DOING PHYSICS WITH PYTHON

COMPUTATIONAL OPTICS

RAY (GEOMETRIC) OPTICS

MATRIX METHODS IN PARAXIAL OPTICS

Ian Cooper

Please email me any corrections, comments, suggestions or additions: **matlabvisualphysics@gmail.com**

DOWNLOAD DIRECTORIES FOR PYTHON CODE

[Google drive]

[GitHub]

S003C.py
Functions to commute the most commonly used transformation matrices.

S003B.py
Transformation matrices: Propagation into a long cylinder with spherical end.
Translation 0 → 1; refraction 1→ 2; translation 2 → 3

MATRIX METHODS FOR RAY PROPAGATION

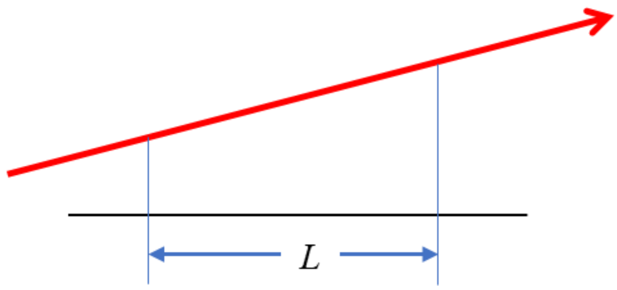
RAY-TRANSFER MATRICES

We can define matrixes for translation, reflection at plane surfaces, reflection at spherical surfaces, refraction at plane surfaces, and refraction at spherical surfaces. By combining appropriate individual matrices in the proper order, it is possible to express any optical system by a 2x2 matrix, which we call the **system matrix**. Table 1 gives a quick summary of some simple ray-transfer matrices.

Table 1 Simple ray-transfer matrices

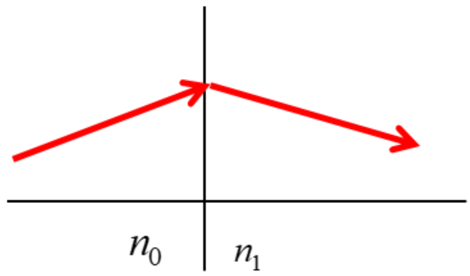
Translation matrix

$$M = \begin{pmatrix} 1 & L \\ 0 & 1 \end{pmatrix}$$



Refraction matrix
plane interface

$$M = \begin{pmatrix} 1 & 0 \\ 0 & n_0 / n_1 \end{pmatrix}$$

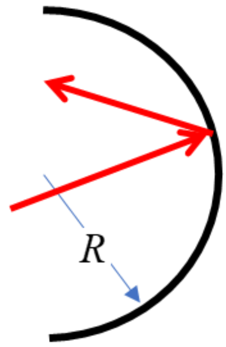


Reflection matrix
spherical mirror

$$M = \begin{pmatrix} 1 & 0 \\ 2/R & 1 \end{pmatrix}$$

convex: $R > 0$

concave $R < 0$

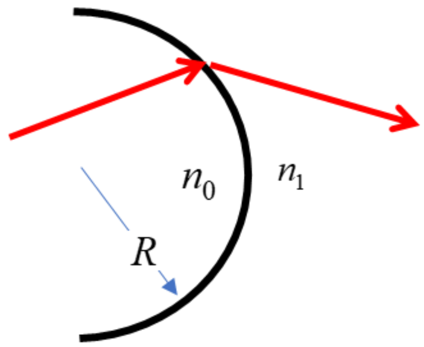


Refraction matrix
spherical interface

$$M = \begin{pmatrix} 1 & 0 \\ \dfrac{n_0 - n_1}{n_1 R} & \dfrac{n_0}{n_1} \end{pmatrix}$$

convex: $R > 0$

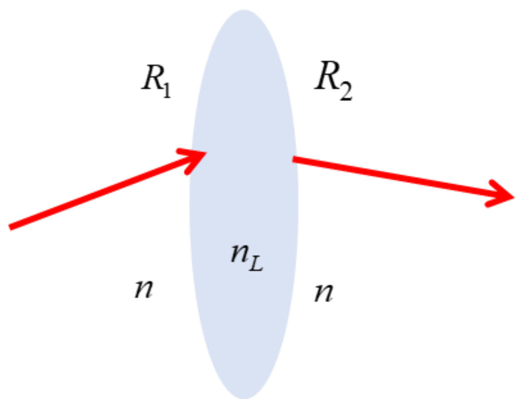
concave $R < 0$



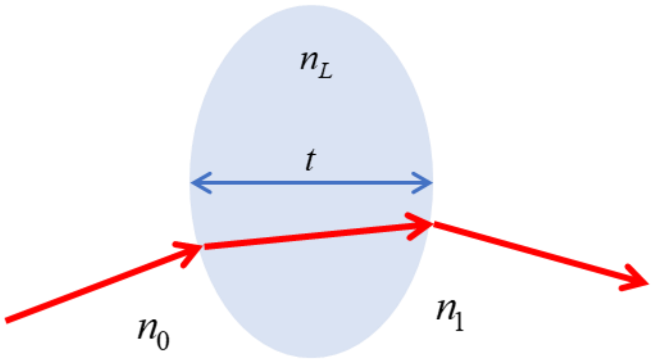
Thin lens matrix

$$M = \begin{pmatrix} 1 & 0 \\ \left(\dfrac{n_L - n}{n}\right)\left(\dfrac{1}{R_2} - \dfrac{1}{R_1}\right) & 1 \end{pmatrix}$$

$$\frac{1}{f} = -\left(\frac{n_L - n}{n}\right)\left(\frac{1}{R_2} - \frac{1}{R_1}\right)$$



Thick lens matrices



$$M = \begin{pmatrix} 1 & 0 \\ \dfrac{n_L - n_1}{n_1 R_2} & \dfrac{n_L}{n_1} \end{pmatrix} \begin{pmatrix} 1 & t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \dfrac{n_0 - n_L}{n_L R_1} & \dfrac{n_0}{n_L} \end{pmatrix}$$

The Python Code **S003C.py** defines the ABCD matrices for the examples given in Table 1. These functions can be used to calculate the transform of an input column vector (y coordinate and elevation angle [slope angle] $\alpha \equiv a$) to the output column vector. Inputs are often indicated by <<< or >>> in the Python Code. The results of running the Code are displayed in the Console Window as shown below.

1 Translation

y0 = 5.000 a0 = 18.000 deg

y1 = 5.628 a1 = 18.000 deg

2 Refraction at a plane interface

y0 = 5.000 a0 = 18.000 deg

y1 = 5.000 a1 = 12.000 deg

3 Reflection from a spherical mirror

y0 = 5.000 a0 = 18.000 deg

y1 = 5.000 a1 = 75.296 deg

4 Refraction at a spherical interface

y0 = 5.000 a0 = 18.000 deg

y1 = 5.000 a1 = 2.451 deg

5 Thin lens matrix

y0 = 5.000 a0 = 18.000 deg

y1 = 5.000 a1 = 8.419 deg

SIGNIFICANCE OF SYSTEM MATRIX ELEMENTS

We can now examine the significance of a **zero** matrix element. The transformation from input (0) to output (1) can be expressed as

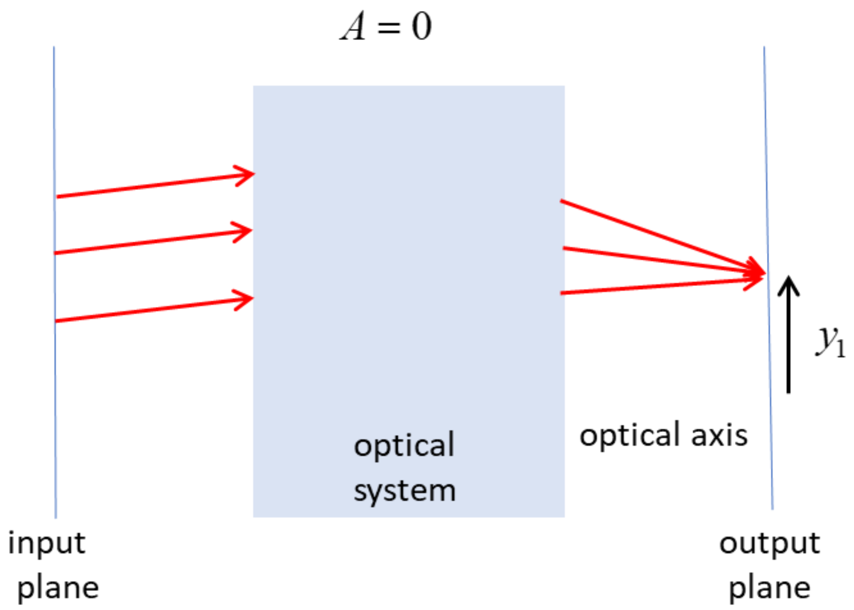
$$\begin{pmatrix} y_1 \\ \alpha_1 \end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} y_0 \\ \alpha_0 \end{pmatrix}$$

$$y_1 = A\,y_0 + B\,\alpha_0$$

$$\alpha_1 = C\,y_0 + D\,\alpha_0$$

***A* = 0** $\quad y_1 = B\,\alpha_0$

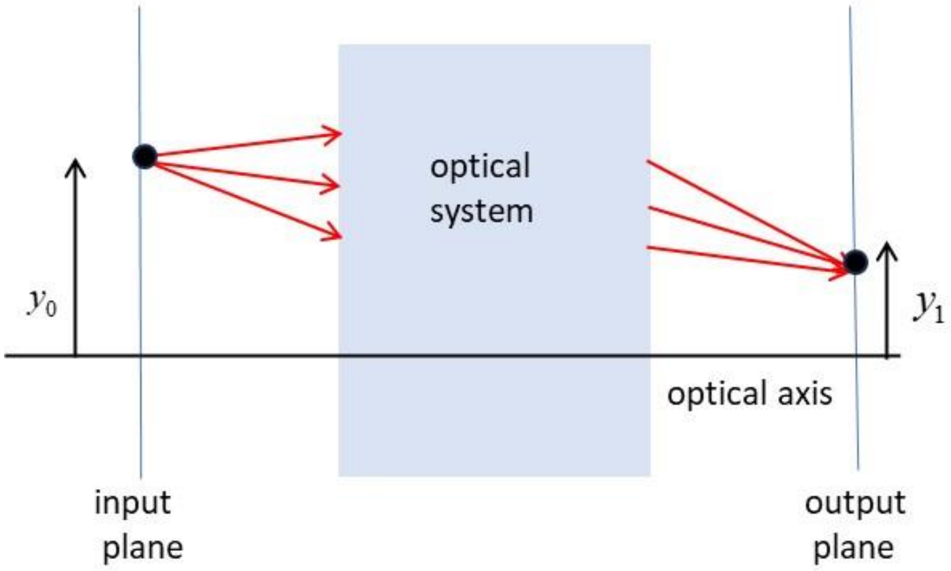
The output height (altitude) y_1 is independent of the height of the input y_0. Therefore, all rays departing the input plane at the same angle, regardless of height, arrive at the same height y_1 at the output plane. The output plane thus functions as the second focal plane.



B = 0 $\quad y_1 = A\,y_0$

The height is independent of α_0. Thus, all rays from a point at height y_0 in the input plane arrive at the same point of height y_1 in the image plane. The points are then related as **object point** and **image point**. The input and output planes correspond to **conjugate planes** for the optical system. Since $A = y_1 / y_0$ the matrix element represents the linear magnification.

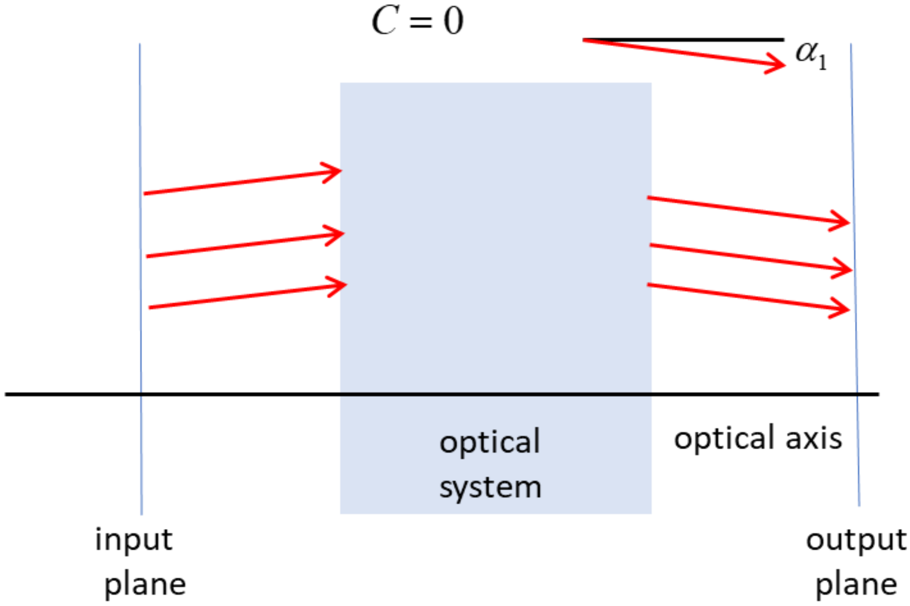
$B=0$



$C = 0$ $\qquad \alpha_1 = D\, y_0$

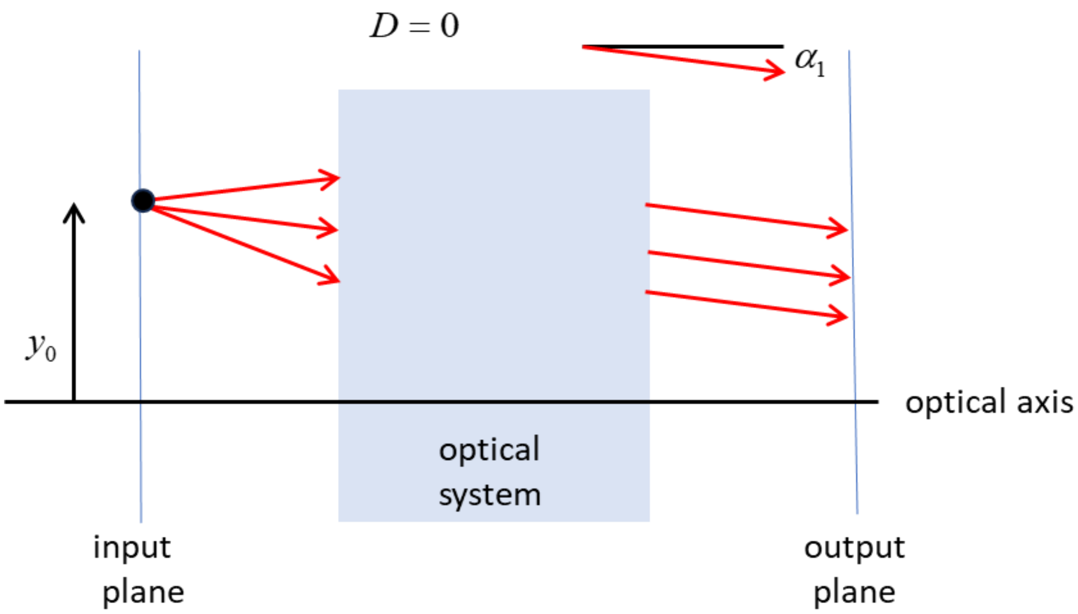
The slope angle α_1 is independent of y_0. A set of parallel input rays produces a set of parallel output rays. The angular magnification is given by $D = \alpha_1 / \alpha_0$.

$C=0$



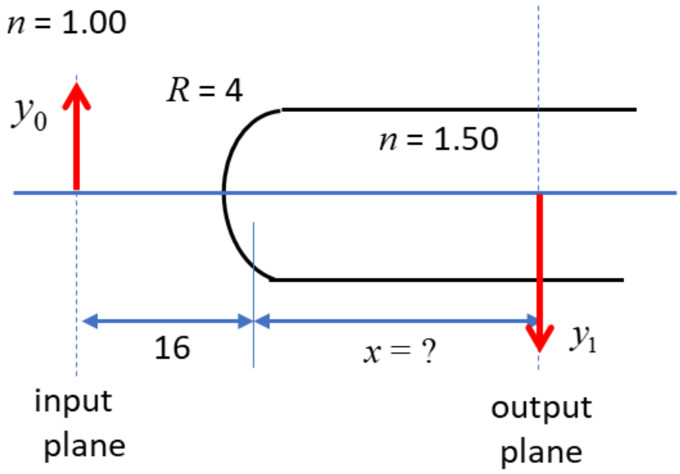
D = 0 $\alpha_1 = C\, y_0$

The output slope angle (elevation) α_1 is independent of the input slope angle α_0. Hence, for a fixed value for y_0, all rays leaving a point in the input plane will have the same angle at the output plane. The input plane thus coincides with the first focal plane of the optical system.



EXAMPLE 1 $B = 0$

A small object is placed at a distance of 16 mm from the left end of a long plastic rod with a spherical end of radius 4 mm. The refractive index of the rod is 1.50 and the object is in air. The output plane is at a distance x from the spherical cap.



We want to determine the image distance x and its lateral magnification. The system matrix consists of the product of three matrices: M1 (translation in air 16 mm), M2 (refraction at spherical surface, M3 (translation in rod x mm).

The Python Code **S003B.py** is used to answer the exercise question. The goal is to find the value of *x* (L23) such that the matrix element **B** is zero. When **B** = 0, the rays from an object point pass through the one image point as shown in figure 1. The answer is x = 24, and the linear magnification is -1 (A = -1). A summary of the simulation parameters is displayed in a the Console Window and ray paths are displayed in a Figure Window.

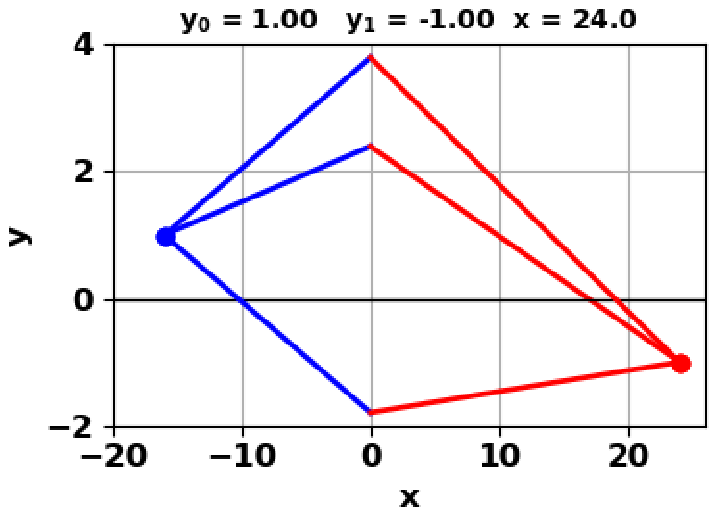


Fig. 1. **B = 0** when x = 24. All rays from a point at height y_0 in the input plane arrive at the same point of height y_1 in the image plane.

The Python Code **S003B.py** calculates each transformation so the path of the ray can be plotted and the image point is also calculated using the system matrix.

```
Example 1: propagation into long cylinder
Object plane  L01 = 16.00
Image plane  x = L23 = 24.00
Input vector:  y0 = 1.000   a0 =  5.000  deg
0 --> 1   Translation
  y1 = 2.396   a1 =  5.000  deg
[[ 1. 16.]
 [ 0.  1.]]

1 --> 2   Refraction
  y2 = 2.396   a2 =  -8.108  deg
[[ 1.         0.        ]
 [-0.08333333  0.66666667]]
```

```
2--> 3   Translation
  y3 = -1.000   a3 =  -8.108  deg
[[ 1. 24.]
 [ 0.  1.]]

System matrix
Input vector:  y0 = 1.000   a0 =  5.000  deg
Output vector:  y1 = -1.000    a1 =  -8.108  deg
A = -1.00   B = 0.00   C = -0.08   D = -0.67
```

When x = 20 (L23 = 20), **B** = 2.67, the rays from one point on the object do not coincide in the image plane (figure 2).

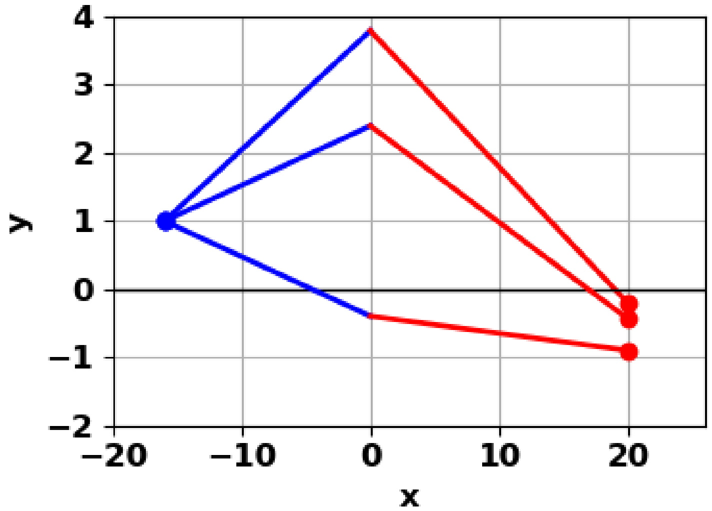


Fig. 2. The three rays from one point in the object plane do not intersect at one point in the image plane when $B \neq 0$.

$$\alpha_0 = 5^{\circ} \quad y_1 = -0.899$$

$$\alpha_0 = 10^{\circ} \quad y_1 = -0.434$$

$$\alpha_0 = -5^{\circ} \quad y_1 = -0.201$$