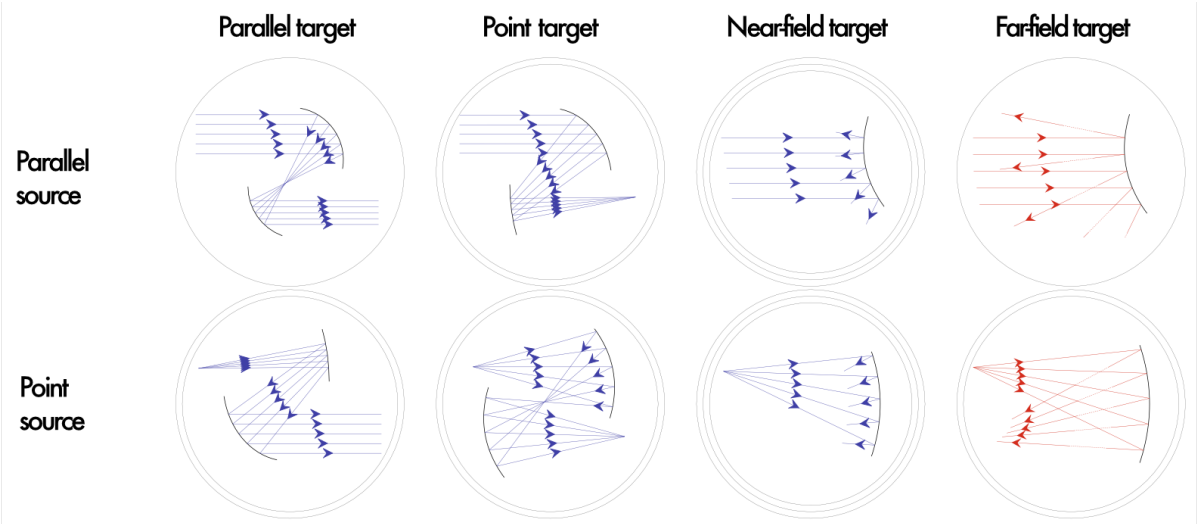
EUROPEAN CONSORTIUM FOR MATHEMATICS IN INDUSTRY

ECMI Modeling Week 2023

University of Szeged, Bolyai Intezet

INVERSE METHODS IN FREEFORM OPTICS

Instructor:	Anthonissen, Martijn	Eindhoven University of Technology
Team:	Cebatarauskas, Paulius	Norwegian University of Science and Technology
	Landwehr, Philipp	Technische Universität Dresden
	Rosztoczy, Csaba	SZTE TTIK Széged
	Röpelinen, Santeri	Lappeenranta-Lahti University of Technology LUT
	Zanrosso, Maddalena	Department of Mathematics "Federigo Enriques" Università degli studi di Milano



August 2023

Contents

1 Abstract

Traditional methods in optical design like ray tracing suffer from slow convergence and are not constructive, i.e., each minimal perturbation of input parameters might lead to "chaotic" changes in the output. However, so-called inverse methods can be helpful in designing optical systems of reflectors and lenses. The equations in $\mathbb{R}^2$ become ordinary differential equations, while in $\mathbb{R}^3$ the equations become partial differential equations. These equations are then used to transform source distributions into target distributions, where the distributions are arbitrary, though assumed to be positive and integrable. In this project, we derive the governing equations and solve them numerically, for the systems presented by our instructor Martijn Anthonissen [Anthonissen et al. 2021]. Additionally, we show how point sources can be derived as a special case of a interval source with directed source interval, i.e., with each point in the source interval there is also an associated unit direction vector which could be derived from a system of two interval sources in $\mathbb{R}^2$. This way, it is shown that connecting source distributions with target distributions can be classified into two instead of three categories.
The resulting description of point sources as a source along an interval with directed rays could potentially be extended to three dimensions, leading to interpretations of point sources as directed sources on convex or star-shaped sets.

Keywords: optical design, freeform optical design, ray tracing, mirror, lens, lenses, Monge-Ampére-Equations, Generated Jacobian Equations, differential equations, ODE, PDE, numerics, numerical solutions, Python, DOPTICS

2 Notation And Conventions

- Following the common notation in the study of optical systems, we study our problem in a Cartesian plane using the x- and z-axis: rays are usually assumed to be emitted in positive z-direction unless explicitly told otherwise.
- Source intervals are assumed to be parallel to the x-axis. Target intervals are usually assumed to be parallel to the x-axis as well, however, their coordinates are being expressed as points with coordinates (y, z), where y is another axis parallel to the x-axis. It is important to use a different axis for their first component for reasons explored in Section 4.
- In particular, we will call the domain of the source function $[x_l, x_r]$ (when it is an interval) and the target interval $[y_l, y_r]$ (or $[y_{1,l}, y_{1,r}]$ and $[y_{2,l}, y_{2,r}]$ when there are two target intervals).
- The target (or targets) are positioned at height L (or L_1 and L_2), while the source interval or point is always going to be at $z = 0$ unless stated otherwise.
- We sometimes use the shorthand notation $\partial_x := \frac{\partial}{\partial x}$ for partial derivatives.
- In our computations, we will often omit the dependencies of the functions, for example we will often write u instead of $u(x)$ for improved readability.
- When talking about equations, LHS means left-hand side, and similarly RHS is the right-hand side of the equation.
- $d(A, B)$ refers to the Euclidean distance between the points A and B in $\mathbb{R}^2$.

- Vectors will by written in bold script. Normalized vectors are additionally indicated by a hat. For example:

$$\hat{\boldsymbol{v}} := \frac{\boldsymbol{v}}{||\boldsymbol{v}||},$$

  where $||\cdot||$ is the Euclidian norm in $\mathbb{R}^2$.
- However, if the argument of the norm is a function, $||\cdot|| := ||\cdot||_\infty$ is the infinity norm unless stated otherwise.

3 Introduction

In our ECMI modeling week project, we studied different configurations of the following problem: given a light source emitting orthogonally from an interval or emitting from a single point and given one or two target intervals, we wish to find one or two mirrors respectively such that the rays from the source illuminate the target(s) the given brightness distribution(s). Those problems were classified by [Anthonissen et al. 2021] as sixteen different models (cf. Figure 1), of which we solved the blue ones. The same figure illustrates which mathematical model is necessary to solve the system in three dimensions: sketches with one single circular line can be solved by a *Standard Monge-Ampére equation* (SMA), systems with two circle lines need to be solved with a *Generalized Monge-Ampére equation* (GMA) while system with three circle lines need *Generated Jacobian equations* (GJE) to be solvable in the formulation suggested by [Anthonissen et al. 2021].
However, some geometric observations can be utilized to reduce the complexity of near-field target problems so that they do not longer require *GJE*s to be solved (compare Section 8).

4 ECMI Modeling Week Challenges

4.1 Problem 1 - Parallel to Near-Field Target

4.1.1 Description

Let us set the stage for the first of our four problems. First, let us look at the source. We fix the source interval on the x-axis at $z = 0$. Our source interval is $[x_l, x_r]$. Then, on this interval, we define a distribution $E : [x_l, x_r] \to \mathbb{R}$. The rays from the source interval are parallel and they are orthogonal to the x-axis, which means that they go straight up. The distribution of the rays from the source is then determined by E.

Now, let us focus on the target interval. This interval is placed at $z = L$ and given by (y_l, y_r). The target interval has another distribution called G, where $G : [y_l, y_r] \to \mathbb{R}$. Both densities E and G have to be integrable. Now our goal is to find a continuous reflector $u = u(x) : [x_l, x_r] \to \mathbb{R}$. In the Cartesian plane, the reflector will be given by $(x, u(x))$ where $x \in [x_l, x_r]$.

4.1.2 Deriving The Equations

Let $\boldsymbol{P} = (x, 0)$, $\boldsymbol{A} = (x, u(x))$, $\boldsymbol{Q} = (y, L)$ where $\boldsymbol{P}$ is a point in our source interval, $\boldsymbol{A}$ is the point on the reflector hit by a ray emitted in $\boldsymbol{P}$, and $\boldsymbol{Q}$ is a point in the target interval. We assume that there is a map $m : [x_l, x_r] \to [y_l, y_r]$ which for each point in the

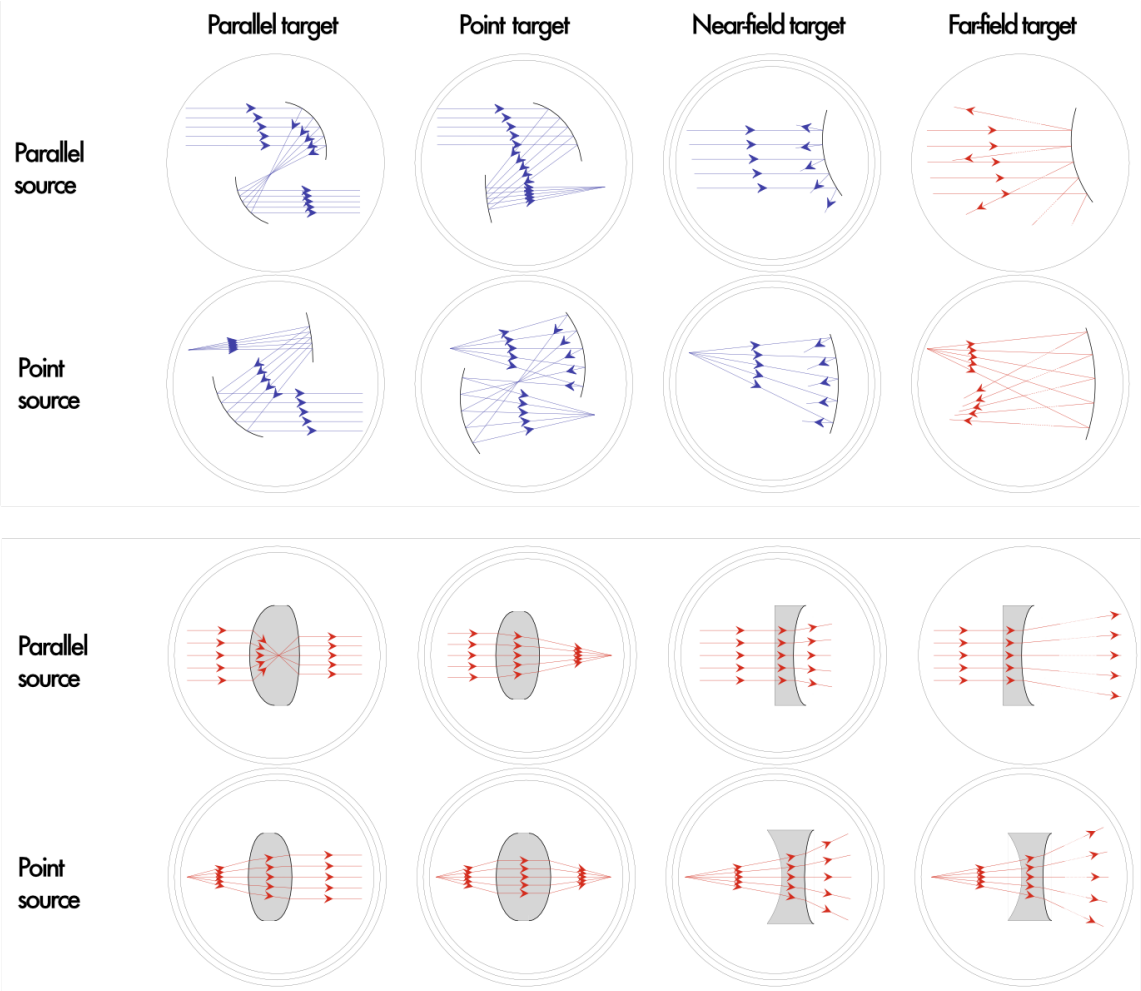


Figure 1: The 16 systems as classified by [Anthonissen et al. 2021]. We solved the *blue* cases during the ECMI Modeling Week.

source interval associates a point in the target interval. This will become important later, but right now we just use this property to derive $\boldsymbol{P}$-dependence of the point $\boldsymbol{Q}$. We will also derive an equation for m later.
The length of the ray emitted at $\boldsymbol{P}$, hitting the reflector at $\boldsymbol{A}$ and hitting the target interval at $\boldsymbol{Q}$ can be calculated as

$$\begin{aligned} V(x,y) &= d(\boldsymbol{P},\boldsymbol{A}) + d(\boldsymbol{A},\boldsymbol{Q}) \\ &= u(x) + \sqrt{(y-x)^2 + (L-u(x))^2}, \end{aligned}$$

where V denotes the Euclidean length of the ray in the Euclidean norm. Hamilton's characteristic function [Lotte Bente Romijn 2021] of the first kind in this case states that $\partial_x V = 0$, which implies that $V = V(y)$ solely depends on y. Thus, we may differentiate the expression above with respect to x and obtain the ODE

$$u'(x) = \frac{y-x}{d(\boldsymbol{A},\boldsymbol{Q}) + u(x) - L} =: f(x,y,u;L). \tag{1}$$

To make this ODE solvable, we need to remove the y dependence from the function f.

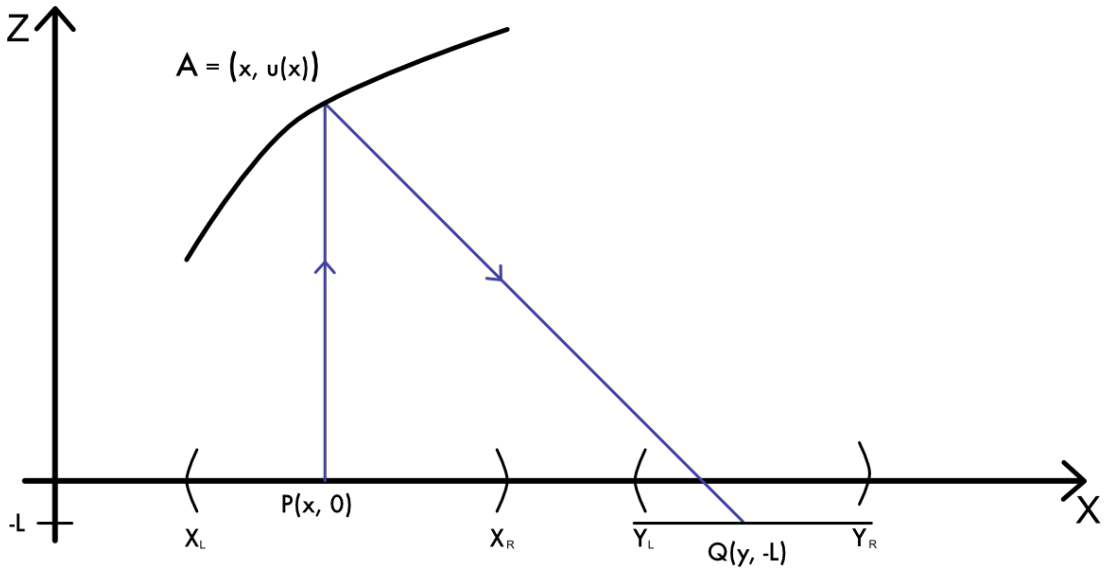


Figure 2: A single ray from a parallel, interval source hits a reflector A before hitting an interval target placed at level $-L$.

Energy conservation and mapping functions
This is where the map m enters the game. By the assumption of energy conservation, the equation

$$\int_B E(\xi)\, d\lambda(\xi) = \int_{m(B)} E(m(\xi))\, d\lambda(\xi)$$

must hold for all $B \in \mathcal{B}([x_L, x_R])$ elements of the Borel set of the domain of the source [L. B. Romijn et al. 2021]. Fortunately, $\mathcal{B}\big([x_L, x_R]\big) = \sigma\big(\{[a,b)\big|x_L \le a < b \le x_R\}\big) = \\ = \sigma\big(\{[a,b]\big|x_L \le a < b \le x_R\}\big) = \sigma\big(\{[x_L,b]\big|x_L \le b \le x_R\}\big)$ [Schilling 2017], and it is sufficient to test energy conservation with Riemann integrals in the equation

$$\int_{x_l}^{x} E(\xi)d\xi = \pm \int_{m(x_l)}^{m(x)} G(y)dy = \pm \int_{x_l}^{x} G(m(\xi))m'(\xi)d\xi. \tag{2}$$

Differentiating and rewriting this expression yields

$$m'(x) = \pm \frac{E(x)}{G(m(x))}. \tag{3}$$

If we choose the $+$, then the initial condition becomes $m(x_l) = y_l$. If we choose the $-$, then the initial condition becomes $m(x_l) = y_r$. So, if we choose the ODE with plus sign in it, then rays do not cross, but if we choose the equation with the minus sign, then the rays will cross.
Now, substituting the equation for y yields $u'(x) = f(x, m(x), u(x))$ with initial condition $u(x_l) = u_0$, where u_0 is a constant that determines the height of where we want the reflector to start. Now, the differential equation can be solved numerically with a common solver like the Gauss method or the Runge-Kutta method (cf. Chapter 6).

4.2 Problem 2 - Parallel Source To Two Targets

This is a generalization of the first problem. Everything about the source is the same. Now instead of having one target we have two targets, T_1 and T_2. We prescribe now the same information for both targets as we did for the first problem, but add an additional subscript 1 or 2 depending on whether they correspond to the first target or the second.

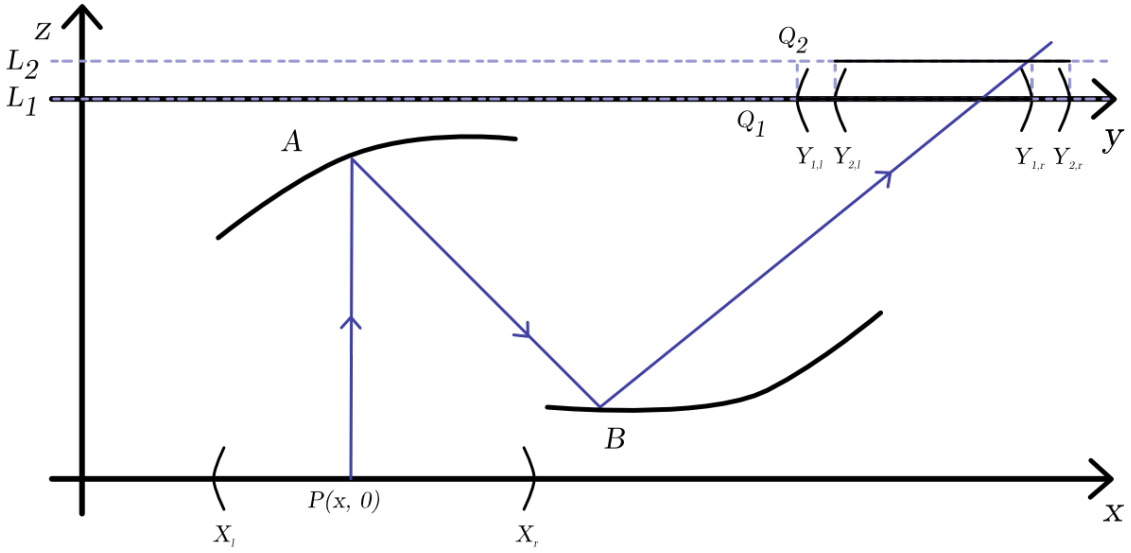


Figure 3: A light ray emitted from a parallel, interval source is reflected by mirrors A and B before hitting two targets consecutively. The y-axis is parallel to the x-axis and (for visual distinction) placed at the L_1-level.

First, we look at T_1 and T_2. Fix points $\boldsymbol{Q_1} = (y_1, L_1)$ and $\boldsymbol{Q_2} = (y_2, L_2)$. We want to find a map m_2 which maps points from T_1 to T_2. This is done as in the first problem, where we computed the map from source to target. This yields the equation

$$m_2'(y_1) = \pm \frac{G_1(y_1)}{G_2(m(y_1))} \tag{4}$$

with initial conditions $m_2(y_{1,l}) = y_{2,l}$ or $m_2(y_{1,l}) = y_{2,r}$. This lets us compute the direction which a ray starting at T_1 has to have. This is given by $\boldsymbol{t(y_1)} = \begin{pmatrix} m(y_1) - y_1 \\ L_2 - L_1 \end{pmatrix}$. Then normalizing this vector we get, $\hat{\boldsymbol{t}} = \frac{\boldsymbol{t}}{\|\boldsymbol{t}\|}$. Now given a point $\boldsymbol{Q_1}$, let $\boldsymbol{B}$ be the point where the ray going through $\boldsymbol{Q_1}$ hits the second reflector. Let us write this as $\boldsymbol{B} = \boldsymbol{Q_1} - w(y_1)\hat{\boldsymbol{t}}$, where w is an unknown function to be determined. The point $\boldsymbol{A}$ that is hit by the ray coming from point $\boldsymbol{P}$, is given by $\boldsymbol{A} = \boldsymbol{P} + u(x)\begin{pmatrix} 0 \\ 1 \end{pmatrix}$. This is done as in Problem 1.

Let us also find the map that maps $\boldsymbol{P}$ to $\boldsymbol{Q_1}$. Solving the equation

$$m_1'(x) = \pm \frac{E(x)}{G_1(m_1(x))}$$

with the initial conditions $m_1(x_l) = y_{1,l}$ or $m_1(x_l) = y_{1,r}$ does the trick. Now we need to derive the equations for u and w.

We begin again by computing the optical path length V_x.

$$\begin{aligned} V(x, y_1) &= d(\boldsymbol{P}, \boldsymbol{A}) + d(\boldsymbol{A}, \boldsymbol{B}) + d(\boldsymbol{B}, \boldsymbol{Q_1}) \\ &= u(x) + d(\boldsymbol{A}, \boldsymbol{B}) + w(y_1) \end{aligned}$$

and we know that V satisfies $\partial_x V = 0$ and $\partial_{y_1} V = \hat{t}_1(y_1)$. The first differential equation tells us that $V = V(y_1)$, i.e., that V does not depend on x. But then the second equation is a solvable ODE, so we can compute V. Now squaring the expression for V and isolating w yields

$$w = \frac{-(u-V)^2 + (y_1 - x)^2 + (L_1 - u)^2}{2\Big(u - V + \hat{t_1}(y_1 - x) + \hat{t_2}(L_1 - u)\Big)}, \quad \text{where } \hat{\boldsymbol{t}} = \begin{pmatrix} \hat{t}_1. \\ \hat{t}_2 \end{pmatrix} \tag{5}$$

Now the LHS is a function of y_1 only. Differentiating both sides with respect to x, we get that the w dependence drops out. This expression can be rewritten as $u'(x) = f(x, y_1, V, \hat{t}_1, \hat{t}_2)$. V is a known function, y_1 can be expressed through x using the map m_1. $\hat{\boldsymbol{t}}$ is also known. Thus we can solve this ODE. Once we have u, you also have w from equation (5).

4.3 Problem 3 - Point Source To Near-Field Target

4.3.1 Deriving The Equations

Now we turn our attention to point sources. This first problem is similar to Problem 1. The target is near field, so everything stays the same there. Now how do we describe our source? First of all, we do not have an interval anymore but a single point. This point is placed at the origin for our convenience. Now the energy distribution is angular. We let the rays shoot out in the angle range $[\theta_l, \theta_r]$. On this interval we prescribe an energy distribution $E : [\theta_l, \theta_r] \to \mathbb{R}$. Now the energy conservation is slightly different. The energy over the source becomes $\int_S E \, ds$, which is a line integral. Let $r(\theta) := (\cos\theta, \sin\theta)$. Then, the energy becomes

$$\begin{aligned} \int_S E \, d\gamma &= \int_{\theta_l}^{\theta_r} E(\boldsymbol{r}(\theta)) \cdot \|\boldsymbol{r}'(\theta)\| \, d\theta = \int_{\theta_l}^{\theta_r} E(\boldsymbol{r}(\theta)) \, d\theta \\ &= \int_{x_l}^{x_r} E(\boldsymbol{s}(x)) \cdot \|\boldsymbol{s}'(x)\| \, dx = \int_{x_l}^{x_r} \tilde{E}(x) \cdot \|\boldsymbol{s}'(x)\| \, dx. \end{aligned}$$

Here $\boldsymbol{s}$ is the stereographic projection given as $\boldsymbol{s}(x) = (s_1, s_2)^\top = \left(\frac{2x}{1+x^2}, \frac{1-x^2}{1+x^2}\right)^\top$. Also $x_r = \boldsymbol{s}^{-1}(\theta_l)$ and $x_l = \boldsymbol{s}^{-1}(\theta_r)$. Now the conservation of energy yields

$$\int_{x_l}^{x} \tilde{E}(y) \cdot \|\boldsymbol{s}'(y)\| \, dy = \int_{m(x_l)}^{m(x)} G(y) \, dy = \pm \int_{x_l}^{x} G\big(m(\xi)\big) \cdot m'(\xi) \, d\xi. \tag{6}$$

Now, differentiating the whole expression from equation (6) with respect to x yields the following DE

$$m'(x) = \frac{\tilde{E}(x)\|\boldsymbol{s}'(x)\|}{G(m(x))}.$$

The map m connects the stereographic coordinate x with y in the target interval, i.e., $m(x) = y$. Similar to equation (2), the mapping may either maintain or invert the order of rays: this again depends on the choice of the sign for the equation (6). The rest of the solution is computed in a similar way as in Problem 1, with the substantial difference in regards to the way we express the reflector: while in Problem 1 the function $u(x)$ indicated the height of the reflector from the x-axis, in this third one the same function indicates the distance from the origin.

After the setting was introduced, we proceed to derive the equations. We indicate with the set $R := \{(x, u(x)s(x))|x \in D_E\}$ the reflector we want to compute: the vector s is the stereographic projection, while x is the stereographic coordinate.
One can easily convince themselves that s is a unit vector. Then, we interpret u as a scaling function. Let us now write the expression for the optical path length

$$V(x, y) = d(\boldsymbol{P}, \boldsymbol{A}) + d(\boldsymbol{A}, \boldsymbol{Q}) = u(x) + d(\boldsymbol{A}, \boldsymbol{Q}),$$

where

$$d(\boldsymbol{A}, \boldsymbol{Q}) = \sqrt{u^2 + y^2 - 2yus_1 + L^2 + 2us_2L}.$$

Now, the optical path length generally depends on x and y where x and y are Cartesian coordinates. However, we transform the x-coordinate so that x here is no longer the Cartesian coordinate, and is the stereographic coordinate and y still remains a Cartesian coordinate here. This means that x is a coordinate that describes the emitting angle of the point light source. Thanks to the following Hamilton's characteristic function

$$V = W^\star, \quad \frac{\partial W^\star}{\partial p_s} = q_s = 0 \tag{7}$$

, it is said that the optical path length depends only on the Cartesian coordinate y. Indeed q_s is the x-coordinate of the source point (which is the origin), while p_s is the first coordinate of the vector s: using the chain rule we get the equation

$$\frac{\partial V}{\partial x} = 0 \tag{8}$$

So the optical path does not depend on the angle, and thus does not depend on the stereographic coordinate x. Thus, $V = V(y)$. The explanation means basically that $\partial_x V = 0$ where x is the stereographic x. The Cartesian x is a constant in our case, because the starting point is the origin and hence the Cartesian x equals to zero, i.e., $x = 0$ where x is the Cartesian x. Now taking the equation for V and differentiating both sides with respect to x, we get that V drops out, because it is independent of x. If you then rewrite that expression such that you isolate u' on its own you get the following ODE:

$$u'(x) = \frac{yus_1' - us_2'L}{\sqrt{u^2 + y^2 - 2yus_1 + L^2 + 2us_2L} + u - 2ys_1 + s_2L}$$

By now we know this is solvable. With additional effort, we can solve systems of point sources for two consecutive targets with densities using systems two mirrors.

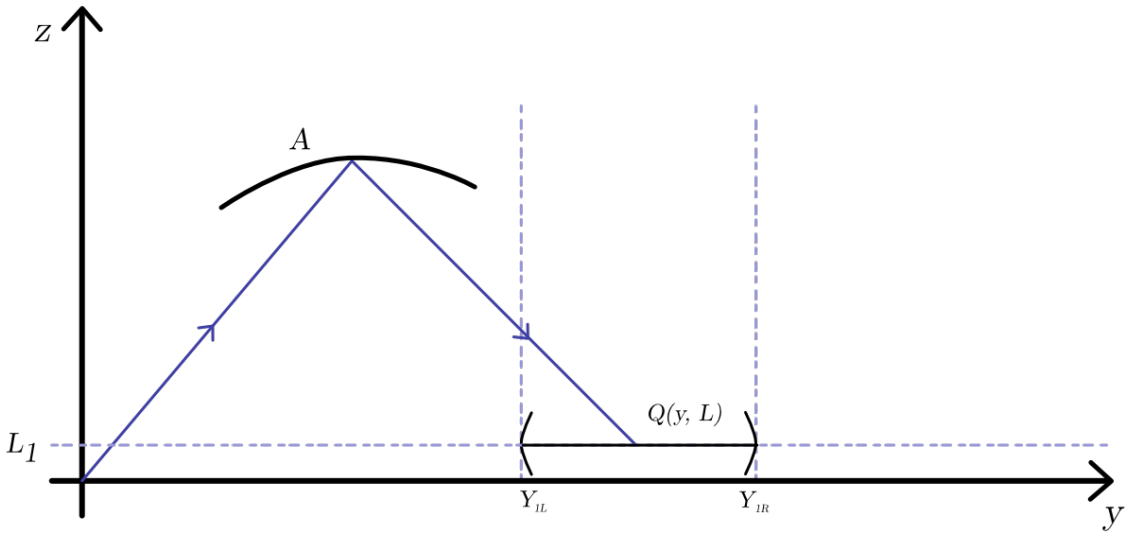


Figure 4: A ray emitted by a point source placed in the origin hits a reflector A and an interval with an intensity function. The angles at which the target is being hit are barely constraint by the problem formulation.

4.4 Problem 4 - Point Source To Two Targets

Solving the problems 2 and 3 gives all the necessary theory to solve problem 4. Problem 4 can be understood as a simple mix of those two. Now, the source is described as in problem 3, while the two targets are described as in problem 2. Again, we first solve two differential equations. The first one is

$$m_1'(x) = \pm \frac{\tilde{E}(x)\|\boldsymbol{s}'(x)\|}{G_1(m_1(x))}, \tag{9}$$

while the second one is

$$m_2'(y_1) = \pm \frac{G_1(y_1)}{G_2(m_2(y_1))}. \tag{10}$$

Using equation (10) lets us compute the direction which a ray starting at T_1 has to have. This direction is given by $\boldsymbol{t}(y) = \begin{pmatrix} m_2(y_1) - y_1 \\ L_2 - L_1 \end{pmatrix}$. Then, normalizing the vector $\boldsymbol{t}$ we obtain $\hat{\boldsymbol{t}} := \frac{\boldsymbol{t}}{\|\boldsymbol{t}\|}$. Now given a point $\boldsymbol{Q_1(y_1)}$, let $\boldsymbol{B(y_1)}$ be the point where the ray going through $\boldsymbol{Q_1(y_1)}$ hits the second reflector. Let us write this as $\boldsymbol{B(y_1)} = \boldsymbol{Q_1(y_1)} - w(y_1)\hat{\boldsymbol{t}}$, where w is an unknown function to be determined. The point $\boldsymbol{A}$ where the ray hits the first reflector is as in problem 3, namely $\boldsymbol{A} = u(x)\boldsymbol{s}(x)$. $\boldsymbol{s}$ is the stereographic projection, while x is the stereographic coordinate. Also you can check that $\boldsymbol{s}$ is a unit vector. Then, u is a scaling function. Let us now write the expression for the optical path length

$$V(x, y_1) = d(\boldsymbol{O}, \boldsymbol{A}) + d(\boldsymbol{A}, \boldsymbol{B}) + d(\boldsymbol{B}, \boldsymbol{Q_1}) = u(x) + d(\boldsymbol{A}, \boldsymbol{B}) + w(y_1).$$

By the same arguments as given in the solution of problem 3, the optical path length $V(x, y_1)$ depends on y_1 only, and we therefore write $V = V(y_1)$. Now by Hamilton's characteristic functions we know that $\frac{dV}{dy_1} = \hat{t}_1(y_1)$. As $\hat{t}_1$ is known, we can compute V

from this equation. Now computing $d(\boldsymbol{A}, \boldsymbol{B})^2 = (V - u - w)^2$ and isolating w, we get the following expression

$$w = \frac{V^2 - 2uV - y_1^2 + 2us_1y_1 - L_1^2 + 2L_1us_2}{2(V - u - y_1\hat{t}_1) + u\hat{t}_1s_1 - L_1\hat{t}_2 + u\hat{t}_2s_2}. \tag{11}$$

Now differentiate both sides with respect to x and rewrite everything in terms of u'. Then you can solve that ODE for u and once you know u you can insert that into the equation for w and then you know w also. Hence, we have illustrated how the problem is being solved.

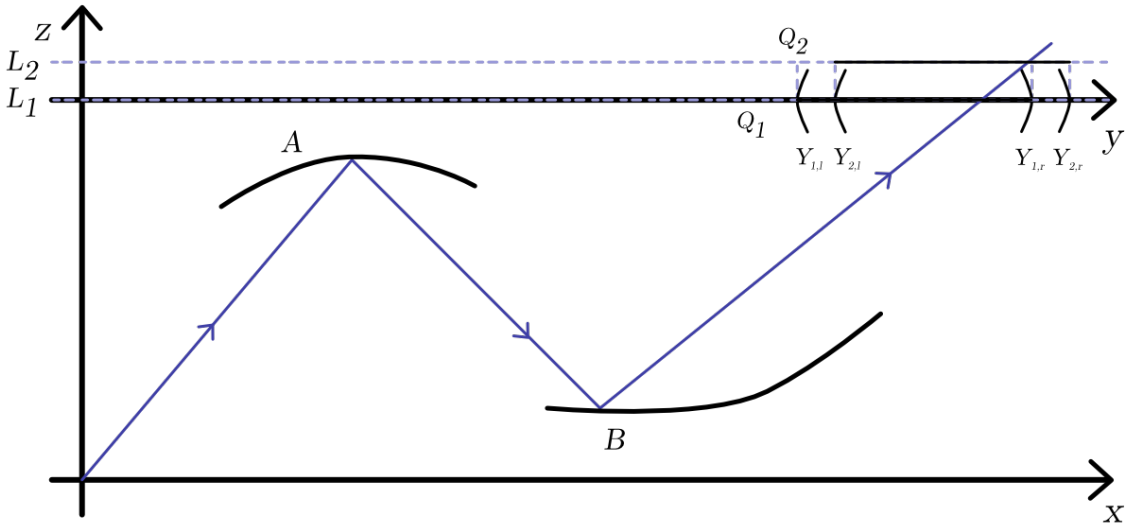


Figure 5: A point source placed in the origin hits two intervals with respective intensity functions. The entering angles at the first are implicitly given by the illumination intensity function of the second interval.

Let $\big(s_1(\theta), s_2(\theta)\big)$ be the normalized emitting direction vector. The mirror A is being described in dependence of θ and the scaling $u(\theta)$, so $A(\theta) = \left\{ u(\theta) \cdot \begin{pmatrix} s_1(\theta) \\ s_2(\theta) \end{pmatrix} \middle| \theta_l \leq \theta \leq \theta_r \right\}$

5 Validation - Ray tracing

We can test the accuracy of our reflectors by simulating paths of light rays and comparing them to the calculated solution. This validation is done separately from the other calculations and the reflectors are defined as sets of line segments instead of continuous functions.

5.1 Splines

5.1.1 Piece-Wise Affine Reflectors

We use a simple ray tracing algorithm to compute the paths. A light ray can be defined as a line segment from point $\boldsymbol{A}$ to point $\boldsymbol{B}$ and reflector segment as a line segment from point $\boldsymbol{C}$ to point $\boldsymbol{D}$. Point $\boldsymbol{B}$ is an arbitrary point from starting point $\boldsymbol{A}$ to the direction of the ray far enough away from $\boldsymbol{A}$ so that $\overline{\boldsymbol{AB}}$ intersects with $\overline{\boldsymbol{CD}}$. This is illustrated in Figure 6.

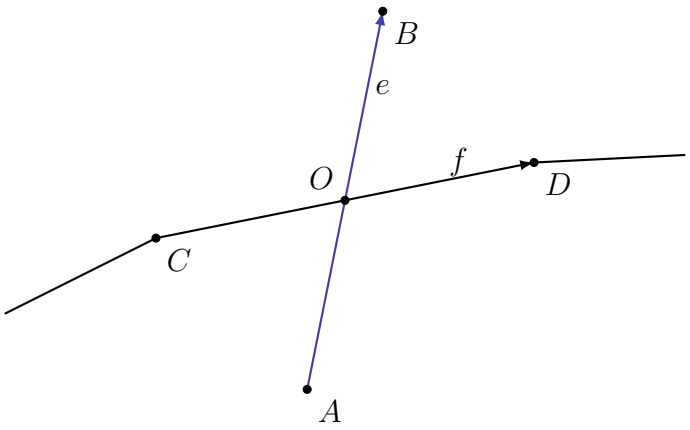


Figure 6: Light ray $\overline{\boldsymbol{AB}}$ intersecting with reflector line segment $\overline{\boldsymbol{CD}}$

First, we must find the reflector segment that intersects with the light ray by calculating value for h:

$$\begin{aligned} \boldsymbol{e} &= \boldsymbol{B} - \boldsymbol{A}, \\ \boldsymbol{f} &= \boldsymbol{D} - \boldsymbol{C}, \\ \boldsymbol{p} &\perp \boldsymbol{e}, \\ \boldsymbol{h} &= \frac{(\boldsymbol{A} - \boldsymbol{C}) \cdot \boldsymbol{p}}{\boldsymbol{f} \cdot \boldsymbol{p}}. \end{aligned}$$

This is repeated for every reflector segment until $0 \leq h \leq 1$, which means that the intersection has been found and the point of intersection $\boldsymbol{O}$ is

$$\boldsymbol{O} = \boldsymbol{h}f.$$

We can now calculate the direction vector $\boldsymbol{r_{ref}}$ of the reflected ray for the intersecting reflector segment:

$$\begin{aligned} \boldsymbol{n} &\perp \boldsymbol{f}, \\ \boldsymbol{r_{ref}} &= \boldsymbol{e} - \frac{2\,(\boldsymbol{e} \cdot \boldsymbol{n})\,\boldsymbol{n}}{||\boldsymbol{n}||^2}. \end{aligned} \tag{12}$$

Vector reflection is illustrated in Figure 7. Same steps are repeated in order to find intersection point and reflection vector for the second reflector. Knowing the starting point, the intersection points and the reflection vectors we can finally plot the ray-traced path of a light ray and compare it to the desired one.

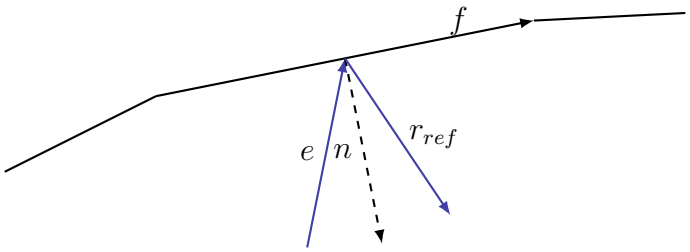


Figure 7: A light ray (blue) is being reflected by a piece-wise affine reflector $\boldsymbol{f}$.

5.1.2 Piece-Wise Cubic Reflectors

Cubic splines have incredibly desirable properties [Deuflhard and Hohmann 2008] and can be calculated numerically almost effortlessly using the *SciPy* Python library. Behind the scenes, spline interpolation solves a *tridiagonal matrix* which is solvable in linear time [Sander 2016] with the *Thomas Algorithm* [Lee 2011]. Instead of simply connecting points for which we solved the differential equation with line segments, our approximation will always be better if we interpolate third degree polynomials which share function value, first and second derivative at the knots.

5.2 Error Estimates For Spline Reflectors

In the following, we assume the source density to emit rays parallelly and at direction $\boldsymbol{s} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$. We abuse this property to ensure that the shape of an reflector $R = \left\{ \big(x, u(x)\big) \middle| x \in D_E \right\}$ is parameterized straight from below, a property which is extremely practical as the shape could now be interpolated by a spline. Assume a straight, parallelly emitting light source. Then, the (first) reflector is expressed as a continuous set of points $\big(x, u(x)\big)$. A ray emitted at $x = x_0$ hits the reflector at the same x-value. Let the target domain be shifted to then,

- **best estimate with the infinity norm:** Let the interpolated function $f \in \mathcal{C}^4([a, b])$ and let s be the interpolating spline.

$$||f - s||_\infty \leq \frac{5}{384} h^4 \cdot ||f^{(4)}||_\infty. \tag{13}$$

  We take the result directly from [Hall and Meyer 1976] and use it without proof.

- **weaker estimate using the infinity norm:** As p-norms are monotonically increasing in value, $||\cdot||_p \underset{p\to\infty}{\nearrow} ||\cdot||_\infty$, we can use estimates for p-norms and, by estimating downwards again, may use those inequalities and replace their respective

p-norm with the ∞-norm. For instance, we can obtain an inequality which uses the $||\cdot||_2$-norm from [Sander 2016] and weaken the result

$$||f - s||_\infty \leq \frac{1}{2} h^{3/2} \cdot ||f^{(2)}||_2 \leq \frac{1}{2} h^{3/2} \cdot ||f^{(2)}||_\infty \tag{14}$$

to obtain a very weak estimate which is not sharp for all non-constant interpolated functions f.

Fortunately, the mirrors are solutions of differential equations and therefore, first derivatives of the interpolated functions are known. Estimates for the second derivative can be calculated numerically using finite differences or even analytically if reasonable.

We assume that light sources are not transparent, i.e. a rays cannot travel through a light source. Especially, they are prohibited from travelling through a light source in opposite direction of emission. Therefore, we would like to introduce some limitations to the setting of our problem to ensure that the solutions of our differential equations are in fact feasible.

5.2.1 Lemma: A Priori Feasibility Of Starting Values For Near-Field Problems

Let $[x_l, x_r]$ be a parallel light source emitting light in positive z-direction. Let L be the target level and let $[y_l, y_r]$ be the target interval. Let $E : [x_l, x_r] =: D_E \to (0, \infty),\ x \mapsto E(x)$ be the source density, let $G : [y_l, y_r] =: D_G \to (0, \infty),\ y \mapsto G(y)$ be the target density and $m(x)$ the mapping. Let $\int_{D_E} E(x)\, dx = \int_{D_G} G(y)\, dy$.
Then,

(i) if $L < 0$ and m is monotonously increasing: The starting values are feasible if and only if

$$y_l > x_l + \frac{x_r - x_l}{u_0}(u_0 - L)$$

or equivalently

$$y_l > x_r + \frac{x_r - x_l}{u_0}(-L)$$

(ii) if $L < 0$ and m is monotonously decreasing: The starting values can only be feasible if

$$y_r > x_l + \frac{x_r - x_l}{u_0}(u_0 - L)$$

Note that this is necessary but not a sufficient condition for feasibility.

(iii) if $L = u_0$ and m is monotonously increasing: The starting values are infeasible.

Proof

(i) The expressions are obviously equivalent.
" $\Longleftarrow$ ": Let the inequality hold. Then, no rays intersect with the light source by simple geometric arguments.
" $\Longrightarrow$ ": Let m be monotonously increasing. In this case, the rays do not cross

and maintain their angular half order. By the assumption of energy conservation, we obtain $m(x_l) = y_l$. The starting values therefore are feasible if the reflected left most ray does not intersect with the source.
The left most ray hits the reflector in the point $\big(x_l,\, u(x_l)\big)$. Between the reflector and the target, the ray travels along the line segment

$$g : d \cdot \begin{pmatrix} y_l - x_l \\ L - u(x_l) \end{pmatrix} + \begin{pmatrix} x_l \\ u(x_l) \end{pmatrix},\ d > 0.$$

We check the hypothetical path a ray would travel if it would hit the right corner of the source:

$$\gamma : d \cdot \begin{pmatrix} x_r - x_l \\ -u(x_l) \end{pmatrix} + \begin{pmatrix} x_l \\ u(x_l) \end{pmatrix},\ d > 0,$$

which gives us the proposed inequality.

(ii) To find a proof analogue to (i), one would require a priori knowledge of $u_r := u(x_r)$ (similar to the knowledge of $u_0 := u(x_l)$). There is no obvious way how to obtain this value without solving the ODE.

(iii) by assumption, the angles at the target must not contain 0. However, if $u_0 = L$, the ray emitted from x_l may only hit the target at $\alpha = 0$, and therefore the assumption is broken. □

5.2.2 Estimates for single reflector, near-field targets

For single reflector problems, we do fortunately not have any cross-dependence between the mirror differential equations. Assume the starting values to be feasible according to Lemma 5.2.1(i).
The reflector u is then given as the solution to the initial value problem

$$\begin{cases} u'(x) & = \frac{m(x)-x}{\sqrt{(x-m(x))^2+(L+u(x))^2}+L+u(x)} \\ u(x_l) & = u_0 \end{cases}$$

By assumption, m does not invert the order of the rays and therefore $u(x)$ is monotonously increasing as a consequence of Lemma 5.2.1(i) because $y_r > y_l$ to create a valid, non-empty interval. As the source emits in positive z-direction, $u_0 > 0$ and the monotonicity of u (which is a direct consequence of $y_l > x_r \implies m(x) > x\, \forall x \in [x_l, x_r]$ and the assumed non-crossing property of the mapping function and implies $u' > 0$), one can obtain $u(x) \geq u_0 > 0\, \forall x \in [x_l, x_r] =: D_E$ with equality only in $u(x_l)$.
Applying all of those observations, we find estimate lower and upper boundaries (which are not sharp on (x_l, x_r) the interior of the domain):

$$\begin{aligned} 0 < u'(x) &= \frac{m(x)-x}{\sqrt{(x-m(x))^2+(L+u(x))^2}+L+u(x)} \leq \frac{m(x)-x}{\sqrt{(x-m(x))^2+L^2}+L} \\ &\leq \frac{m(x)}{\sqrt{m(x)^2+L^2}+L}. \end{aligned} \tag{15}$$

However, this estimate is not bounded for some unfortunate choices of parameters. Furthermore, the shape of the reflector u does not depend on the distance between the

(parallel) light source and the reflector. For geometrical reasons, the shape of u is invariant under translations of the source on the z-axis on the interval $(-\infty, u_0)$.
We now search for a lower and an upper, both very naive boundary for the term $m(x)-x$ which we can then insert back into the larger equation afterwards: $m(x) - x < y_r - x_l$ and $\big(x - m(x)\big)^2 > (x_r - y_l)^2$. Then, we can estimate the derivative less naively as

$$\begin{aligned} u'(x) &= \frac{m(x)-x}{\sqrt{(x-m(x))^2 + (L+u(x))^2} + L + u(x)} < \frac{y_r - x_l}{\sqrt{(x-m(x))^2 + L^2} + L + u_0} \\ &< \frac{y_r - x_l}{\sqrt{(x_r - y_l)^2 + L^2} + L + u_0} \end{aligned} \tag{16}$$

As this term only includes constants and the denominator is strictly larger than zero, u' is in fact bounded and we can use the terribly naive estimate to describe a very weak estimate of an upper bound of the error of the reflector u.
In some (or potentially in all?) cases, we might find a better estimate of the $m(x) - x$ term

$$\begin{aligned} 0 \leq m(x) - x = \int_0^x m'(\xi) - 1 \, d\xi &\leq \int_0^x ||m'||_\infty - 1 \, d\xi \\ &= x \cdot ||m'||_\infty - x = x \cdot \big(||m'||_\infty - 1\big) \end{aligned} \tag{17}$$

Gathering information about m' estimates
The mapping function m' (cf. (3)) has an implicit formulation of the derivative. Let us first consider easy cases of constant densities with affine transformation functions and increase in complexity from there on.
A few observations of the mapping function will aid greatly in simplifying calculations.

- The respective domains D_E and D_G are closed or open such that they could be extended to the closure continuously.
- Translating the density functions E and G along the x (or respectively y) axis does not affect $||E||_\infty$ and $||G||_\infty$

$\implies$ Translate G so that $\inf D_E = \inf D_G = 0$.

- For constant densities $E = E(x)$, $G = G\big(m(x)\big)$, the derivative $m'(x) = \frac{E}{G}$ is constant and therefore, $||m'|| = \frac{E}{G}$.
- If the target density G is constant but the source density E is not, one can obtain the estimate

$$\begin{aligned} ||m'(x)|| &= \frac{\lambda(E)}{\lambda(G)} \cdot |E(x)| \\ ||m'|| &\leq \frac{\lambda(E)}{\lambda(G)} \cdot ||E||, \end{aligned} \tag{18}$$

  where equality is reached at the maximum(s) of E.

5.3 Ray Tracing Errors - Illumination Errors

To describe the accuracy of our ray tracing, we must find a way to describe the error of our ray-traced solution.

5.3.1 Definition: Axioms For Errors

Some basic properties of the error definition should include:

(i) Exact solutions should have error zero.
(ii) Ray tracing is an agent-based verification model in the sense that it takes samples from a continuous "population" of rays which all have an individual error. When we increase the size of the sample, the error should converge.
(iii) The previous statement shall be true regardless of the distribution of the sample. However, the limit generally is not the same.
(iv) Errors must not be negative.
(v) The sample-based error must be identical to the individual error if the sample size is trivial ($n = 1$).

This motivates something like "defining the error as the distance between the calculated hit of the target and the actual target". However, we cannot be entirely sure that the rays actually do hit the target.
If rays are reflected at the wrong angle, for instance because the machine precision was reached already, the deviation from the desired target has another important component: The angle of the ray will most deviate from the computed values in small amounts as well. If we study cases of two target distributions at different levels (cf. Problems 2 and 4), the error at both levels will most likely not be the same.

Angular errors
In this paper, we ignore the angular error and only address the error at the first level instead. We refer to two densities as "similar", if they can be transformed into one another using no other operations than translation and stretching. For two such target densities, we are hopeful that arguments could be found that they share the same error. Potentially, the Intercept Theorem could play a role in a potential proof.

5.3.2 Extrapolated Ray Tracing Error *(ERTE)*

To determine the accuracy of the ray tracing, we extrapolate the shape of the target to ensure that we find an easy to compute way of dealing with rays missing the target:

(i) Interval targets: The target t is an interval on the y-axis at level $z = L$. So basically, the target is the image of a function $f : \mathbb{R} \to \mathbb{R},\ y \mapsto L$ restricted on the domain D_G, e.g. $t = \operatorname{Img} f|_{D_G}$. Note that one could have chosen $f : \mathbb{R} \to \mathbb{R},\ x \mapsto L$ and would have come to the same result.
To extrapolate the target, we simply get rid of the restriction.
(ii) Point targets: Point targets unfortunately have a density in stereographic coordinates. The target is a point $t = (t_x, t_z)$ and can be described as the image of a restricted function $f : \mathbb{R} \to \mathbb{R},\ x \mapsto L$, however, this time restricted on t_x: $t = \operatorname{Img} f|_{t_x}$.
Again, we get rid of the restriction to extrapolate the shape of the target and ensure that we find a method to deal with rays which miss the target.

Definition: Extrapolated Ray Tracing Error *(ERTE)* We assume a system of mirrors which at most has one ray travelling parallel to the (extrapolated) target. Define the ERTE for single rays and a sample of rays as

(i) Single ray: Let t_r be the calculated hit of the target of a ray emitted from somewhere in D_E and $\tilde{t}_r$ be the ray-traced hit of the (extrapolated) target. Then,

$$\begin{aligned}\varepsilon_r := d(t_r, \tilde{t}_r) &= \sqrt{(t_{r_1} - \tilde{t}_{r_1})^2 + (t_{r_2} - \tilde{t}_{r_2})^2} = \sqrt{(t_{r_1} - \tilde{t}_{r_1})^2} \\ &= |t_{r_1} - \tilde{t}_{r_1}|.\end{aligned}$$

(ii) Sample of rays: For a sample of rays $r = 1, 2, ..., n$, we define the error as

$$\varepsilon_n := \frac{\sum_{i=1}^{n} \varepsilon_i}{n}. \tag{19}$$

This error definition satisfies 5.3.1 (i), is obviously positive and satisfies 5.3.1 (iv), and satisfies 5.3.1 (v).
A bit less obvious are 5.3.1 (ii) and (iii): We introduce a case differentiation based on the supremum of the single ray errors: Let $\varepsilon_{\text{sup}} := \sup_{r \in D_E} \varepsilon_r$. If $\varepsilon_{\text{sup}} = \infty$, we set $\varepsilon := \infty$. If $\varepsilon_{\text{sup}} < \infty$, things are a bit more complicated.
The axioms (ii) and (iii) will be studied later as they require more theoretical work to be done before.

5.3.3 Definition: Minimal Distance Ray Tracing Error *(MIRTE)*

While the ERTE was introduced with the intuition of generalizing from interval targets to point targets, one might take the different approach of starting out with a generalizable error definition for point targets.
Interpret the path of a ray between the last contact with a mirror and infinity as a line $r \subsetneq \mathbb{R}^2$.

(i) Single ray: Let t_r be the calculated hit of the target of a ray r. Then,

$$\varepsilon_r := \text{infdist}(t_r, r).$$

(ii) Sample of rays: For a sample of rays $r = 1, 2, ..., n$, we define the error as

$$\varepsilon_n := \frac{\sum_{i=1}^{n} \varepsilon_i}{n}. \tag{20}$$

Again, this error definition satisfies 5.3.1 (i), is positive and satisfies 5.3.1 (iv), and satisfies 5.3.1 (v). We continue to show that this error converges for any underlying distribution after introducing a practical Lemma:

5.3.4 Lemma: Continuity Of The Ray Traced Reflection Projection Of Smooth Reflectors

Let Δ be a partition and let $s \in S_3^2(\Delta)$ be a C^2-spline. Then, the projection of a reflection of a convex, star shaped, parallel light source is continuous.

Proof
In this proof, we just show the two-dimensional case as this paper focuses on two-dimensional problems only.
By the construction of splines, s' is continuously differentiable on its whole domain. But

as the slope of the function s is continuously differentiable, the normal of the slope is continuously differentiable as well: Construct the function of the tangent vectors

$$\boldsymbol{\tau}(s') := \begin{pmatrix} \arcsin s' \\ \arccos s' \end{pmatrix}$$

and the function of the normal vectors as

$$\begin{aligned} \boldsymbol{\nu}(s') &:= \begin{pmatrix} \cos\frac{-\pi}{2} & -\sin\frac{-\pi}{2} \\ \sin\frac{-\pi}{2} & \cos\frac{-\pi}{2} \end{pmatrix} \cdot \boldsymbol{\tau}(s') = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \cdot \tau(s') \\ &= \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \cdot \begin{pmatrix} \arcsin s' \\ \arccos s' \end{pmatrix} = \begin{pmatrix} \arccos s' \\ -\arcsin s' \end{pmatrix} \end{aligned}$$

Therefore, the slope of the normals can be expressed as a function n' of s' with $n' := \sin\big(\arccos s'(x)\big)$, which is continuous. Recall Equation 12: for e fixed, $\sin\big(r_{ref}\big)$ is continuous in n. Hence, the projection onto a plane (which is not parallel to any ray) is continuously differentiable. □

5.3.5 Theorem: Convergence Of The MIRTE

Let $\varphi : D_E \to \mathbb{R}_+,\ x \mapsto \beta \cdot E(x)$ be a strictly positive, continuous and integrable probability density. Let $\Phi(x)$ be the CDF of φ. Let

$$\Delta_n := \left\{ \frac{\min(D_E) + 0 \cdot \lambda(D_E)}{n}, \frac{\min(D_E) + 1 \cdot \lambda(D_E)}{n}, ..., \frac{\min(D_E) + n \cdot \lambda(D_E)}{n} \right\}$$

be an ordered partition of the domain of φ. Call the i-th element of Δ_n δ_{n_i}.
We define the expressions $\overline{I}_n(\varphi) := \sum_{i=1}^{n} \varepsilon_i \cdot \varphi(\delta_{n_i})$ and $\underline{I}_n(\varphi) := \sum_{i=1}^{n} \varepsilon_{i-1} \cdot \varphi(\delta_{n_{i-1}})$.
Then, there exists a finite limit $I(\varphi) = \lim_{n\to\infty} \overline{I}_n(\varphi) = \lim_{n\to\infty} \underline{I}_n(\varphi)$.

Proof
Ignoring the factors ε_i, I is defined as the Riemann integral on the domain of φ. As φ is a probability density, φ is integrable and $\Phi\big(\max(D_E)\big) = \lim_{x\to\infty} \Phi(x) = 1$. Especially, I exists. Reintroducing the factors ε_i, it is important to acknowledge a few of its properties: As MIRTE satisfies the positivity axiom from Definition 5.3.1, all addends in the $\overline{I}$ and $\underline{I}$ terms remain positive. Additionally, MIRTE is bounded: even rays travelling parallel to a target interval have a (finite) minimum distance to their calculated hit point!
As we defined ε_r for every ray emitted from the source, we can define a function $\tilde{\varepsilon} : D_E \to \mathbb{R}_+,\ r \mapsto \varepsilon_r$. We use Lemma 5.3.4 to argue that the smooth spline reflectors from Section 5.1.2 make the function $\tilde{\varepsilon}$ continuous because both the projection of the ray tracing and the calculated target distribution are continuous. As a site note, the smoothness of the reflectors is an essential property without which this proof would fail.
It is known that it the error function $\tilde{\varepsilon}$ is bounded, too, from the fact that ε_r is bounded. So, $\tilde{\varepsilon}$ is continuous and integrable. As the product of two bounded, integrable functions is bounded and integrable [Feldman 2016], $\tilde{\varepsilon} \cdot \varphi$ is integrable and the limit I exists. We define $\varepsilon := I(\varphi)$ and call it the error of the ray tracing. □

5.3.6 Convergence Of The ERTE

We gather numerical indication for convergence instead of giving a rigorous proof of its convergence. Implementation details and results are indicated by the illustrations in Figure 8.

5.3.7 Application

We will stick with the ERTE for the sake of implementation simplicity even though another formulation of the error might sound more promising. We observed a numerical indication both intuitive definitions of errors converge if and only if the other one converges, therefore we argue that the usage of the ERTE error is reasonable.

6 Numerical Implementation

For the implementation of a numerical solver, we decided to go with the programming language Python due to its wide adoption and adaptability to different use cases.

6.1 The DOPTICS Library

Given the fact that (at least no easily findable) library exists to solve inverse problems in Python, we opted to write an Open Source library of our own [Landwehr et al. 2023]. The library runs on Python 3.11 and the latest versions of *NumPy*, *SciPy* and *SymPy* as of August 2023. While the code base runs on relatively performant Python wrappers around *SciPy* and *NumPy* function, the long term goal is to refactor the pure Python code snippets into *Cython* as well to improve performance for critical applications. At the moment, the *DOPTICS* library consists of solvers for two-dimensional problems only.
We used *SymPy* for expression manipulation and symbolic differentiation and *SciPy* for a numerical ODE solver and numerical integration.

DOPTICS development
The development work could be divided into two major parts:

(i) **Prototyping:** making the solver work with explicit arguments, low fragmentation of code into functions.

(i) **Code base development:** Allowing for maximum flexibility and maintainability by structuring the code well, creating and improving the documentation.

A majority of the work of developing the code base started after the ECMI Modelling Week, meanwhile the prototyping was mostly done during the week.
Much consideration was put into achieving higher accuracy or improved performance at run time even if we would not be able to give proper error estimates yet. The most prominent example for this would be the interpolation in Section 5.1.2.
The code available in a GitHub code base [Landwehr et al. 2023], and there are future plans to turn it into a Python library available for installation with *pip* or *conda*.

6.2 Pseudocode Of The Implementation

This section includes both pseudocoded sections as well as actual code snippets from our production codebase.

6.2.1 Solutions Of The Problems

We provide some code snippets from the *DOPTICS* repository to illustrate how we solve the problems from Section 4. The goal of our implementation is to be accessible as possible the broader public. We provide methods for all different problems which solve all possible situations of crossing and non-crossing rays for the user.
A call to one of the solvers might look like the following:

```
# import the two mirror solver package as tms
import src.ode_solving.two_mirrors as tms
# simply call the solver with your parameters
result = tms.solve_two_mirrors_parallel_source_two_targets(
        starting_density=func.uniform,
        target_distribution_1=func.G1,
        target_distribution_2=func.G1, x_span=x_span,
        y1_span=y1_span, y2_span=y2_span, u0=u0, w0=w0,
        l1=l1, l2=l2,
        # color='#a69f3f',
        number_rays=15
)
```

6.2.2 Ray Tracing And Ray Tracing Error

We first begin by importing the open source libraries *Numpy*, *SciPy* and *Matplotlib*.

```
import numpy as np
import matplotlib.pyplot as plt
import scipy as sp
```

Then, we define functions to calculate the intersection of two lines.

```
def line(p1, p2):
    ...
    return ...

def intersection(l1, l2):
    ...
    if d != 0:  # lines intersect!
        ...
        return x, y
    else:  # lines do not intersect
        return np.nan
```

Then, we continue to define the parameters for our problem. All variable names are identical to the ones used in this report.

```
L1 = 8
L2 = 14
y1_span = (10.5, 13.5)
y2_span = (11, 13)
x_span = (0, 3)
```

```
u0 = 4
w0 = 6
E = lambda x: np.exp(-((x - 1.5) / 1)**2 / 2) / (2 * np.pi)**.5
G1 = lambda y1: 1
G2 = lambda y2: 1
pm1 = -1
pm2 = 1
```

We wish must rescale at least one of the densities and find it reasonable to interpret both densities as probability densities. We rescale the initial densities E and $G1$ and call the "E_fixed" and "G_fixed" respectively.

```
if __name__ == '__main__':
    # calculate theoretical target from the mapping between
    # emitting density E and target density G1
    G1_int = sp.integrate.quad(G1, y1_span[0], y1_span[1])[0]
    G1_fixed = lambda y1: G1(y1) / G1_int
    E_int = sp.integrate.quad(E, x_span[0], x_span[1])[0]
    E_fixed = lambda x: E(x) / E_int

    m1_prime = lambda x, m: pm1 * E_fixed(x) / G1_fixed(m)
    y1_0 = y1_span[1]
    m1_solved = sp.integrate.solve_ivp(m1_prime, x_span, [y1_0], dense_output=1)
    m1 = lambda x: m1_solved.sol(x)[0]
```

When the solution for a given set of parameters has already been calculated, we can save computation time by importing the results from a numpy archive file.

```
    # absue that solution was calculated already by importing mirror points
    reflectors = np.load("reflectors.npz")
    A = reflectors["arr_0"]  # A is discrete reflector as solved by ODE
    B = reflectors["arr_1"]  # B is ...
```

We run the error calculation with an increasing number of rays emitted from the source and save in error ε_n to an array to plot it later and study if ε_n appears to converge towards an ε.

```
    RUNS = 1000
    errn = np.zeros(RUNS)
    for n_rays in range(1, RUNS):
        print(n_rays)
        xis = np.linspace(0, 3, n_rays)
        targets = m1(xis)
        # First reflector
        Ps = np.zeros((n_rays, 2))
        As = np.zeros((n_rays, 2))
        us = np.zeros((n_rays, 2))
        for i, x in enumerate(xis):
            for j in range(A.shape[0] - 1):
                a = np.array([x, 0])
                b = np.array([x, 5])
```

```
            c = A[j, :]
            d = A[j + 1, :]
            e = b - a
            f = d - c
            p = np.array([-e[1], e[0]])
            h = np.dot(a - c, p) / np.dot(f, p)
            if 0 <= h <= 1:
                o = c + f * h
                n = np.array([-f[1], f[0]])
                n /= np.linalg.norm(n)
                us[i] = e - 2 * np.dot(e, n) * n
                As[i] = o
                Ps[i] = a
                break
```

After filling in the values for the first reflector, we move on to the second reflector analogously.

```
# Second reflector
for i in range(n_rays):
    for j in range(B.shape[0] - 1):
        ...
        if 0 <= h <= 1:
            ...
            break
```

As the rays were calculated, we can start to calculate the distance between the calculated intersection and the actual intersection with the extrapolated target line to determine the ERTE as in Definition 5.3.2.

```
error = np.zeros(n_rays)
for i in range(n_rays):
    ...
    intersec = intersection(target_line, ray)
    try:
        error[i] = np.abs(targets[i] - intersec[0]) * E(xis[i])
    except TypeError:
        error[i] = np.nan
indices_to_drop = np.argwhere(error == np.nan)
error = np.delete(error, indices_to_drop)
error = error / len(error)
errn[n_rays] = np.nansum(error)
```

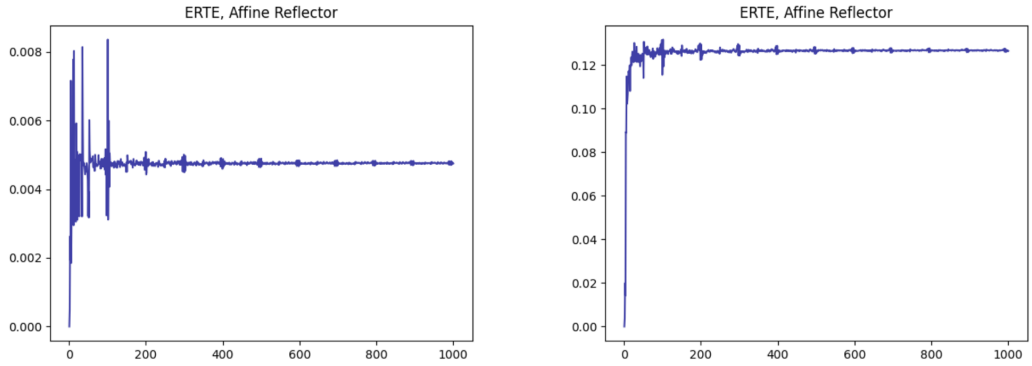
Finally, we can plot the development of the error.

```
plt.plot(np.arange(1, RUNS+1, 1), errn, COLORS['szegedblue'])
plt.title('ERTE, Affine Reflector')
plt.show()
```

Eventually, the plots can be used in the report, for instance in Figure 8.

6.3 ERTE Implementation

Whenever we think about errors, it is important to try and make sure that the deviance from the calculated targets and the ray-traced target hits comes from similar sources only. For instance, for both interval sources and point sources respectively, we expect a dependency of the error on the derivatives of the source and target densities. But as we need to rescale the densities such that their integrals match, we are faced with situation in which we *must* multiply the factors with a scalar already. Our initial The code is undergoing a refactoring process with the aim to turn all densities into probability densities on closed intervals.



(a) The converges but may face periodic "bumps" which decrease in size.

(b) ERTE seems to also converge if the result is false. Its limit is relative large.

Figure 8: Implementation of the ERTE error shows clear indications of numerical covnergence.

7 Results

7.1 Problem 1: Results

Numerical solutions were obtained for all problem cases. The solution for Problem 1 presented in Figure 9 is computed using values $[x_l,\, x_r] = [1,\, 2]$, $[y_l,\, y_r] = [3,\, 7]$, $L = -1$, $u_0 = 4$, $E(x) = 1$ and $G(y) = 0.4 \exp\left(-0.4\,(y-5)^2\right)$.

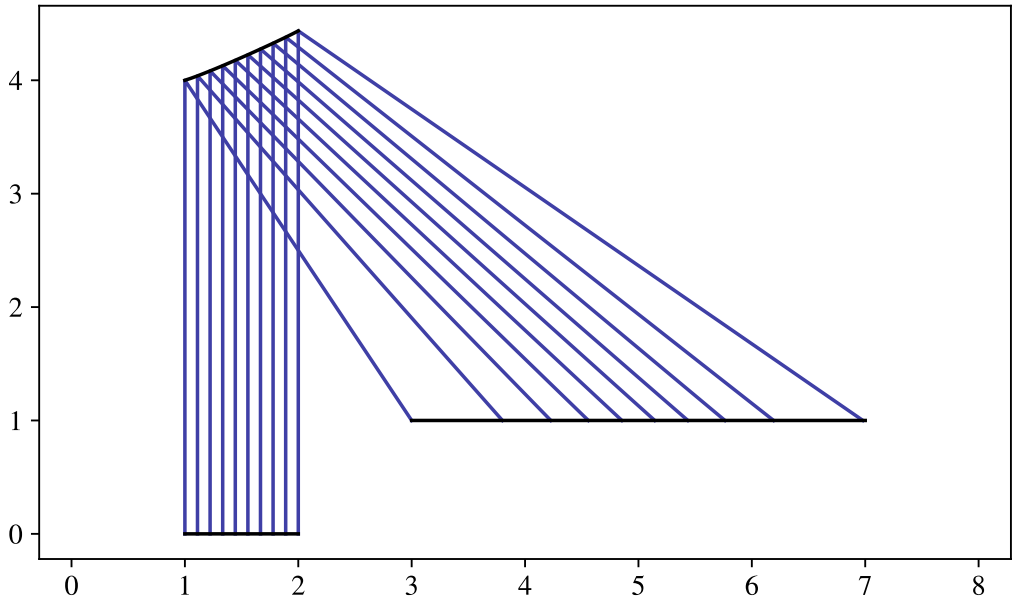


Figure 9: Problem 1 example solution

7.2 Problem 2: Results

The solution for Problem 2 in Figure 10 is computed using values $[x_l,\, x_r] = [0,\, 3]$, $[y_{1,l},\, y_{1,r}] = [10.5,\, 13.5]$, $[y_{2,l},\, y_{2,r}] = [11,\, 13]$, $L_1 = 8$, $L_2 = 14$, $u_0 = 4$, $w_0 = 6$, $E(x) = 0.4 \exp\left(-0.5\,(x-1.5)^2\right)$, $G_1(y_1) = 1$ and $G_2(y_2) = 1$.

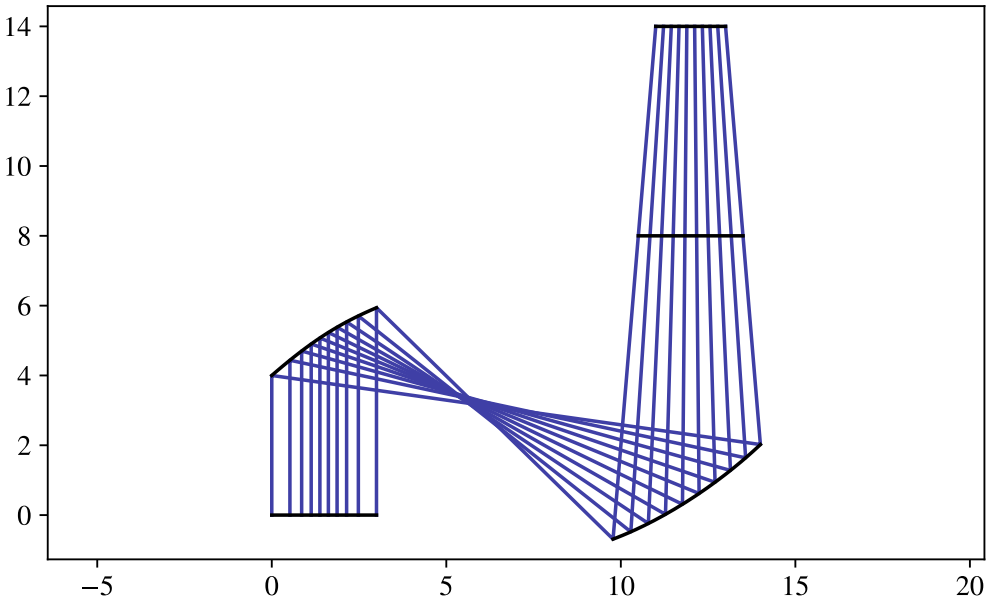


Figure 10: Problem 2 example solution

7.3 Problem 3: Results

The solution for Problem 3 in Figure 11 is computed using values $[\theta_l, \theta_r] = \left[\frac{\pi}{4}, \frac{3\pi}{4}\right]$, $[y_l, y_r] = [4, 7]$, $L = 0$, $u_0 = 5$, $E(\theta) = 1$ and $G(y) = 1$.

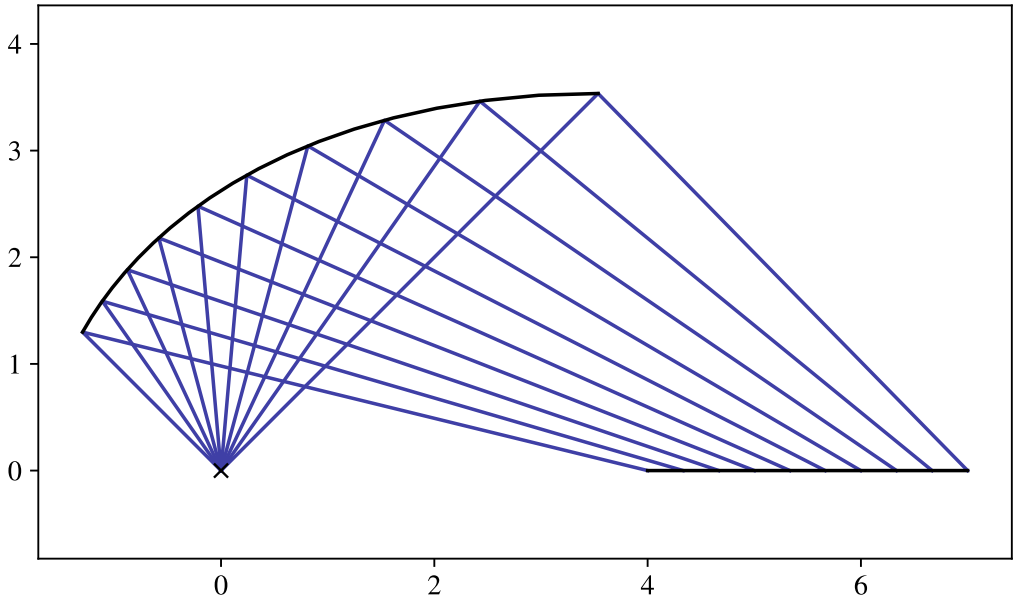


Figure 11: Problem 3 example solution

7.4 Problem 4: Results

The solution for Problem 4 in Figure 12 is computed using values $[\theta_l, \theta_r] = \left[\frac{\pi}{4}, \frac{3\pi}{4}\right]$, $\left[y_{1,l}, y_{1,r}\right] = [6.3, 10.3]$, $\left[y_{2,l}, y_{2,r}\right] = [6, 10]$, $L_1 = 8$, $L_2 = 10$, $u_0 = 7$, $w_0 = 7$, $E(\theta) = 1$, $G_1(y_1) = 1$ and $G_2(y_2) = 1$.

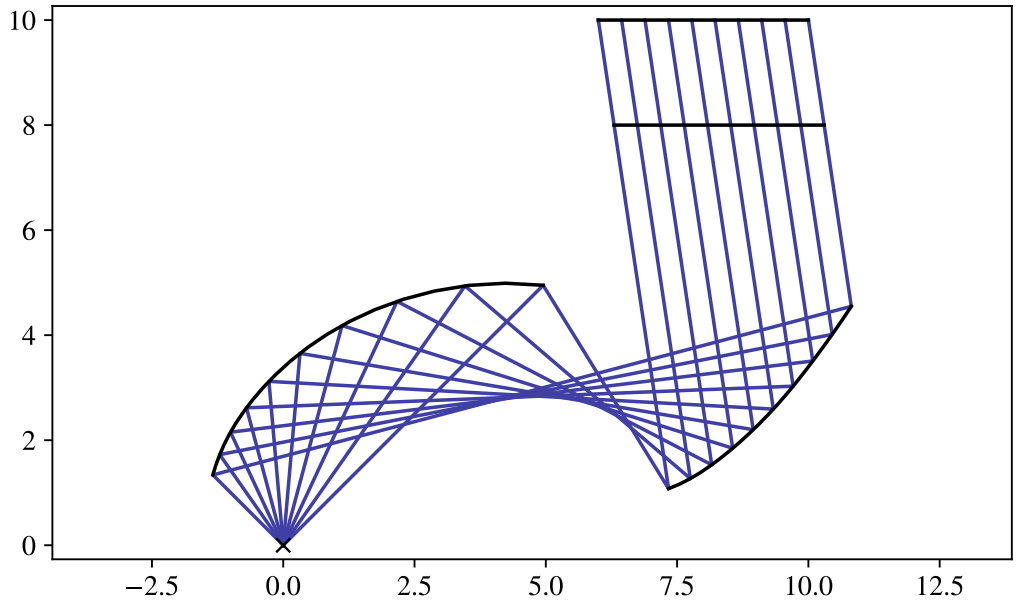
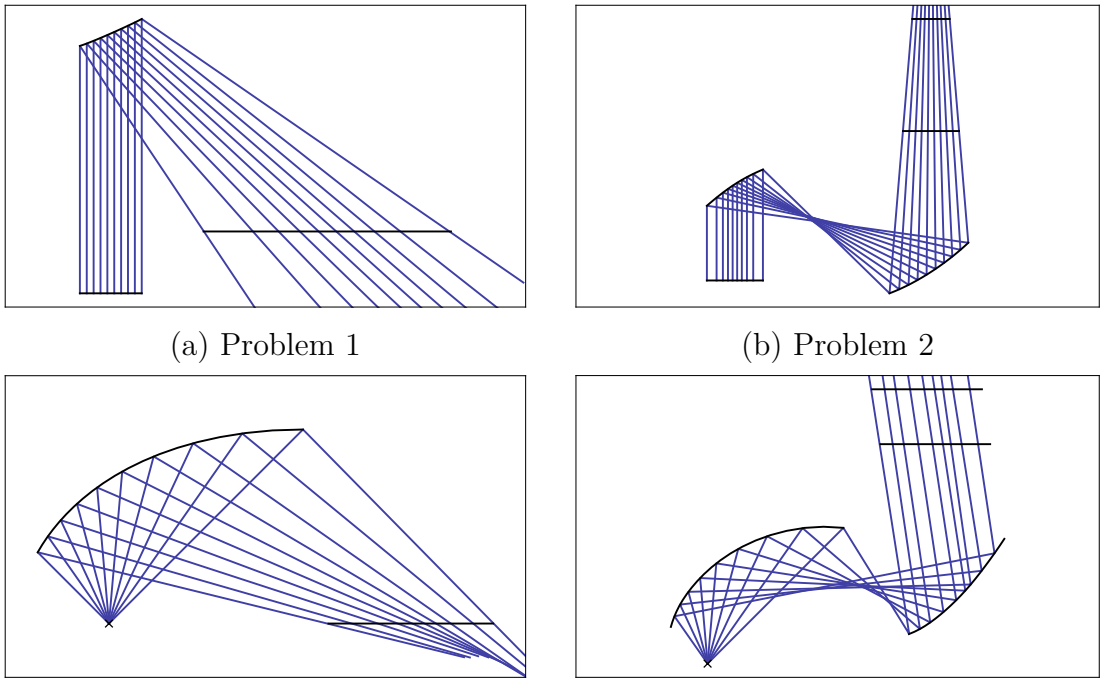


Figure 12: Problem 4 example solution

Accuracy of numerical solutions was tested using ray tracing. Ray-traced paths corresponding to solutions presented above are shown in Figure 13. Simulated paths closely mimic the expected ones. Based on this test, point-source solutions appear to have more error than parallel-source solutions.



(a) Problem 1

(b) Problem 2

(c) Problem 3

(d) Problem 4

Figure 13: Ray tracing the solutions. Especially point sources suffer from relatively high integration errors.

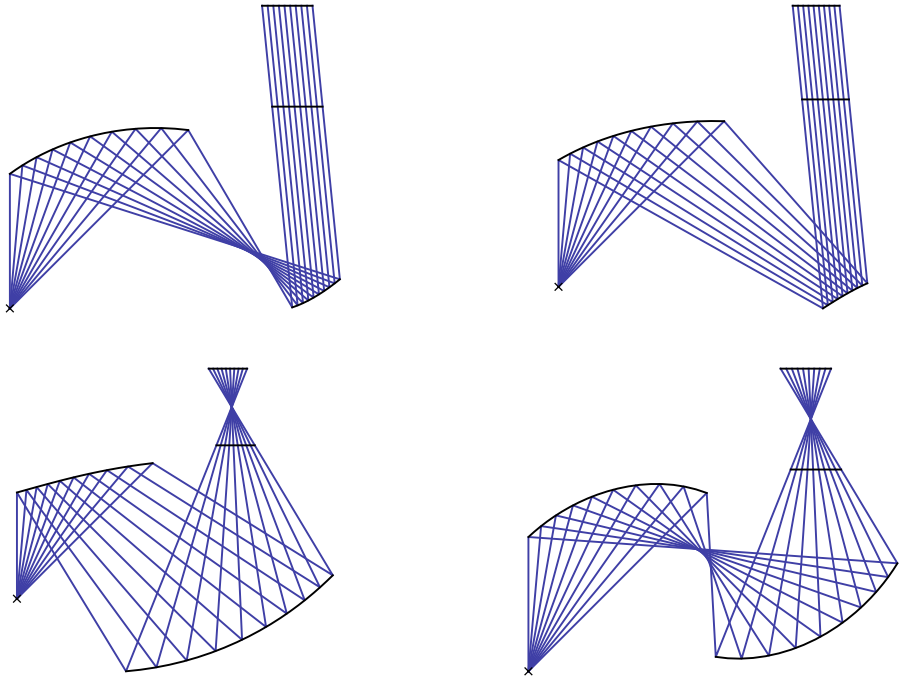


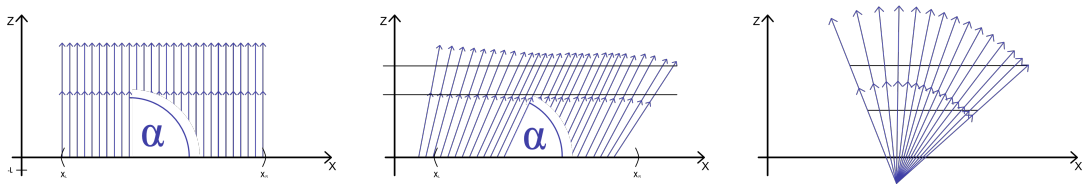
Figure 14: For a given pair of starting values u_0 and w_0, there exist four solutions: rays may or may not cross both between the first and the second mirror and the second mirror and the target.

8 Generalizations In Two Dimensions

An accidental finding during the solving of the challenge of the modeling week led to the observation that at least all mirror near field problems can be rewritten to be special cases of a general form.
Interestingly, in two dimensions, the different differential equations complify or simplify to the same type of ODE respectively. This way, a reformulation of the problems in terms of a directed density, we were able to reduce the number of classifications of near-field optical systems as classified by Anthonissen et al. 2021 from six to one generalized form. Potentially, far-field targets could also be solved by the same approach.

8.1 Directed Densities

The sources and targets proposed by [Anthonissen et al. 2021] shared the property that for all planes intersecting the rays in (positive) direction of travel the "ordering of the rays" was either maintained or inverted. This property is necessary to ensure that the mapping functions m as in Chapter 4 are continuous. Additionally, from here on we also treat source densities as probability densities, i.e. WLOG we assume $\int_D \phi(x)\,dx = 1$ for all densities encountered.



(a) a parallel source: $\alpha = 90°$

(b) two densities: $\alpha = \phi(x)$ is a function of x

(c) intersecting with planes leads to two densities

Figure 15: Illustrations of some light sources indicate how the interpretation as directed densities can be beneficial.

While the more extreme, outlying cases of point sources (cf. Figure 15c) and parallel sources (Figure 15c) obviously maintain this property, the transformation from parallel sources to point sources could be interpreted as follows:

(i) Intersect the rays with a (normal) plane P.

(i) Calculate the median of the source density: $\bar{S} = \frac{S_r + S_l}{2}$

(i) Iteratively shrink the source interval by a shrinking factor $\lambda \in (0, 1)$ while maintaining the density at the plane P.

(i) The point source case is then the limit of the iterative shrinking process.

All the iteratively shrinked source densities maintain ray ordering at the intersecting plane. This motivates further study of directed (free-form) light sources which maintain ray ordering at intersecting planes.
However, this should generally hold for directed densities which maintain ray ordering - even for sources which are neither straight nor points.

8.2 Generalized, Orthogonally Emitting Sources in $\mathbb{R}^2$

Even though most of this should hold true in $\mathbb{R}^3$, we for the beginning only define and show everything in two dimensions to improve readability.
Let M be a parametrized, continuous curve in $\mathbb{R}^2$ which is assumed to emit rays orthogonally in any point in a specified direction. It quickly becomes obvious that the cases presented in Figure 15 can all be transformed into this setting.

8.2.1 Point Light Sources As Orthogonally Emitting Sources

To express point light sources as orthogonal sources, view their projection on balls with radius r.

Unit circles
Let $S := (s_1, s_2)$ be a directed light source with angular density $\varrho : D \subsetneq [-\pi, \pi] \to \mathbb{R}_+$ measurable and continuous with $\lambda\big(\text{supp}(\varrho)\big) < \pi$, $\varrho > 0$, and with $0 \notin \text{supp}(\varrho)$. Let D be open. Then, the segment of the unit circle which S projects on can be described as a manifold M with chart

$$\left(\text{supp}(\varrho), \phi_1 : \xi \mapsto 1 \cdot \begin{pmatrix} \sin(\xi) \\ \cos(\xi) \end{pmatrix} + \begin{pmatrix} s_1 \\ s_2 \end{pmatrix}\right)$$

The light intensity $\psi(x, z)$ on the lit segment of the unit circle can then be described as $\psi(x, z) = \varrho \circ \phi^{-1}(x, z)$.
Thankfully, the derivative of ϕ simplifies to

$$\phi'(\alpha) = \frac{d}{d\alpha}\begin{pmatrix} \sin(\alpha) \\ \cos(\alpha) \end{pmatrix} + \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} = \begin{pmatrix} \cos(\alpha) \\ -\sin(\alpha) \end{pmatrix},$$

and therefore $||\phi'(\xi)|| = 1 \; \forall \xi$ and ϕ^{-1} can be ignored in the integration.

Circles of different size
Let $\varphi : (\alpha_l, \alpha_r) \to \mathbb{R}^2, \varphi(\alpha) := r \cdot \begin{pmatrix} \sin(\alpha) \\ \cos(\alpha) \end{pmatrix} + \begin{pmatrix} s_1 \\ s_2 \end{pmatrix}$ be a parametrization of a circle segment C with center S and radius r. Define the global illumination intensity function

$$\begin{aligned} \psi :& \mathbb{R}^2 \to \mathbb{R}, \\ & \psi(x, z) = \varrho\Bigg(\arcsin\Big(\frac{x - S_2}{\sqrt{(x - S_1)^2 + (z - S_2)^2}}\Big)\Bigg) \cdot \frac{1}{\sqrt{(x - S_1)^2 + (z - S_2)^2}} \end{aligned} \tag{21}$$

as evaluating ρ at the angle any (x, y) has to S. By implicitly defining ϱ as zero everywhere else, we get the illuminated are I as $I = \text{Img}(\psi)$.
As the derivative of the parametrization now is

$$\varphi'(\alpha) = r \cdot \begin{pmatrix} \cos(\alpha) \\ -\sin(\alpha) \end{pmatrix} \iff ||\varphi'(\alpha)|| = r \cdot \left|\left| \begin{pmatrix} \cos(\alpha) \\ -\sin(\alpha) \end{pmatrix} \right|\right| = r \cdot 1 = r, \tag{22}$$

both $\|\varphi'\|$ and ψ disappear

$$\begin{aligned}\int_{(\alpha_l,\alpha_r)} \varrho(\alpha)\,d\alpha &= \int_C \psi\big(\varrho(\alpha)\big)\cdot||\varphi'(\alpha)||\,d\alpha \\ &= \int_{(\alpha_l,\alpha_r)} \varrho\Big(\arcsin\Big(\frac{x-S_2}{d\big((x,z),\,(S_1,S_2)\big)}\Big)\Big)\cdot\frac{r}{d\big((x,z),\,(S_1,S_2)\big)}\,dx \\ &= \int_{(\alpha_l,\alpha_r)} \varrho\Big(\arcsin\Big(\frac{x-S_2}{d\big((x,z),\,(S_1,S_2)\big)}\Big)\Big)\cdot\frac{r}{r}\,dx \\ &= \int_{(\alpha_l,\alpha_r)} \varrho(\alpha)\,d\alpha\end{aligned} \tag{23}$$

and arbitrarily large, orthogonally emitting circles can now be used as light sources.

Transformation to directed densities along intervals
Instead of studying the illumination at the surface of arbitrary balls with sufficient center, we study the directed energy distribution at an intersecting plane P parallel to $Z=0$ at level L_P in emitting direction. Let $\phi:]\alpha_l,\,\alpha_r[\to P$ be a parametrization of P. P can be described in the form

$$\varphi: D\to\mathbb{R}^2,\ \alpha\mapsto \underbrace{\cos(\alpha)\cdot(L_P-S_2)}_{d\big(S,\,P(\alpha)\big)}\cdot\begin{pmatrix}\sin(\alpha)\\ \cos(\alpha)\end{pmatrix}+\begin{pmatrix}S_1\\ S_2\end{pmatrix} \tag{24}$$

and the derivative turns out to be

$$\begin{aligned}\varphi'(\alpha) &= d\big(S,\,P(\alpha)\big)\cdot\begin{pmatrix}\cos(\alpha)\\ -\sin(\alpha)\end{pmatrix} \\ \iff ||\varphi'(\alpha)|| &= d\big(S,\,P(\alpha)\big)\cdot\left|\left|\begin{pmatrix}\cos(\alpha)\\ -\sin(\alpha)\end{pmatrix}\right|\right| = d\big(S,\,P(\alpha)\big)\end{aligned} \tag{25}$$

Recall the global illumination intensity function

$$\begin{aligned}&\psi:\mathbb{R}^2\to\mathbb{R}, \\ &\psi(x,z)=\varrho\Big(\arcsin\Big(\frac{x-S_2}{d\big((x,z),\,(S_1,S_2)\big)}\Big)\Big)\cdot\frac{1}{d\big((x,z),\,(S_1,S_2)\big)}\end{aligned} \tag{26}$$

Then, the intensity of the directed density in P can be calculated as

$$\begin{aligned}\int_{(\alpha_l,\alpha_r)} \varrho(\alpha)\,d\alpha &= \int_C \psi\big(\varrho(\alpha)\big)\cdot||\varphi'(\alpha)||\,d\alpha \\ &= \int_{(\alpha_l,\alpha_r)} \varrho\Big(\arcsin\Big(\frac{x-S_2}{d\big((x,L_p),\,(S_1,S_2)\big)}\Big)\Big)\cdot\frac{d\big((x,L_p),\,(S_1,S_2)\big)}{d\big((x,L_P),\,(S_1,S_2)\big)}\,dx \\ &= \int_{(\alpha_l,\alpha_r)} \varrho(\alpha)\,d\alpha\end{aligned} \tag{27}$$

Meanwhile, the (angle of the) direction in P can be calculated as $\arcsin\Big(\frac{x-S_2}{d\big((x,L_p),\,(S_1,S_2)\big)}\Big)$. This yields the desired pair of interval source and directed density and we can solve as in section 4.2.

9 Conclusion and Future Research

Our work during and after the week discovered a few novel aspects of the inverse optics design approach in two dimensions. The most relevant should be the generalized interpretation of point and interval sources as special cases of directed densities as discussed in Section 8, especially as in Section 8.1.
The interpretation of reflectors as C^2 splines instead affine splines (as in Section 5.1.2) to reduce the reflection imprecision is also a novel approach which should be perfectly generalizable to three dimensions.

Future work may take further looks into how the findings from $\mathbb{R}^2$ can be generalized to $\mathbb{R}^3$, proof the theoretical convergence of the ERTE and show how our ray tracing error definition converges to zero if and only if traditional ray tracing error definitions converge to zero.

10 Group Dynamic

We had a great group dynamic when we participated in the ECMI Modeling Week in Szeged as a group of five, even though English was not our first language. Throughout the challenging week, we communicated well, worked together smoothly, and formed strong connections, which helped us achieve impressive results.
One of our strengths was that we had people with different backgrounds and skills. Each of us had unique ideas and ways of solving problems. Despite the language barrier, we listened to each other and respected everyone's opinions. This created a friendly and creative environment where we came up with innovative solutions.
We also worked well together as a team. We used our individual strengths and helped each other when we faced difficulties. We tried our best to communicate effectively, even though English was not easy for all of us. We divided tasks and kept each other updated in our regular meetings. This made sure that we all knew what was going on and could make decisions together.
In addition, we had a positive and motivating atmosphere within our group: we balanced our work and enjoyed our time together. We celebrated our accomplishments, laughed together, and appreciated each other's contributions. This positivity not only boosted our spirits but also made us more productive and satisfied as a team.
In summary, our group had a strong dynamic during the ECMI Modeling Week in Szeged: we embraced our differences, worked well together, and created a positive environment. Our experience showed us the power of teamwork in overcoming language barriers and reaching our goals.

References

Anthonissen, Martijn J. H. et al. (Sept. 2021). "Unified mathematical framework for a class of fundamental freeform optical systems". In: *Opt. Express* 29.20, pp. 31650–31664. DOI: 10.1364/OE.438920. URL: https://opg.optica.org/oe/abstract.cfm?URI=oe-29-20-31650.

Deuflhard, Peter and Andreas Hohmann (2008). *Numerische Mathematik. Eine algorithmisch orientierte Einführung*. Walter de Gruyter.

Feldman, Joel (2016). *Products Of Riemann Integrable Functions*. URL: https://personal.math.ubc.ca/~feldman/m321/product.pdf.

Hall, Charles A and W Weston Meyer (1976). "Optimal error bounds for cubic spline interpolation". In: *Journal of Approximation Theory* 16.2, pp. 105–122.

Landwehr, Philipp et al. (Aug. 2023). *Philipplndwhr/Doptics: Doptics - differential equation-based optics design tool for Integrated Complex Systems*. URL: https://github.com/philipplndwhr/doptics.

Lee, WT (2011). "Tridiagonal matrices: Thomas algorithm". In: *MS6021, Scientific Computation, University of Limerick*.

Romijn, L. B. et al. (2021). "An Iterative Least-Squares Method for Generated Jacobian Equations in Freeform Optical Design". In: *SIAM Journal on Scientific Computing* 43.2, B298–B322. DOI: 10.1137/20M1338940. eprint: https://doi.org/10.1137/20M1338940. URL: https://doi.org/10.1137/20M1338940.

Romijn, Lotte Bente (2021). "Generated Jacobian Equations in Freeform Optical Design: Mathematical Theory and Numerics". In: *Phd Thesis 1 (Research TU/e / Graduation TU/e), Mathematics and Computer Science*.

Sander, Oliver (2016). *Numerik*. URL: https://gitlab.mn.tu-dresden.de/api/v4/projects/44/jobs/artifacts/master/raw/skript-numerik-sander.pdf?job=compiling.

Schilling, René L (2017). *Measures, integrals and martingales*. Cambridge University Press.