LASER BEAM SHAPING AND TRANSFORMATION USING FREEFORM SURFACE DESIGN

by

Xiangchao Zhu

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
May 3, 2014

Approved by

Hulya Kirkici, Chair, Professor of Electrical and Computer Engineering
Stanley Reeves, Professor of Electrical and Computer Engineering
Michael Baginski, Associate Professor of Electrical and Computer Engineering

Abstract

Lasers are used in many areas including material processing (lithography, semiconductor manufacturing), welding, cutting, drilling, medical procedures (eye surgery and cosmetic skin treatments), and optical data processing. Laser diodes are the most developed lasers in recent years, due to their multiple features like low cost, compactness, electronic compatibility, broad range of wavelengths, and high pulse prepetition frequency values. On the other hand, the irradiance distribution of a laser-diode is elliptical Gaussian, not circular. In order to be widely used commercially, the beams from the laser diodes must be shaped to a desired shape through optical elements by the redistribution of energy profile into other beam irradiance profiles [3].

This work focuses on redistribution of Gaussian profile laser beam irradiance to other forms propagating through an optical media. In this work, single lens system is designed by getting surface parameters which are the coordinates of the surface points through solving the first order partial differential equations based on Snell's law and energy conservation principle. Meantime, a lens array is applied to the multi-laser source with the Gaussian distribution. The simulation results show that this method can be very well used for beam shaping. For the two lens system, the surface parameters are solved by the same method used in one freeform lens design. Meanwhile, another new method based on the separated variable mapping is also used in acquiring the surface parameters. This separated variable mapping means that the surface parameters of the source plane and target plane can be numerically specified based on "source-to-target" [4] or "target-to-source" [5,6]. And the mapping is calculated based on energy conservation principle, Snell’s law, and constancy of optical path length. According to comparison of the simulation results from two different methods, the method based on solving partial differential equations is more effective and efficient to transform a collimated Gaussian laser beam into a rectangular “flat-top” one, and the uniformity is almost 90%. Also, a lens array is applied to the multi-laser source with Gaussian distribution. The simulation results again show that the method based on solving partial differential equations is better than the other one.

Acknowledgments

The author would like to thank his advisor Dr. Hulya Kirkici, who has been giving so much patience and guidance during the past two years that allows the author to finally finish this project. Without her support this project could not come to the current level of performance. The author would like to thank the other two committee members, Dr. Baginski who taught him a lot of knowledge of electromagnetic fields and Dr. Reeves who guided him to gain the knowledge of using MATLAB to do the simulation during the classes. The author also would like to express appreciation to his research group members, thanks for all the helps that they have given to support him. Finally, the author wishes to thank his parents, Zhian Zhu and Hongying Qian, for their consistent encouragement and care.

Table of Contents

List of Tables

List of Illustrations

List of Abbreviations

VCSEL	Vertical Cavity Surface Emitting Laser diode
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
OPL	Optical Path Length
NURBS	Non-uniform rational basis spline

CHAPTER 1

BACKGROUND AND RESEARCH OBJECTIVES

1.1 Laser diode

A laser diode is electrically a P-I-N diode. The active region of the laser diode is in the intrinsic (I) region, and the carriers, electrons and holes, are pumped into it from the N and P regions respectively. While initial diode laser research was conducted on simple P-N diodes, all modern lasers use the double-heterostructure implementation, where the carriers and the photons are confined in order to maximize their chances for recombination and light generation. Unlike a regular diode used in electronics, the goal for a laser diode is that all carriers recombine in the (I) region, and produce light. Thus, laser diodes are fabricated using direct bandgap semiconductors. The laser diode epitaxial structure is grown using one of the crystal growth techniques, usually starting from an N doped substrate, and growing the (I) doped active layer, followed by the P doped cladding, and a contact layer. The active layer most often consists of quantum wells, which provide lower threshold current and higher efficiency [1]. A semiconductor laser diode is shown schematically in Figure 1.1, where the gain medium in the middle is surrounded by the guiding layers. The refractive index of the guiding layers is somewhat greater than that of the surrounding regions (substrate and cladding) so that the light can be confined to a relatively narrow region by total internal reflection. The two ends of the diode in the cross direction (x-y plane) are cleaved to form perfectly smooth, parallel edges, forming a Fabry-Perot resonator, and thus an optical cavity is formed. When the electrical current is injected through the positive electrode and collected at the base-plate on the bottom side of the junction (ground electrode), the original stable state of thermal equilibrium of the junction will be disturbed, and electrons can absorb energy either from heat caused by the collision among the massively increased amount of electrons or from the photons caused by the spontaneous emission which is necessary to initiate the laser oscillation. As a result, there are more electrons in higher energy states than in lower energy states, thus population inversion is achieved in the gain medium within the optical cavity.

When an electron is excited from a lower to a higher energy level, it will not stay that way forever; instead it will go back to the ground state and emit photons. These photons will travel along the waveguide and be reflected several times from the end faces before they are emitted out. As the light wave passes through the cavity, it is amplified by the stimulated emission which generates another photon of the same frequency, travelling in the same direction, with the same polarization and phase as the first photon. This is how laser diode works and generates laser light.

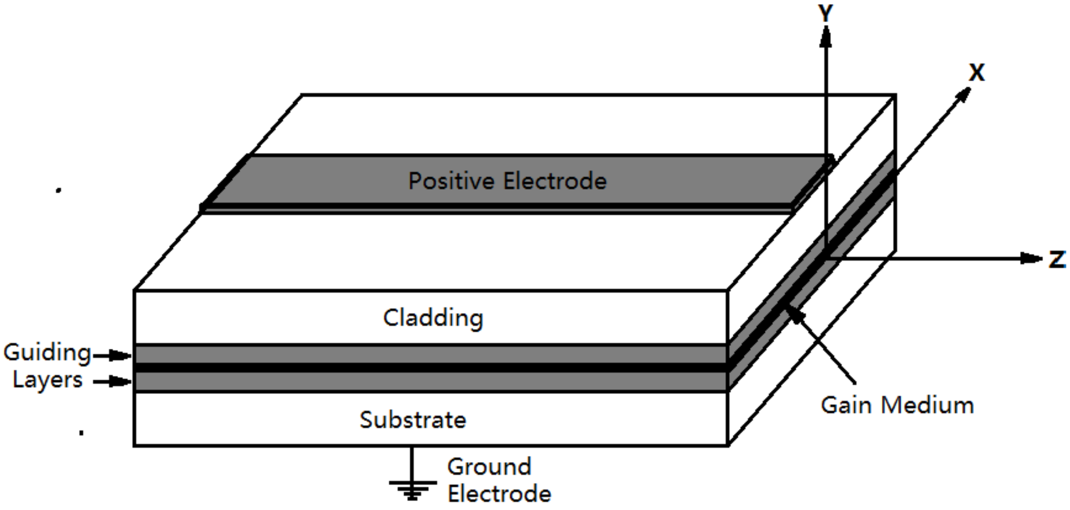


Figure 1.1: A semiconductor laser diode with the p-n junction structure.

1.2 Non-imaging optics

Non-imaging optics (also called anidolic optics) [2] is the branch of optics concerned with the optimal transfer of light radiation between a source and a target. Unlike traditional imaging optics, the techniques involved do not attempt to form an image of the source; instead they intend to form an optimized optical system for optical radiative transfer from a source to a target is desired.

The two design examples that non-imaging optics solves better than imaging optics are solar energy concentration and illumination. Especially, the illumination part is the core concept in this paper. The illumination part is controlling the distribution of light, typically so it is "evenly" spread over some areas and completely blocked from other areas. Typical variables to be optimized at the target include the total radiant flux, the angular distribution of optical radiation, and the spatial distribution of optical radiation. These variables on the target side of the optical system often must be optimized while simultaneously considering the collection efficiency of the optical system at the source.

Examples of non-imaging optical devices include optical light guides, non-imaging reflectors, non-imaging lenses or a combination of these devices. Common applications of non-imaging optics include many areas of illumination engineering (lighting). Examples of modern implementations of non-imaging optical designs include automotive headlamps, LCD backlights, illuminated instrument panels, fiber optic illumination devices, and LED Lights.

1.3 Research Objectives

The laser diode example shown above is the basic edge-emitting semiconductor laser, only one of three kinds of laser diodes. The other two includes double hetero-structure laser diode and vertical cavity surface emitting laser diode (VCSEL), as shown in Figure 1.2. Though different from each other in the structure, these laser diodes work on the same principle, functioning as an optical oscillator by stimulating a chain reaction of photon emission inside a tiny chamber. Thus they produce the same beam characteristics and can share the same beam shaping methods. Throughout this thesis, edge-emitting laser diodes will be referred to as an example by the author to demonstrate how to design and analyze the beam shaping systems involved.

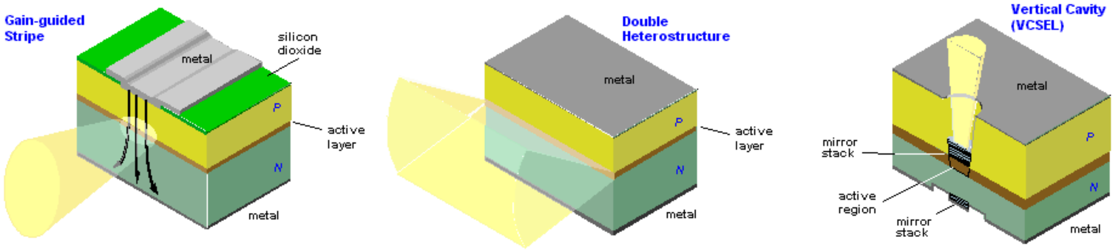


Figure 1.2: Three different structures of laser diodes [7]

As mentioned earlier, collimation, uniform intensity and pre-determined beam shape are the three main features in the regions of laser beam shaping applications. There are many papers presenting either to collimate the beam, to make it circular or rectangular, or to bring the irradiance profile to a uniform irradiance. And most of them use the traditional Galilean or Keplerian beam shaping system. These two kinds of refractive beam shaping systems are constructed by dual aspheric lens [54-55] based on energy conservation principle, constancy of optical path length, and Snell's law. Specifically, the two different off-axis mirror system and two different lens system are developed to circularize, collimate, and expand the elliptical beam of edge-emitting semiconductor laser based LIDAR systems [3].

Traditional beam shaping optics design techniques usually utilize rotational or translational symmetries, where the calculated cross section curve is swept around or along its symmetry axis to generate the 3-D geometry shape. However, the need for beam shaping optics that can distribute light beam in a non-rotational or non-translational manner has strongly increased. This leads to the concept of freeform optical surfaces, which can provide much more controlling freedom. The rapidly advancing manufacture technologies of freeform surfaces stimulate the development of freeform beam shaping elements [8]. The freeform surfaces design method has advantages with high degrees of design freedom that can be used to generate simpler designs with better performance and produce some special illuminations [9]. There are many kinds of freeform surfaces design methods in nonimaging optics: multiparameter optimization [10-15], the "Tailored Freeform Surface Method" [16-24], and the simultaneous multiple surface (SMS) method [25-27].

Inspired by these works, this thesis will design optical systems to transform a collimated elliptical or circular Gaussian laser beam into a uniform rectangular one with/without changing the wavefront. And also this thesis will realize the large area uniform illumination by using 9 lens units. The equations specific to this system are derived analytically based on Snell's Law and Energy Conservation. Then *Matlab* is used to solve these equations. *Matlab* codes are written to calculate and determine the parameters of the optical components. After getting the discrete data of the freeform surface by calculation, *Rhinoceros* are applied for the 3D molding. And then the *Tracepro* are applied to simulate the illumination system. The author expects that this beam-shaping system can be used and also easily modified, and the analysis methods developed in this thesis would be a valuable resource for the scientific community interested in beam shaping systems.

CHAPTER 2

DEVELOPED TECHNIQUES AND TOOLS

2.1 Runge-kutta method

In mathematics, an ordinary differential equation (ODE) is an equation containing a function of one independent variable and its derivatives. The term "ordinary" is used in contrast with the term partial differential equation which may be with respect to more than one independent variable.

Generally, there are two ways to solve the ODE, one is Euler method, and the other is Runge-kutta method. However, the latter is more widely used in solving ODE. There are several reasons that Euler's method is not recommended for practical use, among them, (i) the method is not very accurate when compared to other methods run at the equivalent step size, and (ii) neither is it very stable.

In numerical analysis, the Runge-Kutta methods are an important family of implicit and explicit iterative methods, which are used in temporal discretization for the approximation of solutions of ordinary differential equations. These techniques were developed around 1900 by the German mathematicians C. Runge and M. W. Kutta.

The formula for the fourth order Runge-Kutta method (RK4) is given below. Consider the problem [28]:

$$\begin{cases} y' = f(t, y) \\ y(t_0) = \alpha \end{cases} \tag{2.1}$$

Define h to be the time step size and $t_i = t_0 + ih$. Then the following formula:

$$\begin{cases} w_0 = \alpha \\ k_1 = hf(t_i, w_i) \\ k_2 = hf(t_i + \frac{h}{2}, w_i + \frac{k_1}{2}) \\ k_3 = hf(t_i + \frac{h}{2}, w_i + \frac{k_2}{2}) \\ k_4 = hf(t_i + h, w_i + k_3) \\ w_{i+1} = w_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \end{cases} \tag{2.2}$$

Computes an approximate solution, that is $w_i \approx y(t_i)$.

2.2 First-order quasi-linear partial differential equations (PDEs)

If a PDE can be expressed as follows, then it can be called the quasi-linear PDE:

$$\alpha(x, y, u(x, y))\frac{\partial u(x, y)}{\partial x} + b(x, u, u(x, y))\frac{\partial u(x, y)}{\partial y} = c(x, y, u(x, y)) \tag{2.3}$$

The general solution to this kind of PDE is Finite Difference Method (FDM) and Finite Element Method (FEM). However, the Runge-kutta method can be used to solve the first-order quasi-linear PDE when the boundary conditions are certain. Especially, the fourth order Runge-kutta method is more widely applied to this kind of quasi-linear PDEs.

2.3 Optical path length (OPL)

In optics, optical path length (OPL) or optical distance is the product of the geometric length of the path that light follows through the system, multiplied by the index of refraction of the medium through which it propagates. A difference in optical path length between two paths is often called the optical path difference (OPD). Optical path length is important because it determines the phase of the light and governs interference and diffraction of light as it propagates.

Optical path difference results in phase shift between two previously coherent sources when passed through different mediums. For example a wave passed through glass will appear to travel a greater distance than an identical wave in air. This is because the light beam in the glass will have travelled a greater number of wavelengths due to the higher refractive index of the glass.

The OPD can be calculated from the following equation:

$$OPD = d_1 n_1 - d_2 n_2 \tag{2.4}$$

where d_1 and d_2 are the distances of the ray passing through medium 1 or 2, n_1 is the greater refractive index (e.g., glass) and n_2 is the smaller refractive index (e.g., air).

In a medium of constant refractive index, n, the OPL for a path of physical length d is just

$$OPL = nd \tag{2.5}$$

If the refractive index varies along the path, the OPL is given by

$$OPL = \int_L n(l)\,dl \tag{2.6}$$

where $n(l)$ is the local refractive index as a function of distance, l, along the path L. Fermat's principle states that the path that light takes between two points is the path that has the minimum optical path length [29].

2.4 Fermat's principle

In optics, Fermat's principle or the principle of least time is the principle that the path taken between two points by a ray of light is the path that can be traversed in the least time. This principle is sometimes taken as the definition of a ray of light [30]. However, this version of the principle is not general; a more modern statement of the principle is that rays of light traverse the path of stationary optical length with respect to variations of the path [31]. In other words, a ray of light prefers the path such that there are other paths, arbitrarily nearby on either side, along which the ray would take almost exactly the same time to traverse [32].

Fermat's principle can be used to describe the properties of light rays reflected off mirrors, refracted through different media, or undergoing total internal reflection. It follows mathematically from Huygens ' Principle (at the limit of small wavelength). Fermat's text *Analyse des réfractions* exploits the technique of adequality to derive Snell's law of refraction [33] and the law of reflection.

2.5 Vector Snell's Law

In Figure 2.1 we have an interface between two materials with different indexes of refraction n_1 and n_2. n_1 is the index of refraction of the material you come from, and n_2 of the material you go to (in case of refraction).

The direction vector of the incident ray (= incoming ray) is $\vec{i}$, and we assume this vector is normalized. The direction vectors of the reflected and transmitted rays are $\vec{r}$ and $\vec{t}$. These vectors will be normalized as well. We also have the normal vector $\vec{N}$, orthogonal to the interface and points towards the first material n_1.

The direction vectors of the rays can be split in components orthogonal and parallel to the interface. We call these the normal component $\vec{v_\perp}$ and the tangent component $\vec{v_{||}}$ of a vector $\vec{v}$.

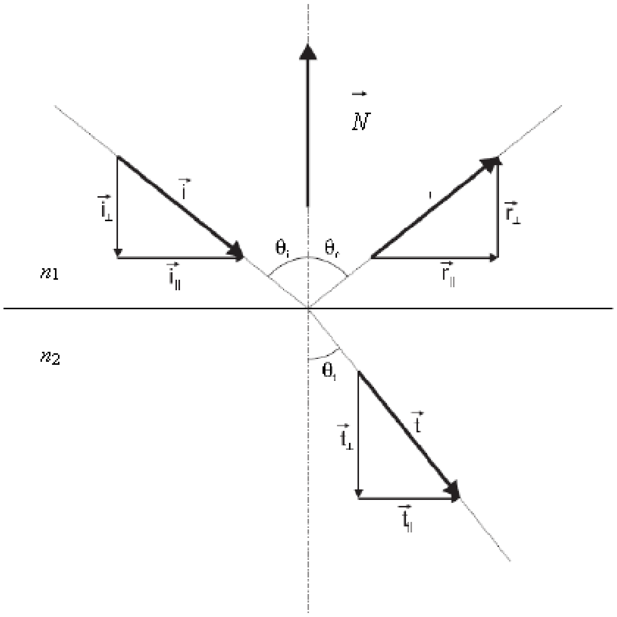


Figure 2.1: Situation for refraction and reflection [34]

The calculation of the refracted ray starts with Snell's law [35] which tells that the products of the index of refraction and sines of the angles must be equal:

$$n_1 \sin\theta_i = n_2 \sin\theta_t \tag{2.7}$$

Then we should try to find a formula for $\vec{t}$. The first thing is to split it up into tangent and normal parts:

$$\vec{t} = \vec{t_{\parallel}} + \overrightarrow{t_{\perp}} \tag{2.8}$$

According to the formula (2.7), the norms of the tangent parts happen to be equal to sines. Hence:

$$\left|\vec{t_{\parallel}}\right| = \frac{\eta_1}{\eta_2}\left|\vec{i_{\parallel}}\right| \tag{2.9}$$

Since $\vec{t_{\parallel}}$ and $\vec{i_{\parallel}}$ are in the same direction, this becomes:

$$\vec{t_{\parallel}} = \frac{\eta_1}{\eta_2}\vec{i_{\parallel}} = \frac{\eta_1}{\eta_2}\left[\vec{i} - \cos\theta_i \overrightarrow{N}\right] \tag{2.10}$$

The other one can be calculated by using the Pythagoras theorem:

$$\overrightarrow{t_{\perp}} = \text{-}\sqrt{1\text{-}\left|\vec{t_{\parallel}}\right|^2}\,\overrightarrow{N} \tag{2.11}$$

Then substitute (2.10) and (2.11) in (2.8) to get the refracted direction vector:

$$\vec{t} = \frac{n_1}{n_2}\vec{i} - \left(\frac{n_1}{n_2}\cos\theta_i + \sqrt{1 - \left|\vec{t_{\parallel}}\right|^2}\right)\overrightarrow{N} \tag{2.12}$$

It is easy to know that the vector $\vec{t_{\parallel}}$ equals to sinθ_t. Finally we get:

$$\vec{t} = \frac{n_1}{n_2}\vec{i} - \left(\frac{n_1}{n_2}\cos\theta_i + \sqrt{1 - \sin^2\theta_t}\right)\overrightarrow{N} \tag{2.13}$$

By Snell’s law, (2.13) can be expressed:

$$\sin^2\theta_t = \left(\frac{n_1}{n_2}\right)^2 \sin^2\theta_i = \left(\frac{n_1}{n_2}\right)^2 (1 - \cos^2\theta_i) \tag{2.14}$$

The last two equations will be used to calculate the refracted direction vector.

2.6 Non-uniform rational basis spline

Non-uniform rational basis spline (NURBS) is a mathematical model commonly used in computer graphics for generating and representing curves and surfaces. It offers great flexibility and precision for handling both analytic (surfaces defined by common mathematical formulae) and modeled shapes.

NURBS are commonly used in computer-aided design (CAD), computer-aided manufacturing (CAM), and computer-aided engineering (CAE) and are part of numerous industry wide standards, such as IGES, STEP, ACIS, and PHIGS. NURBS tools which are found in various 3D modelling and animation software packages.

They can be efficiently handled by computer programs and yet allow for easy human interaction. NURBS surfaces are functions of two parameters mapping to a surface in three-dimensional space. The shape of the surface is determined by control points. NURBS surfaces can represent simple geometrical shapes in a compact form. T-splines and subdivision surfaces are more suitable for complex organic shapes because they reduce the number of control points twofold in comparison with the NURBS surfaces.

2.6.1 NURBS CURVE

In general, editing NURBS curves and surfaces is highly intuitive and predictable. Control points are always either connected directly to the curve/surface or act as if they were connected by a rubber band. Depending on the type of user interface, editing can be realized via an element's control points, which are most obvious and common for Bézier curves, or via higher level tools such as spline modeling or hierarchical editing [36]. A Bézier curve is a parametric curve frequently used in computer graphics and related fields.

The general form of a NURBS curve is [36-38]:

$$r(u) = \frac{\sum_{i=0}^{n} w_i N_{k,i}(u) C_i}{\sum_{i=0}^{n} w_i N_{k,i}(u)} = \sum_{i=0}^{n} N^{*}{}_{k,i}(u) C_i \tag{2.15}$$

Here,

$$N^{*}{}_{k,i}(u) = \frac{w_i N_{k,i}(u)}{\sum_{i=0}^{n} w_i N_{k,i}(u)} \tag{2.16}$$

U={u_0,…,u_m} is a nondecreasing sequence of real numbers, i.e., u_i≤u_i+1, i=0,…,m-1. The u_i are called knots, and U is the knot vector. The ith B-spline basis function of p-degree (order p+1), denoted by $N_{i,p}(u)$, is defined as

$$N_{i,0}(u) = \begin{cases} 1 & if\ u_i \le u \le u_{i+1} \\ 0 & otherwise \end{cases}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \tag{2.17}$$

In this, n is the number of control points C_i and w_i are the corresponding weights. The denominator is a normalizing factor that evaluates to one if all weights are one. The Figure 2.2 shows the NURBS curve, the control points are intersections of the lines.

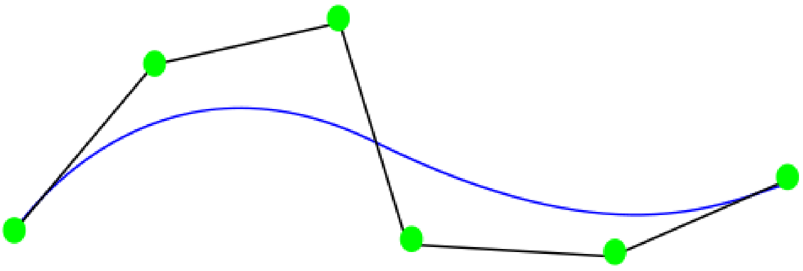


Figure 2.2: NURBS curve [36]

2.6.2 NURBS SURFACE

A B-spline surface is obtained by taking a bidirectional net of control points, two knot vectors, and the products of the univariate B-spline functions.

$$r(u)=\frac{\sum_{i=0}^{n}\sum_{j=0}^{m}w_{i,j}N_{p,i}(u)N_{q,j}(u)C_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m}w_{i,j}N_{p,i}(u)N_{q,j}(u)},\quad 0\le u,v\le 1 \tag{2.18}$$

With

$$\begin{aligned}U&=\{\underbrace{0,\ldots,0}_{p+1},u_{p+1},\ldots,u_{r-p-1},\underbrace{1,\ldots,1}_{p+1}\}\\ V&=\{\underbrace{0,\ldots,0}_{q+1},u_{q+1},\ldots,u_{s-q-1},\underbrace{1,\ldots,1}_{q+1}\}\end{aligned} \tag{2.19}$$

U has *r*+1 knots, and *V* has *s*+1.The Figure 2.3 shows the NURBS surface.

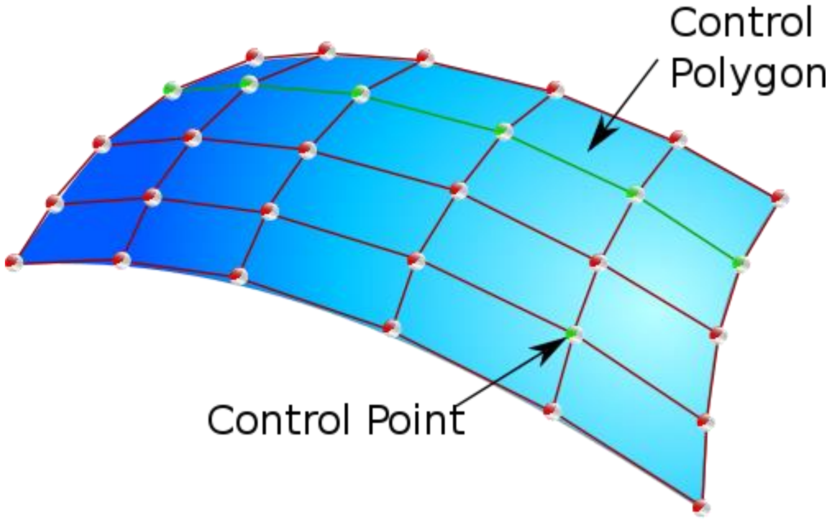


Figure 2.3: NURBS surface

2.7 Top-hat beam

In optics, a top-hat (or top-hat) beam such as a laser beam or electron beam has a near-uniform fluence (energy density) within a circular disk. It is typically formed by diffractive optical elements from a Gaussian beam. Top-hat beams are often used in industry, for example for laser drilling of holes in printed circuit boards. They are also used in very high power laser systems, which use chains of optical amplifiers to produce an intense beam. Top-hat beams are named for their resemblance to the shape of a top hat [39].

The irradiance distribution of the top-hat beam is:

$$I_{out,TH}(R) = \frac{circ(R / R_{\max})}{\pi R_{\max}^2} \tag{2.20}$$

where R_{max} labels the radius of the transverse size of the top-hat beam, and circ(R/R_{max}) is equal to unity for $0 \leq R \leq R_{max}$ and to zero otherwise [40].

Normally, the super-Gaussian, flattened-Gaussian, Fermi-Dirac, and super-Lorentzian are included in the top-hat distributions.

A general flattened output irradiance profile can be written as a product of its normalization constant, $I_0(\sigma, R_0)$, which is determined by requiring the input beam profile to be normalized according to energy conservation, and its functional dependence $f(\sigma,\ R/R_0)$; which depends on the shape parameter σ and on the ratio of the radial coordinate, R; to the beam width, R_0, of each distribution

$$I(\sigma, R/R_0) = I_0(\sigma, R_0) \times f(\sigma, R/R_0) \tag{2.21}$$

The width parameter, R_0, defines a length scale over which the profile decreases to some significant value for the specific profile distribution, such as, e^{-2} of its axial value. The shape parameter σ species the shape of the profile distribution, such as, the power of the radial coordinate of a super-Gaussian profile.

Explicitly, the Fermi-Dirac function is defined by:

$$I_{out,FD}(\beta, R/R_{FD}) = I_{0,FD}(\beta, R_{FD})\left\{1+\exp\left[\beta\left(\frac{R}{R_{FD}}-1\right)\right]\right\}^{-1} \tag{2.22}$$

where R_{FD} corresponds to R_0 in Eq. (2.21) and is the radius at which the output irradiance falls to half of its axial value. β is a dimensionless shape parameter for the Fermi-Dirac function. When β increases, the output profile approaches a more uniform distribution where there is a continuous variation of irradiance from zero to its maximum value on axis.

The super-Lorentzian function is defined by

$$I_{out,SL}(M, R/R_{SL}) = \frac{I_{0,SL}(M, R_{SL})}{\left[1+(R/R_{SL})^{M}\right]} \tag{2.23}$$

where R_{SL} is the radius at which the output irradiance falls to half of its axial value, and M is a dimensionless shape parameter for the super-Lorentzian function. When M increases, the output profile approaches a more uniform distribution where there is a continuous variation of irradiance from zero to its maximum value of $I_{0,SL}(M,R_{SL})$.

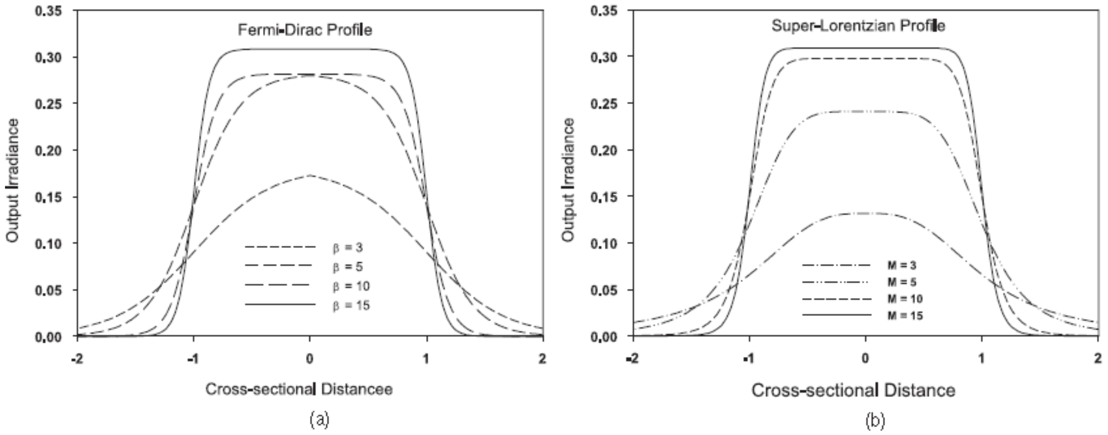


Figure 2.4: (a) Fermi-Dirac and (b) super-Lorentzian output irradiance for shape parameters β and M. All profiles are normalized such that the total energy contained within each irradiance

distribution over all space is equal to unity. The cross-sectional distance is expressed in the normalized units of the beam width.

The super-Gaussian function is defined by

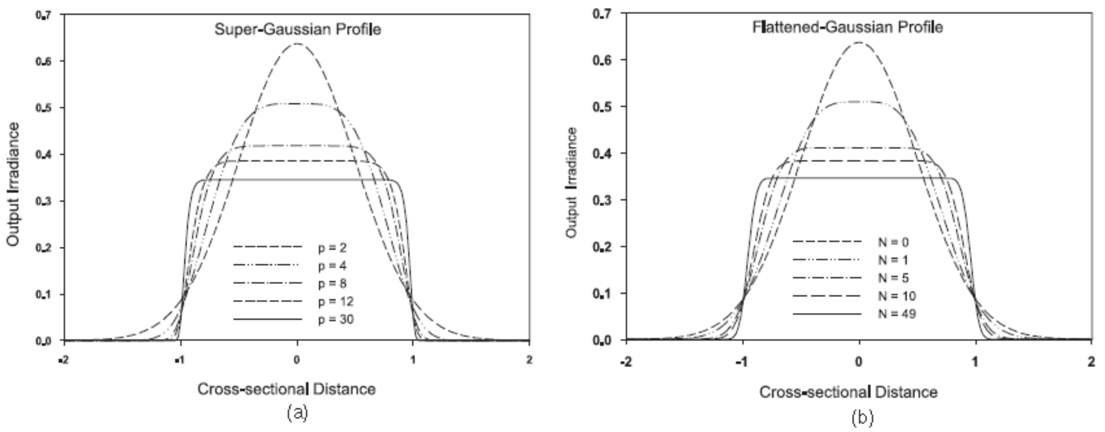
$$I_{out,SG}(p, R/R_{SG}) = I_{0,SG}(p, R_{SG})\exp\left[-2\left(\frac{R}{R_{SG}}\right)^{p}\right] \tag{2.24}$$

where R_{SG} is the radius at which the output irradiance falls to e^{-2} of its axial value, and p is a dimensionless shape parameter for the super-Gaussian function. When p increases, the output profile approaches a more uniform distribution where there is a continuous variation of irradiance from zero to its maximum value of $I_{0,SL}$ (M,R_{SG}).

The flattened-Gaussian function is defined by

$$I_{out,FG}(N, R/R_{FG}) = I_{0,FG}(N, R_{FG})\exp\left[-2(N+1)(R/R_{FG})^{2}\right]\sum_{m,n=0}^{N}\frac{\left[\left(N+1\right)\left(R/R_{FG}\right)^{2}\right]^{n+m}}{n!m!} \tag{2.25}$$

where R_{FG} is the radius at which the output irradiance falls to $\Gamma(N + 1,N + 1)^2 / \Gamma(\mathrm{N} + 1)^2$ of its axial value. N is a dimensionless shape parameter for the flattened-Gaussian function. When p increases, the output profile approaches a more uniform distribution where there is a continuous variation of irradiance from zero to its maximum value of $I_{0,FG}$ (N,R_{FG}).



(a)

(b)

Figure 2.5: (a) super-Gaussian and (b) flattened-Gaussian output irradiance for shape parameters *p* and *N*. All profiles are normalized such that the total energy contained within each irradiance distribution over all space is equal to unity.

2.8 Monte-Carlo method

Monte Carlo methods (or Monte Carlo experiments) are a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results; typically one runs simulations many times over in order to obtain the distribution of an unknown probabilistic entity. The name comes from the resemblance of the technique to the act of playing and recording your results in a real gambling casino. They are often used in physical and mathematical problems and are most useful when it is difficult or impossible to obtain a closed-form expression, or infeasible to apply a deterministic algorithm. Monte Carlo methods are mainly used in three distinct problems classes: optimization, numerical integration and generation of draws from a probability distribution [41].

In physics-related problems, Monte Carlo methods are quite useful for simulating systems with many coupled degrees of freedom, such as fluids, disordered materials, strongly coupled solids, and cellular structures (see cellular Potts model). Other examples include modeling phenomena with significant uncertainty in inputs, such as the calculation of risk in business; and, in math, evaluation of multidimensional definite integrals with complicated boundary conditions. In application to space and oil exploration problems, Monte Carlo-based predictions of failures, cost overruns and schedule overruns are routinely better than human intuition or alternative "soft" methods [42].

The modern version of the Monte Carlo method was invented in the late 1940s by Stanislaw Ulam, while he was working on nuclear weapons projects at the Los Alamos National Laboratory. It was named by Nicholas Metropolis, after the Monte Carlo Casino, where Ulam's uncle often gambled [43]. Immediately after Ulam's breakthrough, John von Neumann understood its importance and programmed the ENIAC computer to carry out Monte Carlo calculations.

CHAPTER 3

METHODOLOGY

3.1 One freeform lens

Based on the Snell's law and the energy conservation, we can deduce from the characteristics of the source and desired illumination, to get a set of first-order partial differential equations [44-48]. These two scientific laws are the basic principle of the freeform lens design. The Snell's law tells how the rays are refracted and reflected, which means the redistribution of the luminous energy. With regard to the energy conservation, the relationship between input energy and output one can be defined, and, furthermore, the input-to-output irradiance mapping can be obtained based on this law using the variable separation method [49]. Then the collimated light freeform lens design method will be built in the Cartesian coordinate system.

Assume $\boldsymbol{I}$ is the vector of the collimated incident light at point p, while $\boldsymbol{O}$ is the vector of the refracted light by the freeform lens from point p to point t, and the normal vector at point p of the freeform lens is $\boldsymbol{N}$. The Cartesian coordinate system is built as shown in Figure 3.1, z is the optical axis.

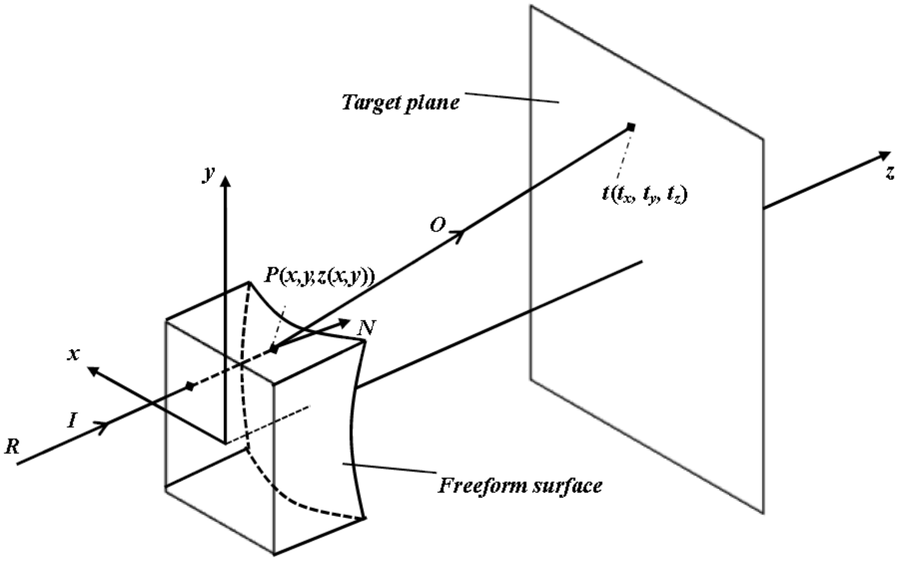


Figure 3.1: Schematic representation of the one freeform lens

The coordinates on *p* and *t* are (x,y,z(x,y)) and (t_x,t_y,t_z), respectively. ***N*** is the normal unit vector to the freeform surface. ***I*** and ***O*** are the unit vectors of the incident and transmitted beams, respectively. Using the concept of differential calculus of multivariate, they can be expressed as:

$$\overrightarrow{N} = \frac{1}{\sqrt{z_x^2 + z_y^2 + 1}}(-z_x, -z_y, 1) \tag{3.1}$$

where, z_x and z_y are the first-order partial derivatives of z with respect to x and y, respectively.

$$\overrightarrow{O} = \frac{1}{\sqrt{(t_x - x)^2 + (t_y - y)^2 + (t_z - z)^2}}(t_x - x, t_y - y, t_z - z) \tag{3.2}$$

After using Snell's law at point *p*, we have the equations as follows:

$$n_o\overrightarrow{O} - n_i\overrightarrow{I} = \overrightarrow{N}\sqrt{n_o^2 + n_i^2 - 2n_o n_i(\overrightarrow{O}\cdot\overrightarrow{I})} \tag{3.3}$$

Here, n_o and n_i indicate the refractive index of the incidence and emergence medium. Further, we can get:

$$n_o O_x - n_i I_x = N_x\sqrt{n_o^2 + n_i^2 - 2n_o n_i(\overrightarrow{O}\cdot\overrightarrow{I})} \tag{3.4}$$

$$n_o O_y - n_i I_y = N_y \sqrt{n_o^2 + n_i^2 - 2 n_o n_i (\vec{O} \cdot \vec{I})} \tag{3.5}$$

$$n_o O_z - n_i I_z = N_z \sqrt{n_o^2 + n_i^2 - 2 n_o n_i (\vec{O} \cdot \vec{I})} \tag{3.6}$$

Here, I_x, I_y, I_z are the three components of ***I***. O_x, O_y, O_z are the three components of ***O***. N_x, N_y, N_z are the three components of ***N***. Moreover, I=(0,0,1), and t_z is a constant, which is defined as the distance between source plane **A** and target plane **C**. Then we can get:

$$z_x = \frac{n_o(x - t_x)}{n_o(t_z - z) - n_i \sqrt{(t_x - x)^2 + (t_y - y)^2 + (t_z - z_0)^2}} \tag{3.7}$$

$$z_y = \frac{n_o(y - t_y)}{n_o(t_z - z) - n_i \sqrt{(t_x - x)^2 + (t_y - y)^2 + (t_z - z_0)^2}} \tag{3.8}$$

Equations (3.7) and (3.8) are the first-order partial differential equations about z_x and z_y, which can be solve by using the Runge-kutta method.

According to light transmission energy conservation condition, the output of source is equal to the flux incident in the target plane.

$$I_{in}(x, y) dxdy = I_{out}(t_x, t_y) dt_x dt_y \tag{3.9}$$

Therefore, the relationship between x, y and t_x, t_y can be acquired from equation (3.9).

3.2 Two freeform lens

3.2.1 Based on Partial Differential Equations (PDEs)

According to the previous section, we know how to design freeform lens based on the Snell's law and energy conservation principle. And the two freeform lens design is similar to the one freeform lens design. One more principle is needed, which is the constancy of optical path length.

The two freeform refractive surface optical system is shown schematically in Figure 3.2.

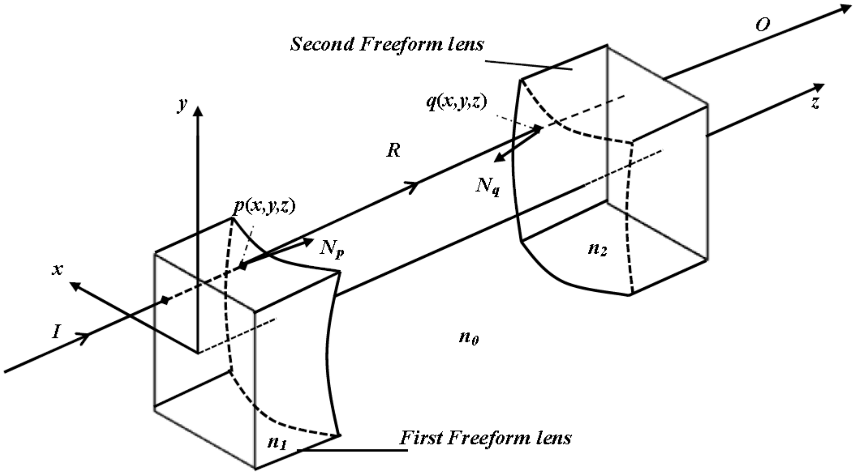


Figure 3.2: Schematic representation of the two freeform lens

Assume ***I*** is the unit vector of the collimated incident light at point *p*, ***R*** is the unit vector of the ray through *p* and *q*, ***O*** is the unit vector of the refracted light by the second freeform lens from point *q* to point *t*, and ***O*** is parallel to the optical axis. n_1, n_0 and n_2 are set as the refractive indices of the mediums on the right side of the first freeform surface, between the two freeform surfaces and on the left side of the second freeform surface, respectively. Similar to the one freeform lens design, ***I***, ***R***, ***O***, ***N*** and the relationships among them can be expressed as:

$$\vec{I} = (0,0,1) \tag{3.10}$$

$$\vec{O} = (0,0,1) \tag{3.11}$$

$$\vec{R} = \frac{1}{\sqrt{(x_q - x_p)^2 + (y_q - y_p)^2 + (z_q - z_p)^2}} (x_q - x_p, y_q - y_p, z_q - z_p) \tag{3.12}$$

$$\overrightarrow{N_p} = \frac{1}{\sqrt{z_{px}^2 + z_{py}^2 + 1}} (-z_{px}, -z_{py}, 1) \tag{3.13}$$

$$n_0 \vec{R} - n_1 \vec{I} = \overrightarrow{N_p} \sqrt{n_0^2 + n_1^2 - 2n_0 n_1 (\vec{R} \cdot \vec{I})} \tag{3.14}$$

Where, x_p, y_p, z_p and x_q, y_q, z_q are the Cartesian coordinates of points p and q on the first and second freeform surfaces, respectively. And $\boldsymbol{N_p}$ is the normal unit vector at point p of the first freeform lens. Then substitute (3.10), (3.11), (3.12) and (3.13) in (3.14) to get the partial differential equations:

$$z_{px} = -\frac{n_0(x_q - x_p)}{n_0(z_q - z_p) - n_1\sqrt{(x_q - x_p)^2 + (y_q - y_p)^2 + (z_q - z_p)^2}} \tag{3.15}$$

$$z_{py} = -\frac{n_0(y_q - y_p)}{n_0(z_q - z_p) - n_1\sqrt{(x_q - x_p)^2 + (y_q - y_p)^2 + (z_q - z_p)^2}} \tag{3.16}$$

The relationship between x_p, y_p, z_p and x_q, y_q, z_q can be calculated based on the constancy of optical path length.

$$n_1 z_p + n_0 \left[\left(x_q - x_p \right)^2 + \left(y_q - y_p \right)^2 + \left(z_q - z_p \right)^2 \right]^{\frac{1}{2}} + n_2 (z_t - z_q) = Const. \tag{3.17}$$

Because $\boldsymbol{I}$ and $\boldsymbol{O}$ are parallel to the optical axis, then x_p=x_s, y_p=y_s, x_q=x_t, y_q=y_t. z_s and z_t are constants. On the other hand, the relationship between x_s, y_s and x_t, y_t can be calculated based on the energy conservation.

$$I_{in}(x_s, y_s) dx_s dy_s = I_{out}(x_t, y_t) dx_t dy_t \tag{3.18}$$

3.2.2 Based on separated variables mapping

This method is to produce desired irradiance distribution whilst forming a prescribed wavefront from a given input beam. To simplify the computation, the input-to-output irradiance mapping is firstly obtained based on Energy conservation using the separated variables method [50]. Thus, both the input and output irradiance distributions must fulfill the restriction that they can be factorized in two orthogonal transverse coordinates, which is still suitable for many laser beam shaping applications. Then, the two freeform optical surfaces are generated simultaneously and point by point corresponding to the input and output ray sequences defined by the first step. The method is fast and can be easy understood by optical engineers since it avoid solving the underlying Monge–Ampère equation [49].

Consider a two-freeform-refractive-surface optical system shown schematically in Figure 3.3 (one or both of the two optical surfaces could also be reflective). The input and output beam are supposed as propagating toward the positive z direction and their wavefronts can be represented as position vectors $\boldsymbol{S} = (x_s,y_s,z_s)$ and $\boldsymbol{T} = (x_t,y_t,z_t)$, respectively. The points on the two freeform surfaces are denoted by $\boldsymbol{P} = (x_p,y_p,z_p)$ and $\boldsymbol{Q} = (x_q,y_q,z_q)$, respectively. n_1, n_0 and n_2 are set as the refractive indices of the mediums on the right side of the first freeform surface, between the two freeform surfaces and on the left side of the second freeform surface, respectively.

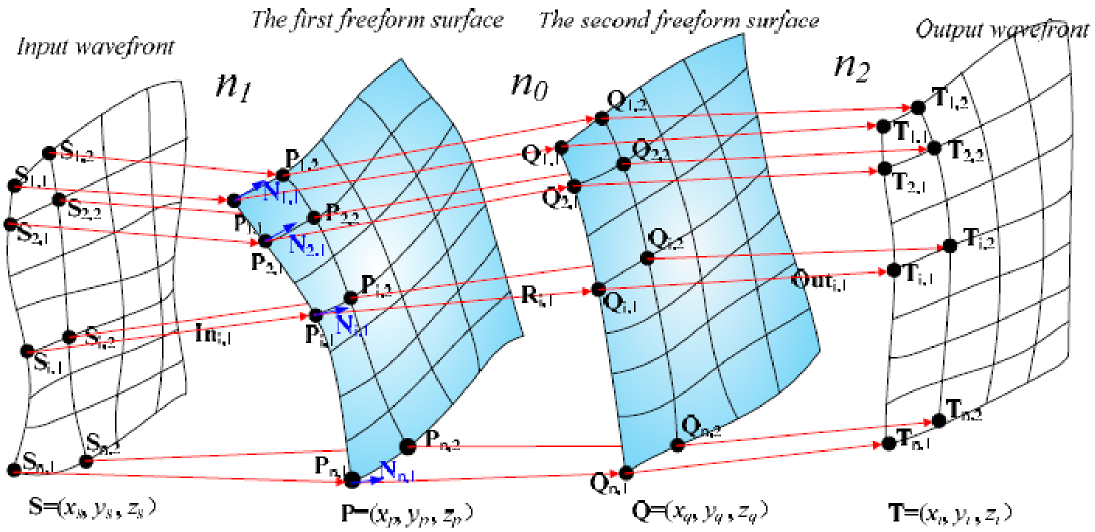


Figure 3.3: Geometrical construction of the double freeform optical surfaces for achieving a specified input-output ray mapping [49].

Let I_{in} and I_{out} denote the prescribed irradiance distributions over the planes perpendicular to the z-axis near the input and output wavefronts, respectively. Their relationship can be described by Energy conservation written as Eq. (3.19):

$$I_{in}(x_s, y_s)dx_s dy_s = I_{out}(x_t, y_t)dx_t dy_t \qquad (3.19)$$

Assume that I_{in} and I_{out} can be separated into a product of two one-dimensional irradiance distributions: $I_{\text{in}}(x_s,y_s) = I_{\text{in},x}(x_s)\, I_{\text{in},y}(y_s)$ and $I_{\text{out}}(x_t,y_t) = I_{\text{out},x}(x_t)\, I_{\text{out},y}(y_t)$. Thus, both (x_s, y_s) and (x_t, y_t) can be numerically specified based on “source-to-target” [50] or “target-to-source” [51,52] variable separation mapping strategies if one of them is predefined. Take “source-to-target variable separation mapping strategy for example, if (x_s, y_s) are equidistantly divided into $n \times m$ rectangular grids: $x_s = x_{s,j}$ and $y_s = y_{s,i}$, $i = 1,2,\ldots n$, $j = 1,2,\ldots m$, then $x_{t,j}$ and $y_{t,i}$ can be calculated by satisfying Eq. (3.20) and Eq. (3.21), respectively.

$$\int_{x_{s,1}}^{x_{s,j}} I_{in,x}(x_s)dx_s \int_{y_{s,1}}^{y_{s,n}} I_{in,y}(y_s)dy_s = \int_{x_{t,1}}^{x_{t,j}} I_{out,x}(x_t)dx_t \int_{y_{t,1}}^{y_{t,n}} I_{out,y}(y_t)dy_t \qquad (3.20)$$

$$\int_{x_{s,1}}^{x_{s,m}} I_{in,x}(x_s)dx_s \int_{y_{s,1}}^{y_{s,i}} I_{in,y}(y_s)dy_s = \int_{x_{t,1}}^{x_{t,m}} I_{out,x}(x_t)dx_t \int_{y_{t,1}}^{y_{t,i}} I_{out,y}(y_t)dy_t \tag{3.21}$$

Then, the input and output unit ray vectors $\mathrm{In}_{i,j}$ and $\mathrm{Out}_{i,j}$ can be obtained as the unit normal vectors at $\boldsymbol{S}_{i,j}(x_{s,j},y_{s,i},z_{s,i,j})$ and $\boldsymbol{Ti,j}(x_{s,j},y_{s,i},z_{s,i,j})$, respectively, as shown in Eq. (3.22) and Eq. (3.23):

$$\boldsymbol{In}_{i,j} = \left[-\left(\frac{\partial z_s}{\partial x}\right)_{x_{s,j},y_{s,i}}, -\left(\frac{\partial z_s}{\partial y}\right)_{x_{s,j},y_{s,i}}, 1 \right] \Bigg/ \sqrt{1+\left(\frac{\partial z_s}{\partial x}\right)_{x_{s,j},y_{s,i}}^2 + \left(\frac{\partial z_s}{\partial y}\right)_{x_{s,j},y_{s,i}}^2} \tag{3.22}$$

$$\boldsymbol{Out}_{i,j} = \left[-\left(\frac{\partial z_t}{\partial x}\right)_{x_{t,j},y_{t,i}}, -\left(\frac{\partial z_t}{\partial y}\right)_{x_{t,j},y_{t,i}}, 1 \right] \Bigg/ \sqrt{1+\left(\frac{\partial z_t}{\partial x}\right)_{x_{t,j},y_{t,i}}^2 + \left(\frac{\partial z_t}{\partial y}\right)_{x_{t,j},y_{t,i}}^2} \tag{3.23}$$

Thus, the input ray sequence is defined both by $\boldsymbol{In}_{i,j}$ and $\boldsymbol{S}_{i,j}$, and the output ray sequence is defined both by $\boldsymbol{Out}_{i,j}$ and $\boldsymbol{T}_{i,j}$, $i = 1,2,\dots n$, $j = 1,2,\dots m$.

The next step is to calculate the data points $\boldsymbol{P}_{i,\,j}$ and $\boldsymbol{Q}_{i,\,j}$ on the two freeform optical surfaces which are desired to transform the input ray sequence into the output ray sequence. From two starting points $\boldsymbol{P}_{1,1}$ and $\boldsymbol{Q}_{1,1}$ e.g. the central points of the first and second freeform surfaces, the normal vector $\boldsymbol{N}_{1,1}$ at $\boldsymbol{P}_{1,1}$ is firstly calculated using the vector form of Snell's law so that the ray emitted from $\boldsymbol{P}_{1,1}$ can reach $\boldsymbol{Q}_{1,1}$, as shown in Eq. (3.24):

$$\boldsymbol{N}_{1,1} = \left(n_0\boldsymbol{R}_{1,1} - n_1\boldsymbol{In}_{1,1}\right) \Big/ \sqrt{n_0^2 + n_1^2 + 2n_0n_1(\boldsymbol{R}_{1,1}\cdot\boldsymbol{In}_{1,1})} \tag{3.24}$$

wherein $\boldsymbol{R}_{1,1}$ denotes the unit vector of the ray through $\boldsymbol{P}_{1,1}$ and $\boldsymbol{Q}_{1,1}$. $\boldsymbol{P}_{2,1}$ can be obtained by intersecting its corresponding input ray to the tangent plane of $\boldsymbol{P}_{1,1}$ [53], which can be formulated as Eq. (3.25) and Eq. (3.26):

$$\left(\boldsymbol{P}_{2,1} - \boldsymbol{P}_{1,1}\right)\cdot \boldsymbol{N}_{1,1} = 0 \tag{3.25}$$

$$\left(\boldsymbol{P}_{2,1} - \boldsymbol{S}_{2,1}\right) \Big/ \left|\boldsymbol{P}_{2,1} - \boldsymbol{S}_{2,1}\right| = \boldsymbol{In}_{2,1} \tag{3.26}$$

Then, $\boldsymbol{Q}_{2,1}$ is obtained by equaling the OPL of the ray through $\boldsymbol{P}_{2,1}$ and itself to the OPL of the ray through $\boldsymbol{P}_{1,1}$ and $\boldsymbol{Q}_{1,1}$, as shown in Eq. (3.27) and Eq. (3.28):

$$n_1\left[\boldsymbol{S}_{2,1},\boldsymbol{P}_{2,1}\right]+n_0\left[\boldsymbol{P}_{2,1},\boldsymbol{Q}_{2,1}\right]+n_2\left[\boldsymbol{Q}_{2,1},\boldsymbol{T}_{2,1}\right]=n_1\left[\boldsymbol{S}_{1,1},\boldsymbol{P}_{1,1}\right]+n_0\left[\boldsymbol{P}_{1,1},\boldsymbol{Q}_{1,1}\right]+n_2\left[\boldsymbol{Q}_{1,1},\boldsymbol{T}_{1,1}\right] \quad (3.27)$$

$$\left(\boldsymbol{T}_{2,1}-\boldsymbol{Q}_{2,1}\right)\Big/\left|\boldsymbol{T}_{2,1}-\boldsymbol{Q}_{2,1}\right|=\boldsymbol{Out}_{2,1} \quad (3.28)$$

wherein [*X*,*Y*] represents the distance between two arbitrary points *X* and *Y*. Then, the required normal vector $\boldsymbol{N}_{2,1}$ at $\boldsymbol{P}_{2,1}$ can be computed so that the ray through $\boldsymbol{P}_{2,1}$ can be refracted to $\boldsymbol{Q}_{2,1}$. The entire starting curves on the two surfaces can be generated by repeating the above process. All the points of the second curve on the first freeform surface can be obtained on the tangent planes of the starting curve's points, written as Eq. (3.29) and Eq. (3.30):

$$\left(\boldsymbol{P}_{i,2}-\boldsymbol{P}_{i,1}\right)\cdot\boldsymbol{N}_{i,1}=0 \quad (3.29)$$

$$\left(\boldsymbol{P}_{i,2}-\boldsymbol{S}_{i,2}\right)\Big/\left|\boldsymbol{P}_{i,2}-\boldsymbol{S}_{i,2}\right|=\boldsymbol{In}_{i,2} \quad (3.30)$$

Then, points of the second curve on the second freeform surface can be computed by equaling their OPLs to that of the ray through $\boldsymbol{P}_{1,1}$ and $\boldsymbol{Q}_{1,1}$, expressed as Eq. (3.31) and Eq. (3.32):

$$n_1\left[\boldsymbol{S}_{i,2},\boldsymbol{P}_{i,2}\right]+n_0\left[\boldsymbol{P}_{i,2},\boldsymbol{Q}_{i,2}\right]+n_2\left[\boldsymbol{Q}_{i,2},\boldsymbol{T}_{i,2}\right]=n_1\left[\boldsymbol{S}_{i,1},\boldsymbol{P}_{i,1}\right]+n_0\left[\boldsymbol{P}_{i,1},\boldsymbol{Q}_{i,1}\right]+n_2\left[\boldsymbol{Q}_{i,1},\boldsymbol{T}_{i,1}\right] \quad (3.31)$$

$$\left(\boldsymbol{T}_{i,2}-\boldsymbol{Q}_{i,2}\right)\Big/\left|\boldsymbol{T}_{i,2}-\boldsymbol{Q}_{i,2}\right|=\boldsymbol{Out}_{i,2} \quad (3.32)$$

All the normal vectors at the points of the second curve on the first freeform surface can then be obtained. We can repeat this process to generate all the surface points and interpolate them to get the entire freeform surfaces [38].

CHAPTER 4

DESIGN EXAMPLES AND RESULTS

This chapter provides design examples to the systems discussed in chapter 3. The parameters of the design examples are calculated with *Matlab* codes developed. The examples will be discussed in two main parts: one freeform lens, and two freeform lens. The *Matlab* and *Tracepro* results are presented and discussed separately.

4.1 One freeform lens

Parameters for the design are listed in Table 4.1, which are the values of the basic known parameters of our design example.

Table 4.1 Parameters of Laser shaping system

Parameter	Value
Wavelength	870 nm
The glass used	BK7
The refractive index at the wavelength	1.509493

Second, we use the equations derived in chapter 3 to obtain the parameters of the freeform surface, which are realized in the software *Matlab*. In detail, the parameters are the three-dimensional Cartesian Coordinates of the data points on the surface. Then the surface can be reconstructed in the software *Rhinoceros* based on the idea of NURBS which is stated in chapter 2.

According to above design steps, four different schemes for the freeform lens with nonrotational symmetry are constructed.

4.1.1 Incident beam is circular Gaussian beam

The other parameters for this beam shaping system is shown in Table 4.2

Table 4.2 Parameters of one freeform lens system

W	4mm
L	4mm
Incident Beam radius	2mm
Incident beam waist w_0 (e^{-2} of beam profile)	1mm
D_x	6mm
D_y	6mm
z_0	500mm

W and L are the width and length of the freeform lens, respectively. Dx and Dy are the length respectively in x and y direction of the output beam in the target plane. z_0 is the distance between the first surface of the freeform lens and the target plane.

In the design, the freeform surface is reconstructed with 201×201 points based on NURBS. Ray tracing is implemented based on the Monte-Carlo method. About 1,000,000 rays are traced in this beam shaping system, the irradiance distribution of the output beam in the target plane is shown in Figure 4.1.

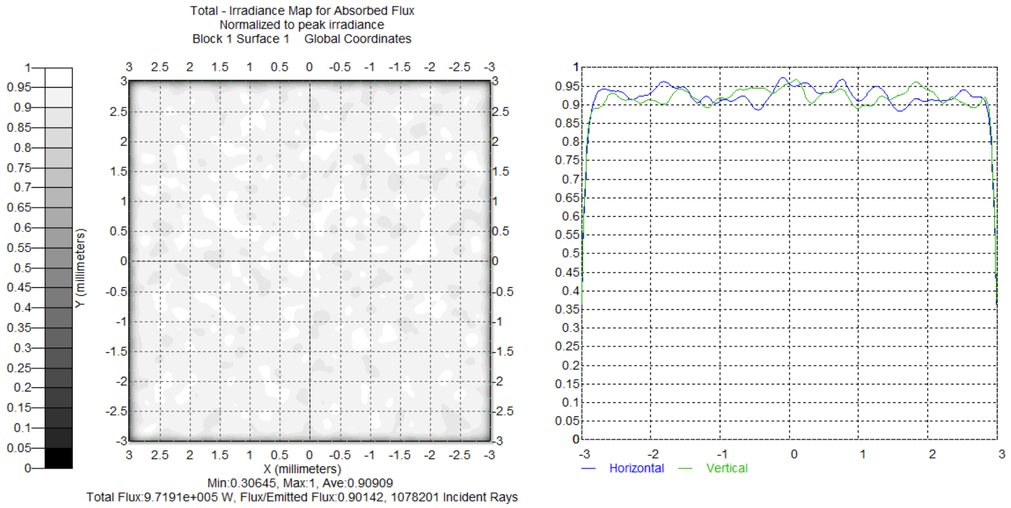


Figure 4.1: Irradiance distribution on the target plane.

From the ray tracing results, the ratio of the power on the desired region to the total power is about 90.9%.

When the source consists of 3 by 3 laser diode arrays, each lens array is composed of 9 lens, which is used to shape the corresponding laser diode. Figure 4.2 is the layout of the system.

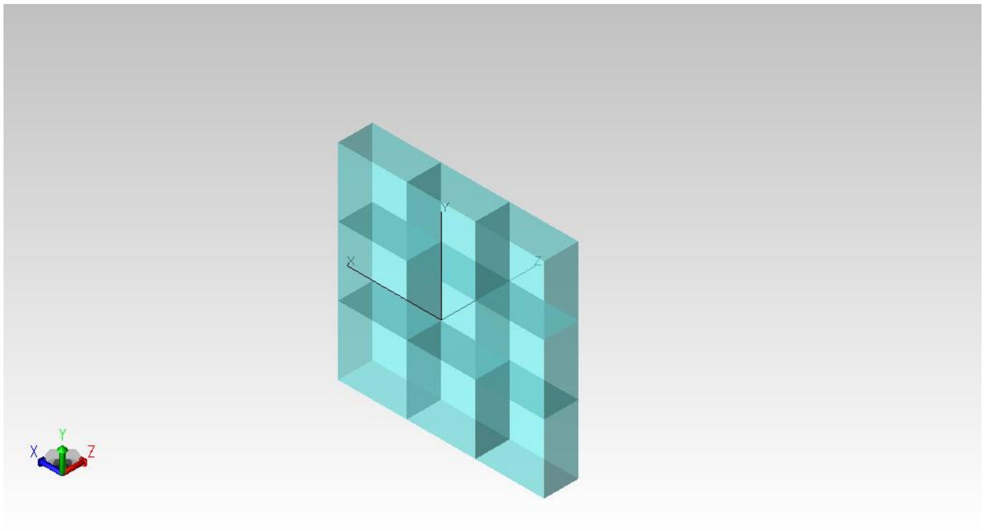


Figure 4.2: Layout of the lens array.

About 10,000 rays are traced in this beam shaping system, the irradiance distribution of the output beam in the target plane is shown in Figure 4.3.

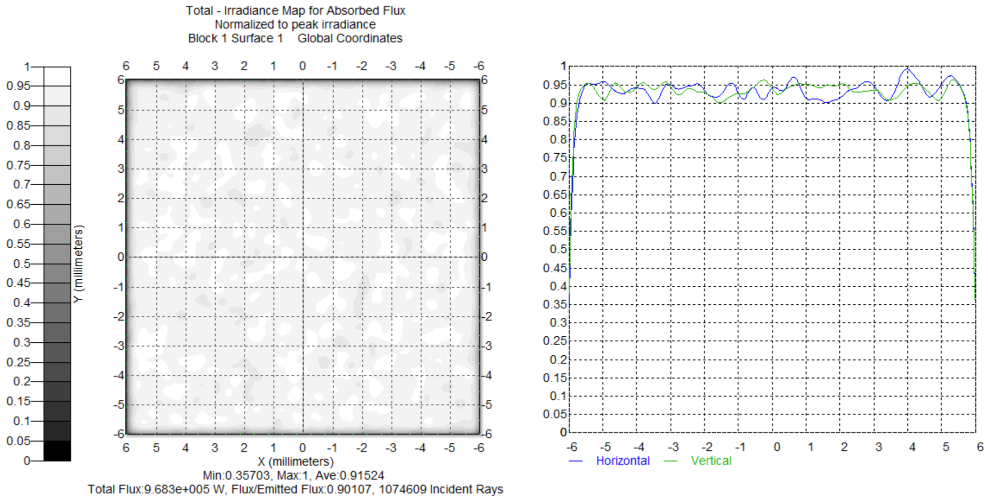


Figure 4.3: Irradiance distribution of the lens array on the target plane.

From the ray tracing results, the ratio of the power on the desired region to the total power is about 91.52%. According to this result, large area illumination can be realized by using the idea of lens array.

4.1.2 Incident beam is elliptical Gaussian beam

The other parameters for this beam shaping system is shown in Table 4.3

Table 4.3 Parameters of one freeform lens system

W	4mm
L	4mm
Incident Beam radius in x direction	2mm
Incident Beam radius in y direction	1.6mm
Incident beam perpendicular waist w_{0x}	1mm
Incident beam parallel waist w_{0y}	0.8mm
D_x	4mm
D_y	4mm
z_0	500mm

Also, W and L are the width and length of the freeform lens, respectively. *Dx* and *Dy* are the length respectively in x and y direction of the output beam in the target plane. z_0 is the distance between the first surface of the freeform lens and the target plane.

In the design, the freeform surface is reconstructed with 201×201 points based on NURBS. Ray tracing is implemented based on the Monte-Carlo method. About 1,000,000 rays are traced in this beam shaping system, the irradiance distribution of the output beam in the target plane is shown in Figure 4.4.

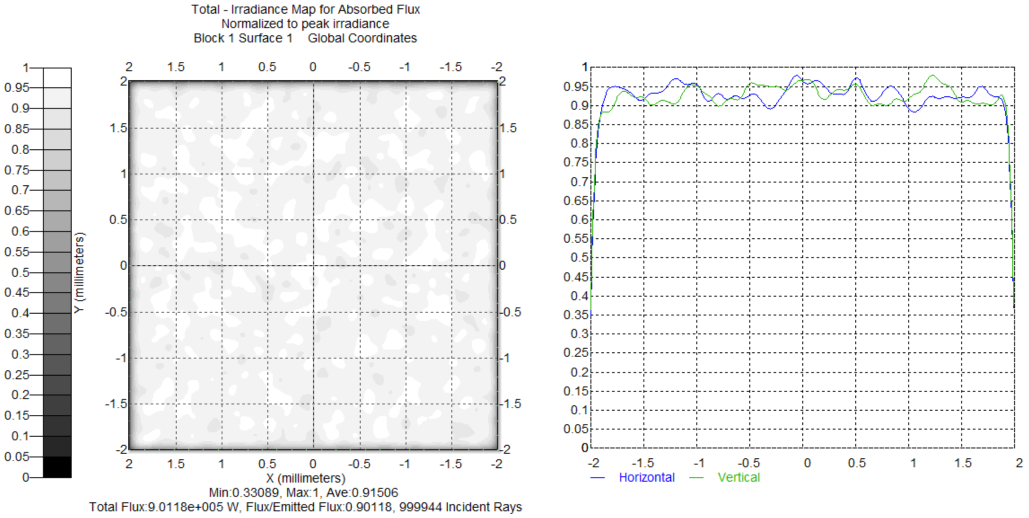


Figure 4.4: Irradiance distribution of the freeform lens on the target plane.

From the ray tracing results, the ratio of the power on the desired region to the total power is about 91.51%.

When the source consists of 3 by 3 laser diode arrays, each lens array is composed of 9 lens, which is used to shape the corresponding laser diode. Still, about 2,000,000 rays are traced in this beam shaping system, the irradiance distributions of the incident and output beam in the input and target plane are shown in Figure 4.5 and Figure 4.6, respectively.

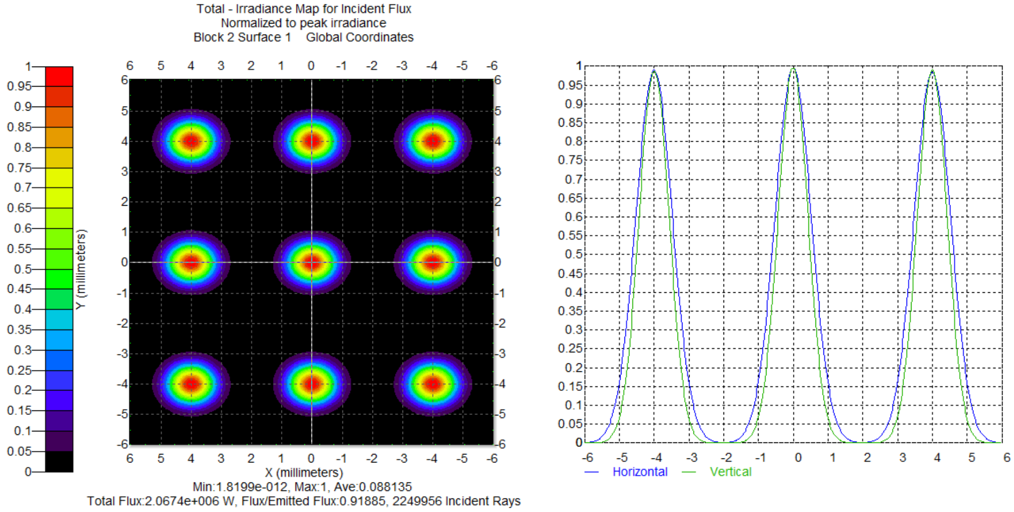


Figure 4.5: Irradiance distribution of the freeform lens on the input plane.

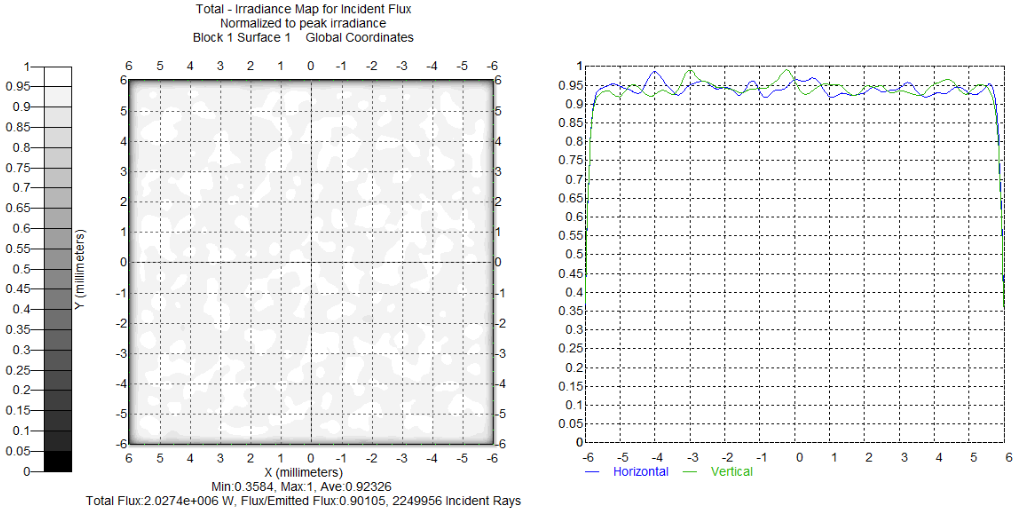


Figure 4.6: Irradiance distribution of the freeform lens on the target plane.

From the ray tracing results, the ratio of the power on the desired region to the total power is about 92.33%.

4.2 Two freeform lens

As mentioned in chapter 3, there are two different methods used in designing two freeform lens. Same parameters are used both in these two methods.

4.2.1 Incident beam is circular Gaussian beam

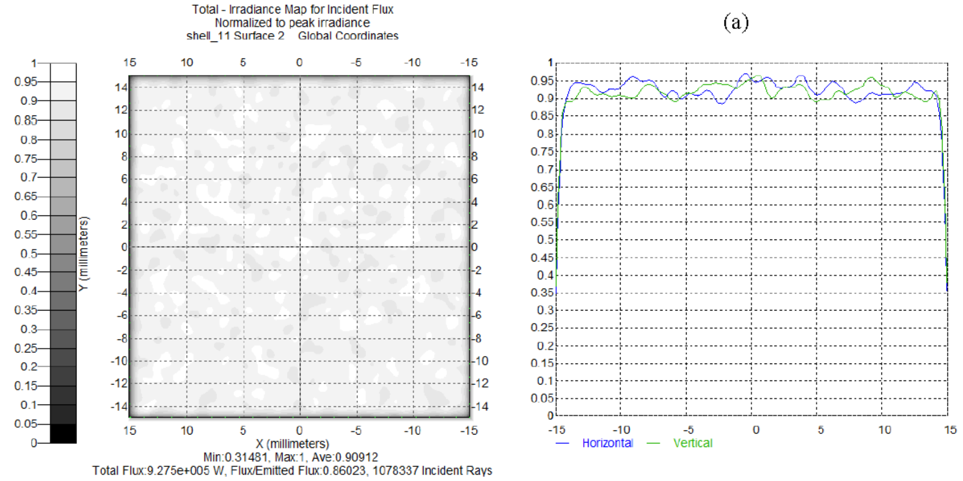
Parameters for the design are listed in Table 4.4, which are the values of the basic known parameters of our design example.

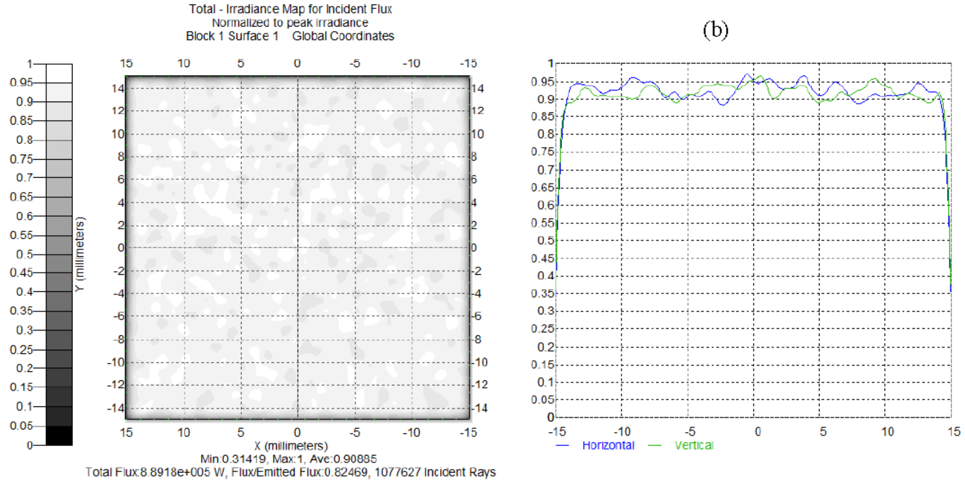
Table 4.4 Parameters of two freeform lens system

Incident beam radius	20mm
Incident beam waist (e^{-2} of beam profile)	10mm
Desired output beam dimension	30mm×30mm
Distance between input and output plane	310mm

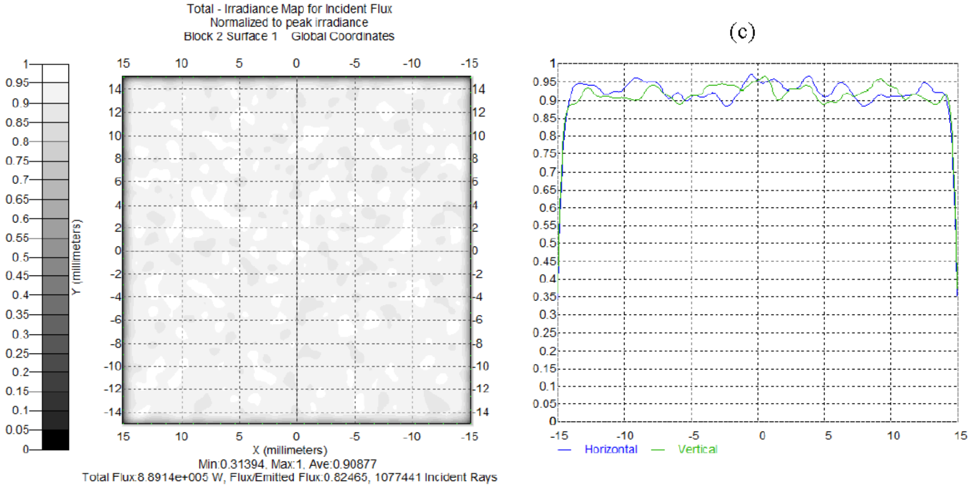
In the design, the freeform surface is reconstructed with 101×101 points based on NURBS. Ray tracing is implemented based on the Monte-Carlo method. About 1,000,000 rays are traced in this beam shaping system.

If the surface parameters are calculated by PDEs, the irradiance distribution of the output beam on four receivers placed at 0mm, 200mm, 500mm and 1000mm away from the flat emitted surface of the secondary lens are shown in Figure 4.7.





Total - Irradiance Map for Incident Flux



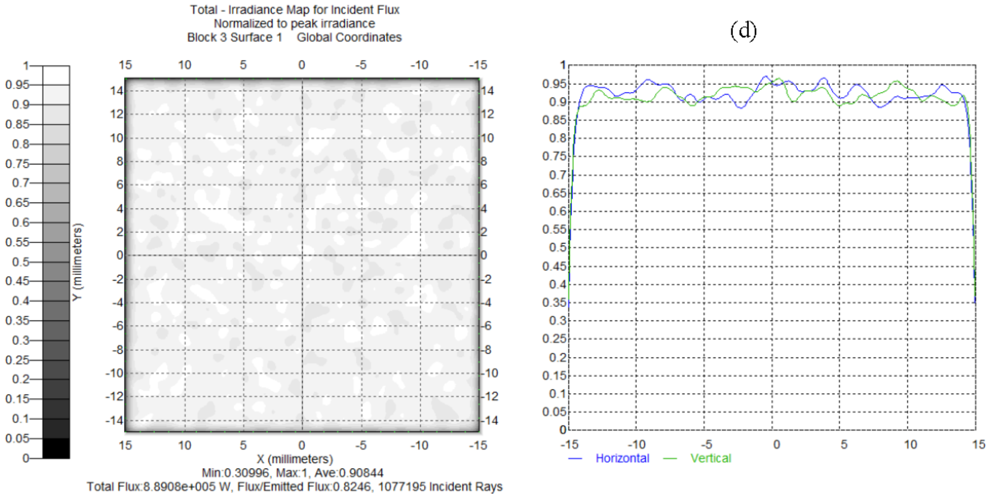
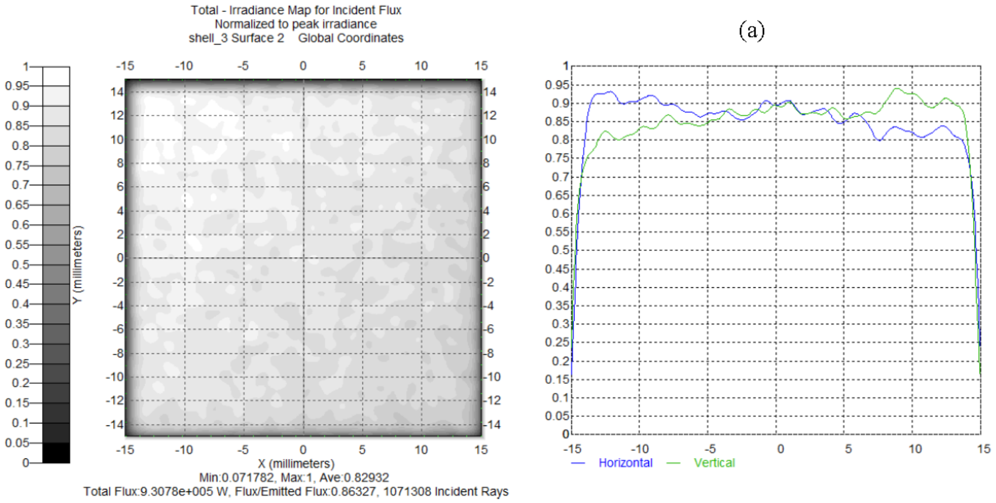


Figure 4.7: In the PDEs method , irradiance distribution on the receivers are placed at (a) 0mm, (b) 200mm, (c) 500mm, (d) 1000mm away from the emitted surface of the second lens.

If the surface parameters are calculated based on variable separation mapping, then the irradiance distribution of the output beam on four receivers placed at 0mm, 200mm, 500mm and 1000mm away from the flat emitted surface of the secondary lens are shown in Figure 4.8.



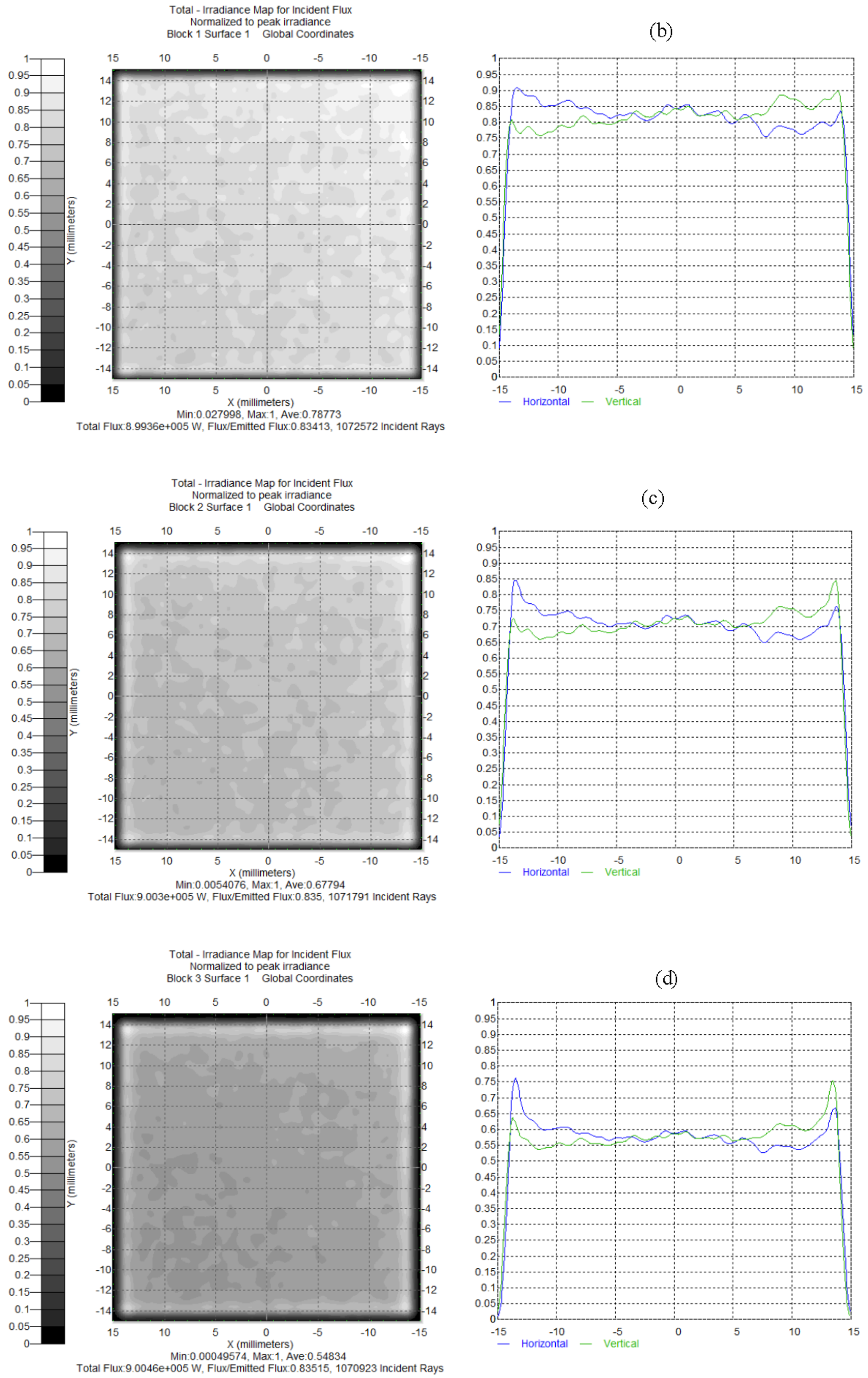

Total - Irradiance Map for Incident Flux
Normalized to peak irradiance
Block 1 Surface 1 Global Coordinates
(b)
Y (millimeters)
X (millimeters)
Min:0.027998, Max:1, Ave:0.78773
Total Flux:8.9936e+005 W, Flux/Emitted Flux:0.83413, 1072572 Incident Rays
Horizontal
Vertical
Total - Irradiance Map for Incident Flux
Normalized to peak irradiance
Block 2 Surface 1 Global Coordinates
(c)
Y (millimeters)
X (millimeters)
Min:0.0054076, Max:1, Ave:0.67794
Total Flux:9.003e+005 W, Flux/Emitted Flux:0.835, 1071791 Incident Rays
Horizontal
Vertical
Total - Irradiance Map for Incident Flux
Normalized to peak irradiance
Block 3 Surface 1 Global Coordinates
(d)
Y (millimeters)
X (millimeters)
Min:0.00049574, Max:1, Ave:0.54834
Total Flux:9.0046e+005 W, Flux/Emitted Flux:0.83515, 1070923 Incident Rays
Horizontal
Vertical
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
0.55
0.6
0.65
0.7
0.75
0.8
0.85
0.9
0.95
1
15
10
5
0
-5
-10
-15
14
12
10
8
6
4
2
0
-2
-4
-6
-8
-10
-12
-14

Figure 4.8: In the variable separation mapping method, irradiance distribution on the receivers are placed at (a) 0mm, (b) 200mm, (c) 500mm, (d) 1000mm away from the emitted surface of the second lens.

From the ray tracing results, in the PDEs method, the ratio of the power on the desired region to the total power is about 90% on four receivers. However, in the variable separation mapping method, the ratio is only about 60%. According to these, we can find that the previous method is more effective to design the two freeform lens to realize uniform illumination. Moreover, the computation is much faster in the PDEs method.

4.2.2 Incident beam is elliptical Gaussian beam

Parameters for the design are listed in Table 4.5, which are the values of the basic known parameters of our design example.

Table 4.5 Parameters of two freeform lens system

Incident beam radius in x direction	26mm
Incident beam radius in y direction	8mm
Incident beam waist w_{0x}	12.7mm
Incident beam waist w_{0y}	3.5mm
Desired output beam dimension	60mm×60mm
Distance between input and output plane	310mm

In the design, the freeform surface is reconstructed with 101×101 points based on NURBS. Ray tracing is implemented based on the Monte-Carlo method. About 1,000,000 rays are traced in this beam shaping system.

The irradiance distribution of the incident and output beam in the source is shown in Figure 4.9.

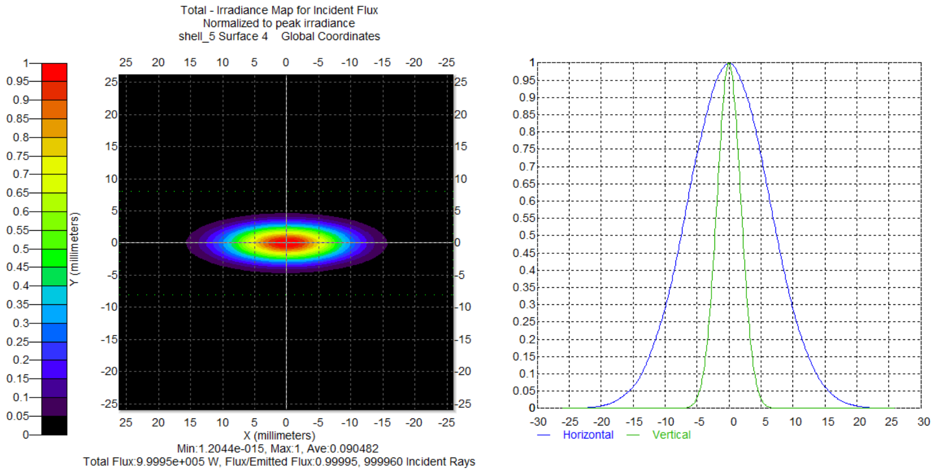
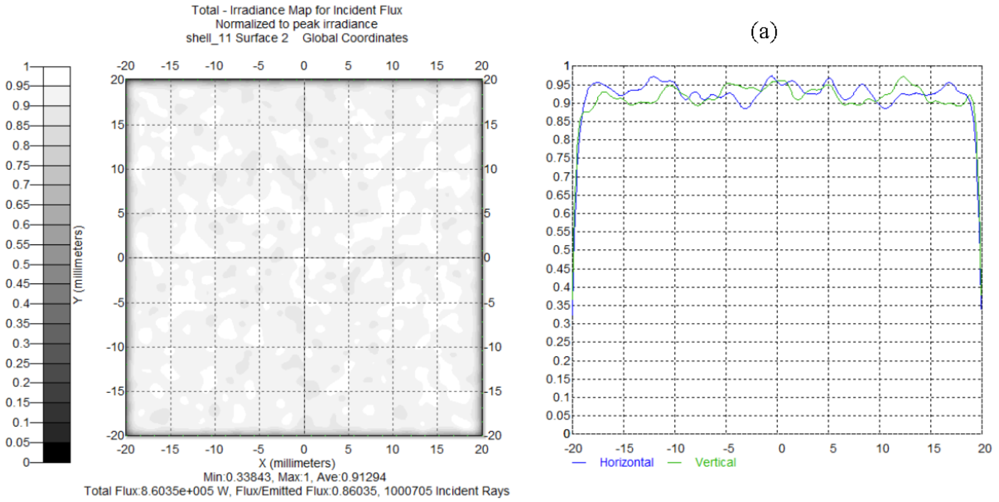


Figure 4.9: Irradiance distribution of the freeform lens on the input plane.

If the surface parameters are calculated by PDEs, the irradiance distribution of the output beam on four receivers placed at 0mm, 200mm, 500mm and 1000mm away from the flat emitted surface of the secondary lens are shown in Figure 4.10.



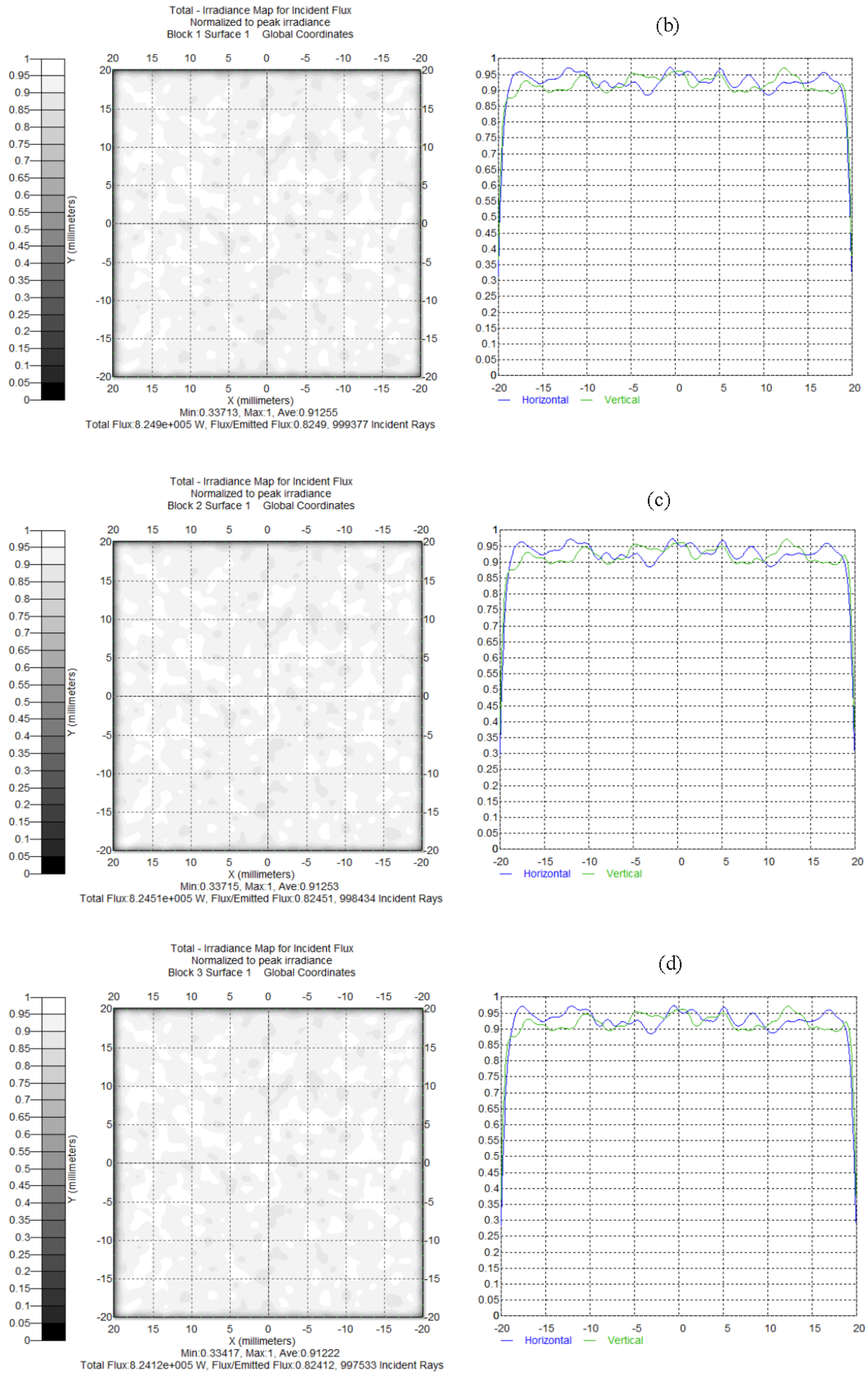
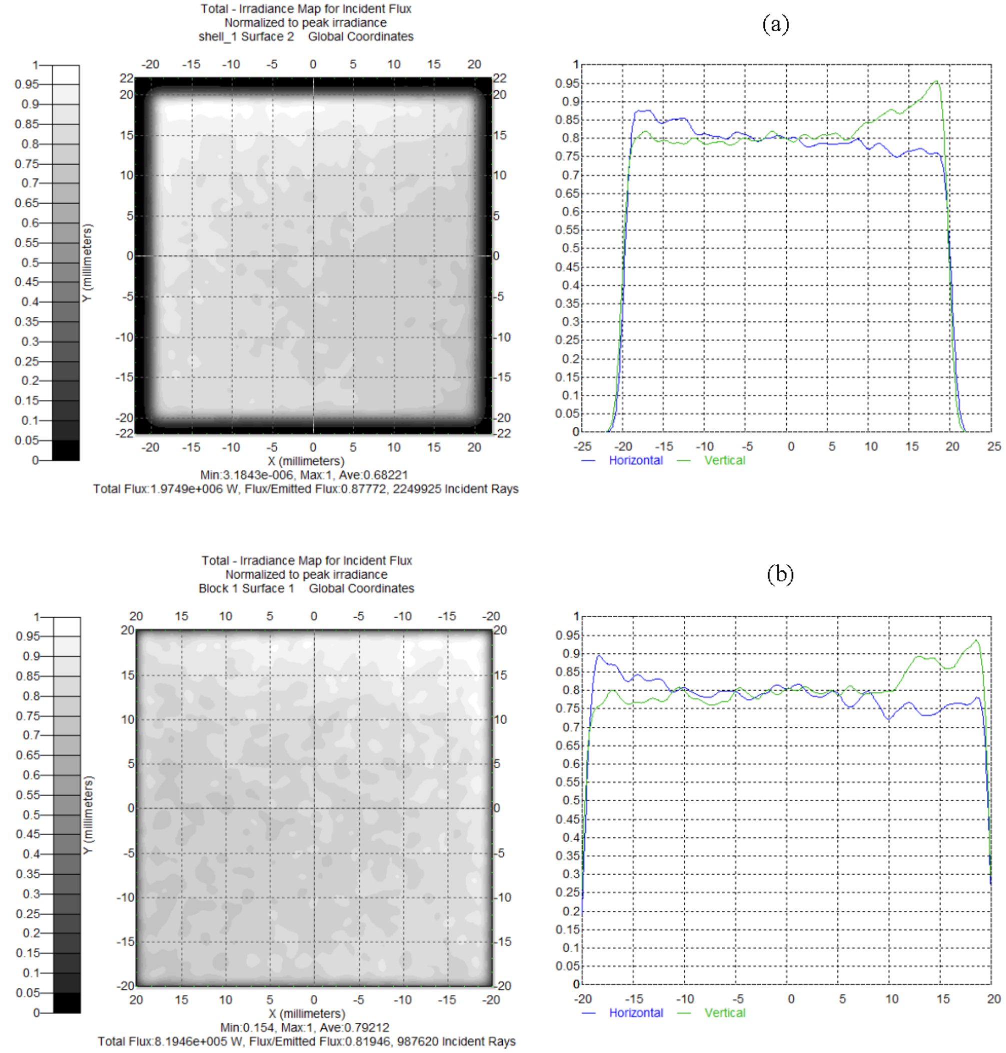
Total - Irradiance Map for Incident Flux
Normalized to peak irradiance
Block 1 Surface 1 Global Coordinates
Y (millimeters)
X (millimeters)
Min:0.33713, Max:1, Ave:0.91255
Total Flux:8.249e+005 W, Flux/Emitted Flux:0.8249, 999377 Incident Rays
(b)
Horizontal
Vertical
Total - Irradiance Map for Incident Flux
Normalized to peak irradiance
Block 2 Surface 1 Global Coordinates
Y (millimeters)
X (millimeters)
Min:0.33715, Max:1, Ave:0.91253
Total Flux:8.2451e+005 W, Flux/Emitted Flux:0.82451, 998434 Incident Rays
(c)
Horizontal
Vertical
Total - Irradiance Map for Incident Flux
Normalized to peak irradiance
Block 3 Surface 1 Global Coordinates
Y (millimeters)
X (millimeters)
Min:0.33417, Max:1, Ave:0.91222
Total Flux:8.2412e+005 W, Flux/Emitted Flux:0.82412, 997533 Incident Rays
(d)
Horizontal
Vertical
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
0.55
0.6
0.65
0.7
0.75
0.8
0.85
0.9
0.95
1
20
15
10
5
0
-5
-10
-15
-20

Figure 4.10: In the PDEs method, irradiance distribution on the receivers are placed at (a) 0mm, (b) 200mm, (c) 500mm, (d) 1000mm away from the emitted surface of the second lens.

If the surface parameters are calculated based on variable separation mapping, then the irradiance distribution of the output beam on four receivers placed at 0mm, 200mm, 500mm and 1000mm away from the flat emitted surface of the secondary lens are shown in Figure 4.11.



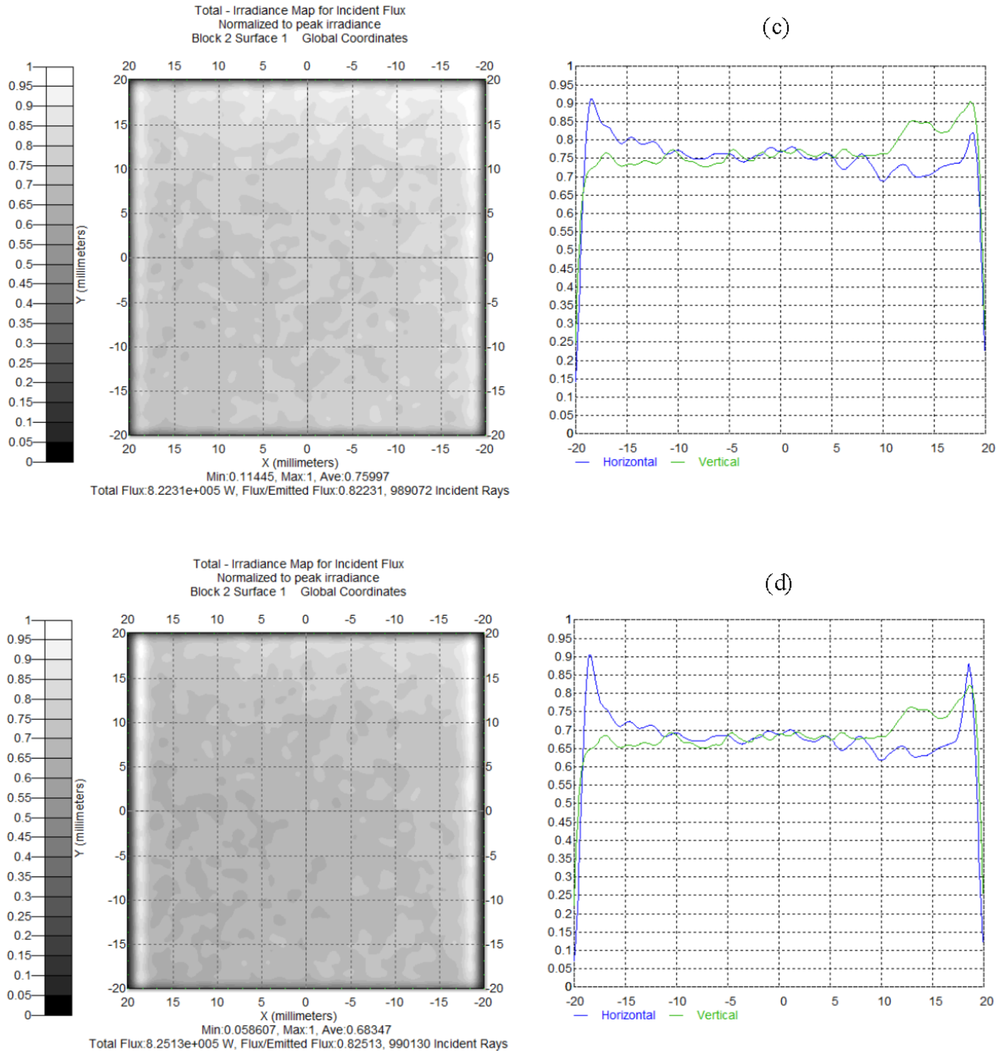


Figure 4.11: In the variable separation mapping method, irradiance distribution on the receivers are placed at (a) 0mm, (b) 200mm, (c) 500mm, (d) 1000mm away from the emitted surface of the second lens.

From the ray tracing results, in the Partial Differential Equations (PDEs) method, the ratio of the power on the desired region to the total power is about 91% on four receivers. However, in the variable separation mapping method, the ratio is only about 82%. According to these, also we can find that the previous method is more effective to design the two freeform lens to realize uniform illumination. Moreover, the computation is much faster in the Partial Differential Equations (PDEs) method.

Above all, the Partial Differential Equations (PDEs) method is more effective and efficient to design freeform lens. And the reconstruction points wouldn't be a big problem in designing. However, more data points should be used in the variable separation mapping method if good illumination result is desired, for example, 400×400 points are normally used in reconstructing the freeform surfaces.

Then Partial Differential Equations (PDEs) method can be used for 3 by 3 laser diode arrays, each lens array is composed of 9 lens, which is used to shape the corresponding laser diode. And the incident beam of circular and elliptical Gaussian shape will be reshaped according to this system.

For circular Gaussian beam, the layout of the system is shown in Figure 4.12, and the irradiance distributions of the incident and output beam in the source and target plane are shown in Figure 4.13 and Figure 4.14, respectively, the target plane is placed at 1000mm away from the emitted surface of the second lens.

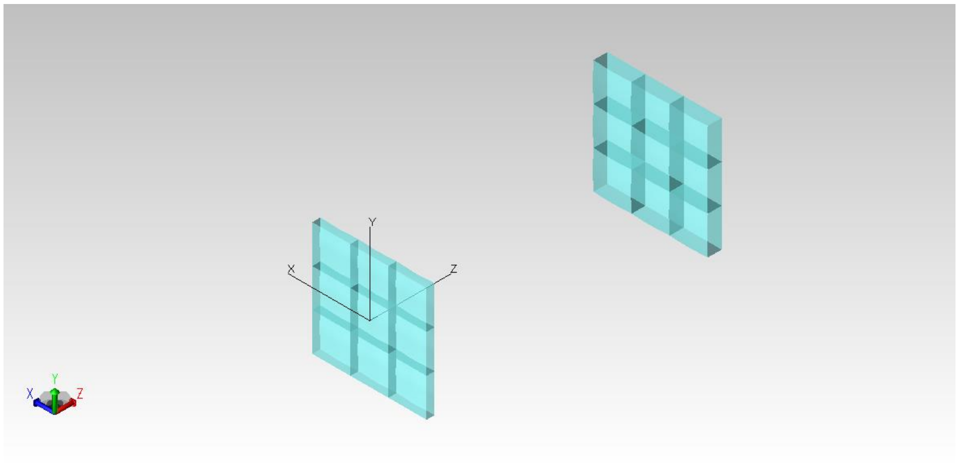


Figure 4.12: The layout of the system with 3-by-3 laser diode arrays, the distribution of incident beam is circular Gaussian.

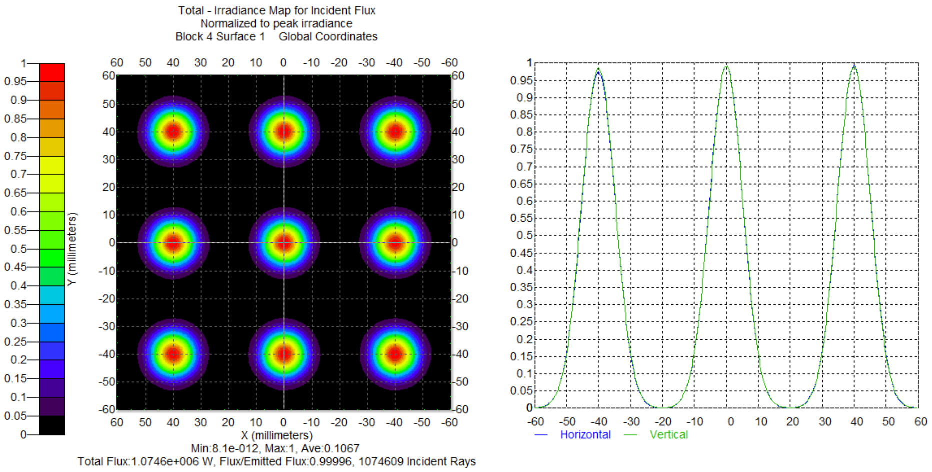


Figure 4.13: Irradiance distribution of the freeform lens on the input plane.

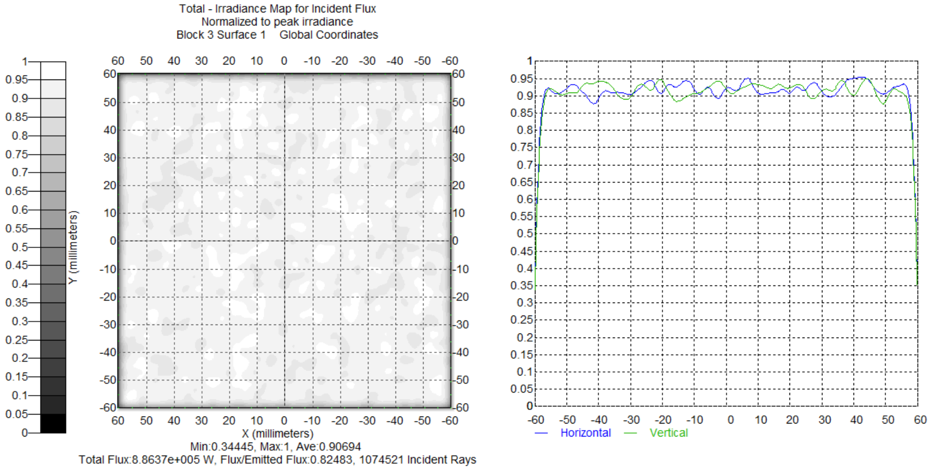


Figure 4.14: Irradiance distribution of the freeform lens on the target plane 1000mm away from the emitted surface of the second lens.

For elliptical Gaussian beam, the layout of the system is shown in Figure 4.15, and the irradiance distributions of the incident and output beam in the source and target plane are shown in Figure 4.16 and Figure 4.17, respectively, the target plane is also placed at 1000mm away from the emitted surface of the second lens.

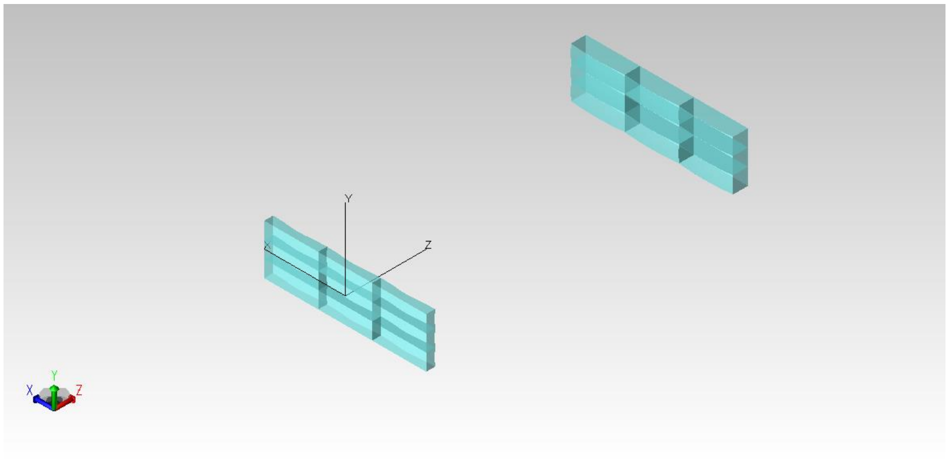


Figure 4.15: The layout of the system with 3-by-3 laser diode arrays, the distribution of incident beam is elliptical Gaussian

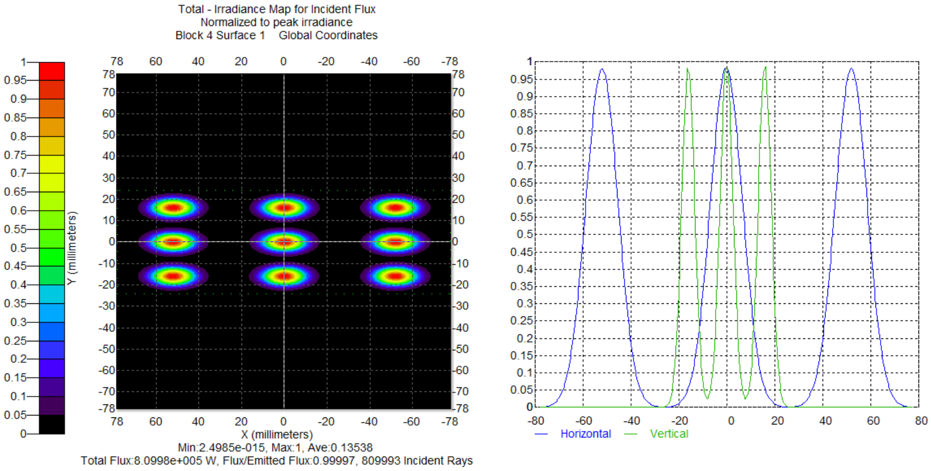


Figure 4.16: Irradiance distribution of the freeform lens on the input plane

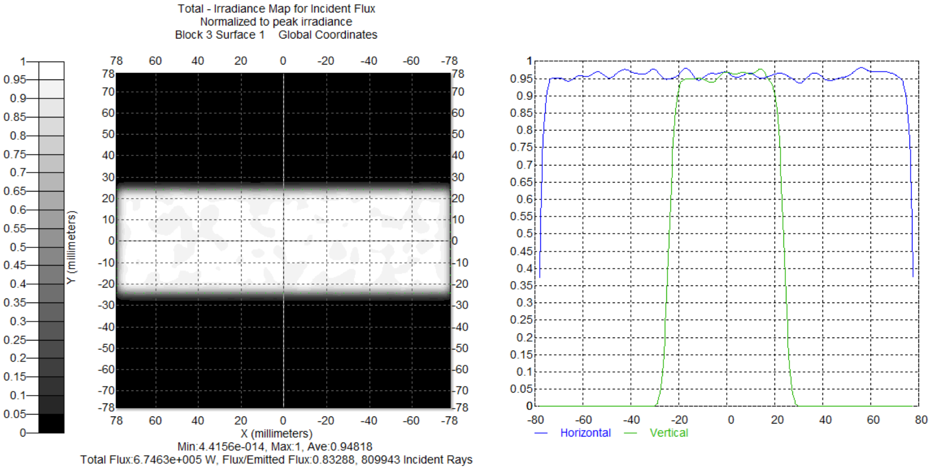


Figure 4.17: Irradiance distribution of the freeform lens on the target plane 1000mm away from the emitted surface of the second lens.

From the stimulated irradiance distributions on the receivers placed at 1000mm away from the emitted surface of the second lens, which are shown in Figure 4.14 and 4.17, the uniformities are 90% and 94%, respectively. In consequence, we can know that the PDEs method is very effective to design the lens array which will reshape irradiance of the laser diode arrays.

CHAPTER 5

CONCLUSIONS

To summarize, the irradiance distribution of the laser diode was mainly concentrated in this work. And the freeform surfaces design method was proposed for transforming this irradiance distribution into uniform one. One freeform lens and two freeform lens were applied to this beam shaping system.

For one freeform lens design, the first order partial differential equations were solved to obtain the parameters of the freeform surface. The simulation results show that this one freeform lens system can effectively transform a collimated Gaussian laser beam into a rectangular “flat-top” one. Furthermore, the lens array was used for the large area illumination, and the simulation results are very good, which means this lens design method is very advantageous. With the help of the lenslet technology, we can simplify the multi-sources shaping problem to the single laser diode shaping problem.

Meantime, for the two freeform lens system, there are two different methods which can be used to calculate the surfaces parameters. One is based on solving partial differential equations, the other is on the basis of variable separation mapping. According to the simulation results, the previous one, the PDEs based method, is more effectively and efficiently in transforming a collimated Gaussian laser beam into a “flat-top” rectangular one with a long depth of field and small divergence.

REFERENCES

[1] Larry Coldren, Scott Corzine, Milan Mashanovitch. "*Diode Lasers and Photonic Integrated Circuits*" (Second Ed.), John Wiley and Sons (2012).

[2] Julio Chaves, *Introduction to Nonimaging Optics*, CRC Press, 2008.

[3] Mert Serkan, "*Laser beam shaping optical system design methods and their application in edge-emitting semiconductor laser-based LIDAR systems*," doctoral dissertation.

[4] L. Wang, K. Y. Qian, and Y. Luo, "*Discontinuous free-form lens design for prescribed irradiance*," *Appl. Opt*. 46(18), 3716–3723 (2007).

[5] Y. Han, X. Zhang, Z. Feng, K. Qian, H. Li, Y. Luo, X. Li, G. Huang, and B. Zhu, "*Variable-separation three dimensional freeform nonimaging optical system design based on target-to-source mapping and micro belt surface construction*, " *Sciencepaper* Online 1–9(2010). http://www.paper.edu.cn/en/paper.php?serial_number=201002-443.

[6] Y. Luo, Z. Feng, Y. Han, and H. Li, "*Design of compact and smooth free-form optical system with uniform illuminance for LED source*," *Opt. Express* 18(9), 9055–9063 (2010).

[7] Computer Desktop Encyclopedia@ 2000 The computer language Co. Inc.

[8] Zexin Feng, Lei Huang, Mali Gong,* and Guofan Jin. "*Beam shaping system design using double freeform optical surfaces*," *Optics Express*, Vol. 21, No. 12 / June 2013.

[9] Rengmao Wu, Zhenrong Zheng,* Haifeng Li, and Xu Liu. "*Constructing optical freeform surfaces using unit tangent vectors of feature data points*," *J. Opt. Soc. Am. A* / Vol. 28, No. 9 / September 2011.

[10] W. J. Cassarly and M. J. Hayford, "*Illumination optimization: the revolution has begun*," *Proc. SPIE* 4832, 258–269 (2002).

[11] T. L. R. Davenport, T. A. Hougha, and W. J. Cassarlya, "*Optimization for illumination systems: the next level of design*," *Proc. SPIE* 5456, 81–90 (2004).

[12] T. R. Davenport, "*3D NURBS representation of surfaces for illumination*," *Proc. SPIE* 4832, 293–301 (2002).

[13] J. Bortz, N. Shatz, and M. Keuper, "*Optimal design of a nonimaging TIR doublet lens for an illumination system using an LED source*," *Proc. SPIE* 5529, 8–16 (2004).

[14] M. G. Turner and K. J. Garcia, "*Optimization using rational Bézier control points and weighting factors*," *Proc. SPIE* 7061, 70610H (2008).

[15] W. Z. Zhang, Q. X. Liu, H. F. Gao, and F. H. Yu, "*Free-form reflector optimization for general lighting*," *Opt. Eng.* 49, 063003 (2010).

[16] Y. Ding, X. Liu, Z. R. Zheng, and P. F. Gu, "*Freeform LED lens for uniform illumination*," *Opt. Express* 16, 12958–12966 (2008).

[17] R. Winston and H. Ries, "*Nonimaging reflectors as functionals of the desired irradiance*," *J. Opt. Soc. Am. A* 10, 1902–1908 (1993).

[18] A. Rabl and J. M. Gordon, "*Reflector design for illumination with extended sources: the basic solutions*," *Appl. Opt*. 33, 6012–6021 (1994).

[19] H. R. Ries and R. Winston, "*Tailored edge-ray reflectors for illumination*," *J. Opt. Soc. Am. A* 11, 1260–1264 (1994).

[20] P. T. Ong, J. M. Gordon, and A. Rabl, "*Tailoring lighting reflectors to prescribed illuminance distributions: compact partialinvolute designs*," *Appl. Opt*. 34, 7877–7887 (1995).

[21] W. Tai and R. Schwarte, “*Design of an aspherical lens to generate a homogenous irradiance for three-dimensional sensors with a light-emitting diode source*,” *Appl. Opt.* 39, 5801–5805 (2000).

[22] H. Ries and J. Muschaweck, “*Tailored freeform optical surfaces*,” *J. Opt. Soc. Am. A* 19, 590–595 (2002).

[23] V. Oliker, “*Optical design of freeform two-mirror beam-shaping systems*,” *J. Opt. Soc. Am. A* 24, 3741–3752 (2007).

[24] Z. R. Zheng, X. Hao, and X. Liu, “*Freeform surface lens for LED uniform illumination*,” *Appl. Opt*. 48, 6627–6634 (2009).

[25] P. Benítez, J. C. Miñano, J. Blen, R. Mohedano, J. Chaves, O. Dross, M. Hernández, J. L. Alvarez, and W. Falicoff, “*SMS design method in 3D geometry: examples and applications*,” *Proc. SPIE* 5185, 18–29 (2004).

[26] F. Muñoz, P. Benítez, O. Dross, J. C. Miñano, and B. Paikyn, “*Simultaneous multiple surface design of compact air-gap collimators for light-emitting diodes*”,*Opt. Eng.* 43, 1522–1530 (2004).

[27] O. Dross, R. Mohedano, P. Benítez, J. C. Miñano, J. Chaves, J. Blen, M. Hernández, and F. Muñoz, “*Review of SMS design methods and real world applications*,” *Proc. SPIE* 5529, 35–47 (2004).

[28] Hazewinkel, Michiel, ed. (2001), "*Runge-Kutta method*", *Encyclopedia of Mathematics, Springer*, ISBN 978-1-55608-010-4

[29] http://en.wikipedia.org/wiki/Optical_path_length

[30] Arthur Schuster, *An Introduction to the Theory of Optics*, *London: Edward Arnold*, 1904

[31] Ghatak, Ajoy (2009), *Optics* (4th ed.), ISBN 0-07-338048-2

[32] Feynman, Richard, *The Feynman Lectures on Physics*, Vol. 1, pp. 26–7

[33] Katz, Mikhail G.; Schaps, David; Shnider, Steve (2013), "*Almost Equal: The Method of Adequality from Diophantus to Fermat and Beyond*", *Perspectives on Science* 21 (3): 7750, arXiv:1210.7750, Bibcode:2012arXiv1210.7750K

[34] B de Greve, "*Reflections and Refractions in Ray tracing*." http://graphics.stanford.edu/courses/cs148-10-summer/docs/2006--degreve--reflection_refraction.pdf

[35] *Eric Weisstein's World of Physics*, Snell's Law, http://scienceworld.wolfram.com

[36] http://en.wikipedia.org/wiki/Non-uniform_rational_B-spline

[37] http://web.cs.wpi.edu/~matt/courses/cs563/talks/nurbs.html

[38] Piegl, L and Tiller, W. "*The NURBS book*." *New York: Springer*, 1997: 188-212.

[39] http://en.wikipedia.org/wiki/Tophat_beam

[40] David L. Shealy[a] and John A. Hoffnagle[b], "*Beam shaping profiles and propagation*." *SPIE Conf*. Laser Beam Shaping VI, Proc. 5876-13, 2005.

[41] http://en.wikipedia.org/wiki/Monte_Carlo_method

[42] Anderson, H.L. (1986). "*Metropolis, Monte Carlo and the MANIAC*". *Los Alamos Science* 14: 96–108.

[43] Baeurle, Stephan A. (2009). "*Multiscale modeling of polymer materials using field-theoretic methodologies: A survey about recent developments*". *Journal of Mathematical Chemistry* 46 (2): 363–426. doi:10.1007/s10910-008-9467-3.

[44] H. Ries and J. Muschaweck, "*Tailoring freeform lenses for illumination*," *Proc. SPIE* 6338, 633808 (2001).

[45] H. Ries and J. Muschaweck, "*Tailored freeform optical surfaces*," *J. Opt. Soc. Am. A* 19, 590-595 (2002).

[46] Y. Ding and P. F. Gu, "*The Freeform Reflector for Uniform Illumination*," *Acta Optica Sinica* 27, 540-544 (2007).

[47] Y. Ding, X. Liu, H. F. Li, and P. F. Gu, "*The design of the freeform reflector for uniform illumination*," *in Proceedings of Asia Display* 2007, Volume 1. (Shanghai, China, 2007), pp. 735-738.

[48] Y. Ding, X. Liu, Z. R. Zheng, and P. F. Gu, "*Freeform LED lens for uniform illumination*," *Opt. Express* 16(17), 12958–12966 (2008).

[49] Zexin Feng, Lei Huang, Mali Gong, and Guofan Jin, "*Beam shaping system design using double freeform optical surfaces*," *Opt. Express* Vol. 21, No. 12 (2013).

[50] L. Wang, K. Y. Qian, and Y. Luo, "*Discontinuous free-form lens design for prescribed irradiance*," *Appl. Opt*. 46(18), 3716–3723 (2007).

[51] Y. Han, X. Zhang, Z. Feng, K. Qian, H. Li, Y. Luo, X. Li, G. Huang, and B. Zhu, "*Variable-separation three dimensional freeform nonimaging optical system design based on target-to-source mapping and micro belt surface construction*," *Sciencepaper Online* 1–9(2010).

[52] Y. Luo, Z. Feng, Y. Han, and H. Li, "*Design of compact and smooth free-form optical system with uniform illuminance for LED source*," *Opt. Express* 18(9), 9055–9063 (2010).

[53] W. B. Elmer, *The optical design of reflectors*, 2nd ed. (*Wiley*, New York, 1980).

[54] F. M. Dickey, S. C. Holswade, and D. L. Shealy, eds., *Laser Beam Shaping Applications* (CRC Press, 2005).

[55] Haotong Ma, Zejin Liu, Pengzhi Jiang, Xiaojun Xu, and Shaojun Du, "*Improvement of Galilean refractive beam shaping system for accurately generating near-diffraction-limited flattop beam with arbitrary beam size*", *Opt. Express* Vol. 19, No. 14.

Appendix

1. Two freeform lens design based on partial differential equations

```
clear all,clc
format long
global z0 nO nI W L N
e1=0;
t1=cputime;
W = 20;
L = 20;
Lx = L;
Ly = W;
w0x = 5;
w0y = 5;
nO = 1;
nI = 1.5095; % BK7 -- 870nm
n1 = nI ;
n2 = nI ;
N = 200;
Dx = 40; % [mm] the size of the output beam in the target plane
Dy = 40;
I0 = 1; % the intensity for the input beam
fzx = @(xp,yp,xq,yq,zp,zq) nO*(xp-xq)/(nO*(zq-zp)-n1*sqrt( (xq-xp)^2 + (yq-yp)^2 + (zq-
zp)^2 )) ;
fzy = @(xp,yp,xq,yq,zp,zq) nO*(yp-yq)/(nO*(zq-zp)-n1*sqrt( (xq-xp)^2 + (yq-yp)^2 + (zq-
zp)^2 )) ;
fun_x_t=@(x,y) exp(-2*(x./w0x).^2).*exp(-2*(y./w0y).^2);
% fplus = @(x,y,tx,ty,z) fzx + fzy;
% % X = zeros(1,N+1);
% % Y = zeros(1,N+1);
% % Z = zeros(N+1,N+1);
% % % % h1 = Lx/N ;
% % % % h2 = Ly/N ;
% % % %
% % % % X = -Lx/2 : h1 : Lx/2 ;
% % % % Y = -Ly/2 : h2 : Ly/2 ;
dx = L/2/N;
h1 = 2*dx;
```

```
h2 = 2*dx;
for i=1:N+1
    for j=1:N+1
        %Space axis
        X(j)=2*((j-1)*dx-L/4);
        Y(i)=2*((i-1)*dx-L/4);
    end
end
E0 = I0 / Dx / Dy * quad2d(fun_x_t,-Lx/2,Lx/2,-Ly/2,Ly/2);   % normalization coefficient ---
for the output beam
% tx = I0 / Dy / E0 * quad2d(fun_x_t,-Lx/2,X,-Ly/2,Ly/2);
% ty = I0 / Dx / E0 * quad2d(fun_x_t,-Lx/2,Lx/2,-Ly/2,Y);
for j = 1:N+1
       tx(j)  = I0 / Dy / E0 * quad2d(fun_x_t,-Lx/2,X(j),-Ly/2,Ly/2);
   end
for i=1:N+1
        ty(i)  = I0 / Dx / E0 * quad2d(fun_x_t,-Lx/2,Lx/2,-Ly/2,Y(i));
   end
tx = tx - Dx/2;
ty = ty - Dy/2;
%% P & Q & S & T
x_s = X ;
y_s = Y ;
x_p = X ;
y_p = Y ;
x_q = tx ;
y_q = ty ;
x_t = tx ;
y_t = ty ;
%% Boundary condition
z_p(1,1) = 8.5 ;
z_p(1,N+1) = 8.5 ;
z_p(N+1,1) = 8.5 ;
z_p(N+1,N+1) = 8.5 ;
z_q(1,1) = 295.5 ;
z_q(1,N+1) = 295.5 ;
z_q(N+1,1) = 295.5 ;
z_q(N+1,N+1) = 295.5 ;
z_s = zeros(N+1,N+1) ;
z_t = 300 * ones(N+1,N+1) ;
% Constancy of the Optical Path Length
Const = n1 * (z_p(1,1) - z_s(1,1)) + nO * sqrt( (x_q(1)-x_p(1))^2 + (y_q(1)-y_p(1))^2 +
(z_q(1,1)-z_p(1,1))^2 ) + n2 * (z_t(1,1)-z_q(1,1)) ;
%% z_p(x,-Ly/2) & z_q(x,-Dy/2)
i=1;
for j = 2:N
```

```
        f1 = feval(fzx,x_p(j-1),y_p(i),x_q(j-1),y_q(i),z_p(i,j-1),z_q(i,j-1));
        % step 2
    xq2 = I0 / Dy / E0 * quad2d(fun_x_t,-Lx/2,x_p(j-1)+h1/2,-Ly/2,Ly/2)  - Dx/2;
    zp2 = z_p(i,j-1)+  h1/2*f1  ;
    ff2 = @(zq2) ([n1 * (zp2- z_s(i,j-1)) + nO * sqrt( (xq2-x_p(j-1)-h1/2)^2 + (y_q(i)-
y_p(i))^2  + (zq2-zp2)^2  ) + n2 * (z_t(i,j-1)-zq2)  - Const]) ;
    zq2 = fsolve(ff2,[z_p(i,j-1)]);
    f2 = feval(fzx,x_p(j-1)+h1/2,y_p(i),xq2,y_q(i),zp2,  zq2);
    % step 3
    zp3 = z_p(i,j-1)+  h1/2*f2  ;
    ff3 = @(zq3) ([n1 * (zp3- z_s(i,j-1)) + nO * sqrt( (xq2-x_p(j-1)-h1/2)^2 + (y_q(i)-
y_p(i))^2  + (zq3-zp3)^2  ) + n2 * (z_t(i,j-1)-zq3)  - Const]) ;
    zq3 = fsolve(ff3,[z_p(i,j-1)]);
    f3 = feval(fzx,x_p(j-1)+h1/2,y_p(i),xq2,y_q(i),zp3,  zq3);
     % step 4
    xq3 = I0 / Dy / E0 * quad2d(fun_x_t,-Lx/2,x_p(j-1)+h1,-Ly/2,Ly/2)  - Dx/2;
    zp4 = z_p(i,j-1)+  h1*f3  ;
    ff4 = @(zq4) ([n1 * (zp4- z_s(i,j-1)) + nO * sqrt( (xq3-x_p(j-1)-h1)^2 + (y_q(i)-y_p(i))^2
+ (zq4-zp4)^2  ) + n2 * (z_t(i,j-1)-zq4)  - Const]) ;
    zq4 = fsolve(ff4,[x_p(i,j-1)]);
    f4 = feval(fzx,x_p(j-1)+h1,y_p(i),xq3,y_q(i),zp4,  zq4);
    z_p(i,j)  = z_p(i,j-1)  + h1*(f1+2*f2+2*f3+f4)/6;
     ff5 = @(zq5) ([n1 * (zp4- z_s(i,j)) + nO * sqrt( (x_q(j)-x_p(j))^2 + (y_q(i)-y_p(i))^2 +
(zq5-z_p(i,j))^2  ) + n2 * (z_t(i,j)-zq5)  - Const]) ;
    z_q(i,j)  = fsolve(ff5,[z_p(i,j)]);
   end
%% z_p(x,Ly/2)  & z_q(x,Dy/2)
i=N+1;
for j = 2:N
      f1 = feval(fzx,x_p(j-1),y_p(i),x_q(j-1),y_q(i),z_p(i,j-1),z_q(i,j-1));
      % step 2
    xq2 = I0 / Dy / E0 * quad2d(fun_x_t,-Lx/2,x_p(j-1)+h1/2,-Ly/2,Ly/2)  - Dx/2;
    zp2 = z_p(i,j-1)+  h1/2*f1  ;
    ff2 = @(zq2) ([n1 * (zp2- z_s(i,j-1)) + nO * sqrt( (xq2-x_p(j-1)-h1/2)^2 + (y_q(i)-
y_p(i))^2  + (zq2-zp2)^2  ) + n2 * (z_t(i,j-1)-zq2)  - Const]) ;
    zq2 = fsolve(ff2,[z_p(i,j-1)]);
    f2 = feval(fzx,x_p(j-1)+h1/2,y_p(i),xq2,y_q(i),zp2,  zq2);
     % step 3
    zp3 = z_p(i,j-1)+  h1/2*f2  ;
    ff3 = @(zq3) ([n1 * (zp3- z_s(i,j-1)) + nO * sqrt( (xq2-x_p(j-1)-h1/2)^2 + (y_q(i)-
y_p(i))^2  + (zq3-zp3)^2  ) + n2 * (z_t(i,j-1)-zq3)  - Const]) ;
    zq3 = fsolve(ff3,[z_p(i,j-1)]);
    f3 = feval(fzx,x_p(j-1)+h1/2,y_p(i),xq2,y_q(i),zp3,  zq3);
      % step 4
    xq3 = I0 / Dy / E0 * quad2d(fun_x_t,-Lx/2,x_p(j-1)+h1,-Ly/2,Ly/2)  - Dx/2;
    zp4 = z_p(i,j-1)+  h1*f3  ;
   ff4 = @(zq4) ([n1 * (zp4- z_s(i,j-1)) + nO * sqrt( (xq3-x_p(j-1)-h1)^2 + (y_q(i)-y_p(i))^2
+ (zq4-zp4)^2 ) + n2 * (z_t(i,j-1)-zq4) - Const]) ;
   zq4 = fsolve(ff4,[z_p(i,j-1)]);
   f4 = feval(fzx,x_p(j-1)+h1,y_p(i),xq3,y_q(i),zp4, zq4);
   z_p(i,j) = z_p(i,j-1) + h1*(f1+2*f2+2*f3+f4)/6;
     ff5 = @(zq5) ([n1 * (z_p(i,j)- z_s(i,j)) + nO * sqrt( (x_q(j)-x_p(j))^2 + (y_q(i)-y_p(i))^2
+ (zq5-z_p(i,j))^2 ) + n2 * (z_t(i,j)-zq5) - Const]) ;
   z_q(i,j) = fsolve(ff5,[z_p(i,j)]);
end
%% z_p(-Lx/2,y) & z_q(-Dx/2,y)
j = 1 ;
for i = 2:N
   g1 = feval(fzy,x_p(j),y_p(i-1),x_q(j),y_q(i-1),z_p(i-1,j),z_q(i-1,j));
   % step 2
   yq2 = I0 / Dx / E0 * quad2d(fun_x_t,-Lx/2,Lx/2,-Ly/2,y_p(i-1)+h2/2) - Dy/2;
   zp2 = z_p(i-1,j)+ h2/2*g1 ;
   gg2 = @(zq2) ([n1 * (zp2 - z_s(i-1,j)) + nO * sqrt( (x_q(j)-x_p(j))^2 + (yq2-y_p(i-1)-
h2/2)^2 + (zq2-zp2)^2 ) + n2 * (z_t(i-1,j)-zq2) - Const]) ;
   zq2 = fsolve(gg2,[z_p(i-1,j)]);
   g2 = feval(fzy,x_p(j),y_p(i-1)+h2/2,x_q(j),yq2,zp2,zq2);
   % step 3
   zp3 = z_p(i-1,j)+ h2/2*g2 ;
   gg3 = @(zq3) ([n1 * (zp3 - z_s(i-1,j)) + nO * sqrt( (x_q(j)-x_p(j))^2 + (yq2-y_p(i-1)-
h2/2)^2 + (zq3-zp3)^2 ) + n2 * (z_t(i-1,j)-zq3) - Const]) ;
   zq3 = fsolve(gg3,[z_p(i-1,j)]);
   g3 = feval(fzy,x_p(j),y_p(i-1)+h2/2,x_q(j),yq2,zp3,zq3);
   % step 4
   yq3 = I0 / Dx / E0 * quad2d(fun_x_t,-Lx/2,Lx/2,-Ly/2,y_p(i-1)+h2) - Dy/2;
   zp4 = z_p(i-1,j)+ h2*g3 ;
   gg4 = @(zq4) ([n1 * (zp4 - z_s(i-1,j)) + nO * sqrt( (x_q(j)-x_p(j))^2 + (yq3-y_p(i-1)-h2)^2
+ (zq4-zp4)^2 ) + n2 * (z_t(i-1,j)-zq4) - Const]) ;
   zq4 = fsolve(gg4,[z_p(i-1,j)]);
   g4 = feval(fzy,x_p(j),y_p(i-1)+h2,x_q(j),yq3,zp4,zq4);
   z_p(i,j) = z_p(i-1,j) + h2*(g1+2*g2+2*g3+g4)/6;
   gg5 = @(zq5) ([n1 * (z_p(i,j)- z_s(i,j)) + nO * sqrt( (x_q(j)-x_p(j))^2 + (y_q(i)-y_p(i))^2 +
(zq5-z_p(i,j))^2 ) + n2 * (z_t(i-1,j)-zq5) - Const]) ;
   z_q(i,j) = fsolve(gg5,[z_p(i,j)]);
end
%% other points
for j = 2:N+1
for i = 2:N
   g1 = feval(fzy,x_p(j),y_p(i-1),x_q(j),y_q(i-1),z_p(i-1,j),z_q(i-1,j));
   % step 2
   yq2 = I0 / Dx / E0 * quad2d(fun_x_t,-Lx/2,Lx/2,-Ly/2,y_p(i-1)+h2/2) - Dy/2;
   zp2 = z_p(i-1,j)+ h2/2*g1 ;
```

```
    gg2 = @(zq2) ([n1 * (zp2 - z_s(i-1,j)) + nO * sqrt( (x_q(j)-x_p(j))^2 + (yq2-y_p(i-1)-h2/2)^2 + (zq2-zp2)^2 ) + n2 * (z_t(i-1,j)-zq2)  - Const]) ;
    zq2 = fsolve(gg2,[z_p(i-1,j)]);
    g2 = feval(fzy,x_p(j),y_p(i-1)+h2/2,x_q(j),yq2,zp2,zq2);
    % step 3
    zp3 = z_p(i-1,j)+  h2/2*g2  ;
    gg3 = @(zq3) ([n1 * (zp3 - z_s(i-1,j)) + nO * sqrt( (x_q(j)-x_p(j))^2 + (yq2-y_p(i-1)-h2/2)^2 + (zq3-zp3)^2 ) + n2 * (z_t(i-1,j)-zq3)  - Const]) ;
    zq3 = fsolve(gg3,[z_p(i-1,j)]);
    g3 = feval(fzy,x_p(j),y_p(i-1)+h2/2,x_q(j),yq2,zp3,zq3);
      % step 4
    yq3 = I0 / Dx / E0 * quad2d(fun_x_t,-Lx/2,Lx/2,-Ly/2,y_p(i-1)+h2)  - Dy/2;
    zp4 = z_p(i-1,j)+  h2*g3  ;
    gg4 = @(zq4) ([n1 * (zp4 - z_s(i-1,j)) + nO * sqrt( (x_q(j)-x_p(j))^2 + (yq3-y_p(i-1)-h2)^2 + (zq4-zp4)^2 ) + n2 * (z_t(i-1,j)-zq4)  - Const]) ;
    zq4 = fsolve(gg4,[z_p(i-1,j)]);
    g4 = feval(fzy,x_p(j),y_p(i-1)+h2,x_q(j),yq3,zp4,zq4);
    z_p(i,j)  = z_p(i-1,j)  + h2*(g1+2*g2+2*g3+g4)/6;
    gg5 = @(zq5) ([n1 * (z_p(i,j)- z_s(i,j)) + nO * sqrt( (x_q(j)-x_p(j))^2 + (y_q(i)-y_p(i))^2 + (zq5-z_p(i,j))^2  ) + n2 * (z_t(i-1,j)-zq5)  - Const]) ;
    z_q(i,j)  = fsolve(gg5,[z_p(i,j)]);
end
end
%% change the sign of y-axis
X1 = X; % unchangable
Y1 = -Y;
tx1 = tx ;
ty1 = -ty ;
z_p = flipud(z_p);
z_q = flipud(z_q);
%% rhino
for i=1:N+1
    for j=1:N+1
        x_S(i,j)  = X1(j);
        x_P(i,j)  = X1(j);
              x_Q(i,j)  = tx1(j);
        x_T(i,j)  = tx1(j);
              y_S(i,j)  = Y1(i);
        y_P(i,j)  = Y1(i);
              y_Q(i,j)  = ty1(i);
        y_T(i,j)  = ty1(i);

    end
end
z_S = z_s ;
z_P = z_p ;
```

```
z_Q = z_q ;
z_T = z_t ;
ii=N+1;
jj=ii*ii;
S_rhino=zeros(jj,3);
for j=1:ii
   S_rhino((j-1)*ii+1:j*ii,1)=x_S(:,j);
   S_rhino((j-1)*ii+1:j*ii,2)=y_S(:,j);
   S_rhino((j-1)*ii+1:j*ii,3)=z_S(:,j);
end
P_rhino=zeros(jj,3);
for j=1:ii
   P_rhino((j-1)*ii+1:j*ii,1)=x_P(:,j);
   P_rhino((j-1)*ii+1:j*ii,2)=y_P(:,j);
   P_rhino((j-1)*ii+1:j*ii,3)=z_P(:,j);
end
Q_rhino=zeros(jj,3);
for j=1:ii
   Q_rhino((j-1)*ii+1:j*ii,1)=x_Q(:,j);
   Q_rhino((j-1)*ii+1:j*ii,2)=y_Q(:,j);
   Q_rhino((j-1)*ii+1:j*ii,3)=z_Q(:,j);
end
T_rhino=zeros(jj,3);
for j=1:ii
   T_rhino((j-1)*ii+1:j*ii,1)=x_T(:,j);
   T_rhino((j-1)*ii+1:j*ii,2)=y_T(:,j);
   T_rhino((j-1)*ii+1:j*ii,3)=z_T(:,j);
end
e1=cputime-t1+e1;
figure
pcolor(x_P,y_P,z_P);
figure
pcolor(x_Q,y_Q,z_Q);
```

2. Two freeform lens design based on separated variables mapping

```
clear all,clc
format long
% OE_2013_Beam shaping system design using double freeform optical surfaces
w0=10; x0=15; % distance from the respective axes at which the intensity has fallen to half
its axial value
y0=15;
betax=40;
betay=40;
fun_in=@(x,y) exp(-2*(x./w0).^2).*exp(-2*(y./w0).^2);
Q_in=quad2d(fun_in,-20,20,-20,20);
fun_out=@(x,y) ((1+exp(betax*(abs(x)./x0-1))).^-1).*((1+exp(betay*(abs(y)./y0-1))).^-1);
```

```
Q_out=quad2d(fun_out,-15,15,-15,15);
AxAy=Q_in./Q_out;  % [xs,ys]=meshgrid(-10:20/400:10,-10:20/400:10);
% xs=linspace(-10,10,401);
% ys=linspace(-10,10,401);
% xt=zeros(1,401);
% yt=zeros(1,401);
% xt(1)=-20;
% yt(1)=-5;
% % % % % % % % % %
N=100;
% % % % % % % % % %
L=20;
% dx : step size
dx=L/N;
for i=1:N+1
   for j=1:N+1
      %Space axis
      xs(j)=2*((j-1)*dx-L/2);
      ys(i)=2*((i-1)*dx-L/2);
%     Gau(n,m)=exp(-(x(m)^2+y(n)^2)/(wo^2));
%     FDrect(n,m)=((1+exp(betax*(abs(x(m))/x0-1)))^-1)*((1+exp(betay*(abs(y(n))/y0-1)))^-1);
%      r=sqrt(x(m)^2+y(n)^2);
%      FDGau(n,m)=(1+exp(beta*(r/R0-1)))^(-1);
   end
end
xtt(1)=-15;
xtt(N+1)=15;
ytt(1)=-15;
ytt(N+1)=15;
% for i=1:size(rrr,2)-1
%     rrr(rrr==0)=eps;
%     RRR(RRR==0)=eps;
%     kk1=((2/pi/g0/(w0)^2)*(rrr(i)/RRR(i))*(1+exp(beta*(RRR(i)/R0-1)))*exp(-2*(rrr(i))^2/w0^2));
%     tt=RRR(i)+kk1*h;
%     tt(tt==0)=eps;
%     kk2=((2/pi/g0/(w0)^2)*(rrr(i+1)/tt)*(1+exp(beta*(tt/R0-1)))*exp(-2*(rrr(i+1))^2/w0^2));
%     RRR(i+1)=RRR(i)+h*(kk1+kk2)/2;
% end

% h=0.0001;
% for m=1:size(xs,2)-1
%     kk1=A3*((1+exp(betax*(abs(xt(m))./x0-1))).^-1).*((exp(-2*(xs(m)./w0).^2)).^-1);
%     tt=xt(m)+kk1*h;
```

```
    %     kk2=A3*((1+exp(betax*(abs(tt)./x0-1))).^-1).*((exp(-2*(xs(m+1)./w0).^2)).^-1);
    %     xt(m+1)=xt(m)+h*(kk1+kk2)/2;
    % end
    % A3=A1*A2/AxAy/A4;  %
    % fun3=@(xx,t)  (1+exp(betax*(abs(xx)./x0-1))).^-1;
    % xx0=xt(1);
    % dxx=0.1;
    % yy0=0;
    % yy_end=0;
    % e1=0;
    % while  yy_end<A3
    %     t1=cputime;
    %     xx=xx0+dxx;
    %     xspan=[xx0  xx];
    %     [t,yy]=ode45(fun3,xspan,yy0);
    %     yy_end=yy(end);
    %     e1=cputime-t1+e1;
    % end
    % for j=2:size(xs,2)
    % %     xsj=xs(j);
    %     fun1=@(x,w0)  exp(-2*(x./w0).^2);
    %     A1 = quadgk(@(x)fun1(x,w0),xs(1),xs(j));
    %     fun2=@(y,w0)  exp(-2*(y./w0).^2);
    %     A2 = quadgk(@(y)fun2(y,w0),ys(1),ys(401));
    %
    %     B2 = 10;
    %     xtt = A1*A2/B2  + xt(1);
    %     xt(j)=xtt;
    % end
% Regardless  the integrand  you could use cumtrapz and interp1
    %
    % xmax=10; %some  large  value
    % x=linspace(0,xmax,1000);
    % f=sin(x);%change  here
    %
    % x0=interp1(cumtrapz(x,f),x,L,'linear')
    %
    % Hope it helps,
    % Per
    % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
    fun2=@(y)  exp(-2*(y./w0).^2);
    A2=quadl(fun2,ys(1),ys(end));
    fun3=@(y)  (1+exp(betay*(abs(y)./y0-1))).^-1;
    A3=quadl(fun3,ytt(1),ytt(end));
    M=N/2;
    q1=[];
```

```
q2=[];
for j=2:M
    fun1=@(x)  exp(-2*(x./w0).^2);
    A1=quadl(fun1,xs(1),xs(j));
    q1=[q1;A1];
    L=A1*A2/AxAy/A3;
    xx=linspace(xtt(1),0,1000000);
    f=(1+exp(betax*(abs(xx)./x0-1))).^-1;
    xx0=interp1(cumtrapz(xx,f),xx,L,'spline');
    xtt(j)=xx0;
       fun5=@(xxx)(1+exp(betax*(abs(xxx)./x0-1))).^-1;
    A5=quadl(fun5,xtt(1),xtt(j));
    q2=[q2;A5];
end
dqx=q1*A2/AxAy/A3-q2;
xxtt=xtt(1,1:M);
xt=[xxtt  0 fliplr(abs(xxtt))];
fun6=@(x)  exp(-2*(x./w0).^2);
A6=quadl(fun6,xs(1),xs(end));
fun7=@(x)  (1+exp(betax*(abs(x)./x0-1))).^-1;
A7=quadl(fun7,xtt(1),xtt(end));
q3=[];
q4=[];
for i=2:M
    fun8=@(y)  exp(-2*(y./w0).^2);
    A8=quadl(fun8,ys(1),ys(i));
    q3=[q3;A8];
    L1=A6*A8/AxAy/A7;
    yy=linspace(ytt(1),0,1000000);
    f1=(1+exp(betay*(abs(yy)./y0-1))).^-1;
    yy0=interp1(cumtrapz(yy,f1),yy,L1,'spline');
    ytt(i)=yy0;
       fun9=@(yyy)(1+exp(betay*(abs(yyy)./y0-1))).^-1;
    A9=quadl(fun9,ytt(1),ytt(i));
    q4=[q4;A9];
end
dqy=q3*A6/AxAy/A7-q4;   yytt=ytt(1,1:M);
yt=[yytt  0 fliplr(abs(yytt))];
% % % % % % % % % % %
ys=-ys;
yt=-yt;
% % % % % % % % % % %
savefile1='xt.mat';
save(savefile1,'xt');
savefile2='yt.mat';
save(savefile2,'yt');
```