Analytic-domain lens design with proximate ray tracing

Nan Zheng,[1] Nathan Hagen,[2,*] and David J. Brady[2,3]

[1]*Department of Physics, Duke University, Durham, North Carolina 27708, USA*
[2]*Department of Electrical and Computer Engineering, Fitzpatrick Institute for Photonics, Duke University, Durham, North Carolina 27708, USA*
[3]*E-mail: david.brady@duke.edu*
[*]*Corresponding author: nhagen@optics.arizona.edu*

Received February 17, 2010; revised June 3, 2010; accepted June 5, 2010; posted June 7, 2010 (Doc. ID 124169); published July 14, 2010

We have developed an alternative approach to optical design which operates in the analytical domain so that an optical designer works directly with rays as analytical functions of system parameters rather than as discretely sampled polylines. This is made possible by a generalization of the proximate ray tracing technique which obtains the analytical dependence of the rays at the image surface (and ray path lengths at the exit pupil) on each system parameter. The resulting method provides an alternative direction from which to approach system optimization and supplies information which is not typically available to the system designer. In addition, we have further expanded the procedure to allow asymmetric systems and arbitrary order of approximation, and have illustrated the performance of the method through three lens design examples. © 2010 Optical Society of America

OCIS codes: 080.2740, 080.6755, 220.3620.

1. INTRODUCTION

The design of an optical system typically proceeds with the use of a modern computer program for performing exact ray tracing. The user first models the optical layout and constructs a merit function, after which the program uses ray trace data to minimize the merit function by adjusting the system parameters. The most common optimization algorithm used is damped least-squares [1], which is fast but has the drawback that it is limited to finding only local minima. If the starting optical configuration is far enough from the global minimum, and the merit function is complicated enough to possess a number of local minima (as in almost all cases of interest), then the chance of converging on the global solution quickly approaches zero. Global optimization techniques [2] such as simulated annealing [3] or genetic algorithms [4] can of course get around this problem, but require extensive computational resources [5]. While it is possible to minimize the computational effort needed by providing the algorithm with a starting configuration which is as close as possible to the global solution, thereby minimizing the search space, the method for doing this relies heavily on the intuition of the designer and on tools developed in classical aberration theory (such as aberration cancellation in symmetric systems).

We present what we believe to be a new design tool which approaches the problem from a different direction, and thus may be helpful in situations where traditional tools get stuck: a ray tracing engine that provides the expression for the rays, to any desired polynomial order of approximation, as an analytical function of the system parameters. The approach is a generalization of the proximate ray tracing method developed by Hopkins [6–8]—a generalization which is made possible through the power of modern computer algebra systems [9] to manipulate large analytical equations.

In its original form, proximate ray tracing involves first analytically calculating the formulas for transfer and refraction at each polynomial order of approximation. A ray trace then involves calculating the *numerical* values of the ray-surface intersection point at each surface, at each order of approximation. The procedure was thus developed as an efficient method of numerically calculating higher-order aberration coefficients, and it requires tracing only a small set of special rays to obtain the aberration coefficient values. The generalization we present here extends proximate ray tracing's analytical approach to the entire design process, without numerical substitution. Thus, rather than obtaining analytical formulas for each individual refraction and transfer step in the trace, we obtain a single formula for the rays at the image plane (and exit pupil). We can thus obtain the aberration coefficients in *analytical* form, in which the functional dependence of the lens merit function on the system parameters is retained. The aim is to show that with the aid of modern computer algebra systems, optical design problems that are currently performed numerically can also be done in the analytical domain, and that this can have significant advantages for understanding the design problem.

A closely related approach for analytical ray tracing was taken by Kondo and Takeuchi [10] through the use of matrices and the selection of a proper vector basis for modeling the nonlinear effects present in ray tracing (up to the desired order of approximation). While similar to the approach presented here, it lacks the conceptual sim-

plicity of proximate ray tracing, and thus we feel that it is a much more cumbersome tool to work with. That is, for many optical engineers, the concept of equating terms of like order in the Taylor expansion of a nonlinear equation is a much more familiar process than that of constructing a nonorthogonal vector basis in which to represent a nonlinear transformation. This conceptual simplicity becomes important for understanding what to do with the large set of polynomial terms generated by either approach.

Kondo and Takeuchi's matrix approach was modified by Lakshminarayanam and Varadharajan [11], and also by Almeida [12], and adapted for use in a computer algebra system [13], but published work has been limited to optical modeling rather than design, in that it does not treat the merit function or the optimization procedure. This misses one of the central strengths of working in the analytical domain: the tractability of polynomial equations allows the use of more robust optimization techniques.

Other attempts at analytical ray tracing have also been made. Walther [14,15] developed an analytical approach which makes use of eikonals rather than rays and is thus less accessible for many optical engineers. Kryszczyński [16] provided some tentative steps toward analytical design based on rays, but only supplies an outline of an algorithm. Although it may at first appear mathematically complex, we hope to show that the proximate ray tracing method makes the analytical approach both accessible and practical.

In the discussion below, we first review Hopkins' method [6–8] and show how it can be readily generalized to arbitrary orders of approximation and to asymmetric systems. We then show how to construct the merit function and optimizer for designing optical systems with this approach and present three example designs. Finally, we conclude with a discussion of the advantages and disadvantages of this technique.

2. TRANSFER

Hopkins [6–8] described proximate ray tracing as an iterative ray tracing technique in algebraic form. The basic approach is

1. A polynomial series expansion for basic and intermediate variables are inserted into the exact ray trace equations.
2. Any sines and cosines are series-expanded, and any multiplications, divisions, and square roots are performed as series operations.
3. Terms of a given order are collected together, and higher-order terms are obtained using lower-order solutions via a triangular set of equations.

Ray tracing consists of two basic operations: transfer and refraction. Using the ray path length w as the transfer parameter, the transfer equations can be written as

$$\mathbf{r}_{s+1} = \mathbf{r}_s + w_s \mathbf{c}_s, \tag{1}$$

where $\mathbf{r}=(x,y,z)$ is the ray position vector, $\mathbf{c}=(c_x,c_y,c_z)$ is the direction cosine vector, and s indicates the surface index (i.e., transfer from surface s to surface $s+1$). To simplify the equations below, we will usually leave the surface index implied, except where it is needed. Here z is taken to be the optical axis, and multiplying w by the refractive index n of the medium gives the ray optical path length. The refraction equations can be written as [[17], p. 133]

$$n(\mathbf{c}\times\mathbf{N}) = n'(\mathbf{c}'\times\mathbf{N}), \tag{2}$$

where $\mathbf{N}=(N_x,N_y,N_z)$ is the normal vector of the surface, and

$$\mathbf{N} = \nabla f(\mathbf{r}), \tag{3}$$

when $f(\mathbf{r})=0$ defines the surface. Note that the optical path length of a ray through the entire system is given by $W=\sum_{s=1}^{S-1} n_s w_s$ when w_s is the ray path length for transfer from surface s to $s+1$.

The first step in the proximate ray trace procedure is to expand all of the relevant variables in the transfer and refraction equations in various orders of approximation. Thus, each ray trace variable is expressed in the form

$$\begin{aligned}
x &= 0 + x^{(1)} + x^{(2)} + x^{(3)} + x^{(4)} + \cdots,\\
y &= 0 + y^{(1)} + y^{(2)} + y^{(3)} + y^{(4)} + \cdots,\\
z &= z^{(0)} + z^{(1)} + z^{(2)} + z^{(3)} + z^{(4)} + \cdots,\\
c_x &= 0 + c_x^{(1)} + c_x^{(2)} + c_x^{(3)} + c_x^{(4)} + \cdots,\\
c_y &= 0 + c_y^{(1)} + c_y^{(2)} + c_y^{(3)} + c_y^{(4)} + \cdots,\\
c_z &= 1 + c_z^{(1)} + c_z^{(2)} + c_z^{(3)} + c_z^{(4)} + \cdots,\\
w &= w^{(0)} + w^{(1)} + w^{(2)} + w^{(3)} + w^{(4)} + \cdots.
\end{aligned} \tag{4}$$

The superscripts in parentheses indicate the order of approximation so that the first nonzero term on the right hand side of each of these equations represents a paraxial variable, and succeeding terms represent the nonlinear dependence on the paraxial variables. The term $z^{(0)}$ represents the axial transfer distance from the surface vertex to the previous surface (and is thus a negative quantity for rays propagating from left to right).

The primary variables used to define the rays, the entrance pupil coordinates (x_{ep},y_{ep}) and the field angles (θ_x,θ_y), are treated as paraxial variables and thus do not have an order-expansion. The final expressions for the rays will give the image coordinates in terms of these primary variables and of the parameters used to define each surface. In aberration theory it is more common to work with normalized field angles $\mathbf{H}$, defined either as $(H_x,H_y)=(1/\theta_{max})(\theta_x,\theta_y)$, where $\theta_{max}\equiv[(\max\{\theta_x\})^2+(\max\{\theta_y\})^2]^{1/2}$, or as $(H_x,H_y)=(\theta_x/\max\{\theta_x\},\theta_y/\max\{\theta_y\})$. In the presentation below, we will continue to use θ rather than H to represent field angles.

The elements of the direction cosine vector for the incident ray are given by $c_x=\sin\theta_x$, $c_y=\sin\theta_y$, and $c_z=\pm\sqrt{1-c_x^2-c_y^2}$ for a ray propagating in the $\pm z$ direction. Thus

$$(c_x^{(1)} + c_x^{(3)} + \cdots) = \theta_x - \frac{1}{3!}\theta_x^3 + \cdots,$$

$$(c_y^{(1)} + c_y^{(3)} + \cdots) = \theta_y - \frac{1}{3!}\theta_y^3 + \cdots,$$

$$(c_z^{(0)} + c_z^{(2)} + c_z^{(4)} + \cdots) = 1 - \frac{1}{2}(\theta_x^2 + \theta_y^2) + \left[\frac{1}{16}(\theta_x^2 + \theta_y^2)^2 + \frac{1}{6}(\theta_x^4 + \theta_y^4)\right] + \cdots.$$

In Hopkins' presentation [6–8] of the proximate ray trace equations, the variables x, y, c_x, and c_y have only odd-order terms, and the variables z, c_z, and w have only even-order terms due to his assumption of rotational symmetry about the optical axis. In order to design more general optical systems, we drop these symmetry assumptions here and use the most general form of the equations.

Substituting the order-expanded variables for $\mathbf{r}$, $\mathbf{c}$, and w into the transfer equations (1) results in

$$0 = z_0^{(0)} + w^{(0)},$$

$$\mathbf{r}_s^{(1)} = \mathbf{r}_0^{(1)} + \mathbf{c}^{(1)}w^{(0)} + \mathbf{c}^{(0)}w^{(1)},$$

$$\mathbf{r}_s^{(2)} = \mathbf{r}_0^{(2)} + \mathbf{c}^{(2)}w^{(0)} + \mathbf{c}^{(1)}w^{(1)} + \mathbf{c}^{(0)}w^{(2)},$$

$$\mathbf{r}_s^{(3)} = \mathbf{r}_0^{(3)} + \mathbf{c}^{(3)}w^{(0)} + \mathbf{c}^{(2)}w^{(1)} + \mathbf{c}^{(1)}w^{(2)} + \mathbf{c}^{(0)}w^{(3)},$$

$$\vdots$$

in which we label the surface before transfer as $s=0$ and that following transfer as simply s. Each line contains only terms of equal order, such that the number of equations in the resulting system is equal to the desired order of approximation. Using the fact that $c_z^{(0)}=1$ and $c_x^{(0)}=c_y^{(0)}=0$, the transfer equations split into

$$w^{(0)} = -z_0^{(0)},$$

$$w^{(1)} = z_s^{(1)} - z_0^{(1)} - c_z^{(1)}w^{(0)},$$

$$w^{(2)} = z_s^{(2)} - z_0^{(2)} - c_z^{(2)}w^{(0)} - c_z^{(1)}w^{(1)},$$

$$\vdots$$

$$x_s^{(1)} = x_0^{(1)} + c_x^{(1)}w^{(0)},$$

$$x_s^{(2)} = x_0^{(2)} + c_x^{(2)}w^{(0)} + c_x^{(1)}w^{(1)},$$

$$\vdots$$

$$y_s^{(1)} = y_0^{(1)} + c_y^{(1)}w^{(0)},$$

$$y_s^{(2)} = y_0^{(2)} + c_y^{(2)}w^{(0)} + c_y^{(1)}w^{(1)},$$

$$\vdots \qquad (5)$$

The zeroth-order equation gives $w^{(0)}=-z_0^{(0)}$, which is simply the axial distance from the previous surface vertex to the current surface vertex. Substituting this result into the first-order equation allows one to solve for $w^{(1)}$, and we can likewise continue to substitute lower-order results to obtain solutions for the higher-order equations. The resulting sequence of values for w, i.e., $(w^{(0)}, w^{(0)}+w^{(1)}, w^{(0)}+w^{(1)}+w^{(2)}, \ldots)$, provides an estimate of the real ray path length to increasing order of approximation. Once all desired orders of w have been solved for, one can then substitute into the equations for x and y. As with w, the sequence $(\mathbf{r}_s^{(0)}, \mathbf{r}_s^{(0)}+\mathbf{r}_s^{(1)}, \mathbf{r}_s^{(0)}+\mathbf{r}_s^{(1)}+\mathbf{r}_s^{(2)}, \ldots)$ provides an estimate of the ray-surface intersection location to increasing order of approximation (see Fig. 1).

One further step is necessary before we can use this procedure to solve this set of equations. Since the various orders of the surface sag z_s are not yet known, we cannot yet solve directly for w. First we need to express z_s in terms of known quantities, and for this we need to perform the order-expansion of the surface equation.

3. SURFACE EQUATION

The order-expansion of the surface equation is obtained by taking its Taylor expansion, shown here for a surface in the form $z=z(x,y)$

$$z_s(x_s,y_s) = z_s(0,0) + x_s\left[\frac{\partial z_s}{\partial x_s}\right]_{(0,0)} + y_s\left[\frac{\partial z_s}{\partial y_s}\right]_{(0,0)} + \frac{1}{2}x_s^2\left[\frac{\partial^2 z_s}{\partial x_s^2}\right]_{(0,0)} + x_s y_s\left[\frac{\partial^2 z_s}{\partial x_s\,\partial y_s}\right]_{(0,0)} + \frac{1}{2}y_s^2\left[\frac{\partial^2 z_s}{\partial y_s^2}\right]_{(0,0)} + \cdots, \qquad (6)$$

in which the $(x_s,y_s)=(0,0)$ subscript on each square bracket indicates that the partial derivatives are evaluated at the axial surface point. From Eq. (6), we next substitute in the order-expansion forms of (x_s,y_s,z_s) and equate terms of equal order, giving

$$z_s^{(0)} = z_s(0,0),$$

$$z_s^{(1)} = x_s^{(1)}\left[\frac{\partial z_s}{\partial x_s}\right]_{(0,0)} + y_s^{(1)}\left[\frac{\partial z_s}{\partial y_s}\right]_{(0,0)},$$

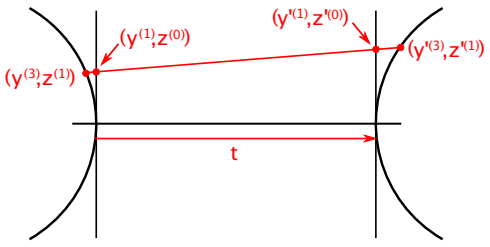

Fig. 1. (Color online) The transfer operation involves taking low-order polynomial approximations of surfaces and modifying the surface intersection coordinates to increasing accuracy as higher orders are traced. Shown here are only the first- and third-order approximations of a spherical surface. The ray path length w from the left to right surfaces depends on the order of approximation, as indicated in Eqs. (5).

$$z_s^{(2)} = x_s^{(2)}\left[\frac{\partial z_s}{\partial x_s}\right]_{(0,0)} + y_s^{(2)}\left[\frac{\partial z_s}{\partial y_s}\right]_{(0,0)} + \frac{1}{2}[x_s^{(1)}]^2\left[\frac{\partial^2 z_s}{\partial x_s^2}\right]_{(0,0)} + \frac{1}{2}[y_s^{(1)}]^2\left[\frac{\partial^2 z_s}{\partial y_s^2}\right]_{(0,0)} + x_s^{(1)}y_s^{(1)}\left[\frac{\partial^2 z_s}{\partial x_s\,\partial y_s}\right]_{(0,0)},$$

$$\vdots$$

In the case of a spherical surface with radius of curvature R whose center of curvature lies on the optical axis, the surface equation can be written as $z=-R+\sqrt{R^2-r_s^2}$ (for $r_s=\sqrt{x_s^2+y_s^2}$). The order-expansion of this surface produces

$$z_s^{(0)} = 0,$$

$$z_s^{(1)} = 0,$$

$$z_s^{(2)} = \frac{[r_s^{(1)}]^2}{2R},$$

$$z_s^{(3)} = 0,$$

$$z_s^{(4)} = \frac{r_s^{(1)}r_s^{(3)}}{R} + \frac{[r_s^{(1)}]^4}{8R^3},$$

$$\vdots$$

where $[r_s^{(1)}]^2 \equiv [x_s^{(1)}]^2+[y_s^{(1)}]^2$. The even-order components are equal to zero here due to rotational symmetry. Now we can see that higher-order forms of z_s can be written in terms of lower-order forms of x_s and y_s. Thus, if we return to Eqs. (5) and substitute in for the order-expanded form of z_s given here, we find that the higher-order equations can all be expressed in terms of lower-order quantities, allowing the full system of equations to be solved.

While we have shown the order-expansion of a spherical surface, one may also define a surface by an arbitrary polynomial in x_s and y_s, such that $z=z(x,y)$ is given by

$$z_s = \alpha_1 x_s + \alpha_2 y_s + \alpha_3 x_s^2 + \alpha_4 x_s y_s + \alpha_5 y_s^2 + \alpha_6 x_s^3 + \alpha_7 x_s^2 y_s + \alpha_8 x_s y_s^2 + \alpha_9 y_s^3 + \cdots. \tag{7}$$

In this case the surface order-expansion gives

$$z_s^{(0)} = 0,$$

$$z_s^{(1)} = \alpha_1 x_s^{(1)} + \alpha_2 y_s^{(1)},$$

$$z_s^{(2)} = \alpha_1 x_s^{(2)} + \alpha_2 y_s^{(2)} + 2\alpha_3[x_s^{(1)}]^2 + \alpha_4 x_s^{(1)} y_s^{(1)} + 2\alpha_5[y_s^{(1)}]^2,$$

$$\vdots$$

4. REFRACTION

The next step in the ray trace procedure is solving for the refracted ray direction. By combining the refraction equations (2) and the equation for the surface normal (3) with the normalization condition for the direction cosine vector, $|\mathbf{c}'|=1$, we can solve for the refracted direction cosines $\mathbf{c}'$,

$$c_x' = \frac{B_x \pm N_x\sqrt{D}}{A_{xy}}, \tag{8}$$

$$c_y' = \frac{B_y \pm N_y\sqrt{D}}{A_{xy}}, \tag{9}$$

$$c_z' = \frac{B_z \pm \sqrt{D}}{A_z}, \tag{10}$$

where

$$A_{xy} = n_2^2 N_z(N_x^2+N_y^2+N_z^2),$$

$$A_z = n_2^2(N_x^2+N_y^2+N_z^2),$$

$$B_x = n_1 n_2 N_z[N_y^2 c_x - N_x N_y c_y + N_z(N_z c_x - N_x c_z)],$$

$$B_y = n_1 n_2 N_z[N_x^2 c_y - N_x N_y c_x + N_z(N_z c_y - N_y c_z)],$$

$$B_z = n_1 n_2[-N_z(N_x c_x + N_y c_y) + (N_x^2+N_y^2)c_z],$$

$$D = n_2^2 N_z^2[n_2^2(N_x^2+N_y^2+N_z^2) - n_1^2(N_z^2(c_x^2+c_y^2) - 2N_x N_z c_x c_z - 2N_y c_y(N_x c_x + N_z c_z) + N_y^2(c_x^2+c_z^2) + N_x^2(c_y^2+c_z^2))], \tag{11}$$

and n_1, n_2 are the refractive indices of the media before and after refraction. In the equations for $\mathbf{c}'$ [Eqs. (8)–(10)], choosing for the solution the positive sign in front of the square root selects a ray propagating in the $+z$ direction.

The square root, multiplication, and division in Eqs. (8)–(10) are each nonlinear procedures and so we must perform each operation in the context of power series to the appropriate order. The order-expansion of the square root can be done by searching for an order-expanded variable α, whose square is equal to D, i.e.,

$$(\alpha^{(0)} + \alpha^{(1)} + \cdots)(\alpha^{(0)} + \alpha^{(1)} + \cdots) = (D^{(0)} + D^{(1)} + \cdots).$$

This involves solving a triangular set of equations,

$$\alpha^{(0)}\alpha^{(0)} = D^{(0)},$$

$$\alpha^{(0)}\alpha^{(1)} + \alpha^{(1)}\alpha^{(0)} = D^{(1)},$$

$$\alpha^{(0)}\alpha^{(2)} + \alpha^{(1)}\alpha^{(1)} + \alpha^{(2)}\alpha^{(0)} = D^{(2)},$$

$$\vdots \tag{12}$$

Substituting the order-expanded variables into definition (11) of D and sorting terms by order, we can obtain the expressions for $D^{(0)}$, $D^{(1)}$, etc. Inserting these into Eqs. (12), we can solve the zeroth-order equation to give $\alpha^{(0)}$. Following the back-substitution procedure, we then use each lower-order solution to solve each higher-order equa-

tion and eventually obtain all of the unknown terms in α up to the desired order.

The next step is to perform the division step in Eqs. (8)–(10). For a division such as $c=b/a$ in which all three variables are taken to have series form (i.e., $c=c^{(0)}+c^{(1)}+c^{(2)}+\cdots$, etc.), we can solve this operation by first multiplying both sides by a,

$$(c^{(0)}+c^{(1)}+\cdots)(a^{(0)}+a^{(1)}+\cdots)=(b^{(0)}+b^{(1)}+\cdots),$$

and once again solving the resulting triangular set of equations for the terms of c,

$$c^{(0)}a^{(0)}=b^{(0)},$$

$$c^{(0)}a^{(1)}+c^{(1)}a^{(0)}=b^{(1)},$$

$$c^{(0)}a^{(2)}+c^{(1)}a^{(1)}+c^{(2)}a^{(0)}=b^{(2)},$$

$$\vdots$$

via back-substitution.

Finally, we also need to obtain the equation for the surface normal vector $\mathbf{N}=\nabla[z(x,y)-z]$ so that

$$N_x=\frac{\partial}{\partial x_s}z_s(x_s,y_s),$$

$$N_y=\frac{\partial}{\partial y_s}z_s(x_s,y_s),$$

$$N_z=-1. \tag{13}$$

As with all other nonprimary variables in the system, we take an order-expansion of the normal vector components in terms of primary variables. The order-expansion for $\mathbf{N}$ takes the form

$$N_x=N_x^{(0)}+N_x^{(1)}+N_x^{(2)}+N_x^{(3)}+N_x^{(4)}+\cdots,$$

$$N_y=N_y^{(0)}+N_y^{(1)}+N_y^{(2)}+N_y^{(3)}+N_y^{(4)}+\cdots,$$

$$N_z=N_z^{(0)}+N_z^{(1)}+N_z^{(2)}+N_z^{(3)}+N_z^{(4)}+\cdots, \tag{14}$$

which we can use to replace the terms on the left hand side of each equation in Eqs. (13). On the right hand side of each equation, we can use the computer algebra system to obtain the derivative of z_s and substitute into the result the order-expansion forms of x_s and y_s. In general, this requires a great deal of analytical work to perform each mathematical operation in order-expansion form, but computer algebra systems can work through these steps without difficulty. For example, for a spherical surface, the partial derivatives needed for Eqs. (13) are

$$\frac{\partial}{\partial x}z(x,y)=\frac{-x}{\sqrt{R^2-x^2-y^2}},$$

$$\frac{\partial}{\partial y}z(x,y)=\frac{-y}{\sqrt{R^2-x^2-y^2}},$$

so that the division, square root, and square operations must be done in order-expansion form. In the case of the polynomial surface example (7), however, we readily obtain the result explicitly,

$$N_x^{(0)}=-\alpha_1,$$

$$N_y^{(0)}=-\alpha_2,$$

$$N_x^{(1)}=-2\alpha_3x_s^{(1)}-\alpha_4y_s^{(1)},$$

$$N_y^{(1)}=-\alpha_4x_s^{(1)}-2\alpha_5y_s^{(1)},$$

$$N_x^{(2)}=-2\alpha_3x_s^{(2)}-\alpha_4y_s^{(2)}-3\alpha_6[x_s^{(1)}]^2-2\alpha_7x_s^{(1)}y_s^{(1)}-\alpha_8[y_s^{(1)}]^2,$$

$$N_y^{(2)}=-\alpha_4x_s^{(2)}-2\alpha_5y_s^{(2)}-\alpha_7[x_s^{(1)}]^2-2\alpha_8x_s^{(1)}y_s^{(1)}-3\alpha_9[y_s^{(1)}]^2,$$

$$\vdots$$

Performing this sequence of operations once for each of Eqs. (8)–(10) gives the solution for the refracted ray direction cosine vector $\mathbf{c}'$. The resulting sequence of values for $\mathbf{c}'$, i.e., $(\mathbf{c}'^{(0)},\mathbf{c}'^{(0)}+\mathbf{c}'^{(1)},\mathbf{c}'^{(0)}+\mathbf{c}'^{(1)}+\mathbf{c}'^{(2)},\ldots)$, provides an estimate of the real refracted ray angle to increasing order of approximation. Note that for an nth-order ray trace, the surface must be expanded to order $n+1$ prior to taking its derivative in order to obtain an nth-order form for the surface normal.

While following this ray trace procedure manually is tedious and error-prone, it can be made fast and robust through the use of modern computer algebra systems. In fact, most such systems provide enough functionality that the entire transfer and refraction operations can each be performed in a couple lines of code, and is typically executed within seconds for systems of modest complexity. (See Appendix A for comments on how to structure the code to help make this possible.)

5. MERIT FUNCTION

The final result of a proximate ray trace calculation is to obtain $x=x^{(1)}+x^{(2)}+\cdots$ and $y=y^{(1)}+y^{(2)}+\cdots$, the position of the ray at the image plane to each order of approximation. Each term is itself a function of the ray coordinate at the entrance pupil $(x_{\mathrm{ep}},y_{\mathrm{ep}})$ and the incident ray angle (θ_x,θ_y) so that we have a polynomial expression in these four variables in addition to all of the parameters used to define the various surfaces and their spacings. In order to design a system, we need to construct a merit function, for which the mean square spot size is a common choice [18]. Denoting the merit function by M, we can write

$$M=\int\int\int\int[x(\,)-x_G(\,)]^2+[y(\,)-y_G(\,)]^2\mathrm{d}x_{\mathrm{ep}}\mathrm{d}y_{\mathrm{ep}}\mathrm{d}\theta_x\mathrm{d}\theta_y, \tag{15}$$

where $(\,)$ represents the variable and parameter dependence of the ray coordinates, i.e., $(x_{\mathrm{ep}},y_{\mathrm{ep}},\theta_x,\theta_y,\ldots)$, in

which the ellipsis indicates the surface parameters of the system. For a system of spherical surfaces in a rotationally symmetric design, the surface parameters take the form $(R_0,t_0,n_0,R_1,t_1,n_1\ldots)$—the radius of curvature, spacing, and refractive index for the successive surfaces. In the merit function integral, (x_G,y_G) is the Gaussian image point, for which we can simply substitute the first-order solution of the ray location at the image plane, $(x^{(1)},y^{(1)})$.

As in the ray trace procedure, if we wish to work with an analytical merit function, we can insert the order-expanded variables for $x(\ )$ and $y(\ )$ in the above integral, collect terms of like order, and perform the integral on each order independently. The resulting expression can be quite long for optical systems of even moderate complexity, and so the use of a computer algebra program is essential here. If the optical system possesses rotational symmetry, we can simplify the merit function expression to have the form

$$M = \int\int\int [x(\ ) - x_G(\ )]^2 + [y(\ ) - y_G(\ )]^2 \mathrm{d}\phi\rho\mathrm{d}\rho\mathrm{d}\theta,$$

in which the primary variables are no longer $(x_{\mathrm{ep}},y_{\mathrm{ep}},\theta_x,\theta_y,\ldots)$ but rather $(\rho,\phi,\theta,\ldots)$. That is, the field angle can be expressed as a scalar, and the pupil location is now expressed in cylindrical coordinates, i.e., $(x_{\mathrm{ep}},y_{\mathrm{ep}}) = (\rho\cos\phi,\rho\sin\phi)$. For the majority of imaging systems, the most appropriate choice of integration range is a rectangular field and a circular pupil. As long as the integration range allows us to obtain analytical functions for polynomial integrands, then it remains possible to obtain an analytical function for the merit function as well. Note that the squaring of the terms in the integrand results in a merit function polynomial of order $2p+4$ after integration, where p is the order of approximation in the ray trace, and the additional four orders arise from the four integrals of Eq. (15).

In addition, if we wish to use the spot centroid rather than the Gaussian image point as our reference for the merit function, a choice which amounts to ignoring the effects of distortion on the image, then we can replace (x_G,y_G) with the appropriate centroid $(\bar{x},\bar{y})$ given by

$$\bar{x}(\ ) = \int\int\int\int x(\ )\mathrm{d}x_{\mathrm{ep}}\mathrm{d}y_{\mathrm{ep}}\mathrm{d}\theta_x\mathrm{d}\theta_y,$$

$$\bar{y}(\ ) = \int\int\int\int y(\ )\mathrm{d}x_{\mathrm{ep}}\mathrm{d}y_{\mathrm{ep}}\mathrm{d}\theta_x\mathrm{d}\theta_y.$$

After constructing the merit function, the final step in designing an optical system is the implementation of an optimization algorithm to determine the system parameters which minimize M. Here we run into many of the same problems encountered by optimization in the existing design software: while local techniques are compact and fast, they typically cannot reach the global solution; while global techniques are capable of finding the optimal solution, they require unrealistic computational resources in order to do so. (Reference [19] provides a useful survey of modern algorithms for solving polynomial equations.) For rotationally symmetric systems of modest complexity or asymmetric systems of low complexity, existing algorithms are capable of locating global minima. Beyond these, one must compromise between the computational resources available and the restriction to local domains. In the examples shown in Section 7 below, for low complexity systems, we use fast global techniques such as Mathematica's [20] NMinimize function, whereas for more complex problems we resort to simulated annealing.

If the designer wishes, it is also possible to constrain the optimization using implicit functions of the system parameters. For example, if we wish to restrict the lens diameters to be within some allowed range, then—given the functional form of the ray at the appropriate surface—we can obtain equations of constraint. For example, for a rotationally symmetric spherical lens, the surface equation gives

$$z(y) = \frac{y^2}{2R} + \frac{y^4}{8R^3} + \cdots,$$

for ray height y and radius of curvature R. Constraining y to be less than some value y_0 allows us to solve for an equation of constraint on R. This can be used by the optimization routine to look for solutions lying only within the valid design space.

The optimization can run into trouble due to the sheer size of the analytical formulas produced by the ray trace, especially for systems with more than a few surfaces and with surfaces having many modeling parameters (such as high-order aspheres). When this happens, one thing that can be done is to fix some of the system parameters and optimize over the remaining ones. For example, if we fix the thickness of a lens, then we can give it the numerical value during ray tracing so that it need not be tracked analytically. This can greatly simplify the resulting expressions and make ray tracing and optimization much faster.

6. EXAMPLE RAY TRACE

Since the discussion up to this point has given the general expressions for the proximate ray tracing technique, we illustrate the approach with a simple example, using the lens shown in Fig. 2 (see also Table 1). This lens model has been chosen such that the example ray trace can be presented easily on a printed page: an $f/2.2$ singlet cylin-

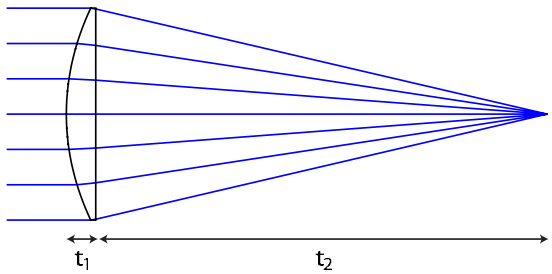

Fig. 2. (Color online) 2D lens model used for the example. Once the analytical model is completed, we substitute the following parameter values to give the lens shown: $t_1=5$ mm, $t_2=76.6667$ mm, and the refractive index is $n=1.5$. The first lens surface is convex, and the second surface is planar. The resulting $f/2.2$ design has an entrance pupil diameter of 32 mm and a focal length of 80 mm.

Table 1. Prescription for the Lens Shown in Fig. 2

	s	Radius of Curvature	Thickness	Index
Pupil	0	∞	0	
Lens front	1	R	t_1	n
Lens back	2	∞	t_2	
Image	3	∞		

drical lens with the (planar) aperture stop placed to coincide with the front surface of the lens. Limiting the ray trace to two dimensions reduces the definition of a ray to its (y,z)-coordinates and its angle θ_y. In the ray trace given below, all subscripts refer to the surface number s, and we limit the approximation to third order for space reasons. To match common practice in ray tracing, the surface equations are expressed in terms of the surface vertex point, with the axial transfer distance inserted into $w^{(0)}$ in the transfer equations.

The ray trace starts at the pupil plane ($s=0$), with the primary variables y_{ep} and θ_y defining all incident rays. The first surface is defined by the equation

$$z = \frac{y^2}{2R} + \frac{y^4}{8R^3}.$$

Applying the transfer equations gives

$$w_0^{(0)} = 0, \quad w_0^{(2)} = \frac{y_{\mathrm{ep}}^2}{2R},$$

$$y_1^{(1)} = y_{\mathrm{ep}}, \quad y_1^{(3)} = \frac{y_{\mathrm{ep}}^2 \theta_y}{2R},$$

$$z_1^{(0)} = 0, \quad z_1^{(2)} = \frac{y_{\mathrm{ep}}^2}{2R},$$

and the refraction equations for the refracted direction cosines are

$$[c_y^{(1)}]_1 = \left[\frac{1}{R}(n-1)y_{\mathrm{ep}} + \theta_y n\right],$$

$$[c_y^{(3)}]_1 = \frac{1}{6R^3}[\theta_y^3 R^3 n - 3\theta_y^2 y_{\mathrm{ep}} R^2 n(n-1) - 3y_{\mathrm{ep}}^3 n(n-1) - 3\theta_y y_{\mathrm{ep}}^2 (n^2 - n - 1)],$$

$$[c_z^{(0)}]_1 = 1,$$

$$[c_z^{(2)}]_1 = -\frac{1}{2R^2}[y_{\mathrm{ep}}^2(1 - 2n + n^2) + y_{\mathrm{ep}} 2n(\theta_y R n - \theta_y R) + \theta_y^2 R^2 n^2].$$

Transferring to the planar back surface of the lens,

$$w_1^{(0)} = t_1,$$

$$w_1^{(2)} = \frac{1}{2R^2}[t_1 y_{\mathrm{ep}}^2 (n-1)^2 + \theta_y^2 R^2 t_1 n^2 - y_{\mathrm{ep}}^2 R + 2\theta_y y_{\mathrm{ep}} R t_1 n(n-1)],$$

$$y_2^{(1)} = \frac{t_1}{R} y_{\mathrm{ep}}(n-1) + y_{\mathrm{ep}} + \theta_y t_1 n,$$

$$y_2^{(3)} = \frac{1}{6R^3}(3\theta_y^2 y_{\mathrm{ep}} R^2 t_1 n(n-1)(1+3n) + \theta_y^3 R^3 t_1 n(3n^2 - 1) + 3y_{\mathrm{ep}}^3 (n-1)[-R + t_1 + t_1 n(n-1)] - 3\theta_y y_{\mathrm{ep}}^2 R(n-1)[R + t_1(3n^2 + n - 1)]),$$

$$z_2^{(0)} = 0,$$

$$z_2^{(2)} = 0,$$

and again refracting,

$$[c_y^{(1)}]_2 = \frac{1}{6nR^3}[6y_{\mathrm{ep}} R^2(n-1) - 6\theta_y R^3 n],$$

$$[c_y^{(3)}]_2 = \frac{-1}{6R^3 n}[\theta_y^3 R^3 n - 3\theta_y^2 y_{\mathrm{ep}} R^2 n(n-1) - 3y_{\mathrm{ep}}^3 n(n-1) + 3\theta_y y_{\mathrm{ep}}^2 R(2n^2 + n - 1)],$$

$$[c_z^{(0)}]_2 = 1,$$

$$[c_z^{(2)}]_2 = \frac{-1}{2R^2 n^2}([y_{\mathrm{ep}}(n-1) + \theta_y R n]^2).$$

Finally, we transfer to the image surface. Here the only quantity of interest is the ray coordinate y, so we omit the expressions for w and z,

$$y_3^{(1)} = \frac{1}{Rn}[y_{\mathrm{ep}}(-t_2 + nR - nt_1 + nt_2 + t_1 n^2) + \theta_y R(nt_2 + n^2 t_1)],$$

$$y_3^{(3)} = \frac{1}{6R^3 n^3}\{y_{\mathrm{ep}}^3[3t_2(n^4 - 3n^2 + 3n - 1) + 3n^3(n-1)(-R + t_1 + t_1 n(n-1))] + \theta_y y_{\mathrm{ep}}^2[3Rt_2 n(n-1)(2n^2 + 4n - 3) - 3Rn^3(n-1)(R + t_1(-3n^2 + n - 1))] + \theta_y^2 y_{\mathrm{ep}}[3R^2 t_2 n^2(n-1)(n+3) + 3R^2 n^4 t_1(n-1)(3n+1)] + \theta_y^3[2R^3 t_2 n^3 + R^3 t_1 n^4(3n^2 - 1)]\}.$$

This is the analytical expression giving the location of all rays at the image plane in terms of the variables defining the incident ray $(y_{\mathrm{ep}}, \theta_y)$, and of the system parameters (R, t_1, t_2, n), to third-order approximation. In order to use this model to design a lens, we construct a merit function,

$$M = \int\int [y_3(\,) - y_3^{(1)}(\,)]^2 \mathrm{d}y_{\mathrm{ep}} \mathrm{d}\theta_y = \int\int [y_3^{(3)}(\,)]^2 \mathrm{d}y_{\mathrm{ep}} \mathrm{d}\theta_y.$$

At this point, we have not defined any of the system parameters, pupil size, or field of view. Defining the latter two allows us to perform the above integrals, and a typical optical design procedure would involve fixing the values of t_1, t_2, and n—or constraining their ranges—prior to searching for the optimum. (If these three parameters are allowed to vary freely, one can always obtain a zero-aberration solution by letting $n \to \infty$ or $t_2 \to \infty$.) For a $\pm 10°$ field of view and a 32 mm pupil diameter, and fixing the system parameters at $t_1 = 5$ mm, $t_2 = 76.6667$ mm, and $n = 1.5$, we find the optimal radius of curvature to be $R = 41.75$ mm.

Even for such a simplified case, we can see that the expressions are lengthy so that performing design in the analytical domain requires interacting with these functions via computer algebra systems. They do, however, provide a much more informative description of the ray intercepts' nonlinear dependence on each given system parameter. This can create a problem of information *overload*, in contrast to the information uncertainty produced by sampled exact ray tracing.

Note that one needs not expand all surfaces to the same polynomial order of approximation. The only restriction here is that the order of approximation in the ray trace needs to be at least as much as the highest order used for the surfaces within the system.

7. DESIGN EXAMPLES

To illustrate the performance and flexibility of our design approach, we show the design of three example systems and compare to results derived with conventional optical design software. While comparisons between conventional and new techniques developed during research are almost invariably unfair, in that more effort is put into optimizing the latter than the former for the specific cases at hand, what we wish to show is that the approach differs qualitatively from conventional methods and is capable of providing equivalent answers in a wide variety of designs so that it shows promise as a new design tool.

The first example is a variant of the example ray trace performed in Section 6, but this time using the lens off-axis and allowing the two surfaces to be freeform x-y polynomials rather than spherical (see also Fig. 3). The system parameters are $t_1 = 5$ mm, $t_2 = 76.6667$ mm, the refractive index $n = 1.5$, and the rectangular field of view is $5° \le \theta_y \le 7°$ and $-1° \le \theta_x \le +1°$. Thus, the equations for the front and back surfaces of the lens are defined as

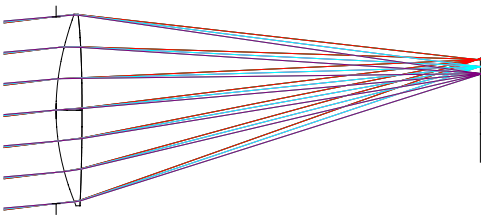

Fig. 3. (Color online) The lens used in the first design example: a freeform x-y polynomial singlet lens with an off-axis field. The system parameters are similar to those of Fig. 2: $t_1 = 5$ mm, $t_2 = 76.6667$ mm, and the refractive index is $n = 1.5$, while the surface parameters are the 14 polynomial coefficients (seven for each surface) out to fourth order. The field of view for this design is $5° \le \theta_y \le 7°$ and $-1° \le \theta_x \le +1°$.

$$z_{\mathrm{front}} = \alpha_1 x^2 + \alpha_2 y^2 + \alpha_3 y^3 + \alpha_4 x^2 y + \alpha_5 x^4 + \alpha_6 y^4 + \alpha_7 x^2 y^2,$$

$$z_{\mathrm{back}} = \beta_1 x^2 + \beta_2 y^2 + \beta_3 y^3 + \beta_4 x^2 y + \beta_5 x^4 + \beta_6 y^4 + \beta_7 x^2 y^2,$$

with the α_s and β_s coefficients left to be determined by the optimizer. Since the system is symmetric about the $x = 0$ plane, all odd-order terms in x are necessarily zero.

In order to compare the results with conventional design approaches, we designed this lens using both the proximate ray trace technique and Zemax [21], with the resulting coefficients shown in Table 2. The figure of merit from each design can be calculated in either the domain used by proximate ray tracing (i.e., an approximate analytical model) or the domain used by Zemax (i.e., an exact sampled model), with the following results:

Merit Function	Design Method	
Domain	Proximate	Zemax
Analytic Approx.	0.00118	0.00205
Sampled Exact	0.07483	0.05635

Note that both merit functions require approximations—truncated order in the case of proximate ray tracing and sampling in the case of Zemax's default method.

The second design example shows a setup appropriate to the design of a lenslet used in a multiscale lens [22]. A multiscale lens involves the use of a standard objective lens combined with a back-end lenslet array used to perform remapping and aberration correction on the image prior to detection. The modeling of these systems can be quite complex since the lenslet elements are designed to have freeform surfaces, and pupil vignetting is both large and varies rapidly with field angle. The number of surfaces present in the system is small, however, making the problem tractable for analytical ray tracing.

Table 2. Surface Parameters Obtained for the First Design Example (an Off-Axis Singlet)[a]

	Proximate	Zemax
α_1	1.188×10^{-2}	1.136×10^{-2}
α_2	1.137×10^{-2}	1.114×10^{-2}
α_3	3.620×10^{-6}	-2.893×10^{-5}
α_4	1.116×10^{-5}	-1.790×10^{-5}
α_5	-3.863×10^{-7}	2.478×10^{-7}
α_6	-1.573×10^{-6}	-1.097×10^{-6}
α_7	1.839×10^{-6}	-8.058×10^{-7}
β_1	-5.649×10^{-4}	-1.124×10^{-3}
β_2	-9.805×10^{-4}	-1.234×10^{-3}
β_3	2.782×10^{-6}	-2.856×10^{-5}
β_4	1.150×10^{-5}	-1.675×10^{-5}
β_5	-1.014×10^{-6}	-1.410×10^{-7}
β_6	-2.030×10^{-6}	-1.400×10^{-6}
β_7	1.034×10^{-6}	-1.467×10^{-6}

[a]The proximate ray trace is performed out to sixth order and optimized with simulated annealing; the Zemax results use damped least-squares optimization.

This example illustrates the design of a single lenslet within the array. The goal of the design is to re-image a section of the initial image plane and also to perform aberration correction so that the re-imaged field contains less blur than the original field. In the case here, the field angles re-imaged by the lenslet are $5.75° < \theta < 8.25°$, the entrance pupil diameter is 8 mm, and the lenslet itself is 4 mm in diameter. For the lenslet shown here, the free-form rear surface is modeled using an x-y polynomial while the front surface is spherical so that the rear surface has the equation

$$z = \alpha_1 x^2 + \alpha_2 y^2 + \alpha_3 y^3 + \alpha_4 x^2 y + \alpha_5 x^4 + \alpha_6 y^4 + \alpha_7 x^2 y^2.$$

The optical layout and prescription for this setup are given in Fig. 4 and Table 3.

Once again performing the proximate ray trace to sixth order and constructing the merit function, the simplicity of this problem allows us to use the general-purpose NMinimize function in Mathematica. Likewise using Zemax's default optimization tool (damped least-squares), we obtain the following results:

Merit Function	Design Method	
Domain	Proximate	Zemax
Analytic Approx.	0.0003428	0.002169
Sampled Exact	0.12713	0.01467

As before, we find that each method is optimal in its own domain, although the two designs are quite similar in shape (see Table 4).

The third design example is a rotationally symmetric nonimaging concentrator, where each of the lens surfaces is even aspheric. The goal here is different from that of the two previous examples in that we are no longer concerned with imaging quality *per se*. Rather, we want to maximize the amount of light we can concentrate onto a given region at the "image plane"—what we call the *concentration region*. Thus, for a given range of incident ray angles, we attempt to maximize the number of rays incident on the concentration region.

Due to symmetry, we can produce an approximate solution by confining the field angles and pupil coordinates to the meridional plane. Thus, we consider a field of view of $-20° \le \theta_y \le +20°$; the entrance pupil diameter is 2.93 mm, the system track length is 3.14 mm, and the image size is $-1\ \text{mm} \le y \le +1$ mm. (This design example is modeled after [[23], pp. 189–192].) The light concentration region is thus 2 mm long, and we wish to maximize the amount of incident light reaching this region. If we confine ourselves to analyzing the system in two dimensions, then the surface equations for this lens are

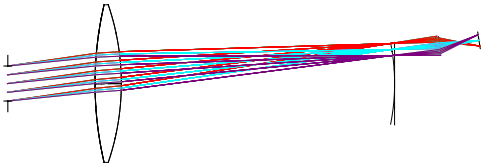

Fig. 4. (Color online) The multiscale lens design example layout and prescription. The objective is fixed, while we attempt to design the lenslet to perform aberration correction on the nominal image (shown by the curved surface between the lenses) and re-image onto a tilted detector array (shown at the far right). The square pupil is 8 mm × 8 mm in size, and the objective lens focal length is 64.45 mm. The lenslet re-images a 2.5° × 2.5° square field of view.

Table 3. Multiscale Lens Design Example Layout and Prescription

	s	Radius of Curvature (mm)	Thickness (mm)	Index
Pupil	0	∞	20	
Obj. lens front	1	79.60	6	1.5168
Obj. lens back	2	−55.80	72.02	
First image	3	∞	62.44	
Coord. break: rotate about x axis by 4.54°				
Lenslet front	4	7.88	1	1.8
Lenslet back	5	∞	9	
Final image	6	∞		

$$z_{\text{front}} = \alpha_1 y^2 + \alpha_2 y^4 + \alpha_3 y^6,$$

$$z_{\text{rear}} = \beta_1 y^2 + \beta_2 y^4 + \beta_3 y^6.$$

A naive attempt at constructing a merit function for this problem would be something like

$$M = \int_{-D_{\text{ep}}/2}^{D_{\text{ep}}/2} \mathrm{d}y_{\text{ep}} \int_{-20°}^{20°} \mathrm{d}\theta_y y^2,$$

where y gives the position of the ray at the image plane and D_{ep} is the diameter of the entrance pupil. While this function penalizes rays which stray too far from the axis, what we really want is a penalty which is zero or very small for rays falling onto the concentration region, but very large for rays falling outside it. The implementation of this approach is tricky, however, as it requires ℓ_1 minimization rather than the much more widely used ℓ_2 minimization techniques, and also it requires that we work directly with image coordinates as primary variables, rather than the object coordinates we have been using up to now. This is a topic we hope to treat at length in a future publication.

An alternative optimization approach takes advantage of the edge ray principle [[23], p. 183]. The phase space for rays propagating through the system (Figs. 5 and 6) are bounded by the square $abcd$. We can choose to map those

Table 4. Surface Parameters Obtained for the Second Design Example (Multiscale Lenslet)[a]

	Proximate	Zemax
α_1	−0.070 26	−0.066 25
α_2	−0.068 24	−0.064 25
α_3	0.000 08	0.000 19
α_4	0.000 02	0.000 17
α_5	0.002 35	0.001 38
α_6	0.002 24	0.001 31
α_7	0.004 02	0.002 38

[a]The proximate ray trace is performed out to sixth order and optimized with Mathematica's NMinimize function; the Zemax results use damped least-squares optimization.

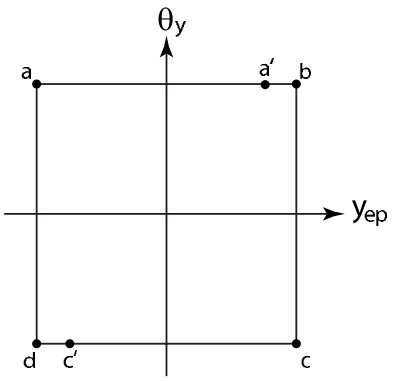

Fig. 5. Phase space of the incident rays.

rays represented by line aa' to the edge of the concentration region—a point we designate as y_m—and map the rays represented by line $a'b$ to the edge of the back surface of the lens—a point we designate as (y_k,z_k). By symmetry, this likewise forces all the rays represented by line cc' to focus near $-y_m$, and all the rays represented by line $c'd$ to focus near $(-y_k,z_k)$. The corresponding merit function can be given the form

$$M = \left[\int_{(D_{ep}/2)-y_{a'}}^{D_{ep}/2} [(y_3 - y_k)^2 + (z_3 - z_k)^2] dy_{ep} + \int_{-D_{ep}/2}^{(D_{ep}/2)-y_{a'}} (y - y_m)^2 dy_{ep} \right]_{\theta=20°},$$

where $y_{a'}$ is the y_{ep} value of ray a'. (The value of $y_{a'}$ is determined during the optimization step.) The coordinate (y_3,z_3) is the ray position on the back surface of the lens—surface 3 in the system. The reason for choosing this merit function is as follows: when the incident ray changes continuously in phase space along the phase-space boundary from point a to b, c, d, and back to a, the corresponding ray at the image plane will also change continuously in phase space and form a closed loop. According to the edge ray principle all the incident rays within the rectangular region $abcd$ in phase space will fall into the closed loop in the phase space of the rays at the image plane, bounded at the image plane within the range $-y_m$ to y_m.

We choose the line aa' to reach the y_m point at the image plane, and also the line $a'b$ to hit the upper edge of the lens back surface. Since a' point has the smallest y_{ep} value among those rays within the phase-space line $a'b$, it should also have the largest ray angle when hitting the upper edge of the back surface so that its ray angle after refraction by the back surface will also be largest. Thus, it will reach the image plane with the largest y coordinate, y_m, and all other rays along the line $a'b$ in phase space will reach the image plane below y_m. Since the ray from point c in phase space hits the image plane at $-y_m$, all rays from the phase-space line bc should reach the image plane inside the concentration region $(-y_m \le y \le y_m)$. And, due to symmetry, the phase-space lines cd and da will fall inside the concentration region as well. Note that this approach is not exact: it is possible to generate a surface which violates these assumptions. Moreover, while we can use this one-dimensional design approach to generate a two-dimensional (2D) surface by rotating the design surface about the axis, the analysis above has ignored skew rays within the system. Nevertheless, we can obtain a useful design using the above approach, and upon optimizing M with the proximate ray trace equations for y and y_3, we obtain the design,

$$\alpha_1 = 0.297\ 73, \quad \beta_1 = -0.038\ 88,$$

$$\alpha_2 = 0.018\ 92, \quad \beta_2 = 0.023\ 36,$$

$$\alpha_3 = 0.001\ 71, \quad \beta_3 = 0.002\ 53,$$

The resulting lens is shown in Fig. 7 together with a diagram illustrating the ray mapping. Figure 8 shows the resulting concentration performance, giving the transmission (portion of rays reaching the concentration region) as a function of the incidence angle.

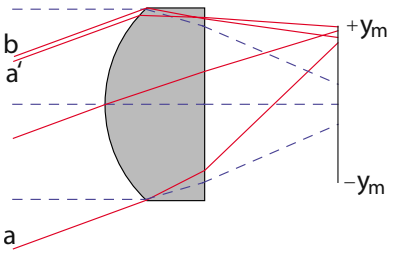

Fig. 6. (Color online) The approximate ray trace used by the edge ray principle for designing a concentrator lens. The labeled rays a, a', and b are the phase-space points given in Fig. 5. Ray a' maps to the edge of the concentration region, at a distance y_m from the optical axis, while ray b is bent downward so that it reaches the image plane at $y < y_m$.

8. CONCLUSION

The first implementation of proximate ray tracing, by Hopkins in 1976 [6–8], appears to have been done as a method of reducing the computational burden and complexity for calculating higher-order aberrations. By sampling a specific set of rays passing through the system, one is able to obtain each of the various aberration coefficients. Our own implementation adapts the proximate ray tracing concept for use in computer algebra systems in order to perform the entire procedure in the analytical domain. This is an important advance in that the analytical formulas provide information of a qualitatively different character than that provided by conventional ray tracing. Some examples of these advantages include: (1) the aberration terms can be simply picked out of the final expression for the optical path length $W = \Sigma_s n_s w_s$ as a function of the incidence angle (H_x,H_y) and pupil coordinates (x_{ep},y_{ep}); (2) the optimization procedure can take advantage of the properties of well-behaved functions (e.g., polynomials) such as their infinite differentiability; (3) there is no need to consider sampling density or similar issues required for discrete ray tracing; and (4) the analytical functional form more clearly shows some of the difficulties that ray tracing can encounter, such as the presence of singularities [25] that can affect the convergence of aber-

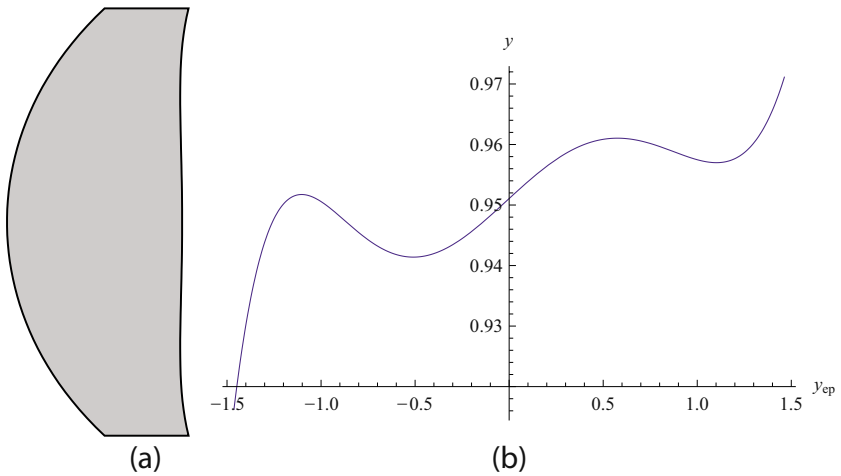

Fig. 7. (Color online) (a) Layout of the concentrator lens obtained via the edge ray principle design method; (b) the profile of ray position y at the image plane as a function of entrance pupil position y_{ep} for $\theta=20°$.

ration series. We plan to present a more thorough discussion of these properties in a subsequent publication.

Each of the advantages stated above can also be given for the matrix approaches to analytical ray tracing [10–13]. The difference between the implementation presented here and the matrix methods is that our approach is more easily adaptable to general-purpose use by having the computer algebra program perform much of the generic analytical calculations, such as obtaining the surface normal, converting coordinate systems, and developing systems of equations out to arbitrary order. In addition, while the research presented here considers only monochromatic systems, we are currently adding wavelength (or wavenumber, depending on the choice of coordinate) as an additional primary variable and developing our code for use in multiwavelength systems. This is an essential additional step for optical system design which will be new to analytical ray tracing.

The three design examples presented in Section 7 illustrate the flexibility of the analytical approach to treat systems of arbitrary symmetry and to produce accurate designs. In all three cases the optimization process in both the proximate and numerical ray trace designs does not require interaction with the designer after defining the first-order properties of the system.

The drawbacks to analytical ray tracing include (1) for all but extreme cases, analytical ray tracing will be much slower than numerical exact ray tracing; (2) the analytical formulas produced by the method can be quite lengthy; and (3) analytical ray formulas are harder to interpret due to their unfamiliarity. While each of these drawbacks is important, they reflect the trade-off of the new information obtained about a given design.

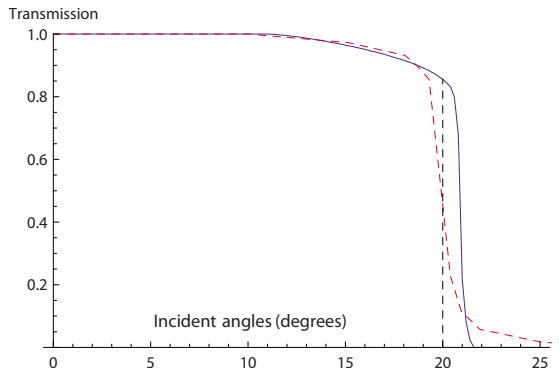

Fig. 8. (Color online) Concentrator example's performance is illustrated by showing the portion of transmitted rays reaching the concentration region as a function of incidence angle. The example shown here (solid line) compares well with that shown in Fig. 8.9 of [24] (dashed line). A vertical line at 20° illustrates the maximum angle of incidence used in the design.

APPENDIX A

Due to the length of the polynomial expressions which need to be manipulated for asymmetric systems and high-order approximations, an important feature of an analytical ray tracing engine is an efficient means of performing each required operation. While [13] presents a code for performing an order-expansion operation, the algorithm used is quite slow for large expressions. An alternative approach which is simple to apply for even very large expressions is the following. For each of the order-expanded variables present in an expression, we can represent the sum as a vector where each element of the vector represents the appropriate order of expansion. Thus, for example, inside the computer algebra system we can represent the variable y as the vector

$$\mathbf{y} = (y^{(0)}, y^{(1)}, y^{(2)}, y^{(3)}, \ldots)^{\mathsf{T}}, \tag{A1}$$

so that the multiplication operation between variables x and y, for example, is computed via vector multiplications as

$$\mathbf{1}^{\mathsf{T}}\mathbf{x}\mathbf{y}^{\mathsf{T}}\mathbf{1},$$

where $\mathbf{1}$ is the vector of appropriate length containing 1's for each element. Here the inner operation, $\mathbf{x}\mathbf{y}^{\mathsf{T}}$, generates a matrix containing all combinations of elements of x with elements of y, i.e.,

$$\mathbf{xy}^{\mathsf{T}} = \begin{pmatrix} x^{(0)}y^{(0)} & x^{(1)}y^{(0)} & x^{(2)}y^{(0)} & \cdots \\ x^{(0)}y^{(1)} & x^{(1)}y^{(1)} & x^{(2)}y^{(1)} & \cdots \\ x^{(0)}y^{(2)} & x^{(1)}y^{(2)} & x^{(2)}y^{(2)} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}.$$

The outer operations perform the summation over all matrix elements. As given here, a computer algebra system needs to check the power of each of the variables used in the expressions and add the powers of all variables within a given term, in order to obtain the order of a given term. It is possible to achieve this in a single step by adding an indicator variable, which we name g. The vector expression for g is given as

$$\mathbf{g} = (1, g, g^2, g^3, \ldots)^{\mathsf{T}}.$$

When we define the vector representation of each variable to be used, instead of Eq. (A1) we use

$$\mathbf{y} \cdot \mathbf{g} = (y^{(0)}, gy^{(1)}, g^2y^{(2)}, g^3y^{(3)}, \ldots),$$

where the "·" operator represents an element-by-element multiplication. The indicator vector $\mathbf{g}$ is likewise applied to all variables used in expressions applying order-expansion methods. When operations such as the x-y multiplication shown above are performed, one now needs only to locate the power of g in order to obtain the order of a given term.

This ability is even more convenient for setting up and solving systems of equations such as those for transfer or refraction. For example, when calculating the expansion of the square root, as in Section 4, instead of manually setting up the matrix of equations and solving them at each nonlinear operation, one needs only to ask the computer algebra system to perform the Taylor expansion of the equation in the variable g. While not optimal in terms of the computational speed, this method greatly simplifies the code, for improved readability and comprehension.

REFERENCES

1. D. Shafer, "Global optimization in optical design," Comput. Phys. **8**, 188–195 (1994).
2. A. E. W. Jones and G. W. Forbes, "An adaptive simulated annealing algorithm for global optimization over continuous variables," J. Global Optim. **6**, 1–37 (1995).
3. S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," Science **220**, 671–680 (1983).
4. X. Chen and K. Yamamoto, "An experiment in genetic optimization in lens design," J. Mod. Opt. **44**, 1693–1702 (1997).
5. R. Horst, P. M. Pardalos, and N. V. Thoai, *Introduction to Global Optimization*, 2nd ed. (Springer, 2000).
6. G. W. Hopkins, "Proximate ray tracing and optical aberration coefficients," J. Opt. Soc. Am. **66**, 405–410 (1976).
7. G. W. Hopkins, "Proximate ray tracing and wave aberration coefficients," J. Opt. Soc. Am. **66**, 942–949 (1976).
8. G. W. Hopkins, "Aberrational analysis of optical systems: a proximate ray trace approach," Ph.D. thesis (University of Arizona, 1976).
9. J. von zur Gathen and J. Gerhard, *Modern Computer Algebra* (Cambridge U. Press, 1999).
10. M. Kondo and Y. Takeuchi, "Matrix method for nonlinear transformation and its application to an optical lens system," J. Opt. Soc. Am. A **13**, 71–89 (1996).
11. V. Lakshminarayanan and S. Varadharajan, "Expressions for aberrations coefficients using nonlinear transforms," Optom. Vision Sci. **74**, 676–686 (1997).
12. J. B. Almeida, "General method for the determination of matrix coefficients for high-order optical system modeling," J. Opt. Soc. Am. A **16**, 596–601 (1999).
13. J. B. Almeida, "Programming matrix optics into Mathematica," Optik (Stuttgart) **116**, 270–276 (2005).
14. A. Walther, "Eikonal theory and computer algebra," J. Opt. Soc. Am. A **13**, 523–531 (1996).
15. A. Walther, "Eikonal theory and computer algebra II," J. Opt. Soc. Am. A **13**, 1763–1765 (1996).
16. T. Kryszczyński, "First steps towards an algebraic method of the optical design in the range of all aberration orders," in *11th Slovak-Czech-Polish Optical Conference on Wave and Quantum Aspects of Contemporary Optics* (1999), pp. 336–342.
17. M. Born and E. Wolf, *Principles of Optics*, 7th ed. (Cambridge U. Press, 1999).
18. G. W. Forbes, "Optical system assessment for design: numerical ray tracing in the Gaussian pupil," J. Opt. Soc. Am. A **5**, 1943–1956 (1988).
19. V. Y. Pan, "Solving a polynomial equation: some history and recent progress," SIAM Rev. **39**, 187–220 (1997).
20. Wolfram Research, Inc., www.wolfram.com.
21. ZEMAX Development Corp., www.zemax.com.
22. D. J. Brady and N. Hagen, "Multiscale lens design," Opt. Express **17**, 10,659–10,673 (2009).
23. N. Shatz and J. C. Bortz, "Global optimization of high-performance concentrators," in *Nonimaging Optics* (Elsevier, 2005), Chap. 11, pp. 265–304.
24. R. Winston, J. C. Miñano, and P. Benítez, *Nonimaging Optics* (Elsevier, 2005).
25. G. W. Forbes, "Extension of the convergence of multivariate aberration series," J. Opt. Soc. Am. A **3**, 1376–1383 (1986).