

Scilab Routines for Calculating Optical Ray Parameters of Thick Multi-Lens Systems

Fulvio Andres Callegari*

Centro de Engenharia, Modelagem, e Ciências Sociais Aplicadas, Universidade Federal do ABC, Santo André, SP, Brasil.
 Avenida dos Estados, 5001 - Bairro Santa Terezinha Santo André - CEP: 09210-580

*Corresponding author: fulvio.callegari@ufabc.edu.br

Received May 06, 2023; Revised June 30, 2023; Accepted July 10, 2023

Abstract It is presented a code using a free software in order to calculate optical parameters for an arbitrary system consisting in N thick lenses submerged in optical media with arbitrary refraction index. This code calculates focus, front and back focal distances, principal planes and image distances for such multi-lens systems. The software was verified by solving numeric examples extracted from well-established optics literature.

Keywords: Matrix Optics, Open-Source Algorithm, Multi-lenses System, Thick Lenses, Back Focal length, Front Focal Length, Principal Planes, Focus

Cite This Article: Fulvio Andres Callegari, “Scilab Routines for Calculating Optical Ray Parameters of Thick Multi-Lens Systems.” *International Journal of Physics*, vol. 11, no. 3 (2023): 153-157. doi: 10.12691/ijp-11-3-4.

1. Introduction

Resolving optical multi-lenses systems can be very laborious if doing by hand. There exists commercial software up to the task and of professional level to perform such assignment [1,2]. They have their correspondent educational free versions, but their code is obviously hidden, and in some cases are far from been user friendly, especially for the beginner optic student. In this paper, as a way of complementing this picture, it is presented an algorithm using an open-source software, Scilab [3], which can be used to calculate some basic optical parameters such as focus, principal planes, front and back focal distances and the image distance (given the object distance), for an optical system composed by ‘N’ spherical thick lenses in cascade merged in N+1 optical media with arbitrary values of index of refraction. This algorithm was verified by solving several numeric examples from publications and text books.

The main objective of this work is to present to the community, a free source code that eventually could be improved in order to perform also more sophisticated task, such as finding pupils, windows, field of view, estimating local of apertures, stops, and the likes. The algorithm that it is presented works for complex optical systems from two thick lenses onwards, and also can be used for the most idealized case of thin lenses.

2. The Optical System

The routine was implemented using Scilab and is based on the optical matrix method, assuming paraxial

approximation, whose complete formalism is described in detail in [4]. The aim is to resolve an optical system as shown in the Figure 1:

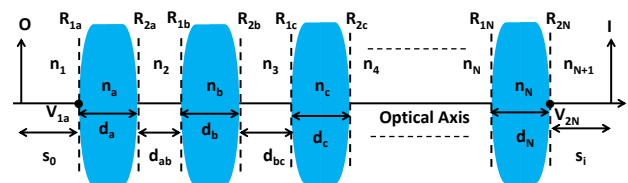


Figure 1. Multi-lenses optical system

The optical system consists in an arbitrary number of ‘N’ spherical thick lenses (being $N > 1$), each one with its glass of specified indices of refraction, $n_a, n_b, \dots, n_N$, thicknesses, $d_a, d_b, \dots, d_N$, and radius, $R_{1a}, R_{2a}, \dots, R_{1N}, R_{2N}$, immersed in optical media with dissimilar indices of refraction, $n_1, n_2, \dots, n_{N+1}$, and separated by distances $d_{ab}, d_{bc}, \dots, d_{N-1,N}$. The horizontal line crossing the center of each lens is the optical axis. The object distance, s_o , is measured with respect to the vertex point ‘ V_{1a} ’, which is the intersection of the first surface of the first lens with the optical axis. The image distance, s_i , is measured with respect to the vertex point ‘ V_{2N} ’, which is the intersection of the second surface of the last lens with the optical axis. Regarding signal convention; object distances will be positive (negative) if the object is to the left (right) of ‘ V_1 ’, image distances will be positive (negative) if the image is to the right (left) of ‘ V_{2N} ’. Concerning the curvature radii of surfaces, they will be positive (negative) if the corresponding surface was convex (concave). This convention is usually adopted in most of the specialized literature.

The idea is to calculate, for the complex system shown in Figure 1, the focal distance, f (also known as effective

focal distance in some texts); the front focal distance, z_f , that is the distance, measure from ' V_{1a} ' on the optical axis, in which an object point (or a source of spherical divergent wave fronts), must be located in order that the optical system produces a bunch of parallel rays after refraction (see Figure 2 the red rays); back focal distance, z_b , which is the distance, measured from ' V_{2N} ', to which the optical system focalizes a bunch of optical ray incident parallel to the optical axis (see the blue rays Figure 2).

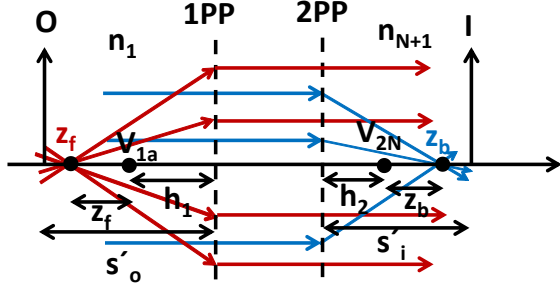


Figure 2. Cardinal points

Front focal point, z_f , related to front focal distance obeys the following signal convention; if positive, it should be located to the left of ' V_{1a} ', if negative, it should be localized to the right of ' V_{1a} '. For back focal point, z_b , the following signal convention should be applied; if positive (negative), it should be localized to the right (left) of ' V_{2N} '.

Also, there will be calculated the two principal planes, 1PP and 2PP, whose distances, h_1 and h_2 are measured with respect to the already defined outermost vertex points of the system, ' V_{1a} ' and ' V_{2N} '. Distance h_1 obeys the following signal convention: if positive (negative), it is located to the right (left) of ' V_{1a} '. For h_2 being positive (negative), h_2 must be located to the right (left) of ' V_{2N} '. These planes are useful in order to simplify the object-image equation for such a complicated system. Also are shown in Figure 2, the quantities s_o' and s_i' , which are object and image distances with respect to the principal planes, respectively. These auxiliary distances are actually those that allow on to describe the optical system through a simple Gaussian type equation. For more details about these cardinal points and the complete analysis of the optical system, the interested reader is referred to [4].

3. The Software

In this section is presented the code used to perform the calculations. The first lines consist in the input data for the system:

```
clear
clc
3. n=list(n1,n2,n3,...,nN+1); // refraction index of
media before, between and after the lenses
4. R=list(R1a, R2a, R1b, R2b,..., R2 N); // Curvature
radii of lenses
5. na=list(na, nb,... ,nN); //refraction index of the
lenses
6. da=list(da, db,... ,dN); //Thickness of lenses
```

```
7. dab=list(dab, dbc,... ,dN-1N); //Distances between
the lenses
8. so = so; //object distance.
```

All data, except the object distance, are gathered using the "list" command. The line 'n' corresponds to the index of refraction of the media surrounding all lenses in order, i.e., the first, ' n_1 ', corresponds to the first index of refraction medium before (to the left of) the first lens, the second, ' n_2 ', corresponds to the index of refraction medium between first and second lenses, etc., until the last one, ' n_{N+1} ', (being N the total number of lenses), corresponds to the index of refraction of the optical medium after (to the right of) the last lens. The fourth line, 'R', gathers the radii of curvature of the lenses. The subindex '1a', stands for the radius of the first surface of the first lens, subindex '2b' for the radius of the second surface of the second lens, etc. In case there is a plane surface for some lens, the number '1e20' will suffice as the value of an "infinite" radius. The line ' n_a ' gathers the index of refraction values for each of the lenses compounding the system. The ' d_a ' line gathers the thickness of each lens and finally, the ' d_{ab} ' line should be filled with the correspondent distances between the lenses.

The next lines show the calculation for the specific case of two lenses:

```
9. if length(dab) == 1 & dab(1) == 0 then
10. nright(1) = na(2);
11. nleft(2) = na(1);
12. nleft(1) = n(1);
13. nright(2) = n(3);
14. for j=1:length(R)/2;
15. gen11(j)=1+(na(j)-nright(j))*da(j)/(na(j)*R(2*j))
16. gen12(j)=-(na(j)-nleft(j))/R(2*j-1)+(na(j)-
nright(j))/R(2*j)-(da(j)/na(j))*((na(j)-
nleft(j))/R(2*j-1))*((na(j)-nright(j))/R(2*j))
17. gen21(j)= da(j)/(na(j))
18. gen22(j)= 1-(na(j)-nleft(j))*da(j)/(na(j)*R(2*j-1))
19. end
20. M11(1)=
gen11(2)*gen11(1)+gen12(2)*((dab(1)/n(2))*gen
11(1)+gen21(1)); //matrix elements for the two
lenses
21. M12(1)=
gen11(2)*gen12(1)+gen12(2)*((dab(1)/n(2))*gen
12(1)+gen22(1));
22. M21(1)=
gen21(2)*gen11(1)+gen22(2)*((dab(1)/n(2))*gen
11(1)+gen21(1));
23. M22(1)=
gen21(2)*gen12(1)+gen22(2)*((dab(1)/n(2))*gen
12(1)+gen22(1));
24. f=-1/M12; // focal distance
25. zf=n(1)*f*M11; //front focal distance
26. zb=n(size(n))*f*M22; // back focal distance
27. h1=n(1)*f*(1-M11); // 1PP distance
28. h2=n(size(n))*f*(M22-1); // 2PP distance
29. soprime=so+h1; // so'
30. siprime=(n(3)*(-1/M12)*soprime)*(1/(soprime-
n(1)*(-1/M12))); // si'
31. si=siprime+h2; // image distance
32. elseif length(dab) == 1 & dab(1) > 0 then
33. nright(1) = n(2);
```

```

34. nleft(2) = n(2);
35. nleft(1) = n(1);
36. nright(2) = n(3);
37. for j=1:length(R)/2;
38. gen11(j)=1+(na(j)-nright(j))*da(j)/(na(j)*R(2*j))
39. gen12(j)=-(na(j)-nleft(j))/R(2*j-1)+(na(j)-
nright(j))/R(2*j)-(da(j)/na(j))*((na(j)-
nleft(j))/R(2*j-1))*((na(j)-nright(j))/R(2*j))
40. gen21(j)= da(j)/(na(j))
41. gen22(j)= 1-(na(j)-nleft(j))*da(j)/(na(j)*R(2*j-1))
42. end
43. M11(1)=
gen11(2)*gen11(1)+gen12(2)*((dab(1)/n(2))*gen
11(1)+gen21(1)); //matrix elements for the two
lenses
44. M12(1)=
gen11(2)*gen12(1)+gen12(2)*((dab(1)/n(2))*gen
12(1)+gen22(1));
45. M21(1)=
gen21(2)*gen11(1)+gen22(2)*((dab(1)/n(2))*gen
11(1)+gen21(1));
46. M22(1)=
gen21(2)*gen12(1)+gen22(2)*((dab(1)/n(2))*gen
12(1)+gen22(1));
47. f=-1/M12;
48. zf=n(1)*f*M11;
49. zb=n(size(n))*f*M22;
50. h1=n(1)*f*(1-M11);
51. h2=n(size(n))*f*(M22-1);
52. soproime=so+h1;
53. siprime=(n(3)*(-1/M12)*soproime)*(1/(soproime-
n(1)*(-1/M12)));
54. si =soprime+h2; // image distance

```

The variables “gen_{ij}” in lines 15-18 are the matrix elements for the individual lenses, for which, in addition to the index of refraction of the lenses themselves, the index of refraction of the media surrounding them, and eventually those of the adjacent lenses (in case the distances between them are null), are used. For that case, the lines 10-13 define the index accordingly. Then, in lines 15-18 the matrix elements are effectively calculated. After that, in lines 20-23 the matrix elements M_{ij} for the two lenses systems are calculated. Finally in lines 24-31 the cardinal points and the image distance are calculated.

The lines 32-54 make the same calculation for the two lenses systems when the distance between them is not null.

The following lines perform the calculation for optical systems with more than two lenses:

```

55. elseif length(dab)>= 2 then
56. for i = 1:length(dab)+1
57. nright(i) = n(i+1);
58. end
59. for i = 1:length(dab)
60. if (dab(i) == 0) then
61. nright(i) = na(i+1);
62. end
63. end
64. for i = 1:length(dab)+1
65. nleft(i) = n(i);
66. end

```

```

67. for i = 1:length(dab)
68. if (dab(i) == 0) then
69. nleft(i+1) = na(i);
70. end
71. for j=1:length(R)/2;
72. gen11(j)=1+(na(j)-nright(j))*da(j)/(na(j)*R(2*j))
73. gen12(j)=-(na(j)-nleft(j))/R(2*j-1)+(na(j)-
nright(j))/R(2*j)-(da(j)/na(j))*((na(j)-
nleft(j))/R(2*j-1))*((na(j)-nright(j))/R(2*j))
74. gen21(j)= da(j)/(na(j))
75. gen22(j)= 1-(na(j)-nleft(j))*da(j)/(na(j)*R(2*j-1))
76. end
77. M11(1)=
gen11(2)*gen11(1)+gen12(2)*((dab(1)/n(2))*gen
11(1)+gen21(1)); //matrix elements for the first
two lenses
78. M12(1)=
gen11(2)*gen12(1)+gen12(2)*((dab(1)/n(2))*gen
12(1)+gen22(1));
79. M21(1)=
gen21(2)*gen11(1)+gen22(2)*((dab(1)/n(2))*gen
11(1)+gen21(1));
80. M22(1)=
gen21(2)*gen12(1)+gen22(2)*((dab(1)/n(2))*gen
12(1)+gen22(1));
81. for k=2:length(na)-1
82. M11(k)=gen11(k+1)*M11(k-
1)+gen12(k+1)*((dab(k)/n(k+1))*M11(k-
1)+M21(k-1)); //matrix elements for three lenses
and beyond
83. M12(k)=gen11(k+1)*M12(k-
1)+gen12(k+1)*((dab(k)/n(k+1))*M12(k-
1)+M22(k-1))
84. M21(k)=gen21(k+1)*M11(k-
1)+gen22(k+1)*((dab(k)/n(k+1))*M11(k-
1)+M21(k-1))
85. M22(k)=gen21(k+1)*M12(k-
1)+gen22(k+1)*((dab(k)/n(k+1))*M12(k-
1)+M22(k-1))
86. end
87. end
88. f=-1/M12(max(k));
89. zf=n(1)*f*M11(max(k));
90. zb=n(size(n))*f*M22(max(k));
91. h1=n(1)*f*(1-M11(max(k)));
92. h2=n(size(n))*f*(M22(max(k))-1);
93. soproime=so+h1;
94. siprime=(n(max(i))*(-
1/M12(max(k)))*soproime)*(1/(soproime-n(1)*(-
1/M12(max(k))));
95. si =soprime+h2; // image distance

```

end

4. Results

The code was tested using three examples solved in the literature.

4.1. Example 1

The first of them is the Tessar lens shown in [5], pages 250-251. The input data are filled as follows:

```
n=list(1,1,1,1,1,1); //refraction index of media before,
between and after the lenses
R=list(1.628,-27.57,-3.457,1.582,1e^20,1.920,1.920,-
2.400); //Curvature radii of lenses
na=list(1.6116,1.6053,1.5123,1.6116); //refraction index
of the lenses
da=list(0.357,0.081,0.217,0.396); //Thickness of lenses
dab=list(0.189,0.325,0.001); //Distances between the
lenses
```

In the last line, i.e., distance between lenses, in order to reproduce the results it was necessary to simulate a thin air layer between the last two lenses, which seems reasonable from the figure of the Tessar shown in the aforementioned reference. The results obtained once the code was run (that can be accessed via variable navigator on Scilab Menu), are:

Table 1. Results for Tessar lens example (this work)

Parameter	Value
M_{11}	[1.24; 1.52; 0.848]
M_{12}	[0.0687; 0.299; -0.197]
M_{21}	[0.463; 1.05; 1.34]
M_{22}	[0.829; 0.862; 0.868]
f	5.07
h_1	0.771
h_2	-0.671
z_b	4.4
z_f	4.3

The first four lines show the successive values of the M_{ij} matrices elements for the system analyzed in this example, calculated iteratively as shown by the lines '77-80' of the code, for the two first lenses, and then by lines 81-85, for the rest of the system. The last figure in each line is the corresponding matrix element of the system, and the optical matrix corresponding to the Tessar lens is:

$$M = \begin{pmatrix} 0.848 & -0.197 \\ 1.34 & 0.868 \end{pmatrix},$$

which is in very good agreement to the result presented in [5]. Also, the values for the focal distance, $f = 5.07$ and the location of principal planes, $h_1 = 0.771$ and $h_2 = -0.671$, are in very good agreement with the results of that book. The units of the cardinal points are the same units of length of radii of curvature of the lens's surfaces, which, of course, should be the same as that of the lens thicknesses.

4.2. Example 2

The next example is from reference [6], where several complex geometric optics systems are analyzed. The first of them presented here is the numerical example 2 of such reference, where a two lenses system is evaluated. The data input lines for such example are:

```
n=list(1,1,1); //refraction index of media before, between
and after the lenses
R=list(7,-7,5,-5); //Curvature radii of lenses
na=list(1.5,1.5); //refraction index of the lenses
da=list(1,1); //Thickness of lenses
dab=list(10); //Distances between the lenses
```

In the next table the results obtained for this example in the reference are shown and compared with the values obtained using the software:

Table 2. Comparison of results Example 2 of [6]

Calculated parameters from [6]	Calculated parameters using code (this work)
$f_e = 22.38578771$	$f = 22.40000000$
$s_4 = -11.3197962$	$z_b = -11.30000000$
$s'_4 = -24.205075$	$z_f = -24.20000000$
$\beta_1 = 46.590861$	$h_f = 46.60000000$
$\beta_2 = -33.70558391$	$h_2 = -33.70000000$

Note that nomenclature for the optical parameters is different, but the physical meaning is the same.

4.3. Example 3

The next case corresponds to the numerical example 4 of the same reference, which consists in a five thick lenses system in between several optical index media, some of them different from air. The input data lines that reproduce such example are:

```
n=list(1.2,1.3,1.2,1.3,1,1); //refraction index of medium
before, between and after the lenses
R=list(15,-25,20,1e20,-20,10,10,-10,1e20,-5); //Curvature
radii of lenses
na=list(1.7,1.6,1.8,1.5,2); //refraction index of the lenses
da=list(1,1,1,0.6,0.4); //Thickness of lenses
dab=list(2,2,0.5,0.2); //distances between the lenses
```

In Table 3 a comparative analysis between the results is shown.

Table 3. Comparison of results Example 4 of [6]

Calculated parameters from [6] (cm)	Calculated parameters using code (this work) (cm)
$f_e = 5.100590992$	$f = 5.100000000$
$s_{10} = 3.61999189$	$z_b = 3.620000000$
$s'_{10} = -2.00978487$	$z_f = -2.010000000$
$\beta_1 = 8.13049406$	$h_f = 8.130000000$
$\beta_2 = -1.48059907$	$h_2 = -1.480000000$

Once more the agreement is excellent.

In the examples verified, there has not been used the line corresponding to object distance (in order to calculate image distance). This feature was corroborated using other examples. Also, the software can be used for the

idealized case of thin lenses, writing simply $d_a = 0$, in the corresponding lenses.

5. Conclusion

In this work, it was presented a free code that calculates basic optical parameters such as focal distance, back and front focal distances, and principal planes, for a complex optical system composed of an arbitrary (minimum two) number of thick lenses surrounded by arbitrary refraction index's optical media. Also, it can be used for calculation of image distance, given the object distance for this system. In addition, idealized cases of multiple thin lenses can also be studied.

References

- [1] <https://lambdare.com/oslo/>. [Accessed June. 04, 2023]
- [2] <https://www.zemax.com/>. [Accessed June. 04, 2023]
- [3] <https://www.scilab.org/>. [Accessed June. 04, 2023]
- [4] Fulvio Andres Callegari. Thick Lenses Systems between Arbitrary Index of Refraction Optical Media. A Simple Optical Model for the Human Eye. *International Journal of Physics*, Vol. 9, No. 6, 2021, pp 259-268. Available: <http://pubs.sciepub.com/ijp/9/6/1>. [Accessed June. 04, 2023]
- [5] Hecht, E. *Optics*, 4ed, Addison Wesley, San Francisco, 2002, 250-251.
- [6] Hassan A. Elagha, "Determination of the effective focal length and cardinal points for a coaxial system of thick lenses," *Appl. Opt.* 61, 6778-6786 (2022).



© The Author(s) 2023. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).