

MASTER

Inverse design of cylindrically symmetric reflector with scattering surfaces

Jiang, S.

Award date:
2017

[Link to publication](#)

Disclaimer

This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain

Inverse Design of Cylindrically Symmetric Reflector with Scattering Surfaces

Master Thesis

Siyuan Jiang

Supervisors:

dr. ir. J.H.M. ten Thije Boonkamp
prof. dr. ir. W.L. IJzerman

Contents

1	Introduction	1
2	Illumination Optics	3
2.1	Lighting terminology	3
2.2	Reflection, refraction and total internal reflection	3
3	The Cylindrically Symmetric Case	5
3.1	Cylindrical coordinate system and energy conservation	5
3.2	Model for the reflector shape	7
3.3	Examples	8
4	Ray Tracing	12
4.1	The procedure of ray tracing	12
4.2	Error estimation	17
5	Light Scattering	21
5.1	Problem formulation	22
5.2	Iterative compensation approach(ICA)	23
5.3	Deconvolution method	25
5.3.1	Discretization	25
5.3.2	Split augmented Lagrangian shrinkage algorithm (SALSA)	27
5.4	Computed examples	28
5.4.1	The model problem	29
5.4.2	Optical design problems	33
6	Conclusions and Future Work	37
6.1	Conclusions	37
6.2	Future work	38

List of Figures

1.1	Reflector geometry with a far field target	2
2.1	Reflection and refraction	4
3.1	The cylindrically symmetric case	5
3.2	Cylindrical coordinate system	5
3.3	Geometry of the 2D problem	7
3.4	Infinitesimal rotation	7
3.5	$t \rightarrow \theta$ relations in the increasing case	10
3.6	$t \rightarrow \theta$ relations in the decreasing case	10
3.7	Reflector profiles in the increasing case	11
3.8	Reflector profiles in the decreasing case	11
4.1	Ray tracing geometry	13
4.2	Ray tracing results in decreasing case	16
4.3	Ray tracing results in increasing case	16
4.4	Histograms of intensity distributions of all examples in Section 3.1, $N_b = 100, N_r = 10^6$	17
4.5	Error with respect to N_r in increasing cases, $N_b = 100$	19
4.6	Coefficients A and B	20
4.7	Error with respect to $N_b, N_r = 10^7$	20
5.1	Scattering on a rough surface.	21
5.2	Illustration of the scattering problem.	22
5.3	Probability density functions of diffusions.	23
5.4	Flowchart of the iterative compensation algorithm	24
5.5	Illustration of trapezoidal rule.	26
5.6	$\mathcal{G}_{\text{ref}}(z)$ and $\mathcal{G}_{\text{desired}}(z)$ of the model problem.	29
5.7	Sparsity pattern of matrix K of size 35×35	30
5.8	Discretization error and condition number.	30
5.9	Residual versus size.	31
5.10	Resulting intensity distribution of deconvolution method, $N_r = 10^7, N_b = 100$	31
5.11	Resulting intensity distributions of ICA. We use the desired target distribution $\mathcal{G}_{\text{desired}}(\theta)$ as the initial predefined target (iteration=0) and the corresponding obtained output is plotted in black curve and labeled as " $\mathcal{G}_{\text{scatt},0}$ ". Obtained outputs after several iterations are plotted in blue curves and labeled as " $\mathcal{G}_{\text{scatt}}$ ".	32
5.12	Evolution of the error during the iterative process of ICA. The error of the deconvolution method is plotted in dashed line for comparison. $N_r = 10^6, N_b = 100$	32
5.13	Resulting intensity distribution $\mathcal{G}_{\text{scatt}}(\theta)$ and corresponding predefined distribution $\mathcal{G}_{\text{ref}}(\theta)$ using SALSA, $N_r = 10^6, N_b = 100$	33
5.14	Resulting intensity distribution using ICA, the output using initial predefined target distribution is labeled as " $\mathcal{G}_{\text{scatt},0}$ ", $N_b = 100, N_r = 10^6$	34
5.15	Result of the deconvolution method.	35

5.16 Results of ICA. We use the desired target distribution $\mathcal{G}_{\text{desired}}(\theta)$ as the initial predefined target (iteration=0) and the corresponding obtained output is plotted in black curve and labeled as " $\mathcal{G}_{\text{scatt},0}$ ". Obtained outputs after several iterations are plotted in blue curves and labeled as " $\mathcal{G}_{\text{scatt}}$ ". 36

List of Tables

3.1	Four examples	8
3.2	Relative errors by solving ODE system	9
3.3	Relative errors by flux matching	9
5.1	Obtained support of resulting intensity distributions in different cases	33

Chapter 1

Introduction

Optical design is a basic problem in lighting technology. It is the process of using a lens or reflector to transform the light from the source to a desired output pattern under certain constraints such as minimizing the loss of light and manufacturing limitations. Optical design has long been a matter of craftsmanship. Around 700BC ancient Egyptians and Mesopotamians started polishing crystals (often quartz) to concentrate light [1]. Theories on light and vision were then developed by ancient Greek philosophers. Earlier studies on optics include geometrical, physical and physiological optics. Along with the development of wave optics and quantum optics, we entered the era of "modern optics" [2]. The development of technique always significantly lags theory. For a long time, design techniques were restricted to point sources and geometries with rotational and translational symmetry. Not until 1932 did Boldyrev derive the general equation governing the reflector shape [16]. From the 80's up to now, more and more numerical methods regarding solving the general equation are devised. Nowadays, with the fast development of the electronic computer, the design process has radically changed.

Forward and inverse methods are two important techniques in optical design. In forward methods, a trial configuration of the optical system is given, then rays are traced and photometric variables at the target are computed. Comparing the desired target with the simulation results, the configuration is altered in some way to optimize the performance of the system. This procedure is repeated many times until a satisfactory result is obtained. The forward method is valuable for many applications and is implemented in several software packages. However, the evaluation of the target distribution is a lengthy operation, therefore forward methods are extremely time consuming when high accuracy is required. Besides, as we will see in Chapter 5, forward method usually have poor convergence performance in optimization processes.

Inverse methods are another useful tool for optical design. Inverse problems can be formulated in the following framework: Given a light source and a target distribution, we aim to construct an optical system that projects a predefined illumination pattern onto the target. The optical surfaces are often modeled as *free-form* surfaces. Often a simple system consisting of one single reflector or lens is desired. Although the inverse method can significantly improve computation speed, real 3D inverse problems lead to highly non-linear second order partial differential equation, called the Monge-Ampère equation. A general overview of the Monge-Ampère equation can be found in [7]. Inverse optical design involves advanced computational methods and cannot be solved by heuristic methods in its most general form, therefore we restrict ourselves to a specific simplified problem.

Throughout this thesis, we consider the following far field problem: the desired target distribution is defined as intensity on angles, as opposed to the near field problem, where the target distribution is defined as illuminance on a plane. The simplification is derived in detail in Chapter 3. As illustrated in Figure 1.1, a point source is located at the original point. The intensity of the emitted light follows the distribution $\mathcal{I}(t)$. The ray with the direction t hits the reflector at point A and is reflected with direction θ . The target distribution is defined by $\mathcal{G}(\theta)$. The angle t is measured counterclockwise with respect to the positive direction of the symmetry axis while the angle θ is measured clockwise with respect to the negative direction of the symmetry axis.

Our goal is to compute the reflector profile that produces a predefined functional dependence of the output angle θ given the source with intensity $\mathcal{I}(t)$. In reality, the light source always has a finite size. When the light source is sufficiently small relative to the dimensions of the optical system, it can be treated as a point source. We first assume that the reflector is specular, then the shape of the reflector surface can be determined by solving differential equations governing specular reflection. We evaluate the performance of the optical system using Monte Carlo ray tracing [17] with a self developed program. Analysis of the error is given as well. In order to make the model realistic, we introduce the transmission mechanism of scattering. Scattering is the deviation from the law of reflection due to irregularities on the reflector surface. We reconsider the design process under the diffuse reflection assumption. For the sake of simplicity, we do not investigate multiple reflections. Practical issues as manufacturability are not considered.

Various mathematical techniques and methods are involved in this thesis. Ordinary differential equations and numerical integration are used in the process of generating the reflector profile. The bisection method is employed to compute the intersection of the ray and the reflector surface. Knowledge of probability is involved both in the error estimation and the formulation of the scattering model. The deconvolution process requires several numerical computational techniques.

This thesis is organized as follows. In Chapter 2, we start with some necessary terminology and theory in illumination optics. The application of these concepts runs through the thesis. In Chapter 3, we deal with the cylindrically symmetric 3D problem with point source, which can be treated as a 2D problem. Four examples are presented to give an intuitive impression of our design. In Chapter 4, Monte Carlo ray tracing is introduced and we derive an estimation of the error. Simulation results are presented to verify the calculations in Chapter 3. In Chapter 5, light scattering is included in the model. We compare two types of methods: deconvolution method and iterative compensation approach. Chapter 6 summarizes our work and give recommendations for future research.

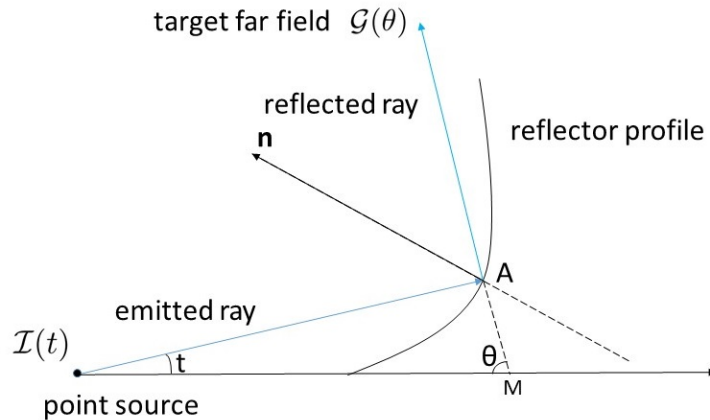


Figure 1.1: Reflector geometry with a far field target

Chapter 2

Illumination Optics

This chapter introduces some basic variables, terminology and the theory from illumination optics. The law of reflection is the main result that is needed in this thesis. The law of refraction is also included for completeness.

2.1 Lighting terminology

Light is electromagnetic radiation transporting energy. The luminous flux Φ_v is the measure of the perceived power of light by the human eye. The usual unit of radiant power is *lumen*. For a point source, the luminous intensity (or simply intensity) is the luminous flux per unit solid angle Ω , i.e.,

$$I_v = \frac{d\Phi_v}{d\Omega}. \quad (2.1)$$

The unit of intensity is *lm/sr*. When light is incident on a surface, the power per unit area is called illuminance:

$$E_v = \frac{d\Phi_v}{dA}, \quad (2.2)$$

where dA denotes the area. The unit of illuminance is *lm/m²*.

2.2 Reflection, refraction and total internal reflection

Reflection is the change in direction of a wavefront at an interface between two different media so that the wavefront returns into the medium from which it originates. As illustrated in Figure 2.1, the incoming light ray is the incident ray which strikes the surface at the incidence point. The light ray that leaves the surface is the reflected ray. The normal is an imaginary line perpendicular to the surface. Let unit vectors \mathbf{s}_i and \mathbf{s}_r denote directions of the incident ray and the reflected ray, and \mathbf{n} denotes the unit normal (point to the side of the incident ray). The law of reflection can be expressed in vector form [6]:

$$\mathbf{s}_r = \mathbf{s}_i - 2(\mathbf{s}_i \cdot \mathbf{n})\mathbf{n}. \quad (2.3)$$

The reflected ray lies in the same plane as the incident ray and the normal. Let θ_i and θ_r denote the incident angle and reflected angle respectively ($\theta_i, \theta_r \in (0, \pi/2)$). Then

$$\cos(\theta_i) = -\mathbf{s}_i \cdot \mathbf{n}, \quad \cos(\theta_r) = \mathbf{s}_r \cdot \mathbf{n}. \quad (2.4)$$

Combining (2.3) and (2.4), we have

$$\mathbf{s}_r \cdot \mathbf{n} = \mathbf{s}_i \cdot \mathbf{n} - 2(\mathbf{s}_i \cdot \mathbf{n})(\mathbf{n} \cdot \mathbf{n}) = -\mathbf{s}_i \cdot \mathbf{n}, \quad (2.5)$$

which leads to the more well-known version "The angle of incidence equals the angle of reflection", i.e.,

$$\theta_i = \theta_r. \quad (2.6)$$

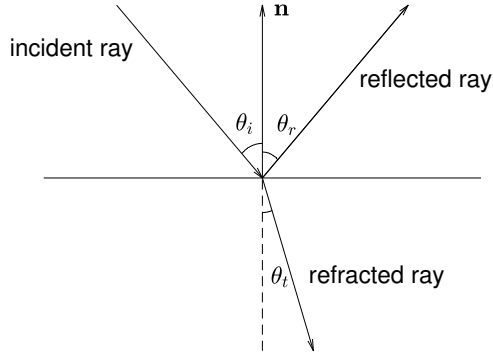


Figure 2.1: Reflection and refraction

Another important concept is refraction. Refraction is the change in direction of propagation of a wave due to a change in its transmission medium. The refractive index n of a medium is the ratio between the speed of light v in that medium and the speed of light c in vacuum. The refractive index of glass is typically 1.5. Let \mathbf{s}_t denote the refracted ray and n_i, n_t denote the refractive indices of material 1 and material 2, then the law of refraction is [8]

$$n_i(\mathbf{s}_i \times \mathbf{n}) = n_t(\mathbf{s}_t \times \mathbf{n}). \quad (2.7)$$

Taking the norm on both sides of (2.7) and note that $\sin \theta_i = |\mathbf{s}_i \times \mathbf{n}|$ and $\sin \theta_t = |\mathbf{s}_t \times \mathbf{n}|$, we can derive the well-known Snell's law:

$$n_i \sin \theta_i = n_t \sin \theta_t. \quad (2.8)$$

The incident ray, refracted ray and the normal are located in one plane. When $\sin \theta_i = \frac{n_t}{n_i}$, the refracted ray is emitted parallel to the surface. If $\sin \theta_i > \frac{n_t}{n_i}$, no refraction occurs. This phenomenon is called total internal reflection (TIR).

Chapter 3

The Cylindrically Symmetric Case

In this chapter, we consider the 3D far field reflector design problem with cylindrical symmetry which allows 2D treatment: We only need to generate the reflector profile in a plane. The full reflector surface can then be obtained by sweeping the curve around its axis of symmetry. This kind of problem is a classical problem in early optical design, which is described in many publications (e.g., [4]). The cylindrical coordinate system is introduced in Section 3.1 to derive a simple representation of energy conservation. Section 3.2 gives the formula to describe the shape of the reflector. Examples are presented in Section 3.3 to verify our calculations of the reflector profiles.

3.1 Cylindrical coordinate system and energy conservation

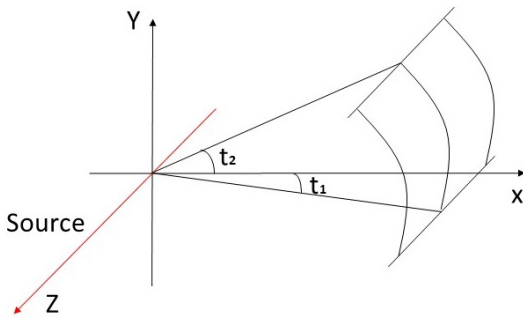


Figure 3.1: The cylindrically symmetric case

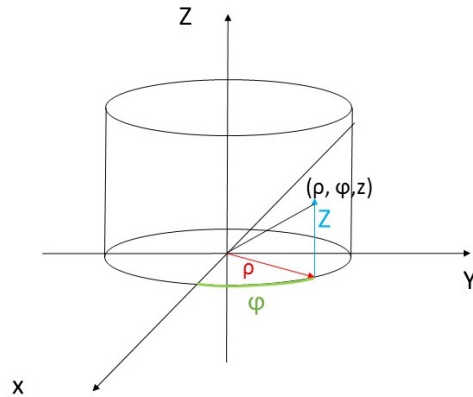


Figure 3.2: Cylindrical coordinate system

As illustrated in Figure 3.1, suppose we have a rectangular xyz -coordinate system in \mathbb{R}^3 , with an infinite linear source l coinciding with the z -axis that is of uniform radiation. To introduce the optical design procedure we first establish the cylindrical coordinate system. The cylindrical coordinate system is a generalization of the two-dimensional polar coordinate system to three dimensions by superposing a height (z) axis. As illustrated in Figure 3.2, a reference plane is chosen and a ray that lies in the plane is selected to be the polar axis. The cylindrical axis is the ray intersecting the polar axis at the original point and is perpendicular to the reference plane. Then the position of an arbitrary point (ρ, φ, z) is specified by the distance from the cylindrical axis, which is called radius and denoted ρ , the angular coordinate, which is referred to as the azimuth and denoted φ , and the altitude z , which is the distance from the reference plane. The radius and the azimuth are called the polar coordinates, as they correspond to a two-dimensional

polar coordinate system. The altitude can be positive or negative, depending on which side the point faces the reference plane. It is usual to choose the Cartesian xy -plane and z -axis as the reference plane and the polar axis respectively. Consequently the cylindrical axis is the Cartesian z -axis. The Cartesian coordinates can be uniquely represented by cylindrical coordinates:

$$\begin{cases} x = \rho \cos \varphi, \\ y = \rho \sin \varphi, \end{cases}$$

the z -coordinate is the same in both systems. If we fix $\rho = 1$, then the emitted rays can be represented by vectors

$$\mathbf{v} := \mathbf{v}(t, z) := (\cos t, \sin t, z) \in \mathcal{S}^1 \times \mathbb{R}, \quad t \in [t_1, t_2], z \in \mathbb{R}, \quad (3.1)$$

where t is the angular coordinate and $\mathcal{S}^1 \times \mathbb{R}$ denotes the surface of the open cylinder with unit radius. The reflector surface $\mathbf{r}(t, z)$ is given by

$$\mathbf{r} := \mathbf{r}(t, z) = (\rho(t) \cos t, \rho(t) \sin t, z), \quad (3.2)$$

where ρ is twice differentiable and strictly positive. The reflected rays are denoted by \mathbf{w} , given in cylindrical coordinates θ, z' (an illustration of angle t and θ is given in Figure 1.1) by

$$\mathbf{w} := \mathbf{w}(\theta, z') := (-\cos \theta, \sin \theta, z') \in \mathcal{S}^1 \times \mathbb{R}, \quad \theta \in [\theta_1, \theta_2], z' \in \mathbb{R}. \quad (3.3)$$

We have a function $\mathcal{I}(t, z)$ and a function $\mathcal{G}(\theta, z')$ which describe the intensity distribution of the infinitesimal line element dl at position $(0, 0, s)$ for arbitrary s and the desired target, respectively. We assume a one-to-one correspondence between incident and reflected rays. Then by energy conservation we have

$$\mathcal{I}(t, z) d\mathcal{S}(t, z) = \mathcal{G}(\theta, z') d\mathcal{S}(\theta, z'), \quad (3.4)$$

where $d\mathcal{S}(t, z)$ and $d\mathcal{S}(\theta, z')$ can be represented in terms of the underlying cylindrical coordinates:

$$d\mathcal{S}(t, z) = \left| \frac{\partial \mathbf{v}}{\partial t} \times \frac{\partial \mathbf{v}}{\partial z} \right| dt dz = dt dz, \quad (3.5)$$

$$d\mathcal{S}(\theta, z') = \left| \frac{\partial \mathbf{w}}{\partial \theta} \times \frac{\partial \mathbf{w}}{\partial z'} \right| d\theta dz' = d\theta dz'. \quad (3.6)$$

Therefore we have

$$\mathcal{I}(t, z) dt dz = \mathcal{G}(\theta, z') d\theta dz'. \quad (3.7)$$

Integrating both sides of (3.7) leads to the global energy conservation

$$\int_{t_1}^{t_2} \int_{-\infty}^{\infty} \mathcal{I}(t, z) dz dt = \int_{\theta_1}^{\theta_2} \int_{-\infty}^{\infty} \mathcal{G}(\theta, z') dz' d\theta. \quad (3.8)$$

The uniformity of the source implies that \mathcal{I} and \mathcal{G} are independent of z and z' , respectively, therefore (3.8) can be further simplified to

$$\int_{t_1}^{t_2} \mathcal{I}(t) dt = \int_{\theta_1}^{\theta_2} \mathcal{G}(\theta) d\theta. \quad (3.9)$$

It is worth remarking that the form of energy conservation differs in problems with different geometry. Now the 3D problem is reduced to a 2D problem. We can use the geometry as well as the energy conservation to solve the problem.

3.2 Model for the reflector shape

In this section we derive the formula governing the shape of the reflector from the 2D geometry, which depends on the angles t and θ . In order to design the optical surface, the correspondence (or called mapping) $t \rightarrow \theta$ must be computed. This mapping can be found by energy conservation and the law of reflection.

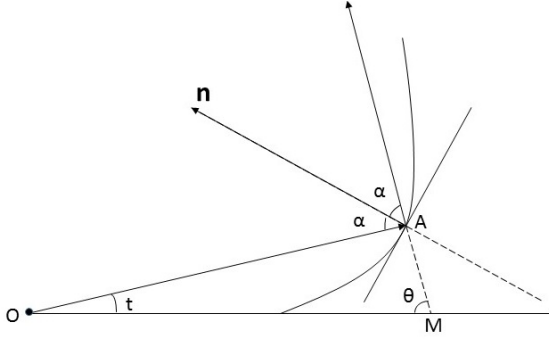


Figure 3.3: Geometry of the 2D problem

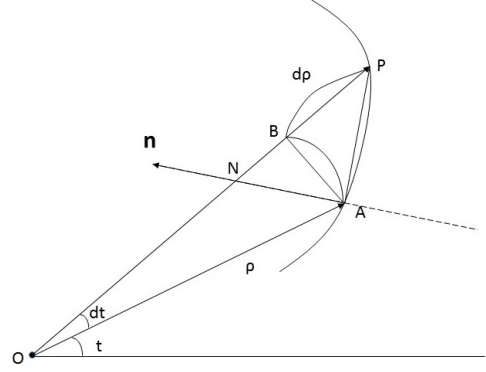


Figure 3.4: Infinitesimal rotation

As illustrated in Figure 3.3, the emitted ray OA hits the reflector at point A with incident angle α and we observe that $\pi - (t + \theta) = \angle OAM = \pi - 2\alpha$ thus $\alpha = (t + \theta)/2$. The distance between points O and A is ρ . In Figure 3.4 we rotate OA over an infinitesimal angle dt to obtain another emitted ray OP which intersects the reflector at P . N is the intersection of OP and the normal. By drawing a circular arc \overline{AB} with $|OA| = |OB|$, we obtain the difference $d\rho$. According to the geometry, the following relation holds in the limit $dt \rightarrow 0$

$$OA \perp AB \text{ and } AP \perp \mathbf{n}. \quad (3.10)$$

Thus, we conclude that $\angle PAB = \pi/2 - \angle NAB = \alpha$ and then the segment $|AB| = d\rho/\tan \alpha$. Since the arc $|\overline{AB}| = \rho dt$, the relation $|\overline{AB}| = |AB|$ leads to

$$\frac{1}{\rho} \frac{d\rho}{dt} = \tan \alpha = \tan\left(\frac{t + \theta}{2}\right). \quad (3.11)$$

Since $\tan \alpha$ has a singularity at $\alpha = \frac{\pi}{2}$, we restrict ourselves to $t + \theta \leq \pi$. Integrating from t_1 to t on both sides of (3.11) yields

$$\rho(t) = \rho_1 \exp\left\{\int_{t_1}^t \tan\left(\frac{s + \theta(s)}{2}\right) ds\right\}, \quad (3.12)$$

where $\rho_1 = \rho(t_1)$ is a scaling parameter which can adjust the size of the optical system. For the point source, ρ_1 has no effect on the design and we fix it at 1. But if we use finite source instead, it will be crucial, because the scale determines how likely the finite source can be approximated by the point source.

The $t \rightarrow \theta$ mapping that we need to know to compute $\rho(t)$ can be derived from (3.9). Here we only consider monotonic solutions, which can be easily computed and guarantee the smoothness of the reflectors. First we consider the increasing solution which denoted by θ^+ . For an arbitrary $t \in [t_1, t_2]$, we have

$$\int_{\theta_1}^{\theta^+(t)} \mathcal{G}(\theta') d\theta' = \int_{t_1}^t \mathcal{I}(t') dt'. \quad (3.13)$$

For the increasing case, differentiating with respect to t on both sides of (3.13) yields

$$\mathcal{G}(\theta^+)(\theta^+)'(t) = \mathcal{I}(t) \implies (\theta^+)'(t) = \frac{\mathcal{I}(t)}{\mathcal{G}(\theta^+)}, \quad (3.14)$$

with initial condition $\theta^+(t_1) = \theta_1$. Similarly, for the decreasing case we have

$$(\theta^-)'(t) = -\frac{\mathcal{I}(t)}{\mathcal{G}(\theta^-)}, \quad (3.15)$$

with initial condition $\theta^-(t_1) = \theta_2$. For given $\mathcal{I}(t)$ and $\mathcal{G}(\theta)$ the reflector profile can be computed by solving the following initial value problems.

$$\text{Increasing case } \begin{cases} \theta'(t) = \frac{\mathcal{I}(t)}{\mathcal{G}(\theta)} & \theta(t_1) = \theta_1, \\ \rho'(t) = \rho \tan\left(\frac{t+\theta(t)}{2}\right) & \rho(t_1) = \rho_1. \end{cases} \quad (3.16)$$

$$\text{Decreasing case } \begin{cases} \theta'(t) = -\frac{\mathcal{I}(t)}{\mathcal{G}(\theta)} & \theta(t_1) = \theta_2, \\ \rho'(t) = \rho \tan\left(\frac{t+\theta(t)}{2}\right) & \rho(t_1) = \rho_1. \end{cases} \quad (3.17)$$

However, this method requires nice analytical properties of $\mathcal{I}(t)$ and $\mathcal{G}(\theta)$, which is, as we will see in Chapter 5, not always guaranteed.

Alternatively, we can find the $t \rightarrow \theta$ mapping by matching two cumulative distributions: Compute the cumulative flux as a function of t and also a function of θ and then match the points of these two functions. For implementation, interpolation is utilized for discrete data. The reflector profile can be computed by numerically integrating (3.12) afterwards. For practical problems, $\mathcal{I}(t)$ and $\mathcal{G}(\theta)$ are typically given in discrete forms, thus it is more convenient to implement cumulative distributions matching. More importantly, this method is more robust than the ODE solver: The predefined intensity distribution that we obtained in the deconvolution or iterative compensation process typically presents oscillation and the ODE solver often fails to find the $t \rightarrow \theta$ mapping in that case. We will compare these two methods by examples in the next section.

3.3 Examples

We choose four relatively easy examples with analytical solutions to numerically verify our calculations, see Table 3.1. These examples cover different types of target distribution: The first one requires a uniform target and the resulting $t \rightarrow \theta$ relation is linear. The $t \rightarrow \theta$ relation in the second example is polynomial, which lead to an algebraic fraction target distribution. The target in the third example is again an algebraic fraction but with a tangent $t \rightarrow \theta$ relation. The required target in the last example is a step function. These examples also reveal difficulties we may encounter in the subsequent design. The linear, tangent and piecewise linear examples are given in [4].

Table 3.1: Four examples

	linear	polynomial	tangent	piecewise linear
$[t_1, t_2]$	$[-\pi/4, \pi/4]$	$[\pi/5, \pi/4]$	$[\pi/4, \pi/2]$	$[-\pi/4, \pi/4]$
$[\theta_1, \theta_2]$	$[-\pi/4, \pi/4]$	$[\pi^3/125, \pi^3/64]$	$[-\pi/8, \pi/8]$	$[-\pi/2, \pi/2]$
$\mathcal{I}(t)$	1	1	1	1
$\mathcal{G}(\theta)$	1	$\frac{1}{\theta^{2/3}}$	$\frac{4\pi}{\pi^2 + 64\theta^2}$	$\mathcal{G}_{\text{PL}}(\theta)$
$\theta^+(t)$	t	t^3	$\frac{\pi}{8} \tan(2t - \frac{3\pi}{4})$	$\theta_{\text{PL}}^+(t)$
$\theta^-(t)$	$-t$	$\theta_{\text{poly}}^-(t)$	$-\frac{\pi}{8} \tan(2t - \frac{3\pi}{4})$	$\theta_{\text{PL}}^-(t)$

$$\theta_{\text{poly}}^-(t) = \left(\frac{9\pi - 20t}{20} \right)^3.$$

$$\mathcal{G}_{\text{PL}}(\theta) = \begin{cases} 6/7 & \text{for } \theta \in [-\pi/2, -\pi/4), \\ 2/7 & \text{for } \theta \in [-\pi/4, \pi/4), \\ 4/7 & \text{for } \theta \in [\pi/4, \pi/2]. \end{cases}$$

$$\theta^+_{\text{PL}}(t) = \begin{cases} 7t/6 - 5\pi/24 & \text{for } t \in [-\pi/4, -\pi/28), \\ 7t/2 - \pi/8 & \text{for } t \in [-\pi/28, 3\pi/28), \\ 7t/4 + \pi/16 & \text{for } t \in [3\pi/28, \pi/4]. \end{cases}$$

$$\theta^-_{\text{PL}}(t) = \begin{cases} -7t/4 + \pi/16 & \text{for } t \in [-\pi/4, -3\pi/28), \\ -7t/2 - \pi/8 & \text{for } t \in [-3\pi/28, \pi/28), \\ -7t/6 - 5\pi/24 & \text{for } t \in [\pi/28, \pi/4]. \end{cases}$$

In order to evaluate the accuracies of these two methods, we compare the relative error, which is defined as following

$$\text{relative error} = \frac{\|\theta_{\text{num}} - \theta_{\text{exac}}\|_2}{\|\theta_{\text{exac}}\|_2}, \quad (3.18)$$

where θ_{num} is the numerical results on the sample points and θ_{exac} is the correspond exact solution. Relative errors in the four examples are shown in Table 3.2 and Table 3.3. In the first method, we use the matlab ode45-solver to compute the solution of the ODE system. Although the error-control parameter 'RelTol' [24] has little impact on the final results, it is set to be 10^{-5} for clarity. The matching of the cumulative distributions is a way of applying energy conservation, and hence requires numerical integration, which is accomplished by the trapezoidal method. The number of sample points is fixed at 1000 in both methods. Since both methods involve numerical integration, their accuracies depend on the high-order derivatives of the target distribution $\mathcal{G}(\theta)$. As we can see, for the linear example with zero high-order derivative, solutions are precise (up to numerical precision). But for nonlinear solutions in polynomial and tangent cases, the high-order derivatives are nonzero, which yield much larger errors. In the piecewise linear case, where high order doesn't exist, the largest error is observed.

Table 3.2: Relative errors by solving ODE system

$\frac{\ \theta_{\text{num}} - \theta_{\text{exac}}\ _2}{\ \theta_{\text{exac}}\ _2}$	linear	polynomial	tangent	piecewise linear
increasing	2.14×10^{-16}	1.18×10^{-9}	1.50×10^{-6}	5.8×10^{-3}
decreasing	2.14×10^{-16}	9.49×10^{-10}	9.76×10^{-6}	2.00×10^{-3}

Table 3.3: Relative errors by flux matching

$\frac{\ \theta_{\text{num}} - \theta_{\text{exac}}\ _2}{\ \theta_{\text{exac}}\ _2}$	linear	polynomial	tangent	piecewise linear
increasing	1.16×10^{-17}	3.47×10^{-9}	1.98×10^{-7}	2.19×10^{-4}
decreasing	1.16×10^{-17}	3.47×10^{-9}	1.98×10^{-7}	2.19×10^{-4}

Figure 3.5 and Figure 3.6 show the numerical and exact solutions for all examples, both increasing and decreasing cases, respectively. Corresponding reflector profiles are shown in Figure 3.7 and Figure 3.8. Note figures are plotted in different scales. Usually the reflector scales vary in different configurations. As we will see in Chapter 4, the increasing case and decreasing case correspond to diverging and converging beams, respectively. The reflector shapes in the polynomial and tangent cases don't have analytical expressions.

Conclusion

The 3D cylindrically symmetric reflector design problem with point source and specular reflection assumption can be viewed as a completely solved issue: First find the $t \rightarrow \theta$ relation and then compute the reflector profile by integration. We propose two alternative methods to calculate the $t \rightarrow \theta$ relation: One is solving the ODE system and the other is comparing the cumulative flux.

Both methods perform well. The realizations of a desired target are not unique and the solution may be properly chosen according to some practical guidelines.

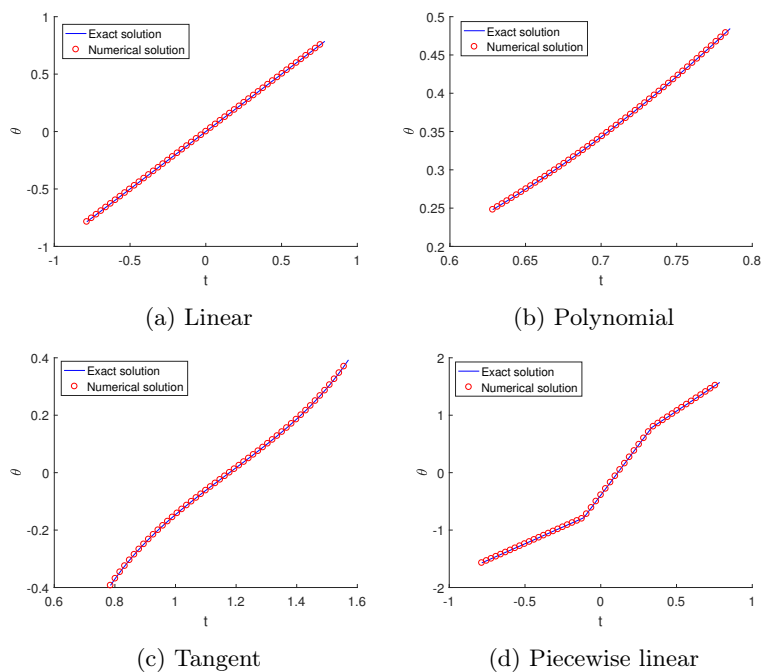


Figure 3.5: $t \rightarrow \theta$ relations in the increasing case

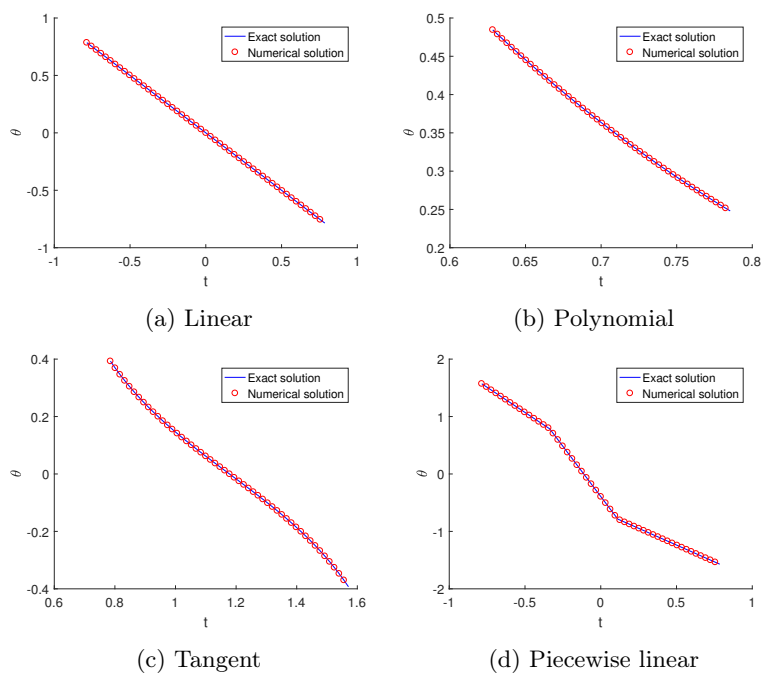


Figure 3.6: $t \rightarrow \theta$ relations in the decreasing case

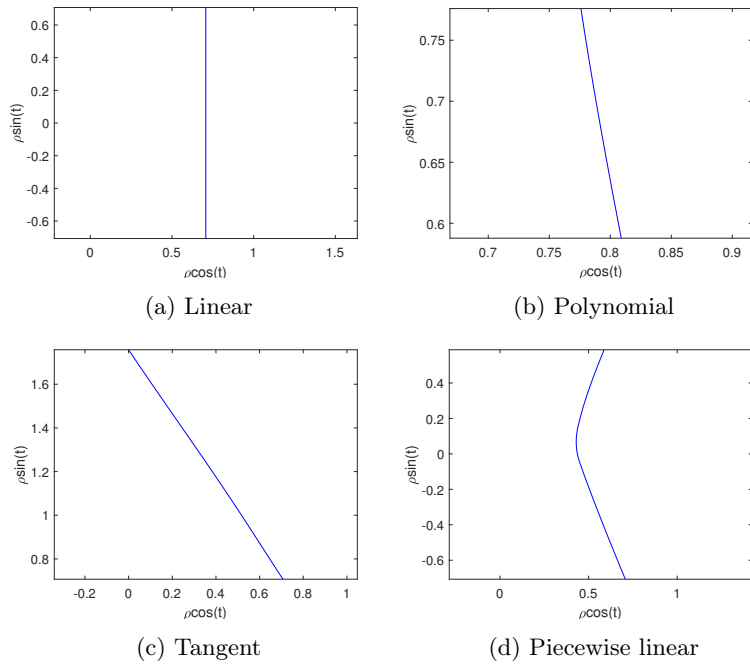


Figure 3.7: Reflector profiles in the increasing case

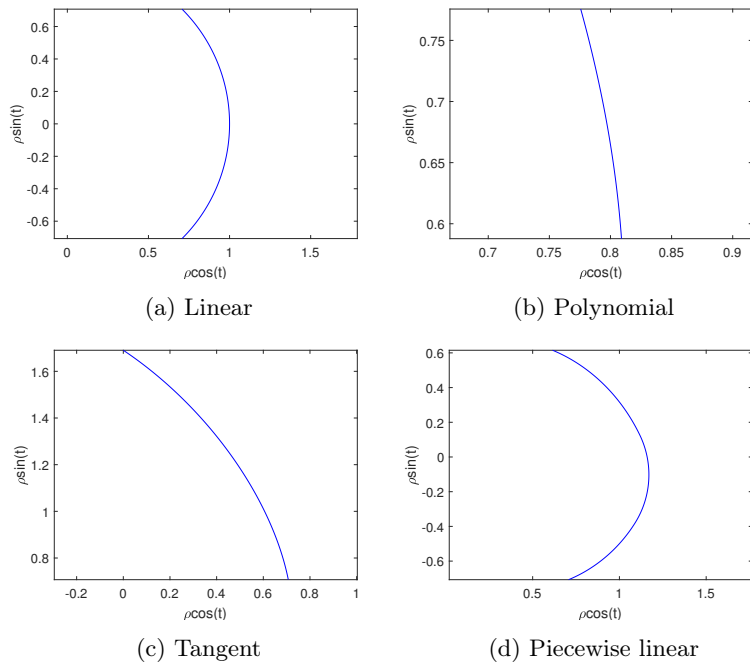


Figure 3.8: Reflector profiles in the decreasing case

Chapter 4

Ray Tracing

Every optical system needs to be verified that it fulfills the claimed specifications. This step used to be done by prototyping and measuring. Nowadays, with the advent of computers, it is possible to do experimental verification by simulation, which greatly saves time and money. Ray tracing is a widely used simulation technique for optical design: A large collection of rays that originate from the source are traced to the target through the optical system. Photometric variables at the target are computed from these rays. Ray tracing techniques are implemented in a wide range of free software and commercial software [22]. Out of the many methods to implement the ray tracing process, the most widely used method is Monte Carlo ray tracing. Monte Carlo methods rely on repeated random sampling to obtain numerical results. They are proper techniques for the evaluation of photometric quantities, since the emission of the ray from the source and the corresponding reflection occurring at the reflector surface can both be considered as stochastic processes. Monte Carlo ray tracing consists of three steps:

1. Generation of a sample of rays according to the intensity distribution $\mathcal{I}(t)$.
2. Propagation of the rays through the optical system.
3. Collection of the rays and computation of $\mathcal{G}(\theta)$.

Since we are interested in the far field problem, all the reflected angles are collected. In order to estimate the target intensity distribution, the entire range of values is divided into adjacent and disjoint small intervals, which are usually called bins, and then the number of rays that fall into each bin is counted. Although not necessary, the bins are often of equal size such that the number of rays in each bin is proportional to the relative intensity. In this chapter, we design a ray trace program to evaluate the performance of the optical system. Section 4.1 elaborates the procedure of tracing one single ray. Issues of the computation of the intersection and the calculation of reflected angles are discussed. To measure the deviation from the desired target, we define the error and give an estimation in Section 4.2. A comparison between the real error and the estimation is presented at the end.

4.1 The procedure of ray tracing

First we declare the notation in this section. In the previous chapters, we used $\{t, \theta, \rho\}$ as the parametrization of the optical system. In this section, we utilize τ to indicate the direction of the emitted ray and the corresponding reflected direction is denoted by $\tilde{\theta}$. This section involves transformation between Cartesian coordinates and polar coordinates. For clearance, we add subscripts c and p to indicate Cartesian and polar coordinates, respectively. To determine the photometric variables we need to compute the path of a large number of rays and collect the reflected angles in bins. This gives a ray distribution from which we can compute the intensity.

Tracing one single ray follows the following steps:

- Randomly choose a ray with direction $\tau \in [t_1, t_2]$.
- Compute the intersection of the ray and the reflector, and compute the normal at the intersection.
- Calculate the reflected ray using the reflection law (2.3).
- Determine the direction $\tilde{\theta}$.

We will elaborate this procedure. The reflector is represented by polar coordinates $(t, \rho(t))_p$. Usually we don't have an explicit expression to describe the profile, instead a set of discrete sample points $\{(t_1, \rho_1)_p, (t_2, \rho_2)_p, \dots, (t_n, \rho_n)_p\}$ are used for the calculation.

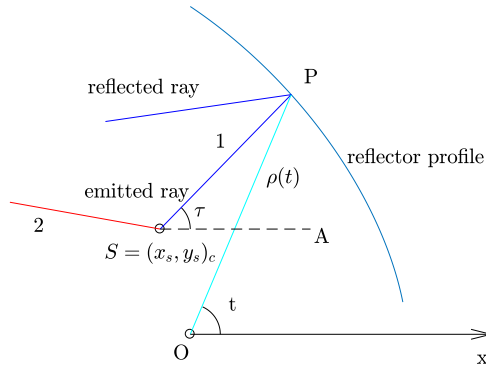


Figure 4.1: Ray tracing geometry

We only consider the point source. As shown in Figure 4.1, the location of the source is $S = (x_s, y_s)_c$. An emitted ray either hits the reflector, like ray 1, or escapes from the system, like ray 2. For an arbitrary point $P = (t, \rho(t))_p$ on the reflector, $\angle PSA$ (measured counterclockwise with respect to x -axis) can be expressed in terms of the angular coordinate t by the inner product of \vec{SP} and \mathbf{e}_x :

$$\begin{aligned} \angle PSA(t) &= \text{sign}(t) \cdot \arccos\left(\frac{\vec{SP} \cdot \mathbf{e}_x}{|\vec{SP}|}\right) \\ &= \text{sign}(t) \cdot \arccos\left(\frac{\rho(t) \cos t - x_s}{\sqrt{[\rho(t) \cos t - x_s]^2 + [\rho(t) \sin t - y_s]^2}}\right), \quad \mathbf{e}_x = (1, 0). \end{aligned}$$

Define the function f as follows

$$f(t) = \angle PSA - \tau, \quad \tau \in [t_1, t_2], \quad (4.1)$$

where τ is the direction angle of the emitted ray. $f(t)$ is the difference between $\angle PSA$ and τ . In order to compute the intersection, we need to solve the equation $f(t) = 0$. We only consider the simplest case that $f(t)$ is continuous and has a unique root in $[t_1, t_2]$. We can use the bisection method to find the root of (4.1). This procedure is summarized in Algorithm Bisection. The inputs for the algorithm are endpoints values a, b , function f , tolerance tol and maximum iterations n_{\max} . As a constraint, $f(a) \cdot f(b) < 0$ must hold, otherwise the root might not exist. At each step the algorithm computes the midpoint value $c = (a + b)/2$ and the corresponding function value $f(c)$. Unless c itself is a root, which is possible and will unluckily take more time to find the root, there are only two possibilities: either $f(a) \cdot f(c) < 0$ and $[a, c]$ brackets the root or $f(b) \cdot f(c) < 0$ and

$[c, b]$ brackets the root. In this way, the size of the interval that contains the root is reduced to half at each step. The maximum number of iterations n_{\max} is set to be $\lceil \log_2 \frac{b-a}{tol} \rceil$, where tol is the tolerance. If the number of iteration exceeds n_{\max} and no root has been found, then the algorithm returns Inf. This corresponds to the situation that the ray escapes from the system.

The next issue is to compute the slope of the tangent line to the reflector at an arbitrary point $(t, \rho(t))_p$. We first express the Cartesian coordinates x, y in terms of polar coordinates:

$$x = \rho(t) \cos t, \quad (4.2a)$$

$$y = \rho(t) \sin t. \quad (4.2b)$$

Differentiating both equations with respect to t yields

$$\frac{dx}{dt} = \rho'(t) \cos t - \rho(t) \sin t, \quad (4.3a)$$

$$\frac{dy}{dt} = \rho'(t) \sin t + \rho(t) \cos t. \quad (4.3b)$$

The inward-pointing normal at point $(t, \rho(t))_p$ is given by

$$\mathbf{n} = - \begin{bmatrix} \rho'(t) \sin t + \rho(t) \cos t \\ -\rho'(t) \cos t + \rho(t) \sin t \end{bmatrix}. \quad (4.4)$$

Using(3.11) to eliminate $\rho'(t)$, the inward-pointing normal can be expressed in t and $\rho(t)$, i.e.,

$$\mathbf{n} = \begin{bmatrix} -\rho(t) \tan\left(\frac{t+\theta(t)}{2}\right) \sin t - \rho(t) \cos t \\ \rho(t) \tan\left(\frac{t+\theta(t)}{2}\right) \cos t - \rho(t) \sin t \end{bmatrix}. \quad (4.5)$$

Combining (3.11), (4.4) and (4.5) yields algorithm CN. We can use spline interpolation to generate consecutive θ, ρ from the discrete data set $\{(t_1, \theta_1, \rho_1), (t_2, \theta_2, \rho_2), \dots, (t_n, \theta_n, \rho_n)\}$. Let $\tilde{\mathbf{n}}$ denote the normalized normal, then the reflected ray \mathbf{s}_2 can be calculated by the law of reflection

$$\mathbf{s}_2 = \mathbf{s}_1 - 2(\mathbf{s}_1, \tilde{\mathbf{n}})\tilde{\mathbf{n}}. \quad (4.6)$$

Finally, the angle $\tilde{\theta}$ is determined by

$$\cos \tilde{\theta} = \mathbf{s}_2 \cdot (-\mathbf{e}_x), \quad \mathbf{e}_x = (1, 0). \quad (4.7)$$

To compute the intensity distribution, we uniformly partition the target interval in small bins with size $h = \frac{\theta_2 - \theta_1}{N_b}$, i.e.,

$$[\theta_1, \theta_2] = [\xi_1, \xi_2] \cup [\xi_2, \xi_3] \cup \dots \cup [\xi_{N_b-1}, \xi_{N_b}] \cup [\xi_{N_b}, \xi_{N_b+1}], \quad \xi_i = \theta_1 + (i-1)h.$$

We compute the angle $\tilde{\theta}$ of each ray and collect these in bins. The intensity at an angle η is estimated by

$$\mathcal{G}_{\text{num}}(\eta) = \mathcal{G}_i := \frac{N_{r_i}}{N_r \cdot h}, \quad \eta \in [\xi_i, \xi_{i+1}], \quad i = 1, 2, \dots, N_b, \quad (4.8)$$

where N_{r_i} is the number of rays that arrive in the bin $[\xi_i, \xi_{i+1}]$ and N_r is the total number of rays. The ray tracing procedure is outlined in Algorithm RT with inputs source location S , reflector parametrization $\{t, \theta, \rho\}$ and parameters N_r, N_b . Algorithm Bisection and Algorithm CN are invoked in step 3 and step 5, respectively. By collecting the output angles in bins we can plot bars with heights \mathcal{G}_i to construct a histogram for estimating the target distribution produced by the optical system. The histogram is normalized such that the total area is 1 corresponding to the continuous case that the integration of the normalized target distribution is 1.

Algorithm Bisection Compute intersection

Input: $a, b, f, \tau, tol, n_{\max}$ **Output:** root

```
1: Define  $c := (a + b)/2, n := 0$ 
2: while  $|f(c)| > tol$  and  $n < n_{\max}$  do
3:   if  $f(b) \cdot f(c) \leq 0$  then
4:      $a := c$ 
5:   else
6:      $b := c$ 
7:   end if
8: end while
9: if  $n < n_{\max}$  then
10:  root :=  $c$ 
11: else
12:  root := Inf
13: end if
```

Algorithm CN Compute Normal

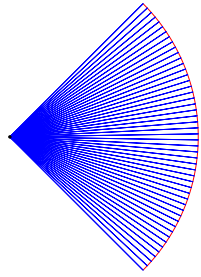
Input: t_0, θ, ρ **Output:** \mathbf{n}

```
1:  $dx = \rho(t_0) \tan(\frac{t_0 + \theta(t_0)}{2}) \cos(t_0) - \rho(t_0) \sin t_0$ 
2:  $dy = \rho(t_0) \tan(\frac{t_0 + \theta(t_0)}{2}) \sin(t_0) + \rho(t_0) \cos t_0$ 
3:  $\mathbf{n} = (-dy, dx)$  ▷ Use spline interpolation to generate  $\theta, \rho$ 
```

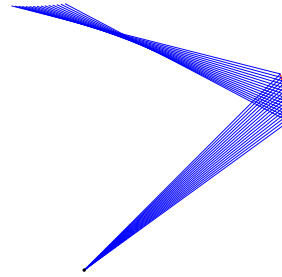
Algorithm RT Ray Tracing

Input: $S, t, \theta, \rho, N_r, N_b$ **Output:** θ

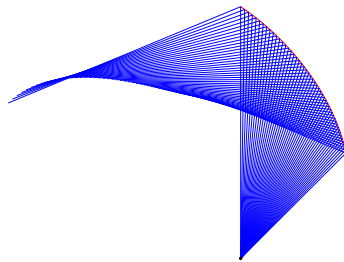
```
1: for  $i = 1$  to  $N_r$  step 1 do
2:   Randomly choose a direction angle  $\tau_0$ 
3:   Compute the intersection
4:   if intersection exists then
5:     Compute emitted ray  $\mathbf{s}_1$  and normal  $\tilde{\mathbf{n}}$ 
6:     Compute reflected ray  $\mathbf{s}_2 = \mathbf{s}_1 - 2(\mathbf{s}_1, \tilde{\mathbf{n}})\tilde{\mathbf{n}}$ 
7:     Compute  $\tilde{\theta}$ 
8:   end if
9: end for
```



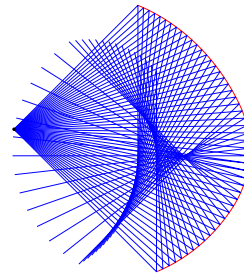
(a) Linear



(b) Polynomial

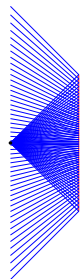


(c) Tangent

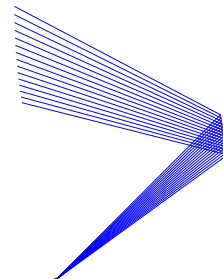


(d) Piecewise linear

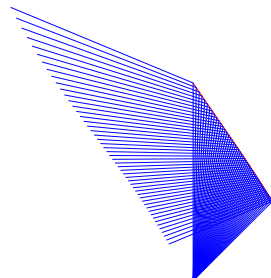
Figure 4.2: Ray tracing results in decreasing case



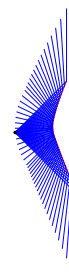
(a) Linear



(b) Polynomial



(c) Tangent



(d) Piecewise linear

Figure 4.3: Ray tracing results in increasing case

Figure 4.2 and Figure 4.3 show the reflector profiles and the traced rays in all examples in Section 3.3. The source is indicated with a black solid circle and the reflector profile is plotted in red. The increasing and decreasing solutions correspond to diverging and crossing ray bundles. These two types of solutions generate reflectors of different shape and size. The crossing solutions typically yield more compact reflectors, which is a better choice from a practical point of view. Note Figure 4.2d presents multiple reflections and thus doesn't fulfill the single reflection requirement. Faceted reflectors can be utilized for optical systems allowing multiple reflections. Due to the aforementioned reason, we will not consider this case.

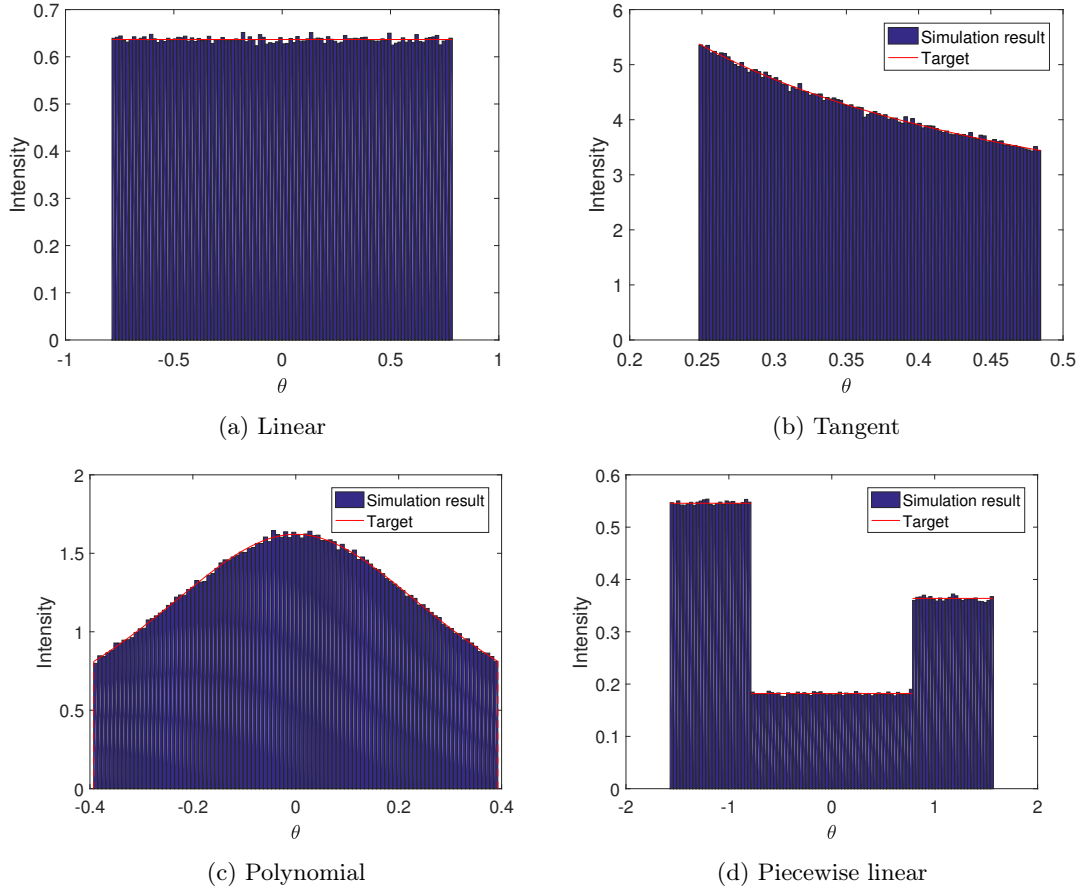


Figure 4.4: Histograms of intensity distributions of all examples in Section 3.1, $N_b = 100$, $N_r = 10^6$.

Figure 4.4 shows the simulation results of all examples. The experiments are conducted in the increasing case. Results of the decreasing case are similar. Statistical noise can be observed because ray tracing involves statistical processes and therefore can not precisely assess the performance of the optical system. The accuracy of Monte Carlo simulation depends on the number of rays and bins. As we will see in the next section, the error is inversely proportional to the square root of the number of rays which significantly increases the computational time when higher accuracy is required.

4.2 Error estimation

The error, or more formally called merit function in the literature (e.g., [18]) quantifies the departure of the obtained distribution from the desired target distribution. It is also a key aspect of the optimization process (e.g., the iterative compensation process in Chapter 5). Although

the definition of error is not unique, intensity values at each sample point are usually used for error measurement. A reasonable measure of the error is the average deviation between the real midpoint value of \mathcal{G} in bin i and the estimation \mathcal{G}_i which can be determined by the derivative of $\mathcal{G}(\theta)$ and the standard deviation of the random variables \mathcal{G}_i . We first derive the expression of the standard deviation. Since $\mathcal{G}(\theta)$ describes the energy distribution (in some sense can be represented by the number of rays with specific direction angle), it gives information on the probability distribution of the direction angle of randomly chosen rays. Let $\tilde{\mathcal{G}}(\theta)$ denote the normalized target intensity distribution, given by $\tilde{\theta}$

$$\tilde{\mathcal{G}}(\theta) = \frac{\mathcal{G}(\theta)}{\int_{\theta_1}^{\theta_2} \mathcal{G}(\theta) d\theta}. \quad (4.9)$$

For a ray with emitted direction τ , $\tilde{\theta}(t)$ falls into $[\xi_i, \xi_{i+1}]$ with probability $P_i = \int_{\xi_i}^{\xi_{i+1}} \tilde{\mathcal{G}}(\theta) d\theta$. Since the sampling of rays is independent and the outgoing angle $\tilde{\theta}$ follows the distribution $\tilde{\mathcal{G}}(\tilde{\theta})$, the number of rays falling into $[\xi_i, \xi_{i+1}]$ hence follows the binomial distribution:

$$\mathbb{P}[N_{ri} = k] = \binom{N_r}{k} P_i^k (1 - P_i)^{N_r - k}, \quad (4.10)$$

with mean $N_r P_i$ and variance $N_r P_i (1 - P_i)$. The mean μ_i and variance σ_i^2 of $\mathcal{G}_i = \frac{N_{ri}}{N_r \cdot h}$ are

$$\mu_i = \frac{P_i}{h}, \quad \sigma_i^2 = \frac{P_i(1 - P_i)}{N_r \cdot h^2}. \quad (4.11)$$

We give the definition of the ray tracing error as follows:

$$\text{error} = \frac{1}{N_b} \sum_{i=1}^{N_b} \left| \tilde{\mathcal{G}}(\xi_{i+\frac{1}{2}}) - \mathcal{G}_i \right|, \quad (4.12)$$

where $\xi_{i+\frac{1}{2}} = \frac{\xi_i + \xi_{i+1}}{2}$ is the midpoint of interval $[\xi_i, \xi_{i+1}]$. We derive an estimation of the error as follows.

$$\begin{aligned} \text{error} &= \frac{1}{N_b} \sum_{i=1}^{N_b} \left| \tilde{\mathcal{G}}(\xi_{i+\frac{1}{2}}) - \mu_i + \mu_i - \mathcal{G}_i \right| \\ &\leq \frac{1}{N_b} \sum_{i=1}^{N_b} \left[\left| \tilde{\mathcal{G}}(\xi_{i+\frac{1}{2}}) - \frac{1}{h} \int_{\xi_i}^{\xi_{i+1}} \tilde{\mathcal{G}}(\theta) d\theta \right| + \left| \mu_i - \mathcal{G}_i \right| \right] \\ &= \frac{1}{N_b} \sum_{i=1}^{N_b} \left[\left| \frac{1}{h} \int_{\xi_i}^{\xi_{i+1}} \tilde{\mathcal{G}}(\xi_{i+\frac{1}{2}}) - \tilde{\mathcal{G}}(\theta) d\theta \right| + \left| \mu_i - \mathcal{G}_i \right| \right] \\ &= \frac{1}{N_b} \sum_{i=1}^{N_b} \left[\left| \frac{1}{h} \int_{\xi_i}^{\xi_{i+1}} \tilde{\mathcal{G}}'(\xi_{i+\frac{1}{2}}) \cdot (\theta - \xi_{i+\frac{1}{2}}) + \frac{\tilde{\mathcal{G}}''(\xi_{i+\frac{1}{2}})}{2} \cdot (\theta - \xi_{i+\frac{1}{2}})^2 + o((\theta - \xi_{i+\frac{1}{2}})^2) d\theta \right| + \left| \mu_i - \mathcal{G}_i \right| \right] \\ &\leq \frac{1}{N_b} \sum_{i=1}^{N_b} \left[\left| \frac{1}{h} \int_{\xi_i}^{\xi_{i+1}} \tilde{\mathcal{G}}'(\xi_{i+\frac{1}{2}}) \cdot (\theta - \xi_{i+\frac{1}{2}}) d\theta \right| \right. \\ &\quad \left. + \left| \frac{1}{h} \int_{\xi_i}^{\xi_{i+1}} \frac{\tilde{\mathcal{G}}''(\xi_{i+\frac{1}{2}})}{2} \cdot (\theta - \xi_{i+\frac{1}{2}})^2 + o((\theta - \xi_{i+\frac{1}{2}})^2) d\theta \right| + \left| \mu_i - \mathcal{G}_i \right| \right] \\ &= \frac{1}{N_b} \sum_{i=1}^{N_b} \left[\left| \frac{1}{h} \int_{\xi_i}^{\xi_{i+1}} \frac{\tilde{\mathcal{G}}''(\xi_{i+\frac{1}{2}})}{2} \cdot (\theta - \xi_{i+\frac{1}{2}})^2 + o((\theta - \xi_{i+\frac{1}{2}})^2) d\theta \right| + \left| \mu_i - \mathcal{G}_i \right| \right]. \end{aligned}$$

When N_r and N_b are large enough, note $h = \frac{\theta_2 - \theta_1}{N_b}$ and $P_i \approx h \cdot \tilde{\mathcal{G}}(\xi_{i+\frac{1}{2}})$, we can estimate the error by omitting terms of higher order and replacing $|\mu_i - \mathcal{G}_i|$ with the standard deviation σ_i

$$\begin{aligned}
\text{error} &\approx \frac{1}{N_b} \sum_{i=1}^{N_b} \left[\left| \frac{1}{h} \int_{\xi_i}^{\xi_{i+1}} \frac{\tilde{\mathcal{G}}''(\xi_{i+\frac{1}{2}})}{2} \cdot (\theta - \xi_{i+\frac{1}{2}})^2 d\theta \right| + \sigma_i \right] \\
&\leq \frac{1}{N_b} \sum_{i=1}^{N_b} \left[\frac{|\tilde{\mathcal{G}}''(\xi_{i+\frac{1}{2}})|}{24} h^2 + \sigma_i \right] \\
&= \left(\frac{1}{24N_b} \sum_{i=1}^{N_b} |\tilde{\mathcal{G}}''(\xi_{i+\frac{1}{2}})| \right) \frac{(\theta_2 - \theta_1)^2}{N_b^2} + \left(\frac{1}{N_b} \sum_{i=1}^{N_b} \sqrt{P_i(1-P_i)} \right) \frac{1}{h} \sqrt{\frac{1}{N_r}} \\
&\approx \left(\frac{1}{24N_b} \sum_{i=1}^{N_b} |\tilde{\mathcal{G}}''(\xi_{i+\frac{1}{2}})| \right) \frac{(\theta_2 - \theta_1)^2}{N_b^2} + \left(\frac{1}{N_b} \sum_{i=1}^{N_b} \sqrt{h\tilde{\mathcal{G}}(\xi_{i+\frac{1}{2}})} \right) \frac{1}{h} \sqrt{\frac{1}{N_r}} \\
&= \left(\frac{(\theta_2 - \theta_1)^2}{24N_b} \sum_{i=1}^{N_b} |\tilde{\mathcal{G}}''(\xi_{i+\frac{1}{2}})| \right) \frac{1}{N_b^2} + \frac{1}{\sqrt{\theta_2 - \theta_1}} \left(\frac{1}{N_b} \sum_{i=1}^{N_b} \sqrt{\tilde{\mathcal{G}}(\xi_{i+\frac{1}{2}})} \right) \sqrt{\frac{N_b}{N_r}}.
\end{aligned}$$

Let A and B denote the coefficients

$$A = \frac{(\theta_2 - \theta_1)^2}{24N_b} \sum_{i=1}^{N_b} |\tilde{\mathcal{G}}''(\xi_{i+\frac{1}{2}})|, \quad B = \frac{1}{\sqrt{\theta_2 - \theta_1}} \left(\frac{1}{N_b} \sum_{i=1}^{N_b} \sqrt{\tilde{\mathcal{G}}(\xi_{i+\frac{1}{2}})} \right). \quad (4.13)$$

We obtain the error estimation of ray tracing

$$\text{error} \approx A \frac{1}{N_b^2} + B \sqrt{\frac{N_b}{N_r}}. \quad (4.14)$$

The last approximation holds because $1 - P_i \approx 1$ when N_b is large. To compute the estimation we first need to calculate the second order derivative of $\tilde{\mathcal{G}}(\theta)$. Figure 4.5 reveals that the error decreases asymptotically to 0 as $\frac{1}{\sqrt{N_r}}$.

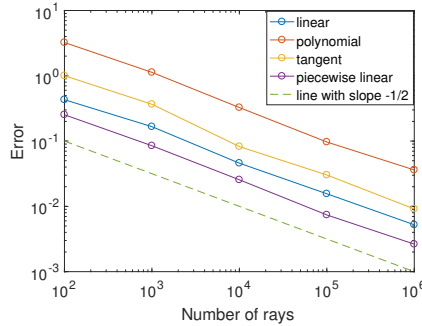


Figure 4.5: Error with respect to N_r in increasing cases, $N_b = 100$.

The dependency of the error on N_b is more complicated. The coefficients A and B both depend on N_b . But since they are the average value, we expect them to be constants when N_b is large. We compute A and B with various choices of N_b and the results is shown in Figure 4.6. Despite the change at the beginning, which corresponds to very small N_b and thus is somewhat meaningless, both coefficients remain stable when N_b is larger than 100. Then we compare the real error with the estimation. The results of the linear, tangent and polynomial examples are shown in Figure 4.7a-4.7c. In the linear case, due to the uniform intensity distribution ($\tilde{\mathcal{G}}'(\theta) = 0$), $A = 0$ and we get rid of the error in the first term. Thus the estimation is proportional to $\sqrt{N_b}$. And it

coincides well with the real error. In contrast, in the tangent and polynomial examples, when N_b is relatively small, omitting higher order terms causes large difference between the estimation and the real error. For the tangent example, as N_b increases, the estimation coincides better with the real error. But for the polynomial example, deviation of estimation from the real error is larger. But they present the same trend with the increase in the number of bins. Notice in both cases, as we can see from Figure 4.6, that the coefficient B is much larger than the coefficient A , therefore the second term dominates the error, that is, the error is proportional to the square root of N_b .

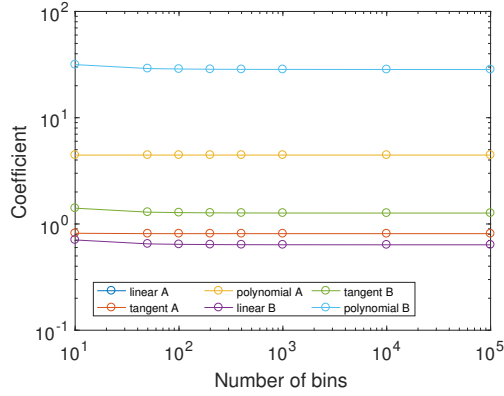


Figure 4.6: Coefficients A and B

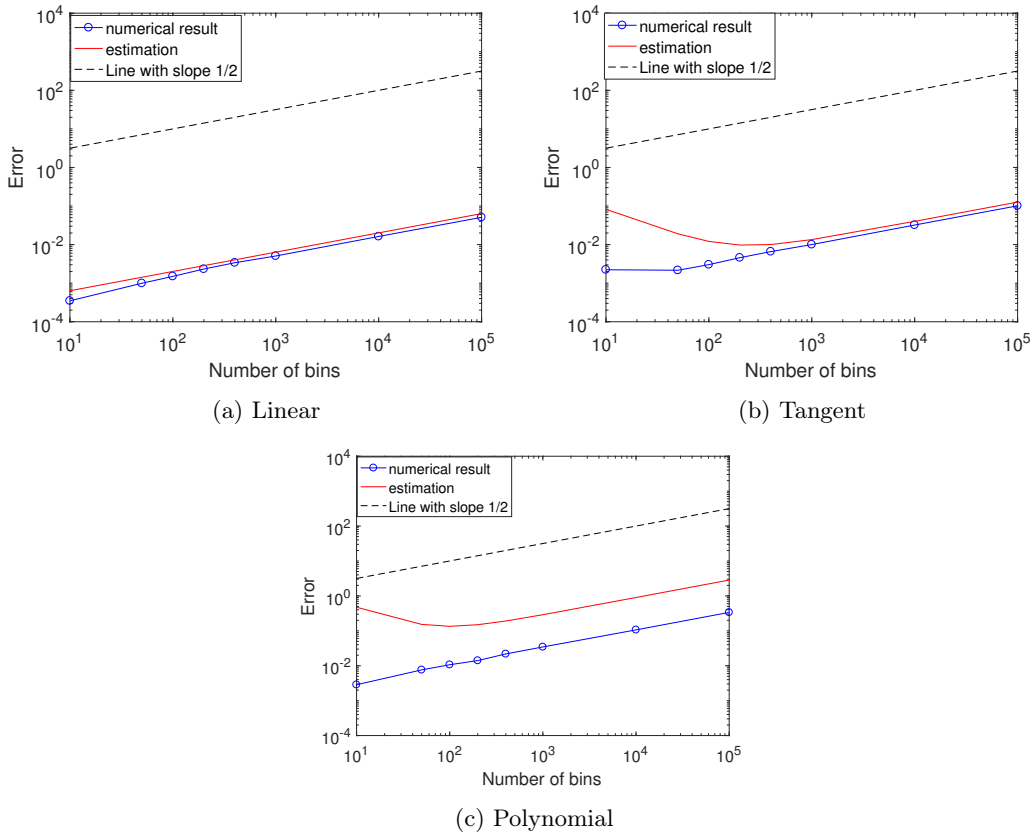


Figure 4.7: Error with respect to N_b , $N_r = 10^7$.

Chapter 5

Light Scattering

Light scattering is the deviation of light from the law of reflection. If a propagating light ray is incident on a perfectly smooth surface, the direction of the reflected light is determined by the law of reflection. In more realistic scenarios, as illustrated in Figure 5.1, the reflected light always contains a diffused component caused by scattering. Possible reflected rays are indicated by multiple rays in red in the figure. An idealized scatterer is the perfectly diffuse reflector which scatters light according to Lambert's cosine law. The scattering mechanism reveals the micro-structure and spatial configuration of the scattering medium and thus plays an important role in many areas. Rayleigh first investigated the reflection of acoustic waves in 1896 [19]. In 1919, Chenmoganadam derived a theory of scattered light based on the phase shift of the reflected beam due to the rough surface [20]. A quantity called the bidirectional reflectance distribution function (BSDF [3]) is introduced by Nicodemus in 1970 to completely describe scattering. In optical design, the inverse scattering problem is that of determining the properties of the scattering object (in our case it is the shape of the reflector), given the incident distribution and the scattering profile. In practice, since BSDF varies with different optical surface materials, a common approach is to determine the scattering mechanism from scattered light measurements. In our model, we simply adopt scattering that follows a certain probability distribution.

This chapter is organized as follows: Section 5.1 formally formulates the problem. Section 5.2 introduces the iterative compensation approach, which is initially used for optical design with extended source. In Section 5.3, a deconvolution algorithm is presented. We apply the iterative method and the deconvolution method to the model problem in Section 5.4 to investigate accuracy and efficiency. Finally these are applied to optical design problems.

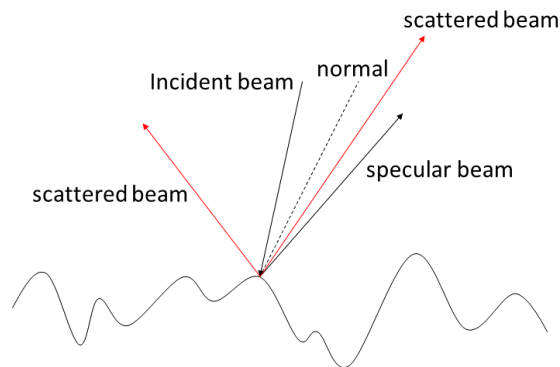


Figure 5.1: Scattering on a rough surface.

5.1 Problem formulation

To establish the model of diffuse reflection, we consider the propagation of a light ray as a stochastic process: A ray is emitted from the source with random direction τ and corresponding specularly reflected direction $\tilde{\theta}$, which is deterministic. The deviation $\Delta\theta$ occurs when the ray is reflected by the reflector, which is also a random variable following a certain distribution. The real reflected direction $\hat{\theta}$ consists of the specular component $\tilde{\theta}$ and the diffused component $\Delta\theta \in [-\pi/2 - \tilde{\theta}, \pi]$, i.e.

$$\hat{\theta} = \tilde{\theta} + \Delta\theta. \quad (5.1)$$

$\hat{\theta}$ is the sum of two random variables and thus is a random variable as well. To mathematically model the scattering problem, we first introduce some notations. Given a source with intensity distribution $\mathcal{I}(t)$, we aim to obtain a desired target distribution $\mathcal{G}_{\text{desired}}(\theta)$. With an imaginary specular reflector, we can obtain the output $\mathcal{G}_{\text{ref}}(\theta)$. Considering the scattering effect, we actually get $\mathcal{G}_{\text{scatt}}(\theta)$. Throughout this chapter, we always consider normalized intensity distributions. Then only the relative shape of the intensity distribution matters. We can scale the intensity distribution by adjusting the total energy of the source. In this setting, $\mathcal{G}_{\text{scatt}}(z)$ and $\mathcal{G}_{\text{ref}}(z)$ are actually the probability density functions of $\hat{\theta}$ and $\tilde{\theta}$, respectively. Let $k(z)$ denote the probability density function of $\Delta\theta$. Suppose $\tilde{\theta}$ and $\Delta\theta$ are independent, by conditioning on the value of $\tilde{\theta}$, we have

$$\begin{aligned} \mathcal{G}_{\text{scatt}}(z) &= \mathbb{P}(\hat{\theta} = z) = \int_{-\pi}^{\pi} \mathbb{P}(\tilde{\theta} = t) \mathbb{P}(\Delta\theta = z - t | \Delta\theta = t) dt \\ &= \int_{-\pi}^{\pi} \mathbb{P}(\tilde{\theta} = t) \mathbb{P}(\Delta\theta = z - t) dt \\ &= \int_{-\pi}^{\pi} \mathcal{G}_{\text{ref}}(t) k(z - t) dt \\ &= (\mathcal{G}_{\text{ref}} * k)(z). \end{aligned} \quad (5.2)$$

The probability distribution of $\hat{\theta}$ is the convolution of $\mathcal{G}_{\text{ref}}(z)$ and $k(z)$. The relation (5.2) reveals a fact that the output is "contaminated" due to diffuse reflection and thus presents a discrepancy from the desired target. This procedure is illustrated in Figure 5.2. Of course we hope that $\mathcal{G}_{\text{scatt}}(\theta)$ is close to the desired target distribution $\mathcal{G}_{\text{desired}}$.

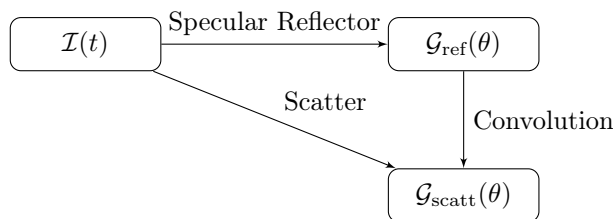


Figure 5.2: Illustration of the scattering problem.

In order to obtain the desired target distribution, modifications in the input should be considered to eliminate the effect of scattering. In other words, we should use a predefined target distribution, instead of the desired target distribution, as input to compute the reflector profile. This issue can be handled in two ways. The first method is natural: we compute a predefined target distribution by solving the deconvolution problem and use it as the input to generate the reflector profile. That is, we want to find a $\mathcal{G}_{\text{ref}}(\theta)$ such that

$$\mathcal{G}_{\text{desired}}(z) = (\mathcal{G}_{\text{ref}} * k)(z). \quad (5.3)$$

The other is an iterative method: First give an initial guess of the predefined target distribution. In each iteration, it is updated according to the relationship between the desired target distribution

and the obtained distribution to "compensate" for the effects of scattering. We will investigate both methods in this chapter. The probability density function $k(t)$ describes the scattering properties of the reflector. As aforementioned, in practical problems, the characteristic of scattering is determined from measurements. In this thesis, for the sake of convenience, we use diffusions that follow specific distributions to test our algorithms. These two distributions are given below

$$k_1(t) = \begin{cases} \frac{1}{2t_0} & \text{for } t \in [-t_0, t_0], \\ 0 & \text{otherwise,} \end{cases} \quad k_2(t) = \begin{cases} \frac{1-(2ct)^2}{4B(2,2)} & \text{for } t \in [-\frac{1}{2c}, \frac{1}{2c}], \\ 0 & \text{otherwise,} \end{cases} \quad (5.4)$$

where t_0, c are constants and $B(x, y)$ is the beta function, i.e.,

$$B(x, y) = \int_0^1 t^{x-1}(1-t)^{y-1} dt. \quad (5.5)$$

$k_1(t)$ is the uniform distribution, which is the simplest probability density function. $k_2(t)$ is a variant of the beta distribution probability density function. Despite the somewhat complicated form, it is essentially a polynomial inside its compact support and has properties of symmetry and continuity. Figure 5.3 shows probability density functions of these two types of diffusion.

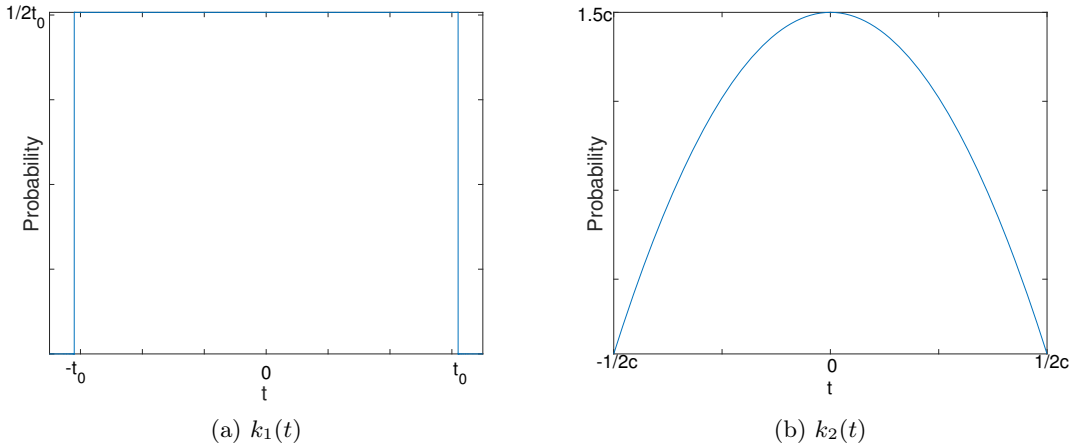


Figure 5.3: Probability density functions of diffusions.

5.2 Iterative compensation approach(ICA)

The iterative approach was first introduced by Rabl in [9], which was originally used to generalize the optical design from point source approximation to real extended source. In 2007, Bortz and Shatz applied an iterative approach to improve the beam-shaping performance of generalized functional designs [10]. The iterative compensation approach works in the following way: An initial design is computed by an available method and evaluated by ray tracing. The obtained result is compared to the desired target. Then the design is modified according to the observed discrepancy from the goal. This procedure is repeated many times until a satisfactory result has been reached. In our case, we use the method developed in Chapter 3 to generate the reflector and use the ray tracing technique developed in Chapter 4 to evaluate the performance of the optical system. To modify our design, we can either alter the predefined target in each iteration or modify the reflector shape. To elaborate the iterative process, we first specify some terminology. Our goal is to obtain the desired output denoted by $\mathcal{G}_{\text{desired}}(\theta)$ on the target, which is defined on the interval $[\theta_1, \theta_2]$. In every iteration, we need to predefine the target $\mathcal{G}_{\text{ref}}(\theta)$ to generate the reflector profile. We evaluate the actual intensity distribution $\mathcal{G}_{\text{scatt}}(\theta)$ using ray tracing. For perfect specular reflections, when $\mathcal{G}_{\text{ref}}(\theta) = \mathcal{G}_{\text{desired}}(\theta)$, $\mathcal{G}_{\text{scatt}}(\theta)$ is necessarily equal to $\mathcal{G}_{\text{desired}}(\theta)$.

However, for diffuse surfaces, there will always be some discrepancy between the predefined output and the target. In order to compensate the departure from the target, we apply correction to the predefined target, either enhance or reduce. More specifically, as illustrated in Figure 5.4, we first give an initial input, e.g., simply use the target intensity distribution $\mathcal{G}_{\text{desired}}(\theta)$. Then generate the reflector profile and evaluate the output distribution. If the error is sufficiently small, then we stop and get the the reflector profile, otherwise the initial design is revised according to the difference. Now we still need to give a definition of the error. Analogous to Chapter 4, we uniformly divide the target interval in small bins and define the error as the average deviation between the target midpoint values and actual midpoint values (represented by corresponding estimation):

$$[\theta_1, \theta_2] = [\xi_1, \xi_2) \cup [\xi_2, \xi_3) \cup \dots \cup [\xi_{N_b-1}, \xi_{N_b}) \cup [\xi_{N_b}, \xi_{N_b+1}], \quad \xi_i = \theta_1 + (i-1)h, h = \frac{\theta_2 - \theta_1}{N_b}, \quad (5.6)$$

$$\text{error} = \frac{1}{N_b} \sum_{i=1}^{N_b} \left| \mathcal{G}_{\text{scatt}}(\xi_{i+\frac{1}{2}}) - \mathcal{G}_{\text{desired}}(\xi_{i+\frac{1}{2}}) \right|, \quad (5.7)$$

where $\xi_{i+\frac{1}{2}} = \frac{\xi_i + \xi_{i+1}}{2}$.

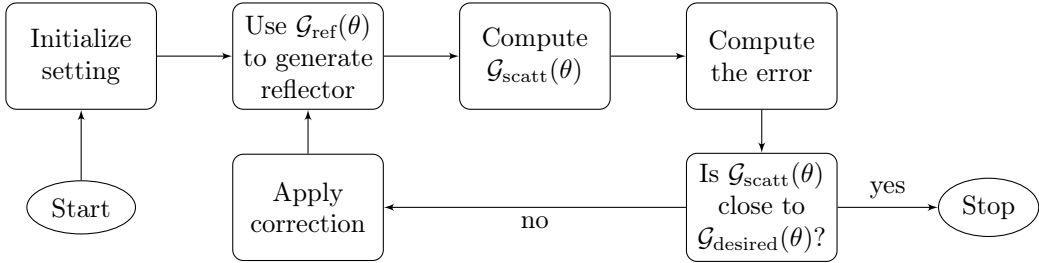


Figure 5.4: Flowchart of the iterative compensation algorithm

The correction of the predefined target can be defined in different ways. A simple modification is to update $\mathcal{G}_{\text{ref}}(\theta)$ as follows

$$\mathcal{G}_{\text{ref,new}}(\theta) = \begin{cases} \alpha \mathcal{G}_{\text{ref,old}}(\theta) + (1 - \alpha) \mathcal{G}_{\text{ref,old}}(\theta) \cdot \left(\frac{\mathcal{G}_{\text{desired}}(\theta)}{\mathcal{G}_{\text{scatt}}(\theta)} \right)^\beta \mathcal{G}_{\text{desired}}(\theta) \neq 0, \\ \mathcal{G}_{\text{max}}(\theta) & \mathcal{G}_{\text{scatt}}(\theta) = 0 \text{ or } \mathcal{G}_{\text{ref,new}}(\theta) > \mathcal{G}_{\text{max}}(\theta), \end{cases} \quad (5.8)$$

where $0 \leq \alpha \leq 1, \beta > 0$ are factors used to control the strength of the compensation. Issue arises when $\mathcal{G}_{\text{scatt}}(\theta) = 0$ that $\mathcal{G}_{\text{ref,new}}(\theta)$ tends to infinity. Therefore we restrict $\mathcal{G}_{\text{ref,new}}(\theta)$ to a maximum $\mathcal{G}_{\text{max}}(\theta)$. For instance, we may use $k \cdot \mathcal{G}_{\text{desired}}(\theta)$ as $\mathcal{G}_{\text{max}}(\theta)$, where k is a scale factor. It also helps to avoid the case that the increase in $\mathcal{G}_{\text{ref}}(\theta)$ doesn't rise the actual intensity and thus in return causes instability in $\mathcal{G}_{\text{ref}}(\theta)$.

Alternatively we can apply correction to the reflector shape, which is achieved by altering the predefined $t \rightarrow \theta$ relation $\theta_{\text{ref}}(t)$. This approach is based on cumulative flux distribution and is proposed in [11]. We first give an initial guess of the $t \rightarrow \theta$ relation and then use it to generate the reflector profile. Next we evaluate the actual intensity distribution and compute the cumulative flux functions of the desired and the actual distribution, which are denoted by $F_{\text{desired}}(\theta)$ and $F_{\text{scatt}}(\theta)$, respectively.

$$F_{\text{desired}}(\theta) = \int_{\theta_1}^{\theta} \mathcal{G}_{\text{desired}}(\theta') d\theta'. \quad (5.9)$$

Since we only have numerical results of $\mathcal{G}_{\text{scatt}}(\theta)$, the cumulative flux of the actual distribution $F_{\text{scatt}}(\theta)$ needs to be computed by numerical integration. By matching the desired cumulative flux of θ and the cumulative flux of t we obtain the desired $t \rightarrow \theta$ relation $\theta_{\text{desired}}(t)$. Analogously we obtain the actual $t \rightarrow \theta$ relation $\theta_{\text{scatt}}(t)$. By comparing $\theta_{\text{desired}}(t)$ and $\theta_{\text{scatt}}(t)$ we obtain the following update rule:

$$\theta_{\text{ref,new}}(t) = \theta_{\text{ref,old}}(t) + \theta_{\text{desired}}(t) - \theta_{\text{scatt}}(t). \quad (5.10)$$

The results of these two update rules are very similar, and thus we only present subsequently results using update rule (5.8). Generally, there is no guarantee that iterative compensation approaches will converge to a solution (in some cases solutions simply don't exist).

5.3 Deconvolution method

Deconvolution problems arise in many different fields such as image restoration, signal processing and nonparametric statistics, etc. It is usually ill-posed and thus is an algorithm-based process. There are various families of deconvolution algorithms, each with their own advantages and limitations. An overview of deconvolution algorithms is given in [14]. Various commercial software packages are available for deconvolution. In this section, we consider the following issue: Given $k(z)$ and $\mathcal{G}_{\text{desired}}(z)$, compute $\mathcal{G}_{\text{ref}}(z)$ from (5.3) via a deconvolution technique. To solve this problem, we first discretize (5.3) to obtain the matrix-vector form of the convolution and then solve the linear system of equations. Here we only introduce an algorithm that is relatively easy to implement: Split Augmented Lagrangian Shrinkage Algorithm (SALSA) [5].

5.3.1 Discretization

Discretization is the first step towards numerical computation. Convolution is essentially a process of integration and thus (5.3) can be discretized by quadrature rules. A general form of an integral reads

$$I[u] = \int_c^d u(t) dt. \quad (5.11)$$

A class of numerical methods work in the following framework. Find an approximating family $\{u_n(t)|n \geq 1\}$ to replace the integrand $u(t)$ and define the numerical integration as

$$I_n[u] = \int_c^d u_n(t) dt = I[u_n]. \quad (5.12)$$

The error of the numerical integration is defined as

$$E_n[u] = I[u] - I_n[u] = \int_c^d [u(t) - u_n(t)] dt. \quad (5.13)$$

The approximations $u_n(t)$ are required to satisfy

$$\|u - u_n\|_\infty \rightarrow 0 \text{ as } n \rightarrow \infty, \quad (5.14)$$

so that the error will converge to 0 as $n \rightarrow \infty$.

$$|E_n(u)| \leq \int_c^d |u(t) - u_n(t)| dt \leq (d - c)\|u - u_n\|_\infty. \quad (5.15)$$

Usually $\{u_n(t)|n \geq 1\}$ are chosen such that $I(u_n)$ is easy to evaluate. Most numerical integration formulas use polynomial or piecewise polynomial interpolation to define $u_n(t)$. The evaluation has the following form:

$$I_n[u] = \sum_{i=1}^{n+1} w_i u(t_i) \quad n \geq 1. \quad (5.16)$$

The coefficients w_i are called integration weights and t_i are the integration nodes, which usually lie in $[c, d]$. Standard methods have weights and nodes of simple form or they are tabulated in manuals and therefore it is not necessary to explicitly construct function $u_n(t)$. Here we introduce the well-known trapezoidal rule. The simple trapezoidal rule approximates $u(t)$ by the straight line joining $(c, u(c))$ and $(d, u(d))$. By integrating this linear function we obtain the approximation

$$I_1[u] = \frac{d - c}{2} [u(c) + u(d)]. \quad (5.17)$$

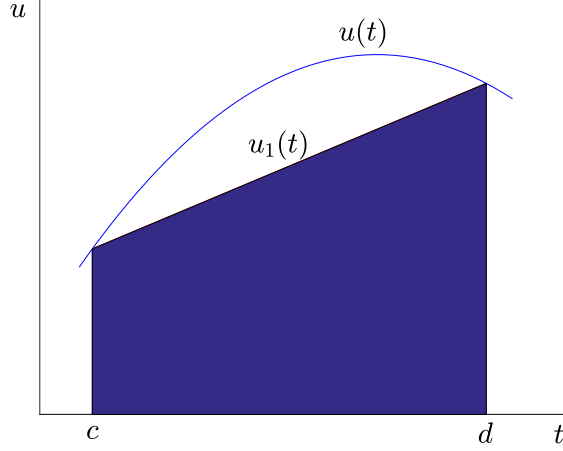


Figure 5.5: Illustration of trapezoidal rule.

This is the area of the trapezoid illustrated in Figure 5.5. When $d - c$ is large, the trapezoidal rule (5.17) is inaccurate. Then we can break the integral into the sum of integrals over small subintervals and apply (5.17) on each subinterval. For convenience, though not necessary, we divide $[c, d]$ into $n - 1$ intervals with the same width $h = \frac{d-c}{n-1}$. This gives the composite trapezoidal rule

$$I_n(u) = h \left(\frac{1}{2} u_1 + u_2 + \cdots + u_{n-1} + \frac{1}{2} u_n \right), \quad n \geq 1, \quad (5.18)$$

where $u_i = u(t_i)$, $t_i = c + (i - 1)h$. Suppose $u(t)$ is twice continuously differentiable, then the error of (5.18) is [25]

$$E_n(u) = -\frac{h^2(c-d)}{12} u''(\eta), \quad \eta \in (c, d). \quad (5.19)$$

Now we can discretize (5.3). Suppose $\mathcal{G}_{\text{desired}}, k, \mathcal{G}_{\text{ref}}$ are all compactly supported. Let $[a, b]$ and $[c, d]$ denote the support of $\mathcal{G}_{\text{desired}}$ and \mathcal{G}_{ref} , respectively. We divide the interval $[c, d]$ into n subintervals with equal width and use the trapezoidal rule to approximate $\mathcal{G}_{\text{desired}}(z)$:

$$\mathcal{G}_{\text{desired}}(z) \approx \frac{h}{2} \sum_{j=1}^{n-1} \left(\mathcal{G}_{\text{ref}}(t_j) k(z - t_j) + \mathcal{G}_{\text{ref}}(t_{j+1}) k(z - t_{j+1}) \right), \quad t_j = c + (j-1)h, \quad h = \frac{d-c}{n-1}. \quad (5.20)$$

By substituting m different values z_1, z_2, \dots, z_m in $\mathcal{G}_{\text{desired}}(z)$ and replacing $\mathcal{G}_{\text{desired}}(z_i)$, $k(z_i - t_j)$, $\mathcal{G}_{\text{ref}}(t_j)$ with g_i, K_{ij}, f_j , respectively, we obtain the matrix-vector form of the convolution:

$$\mathbf{g} = K\mathbf{f}, \quad \mathbf{g} \in \mathbb{R}_{\geq 0}^{m \times 1}, K \in \mathbb{R}_{\geq 0}^{m \times n}, \mathbf{f} \in \mathbb{R}_{\geq 0}^{n \times 1}, \quad (5.21)$$

where

$$\mathbf{g} = \begin{bmatrix} g_1 \\ \vdots \\ g_i \\ \vdots \\ g_m \end{bmatrix}, K = \frac{h}{2} \begin{bmatrix} k_{11} & \cdots & 2k_{1j} & \cdots & k_{1n} \\ \vdots & & \vdots & & \vdots \\ k_{i1} & \cdots & 2k_{ij} & \cdots & k_{in} \\ \vdots & & \vdots & & \vdots \\ k_{m1} & \cdots & 2k_{mj} & \cdots & k_{mn} \end{bmatrix}, \mathbf{f} = \begin{bmatrix} f_1 \\ \vdots \\ f_j \\ \vdots \\ f_n \end{bmatrix}.$$

Generally, the system is overdetermined, i.e., there are more equations than unknowns ($m \geq n$). Thus problem (5.21) can be regarded as a least squares problem defined by

$$\min_{\mathbf{f}} \frac{1}{2} \|\mathbf{g} - K\mathbf{f}\|_2^2, \quad \mathbf{g} \in \mathbb{R}_{\geq 0}^{m \times 1}, K \in \mathbb{R}_{\geq 0}^{m \times n}, \mathbf{f} \in \mathbb{R}_{\geq 0}^{n \times 1}. \quad (5.22)$$

Now we introduce the split augmented Lagrangian shrinkage algorithm (SALSA) to solve (5.22).

5.3.2 Split augmented Lagrangian shrinkage algorithm (SALSA)

Due to the ill-posedness of deconvolution [23], naively solving (5.22) usually doesn't work. We need to apply regularization to make the problem less sensitive. An alternative formulation applies a regularizer to the objective function, leading to the form

$$\min_{\mathbf{f}} \frac{1}{2} \|\mathbf{g} - K\mathbf{f}\|_2^2 + \tau\phi(\mathbf{f}). \quad (5.23)$$

$\phi(\mathbf{f})$ is the regularizer and τ is the regularization parameter that needs to be specified. We choose $\phi(\mathbf{f}) = \|\mathbf{f}\|_1$. An approach based on the principle of variable splitting ([12]) is proposed in [5]. The idea is to split the variable \mathbf{f} into a pair of variables, say \mathbf{f}_1 and \mathbf{f}_2 , each serves as the argument of the two functions with additional constraint, i.e.

$$\begin{aligned} \min_{\mathbf{f}, \mathbf{v}} \quad & h_1(\mathbf{f}_1) + h_2(\mathbf{f}_2), \\ \text{s.t.} \quad & G\mathbf{f}_1 = \mathbf{f}_2, \end{aligned} \quad (5.24)$$

where $h_1(\mathbf{f}_1) = \frac{1}{2} \|\mathbf{g} - K\mathbf{f}_1\|_2^2$, $h_2(\mathbf{f}_2) = \tau\phi(\mathbf{f}_2)$ and $G = I$ in this particular case. Although we introduce more variables and an extra equality constraint, problem (5.24) can be solved by the augmented Lagrangian method. To this purpose, we first introduce the augmented Lagrangian. Consider the following constrained optimization problem

$$\begin{aligned} \min_{\mathbf{z}} \quad & E(\mathbf{z}), \\ \text{s.t.} \quad & A\mathbf{z} - \mathbf{b} = \mathbf{0}. \end{aligned} \quad (5.25)$$

The so-called augmented Lagrangian function takes the equality constraint in (5.25) into account by augmenting the objective function with a weighted sum of the constraint functions.

$$\mathcal{L}(\mathbf{z}, \boldsymbol{\lambda}, \mu) = E(\mathbf{z}) + \boldsymbol{\lambda}^T (\mathbf{b} - A\mathbf{z}) + \frac{\mu}{2} \|\mathbf{b} - A\mathbf{z}\|_2^2, \quad (5.26)$$

We refer to $\boldsymbol{\lambda}$ as the vector of Lagrange multipliers. Besides the weighted sum, the objective function also includes the penalty term $\|A\mathbf{z} - \mathbf{b}\|_2^2$ with the penalty parameter $\mu \geq 0$ to robustify the algorithm [21].

$\mathcal{L}(\mathbf{z}, \boldsymbol{\lambda}, \mu)$ can be minimized by so-called augmented Lagrangian method (ALM) [13] in the following way: First fix $\boldsymbol{\lambda}$ and minimize $\mathcal{L}(\mathbf{z}, \boldsymbol{\lambda}, \mu)$ with respect to \mathbf{z} , then update $\boldsymbol{\lambda}$. This procedure is repeated several times until some convergence criterion is satisfied, which yields the following algorithm

Algorithm ALM Augmented Lagrangian method

- 1: Choose μ, \mathbf{z}_0 and λ_0 , set $k = 0$.
 - 2: **repeat**
 - 3: $\mathbf{z}_k \in \arg \min_{\mathbf{z}} \mathcal{L}(\mathbf{z}, \boldsymbol{\lambda}_k, \mu)$
 - 4: $\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \mu(\mathbf{b} - A\mathbf{z}_{k+1})$
 - 5: $k \leftarrow k + 1$
 - 6: **until** stopping criterion is satisfied.
-

Problem (5.24) can be transformed to problem (5.25) by using the following definitions

$$\mathbf{z} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{bmatrix}, \quad \mathbf{b} = \mathbf{0}, \quad A = [I \quad -I], \quad (5.27)$$

and

$$E(\mathbf{z}) = h_1(\mathbf{f}_1) + h_2(\mathbf{f}_2) = \frac{1}{2} \|\mathbf{g} - K\mathbf{f}_1\|_2^2 + \tau\phi(\mathbf{f}_2). \quad (5.28)$$

Substituting definitions (5.27) and (5.28) into the ALM algorithm yields the SALSA method:

Algorithm SALS Split augmented Lagrangian shrinkage algorithm

- 1: Choose $\mu, \mathbf{v}_0, \mathbf{d}_0$, set $k = 0$.
 - 2: **repeat**
 - 3: $\mathbf{f}'_k = \mathbf{v}_k + \mathbf{d}_k$
 - 4: $\mathbf{f}_{k+1} = \arg \min_{\mathbf{f}} \|K * \mathbf{f} - \mathbf{g}\|_2^2 + \mu \|\mathbf{f} - \mathbf{f}'_k\|_2^2$
 - 5: $\mathbf{v}'_k = \mathbf{f}_{k+1} - \mathbf{d}_k$
 - 6: $\mathbf{v}_{k+1} = \arg \min_{\mathbf{v}} \tau \phi(\mathbf{v}) + \frac{\mu}{2} \|\mathbf{v} - \mathbf{v}'_k\|_2^2$
 - 7: $\mathbf{d}_{k+1} = \mathbf{d}_k + \mathbf{f}_{k+1} - \mathbf{v}_{k+1}$
 - 8: $k \leftarrow k + 1$
 - 9: **until** stopping criterion is satisfied.
-

A good rule of thumb that $\mu = 0.1\tau$ is recommended in [5]. The initial vectors $\mathbf{v}_0, \mathbf{d}_0$ are simply chosen as zero vectors if no prior information is known. The algorithm will stop if relative change in the objective function falls below tolerance, whose default value is 10^{-5} . The convergence of SALS is guaranteed by the following theorem [15]:

Theorem 1 Consider problem (5.24), where h_1 and h_2 are closed, proper convex functions, and G has full column rank. Consider arbitrary $\mu > 0$ and $\mathbf{v}_0, \mathbf{d}_0$. Let $\{\eta_k \geq 0, k = 0, 1, \dots\}$ and $\{\nu_k \geq 0, k = 0, 1, \dots\}$ be two sequences such that

$$\sum_{k=0}^{\infty} \eta_k < \infty \text{ and } \sum_{k=0}^{\infty} \nu_k < \infty.$$

Consider three sequences $\{\mathbf{f}_k, k = 0, 1, \dots\}$, $\{\mathbf{v}_k, k = 0, 1, \dots\}$ and $\{\mathbf{d}_k, k = 0, 1, \dots\}$ that satisfy

$$\begin{aligned} \eta_k &\geq \left\| \mathbf{f}_{k+1} - \arg \min_{\mathbf{f}} \left(h_1(\mathbf{f}) + \frac{\mu}{2} \|G\mathbf{f} - \mathbf{v}_k - \mathbf{d}_k\|_2^2 \right) \right\|, \\ \nu_k &\geq \left\| \mathbf{v}_{k+1} - \arg \min_{\mathbf{v}} \left(h_2(\mathbf{v}) + \frac{\mu}{2} \|G\mathbf{f}_{k+1} - \mathbf{v} - \mathbf{d}_k\|_2^2 \right) \right\|, \\ \mathbf{d}_{k+1} &= \mathbf{d}_k + G\mathbf{f}_{k+1} - \mathbf{v}_{k+1}. \end{aligned}$$

Then, if (5.24) has a solution, the sequence $\{\mathbf{f}_k\}$ converges, $\mathbf{f}_k \rightarrow \mathbf{f}^*$, where \mathbf{f}^* is a solution of (5.24). If (5.24) does not have a solution, then at least one of the sequences \mathbf{v}_k or \mathbf{d}_k diverges.

In our case, $G = I$ and the full rank requirement of G is satisfied. If the minimization problems in step 4 and 6 are exactly solved (up to numerical precision), we find two sequences $\{\eta_k = 0, k = 0, 1, \dots\}$ and $\{\nu_k = 0, k = 0, 1, \dots\}$ that satisfy the conditions of Theorem 1 and thus it can be invoked to guarantee the convergence of SALS. The minimization problem in step 4 is transformed to the following linear system of equations by setting the derivative of $\|K * \mathbf{f} - \mathbf{g}\|_2^2 + \mu \|\mathbf{f} - \mathbf{f}'_k\|_2^2$ to zero.

$$\mathbf{f}_{k+1} = (K^H K + \mu I)^{-1} (K^H \mathbf{g} + \mu \mathbf{f}'_k). \quad (5.29)$$

The minimization problem in step 6 is by definition the Moreau proximal mapping of ϕ to \mathbf{f}'_k . When $\phi(\mathbf{v})$ is chosen to be $\|\mathbf{v}\|_1$, the solution is

$$\mathbf{f}_{k+1} = \text{sign}(\mathbf{f}'_k) \max\{|\mathbf{f}'_k| - \tau/\mu, 0\}. \quad (5.30)$$

For more details, we refer to the original paper [5].

5.4 Computed examples

In this section, we first test two algorithms with a model problem that has an analytical solution to investigate their accuracy and efficiency. It is not easy to compare algorithms because the

results obtained depend on multiple factors such as algorithm parameters and properties of input functions. For instance, how we choose the regularizer $\phi(\mathbf{f})$ and the parameter μ may have an impact on the performance of SALSA. But usually there is no clear clue to determine these parameters. Speed is another important aspect. It is also difficult to compare the speed of different algorithms, because it depends on the way the algorithms are implemented, the stopping criterion, etc. We will therefore not detail speed here. Generally, the deconvolution method takes less than 1 minute to compute $\mathcal{G}_{\text{ref}}(\theta)$ while the iterative compensation approach takes about 3 minutes to perform one iteration tracing 10^6 rays. Finally we apply them to several optical design problems.

5.4.1 The model problem

First, we appropriately choose the density function of the specular component $\tilde{\theta}$ and diffused component $\Delta\theta$ such that they are both continuous and have compact supports. In this experiment, we predefine $\mathcal{G}_{\text{ref}}(z), k(z)$ as raised cosine distributions. The range of $k(z)$ is half of that of $\mathcal{G}_{\text{ref}}(z)$.

$$\mathcal{G}_{\text{ref}}(z) = \begin{cases} \frac{2}{\pi}(1 + \cos 4z) & \text{for } z \in [-\frac{\pi}{4}, \frac{\pi}{4}], \\ 0 & \text{otherwise,} \end{cases} \quad k(z) = \begin{cases} \frac{4}{\pi}(1 + \cos 8z) & \text{for } z \in [-\frac{\pi}{8}, \frac{\pi}{8}], \\ 0 & \text{otherwise.} \end{cases}$$

Then $\mathcal{G}_{\text{desired}}(z)$ can be expressed as the convolution of $\mathcal{G}_{\text{ref}}(z)$ and $k(z)$, which is a piecewise trigonometric function. Note the support has been extended to $(-3\pi/8, 3\pi/8)$. Figure 5.6 shows the images of $\mathcal{G}_{\text{ref}}(z)$ and $\mathcal{G}_{\text{desired}}(z)$.

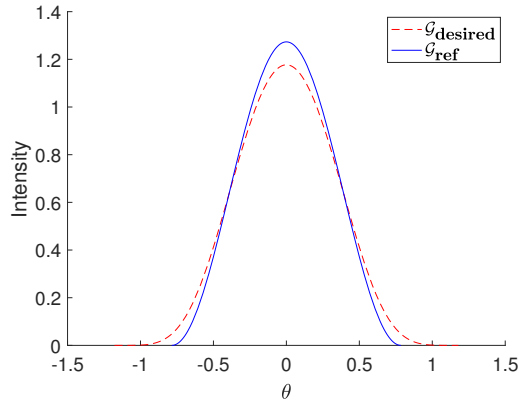


Figure 5.6: $\mathcal{G}_{\text{ref}}(z)$ and $\mathcal{G}_{\text{desired}}(z)$ of the model problem.

$$\mathcal{G}_{\text{desired}}(z) = \frac{1}{3\pi^2} \cdot \begin{cases} 9\pi + 24z + 8 \cos(4z) - \sin(8z) & \text{for } z \in [-3\pi/8, -\pi/8], \\ 2(3\pi + 8 \cos(4z)) & \text{for } z \in [-\pi/8, \pi/8], \\ 9\pi - 24z + 8 \cos(4z) + \sin(8z) & \text{for } z \in [\pi/8, 3\pi/8], \\ 0 & \text{otherwise.} \end{cases}$$

The optical design goal is to produce the desired output intensity $\mathcal{G}_{\text{desired}}(\theta)$ utilizing the uniformly distributed source

$$\mathcal{I}(t) = \frac{2}{\pi}, \quad t \in [-\pi/4, \pi/4].$$

We first test the deconvolution algorithm. To solve the least squares problem (5.22), we discretize the convolution using scheme (5.20). K is the matrix representation of the convolution operator. It has the sparse property. Figure 5.7 shows the sparsity pattern of K . The nonzero entries are indicated by solid circles.

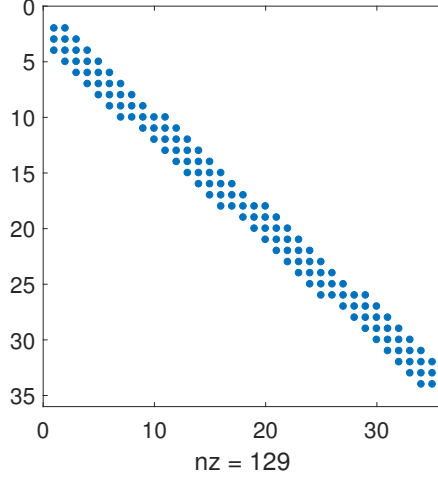
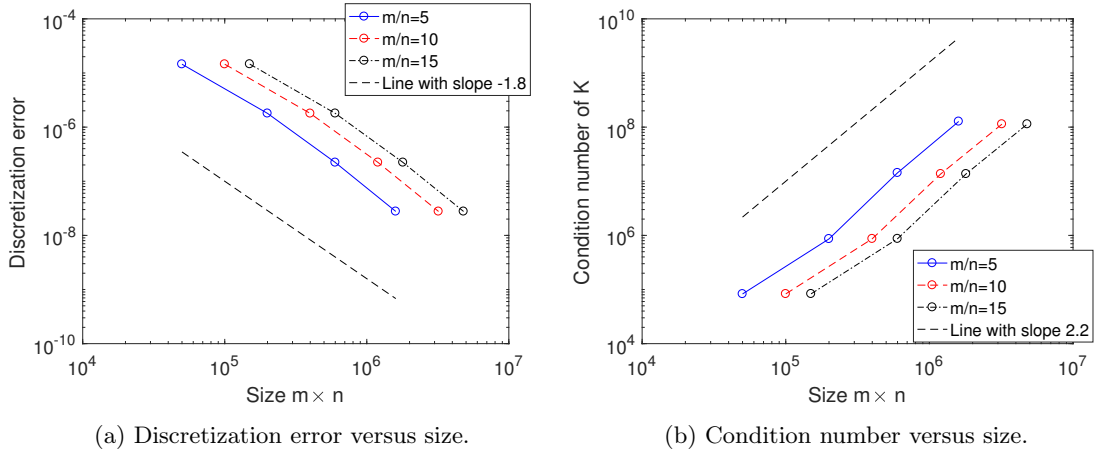


Figure 5.7: Sparsity pattern of matrix K of size 35×35 .



(a) Discretization error versus size.

(b) Condition number versus size.

Figure 5.8: Discretization error and condition number.

In the model problem, we know all functions $\mathcal{G}_{\text{ref}}(z)$, $k(z)$ and $\mathcal{G}_{\text{desired}}(z)$ and thus can compute the corresponding discrete version \mathbf{f} , K and \mathbf{g} . Due to the error in the numerical integration, we always get the discretization error $\|\mathbf{g} - K\mathbf{f}\|_2$, which depends on the grid spacing h . We compute the discretization error as a function of the size $m \times n$. Increasing n means smaller step size in the numerical integration, which will improve the accuracy. As shown in Figure 5.8a, the discretization error is independent of the ratio m/n and it decreases as the size increases. Nevertheless, another problem arises: As we can see from Figure 5.8b, the condition number of K , which is an indicator of the ill-posedness, rises at the same time. Again it is independent of the ratio m/n . The problem with matrices of larger condition number is more sensitive to the error. Then we need a tradeoff between the discretization error and the condition number. To compare the accuracy, we use SALSA to solve the least squares problem (5.22) with different matrix sizes. Let \mathbf{f}_{num} denote the numerical solution. The residual is defined as

$$\text{residual} = \|\mathbf{g} - K\mathbf{f}_{\text{num}}\|_2. \quad (5.31)$$

The results are presented in Figure 5.9. In this experiment, $m = 3000$, $n = 200$ is the best choice. But generally it varies in different settings. We will fix m and n at 3000 and 200 in the subsequent calculations.

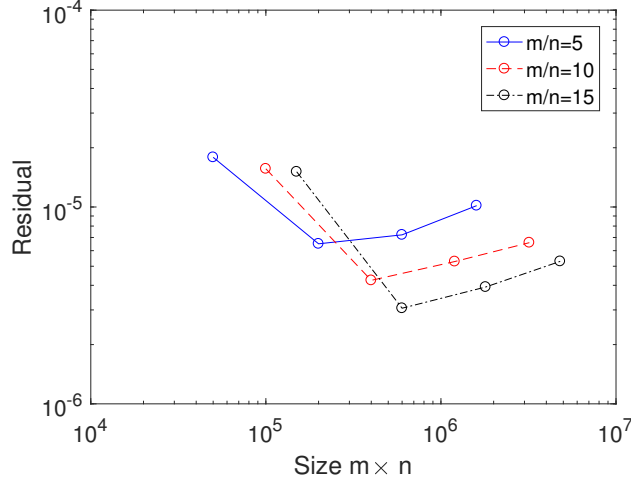
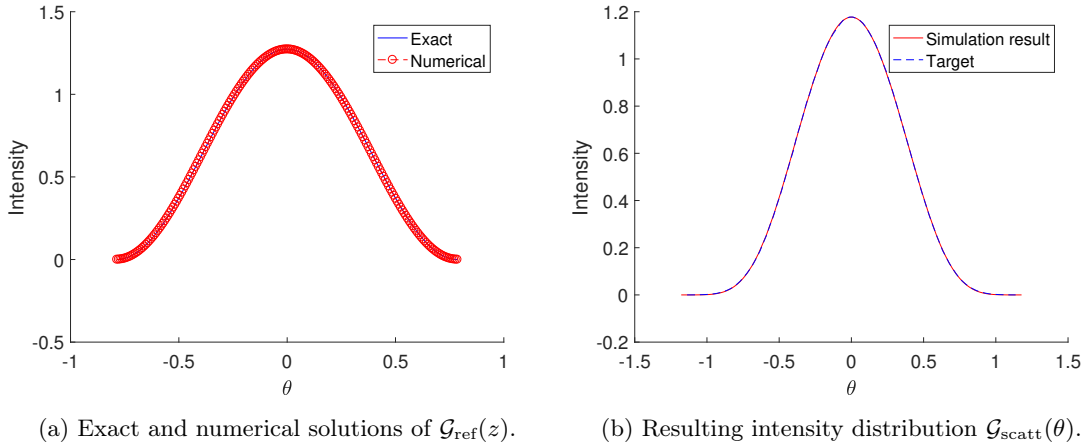


Figure 5.9: Residual versus size.



(a) Exact and numerical solutions of $\mathcal{G}_{\text{ref}}(z)$.

(b) Resulting intensity distribution $\mathcal{G}_{\text{scatt}}(\theta)$.

Figure 5.10: Resulting intensity distribution of deconvolution method, $N_r = 10^7$, $N_b = 100$.

The numerical approximation of $\mathcal{G}_{\text{ref}}(z)$ is displayed in Figure 5.10a, which is used as the predefined target to generate the reflector. Figure 5.10b shows the resulting intensity distribution. We traced 10^7 rays to obtain a fairly accurate simulation result such that it is hardly distinguishable from the desired target.

Next we examine the iterative compensation approach. We apply scheme (5.8) with parameters $\alpha = 0.2$, $\beta = 1$ to modify the predefined target distribution. The output intensity distributions using the iterative approach method are displayed in Figure 5.11. The black curve indicates the result obtained using the initial predefined distribution. A discrepancy can be observed at the center and the flanks. After 6 iterations, the output is close to the desired distribution. The convergence rate can be controlled by altering the parameters α and β , nonetheless they have little impact on the final intensity profile.

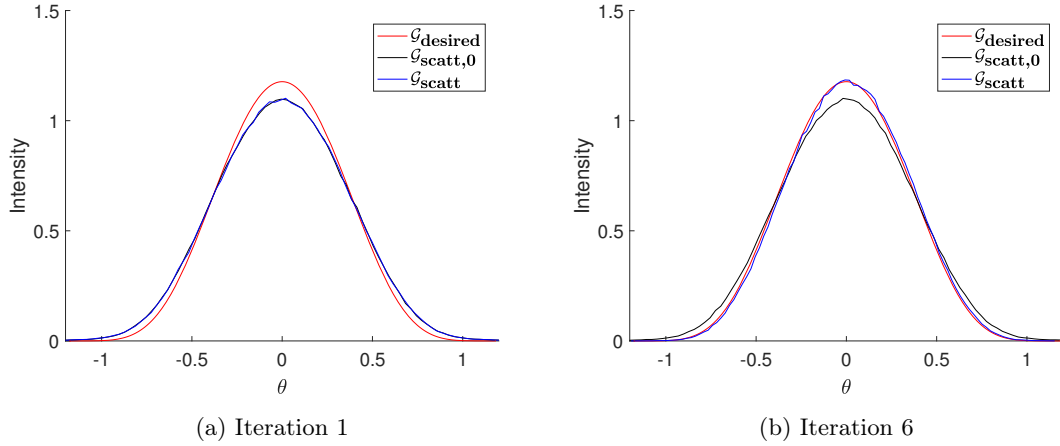


Figure 5.11: Resulting intensity distributions of ICA. We use the desired target distribution $\mathcal{G}_{\text{desired}}(\theta)$ as the initial predefined target (iteration=0) and the corresponding obtained output is plotted in black curve and labeled as " $\mathcal{G}_{\text{scatt},0}$ ". Obtained outputs after several iterations are plotted in blue curves and labeled as " $\mathcal{G}_{\text{scatt}}$ ".

Figure 5.12 shows the evolution of the error with different α values (β is fixed at 1). These curves all present a similar trend: After a fast declining phase, the error reaches the minimum and then fluctuates and increases again. Note that smaller α means stronger compensation strength and the curve reaches the minimum with less iterations. The minimum error of different curves are close. In order to compare the errors of ICA and SALSA and avoid the disturbance of the measure error (error in Monte Carlo ray tracing), all simulations are performed with $N_r = 10^6$ and $N_b = 100$ in both methods. The error of SALSA is much smaller than the errors of ICA.

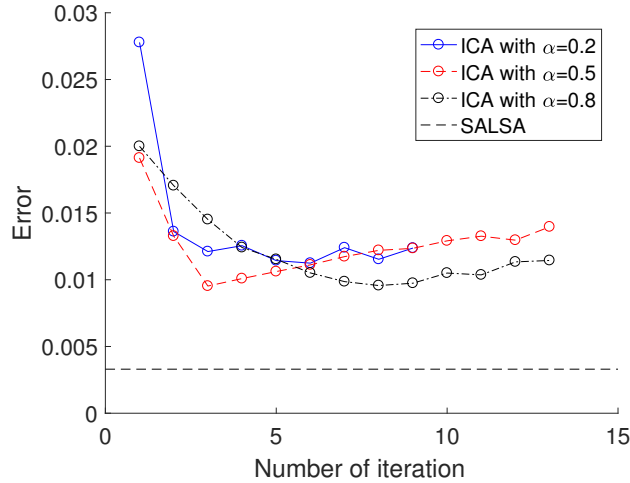


Figure 5.12: Evolution of the error during the iterative process of ICA. The error of the deconvolution method is plotted in dashed line for comparison. $N_r = 10^6$, $N_b = 100$.

In the model problem with ideal setting, SALSA works very well. There are several reasons for the good performance of SALSA: Both $\mathcal{G}_{\text{ref}}(z)$ and $k(z)$ are smooth, which decreases the discretization error. What's more, we know exactly the support $[c, d]$ of $\mathcal{G}_{\text{ref}}(z)$. Usually it serves as a input variable. Although for problems with analytical solution it might not matter much, the solution process for a general problem could be quite sensitive to the specified support $[c, d]$. On

the other hand, ICA is, to some extent, a blind method. Although it can't guarantee monotone decrease in the error, it achieves an overall good result.

5.4.2 Optical design problems

We will consider two optical design problems in this section: The first problem is the tangent example in Section 3.1, whose desired target distribution has a sharp cutoff. We would also like to investigate the effect of the diffusion extent in this problem. In the second problem, we require a uniform central distribution with gradual decay at the edge.

the tangent example

In the first example, the required output intensity distribution is symmetrically defined over output angles between $-\pi/8$ and $\pi/8$. We adopt diffusion $k_1(t)$ with several values parameter t_0 . We first use SALSA to compute the predefined intensity distribution and then use the predefined distribution to generate the reflector. Finally, 10^6 rays are traced to evaluate the output intensity distribution. The support of $\mathcal{G}_{\text{desired}}(z)$ in (5.3) is given while no prior information about the support of $\mathcal{G}_{\text{ref}}(z)$ is available. As a rule of thumb, it is set to be slightly larger than the support of $\mathcal{G}_{\text{ref}}(z)$. Parameters m and n are fixed at 12000 and 200 respectively.

The results we obtained using SALSA are displayed in Figure 5.13 and Table 5.1. As shown in Figure 5.13b, the predefined intensity distribution is extended to compensate the inadequacy of the intensity at the edges, but it also causes extra energy outside the desired range. Fluctuations are also observed, which cause difficulty in solving ODE system (3.16) and (3.17). Thus we employ the flux matching method to solve the $t \rightarrow \theta$ mapping. The resulting intensity distributions obtained are shown in Figure 5.13a. Due to the loss of energy outside the desired domain, the resulting intensity is overall lower than the target but it preserves the shape. The diffusion extent (i.e., the parameter t_0) has significant effect on the final result. Larger extent will cause more serious smearing at the edge. As we can see from Table 5.1, diffusion with extent $\pi/32$ and $\pi/16$ slightly extend the support of the output intensity distribution while diffusion with extent $\pi/8$ doubles the support.

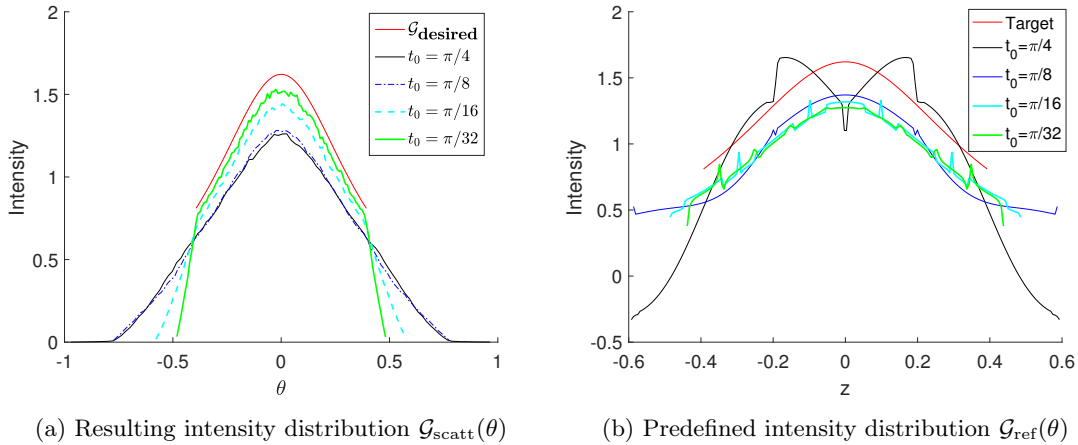


Figure 5.13: Resulting intensity distribution $\mathcal{G}_{\text{scatt}}(\theta)$ and corresponding predefined distribution $\mathcal{G}_{\text{ref}}(\theta)$ using SALSA, $N_r = 10^6$, $N_b = 100$.

Table 5.1: Obtained support of resulting intensity distributions in different cases

t_0	target	$\pi/32$	$\pi/16$	$\pi/8$	$\pi/4$
Support	$[-0.3927, 0.3927]$	$[-0.4861, 0.4864]$	$[-0.5831, 0.5827]$	$[-0.7811, 0.7813]$	$[-0.9816, 0.9746]$

Next we apply the iterative compensation approach to this problem. Again we adopt the diffusion model $k_1(t)$ and set $t_0 = \frac{\pi}{16}$. The desired target is used as the initial predefined distribution, which is revised in every iteration according to (5.8) with parameters $\alpha = 0.7, \beta = 1$. The results are shown in Figure 5.14.

The iterative compensation approach is not feasible in this case. In the first stage, the intensity around the peak is dramatically compensated, nevertheless, smearing of the edges of the target distribution remains serious. And we can see from Figure 5.14d that the error decreases in this stage. However overshoot is observed after iteration 5 and the error increases. One of the reasons for the smearing is that the iterative compensation approach is ineffective when the desired intensity is 0. This occurs when points where predefined target values are 0 yield nonzero actual values, but we cannot "compensate" it by predefined a negative target value. The intensity can never be reduced and according to energy conservation, there are always deficiencies at the edges. An alternative solution is to reduce the support of the predefined target distribution, but this causes other problems. We will not further investigate this issue in this thesis.

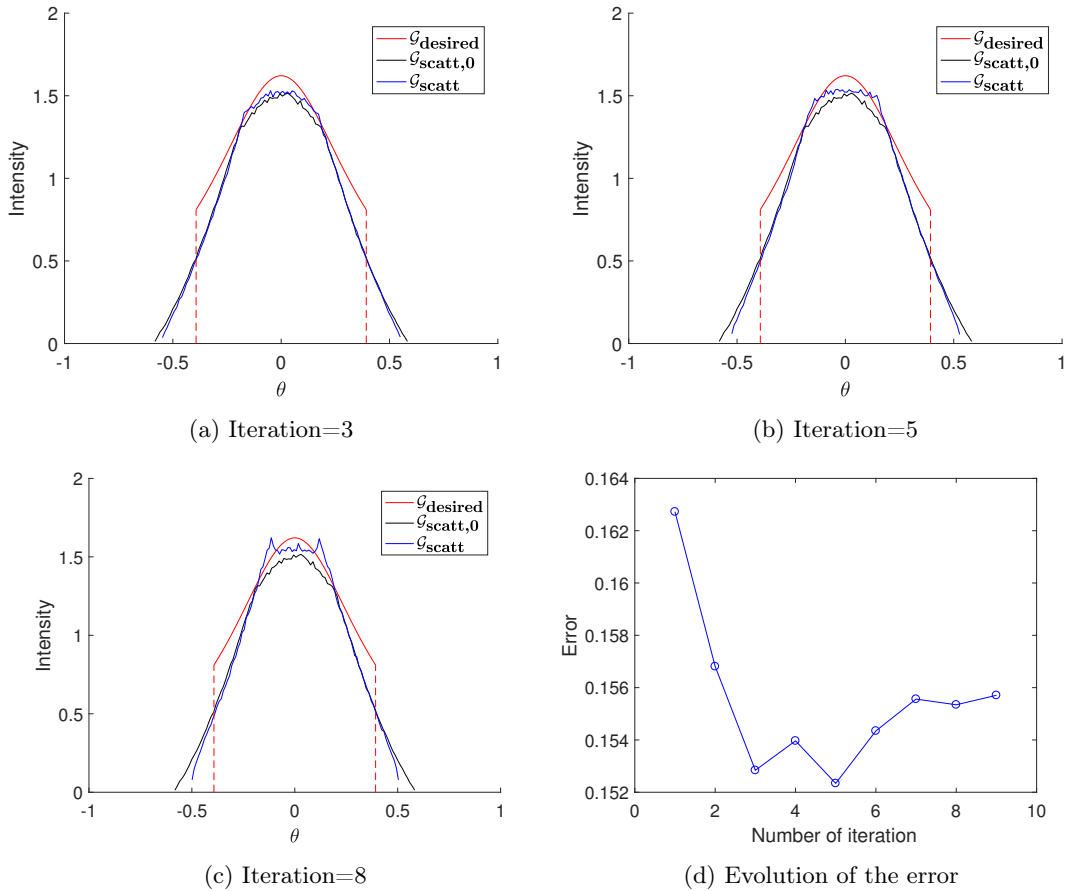


Figure 5.14: Resulting intensity distribution using ICA, the output using initial predefined target distribution is labeled as " $\mathcal{G}^{\text{scatt},0}$ ", $N_b = 100, N_r = 10^6$.

Uniform target example

In this example, we require a uniform distribution at the center which gradually decays to 0 at the edges. More specifically, the desired target is defined as follows

$$\mathcal{G}_{\text{desired}}(\theta) = \frac{4}{3\pi} \cdot \begin{cases} 0 & \text{for } \theta \in (-\infty, -\pi/4) \cup (\pi/4, +\infty), \\ 1 - \cos(8\theta) & \text{for } \theta \in [-\pi/4, -\pi/8] \cup (\pi/8, \pi/4), \\ 2 & \text{for } \theta \in [-\pi/8, \pi/8]. \end{cases}$$

The source is again uniform in the range of $[-\pi/4, \pi/4]$. The diffusion density function is $k_2(t)$ with support $[-0.25, 0.25]$.

A commonly observed effect of scattering is the gentle fall off and consequent deficiency in the uniform portion. These two methods once again display different features.

We first test the deconvolution method. Similarly, we discretize (5.3) with parameters $m = 8000, n = 200$ and use SALSA to solve the linear squares problem (5.22). The solution is used as the predefined target distribution to generate the reflector profile. As can be seen from Figure 5.15, the output intensity distribution preserves the shape at the center well while producing smearing at the edges. The distribution extent is doubled.

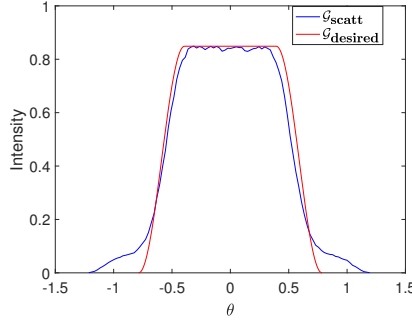


Figure 5.15: Result of the deconvolution method.

ICA performs better in this case. By compensating the predefined intensity distribution according to (5.8) with parameters $\alpha = 0.5, \beta = 1$ during the design process, a series of reflector profiles are obtained. The convergence history of the iterative algorithm is shown in Figure 5.16a-5.16d, which is the comparison between original result and the result after iteration. The iterative compensation approach remarkably reduces the discrepancy between the target and the obtained intensity distribution: As opposed to the deconvolution method, sharper fall offs are observed from iteration 1 to iteration 9 and the intensity is significantly compensated at the center. The most noticeable characteristics are an increasing rise at the edges. In other words, the iterative compensation approach creates the oscillation which is also reflected in the error plot Figure 5.16e: The error is reduced by a factor of 1.5 from an initial value of 0.037 to a final value of 0.024 after the eighth iteration, but slightly increases after the ninth iteration.

Conclusion

We proposed two methods to eliminate scattering effect in optical design. Generally, the deconvolution and iterative compensation approach are not feasible in all cases. For problems that have a solution, both methods perform quite well and deconvolution method is even more accurate. When the solution is not guaranteed, it's also worth employing these methods to enhance the design.

The first method is that first solve the deconvolution to obtain a predefined target distribution and use this distribution to generate the profile. Deconvolution is a computational technique belonging to the ill-posed problem category. Generally, the exact solution of the deconvolution

problem might not exist. Thus we use SALSA to solve the corresponding least squares problem. The accuracy of this method depends on the size of the coefficient matrix and the specified algorithm parameters. We determine these variables by trial and error. The resulting output intensity distribution usually coincides well with the desired target at the center, but it contains smearing at the edges. It is also a relatively efficient method compared to the iterative compensation approach, although the algorithm requires large amount of computation.

The convergence of the iterative compensation approach is not theoretically guaranteed, but it is capable in many cases. The main advantage of this method is its flexibility. We only need to determine a few parameters that control the strength of compensation. One disadvantage is that ICA has no effect on the points where the target intensity is 0. Thus ICA might not be a wise choice for targets with sharp cutoff. In general, ICA yields more oscillatory solutions, sometimes quite large overshoots can be observed. Another obvious drawback of ICA is that it invokes ray tracing in every iteration and thus is much more time-consuming than deconvolution.

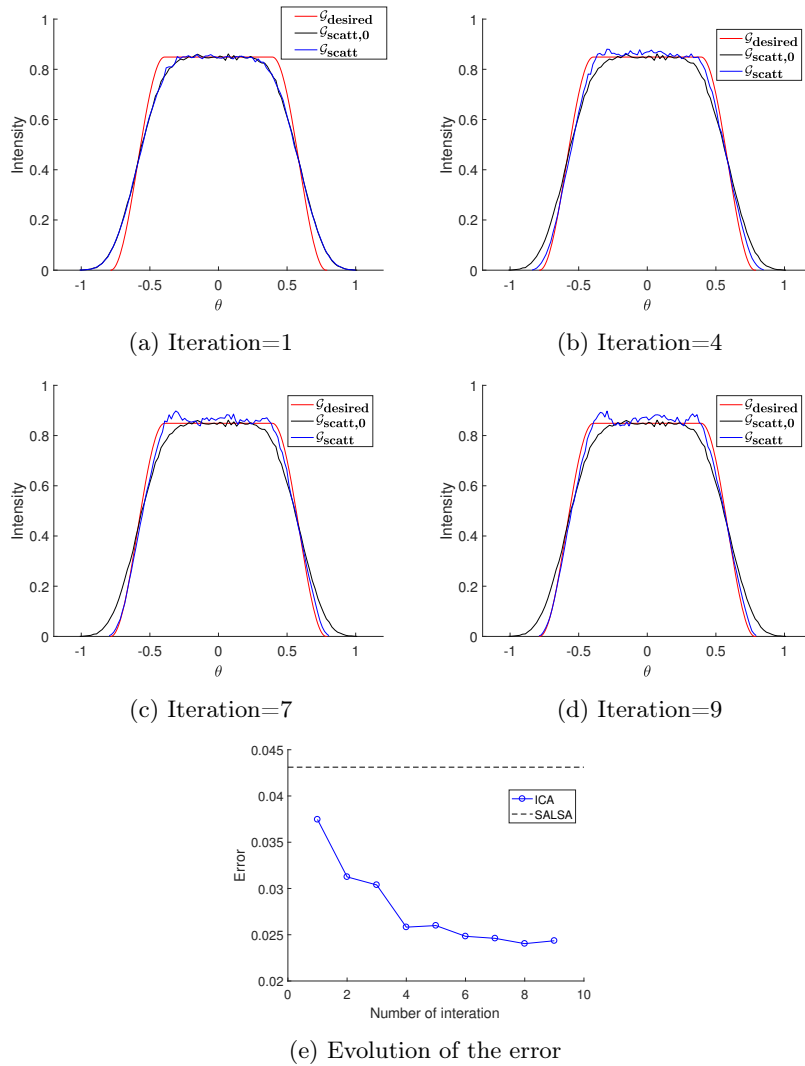


Figure 5.16: Results of ICA. We use the desired target distribution $\mathcal{G}_{\text{desired}}(\theta)$ as the initial pre-defined target (iteration=0) and the corresponding obtained output is plotted in black curve and labeled as " $\mathcal{G}_{\text{scatt},0}$ ". Obtained outputs after several iterations are plotted in blue curves and labeled as " $\mathcal{G}_{\text{scatt}}$ ".

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this thesis, we investigated the reflector design technique for the cylindrically symmetric case with linear source. Due to the symmetry, this 3D problem can be transformed to a 2D problem with point source. The basic theories that we used are the law of reflection and energy conservation. The main results of this thesis consist of three parts:

- In Chapter 3, we proposed the technique to generate the reflector profile under the specular reflection assumption.
- In Chapter 4, Monte Carlo ray tracing technique was introduced to simulate the performance of the optical system. We also derived an estimation of the error. The computation results in Chapter 3 were evaluated and verified.
- In Chapter 5, scattering was taken into account and we implement two algorithms to solve the scattering problem.

We will describe in detail our work in these three areas. Our first step is to create a reflector generation algorithm, which serves as the headstone of the subsequent work. First we derive an ordinary differential equation governing the shape of the reflector and obtain the integral expression of the solution with respect to the direction angles t and θ . The relation $t \rightarrow \theta$ is determined using energy conservation by solving the integral equation. This procedure can be accomplished either by the built-in function `ode45` of matlab or by matching the cumulative flux of the source and that of the target. The accuracies of both methods depend on the smoothness of the desired target distribution. In general, the flux matching method is more accurate. What's more important, it is also more robust against the oscillations in the input function and thus is recommended for calculation.

To simulate the output of the optical system, we employ Monte Carlo ray tracing to trace the propagation of a light ray through the system: We randomly choose a ray emitted from the source, compute the intersection of the ray with the reflector and calculate the reflection direction $\tilde{\theta}$. The output intensity distribution is given after tracing millions of rays and collect their reflection direction $\tilde{\theta}$ in bins. The ray tracing error depends on the number of rays N_r and the number of bins N_b . Unfortunately it is a relatively slow procedure, the error decreases according to $1/\sqrt{N_r}$ and the dependence on N_b may vary in different case. By experimental verification, all designs in Chapter 3 were demonstrated to be valid.

Reflectors with diffuse reflection is a more realistic model. Therefore we include the scattering effect in our design procedure. The propagation of rays is treated as a stochastic process. A ray is emitted with a random angle and it hits the reflector at a certain point and then deviates from the original direction with a random deviation angle following some specific distribution. The output intensity distribution is the deconvolution of the specularly reflected output and the underlying

density function of the deviation angle. We have two alternative methods to handle this problem. One is that first solve the deconvolution problem and then use the result as input to generate the reflector. The other is to iteratively modify the predefined target to compensate the scattering effect. These methods both perform well in the model problem.

Deconvolution is an ill-posed process. We employ SALSA to solve this problem. SALSA is a heuristic algorithm that converges fast and is robust to the error. However, how to determine the setting of parameters, which may have significant effect on the results, remains unknown. We manually adjusted them during our calculation. A common effect of deconvolution method is the serious smearing at the edges.

The iterative compensation approach is flexible, regardless of the desired target distribution and the diffusion density function. It is inefficient when the predefined target distribution yields serious smearing at the edges. Thus the initial predefined distribution should be considered as a variable. It seems that this method ultimately leads to overshooting in the obtained intensity distribution and it's important to control the strength of the compensation.

In conclusion, we completely accomplished the task of reflector design under the specular reflection assumption and the evaluation of optical design. Problems including scattering effect are studied and still need improvement.

6.2 Future work

The work is not finished yet, and a couple of issues remain to be investigated:

- The first one is how to determine the initial predefined target for the iterative compensation approach. As we have seen in Chapter 5, this initial guess has significant effect on the final results and simply using the desired target is not always a wise choice.
- Moreover, all the examples in this thesis are somewhat unrealistic: Usually the intensity of the source changes according to the emitted angle, rather than uniform. The desired target distributions we used are also ideal. The characterisation of the scattering property is itself a complicated topic. Thus we cannot adopt a simple function to describe the scattering. Considering these reasons, it is necessary to test our algorithms on some real data sets.
- Another challenge is to extend our design to a finite source. This requires totally different techniques and will be much more difficult.

Bibliography

- [1] "*History of Optics and Lenses*". <http://www.glasseshistory.com/glasses-history/history-of-optics/>.
- [2] "*History of optics - Wikipedia*". https://en.wikipedia.org/wiki/History_of_optics.
- [3] F. E. Nicodemus. *Reflectance nomenclature and directional reflectance and emissivity*. Applied Optics/ Vol. 9, No. 6, June 1970.
- [4] M. Maes. *Mathematical methods for reflector design*. PhD thesis, University of Amsterdam, 1997.
- [5] Manya V. Afonso, José M. Bioucas-Dias, and Mário A. T. Figueiredo. *Fast Image Recovery Using Variable Splitting and Constrained Optimization*. Advances in Neural Information Processing Systems 15 (NIPS 2002), pages 1065-1072, December 2002.
- [6] A. S. Glassner. *An introduction to Ray Tracing*. Academic Press, 1991.
- [7] C. E. Gutiérrez. *The Monge–Ampère Equation*. Progress in Nonlinear Differential Equations and Their Applications, vol. 44, 2001.
- [8] C. Prins. *Inverse Methods for Illumination Optics*. PhD thesis, Eindhoven University of Technology, 2014.
- [9] P.T. Rabl, A. Ong, J.M. Gordon, and W. Cai. *Iterative algorithm for reflector design for non-isotropic sources*. In Proc. SPIE, volume 2538, 1995.
- [10] J. Bortz and N. Shatz. *Iterative generalized functional method of nonimaging optical design*. In R. Winston & R. J. Koschel, editor, Nonimaging Optics and Efficient Illumination Systems IV, volume 6670, 2007.
- [11] W. J. Cassarly. *Iterative reflector design using a cumulative flux compensation approach*. In Proc. SPIE, volume 7652, pages 76522L–76522L–9, 2010.
- [12] R. Courant. *Variational methods for the solution of problems with equilibrium and vibration*. Bulletin of the American Mathematical Society, vol. 49, pp. 123, 1943.
- [13] J. Nocedal, S. J. Wright. *Numerical Optimization*. 2nd Edition, Springer, 2006.
- [14] Jean-Baptiste Sibarita. *Deconvolution Microscopy*. Adv Biochem Engin/Biotechnol (2005) 95: 201– 243.
- [15] J. Eckstein and D. Bertsekas, *On the DouglasRachford splitting method and the proximal point algorithm for maximal monotone operators* Mathematical Programming, vol. 5, pp. 293-318, 1992.
- [16] N. G. Boldyrev. *About calculation of asymmetrical specular reflectors*. Svetotekhnika 7, 7-8 (1932).

- [17] P. Dutré, K. Bala, and P. Bekaert. *Advanced global illumination*. (AK Peters, Wellesley, Mass., 2006)
- [18] W. J. Cassarly. *Illumination merit functions*. presented at Nonimaging Optics and Efficient Illumination Systems IV, San Diego, CA, USA, SPIE Vol. 6670 (2007).
- [19] Rayleigh. *The Theory of Sound*. Vol. II, Dover, N. Y. (1945)
- [20] T. K. Chenmoganadam. *On the Specular Reflection from Rough Surfaces*. Phys. Rev. 13:96-101 (1919).
- [21] A. N. Iusem. *Augmented Lagrangian Methods and Proximal Point Methods for Convex Optimization*. Investigacion Operativa, Volume 8, pp. 1-3, July -December 1999.
- [22] "*List of ray tracing software*". https://en.wikipedia.org/wiki/List_of_ray_tracing_software
- [23] SEDKI M. RIAD. *The Deconvolution Problem: An Overview*. Proceeding of The IEEE VOL. 74, NO. 1, January 1968.
- [24] "*matlab documentation*". <https://nl.mathworks.com/help/matlab/ref/odeset.html>
- [25] K. E. Atkinson *An Introduction to Numerical Analysis*. Wiley, 1989, pp.253.