Non-imaging Optics for LED-Lighting

Jan Bouwe van den Berg (VU University), Rui Castro (Eindhoven University of Technology), Jan Draisma (Eindhoven University of Technology), Joep Evers (Eindhoven University of Technology), Maxim Hendriks (Eindhoven University of Technology), Oleh Krehel (Friedrich-Alexander-Universität Erlangen-Nürnberg), Ivan Kryven (University of Amsterdam), Karin Mora (University of Bath), Botond Szabó (Eindhoven University of Technology), Piotr Zwiernik (Eindhoven University of Technology)

Abstract

In this report, several methods are investigated to rapidly compute the light intensity function, either in the far field or on a finite-distance screen, of light emanating from a light fixture with a given shape. Different shapes are considered, namely polygonal and (piecewise) smooth. In the first case, analytic methods are sought to circumvent the use of Monte Carlo methods and ray-tracing with large sample size. In the second case, refinements of the Monte Carlo method (notably using a bootstrap procedure) are devised to minimize the number of samples needed for a good approximation of the intensity function.

KEYWORDS: optics, light, geometry, reflection, light ray, phase space, far field, polygon, ray-tracing, root approximation, Monte Carlo, bootstrap

1 Introduction

Philips Lighting is interested in assessing and improving the design of optical systems. These systems consist e.g. of lenses and mirrors and transfer light from a source to a target. To obtain characteristics of the light exiting the device (e.g. luminous intensity, color point and brightness), an efficient way of relating the input from the light source to the output at the target is needed. Currently, Monte Carlo methods are used which sample from the space of all possible rays that are emitted from the source. Using geometrical optics, it is possible to trace the path of each ray through the system. If a sufficiently large sample of rays is taken the corresponding distribution of rays over the target is representative for the output created by the device in reality. The words 'sufficiently large' are crucial here since in practice up to a million paths need to be traced to obtain reasonable results. In view of the fact that one wants to improve the design and redo the calculations step-by-step a less-involved procedure would be beneficial.

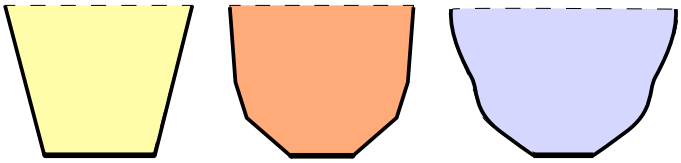


Figure 1: Different cup shapes in order of increasing difficulty: faceted (2 or more) and piecewise smooth.

The request to the Philips SWI 2012 group (here, formulated in its widest form) was to come up with a better (less time-consuming) approach. In view of the limited amount of time available, the problem owners indicated that they would not be disappointed if only the 2D case was considered. However, a short overview of further ideas on the 3D case would be highly appreciated.

During the week the team considered only the following:

- Devices consisting of mirrors. Lenses are not taken into consideration.
- Luminous intensity and no other photometric quantities (like color and brightness).

Moreover, the shapes of the light fixtures, also referred to as cups, can be subdivided in three classes, cf. Figure 1. Firstly, we consider two-faceted cups: devices with one flat wall at either side of the light source (Figure 1, left). Secondly, there are cups with piecewise linear walls, here called multi-faceted (Figure 1, center). Finally, the most general devices we consider are those whose sides are only required to be piecewise smooth (Figure 1, right). In order to assess the quality of the light bundle emitted from a lighting device, a screen is placed at some specific distance from, and parallel to the source. Each ray emitted from the source can be characterized by its position and angle. Once it arrives at the target again, it does so at a certain position and under a certain angle. For both the source and target, the position and angle constitute a two-dimensional phase space. Hence, the effect of the lighting device can be represented by a mapping from the source phase space, $\text{Phase}_{\text{source}}$, to the target phase space, $\text{Phase}_{\text{target}}$. Assume that X and Θ are the spaces of position and angle of a ray, respectively, when emitted from the source. Analogously, the position and angle at the target are in $\tilde{X}$ and $\tilde{\Theta}$, respectively. Under these assumptions, we have

$$\text{Phase}_{\text{source}} = X \times \Theta$$
$$\text{Phase}_{\text{target}} = \tilde{X} \times \tilde{\Theta}.$$

In practice ,i.e. in case of real, manufactured devices, either the target screen is at a relatively large distance from the device or one does not know before-

hand where the target ,e.g. a wall in the customer's home, will be. In such situations it makes sense to consider the so-called *far field* **F**, only. This far field can be considered as a target screen 'at infinity' where one only regards the angle-component of the target phase space and disregards the position-component, then

$$\mathbf{F} = \left(-\frac{\pi}{2}, \frac{\pi}{2}\right).$$

The further away a screen is from the source the better the far field approximation becomes.

The source emits light but it need not necessarily do so equally strong in all directions from all points. We model this by the intensity function $I_{\text{source}} : \text{Phase}_{\text{source}} \to \mathbf{R}_{\geq 0}$. Philips assumed

$$I_{\text{source}}(x, \theta) = \mathcal{I}_0 \cos\theta, \tag{1}$$

where $\mathcal{I}_0$ is a given constant. A source obeying this law is called *Lambertian*. Note that, in general, a suitable composition of I_{source} and the mapping $\text{Phase}_{\text{source}} \to \text{Phase}_{\text{target}}$ yields a corresponding function $I_{\text{target}} = I_{\text{target}}(\tilde{x}, \tilde{\theta})$ on the target phase space which contains the information on the luminous quantities we are interested in. Our goal is to compute this function efficiently. By integrating I_{target} over $\tilde{x}$, a quantity $I_{\mathbf{F}}$ called *intensity* is obtained, whereas the *illumination* follows from integrating over $\tilde{\theta}$.

2 Two-dimensional faceted cups

2.1 The 2-facet cup

One of the simplest shapes of a light fixture from the viewpoint of mathematical analysis is the 2-facet cup comprised only of straight lines. This axis-symmetric fixture consists of the base (light source) and two inclined edges which are 100% reflective and equally long.

Let us fix some notation as illustrated in Figure 2. A Cartesian coordinate system is set up by letting the origin O be the intersection of the lines through the reflective edges. The x-axis is parallel to the fixture's base and the y-axis divides the cup into two equal, symmetrical parts. Further, let a denote the distance of the base to the origin, let γ be the angle between the y-axis and each fixture edge, and let h be the height of the cup.

We assume an emitted light ray to be a straight line with initial position (x_0, a) where $x_0 \in (-a\tan\gamma, a\tan\gamma)$ and initial angle θ which is measured anti-clockwise to the y-axis. A light ray normal to the source, i.e. $\theta = 0$, corresponds to the direction $(0, 1)$.

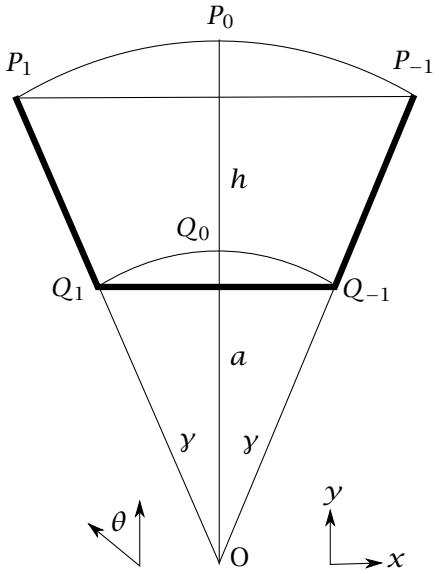


Figure 2: The 2-facet cup (bold lines) in our Cartesian reference frame.

Furthermore, we define the point $P_0 := (0, (a+h)/\cos\gamma)$ to lie on the y-axis and as far away from the origin as the highest point of the cup's edge. Similarly, $Q_0 := (0, a/\cos\gamma)$ has the same distance to the origin as the bottom corners of the cup.

In this particular case, we have:

$$\text{Phase}_{\text{source}} = X \times \Theta = [-a\tan\gamma, a\tan\gamma] \times \left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$$
$$\text{Phase}_{\text{target}} = \tilde{X} \times \tilde{\Theta} = [-(a+h)\tan\gamma, (a+h)\tan\gamma] \times \left(-\frac{\pi}{2}, \frac{\pi}{2}\right).$$

In the following, we address how to analytically determine the number of reflections the light ray undergoes before leaving the fixture. When the light ray meets the cup's wall we reflect the cup at that edge instead of the ray itself. Then the light ray, now a straight line, can be easily traced. This is illustrated in Figure 3 where the red and blue rays are reflected once and twice, respectively. This is an example cup that Philips suggested for study purposes.

For the general case we define the lines $P_{2k+1}Q_{2k+1}$ and $P_{2k-1}Q_{2k-1}$ to be the cup's edges after k reflections where

$$P_k := \begin{pmatrix} \cos\gamma & -\sin\gamma \\ \sin\gamma & \cos\gamma \end{pmatrix}^k P_0, \qquad \text{for } k \in \mathbf{Z}$$

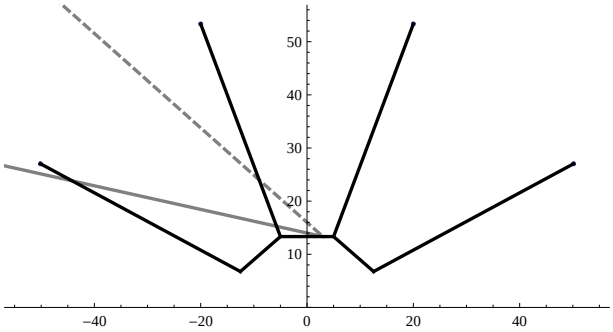


Figure 3: The 2-facet cup with reflections into its sides. Two emitted light rays are shown; the dashed ray with initial conditions (IC) $(x,\theta) = (3, 0.844)$ undergoing one reflection and the other with IC $(x,\theta) = (3, 1.352)$ undergoing two reflections. Cup parameters are $a = 40/3$, $\gamma = \arctan(3/8)$ and $h = 40$ (the Philips example).

are clockwise, $k < 0$, and anti-clockwise, $k > 0$, rotations of P_0 by an angle γ. A similar expression can be obtained for Q_k. Also, $P_{k,1}$ and $P_{k,2}$ are the x and y coordinates of P_k, respectively. The same notation is adopted for any other point, e.g. Q_k.

Then, it becomes obvious that for a fixed cup geometry there is a maximum number of reflections a ray can undergo before escaping the cup. This is the first number $j \in \mathbf{N}$ such that the y-coordinate of the point P_{2j+1} satisfies

$$P_{2j+1,2} \leq a$$

for $\theta \in (0, \pi/2)$. By symmetry we obtain a similar condition for initial ray angle $\theta \in (-\pi/2, 0)$ and y-coordinate of the top edge point $P_{2j-1,2}$ with $j < 0$.

We thus have a finite set $K \subset \mathbf{Z}$ that indexes regions in $\text{Phase}_{\text{source}}$ for which the rays show the same reflective behavior. Define $M_k \subseteq \text{Phase}_{\text{source}}$ $(k \in K)$ to be the set of rays that will undergo $|k|$ reflections before hitting the target where we define the first reflection with the left edge, P_1Q_1 to be positive, i.e. $k > 0$, and negative otherwise, i.e. $k < 0$. If $k = 0$, the ray will not hit any edge at all before reaching the target.

The set $K \subset \mathbf{Z}$ can be determined by solving $P_{k,2} > a$, resulting in

$$K = \left\{ -\left\lfloor \frac{1}{2\gamma} \arccos\left(\frac{a\cos\gamma}{a+h}\right) + \frac{1}{2} \right\rfloor, \ldots, \left\lfloor \frac{1}{2\gamma} \arccos\left(\frac{a\cos\gamma}{a+h}\right) + \frac{1}{2} \right\rfloor \right\}$$

where $\lfloor \cdot \rfloor$ denotes the floor function, giving the nearest smaller integer. Motivated by Figure 3, we calculate equations for the boundaries of M_k since points on such a boundary correspond to planar rays from (x, a) to P_{2k+1}. Considering a suitable triangle in the figure yields a linear equation for those boundaries in

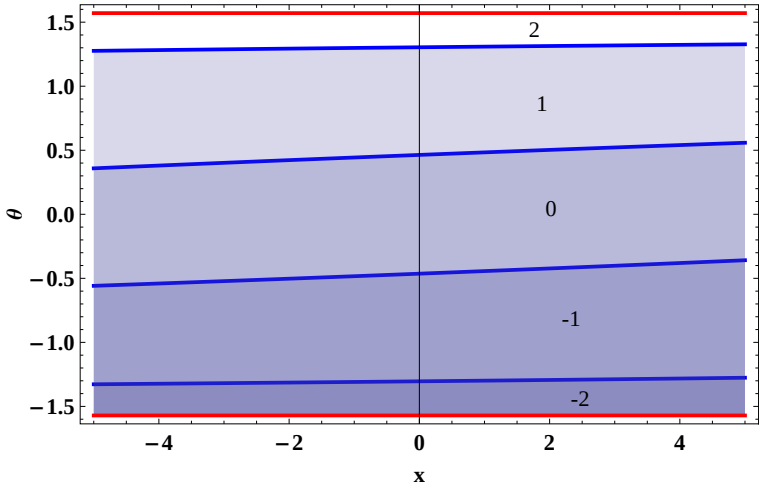


Figure 4: The division of $\text{Phase}_{\text{source}}$ into the regions M_k for $a = 40/3$, $\gamma = \arctan(3/8)$, $h = 40$ (the Philips example).

the coordinates $(x, \tan\theta)$

$$\tan(\theta) = -\frac{P_{2k+1,1} - x}{P_{2k+1,2} - a}.$$

Thus, the boundaries are simply straight lines in the $(x, \tan\theta)$-plane. The subdivision of the phase space $\text{Phase}_{\text{source}}$ into these regions is shown in Figure 4; note that the figure shows the (x, θ)-plane. In this specific case, the boundaries appear straight in these coordinates as well.

To compute the intensities I_{target} or $I_{\mathbf{F}}$ we have to re-trace a ray $(\tilde{x}, \tilde{\theta}) \in \text{Phase}_{\text{target}}$, respectively $\tilde{\theta} \in \mathbf{F}$, back to a set of rays (x, θ) in $\text{Phase}_{\text{source}}$. To do this, we first follow an emitted ray (x, θ). The reflection process is represented by a transformation $T : \text{Phase}_{\text{source}} \to \text{Phase}_{\text{target}}$ which we can describe in several parts, using the restrictions $T|_{M_k}$.

From Figure 5 we conclude that reflection in the positive, respectively negative edge result in the transformations

$$\begin{aligned} \text{Ref}_+ &: \theta \mapsto 2\gamma - \theta \\ \text{Ref}_- &: \theta \mapsto -2\gamma - \theta \end{aligned}$$

Since rays are alternatively reflected on both sides of the cup, if they are reflected at all, we derive, for a general ray in M_k,

$$\text{pr}_2 \circ T|_{M_k}(x, \theta) = (-1)^{k+1}(2k\gamma - \theta). \tag{2}$$

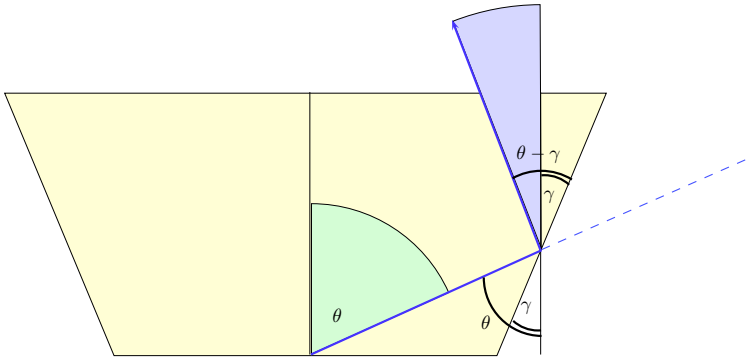
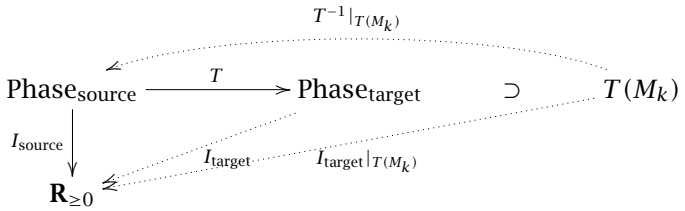


Figure 5: How the angle θ is affected by reflection.

Here, pr_2 denotes the function that projects a vector on its second component, in this case the $\tilde{\theta}$-component of $\mathrm{Phase}_{\mathrm{target}}$. This is a generic function that can be used on any space, and depends on the basis chosen for that space.
Note that it follows from (2) that parallel rays in the same subset M_k of $\mathrm{Phase}_{\mathrm{source}}$ will result in the same final $\tilde{\theta}$.
Now, we can formulate the problem of computing the intensity I_{target} in more detail. For a point $(\tilde{x}, \tilde{\theta}) \in \mathrm{Phase}_{\mathrm{target}}$ we can compose the inverse map T^{-1} with the source intensity. We say that we *push forward* the function I_{source} using T or that we *pull it back* using T^{-1}. To get explicit formulas we have to split up the computation over the different regions $T^{-1}|_{T(M_k)}$ for $k \in K$, since on each region T^{-1} has a different analytic description. This is visualized by the diagram



For simplicity, we concentrate for the moment only on the far field intensity $I_{\mathbf{F}}$.[1] Since, for any $\tilde{\theta}$ in the far field, multiple values of x could in principle have resulted in this final direction we have to integrate over these values on $\mathrm{Phase}_{\mathrm{target}}$ to find the intensity on $\mathbf{F}$, i.e.

$$I_{\mathbf{F}}(\tilde{\theta}) = \sum_{k \in K} \int_{x \in X} I_{\mathrm{source}}(x, (\mathrm{pr}_2 \circ T|_{M_k})^{-1}(\tilde{\theta})) \cdot \updownarrow_{(x, (\mathrm{pr}_2 \circ T|_{M_k})^{-1}(\tilde{\theta})) \in M_k} dx. \qquad (3)$$

[1] Later, we will see, however, that this might actually complicate the computations.

In the case of the 2-facet cup, the regions M_k yield (for each x) intervals with boundaries depending on k and θ for the pull-back integration regions

$$\{x \in X : (x, (\mathrm{pr}_2 \circ T|_{M_k})^{-1}(\tilde{\theta})) \in M_k\},$$

and the previous expression reduces to

$$I_{\mathbf{F}}(\tilde{\theta}) = \sum_{k \in K} \int_{x_{\min}(k,\tilde{\theta})}^{x_{\max}(k,\tilde{\theta})} I_{\mathrm{source}}(x, (\mathrm{pr}_2 \circ T|_{M_k})^{-1}(\tilde{\theta}))dx,$$

where $x_{\min}(k, \tilde{\theta})$ and $x_{\max}(k, \tilde{\theta})$ are the lower and upper bounds of the aforementioned intervals. Since the suggested Lambertian source intensity function (1) is in fact independent of x the above equation simplifies further

$$I_{\mathbf{F}}(\tilde{\theta}) = \mathcal{I}_0 \sum_{k \in K} \cos T_k^{-1}(\tilde{\theta}) \max\{0, x_{\min}(k, \tilde{\theta}) - x_{\max}(k, \tilde{\theta})\}$$

We can compute an analytic expression for this function given parameters (a, γ, h). Utilizing the formulae obtained above for the boundaries of M_k, we derive that

$$\begin{aligned} x_{\max}(\tilde{\theta}) &= \max\{-a\tan\gamma, (P_{2k+1,1} - (P_{2k+1,2} - a)\tan(-T^{-1}(\tilde{\theta}))\} \\ x_{\min}(\tilde{\theta}) &= \min\{a\tan\gamma, (P_{2k-1,1} - (P_{2k-1,2} - a)\tan(-T^{-1}(\tilde{\theta}))\} \end{aligned}$$

Figure 6 shows the intensity profile $I_{\mathbf{F}}$ computed for the Philips example.

To check for correctness of computations: the area under this curve is identical to the integral of the original intensity, corresponding to the total amount of emitted energy which should be conserved. For the given Lambertian source, this energy equals twice the width of the bottom of the cup:

$$E = \int_{\theta=-\frac{\pi}{2}}^{\theta=\frac{\pi}{2}} \int_{x=-a\tan\gamma}^{x=a\tan\gamma} I_0 \cos\theta dx d\theta = 2a\tan(\gamma)\mathcal{I}_0 \cdot \int_{\theta=-\frac{\pi}{2}}^{\theta=\frac{\pi}{2}} \cos\theta d\theta = 4a\tan(\gamma)\mathcal{I}_0.$$

Philips apparently chose their cup shape carefully. Not all cup shapes yield an intensity profile that looks as simple as the one in Figure 6. Specifically, it need not be unimodal even for the 2-facet case, for example see Figure 7 with parameters $a = 3$, $\gamma = \arctan(5/3)$ and $h = 10$ in the far field. We see a bright core appear straight in front of the fixture. Just left and right from the center of the graph there is a dip. This means that on a distant screen a relatively dark ring is present around the center of the illuminated zone which in turn is surrounded by a bright ring. We could call the presence of the dark ring the 'flashlight' or 'torch' effect; this phenomenon is often observed when using these devices even though the reflective sides of a flashlight are mostly not flat.

Instead of considering the intensity on the far field, we could also try to compute it on a nearby target phase space. The intensity on the far field can be obtained from it by integrating over x.

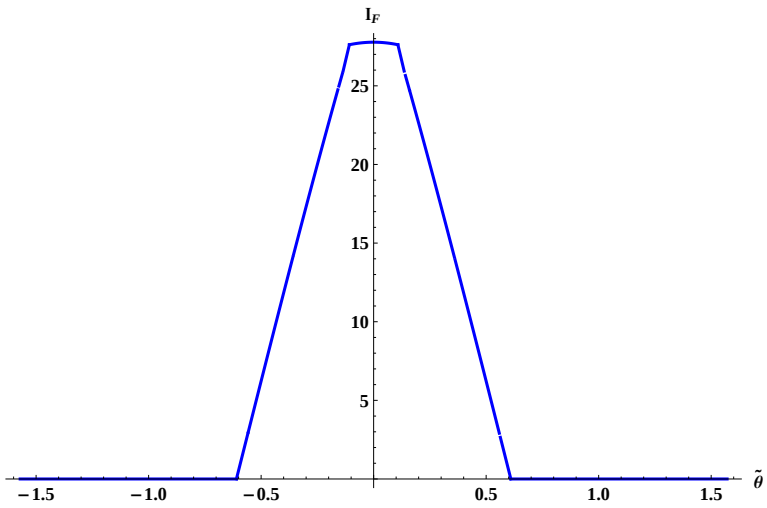


Figure 6: The intensity on **F** for $a = 40/3$, $\gamma = \arctan(3/8)$, $h = 40$ (the Philips example).

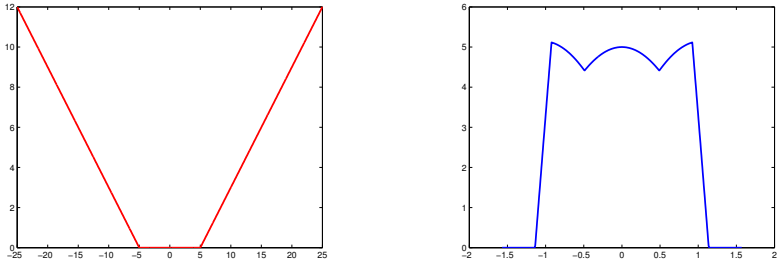


Figure 7: The cup and the intensity function $I_{\mathbf{F}}$ for $a = 3$, $\gamma = \arctan(5/3)$, $h = 10$.

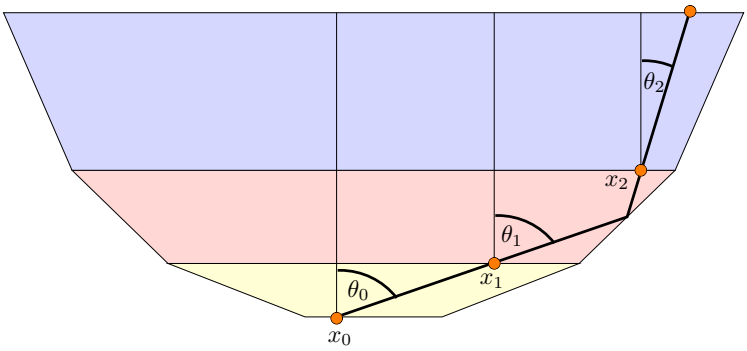
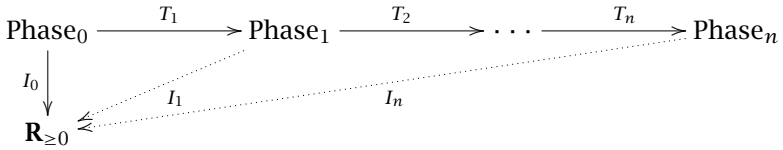


Figure 8: A multi-facet cup can be viewed as a stack of 2-facet cups.

This method allows us to study the symmetric (to the y-axis) multifaceted cup by subdividing the cup into identical subproblems, i.e. by viewing it as a stack of 2-facet cups. The light rays travel from the emitter to the top of the first cup, which is at the same time the bottom of the second. We can consider the continuation after that as a newly emitted ray from the bottom of this second level, see Figure 8. In total, this gives rise to a set of phase spaces

$$\text{Phase}_{\text{source}} =: \text{Phase}_0, \text{Phase}_1, \ldots, \text{Phase}_n := \text{Phase}_{\text{target}}\,.$$

Furthermore, we have a map $I_0 := I_{\text{source}} : \text{Phase}_0 \to \mathbf{R}_{\geq 0}$. Now, if we know the maps $T_i : \text{Phase}_{i-1} \to \text{Phase}_i$ then we also know the ras's path from the bottom of a 2-facet cup to the top and, more importantly, their inverses which means we can compute the intensity on $\text{Phase}_{\text{target}}$. Schematically, we want to push I_0 forward iteratively to the higher phase spaces:



The resulting intensity is

$$I_{\text{target}} = I_0 \circ T_1^{-1} \circ T_2^{-1} \circ \cdots \circ T_n^{-1}.$$

Several problems arise when one tries to implement this technique. Firstly, the maps T_i are not surjective: there are rays at the top of a 2-facet cup that could not have been emitted from the source irrespective of the number of reflections. A simple example one can think of is a ray exiting the top of the cup at the far left, with direction perpendicular to the left boundary. This example shows that the inverse map T_i^{-1} is not defined everywhere and we must keep track of its

domain. Secondly, the definition of T_i is piecewise, depending on the number of reflections, just as we saw above.

The description of T_i^{-1} on each piece where it is analytic is not very difficult. In order to deduce the formula for this map take a ray exiting the cup at $(\tilde{x}, a+h)$ in direction $\tilde{\theta}$. We draw a straight line downwards in direction $\tilde{\theta}+\pi$, retracing the light ray, see Figure 8. Assume that this straight line intersects the polygon $\ldots Q_{-1}Q_0Q_1\ldots$ in (x', y'), that the line segment $(\tilde{x}, a+h)(x', y')$ crosses k reflected sides of the cup and lies to the side of Q_1. The segments correspond to a ray emitted at (x, a) in direction θ that was reflected k times. We quickly deduce that

$$\begin{pmatrix} \cos\gamma & -\sin\gamma \\ \sin\gamma & \cos\gamma \end{pmatrix}^{2k} \begin{pmatrix} x \\ a \end{pmatrix} = \begin{pmatrix} x' \\ y' \end{pmatrix}$$
$$\tilde{x} - x' = -\tan\tilde{\theta}(a+h-y')$$

from which we compute

$$x = \frac{\tilde{x} + (a+h)\tan\tilde{\theta} + a\sin 2k\gamma - a\cos 2k\gamma\tan\tilde{\theta}}{\cos 2k\gamma + \sin 2k\gamma\tan\tilde{\theta}}.$$

Earlier, we already calculated $\mathrm{pr}_2 \circ T|_{M_k}$, and we see that

$$\theta = 2k\gamma + (-1)^k\tilde{\theta}.$$

This gives a complete description of $T^{-1}|_{T(M_k)}$. Due to time constraints we stop short of implementing this and producing a graph of I_{target} for the 2-facet cup.

2.2 Multifacet cups, following the beam backwards

Now, we describe an alternative, rather direct way to implement the computation for $I_{\mathbf{F}}$ of the multi-faceted cup by viewing such a cup as a stack of two-faceted cups.

Instead of starting at the source let us consider the part of the light emanating from the cup that travels in the direction of arbitrary angle $\tilde{\theta}$. Denote a whole interval of parallel rays a *beam*. Its width is the size of a perpendicular cross section of it. The goal is to determine the intensity of the light emitted in direction $\tilde{\theta}$, which corresponds to the width of the beam emanated in this direction. So we follow this beam back in time. Geometrically, we follow it down its path from top to bottom through the stack of cups. It enters the top cup on the line segment $[-(a+h)\tan\gamma, (a+h)\tan\gamma] \times \{a+h\}$. Inside the top cup the beam splits into several pieces (sub-beams), each characterized by the number of reflections in the top cup, where one also has to distinguish between the first reflection being a left or right reflection, as discussed before. These reflected

sub-beams reach the bottom $[-a\tan\gamma, a\tan\gamma]\times\{a\}$ of the top cup in intervals $[x_k,\bar{x}_k]\times\{a\}$ with angle $\theta_k=(-1)^{k+1}(2k\gamma-\tilde{\theta})$. Here k is an index running from $-\lfloor(\pi/2-\tilde{\theta})/(2\gamma)\rfloor$ to $\lfloor(\pi/2+\tilde{\theta})/(2\gamma)\rfloor$. Each interval $[x_k,\bar{x}_k]$ is determined by rotating the bottom of the cup over an angle $2k\gamma$, finding the intersection of this rotated line segment with the light beam, and rotating back. One also needs to take into account the number of reflections (odd or even).

Since the bottom of the upper cup is the top of the cup under it, we now repeat the above process in this next cup (with appropriate a, h and γ) for each of the sub-beams. We then continue inductively until we reach the bottom of the final (lowest) cup. This is where the light source is located, hence we can determine the contribution of the intensity in each sub-beam to the total intensity in the direction $\tilde{\theta}$.

This algorithm has been implemented recursively in Matlab. An example is shown in Figure 9.

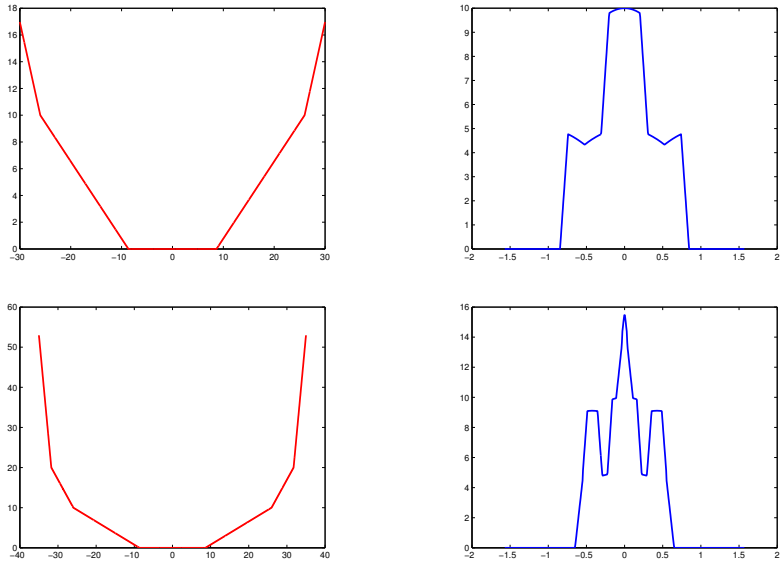


Figure 9: Top row A 4-facet cup with parameters $a_1=5$, $\gamma_1=\pi/3$, $h_1=10$, $a_2=22.32$, $\gamma_2=\pi/6$, $h_2=20$, and the resullting intensity $I_{\mathbf{F}}$. Bottom row A 6-facet cup with parameters $a_1=5$, $\gamma_1=\pi/3$, $h_1=10$, $a_2=22.32$, $\gamma_2=\pi/6$, $h_2=20$, $a_3=28.09$, $\gamma_3=\pi/32$, $h_3=30$, and the resulting intensity $I_{\mathbf{F}}$. These graphs were obtained using a recursive computation by following a beam of light downwards.

2.3 Polygonal cups

In this section we generalize some of the previous results for cup-shaped fixtures even more, namely to (still two-dimensional) polygonal fixtures. To this end, let $Q \subseteq \mathbf{R}^2$ be a two-dimensional convex polygon. To each edge e of Q we associate a phase space Phase_e whose points parameterize rays leaving e into the interior of Q. Formally, Phase_e is the Cartesian product $e \times \mathbf{R}$, where the first component records the point where the ray leaves e and the second component records the tangent of the angle that the ray makes with the inward pointing normal vector to e.

The rationale for choosing the tangent rather than some other function of that angle is the following beautiful fact: *straight line(segment)s in* Phase_e *correspond bijectively to pencils of rays going through a common point.* Let us elaborate a bit on this. For $x \in e$ the vertical straight line $\{x\} \times \mathbf{R} \subseteq \text{Phase}_e$ corresponds to all rays emanating from x into Q, and for $t \in \mathbf{R}$ the horizontal straight line segment $e \times \{t\}$ corresponds to all lines emanating from e in the direction given by t (and hence intersecting in a common point at infinity in the projective plane). Let a, b be the endpoints of e, listed in clockwise order on vertices of Q. Fix distinct $t_a, t_b \in \mathbf{R}$ and draw the lines l_a, l_b (not just half-lines) from a, b with directions given by t_a, t_b, and let p be their common intersection. If $t_a < t_b$, then p lies on the same side of e as Q, while if $t_a > t_b$, then p lies "behind" e. In either case, a straightforward computation shows that the straight line segment in Phase_e between (a, t_a) and (b, t_b) parameterizes precisely the lines going through p.

Given two distinct edges e and f we define the set $K_{ef} \subseteq \text{Phase}_e$ as the set of all rays emitted from e that hit f next. This is, in fact, a convex polyhedron in Phase_e. To see this, we distinguish two cases. First assume that e and f are not adjacent in Q. Let a, b be the vertices of e in clockwise order, and let c, d be the vertices of f in clockwise order. Then the boundary of K_{ef} consists of all rays from vertex a to edge f, from vertex b to edge f, from edge e to vertex c, or from edge e to vertex d. Let $t_{ac}, t_{ad}, t_{bc}, t_{bd}$ denote the tangents of the angles that the lines ac, ad, bc, bd make with the inward normal of e. Then the observation above shows that K_{ef} is the convex quadrangle in Phase_e with corners $(a, t_{ac}), (b, t_{bc}), (b, t_{bd}), (a, t_{ad})$ (in clockwise order), see Figure 10.

Next, assume that e and f are adjacent. We first assume that f follows e in clockwise order; the opposite ordering is treated similarly. Let a, b be the vertices of e and let b, d be the vertices of f. Now the boundary of K_{ef} consists of the rays from a to f, forming the vertical half-line from (a, t_{ad}) downwards, the rays from b to f, forming the half-line from (b, t_{bd}) downwards [2], and the rays from e to f, forming the segment between (a, t_{ad}) to (b, t_{bd}). Hence, by abuse of notation, K_{ef} is the convex hull of $(a, -\infty), (b, -\infty), (b, t_{bd}), (a, t_{ad})$. See Figure 11, also for the case where e follows f in clockwise order and where

[2] to see that this is the right half-line, use a limiting argument: consider lines from a point $x \in e$ close to b to f

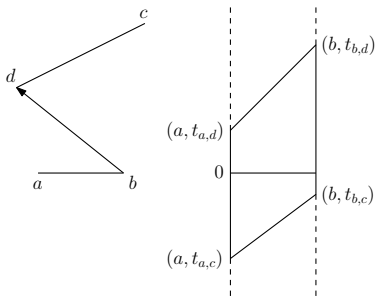


Figure 10: The polyhedron K_{ef} for e, f not adjacent.

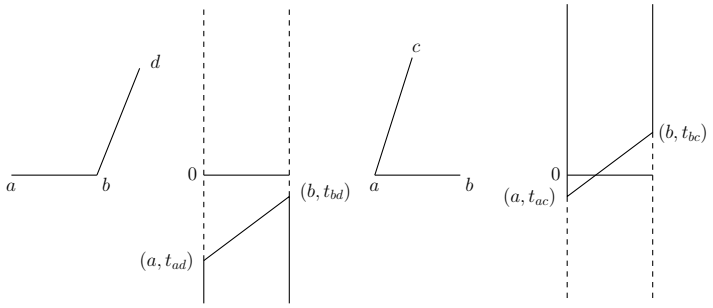


Figure 11: The polyhedron K_{ef} for e, f adjacent.

K_{ef} is the convex hull of $(a, t_{ac}), (b, t_{bc}), (b, +\infty), (a, +\infty)$.

Still for distinct edges e, f of Q we define the map $T_{ef} : K_{ef} \to \text{Phase}_f$ that takes a ray in Phase_e traveling from $x \in e$ to $y \in f$ and returns the ray in Phase_f traveling back from y to x (we do *not* yet reflect here). This map is *not* affine-linear. Indeed, as before, label the endpoints of e by a, b and those of f by c, d. Let u_e, u_f be inward pointing normals of e, f of the same lengths as e, f, respectively. Then the second component of $T_{ef}(x, t)$ equals the rational expression

$$\frac{[u_e + t(a-b)] \cdot [c-d]}{[u_e + t(a-b)] \cdot u_f}$$

in t, where $\cdot$ is the dot product on $\mathbf{R}^2$. The first component is computed by intersecting with f the ray emanating from $x \in e$ in direction t with f.

Note that T_{ef} is a bijection between K_{ef} and K_{fe} with inverse T_{fe}. Strictly speaking, this is only true for the *interiors* of these polyhedra, but it will be convenient in our computations to use their boundaries in our computations. To this end, when $f = (b, d)$ follows $e = (a, b)$ in clockwise order of edges, it will by convenient to set $T_{ef}(a, -\infty), T_{ef}(b, -\infty) := (b, t_{ba})$, where t_{ba} is the tangent of the (b, a) makes with the inward normal to f. Similarly, we set $T_{ef}(b, t_{bd})$

equal to *both* (d,∞) and (b,∞) (these should really be thought of as one and the same point in the projective plane). When $e=(a,b)$ follows $f=(d,b)$, we set $T_{ef}(a,\infty), T_{ef}(b,\infty) := (a,t_{ab})$ and $T_{ef}(a,t_{ac})$ equal to *both* $(c,-\infty)$ and $(a,-\infty)$. Note that if we use, on any edge $e=(a,b)$, the coordinate t for the point $(1-t)a+tb$, then T_{ef} is orientation reversing.

In spite of the fact that T_{ef} is not affine-linear, it *does* have the property that the pre-image $T_{ef}^{-1}m$ of a straight line segment $m\in \mathrm{Phase}_f$ is a straight line segment in K_{ef}. Indeed, the line segment m corresponds to (part of) the pencil of lines from f going through some common point p (possibly behind f or even at infinity). But then then all lines in $T_{ef}^{-1}m$ go through p, as well, and hence lie on a line segment in Phase_e. This argument shows that the bijection $T_{ef}: K_{ef}\to K_{fe}$ maps polyhedra to polyhedra. If a polyhedron in K_{ef} is given by its clockwise list of corners, then mapping T_{ef} to that list gives the counterclockwise list of corners. When e and f are adjacent, some of the corners may be mapped to an ordered "pair" of points at infinity.

For tracing rays traveling through Q we will need the maps $\tau_e : \mathrm{Phase}_e \to \mathrm{Phase}_e,\ (x,t)\mapsto(x,-t)$; these reflect rays in the normal vector u_e.

Now, consider a sequence $s=(e_1,\dots,e_m)$ of edges of Q, where $e_i\neq e_{i+1}$ for all i. We claim that the set K_s of rays in Phase_{e_1} that travel to e_2, and then after reflection to e_3, etc, and finally to e_m, form a (possibly empty) convex polyhedron. We have already seen this for $m=2$, where K_{e_1,e_2} is as above. For $m=1$ it is also true if we agree that $K_{(e_1)}$ is just all of Phase_{e_1}. For $m>2$ let $s'=(e_2,\dots,e_m)$ be the tail of s. Then $K_s = T_{e_1e_2}^{-1}\tau_{e_2}K_{s'}$, and the claim follows from the observation that τ_{e_2} is affine-linear and that $T_{e_1e_2}$ pulls back straight line segments to straight line segments. This also gives an inductive *algorithm* for computing K_s: intersect $\tau_{e_2}K_{s'}\subseteq\mathrm{Phase}_{e_2}$ with K_{e_2,e_1}. This gives a polyhedron. Map its corners into Phase_e by means of $T_{e_2e_1}$. The polyhedron with the corners thus obtained is K_s.

Now that we have an algorithm for computing K_s for any sequence of edges, we single out two edges source, target of Q representing the source and the target, respectively. We want to enumerate all sequences s of edges, starting with source and ending with target, for which K_s is non-empty (or even better, an honest, two-dimensional polyhedron), and then we want to draw the corresponding polyhedral subdivision of $\mathrm{Phase}_{\mathrm{source}}$. One issue with this is that for some triples $(Q,\mathrm{source},\mathrm{target})$ this subdivision is not finite. We will assume, however, that it is. Then it can be computed recursively as follows: given a string $s'=(e_2,\dots,e_m)$ with $m\geq 1$ for which $K_{s'}\subseteq\mathrm{Phase}_{e_2}$ is non-empty (respectively, two-dimensional), check whether $e_2=\mathrm{source}$. If so, then return s' and the corresponding polyhedron. If not, let e_1 run through all edges of Q distinct from e_2, set $s:=(e_1,\dots,e_m)$, and compute K_s from $K_{s'}$ as above. If it is non-empty (respectively, two-dimensional), call the procedure just described with the sequence s.

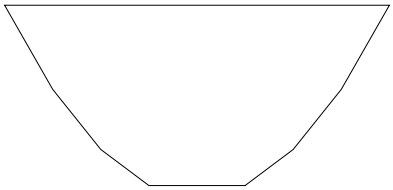


Figure 12: The polygon Q.

An example

Consider the polygon Q with vertices $(\pm i, (i^2-1)/4)$ for $i = 1, 2, 3, 4$, a reasonable approximation of a parabolic collimator, see Figure 12. We label the edges $1, \ldots, 8$ in clockwise order; edge 1 is source, edge 5 is target.

It turns out that there are 15 strings s for which K_s is non-empty. They are:

$$(1,5),(1,2,5),(1,3,5),(1,2,3,5),(1,4,5),$$
$$(1,2,4,5),(1,3,4,5),(1,2,3,4,5),(1,6,5),(1,7,6,5),$$
$$(1,8,7,6,5),(1,8,6,5),(1,7,5),(1,8,7,5),(1,8,5).$$

The corresponding polyhedra in $\text{Phase}_{\text{source}}$ and $\text{Phase}_{\text{target}}$ are depicted in Figure 13. Another example, now with 4 replaced by 7 to resemble a parabola even closer, gives rise to Figure 14.

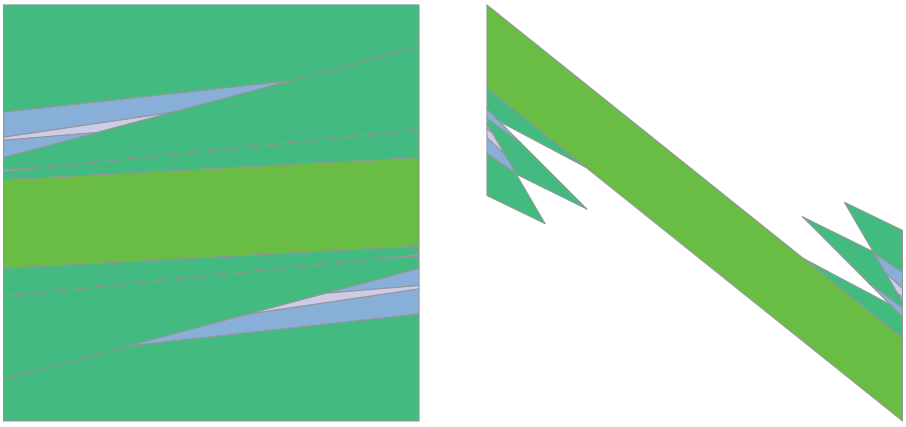


Figure 13: Left The polyhedra K_s—lighter colors indicate more reflections. The line segments separate K_s for different s. Two of the regions are too small to discern. Right The images in $\text{Phase}_{\text{target}}$ of the polyhedra K_s.

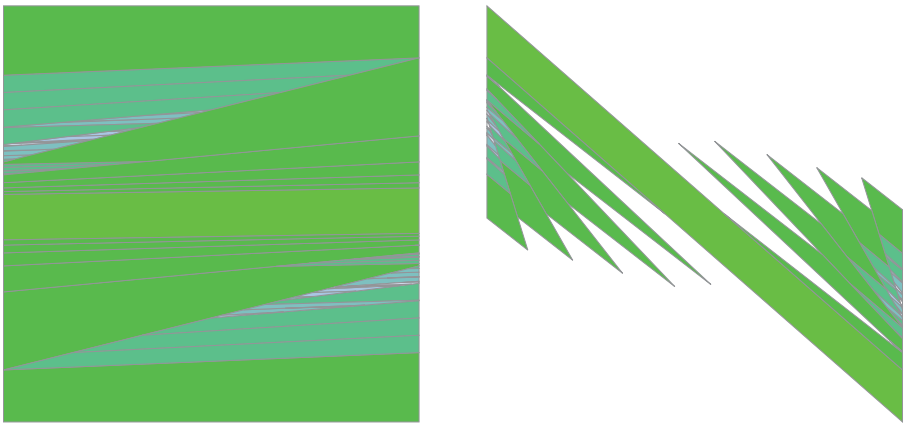


Figure 14: Left The polyhedra K_s—lighter colors indicate more reflections. The line segments separate K_s for different s. Two of the regions are too small to discern. Right The images in $\text{Phase}_{\text{target}}$ of the polyhedra K_s.

3 Two-dimensional smooth cups

In this section we will study a 2-dimensional cup with smooth edges. The aim is to find an efficient numerical algorithm to reconstruct the partition of $\text{Phase}_{\text{source}}$. This was also suggested by the problem owners. For now, this method still has a somewhat limiting restriction on the cup shape, the fourth assumption below. Possibly, it can be removed with a little further research and small adaptations of the method. We assume

1. $p(x) \in C^1((-1,1))$, where $p(x)$ is function that defines the shape of the cup;

2. $X = (h_{-1}, h_1) \times \{p(h_1)\}$, defined so that the light source occupies the whole space between the sides of the cup at height $p(h_1) \in [0, p(1))$;

3. $\tilde{X} = (-1,1) \times \{p(1)\}$;

4. the tangent to the cup does not tend to a vertical when approaching the left or right cup edge

Each point of the source emits light with angle $\theta \in (-\pi/2, \pi/2)$ and we deal with the space $\text{Phase}_{\text{source}}$ as before. We present results for a cup described by $p(x) = 10x^6$. The space $\tilde{X}$ is delimited by end points $P_{-1} = (-1, p(-1))$ and $P_1 = (1, p(1))$. Our example light source is the horizontal strip $(-0.5, 0.5) \times \{0.3\}$, see Figure 15.

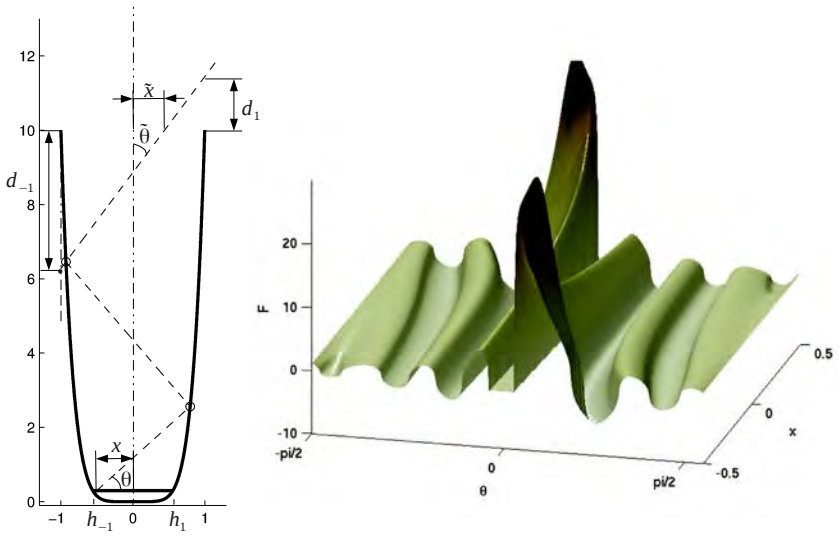


Figure 15: Left A smooth cup described by $p(x) = 10x^6$. The light source is the segment $(-0.5, 0.5) \times \{0.3\}$. The dashed line is the trajectory of a light ray with two reflections. Right The continuous function F is defined so that it vanishes on the boundaries of the sets M_k.

3.1 Reconstruction of the partition for $\mathrm{Phase}_{\mathrm{source}}$

We deal with the numerical reconstruction of the partition M_k of $\mathrm{Phase}_{\mathrm{source}}$ where

$$\bigcup_k \overline{M_k} = \mathrm{Phase}_{\mathrm{source}}, \qquad \bigcap_k M_k = \emptyset. \tag{4}$$

Since each set in the partition is a region (connected, open) in $\mathrm{Phase}_{\mathrm{source}}$ it is enough to efficiently reproduce the union $S = \cup_k \partial M_k$ of the boundaries separating the open sets M_k. It is easy to show that a ray corresponding to a point $s \in S$ passes exactly through one of the points P_{-1} or P_1. Note that the points $P_{-1,1}$ are in $\partial\tilde{X}$ and not in $\tilde{X}$. These facts play a key role in the method of the current section.

We denote $T(x,\theta)$ to be a ray-tracing map – a function that determines the angle and position of the exiting ray for the angle and position of the emitted ray

$$\begin{aligned} T : \mathrm{Phase}_{\mathrm{source}} &\longrightarrow \mathrm{Phase}_{\mathrm{target}} \\ (x,\theta) &\mapsto (\tilde{x},\tilde{\theta}). \end{aligned}$$

The map $T(x,\theta)$ is continuous when $(x,\theta) \in M_k$ and has a jump discontinuity when $(x,\theta) \in \partial M_k$. This fact dramatically affects the convergence speed of common numerical methods. Better results can be achieved by introducing the continuous function

$$F : \mathrm{Phase}_{\mathrm{source}} \longrightarrow \mathbf{R}$$

that we will now describe. Moreover, we define $d_{-1,1} : \mathrm{Phase}_{\mathrm{target}} - \tilde{X} \times \{0\} \to \mathbf{R}$ by

$$\begin{aligned} d_{-1}(\tilde{x},\tilde{\theta}) &:= \cot\tilde{\theta}(1+\tilde{x}) \\ d_1(\tilde{x},\tilde{\theta}) &:= -\cot\tilde{\theta}(1-\tilde{x}). \end{aligned}$$

The geometrical interpretation is that d_{-1} is the signed vertical distance from P_{-1} to the intersection of the line $x = -1$ with the line on which the exiting ray lies. It is positive if the intersection point lies above P_{-1}. Similarly, d_1 is the signed vertical distance from P_1 to the intersection of the line $x = 1$ with the line on which the exiting ray lies. It is positive if the intersection point lies above P_1. This is illustrated in Figure 15. Note that for $\theta = 0$, also $\tilde{\theta} = 0$, the light ray goes straight up, and d_{-1}, d_1 are undefined. Moreover, in the vicinity of the line $\theta = 0$ in $\mathrm{Phase}_{\mathrm{source}}$, d_{-1} and d_1 are unbounded. Our function F will handle this phenomenon, which is unpleasant for root finding. More importantly, if the light ray grazes an upper edge of the cup, either $d_{-1} \circ T$ or $d_1 \circ T$ has a jump discontinuity across zero and the other is zero, see Figure 16. The points for which this is the case form S. This discontinuity is bad for an approximation algorithm to localize the zeros. The function F, that we now define,will remedy

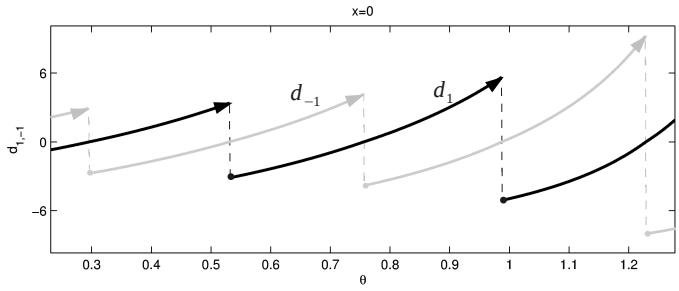


Figure 16: The functions $d_{-1} \circ T$ and $d_1 \circ T$ alternate their signs around any $s \in S$ when fixing x and varying θ. However, only one of the two is continuous at s.

this problem and allow us to find the zeros of both functions simultaneously. We set

$$F(x,\theta) := \begin{cases} (d_{-1} \circ T)(x,\theta) \cdot (d_1 \circ T)(x,\theta) \cdot \operatorname{sign}(\theta)\operatorname{sign}(\mathrm{pr}_2 \circ T(x,\theta)), & (x,\theta) \notin M_0 \\ F_0(x,\theta), & (x,\theta) \in M_0 \end{cases}$$

$$= \begin{cases} d_{-1}(\tilde{x},\tilde{\theta}) \cdot d_1(\tilde{x},\tilde{\theta}) \cdot \operatorname{sign}(\theta \cdot \tilde{\theta}), & d_{-1}(x,\theta) \cdot d_1(x,\theta) < 0 \\ F_0(x,\theta), & d_{-1}(x,\theta) \cdot d_1(x,\theta) \geq 0 \end{cases}$$

outside of S. Here F_0 is a simple continuous function of our choosing that is strictly positive on M_0 and zero on ∂M_0. It is introduced to overcome the discontinuity problem around $\theta = 0$. An example of a suitable F_0 is

$$F_0(x,\theta) = -\left(\theta - \arctan\left(\frac{p(1) - p(h_1)}{x-1}\right)\right)\left(\theta - \arctan\left(\frac{p(-1) - p(h_{-1})}{x+1}\right)\right).$$

It turns out that F is a continuous function on the whole $\mathrm{Phase}_{\mathrm{source}}$ and equal to zero precisely on S. The reason this is true is that when one of the compositions $d_{-1} \circ T$ and $d_1 \circ T$ continuously approaches zero, when we vary the ray such that it passes one of the edge points P_{-1}, P_1, the other stays bounded.

The other important property of F is that when we cross one of its zeros, the signs of both d_{-1} and d_1 switch. But also, the sign of $\tilde{\theta}$ switches. For completeness, the signs of d_{-1}, d_1 and $\tilde{\theta}$ are $(-,+,-)$, respectively, if the last reflection is on the cup's left side and $(+,-,+)$, respectively, if it is on the right. Also, the factor $\operatorname{sign}(\theta)$ ensures that sign switching occurs at ∂M_0. As a consequence, the function F always changes its sign at zeros. This is crucial to root approximation algorithms.

In summary, the problem of reconstructing (4) has now been expressed in terms of F

$$s \in \bigcup_{k \in K} \partial M_k \iff F(s) = 0. \tag{5}$$

Faced with problem (5) one has a vast choice of numerical root-finding methods. For example, both the secant method and inverse interpolation have been implemented in many mathematical packages, like GNU Octave or Matlab.

Usually, an implementation of a root-finding algorithm expects only a one-dimensional function as an input. Therefore, the following strategy might be useful. Fix a sequence of equally spaced points $(x_i)_{i=0}^n$, $x_i \in X$, so that $x_{i+1} - x_i = d$ is a small number, called a discretization step. We now iteratively solve the one-dimensional sub-problems $F(x_i, \theta) = 0$ for θ. To do this in a smart way, when computing the solution set $\{x_i\} \times \Theta_i$, we use the previously computed solution sets $\{x_j\} \times \Theta_j$ for $j = 1, 2, \ldots, i-1$ to determine the starting points for a one-dimensional root-finding algorithm.

More generally speaking, one may use a whole ensemble of numerical tricks introduced for ODE, e.g. self adaptive grids, multi-step methods. For example, while reconstructing trajectory in $\text{Phase}_{\text{source}}$, an algorithm may automatically vary the discretization step d so the overall result meets some error estimate, using as few nodes as possible. This is illustrated in Figure 17 on the left. The nodes are more densely distributed around critical places.

Despite the case with ODE systems, root-finding is a much "better behaved" process since no accumulative error is introduced. A root-finding algorithm will always converge to an exact value within margins of the predefined tolerance.

4 Monte-Carlo methods for intensity computation

A simple and practical, albeit computationally intensive way to compute the intensity profile for a given reflector is to use a Monte-Carlo method to perform essentially numerical integration. Such an approach is part of a larger class of generically known as Monte-Carlo (MC) methods. These rely mostly on simulation of the processes of interest (either random or not), and use the outcomes of the simulation to compute important quantities (in this case, the intensity profile). These methods are simple enough, can be endowed with performance guarantees, and rely on the fact that it is generally easy and quick to simulate the desired processes. The latter point is critical, and it is often the bottleneck of such approaches. For the problem under consideration simulation of the process involves ray-tracing which, despite its simplicity, can be too computationally expensive if one is required to compute several millions of rays to ensure the desired performance. Therefore a naïve and straightforward application of such methods can still be prohibitive. In this section we discuss two possible approaches to *bootstrap* the basic MC approach, which will use a small number of ray-tracing experiments to reconstruct the intensity profile to a high accuracy, by relying on the analytical considerations of the previous sections, as well as by using clever sampling strategies to choose only "important" rays to simulate.

The overall goal of this section is to compute the intensity function at the target

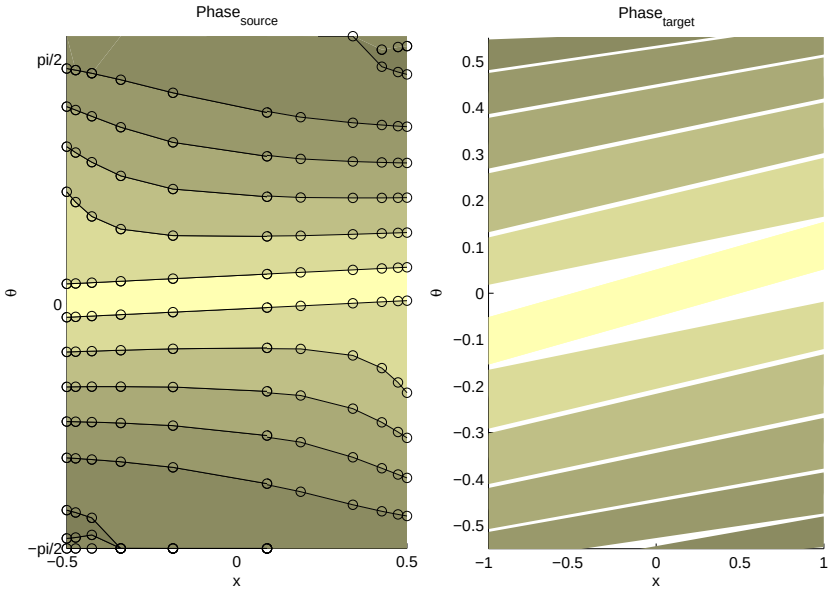


Figure 17: Left The partition of $\text{Phase}_{\text{source}}$ into M_k for the cup described in Figure 15 was obtained numerically. The $\circ$-signs mark grid points computed with a mix of the secant method and inverse interpolation root-finding algorithms. Right The partition of $\text{Phase}_{\text{target}}$ into regions with same number of reflections k. In both pictures, darker colors correspond to a higher number of reflections.

for different designs of optical systems. As was seen before, this requires both knowledge of the transformation T as well as a slightly complicated pull-back integration step. As should be clear by now, even for very simple two-dimensional faceted reflector designs deriving an analytical solution is already rather involved. Therefore extension to non-faceted cups and the three-dimensional setting is likely to be a daunting task. Another possibility is the use of Monte-Carlo integration to approximate the intensity function I_{target}. For a single ray, parameterized by a source position x and source angle θ we can evaluate the transformation function T and therefore the target angle $\tilde{\theta}$ and target position $\tilde{x}$. We restrict ourselves to the far field problem explained in the introduction, hence we are not interested in the target position. By collecting many such samples the integral (3) can be computed numerically to a desired precision level. Unfortunately, to have any reasonable accuracy this method requires a huge number of samples and it is not efficient. However, leveraging on the information for the previous section a different approach can be taken.

The analytical results of the previous sections inform us of the nature of the source phase space which for faceted cups can be decomposed into several polygonal (and furthermore convex) regions in which rays display the same reflective behavior. Since for all rays in each region the reflective behavior is qualitatively the same (i.e. these rays "hit" exactly the same reflective surfaces) this means that two "close-by" rays within the same region (i.e. rays starting at nearby points x_1 and x_2, with similar starting angles θ_1 and θ_2) will display nearly the same behavior. In fact for faceted cups given a few rays in each region it is possible to quickly infer the behavior of *any other ray* in that same region. This means that a Monte-Carlo approach only needs to trace a very small number of rays in each region. The previous statement is obviously conditional on the knowledge of the way the input phase-space is partitioned. In this section we describe two ways of using MC methods based on this rationale:

1. The partition of $\text{Phase}_{\text{source}}$ is given – this can be accomplished either using analytical methods (see Section 2), or through clever use of ray-tracing (see Section 3).

2. The regions partitioning $\text{Phase}_{\text{source}}$ into equivalence classes of reflective behavior can be assumed to be convex (or nearly convex). In this case we propose an algorithm that carefully chooses rays to be traced, and effectively estimates the desired partition of the input phase space, while also computing the intensity profile.

4.1 Method 1: Known input phase space partition

Assume we know how the input phase space is partitioned into regions with the same reflective behavior. In particular, we know for each ray $(x, \theta) \in X \times \Theta$ the number of reflections it experiences. For multifaceted cups this information can

be provided analytically. For smooth-shaped reflectors the method in Section 3 can be used to approximately get this information. The method described is similar in spirit to the original naïve MC method, and relies on the characterization of the behavior of a large number of rays. However, for most of these rays this characterization does not require invoking a ray-tracing routine, therefore dramatically decreasing the computational demands.

The proposed method begins by generating a small sample of n_1 points uniformly distributed over the input phase space $\text{Phase}_{\text{source}}$. Let this sample be denoted as the *preliminary sample*

$$\{(x_i, \theta_i)\}_{i=1}^{n_1} .$$

For each one of these points we run a ray-tracing routine. This means that for each point (x_i, θ_i) in the preliminary sample we know both the target angle $\tilde{\theta}(x_i, \theta_i)$ and the number of reflections $r(x_i, \theta_i)$. This step forms the first stage of the method.

In the second stage we generate a much larger uniform sample of n_2 points from the source phase-space, given by the set

$$\{(x_j, \theta_j)\}_{j=n_1+1}^{n_1+n_2}$$

which we denote by the *estimation sample*. Now, instead of running the ray-tracing routine for each of these points we are going to make use of the assumed knowledge about the phase-space. In particular, with that knowledge we can compute the number of reflections $r(x_j, \theta_j)$ for any point in our sample, without resorting to ray-tracing. So all that is left to be calculated is the exit angle $\tilde{\theta}(x_j, \theta_j)$. Let $j \in \{n_1 + 1, \ldots, n_1 + n_2\}$ and consider the point (x_j, θ_j). Now simply take the three "nearest" rays with the same number of reflections in the preliminary sample.[3] Define

$$d(i, j) \equiv \sqrt{(x_i - x_j)^2 + (\theta_i - \theta_j)^2} , \tag{6}$$

and let $G_j = \{i \in \{1, \ldots, n_1\} : r(x_i, \theta_i) = r(x_j, \theta_j)\}$. Finally, let $i_1, i_2, i_3 \in G_j$ be three distinct points so that for any $i \in G_j$ (unequal to i_1, i_2, i_3) we have $d(i, j) \geq \max\{d(i_1, j), d(i_2, j), d(i_3, j)\}$. We determine $\tilde{\theta}(x_j, \theta_j)$ by linear interpolation using the three nearest neighbours. In other words, determine $\tilde{\theta}(x_j, \theta_j)$ so that the points

$$(x_{i_1}, \theta_{i_1}, \tilde{\theta}(x_{i_1}, \theta_{i_1})),\ (x_{i_2}, \theta_{i_2}, \tilde{\theta}(x_{i_2}, \theta_{i_2})),\ (x_{i_3}, \theta_{i_3}, \tilde{\theta}(x_{i_3}, \theta_{i_3})),\ \text{and}\ (x_j, \theta_j, \tilde{\theta}(x_j, \theta_j))$$

are all co-planar. This simple computation is done for all the points in the estimation sample, and can be significantly faster than performing ray-tracing. For

[3] Note that n_1 needs to be sufficiently large, such that each region contains at least three rays from the preliminary sample.

faceted reflectors (in two and three dimensions) this interpolation is actually exact, so in the end its outcome is essentially equivalent to ray-tracing, but less computationally demanding. Finally, with all these in hand, one can just proceed as in regular Monte-Carlo simulation to compute the intensity function, using the $n_1 + n_2$ points.

Remarks: At this point it is important to issue a number of remarks.

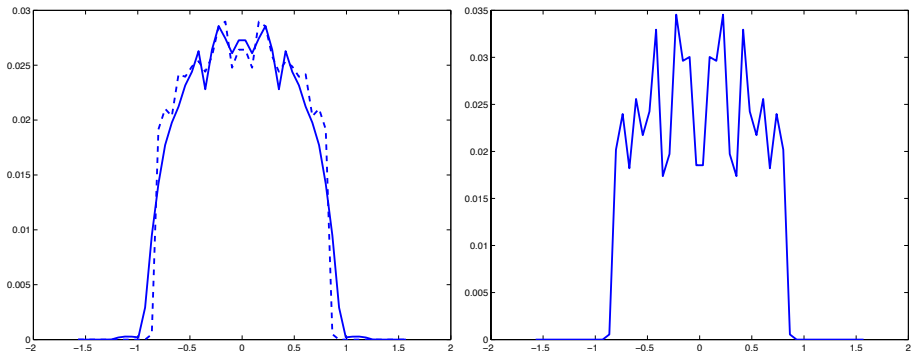
- For faceted reflectors the interpolation step is not approximate, as the exit angle does not depend on the ray position within each of the regions of the source phase-space, and the source and target angles satisfy linear relations.

- The number of reflections does not uniquely identify each of the regions in the phase space, but for our purposes (in two dimensions) it does suffice. This method can easily be extended to resolve this problem, by using instead a label for each reflective path.

- This method will work extremely well *provided* the partition of the phase space is accurate, otherwise it will introduce some systematic errors that can affect the intensity profile estimation. This can be rather undesirable and motivates the second method proposed (see also the simulation results below).

- The notion of distance in (6) is obviously open to discussion as it does not reflect the physics of the problem. The above choice was taken only for simplicity and there are certainly other distances that can be considered.

To assess the potential of this method we used it to compute the intensity profile of the smooth reflector similar to the one used in Section 3. The method described in that section gave rise to the estimated partition of the input phase space, which resembles the one depicted in Figure 17. Using this partition as input to the method described above we constructed an estimate of the intensity profile. In Figure 18a we took $n_1 = 512$ preliminary rays, and ran this method using $n_2 = 2^{18}$. Also in this figure is the result of applying the naïve Monte-Carlo approach with $n_1 = 2^{18}$ rays (this is essentially the true intensity profile). Note that qualitatively the profile obtained by bootstrapping only 512 rays is similar to the "true" profile, however, there are striking differences that cannot be ignored. This systematic error is due mostly to misspecification of the input phase space partition, in which curves were approximated by interpolation of several grid points (see Section 3 for details). To mitigate this drawback we propose a second method that does not rely on such prior knowledge about the input phase space.



(a) (b)

Figure 18: Intensity profile of a smooth reflector cup obtained using bootstrapped MC method 1 using 512 rays (solid line in panel a), and using regular MC with 2^{18} rays (dashed line in panel a). In panel b the result obtained using regular MC with only 512 rays is also depicted.

4.2 Method 2: Unknown partition

In this subsection we give a modified version of the MC method, this time not relying on the complete specification of the input phase space partition. The problem with the application of Method 1, proposed in the preceding paragraphs, is that one needs to have a very accurate description of the input phase space partition and small errors in such description can give rise to significant errors in the intensity profile. Whenever one has an analytical description of the partitioning this is not a problem, but when numerical methods are used to construct such a partition then it becomes difficult to provide any performance guarantees.

The procedure we propose in this section makes very few assumptions about the true partition of $\text{Phase}_{\text{source}}$ (where each region corresponds to a certain reflectivity path). In particular these regions are assumed to be connected (true for any convex reflector). Furthermore we assume these regions are convex when $\text{Phase}_{\text{source}}$ is parameterized by position x and tangent of the source angle $\tan\theta$. The second assumption is true for two-dimensional faceted cups (as the partition regions are convex polygons in such parametrization), but in general false for smooth cups. Nevertheless, if the curvature of these boundaries is modest this assumption is approximately true locally which is all that is required by the method (see more remarks on this later on).

The first step of this method is the same as before: generate a uniformly distributed *preliminary sample* of n_1 points in $\text{Phase}_{\text{source}}$.

$$\{(x_i, \theta_i)\}_{i=1}^{n_1} .$$

As before, for each of these points we run the ray-tracing routine. This means

that for each point (x_i, θ_i) in the preliminary sample, we know both the target angle $\tilde{\theta}(x_i, \theta_i)$ and the number of reflections $r(x_i, \theta_i)$. The second step of the method begins in the same fashion as before, by generating a much larger uniform sample of n_2 points from the source phase space, given by the set

$$\{(x_j, \theta_j)\}_{j=n_1+1}^{n_1+n_2} ,$$

which we denote by the *estimation sample*.

Now, take $j \in \{n_1 + 1, \ldots, n_1 + n_2\}$. Begin by identifying the $K \geq 3$ closest neighbours in the preliminary sample (using again the distance in (6)). For our purposes $K = 12$ was a reasonable option but the proposed method is not terribly sensitive to this choice. For concreteness let $i_1, \ldots, i_K$ denote the K neighbours. The next step is to divide the set of K neighbours into groups of points with the same number of reflections (alternatively the same reflective path). More concretely, for a certain number of reflections $c \in \mathbf{N}_0$ define

$$G_c = \{k \in \{1, \ldots, K\} : r(x_{i_k}, \theta_{i_k}) = c\} .$$

For each $c \in \mathbf{N}_0$ we check if the point $(x_j, \tan\theta_j)$ is *inside* the convex hull of the points $(x_{i_k}, \tan\theta_{i_k})$, with $k \in G_c$. If our assumptions hold then this will happen for at most one group and ensures that the number of reflections for ray (x_j, θ_j) is exactly given by the number of reflections of that group. Suppose this condition holds for c_j^* reflections. Then we are guaranteed that $r(x_j, \theta_j) = c_j^*$ and so we can use all points in $G_{c_j^*}$ to estimate the target angle which can be easily done by linear regression. If the convex hull test is negative for all values c then we cannot identify the point membership and need to run the ray-tracing routine, after which we can add this ray to the preliminary sample rays. Finally, once all the points in the estimation sample have been processed one can proceed with the regular MC integration, as before.

Remarks: Before discussing some experimental results it is important to issue a number of remarks:

1. Note that this method simultaneously identifies the partition of $\text{Phase}_{\text{source}}$ and estimates the intensity profile. Under the faceted cup assumption the partition is convex and the target angle is a linear function of the input angle. This means that the method is exact, in the sense that its outcome is the same as if all the rays were ray-traced. However, the number of ray-traced points is much smaller, mostly consisting on rays near the partition boundaries.

2. The convex partition assumption is required to ensure that the partition membership test is accurate. When this assumption is not met a non-vanishing bias in the estimation of the intensity profile will most likely be present. Nevertheless this bias should typically be quite small: the approximation quality depends on the curvature of the boundaries between

the various regions of the $\text{Phase}_{\text{source}}$ space, and on the average distance between neighboring ray-traced points. This means that in practice the convexity assumption is essentially valid for most smooth cups one might want to consider. Therefore, the primary source of error is the linear regression step. A possible way to ensure there is no asymptotic bias (due to non-convexity) is to modify the proposed procedure in the following way: whenever the convex hull test is positive one "flips" a coin such that with probability $p > 0$ (but small) the corresponding ray-trace is collected.

3. Higher-order regression models can also be considered, and will likely reduce the bias created by using simple linear regression.

4. With the current approach one always ends up at a uniform sample of rays from $\text{Phase}_{\text{source}}$. This is not necessarily the best way to proceed, although it is convenient for the simple MC integration method used. One could use the proposed sampling ideas to construct non-uniform samples that might increase the efficiency of the numerical integration. This is a possibility for future research and was not explored in the current work.

5. It is possible to endow the MC integration step with performance guarantees, which means that, for two-dimensional faceted cups, the performance of this method can be fully characterized. We conjecture that, for three-dimensional faceted reflectors with planar polygonal sources, similar convexity assumptions hold. This means that this method can also be endowed with performance guarantees in that case.

We illustrate this method for the two-faceted cup and the smooth cup described in Section 3. First, we deal with the two-faceted cup. We choose a preliminary sample size $n_1 = 2^9$ and the estimation size $n_2 = 2^{19}$. In Figure 19a we plot the obtained intensity profile. Compare this with the analytical result in Figure 6. Note that in this case the number of calls to the ray-tracing subroutine was only 31906. In Figure 19b we applied the naïve MC method with $n_1 = 31000$ and in Figure 19c we used the same naïve method with $n_1 = 2^{19}$ rays. Note that panels a and c are nearly identical but the latter required a large number of ray-tracing operations. It is important to notice that we invoke the ray-tracing routine mostly for points near the partition boundaries. Figure 20a illustrates this, where we plot the number of reflections vs. source position and angle. In green are the points in the preliminary sample, in red the points of the estimation sample that were ray-traced, and in blue are the points for which we used linear regression. In Figure 20b we zoom in on a particular region so that this phenomena is more visible.

Next we consider a smooth reflector cup, similar to the one of Section 3. For this reflector cup ray-tracing can be rather time consuming, as the typical number of reflections is relatively high. Furthermore, one must use numerical methods in the ray-tracing routine. To illustrate our method we ran it using

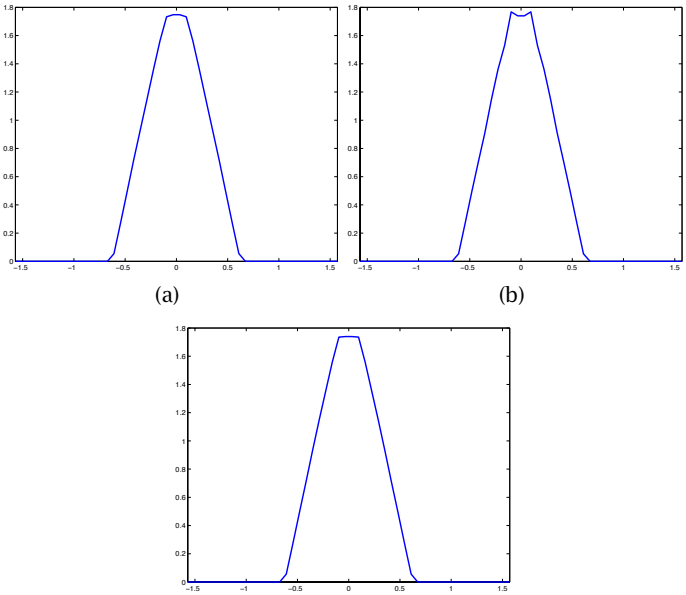
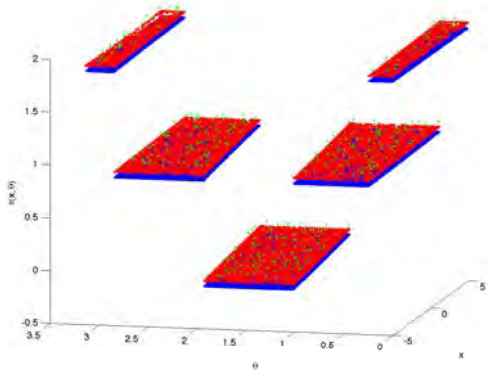
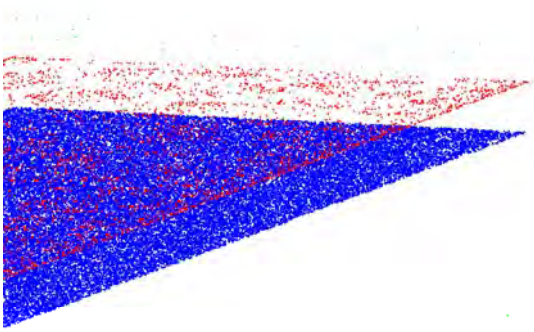


Figure 19: Intensity profile for the faceted reflector cup of Figure 2. In panel a we see the result of our method (bootstrapped MC). In panel b we see the result of a naïve MC method using 31906 rays, and in panel c we display the result for a naïve MC that uses 2^{19} rays.



(a)



(b)

Figure 20: Number of reflections for each sampled point in the phase space. In green are the points in the preliminary sample, in red the points that were later sampled using ray-tracing, and in blue all the points for which we used linear regression. Panel b zooms in a particular part of the plot.

preliminary sample of size $n_1 = 2^9$ and an estimation sample of size $n_2 = 2^{19}$. In the end the method called the ray-tracing subroutine only 38006 times. The resulting intensity plot is depicted in Figure 21a, and computing it took only a few minutes using a rather crude implementation in Matlab. This can be compared with the resulting intensity profile when using a naïve MC method with $n = 2^{18}$ samples which took several hours to compute on the same machine, and is depicted in Figure 21b. As one can see, the two profiles are nearly identical, demonstrating the efficiency of the proposed method. Furthermore, the differences between the two figures are within the accuracy of the Monte-Carlo integration methods when $n_2 = 2^{18}$ rays.

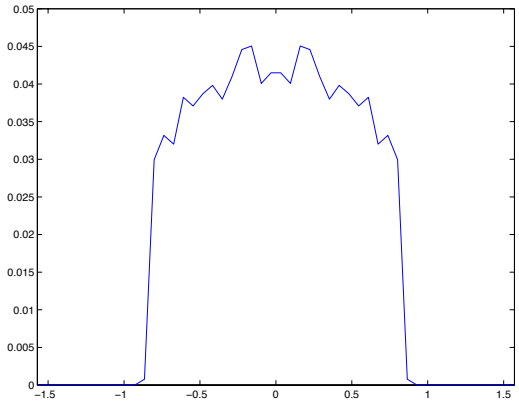
Finally, in Figure 22 we plot the number of reflections as a function of a ray's position and initial angle. Unlike in the faceted cup, the input phase space partition has many more regions, some with a large number of reflections. Nevertheless calls to the ray-tracing routine were done mostly for rays in the boundary of each region.

5 Conclusions and Outlook

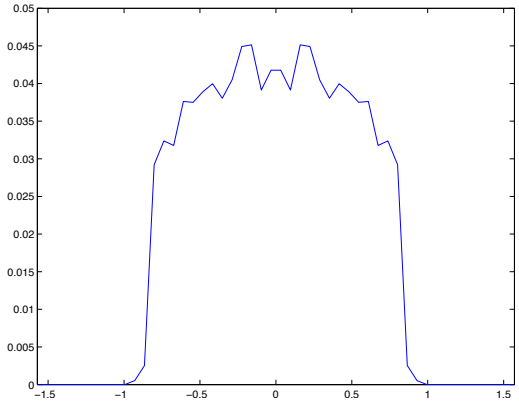
This report shows that, for faceted cups, the analytic approach already yields a lot of information about the target phase space (see Section 2). We started from the relatively simple two-faceted cup (Subsection 2.1) and the philosophy is now that a (symmetric) multi-faceted cup can be regarded as a stack of two-faceted cups where the luminous input in a cup is given by the output of the underlying one. The corresponding results are given in Subsection 2.2. A different class of faceted cups was treated in Subsection 2.3 in which we considered general (convex) polygonal devices.

Section 3 is in two ways different from the preceding ones. First of all, we switched from faceted cups to smooth ones. Secondly, the exact analytic approach was no longer applicable. We proposed a method to approximate the partition of the phase space (both at the source and at the target) numerically. This does not involve random sampling and is based on a completely deterministic method. Moreover, it makes computations quite inexpensive and gives an insight into the solution of the 3-dimensional problem.

We used the information obtained about the partition of the phase space to develop a smarter Monte-Carlo sampling method in Section 4. The first method proposed uses specific information about the subdivision of the source phase space. This is no problem when one indeed has this exact information. However, if only approximate information is available (e.g. it was obtained by the method of Section 3), the corresponding errors in the input might be transferred into the output. Therefore, a second method is introduced that relies on less *a priori* knowledge about the partition.



(a)



(b)

Figure 21: Intensity profile for the a smooth reflector cup. In panel a we see the result of Method 2 (bootstrapped MC), which made use of 38006 calls to the ray-tracing routine. In panel b we see the result of a naïve MC method using 2^{18} calls to the ray-tracing routine. As one can see, differences between the two panels are very minute.

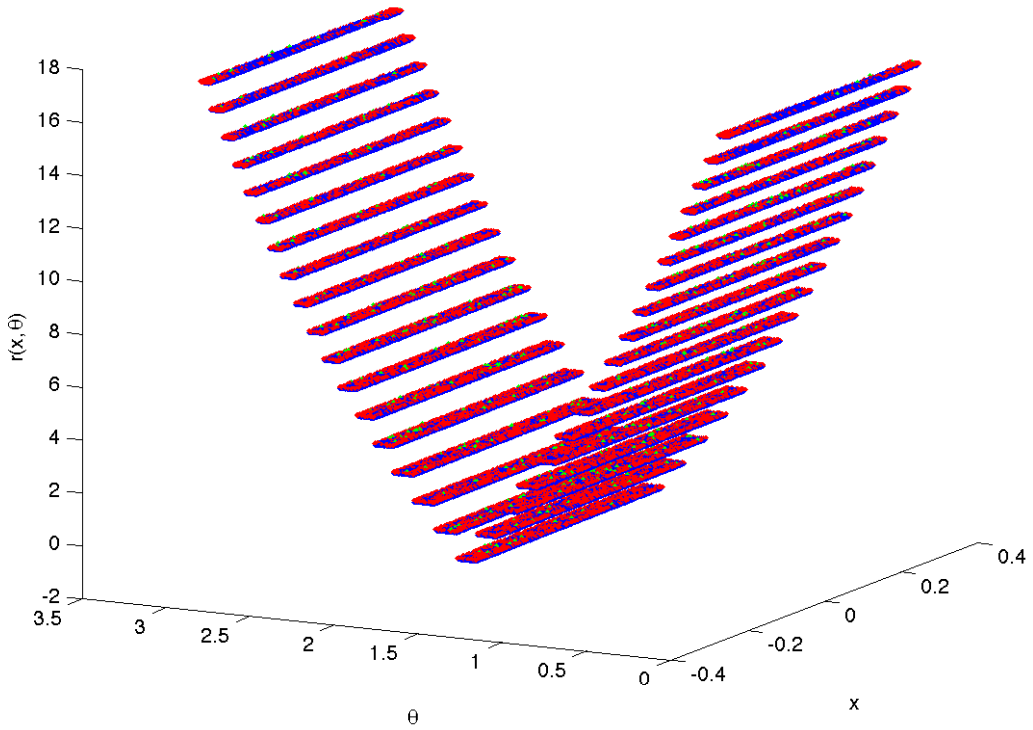


Figure 22: Number of reflections for each sampled point in the phase space, for the smooth cup. In green are the points in the preliminary sample, in red the points that were later sampled using ray-tracing, and in blue all the points for which we used linear regression.

To conclude this report we give some suggestions and directions for future work interesting from either a mathematical or a practical viewpoint. These originate from thoughts and questions that arose during and after the week but did not evolve into concrete results yet.

1. What is the area of the regions in phase space for the 2-facet cup?
2. For what cup shapes are the regions in $\text{Phase}_{\text{source}}$ and $\text{Phase}_{\text{target}}$ convex?
3. What are the θ for which the intensity graph of the far field is non-smooth?
4. Can we express the intensity function on the far field piecewise by elementary (trigonometric) functions when the cup is multifaceted?
5. How to model light rays that graze the surface and follow it, instead of reflecting? Wilbert IJzerman mentioned during the week that these rays arise and can form caustics.

Acknowledgments

We would like to thank Teus Tukker, Ferry Zijp, and Wilbert IJzerman (Philips Lighting) for posing their problem, and for their help during the week. Furthermore, we would like to thank the organizers of SWI 2012 for the opportunity to work together for a week and letting us use the facilities of Eindhoven University of Technology. Special thanks go to Mark Peletier, not only for thinking along about our problem, but also for inviting us all over to his house for a wonderful evening full of tasty desserts.

References

During the Study Group, we did of course not have occasion to study existing literature on the topic in detail. Points of entry suggested by the Philips team are the following.

[1] D. Rausch and A. M. Herkommer. Phase space approach to the use of integrator rods and optical arrays in illumination systems. *Adv. Opt. Techn.*, 1, 2012.

[2] H. Ries and A. Rabl. Edge-ray principle of nonimaging optics. *JOSA A*, 11:pp. 2627–2632, 1994.

[3] R. Winston, J. C. Miñano, and P. Benítez. *Nonimaging Optics*. Academic Press, 2005.