1075 nm appears more pronounced in the RayFlare simulation, likely due to different RCWA methods being used with different numbers of Fourier orders (the number of orders used was not reported in the OPTOS paper). Fig. 15 shows the redistribution matrices (summarized over all the azimuthal angles $\phi$) for light incident from inside the structure on the front surface pyramids, and for the rear-side diffraction grating.

## 2.10 Application of integrated optical modelling

A useful application of the RayFlare framework is studying the effect of thin-film layers deposited conformally onto a surface texture, e.g. pyramids. An example of an application of such a structure in PV is a perovskite

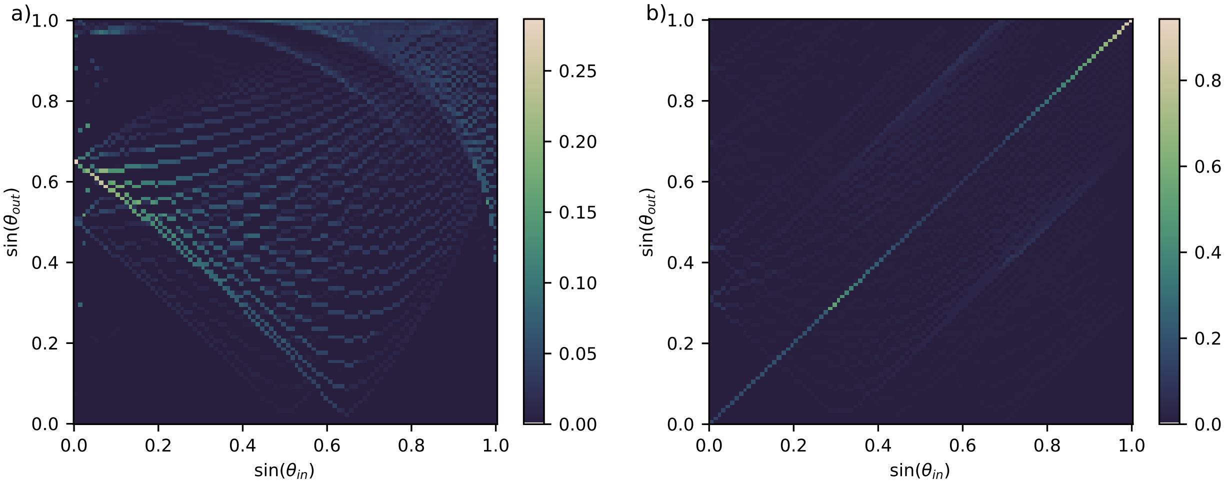

Figure 15: Reflection angular redistribution matrices, summarized per polar angle $\theta$, for light with vacuum wavelength 1100 nm incident on a) inverted pyramids and b) a crossed diffraction grating with period of 1 µm. In both cases, incidence from inside the cell structure (Si) is considered. Note that the pyramids are inverted when considered from the outside of the cell, so the rays traced will actually encounter upright pyramids when incident from the silicon.

Table 1: Layer structure of the perovskite/Si tandem cell. The layer thicknesses and materials are taken from [22]. The sources for optical constant data for each material is also given.

| **Material** | **Role** | **Thickness (nm)** | **Optical constant source** |
|---|---|---|---|
| $MgF_2$ | | 100 | [33] |
| IZO | | 110 | [34] $r(O_2) = 0.10\%$, annealed |
| $SnO_2$ | | 10 | $n = 2$ |
| $C_{60}$ | | 15 | [35] |
| LiF | | 1 | [28] |
| Perovskite | | 440 | [36]* |
| Spiro-TTB | | 12 | $n = 1.65$ |
| a-Si (n-type) | | 6.5 | measured$^{\dagger}$ [37] |
| a-Si (intrinsic) | | 6.5 | measured [37] |
| c-Si | | 260,000 | [32] |
| a-Si (intrinsic) | | 6.5 | measured [37] |
| a-Si (p-type) | | 6.5 | measured [37] |
| ITO | | 240 | measured [37] |
| Ag | | semi-infinite | [38] |

* = The optical constants for the perovskite were based on the data for the films deposited with a 10% evaporation rate of CsBr to $PbI_2$ and a ratio of 1:2 FAI to FABr in the spin-coated solution, designated as 'CsBr 10%, 1:2' in the source [36], which is listed as having a bandgap of 1.65 eV. The best cell performance in [22] was actually achieved with a CsBr/$PbI_2$ perovskite layer with a slightly lower bandgap of 1.6 eV. The optical constants used in the simulation were constructed by taking the optical constants of the 1.65 eV bandgap material and shifting the $n$ and $k$ values down by 0.05 eV over the full wavelength range.

$^{\dagger}$= measured through spectroscopic ellipsometry at Imperial by Dr. Diego Alonso Álvarez.

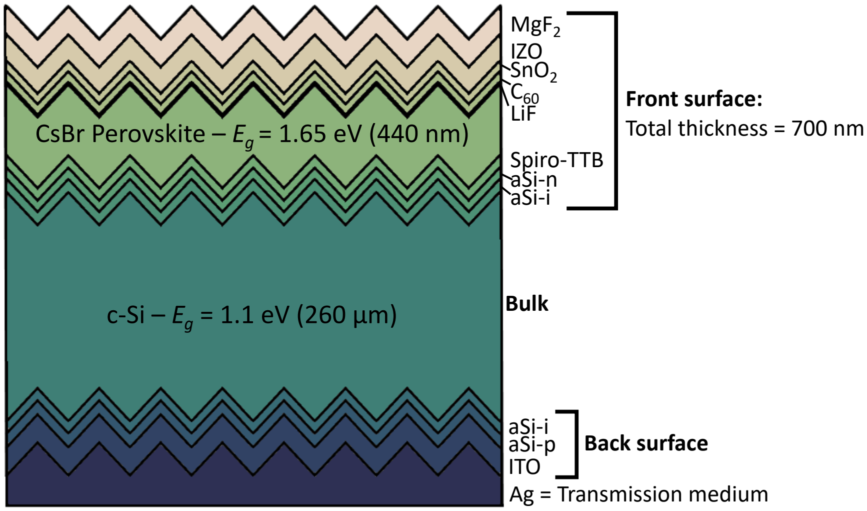

Figure 16: Structure of the perovskite/Si tandem cell structure. The incidence medium is air, with $n = 1$, while the semi-infinite transmission medium is silver. The front surface includes the perovskite, while the bulk material in RayFlare is the thick c-Si layer.

on silicon tandem cell, currently a device structure generating a large amount of research interest. One possible device structure involves depositing the relatively thin perovskite layer, and the necessary contacting and anti-reflection coating layers, conformally onto the pyramid structure of the Si, such as the device presented in Sahli et al. [22]. This preserves the well-optimized and effective use of pyramids to reduce front surface reflection and increase the path length of light in the silicon cell and increase absorption. The structure from [22] is shown in Fig. 16, with the details of the layer structure and the source of the optical constants used in these simulations shown in Table 1. The layer thicknesses in the stack are as reported in [22].

This structure was modelled using RayFlare's combined TMM/ray-tracing approach to calculate the redistribution matrices for each surface, with the Si as the bulk coupling medium across which the angular matrix method is applied. The overall result, with absorption in each layer, reflection, and transmission into the Ag substrate, is shown in Fig. 17. Examples of the matrices generated by RayFlare using the integrated ray-tracing/TMM lookup-table approach are shown in Fig. 18, showing the redistribution matrix for reflection and transmission of light incident from inside the cell on the front pyramidal surface. The TMM lookup-tables used during ray-tracing for the front surface are shown in Fig. 19, for light incident from outside the cell (left column) and inside the cell (right column). Fig. 19a shows that front surface reflection is low except at extremely glancing incidence angles[4], with some visible Fabry-Perot oscillations. In Fig. 19c and e we see low transmission and high absorption, respectively, at wavelengths below the bandgap of the perovskite (≈ 700 nm), where most incident light is absorbed in the surface layers, and high transmission into the Si bulk at wavelengths above the bandgap. For rear incidence, we can clearly see the effect of total internal reflection in Fig. 19d, showing no transmission out of the cell into air above incidence angle of ≈ 15° (with the exact value of the critical angle depending on the refractive index of the layers and thus the wavelength). Light incident from inside the cell can also be absorbed in the front surface layers (Fig. 19f).

[4] For normally incident light incident on a surface with regular pyramids with elevation angle 55°, the only possible local angle of incidence is 35°; however, because generating the lookup-tables is fast, the full angular space is considered.

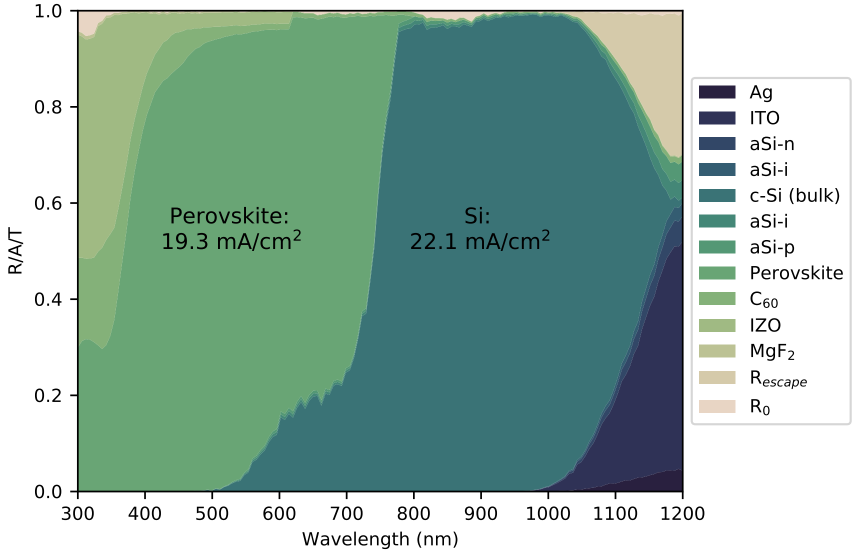

Figure 17: Reflection (direct and escape), absorption per layer, and transmission into the Ag substrate calculated using RayFlare for the perovskite/Si tandem cell structure shown in Fig. 16. The photogenerated currents are shown, calculated for an AM1.5G solar spectrum. The front and rear surfaces were described using the integrated TMM/ray-tracing framework.

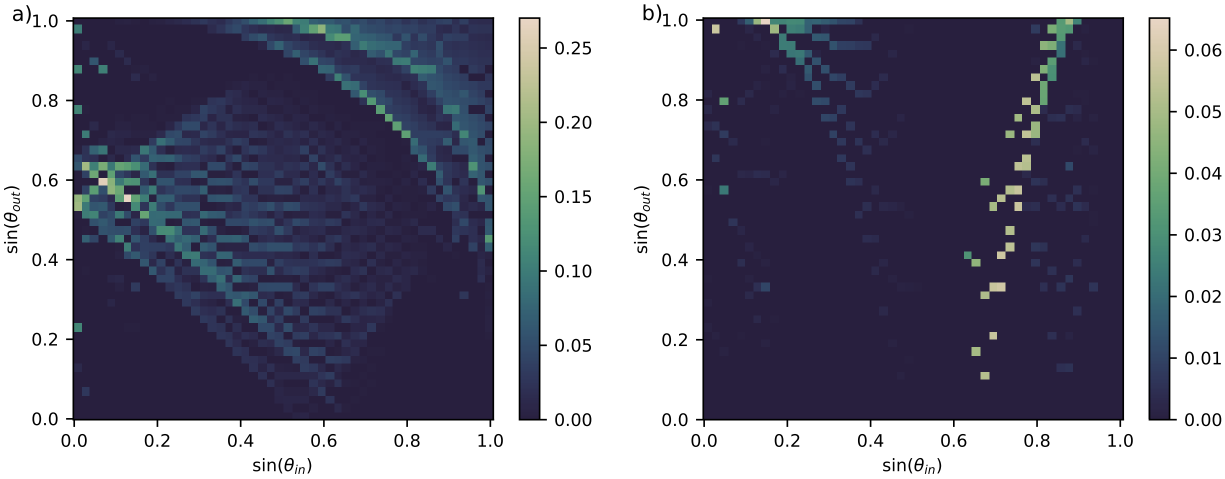

Figure 18: Examples of redistribution matrices calculated for the perovskite/Si tandem cell calculated using 100 incident/outgoing bins for the polar angle $\theta$ with azimuthal discretization $c_{azimuth} = 0.25$. These matrices describe a) reflection back into the cell and b) transmission into the incidence medium (air) for light which is incident from inside the Si on the pyramidal front surface. These matrices are summarized per $\theta$ bin (all the azimuthal $\phi$ bins are added together and normalized).

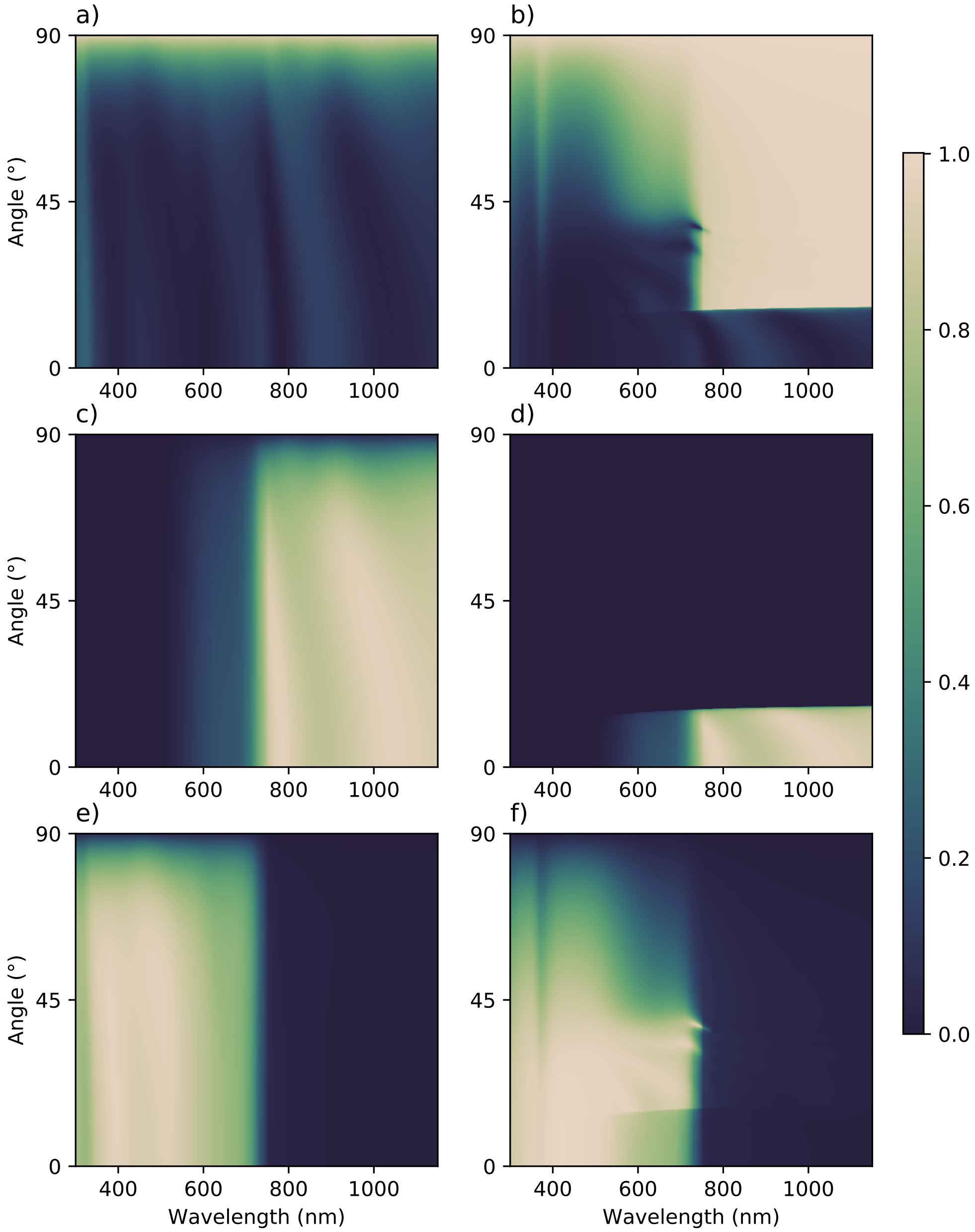

Figure 19: Example of the lookup-tables produced by RayFlare, in this case for the front surface of the perovskite/Si structure in Fig. 16; this information is used by the ray-tracing algorithm. The plots show incidence angle and wavelength dependence for the probability of: a) Reflection for front incidence (from air); b) reflection for rear incidence (from Si); c) transmission for front incidence (from air); d) transmission for rear incidence (from air); e) total absorption in the surface layers for front incidence; f) total absorption in the surface layers for rear incidence.

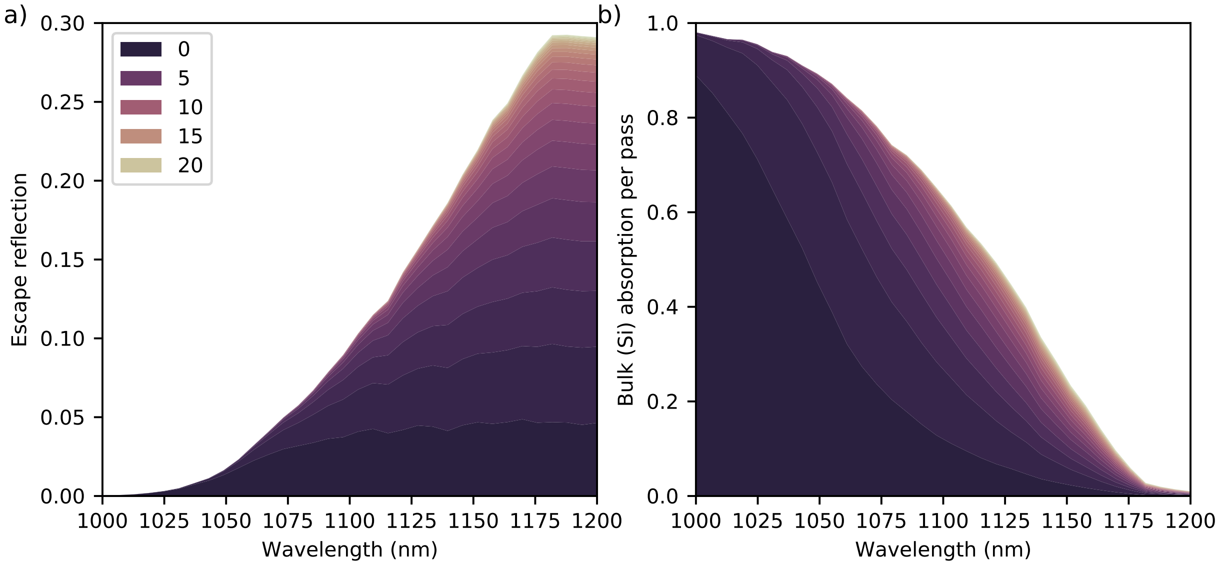

Figure 20: a) Escape reflection per interaction with the front surface and b) absorption in the bulk Si per pass for the perovskite/Si structure. The different colours represent the contribution of each pass/interaction, as shown in the legend.

Fig. 20 shows escape reflection (light transmitted through the front Si surface from inside the cell) and absorption in the bulk during the first 25 interactions with the surface (for the reflection) or passes through the cell (for the absorption). At short wavelength, all the light is absorbed before reaching the front surface again, escape reflection is low, and absorption in a single pass in the Si is high. As the wavelength increases, light is able to make more passes through the cell without being absorbed, leading to increased escape reflection through the front surface as the light has an increasing number of chances to be transmitted out of the cell as the number of interactions with the front surface increases close to the bandgap of Si. These values are calculated from the $\vec{v}$ vectors.

Fig. 21a shows the absorption profile in the front surface layers of the structure, calculated as outlined in Section 2.5, at three different incident wavelengths. At short wavelengths, there is considerable parasitic absorption in the IZO layer, and the absorption profile at the start of the perovskite layer is very sharp as $\alpha$ is high at short wavelengths. At an intermediate wavelength of 540 nm, closer to the bandgap of the perovskite layer, the absorption profile is more extended throughout the layer, and thin-film interference is clearly visible. At long wavelengths, almost no light is absorbed in the deposited layers, and light is instead able to reach the c-Si cell.

Fig. 21b shows the effect of replacing the textured back surface of the Si cell with a perfect mirror, or keeping the texture the same and increasing the Si thickness to 360 μm (from 280 μm), on the long-wavelength absorption in the Si bulk. A perfect mirror actually performs slightly worse than the realistic structure; this is because the real back surface is already a good reflector due to the silver deposited on the rear, and in addition randomizes the ray directions further so behaves more 'Lambertianly' than the perfect mirror, which reflects but does not scatter light into more oblique angles. As expected, increasing the c-Si thickness increases the absorption near the bandgap. Both of these results are extremely quick to calculate; the first case requires only the matrices for the rear surface to be changed to the trivial matrix for a perfect mirror (see Section 2.7), while changing the bulk thickness only requires the matrix **D** to be changed, which

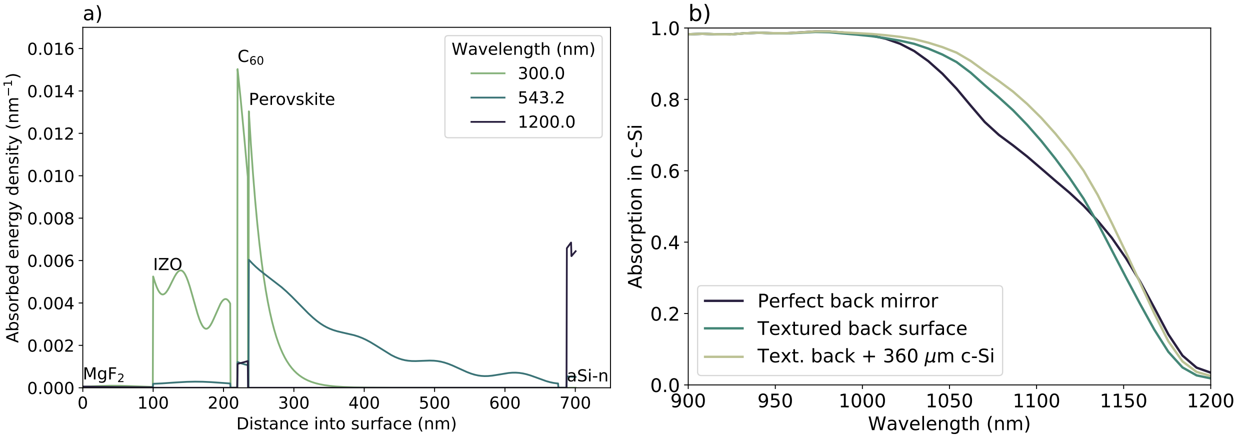

Figure 21: a) Absorption profile in the surface layers deposited on the pyramidal c-Si surface of the structure in Fig. 16 at three different wavelengths. b) Comparison of the absorption in Si at long wavelengths for the original textured back surface, a perfect mirror, or the original textured surface with an increase in the c-Si thickness.

uses only straightforward Beer-Lambert calculations.

## 2.11 Planned further work

Multiple relatively simple improvements to RayFlare could significantly decrease computation time. For ray-tracing, automatically checking the convergence as rays are traced and terminating the process at each wavelength once some sufficiently small standard deviation or confidence interval (adjustable by the user) has been reached prevents an unnecessarily large number of rays being traced for e.g. a simple interface or at wavelengths where the bulk is highly absorbing. For the matrix framework, populating the matrices can be time-consuming depending on the surfaces and methods chosen, and depending on the wavelength, some matrix entries may not be important; for instance, at wavelengths where the bulk material is very highly absorbing, the entries for the interaction with the rear surface, or for interaction with light encountering the front surface from inside the cell are not needed since no light will reach these surfaces at these wavelengths. Even for weaker bulk absorption, if the front surface scatters only into specific angles during the first interaction of light with the surface (e.g. a pyramidal surface), only the entries corresponding to angles into which the light will be scattered in the bulk need to be calculated for the rear surface redistribution matrices, as long as the light is able to make less than three passes through the cell (i.e. it does not reach the front surface again after interacting with the rear surface). If the rear surface is planar, this further reduces the number of matrix entries which have to be calculated. The ray-tracing and rigorous coupled-wave analysis modules, which can be slow due to their computational intensity, already support parallel implementation to calculate results for different wavelengths simultaneously on different threads; further improvements to the code speed can be made by running the computation using CPU (central processing unit) or GPU (graphics processing unit) acceleration, e.g. through the Python package numba.

RayFlare shares functionality with the optical modelling side of Solcore; in addition to the matrix framework, structures using any of the optical modelling methods (TMM, ray-tracing, ray-tracing with TMM or RCWA) for the full stack can also be defined. All these methods allow absorption profiles to be calculated, which allow a generation profile to be generated for Solcore's electrical solvers. An absorption profile can also be calculated using the matrix framework. Therefore, an easy way for RayFlare to interface with Solcore, and generate absorption profiles which can be used directly by Solcore's electrical solvers, will expand both the optical methods available for use with Solcore and allow direct use of the results of RayFlare in electrical models.

# References

[1] Juan C Minano. "Optical confinement in photovoltaics". In: *Physical Limitations to Photovoltaic Energy Conversion*. Bristol: Adam Hilger, 1990, pp. 50–83.

[2] A. Mellor et al. "Upper limits to absorption enhancement in thick solar cells using diffraction gratings". In: *Progress in Photovoltaics: Research and Applications* 19.6 (Sept. 2011), pp. 676–687. ISSN: 10627995. DOI: 10.1002/pip.1086. arXiv: 1303.4604.

[3] E. Yablonovitch and G.D. Cody. "Intensity enhancement in textured optical sheets for solar cells". In: *IEEE Transactions on Electron Devices* 29.2 (Feb. 1982), pp. 300–305. ISSN: 0018-9383. DOI: 10.1109/T-ED.1982.20700.

[4] Martin A. Green. "Lambertian light trapping in textured solar cells and light-emitting diodes: Analytical solutions". In: *Progress in Photovoltaics: Research and Applications* 10.4 (2002), pp. 235–241. ISSN: 10627995. DOI: 10.1002/pip.404.

[5] A. Mellor et al. "Interstitial light-trapping design for multi-junction solar cells". In: *Solar Energy Materials and Solar Cells* 159 (2017), pp. 212–218. ISSN: 09270248. DOI: 10.1016/j.solmat.2016.09.005.

[6] Corsin Battaglia et al. "Light Trapping in Solar Cells: Can Periodic Beat Random?" In: *ACS Nano* 6.3 (Mar. 2012), pp. 2790–2797. ISSN: 1936-0851. DOI: 10.1021/nn300287j.

[7] Kylie R. Catchpole et al. "Plasmonics and nanophotonics for photovoltaics". In: *MRS Bulletin* 36.6 (2011), pp. 461–467. ISSN: 08837694. DOI: 10.1557/mrs.2011.132.

[8] Martin A. Green and Supriya Pillai. "Harnessing plasmonics for solar cells". In: *Nature Photonics* 6.3 (2012), pp. 130–132. ISSN: 17494885. DOI: 10.1038/nphoton.2012.30.

[9] Patrick Campbell and Martin A. Green. "Light trapping properties of pyramidally textured surfaces". In: *Journal of Applied Physics* 62.1 (1987), pp. 243–249. ISSN: 00218979. DOI: 10.1063/1.339189.

[10] R. Santbergen and R.J.C. van Zolingen. "The absorption factor of crystalline silicon PV cells: A numerical and experimental study". In: *Solar Energy Materials and Solar Cells* 92.4 (Apr. 2008), pp. 432–444. ISSN: 09270248. DOI: 10.1016/j.solmat.2007.10.005.

[11] Nico Tucher et al. "3D optical simulation formalism OPTOS for textured silicon solar cells". In: *Optics Express* 23.24 (2015), A1720. ISSN: 1094-4087. DOI: 10.1364/OE.23.0A1720.

[12] Peter Y. Yu and Manuel Cardona. *Fundamentals of Semiconductors*. Graduate Texts in Physics. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. ISBN: 978-3-642-00709-5. DOI: 10.1007/978-3-642-00710-1.

[13] Jessica R. Piper and Shanhui Fan. "Broadband Absorption Enhancement in Solar Cells with an Atomically Thin Active Layer". In: *ACS Photonics* 3.4 (Apr. 2016), pp. 571–577. ISSN: 2330-4022. DOI: 10.1021/acsphotonics.5b00510.

[14] O. Isabella et al. "Advanced light trapping scheme in decoupled front and rear textured thin-film silicon solar cells". In: *Solar Energy* 162.October 2017 (2018), pp. 344–356. ISSN: 0038092X. DOI: 10.1016/j.solener.2018.01.040.

[15] Stéphane Collin and Maxime Giteau. "New limits for light-trapping with multi-resonant absorption". In: *2018 IEEE 7th World Conference on Photovoltaic Energy Conversion, WCPEC 2018 - A Joint Conference of 45th IEEE PVSC, 28th PVSEC and 34th EU PVSEC*. Institute of Electrical and Electronics Engineers Inc., Nov. 2018, pp. 3460–3462. ISBN: 9781538685297. DOI: 10.1109/PVSC.2018.8547477.

[16] Steven J. Byrnes. "Multilayer optical calculations". In: (Mar. 2016), pp. 1–20. arXiv: 1603.02720.

[17] K. C. Johnson. *GD-Calc_Intro.pdf*. 2005. URL: http://kjinnovation.com/GD-Calc.html.

[18] Liping Liu et al. "Heat dissipation performance of silicon solar cells by direct dielectric liquid immersion under intensified illuminations". In: *Solar Energy* 85.5 (May 2011), pp. 922–930. ISSN: 0038092X. DOI: 10.1016/j.solener.2011.02.007.

[19] Pochi Yeh. *Optical Waves in Layered Media*. John Wiley & Sons, 1988, pp. 308–316.

[20] Hung-Ling Chen et al. "A 19.9%-efficient ultrathin solar cell based on a 205-nm-thick GaAs absorber and a silver nanostructured back mirror". In: *Nature Energy* 4.September (2019). ISSN: 2058-7546. DOI: 10.1038/s41560-019-0434-y.

[21] Alexander Mellor et al. "Nanoimprinted diffraction gratings for crystalline silicon solar cells: implementation, characterization and simulation." In: *Optics express* 21 Suppl 2.March (2013), A295–304. ISSN: 1094-4087. DOI: 10.1364/OE.21.00A295.

[22] Florent Sahli et al. "Fully textured monolithic perovskite/silicon tandem solar cells with 25.2% power conversion efficiency". In: *Nature Materials* 17.9 (2018), pp. 820–826. ISSN: 14764660. DOI: 10.1038/s41563-018-0115-4.

[23] Rudi Santbergen et al. "GenPro4 Optical Model for Solar Cell Simulation and Its Application to Multijunction Solar Cells". In: *IEEE Journal of Photovoltaics* 7.3 (May 2017), pp. 919–926. ISSN: 2156-3381. DOI: 10.1109/JPHOTOV.2017.2669640.

[24] Johannes Eisenlohr et al. "Matrix formalism for light propagation and absorption in thick textured optical sheets". In: *Optics Express* 23.11 (2015), A502. ISSN: 1094-4087. DOI: 10.1364/OE.23.00A502.

[25] Yang Li et al. "Angular matrix framework for light trapping analysis of solar cells". In: *Optics Express* 23.24 (Nov. 2015), A1707. ISSN: 1094-4087. DOI: 10.1364/OE.23.0A1707.

[26] Benjamin Lipovšek, Janez Krč, and Marko Topič. "Optical model for thin-film photovoltaic devices with large surface textures at the front side". In: *Informacije MIDEM* 41.4 (2011), pp. 264–271. ISSN: 03529045.

[27] Phoebe Pearce and Nicholas Ekins-Daukes. "Open-source integrated optical modelling with RayFlare". In: *2019 IEEE 46th Photovoltaic Specialists Conference (PVSC)*. IEEE, June 2019, pp. 2627–2633. ISBN: 978-1-7281-0494-2. DOI: 10.1109/PVSC40753.2019.8980868.

[28] D. Alonso-Álvarez et al. "Solcore: a multi-scale, Python-based library for modelling solar cells and semiconductor materials". In: *Journal of Computational Electronics* 17.3 (Sept. 2018), pp. 1099–1123. ISSN: 1569-8025. DOI: 10.1007/s10825-018-1171-3. arXiv: 1709.06741.

[29] A Riverola et al. "Mid-infrared emissivity of crystalline silicon solar cells". In: *Solar Energy Material and Solar Cells* 174.July 2017 (2018), pp. 607–615. DOI: 10.1016/j.solmat.2017.10.002.

[30] Nico Tucher et al. “Optical modeling of structured silicon-based tandem solar cells and module stacks”. In: *Optics Express* 26.18 (2018), A761. ISSN: 1094-4087. DOI: 10.1364/OE.26.00A761.

[31] PV Lighthouse. *Wafer ray tracer*. URL: https://www2.pvlighthouse.com.au/calculators/wafer%20ray%20tracer/wafer%20ray%20tracer.html (visited on 05/14/2020).

[32] Martin A. Green. “Self-consistent optical parameters of intrinsic silicon at 300 K including temperature coefficients”. In: *Solar Energy Materials and Solar Cells* 92.11 (2008), pp. 1305–1310. ISSN: 09270248. DOI: 10.1016/j.solmat.2008.06.009.

[33] Luis V. Rodríguez-de Marcos et al. “Self-consistent optical constants of MgF2, LaF3, and CeF3 films”. In: *Optical Materials Express* 7.3 (Mar. 2017), p. 989. ISSN: 2159-3930. DOI: 10.1364/OME.7.000989.

[34] Monica Morales-Masis et al. “Low-Temperature High-Mobility Amorphous IZO for Silicon Heterojunction Solar Cells”. In: *IEEE Journal of Photovoltaics* 5.5 (2015), pp. 1340–1347. ISSN: 21563381. DOI: 10.1109/JPHOTOV.2015.2450993.

[35] S. L. Ren et al. “Ellipsometric determination of the optical constants of C 60 (Buckminsterfullerene) films”. In: *Applied Physics Letters* 59.21 (1991), pp. 2678–2680. ISSN: 00036951. DOI: 10.1063/1.105907.

[36] Jérémie Werner et al. “Complex Refractive Indices of Cesium-Formamidinium-Based Mixed-Halide Perovskites with Optical Band Gaps from 1.5 to 1.8 eV”. In: *ACS Energy Letters* 3.3 (2018), pp. 742–747. ISSN: 23808195. DOI: 10.1021/acsenergylett.8b00089.

[37] D. Alonso-Álvarez et al. “Thermal emissivity of silicon heterojunction solar cells”. In: *Solar Energy Materials and Solar Cells* 201.May (2019), p. 110051. ISSN: 09270248. DOI: 10.1016/j.solmat.2019.110051.

[38] Yajie Jiang, Supriya Pillai, and Martin A. Green. “Realistic Silver Optical Constants for Plasmonics”. In: *Scientific Reports* 6.1 (2016), p. 30605. ISSN: 2045-2322. DOI: 10.1038/srep30605.

# Appendices

## A Wavevectors and power flow in absorbing media

For a general plane wave (assuming without loss of generality that we are in the $x - z$ plane, where the incidence medium lies at $z = 0$):

$$\vec{k_f} = \frac{2\pi(n + i\kappa)}{\lambda}(\sin\tilde{\theta}, 0, \cos\tilde{\theta}) \tag{26}$$

If $k = 0$ (non-absorbing medium), this is fine. But if we have $k \neq 0$, strange things happen. Specifically, $\tilde{n}\sin\tilde{\theta}$ may not be a real number, which implies that there is dissipation of power in the direction parallel to the incident medium, i.e. not an infinite plane wave. So $\tilde{n}\sin\tilde{\theta}$ has to be real, and if $\tilde{n}$ is complex, this means $\sin\tilde{\theta}$ is complex, and thus $\tilde{\theta}$ is complex. This is equivalent to saying Snell's law ($\tilde{n}_0\sin\theta_0 = \tilde{n}_1\sin\theta_1$) holds even for interfaces between absorbing media.

Let's say we know the angle $\alpha$ from which *power*, i.e. the real part of the Poynting vector, is incident. Can we relate this to a complex angle $\tilde{\theta}$ for the wavevector $k_f$? The time-averaged Poynting vector $\langle\vec{S}\rangle$ can be related to the electric and magnetic fields through[5]:

$$\begin{aligned}
&\vec{k_f} = \tfrac{2\pi\tilde{n}}{\lambda}(\sin\tilde{\theta}, 0, \cos\tilde{\theta})\\
&\vec{E_f} = (0, E_f, 0)\\
&\vec{H_f} \propto \vec{k_f} \times \vec{E_f}\\
&\vec{H_f} \propto (-\cos\tilde{\theta}, 0, \sin\tilde{\theta})\\
&\langle\vec{S}\rangle = \mathrm{Re}\{\vec{E_f} \times \vec{H_f^*}\}\\
&\langle\vec{S}\rangle \propto \mathrm{Re}\{(\tilde{n}\sin\tilde{\theta})^*, 0, (\tilde{n}\cos\tilde{\theta})^*\} = \mathrm{Re}\{\tilde{n}\sin\tilde{\theta}, 0, \tilde{n}\cos\tilde{\theta}\}
\end{aligned} \tag{27}$$

Since we are asserting that for the wave to be a plane wave, $\tilde{n}\sin\tilde{\theta}$ is real, this reduces to:

$$\langle\vec{S}\rangle \propto (\tilde{n}\sin\tilde{\theta}, 0, \mathrm{Re}\{\tilde{n}\cos\tilde{\theta}\}) \tag{28}$$

Where we have used notation as used by Byrnes [16] and the standard expression for real power from the Poynting vector. The last step in (27) uses the fact that the real part of a complex number is the same as the real part of its complex conjugate. So say that we know *power*, or rather, want to assert that power is incident from an angle $\alpha$, this angle is given by:

$$\tan\alpha = \frac{\tilde{n}\sin\tilde{\theta}}{\mathrm{Re}\{\tilde{n}\cos\tilde{\theta}\}} \tag{29}$$

Taking the real part of $\tilde{n}\sin\tilde{\theta}$ is not shown explicitly since, as discussed above, it must be real. Some notation:

$$\begin{aligned}
k_x &= \tilde{n}\sin\tilde{\theta}\\
k_{z,r} &= \mathrm{Re}\{\tilde{n}\cos\tilde{\theta}\}\\
k_{z,i} &= \mathrm{Im}\{\tilde{n}\cos\tilde{\theta}\}
\end{aligned} \tag{30}$$

Where the $k$ notation represents that these quantities are proportional to the vector entries of the

[5] The following is for *s*-polarization, but as it turns out, things work out the same for *p*-polarization (the fact that $\mathrm{Re}\{\vec{E_f} \times \vec{H_f^*}\} = \mathrm{Re}\{\vec{E_f^*} \times \vec{H_f}\}$ for the Poynting vector is useful in that case so the maths is the same).

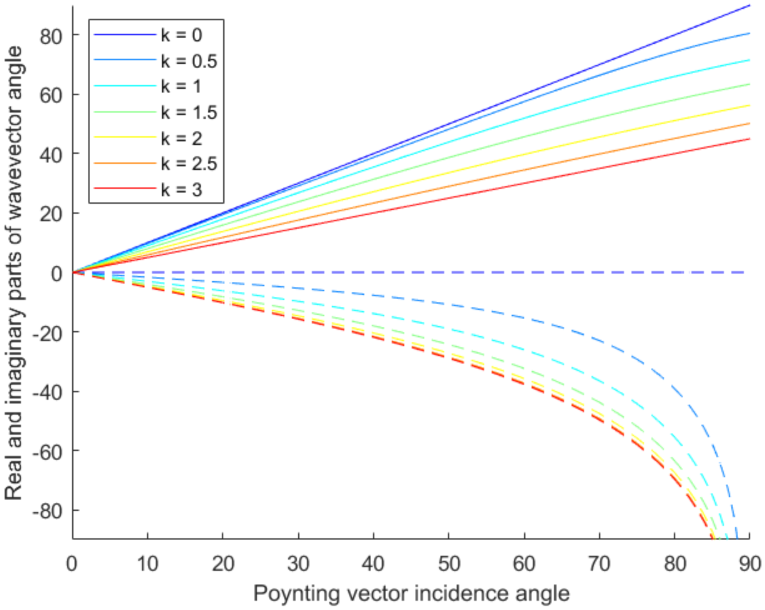

Figure 22: Variation of the real (solid lines) and imaginary (dashed lines) part of the complex wavevector angle $\tilde{\theta}$ where the Poynting vector is travelling at a certain incidence angle in a medium with refractive index $\tilde{n} = 3 + i\kappa$ with varying $\kappa$.

wavevector, and the last two lines are the real and imaginary parts of $k_z$. That means that:

$$\tan\alpha = \frac{k_x}{k_{z,r}} \tag{31}$$

Clearly, for the *complex* angle $\tilde{\theta}$ of the wavevector, we have (referring back to (27)):

$$\tan\tilde{\theta} = \frac{\tilde{n}\sin\tilde{\theta}}{\tilde{n}\cos\tilde{\theta}} = \frac{k_x}{k_z} = \frac{k_x}{k_{z,r} + ik_{z,i}} \tag{32}$$

Some identities we can use: since $\sin^2\theta + \cos^2\theta = 1$ (even for complex angles!), $k_x^2 + (k_{z,r} + ik_{z,i})^2 = \tilde{n}^2 = (n + i\kappa)^2$. Separating out real and imaginary parts gives us two equations:

$$\begin{aligned} &k_x^2 + k_{z,r}^2 - k_{z,i}^2 = n^2 - \kappa^2 \\ &k_{z,r}k_{z,i} = n\kappa \end{aligned} \tag{33}$$

So if we know $\alpha$ and want to find $\theta$, we should express $\tan\theta$ in terms of things we know ($n$, $\kappa$ and $\alpha$). I actually solved for $k_{z,r}$, since it is easy to then get $k_{z,i}$ and $k_x$ through the identities above.

Squaring (31) and plugging the first identity from (33) in to get rid of $k_x^2$:

$$\tan^2\alpha = \frac{k_x^2}{k_{z,r}^2} = \frac{n^2 - \kappa^2 - k_{z,r}^2 + k_{z,i}^2}{k_{z,r}^2} \tag{34}$$

Then getting rid of $k_{z,i}^2$ by substituting the second identity from (33):

$$\tan^2\alpha = \frac{n^2 - \kappa^2 - k_{z,r}^2 + \frac{n^2\kappa^2}{k_{z,r}^2}}{k_{z,r}^2} \tag{35}$$

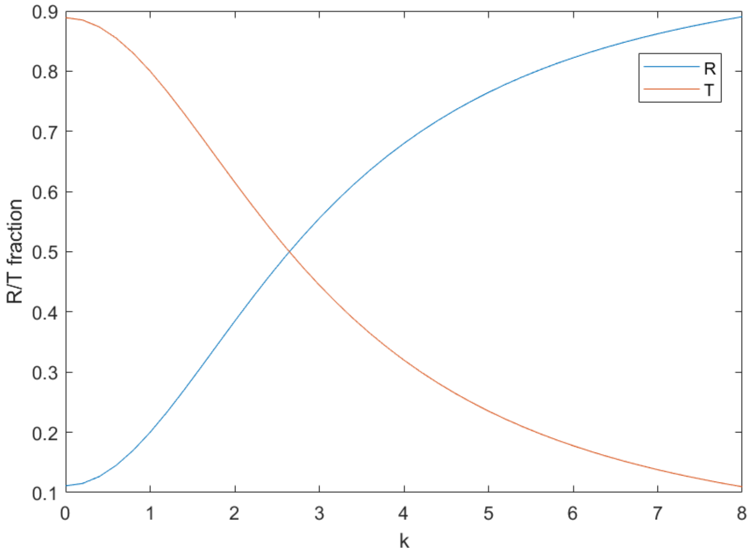

Figure 23: $R$ and $T$ for unpolarized (average of s and p-polarization) light incident from air ($n = 1$) on a medium with $n$ (real part of the refractive index) = 2 and varying $\kappa$.

Now by multiplying both sides by $k_{z,r}^4$ we get:

$$\tan^2 \alpha k_{z,r}^4 + (\kappa^2 - n^2)k_{z,r}^2 - n^2\kappa^2 = 0 \tag{36}$$

This is a quadratic equation for $x = k_{z,r}^2$. As it turns out, we have to take the positive root for the solution to be real. The answer is:

$$k_{z,r} = \cos\alpha\sqrt{\frac{1}{2}(n^2 - \kappa^2) + \frac{1}{2}\sqrt{((\kappa^2 - n^2)^2 + 4n^2\kappa^2 \sec^2\alpha}} \tag{37}$$

This can of course be used to find a direct expression for $\tilde{\theta}$, but if you have numerical inputs for $n$, $\kappa$ and $\alpha$, the quantities $k_{z,r}$ and thus $k_{z,i}$, $k_x$ and $\tilde{\theta}$ are all easy to calculate. Note that if $\kappa = 0$, this expression reduces to:

$$k_{z,r} = n\cos\alpha \tag{38}$$

as expected.

If you know complex $\tilde{\theta}$ but want to calculate $\alpha$, that is much more straightforward, as you can simply use (31) directly. The reason we want to use $\alpha$ in an angular matrix formalism is that we want to know about how incident power is distributed and absorbed throughout the cell, and we are binning in terms of the angular directions in which power travels. In many cases (i.e. media which are not strongly absorbing) these are equivalent to the wavevector directions, but in strongly absorbing media (e.g. metals, direct semiconductors or highly doped layers) this assumption may not be valid. Fig. 22 shows the difference between $\alpha$ and the real and imaginary parts of $\tilde{\theta}$ for increasing values of $\kappa$. Fig. 23 shows how $R$ and $T$ vary with increasing $\kappa$ of the transmission interface, showing that for highly absorbing media assuming the lossless case will give completely incorrect results. Fig. 24 shows a situation in which total internal reflection (TIR) occurs for a non-absorbing medium, and how $R$ is affected by increasing absorption in the transmission medium.

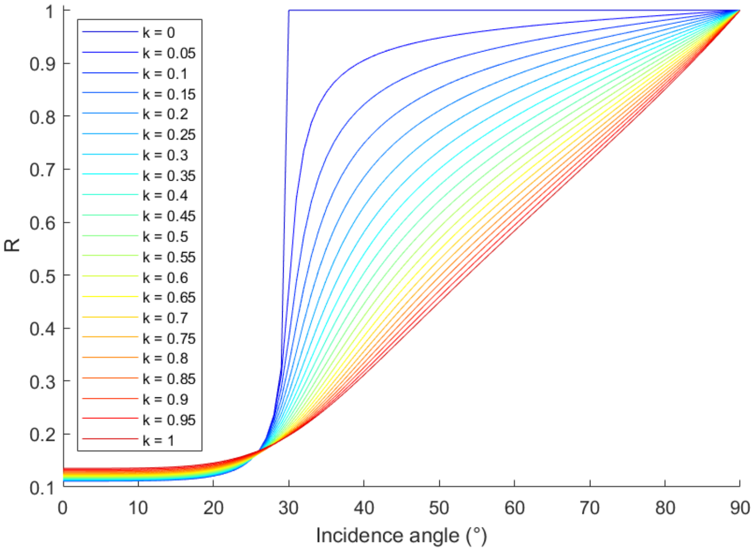

Figure 24: $R$ for light incident from a medium with $n = 4$ on a medium with $n = 2$ and varying imaginary part of the refractive index $\kappa$, at incidence angles $\theta$ between 0 and 90 degrees.

# B Convergence of ray-tracing calculations

Broadly speaking, increasing the number of rays used in the Monte Carlo ray-tracing algorithm will lead to more accurate results and longer computation times; however, there are multiple ways in which the number of rays can be increased. In order to accurately capture the behaviour of the surface, it is necessary to scan across an entire unit cell. For instance, if the number of rays is set at 2500 per wavelength, it is possible to trace 2500 rays incident on exactly the same point on the surface's unit cell, or 25 points on the surface 100 times, or 2500 points once. In addition, the scanned points can be chosen to cover the surface in a systematic way (e.g. by generating a grid of $n_x$ points in the $x$-direction and $n_y$ points in the $y$-direction and scanning the same number of points ($(n_x \times n_y)$ until the total number of rays is reached) or by generating a random position for each ray.

Fig. 25 shows how the number of surface points and total rays traced affects the absorption calculated, using a regular grid to generate the $(x, y)$ position at which the incident ray will strike the unit cell. The structure used is 200 µm of Si with regular inverted pyramids with size 5 µm on the front surface and a planar rear surface. The absorption in the Si was calculated at 300-1200 nm at 20 nm intervals. The RMS difference, summed over all wavelengths, in Fig. 25a is calculated with respect to the result with 50000 rays and 50 $x/y$ points ($50^2 = 2500$ surface points), the largest number of rays and surface points. The fraction of light absorbed in the Si, weighted by the absorption coefficient $\alpha$ at that wavelength, is shown in Fig. 25b. This weighting is used since as the Si becomes more transparent, more rays will have multiple interactions with the interfaces, meaning any systematic deviation in the calculated result should increase, thus weighting these points more heavily will emphasize systematic differences. The RMS difference is sensitive to both systematic deviations between different calculations, and fluctuations between adjacent wavelengths points due to the stochastic nature of the simulation, while the weighted absorption should be less sensitive to the stochastic fluctuations. It is clear from both figures that calculations with a relatively small number of surface points deviate strongly in both metrics, while increasing the number of surface points leads to convergence. Increasing the number of rays decreases the RMS difference, though the additional benefit after 20,000 rays or so is quite minimal. Increasing the number of rays has a much smaller effect on the weighted absorption, indicating that, as expected, the overall trend is correct even for a smaller number of rays but increasing the number of rays decreases the size of random fluctuations due to the stochastic nature of the calculations. Both metrics indicate that in general, increasing the number of surface points is a much more efficient way of obtaining accurate result than only increasing the number of rays.

Fig. 26 shows the full absorption results for several combinations of the number of rays and surface points. The result with a single $x/y$ point (not shown in Fig. 25 due to its large offset making it difficult to see the rest of the results) is extremely far off, predicting a much higher A; this is expected, since the single ray strikes in the centre of the unit cell, at the apex of pyramid, meaning the ray will generally be reflected many times within this pyramid and have a very well-randomized (Lambertian-like) path inside the cell compared to a ray striking the pyramid closer to its base. The other results show convergence appears to happen quickly, and the number of surface points (different line colors) is more important for convergence than the number of rays (different line types).

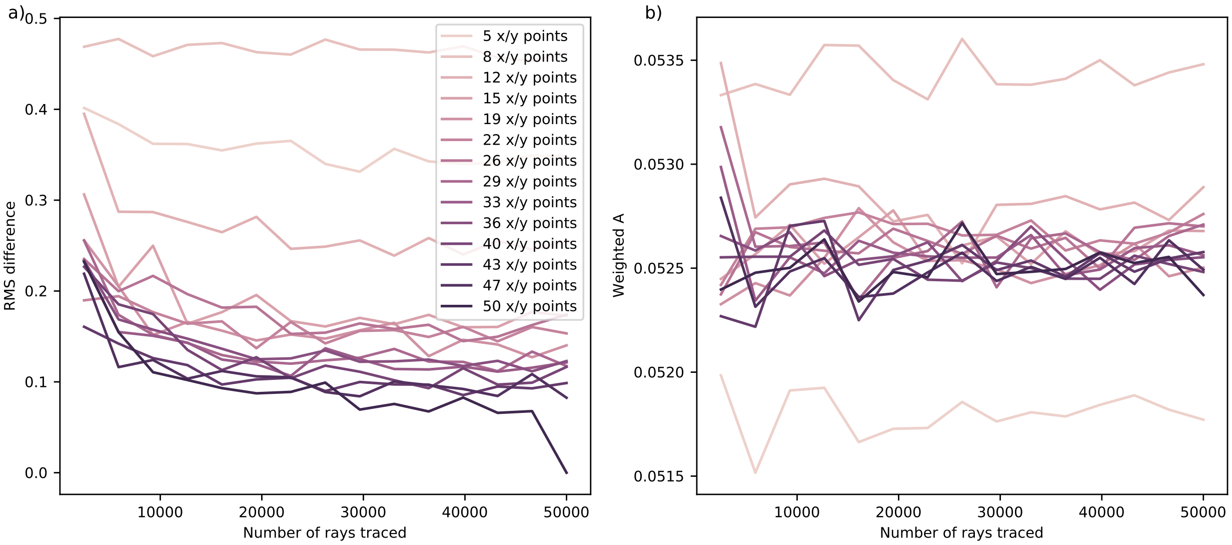

Figure 25: a) Summary of the RMS difference in the calculated absorption, summed over wavelength values shown in Fig. 26, for different numbers of surface points and total number of rays traced. The reference values (RMS error = 0) were taken from the simulation with the maximum number of rays and surface points (50,000 rays and 50 $x$ and $y$ surface points, 2500 surface points total). b) The fraction of light absorbed in the Si, weighted by the absorption coefficient of Si, depending on the number of rays and surface points traced.

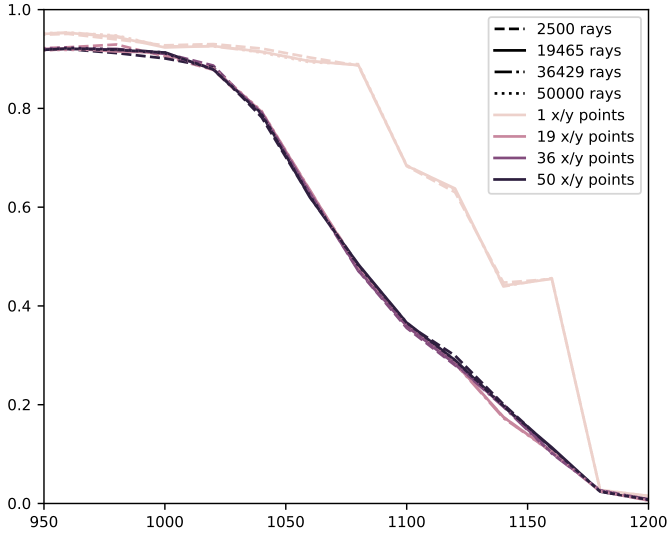

Figure 26: Absorption vs. wavelength for a 200 µm Si wafer with inverted, regular pyramids on the front surface and a planar rear surface using different numbers of rays and $x$ and $y$ scanning the surface (e.g. for 36 $x/y$ points a total of $36^2 = 1296$ surface points were scanned.