

MOTO: a Matlab Object-oriented programming Toolbox for Optics

Eric Anterrieu* and José-Philippe Pérez

*Laboratoire d'Astrophysique de l'Observatoire Midi-Pyrénées
Université Paul Sabatier & CNRS-UMR5572*

14 avenue Édouard Belin - 31400 Toulouse - France

Phone: (+33) 5-6133-2929 / Fax: (+33) 5-6133-2840

**Corresponding author: Eric.Anterrieu@ast.obs-mip.fr*

The ray optics is the branch of optics in which all the wave effects are neglected: the light is considered as travelling along rays which can only change their direction by refraction or reflection. On one hand, a further simplifying approximation can be made if attention is restricted to rays travelling close to the optical axis and at small angles: the well-known linear or paraxial approximation introduced by GAUSS. On the other hand, in order to take into account the geometrical aberrations, it is sometimes necessary to pay attention to marginal rays with the aid of a ray tracing procedure. This contribution describes a toolbox for the study of optical systems which implements both approaches. It has been developed in the framework of an educational project, but it is general enough to be useful in most of the cases. © 2007 Optical Society of America

OCIS codes: 000.2060 (Education), 080.0080 (Geometrical optics).

1. Introduction

Within the frame of the linear approximation, the properties of the rays travelling through an optical system can be treated with an elegant and powerful matrix formalism [1,2]. The transfer matrix between two conjugate planes is obtained with the product between elementary matrices which account for propagation through an homogeneous medium and refraction (or reflection) at an interface. When studying optical systems within the GAUSS approximation, it is sometimes necessary to determine the location and the size of the image of an object given by the optical system. This can be done in a geometrical manner by considering particular rays. However, the power and the cardinal elements, which are easily obtained from the transfer matrix, provide an accurate way to reach this goal. The NEWTON and the DESCARTES relations suffer from a restricting assumption: the optical system should be a focal one (otherwise the cardinal elements would not be defined). This is not the case of the homographic relation which is still valid for non-focal optical systems.

When studying complex optical systems, it is necessary to determine the path of the light with a greater accuracy than that obtained in the paraxial approximation. This may be done with the aid of elementary geometry, by successive application of the SNELL-DESCARTES laws of refraction (or reflection). This method, which is known as ray tracing, is intensively used in the practical study of complex optical instruments. Since in an ideal system all rays that form an image are concurrent at the same image point, only two rays need to be traced to determine the image point. However, because of geometrical aberrations, marginal rays are not concurrent at a single point, whereas paraxial ones are. This is why ray tracing is the only way to properly take into account aberrations in an optical system without referring to SEIDEL or ZERNIKE polynomials. Indeed, whatever the order used, these polynomials provide only an approximation of the geometrical aberrations while ray-tracing, which could be implemented for any interfaces (plane, spherical, parabolical, . . .), gives the true vision of the propagation through the optical system.

Both approaches have been implemented in MATLAB with the aid of the Object Oriented Programming (OOP) facilities of this language [3]. This work has led to the development of a toolbox for MATLAB which provides a quick and easy way to manipulate both dioptric and catadioptric systems, to study them interactively in the paraxial approximation (whether they are focal or non-focal) and also with attention paid to marginal rays (whether they propagate in homogeneous media or not). The toolbox conforms to MATLAB for both programming rules and inline help [4]. High level functions are supported by graphical user interfaces. This toolbox, which will be made available to the public, may facilitate the education and training of students from college to university and of professionals like those in the medical field for example. Moreover, it may also be used as a scientific research tool since the ray tracing approach gives a significant improvement and could be very helpful for a better understanding of optical systems, like the eye for example.

2. Elementary objects and functions

The Object Oriented Programming (OOP) capabilities of MATLAB are used for implementing elementary classes which contains all the necessary characteristics of elementary optical systems. These elementary classes are used for computing the cardinal elements and the transfer matrices within the GAUSS approximation, but also for drawing the path of rays travelling through an optical system whatever their inclination with the optical axis. Such an optical system is a vector which contains the individual elements. This vector is returned by the `sysopt` function which accepts a variable number of arguments and compiles/checks them in order to build the complete optical system.

2.1. Elementary objects

The file `dioptr.m` in the directory `@dioptr` contains the definition of the classe named `dioptr`. It is used for implementing refracting interfaces which are characterised by their location on the optical axes, their radius of curvature and the refractive indexes of the two media. Likewise, the file `mirror.m` in the directory `@mirror` contains the definition of the classe named `mirror`. It is used for implementing reflecting interfaces which are characterised by their location on the optical axes, their radius of curvature and the refractive index of the media. Finally, the file `diaphragm.m` in the directory `@diaphragm` contains the definition of the classe named `diaphragm`.

The functions defined in the private directories perform elementary actions on the corresponding optical subsystem like setting some characteristics, returning the transfer matrix \mathcal{R} or the propagation one \mathcal{T} , computing the intersection of a ray with the interface, ...

2.2. Transfer matrix

When considering complex optical system consisting of regions of free space with a constant refractive index separated by spherical refracting/reflecting surfaces between the input and the output front planes, E_{xy} and S_{xy} , the optical system is homogeneous step by step. Propagation through this system can be treated with the elementary matrices \mathcal{R} and \mathcal{T} corresponding to the refraction/reflection at an interface and to the propagation through free space [1]. The product of these elementary matrices, *written from right to left* following the path of the light, is the transfer matrix of the optical system within the GAUSS approximation:

$$T_{ES} = \mathcal{T}(\overline{S_p S}) \mathcal{R}(S_p) \cdots \mathcal{T}(\overline{S_1 S_2}) \mathcal{R}(S_1) \mathcal{T}(\overline{E S_1}).$$

The function `matrix` returns in the variable `T` the transfer matrix T_{ES} of the optical system between the input plane E_{xy} and the output plane S_{xy} :

```
T = matrix(soc{2});
for k=3:length(soc)
    T = matrix(soc{k-1},soc{k})*T;
    T = matrix(soc{k})*T;
end
```

The vector `soc` contains the individual elements of the optical system. Since the first cell of `soc` is used for storing the positions of the input and output planes E_{xy} and S_{xy} as well as the refractive indexes n_o and n_i of the initial and final media, `T` is initialized with the transfer matrix of the first element reached by the light, here stored in the second cell. Then, products with propagation and transfer matrices are used up to the last element to obtain the final transfer matrix between E_{xy} and S_{xy} .

The function `power` returns the opposite of `T(2,2)` which is by definition the refractive power V of the optical system.

2.3. Cardinal elements

The function `focal` returns in the two variables `fo` and `fi` the object and image focal lengths $f_o = -n_o/V$ and $f_i = n_i/V$:

```
fo = -No/V;
fi = Ni/V;
```

where `No` and `Ni` are the refractive indexes n_o and n_i of the initial and final media, and `V` is the power V of the optical system as returned by the `power` function.

The function `cardinal` returns in the six variables `EFo`, `SFi`, `EHo`, `SHi`, `ENo` and `SNi` the location of the focal planes ($\overline{EF_o}$ and $\overline{SF_i}$), the principal planes ($\overline{EH_o}$ and $\overline{SH_i}$) and the nodal points ($\overline{EN_o}$ and $\overline{SN_i}$) with respect to the input and output planes E_{xy} and S_{xy} :

$$\begin{aligned} \text{EFo} &= \text{fo} * \text{T}(2,2); & \text{EHo} &= \text{fo} * (\text{T}(2,2) - 1); & \text{ENo} &= \text{fo} * (\text{T}(2,2) - \text{Ni}/\text{No}); \\ \text{SFi} &= \text{fi} * \text{T}(1,1); & \text{SHi} &= \text{fi} * (\text{T}(1,1) - 1); & \text{SNi} &= \text{fi} * (\text{T}(1,1) - \text{No}/\text{Ni}); \end{aligned}$$

where `T` is the transfer matrix T_{ES} of the optical system returned by the `matrix` function.

2.4. Algebraic determination of an image point within Gauss approximation

The function `homographic` returns in the variable `SAi` the position $z_i = \overline{SA_i}$ of an image A_i :

$$\text{SAi} = \text{Ni} * (\text{T}(1,1) * (\text{EAo}/\text{No}) - \text{T}(1,2)) / (-\text{T}(2,1) * (\text{EAo}/\text{No}) + \text{T}(2,2));$$

where `EAo` is the position $z_o = \overline{EA_o}$ of the object A_o . The function `invhomographic` performs the reverse operation:

$$\text{EAo} = \text{No} * (\text{T}(2,2) * (\text{SAi}/\text{Ni}) + \text{T}(1,2)) / (\text{T}(2,1) * (\text{SAi}/\text{Ni}) + \text{T}(1,1));$$

These two functions also return in the three variables `Gt`, `Ga` and `Gl` the transversal, angular and longitudinal magnifications G_t , G_a and G_l :

$$\text{Gt} = \text{TaoAi}(1,1); \quad \text{Ga} = \text{TaoAi}(2,2) * \text{No}/\text{Ni}; \quad \text{Gl} = \text{Gt}/\text{Ga};$$

where `TaoAi` is the transfer matrix between the conjugate planes A_oxy and A_ixy :

$$\text{TaoAi} = [1 \ \text{SAi}/\text{Ni}; \ 0 \ 1] * \text{T} * [1 \ -\text{EAo}/\text{No}; \ 0 \ 1];$$

and is easily derived from the transfer matrix T_{ES} of the optical system and the transfer matrices describing the propagation between the front planes A_oxy and E_{xy} on one hand, S_{xy} and A_ixy on the other hand.

The function `descartes` returns in the variable `HiAi` the position $p_i = \overline{H_iA_i}$ of an image A_i computed with the aid of DESCARTES's relation with regards to the principal points:

$$\text{HiAi} = \text{Ni} * (\text{V} + \text{No}/\text{HoAo});$$

where `HoAo` is the position $p_o = \overline{H_oA_o}$ of the object A_o . The function `invdescartes` performs the reverse operation:

$$\text{HoAo} = \text{No} * (\text{Ni}/\text{HiAi} - \text{V});$$

In addition, these functions also return the three magnifications:

$$\text{Gt} = (\text{No}/\text{Ni}) * (\text{HiAi}/\text{HoAo}); \quad \text{Ga} = \text{HoAo}/\text{HiAi}; \quad \text{Gl} = \text{Gt}/\text{Ga};$$

Finally, the `newton` function returns in the variable `FiAi` the position $\sigma_i = \overline{F_iA_i}$ of an image A_i computed with the aid of NEWTON's relation with regards to the focal points:

$$\text{FiAi} = -(\text{No}/\text{Ni}) / \text{V}^2 / \text{FoAo};$$

where `FoAo` is the position $\sigma_o = \overline{F_oA_o}$ of the object A_o . The function `invnewton` performs the reverse operation:

$$\text{FoAo} = (\text{No}/\text{Ni}) / \text{V}^2 / \text{FiAi};$$

In addition, these functions also return the three magnifications:

$$\text{Gt} = -(\text{V} * \text{FiAi}) / \text{Ni}; \quad \text{Ga} = -\text{No} / (\text{V} * \text{FiAi}); \quad \text{Gl} = \text{Gt}/\text{Ga};$$

or

$$\text{Gt} = \text{No} / (\text{V} * \text{FoAo}); \quad \text{Ga} = (\text{V} * \text{FoAo}) / \text{Ni}; \quad \text{Gl} = \text{Gt}/\text{Ga};$$

2.5. Graphical determination of an image point within Gauss approximation

The function `plotcardinal` plots the position of the cardinal elements on an optical axis where the input and output planes E_{xy} and S_{xy} of the optical system are also drawn.

For a given object A_oB_o at a distance $z_o = \overline{EA_o}$ from E_{xy} and for the corresponding image A_iB_i located at a distance $z_i = \overline{SA_i}$ from S_{xy} , the function `plotgaussrays` plots the three particular rays:

- the ray entering the system from B_o parallel to the optical axis emerges from the system by crossing the image focal point F_i towards B_i ,
- the ray entering the system from B_o and crossing the object focal point F_o emerges parallel to the system towards B_i ,
- the ray entering the system from B_o and crossing the object nodal point N_o emerges parallel to its incident direction and crossing the image nodal point N_i towards B_i .

2.6. Ray tracing

When propagating within the optical system, the rays are refracted/reflected by the successive interfaces. Between these interfaces, light is travelling in straight line in an homogeneous media. The corresponding rays are characterized by a set of coordinates (z, xy) (namely the coordinates of the intersection point with each interface) and by an inclination angle α with the optical axis which are computed with the aid of SNELL-DESCARTES laws. The function `rays` returns in the variables `z`, `xy` and `a` the successive locations and inclinations of these rays:

```
for k=2:length(soc)
    [z(k),xy(k),a(k)] = ray(soc{k},z(k-1),xy(k-1),a(k-1));
end
z(k+1) = ze;
xy(k+1) = xy(k)+tan(a(k))*(ze-z(k));
a(k+1) = a(k);
```

The function `ray` is a private function of the elementary classes described in section 2.1 which propagates the ray through each individual element of the optical system `soc`. The function `rays` is ended by computing the propagation of the ray after the optical system up to a plane, namely a virtual screen. Finally, the path of the rays can be plotted with the function `plotrays`.

3. Application in homogeneous media

The functions described in the previous section are used here for studying a complex optical system, namely the human eye. The dimensions of the eye and the characteristics of its optical components vary greatly from person to person, and some further depend upon accommodation level, age and certain pathological conditions. Despite these variations, average values have been used to construct representative or schematic eyes. The standard model used for this work is the LE GRAND model [5] which is a four interfaces model: the two first interfaces correspond to the cornea, the two last ones constitute the lens. The function `LeGrand` returns all the elements of this model when it is relaxed:

```
cornea1 = dioptr('cornea',0.00E-03, 7.80E-03,1.0000,1.3771);
cornea2 = dioptr('cornea',0.55E-03, 6.50E-03,1.3771,1.3374);
iris    = diaphragm('iris' ,3.60E-03,10.20E-03,6.00E-03);
lens1   = dioptr('lens' ,3.60E-03,10.20E-03,1.3374,1.4200);
lens2   = dioptr('lens' ,7.60E-03,-6.00E-03,1.4200,1.3360);
```

or when it accommodates. However, in the present study, we will concentrate on the relaxed eye:

```
>> eye = LeGrand('relaxed');
```

3.1. Gauss approximation

The transfer matrix is easily obtained with the `matrix` function:

```
>> T = matrix(eye)
T = 0.7446    0.0054
    -59.9404    0.9044
```

Then the focal distances are returned by the `focal` function:

```
>> [fo,fi] = focal(eye)
fo = -0.0167
fi = 0.0223
```

Finally, the cardinal elements are computed with the `cardinal` function:

```
>> [EFo,SFi,EHo,SHi,ENo,SNi] = cardinal(eye)
EFo = -0.0151
SFi = 0.0166
EHo = 0.0016
SHi = -0.0057
ENo = 0.0072
SNi = -0.0001
```

and they are represented in Fig. 1 with the `plotcardinal` function:

```
>> figure(1); plotcardinal(eye);
```

The total power of the eye is returned by the `power` function:

```
>> V = power(eye)
V = 59.9404
```

It can be checked that approximately two third of this power is due to the cornea:

```
>> cornea = LeGrand('relaxed','cornea');
>> V1 = vergence(cornea)
V1 = 42.3564
>> lens = LeGrand('relaxed','lens');
>> V2 = vergence(lens)
V2 = 21.7787
```

The optical distance e between the cornea and the lens can therefore be computed with the aid of GULLSTRAND's formula. Indeed, since the refractive index of the aqueous, the medium between the posterior face of the cornea and the anterior face of the lens, is 1.3374, we have:

```
>> e = 1.3374*(V1+V2-V)/(V1*V2)
e = 0.0061
```

It can be verified that e is equal to the distance between the image principal plane of the cornea and the object principal plane of the lens by computing the cardinal elements of these two optical subsystems.

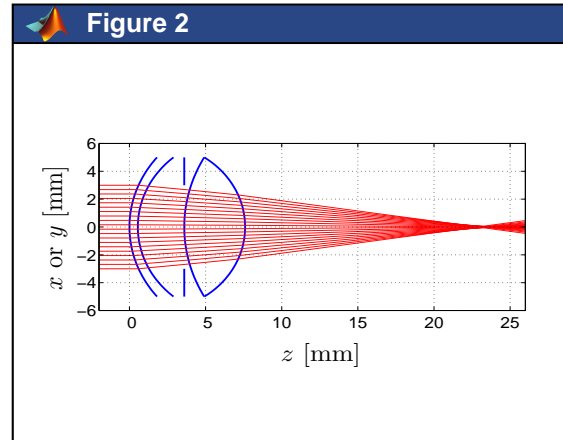
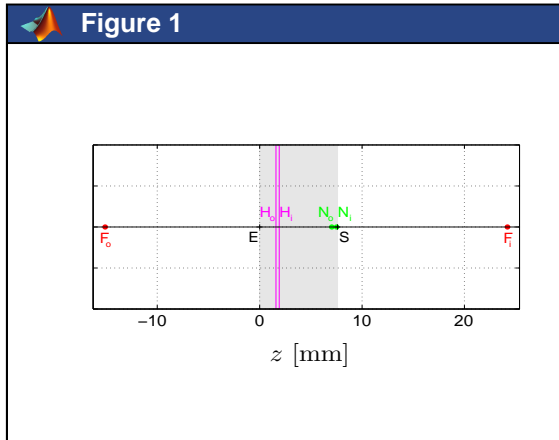
The iris is the aperture stop of the eye. The position EP_e of input pupil P_e with respect to the input plane E_{xy} can be obtained with the aid of the `invhomographic` function:

```
>> [EPe,Gt] = invhomographic(cornea,3.05E-03)
EPe = 0.0030
Gt = 0.8841
```

since it is the conjugate point of the aperture stop given by the optical elements in front of the latter (i.e. here the cornea). It is located 3.04 mm behind E_{xy} , that is to say 0.56 mm in front of the iris. Taking into account the transversal magnification, its diameter is equal to 6.78 mm. Likewise, the location of the output pupil P_s with respect to the output plane S_{xy} is obtained with the aid of the `homographic` function:

```
>> [SPs,Gt] = homographic(lens,0.00E-03)
SPs = -0.0039
Gt = 1.0411
```

since it is the image of the aperture stop given by the optical elements behind the latter (i.e. here the lens). It is located 3.92 mm in front of the output plane S_{xy} , that is to say only 0.08 mm behind the iris. Accounting for the transversal magnification, its diameter is equal to 6.25 mm.



3.2. Ray tracing and aberrations

We now consider an incident beam of 20 rays parallel to the optical axis. The diameter of this beam is, for example, 6 mm, and the path of the rays will be plotted from a plane located 2 mm in front of the cornea to another plane 26 mm behind it in the vicinity of the retina:

```
>> [z,xy,alpha] = rays(eye,-2E-03,linspace(-3E-03,3E-03,20),0,26E-03);
```

The result is represented on Fig. 2 with the aid of the `plotsysopt` and `plotrays` functions:

```
>> figure(2); plotsysopt(eye,5.0E-03); plotrays(z,xy);
```

It can be observed that the eye is not an optical system which can be only studied within the GAUSS approximation. For example, the localization of the retina, that is to say the localization of the image focal plane, requires a ray tracing procedure. From the final coordinates and inclination of the rays when they hit the plane located 26 mm in the back of the cornea, the equation of the rays emerging from the lens can be computed:

```
>> a = tan(alpha(:,end));
>> b = xy(:,end) - a.*z(:,end);
```

Then, the distance $z = -b/a$ from the origin where the rays cross the optical axis is computed:

```
>> z = -b./a;
>> r = abs(xy(:,1));
```

The variations of the radius of the incident beam r wrt. z are shown on Fig. 3:

```
>> figure(3); plot(z*1.0E+03,r*1.0E+03);
```

This figure illustrates the fact that incident rays parallel to the optical axis are all the more convergent when they are far from the optical axis. Moreover, all the rays cross the optical axis before the image focal plane whose location in the GAUSS approximation is 24.2 mm from the anterior face of the cornea. Finally, the diameter d of the spot in a plane located at a given distance z from the anterior face of the cornea can be computed:

```
>> z = (21:0.01:26);
>> for k=1:length(z), d(k) = 2*max(a*z(k) + b); end
```

The variations of d wrt. z are shown on Fig. 4:

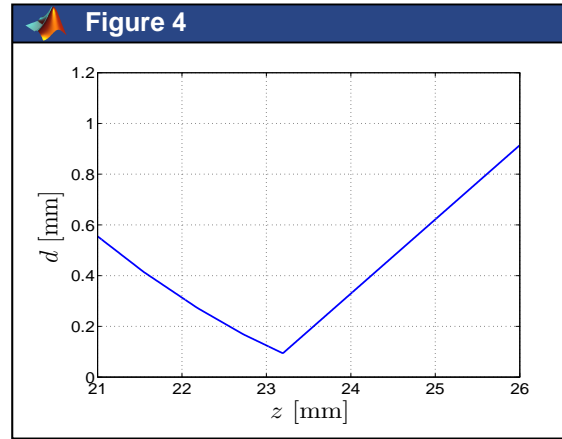
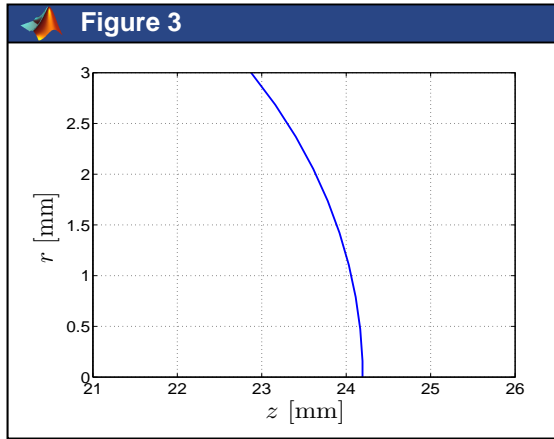
```
>> figure(4); plot(z*1.0E+03,d*1.0E+03);
```

It can be observed that the smallest spot, whose diameter d is here about 0.1 mm, is not in the focal plane but in a plane which is 1 mm in front of it:

```
>> [dmin,k] = min(d);
>> [d(k)*1E+03, z(k)*1E+03]
ans = 0.0948 23.1900
```

since it is located 23.2 mm behind the anterior face of the cornea compared to 24.2 mm in the GAUSS approximation.

The following figures show that the localization of the retina is not a matter of GAUSS approximation but requires a ray tracing procedure since it is always over-estimated by the former approach.



3.3. LASIK

Nowadays, some ametropias of the eye (myopia and hypermetropia) could be corrected by means of a modification of the shape of the anterior face of the cornea. A photo-ablation of a small piece of the cornea is obtained with the aid of an excimer laser. After cicatrizing, the radius of curvature of the anterior face of the cornea is modified. The refractive power of the cornea is changed accordingly, and the ametropic eye becomes emmetropic. This surgical technique is known under the acronym LASIK which stands for Laser Assisted In-situ Keratomileusis. One of the problems encountered in practice is to evaluate the amount of cornea removal in order to give to the cornea the expected refractive power [6]. We show in this section that a ray tracing approach could be very helpful to achieve this goal since it is the only way to properly take into account aberrations of the cornea [7].

The free parameters for the photo-ablation of a piece of cornea are the thickness ε of the corneal tissue which is burned as well as the diameter D of the surgical field. The goal is to change the radius of curvature R of the anterior face of the cornea so that, after cicatrizing, the new radius of curvature R' leads to a better focussing of the rays on the retina. Both radii are related to each other with the following equation [1]:

$$R' = \frac{(h - \varepsilon)^2 + (D/2)^2}{2(h - \varepsilon)} \quad \text{with} \quad h = R - \sqrt{R^2 - (D/2)^2}.$$

Measurements of keratometry and pachymetry of an eye suffering from myopia have led to the following numbers:

```
>> cornea1 = dioptr('cornea',0.00E-03, 7.25E-03,1.0000,1.3771);
>> cornea2 = dioptr('cornea',0.41E-03, 5.25E-03,1.3771,1.3374);
>> iris    = diaphragm('iris' ,3.45E-03,10.33E-03,6.00E-03);
>> lens1   = dioptr('lens' ,3.45E-03,10.33E-03,1.3374,1.4200);
>> lens2   = dioptr('lens' ,7.60E-03,-6.17E-03,1.4200,1.3360);
>> eye     = sysopt(cornea1,cornea2,iris,lens1,lens2);
```

The diameter of the surgical field D has been fixed to 6 mm. The radius of curvature of the reshaped anterior face of the cornea R' is here computed for different values of the thickness ε :

```
>> D = 6.00E-03;
>> R = 7.25E-03;
>> h = R - sqrt(R^2 - (D/2)^2);
>> e = (0:150)*1.0E-06;
>> R = ((h-e).^2 + (D/2)^2)./(2*(h - e));
```

The variations of R' wrt. ε are represented on Fig. 5:

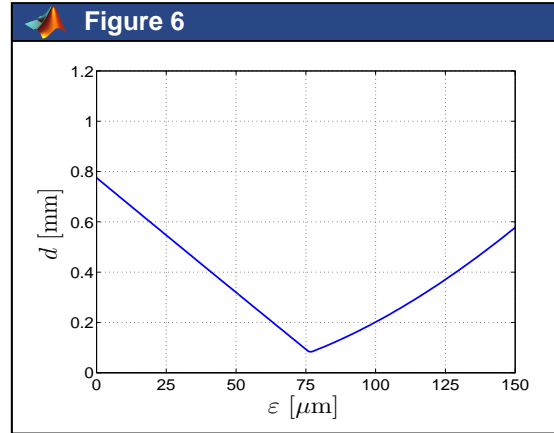
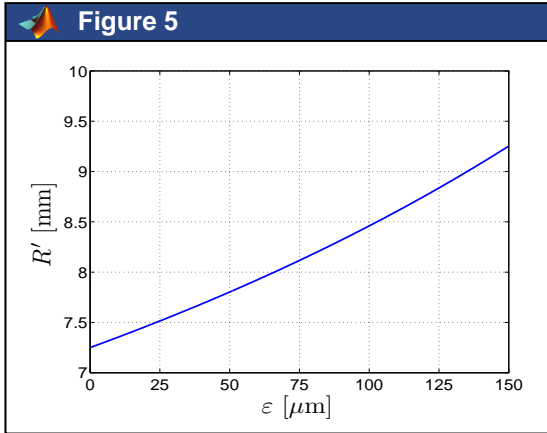
```
>> figure(5); plot(e*1.0E+06,R*1.0E+03);
```

Measurements have shown that the retina is located 17.2 mm behind the posterior face of the lens, that is to say 24.8 mm behind the anterior face of the cornea. The diameter d of the spot on the retina is computed for an incident beam of 20 rays parallel to the optical axis. The width of this beam is taken equal to D :

```
>> for k=1:length(e)
>> eye{2} = set(eye{2},'Position',e(k),'Curvature',R(k));
>> [z,xy,alpha] = rays(eye,-2.0E-03,linspace(-D/2,D/2,20),0,26.0E-03);
>> a = tan(alpha(:,end));
>> b = xy(:,end) - a.*z(:,end);
>> d(k) = 2*max(a*24.8E-03 + b);
>> end
```

The variations of d wrt. ε are represented on Fig. 6:

```
>> figure(6); plot(e*1.0E+06,d*1.0E+03);
```



The thickness of the tissue to be burned corresponds to the value of ε for which d is minimal:

```
>> [dmin,k] = min(d);
>> [e(k)*1.0E+06, d(k)*1.0E+03, R(k)*1.0E+03]
ans = 77 0.0829 8.1424
```

These values can be compared to those before correction, namely those for $\varepsilon = 0$:

```
>> [e(1)*1.0E+06, d(1)*1.0E+03, R(1)*1.0E+03]
ans = 0 0.7754 7.2500
```

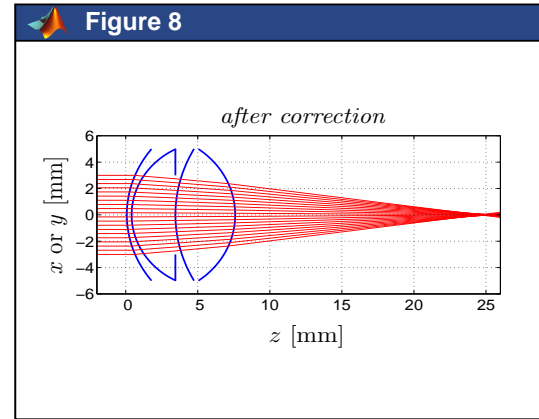
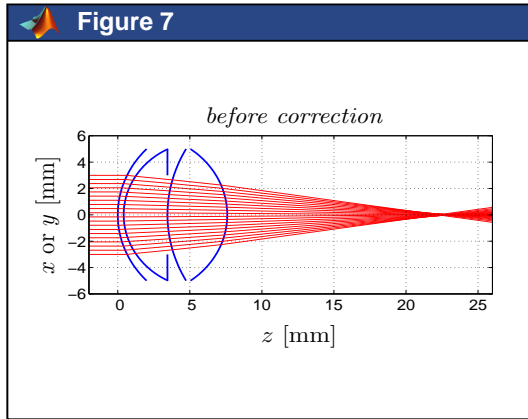
The thickness of the corneal tissue to be burned is $\varepsilon = 77 \mu\text{m}$ in the simulated surgical conditions. The diameter of the spot in the retina plane is therefore reduced from $d = 0.78 \text{ mm}$ down to less than $d = 0.09 \text{ mm}$. To reach this goal, the radius of curvature of the anterior face of the cornea has increased from $R = 7.25 \text{ mm}$ up to $R' = 8.14 \text{ mm}$.

The consequence of this change of the radius of curvature is illustrated in Figs. 7 and 8 where the path of rays are plotted before and after corneal tissue removal:

```
>> eye{2} = set(eye{2},'Position',e(1),'Curvature',R(1));
>> [z,xy,alpha] = rays(eye,-2.0E-03,linspace(-D/2,D/2,20),0,26.0E-03);
>> figure(7); plotsysopt(eye,5.0E-03); plotrays(z,xy); title('before correction');
>> eye{2} = set(eye{2},'Position',e(k),'Curvature',R(k));
>> [z,xy,alpha] = rays(eye,-2.0E-03,linspace(-D/2,D/2,20),0,26.0E-03);
>> figure(8); plotsysopt(eye,5.0E-03); plotrays(z,xy); title('after correction');
```

Paying attention to the path of the rays close to the retina plane located 24.8 mm behind the anterior face of the cornea, it is observed that after correction the rays are converging on the retina whereas they were converging before the retina plane before correction since the eye was suffering from myopia.

The same simulation can be conducted with different values for the diameter D of the surgical field. The corresponding numbers show that, like expected, the thickness ε of the corneal tissue to be burned for reaching the same goal is an increasing function of D .



4. Application in non-homogeneous media

The capabilities of MATLAB to solve ordinary differential equations (ODE) is used here for studying the propagation of light in non-homogeneous media. The ODE which describes the path of the light is:

$$\frac{d}{ds} \left(n \frac{d\mathbf{r}}{ds} \right) = \mathbf{grad} n,$$

where \mathbf{r} denotes the position vector of a point on the ray which is a function of the length of arc s of the ray [1,2]. For the numerical implementation of a ray tracing procedure, it is convenient to set $dl = ds/n$ [1] so that the previous ODE reads:

$$\frac{d^2\mathbf{r}}{dl^2} = \frac{1}{2}\mathbf{grad} n^2.$$

In this study, light is propagating in a plane. In cartesian coordinates x and y , the previous vector equation reduces to a system of two equations of second order:

$$\begin{aligned} \frac{d^2x}{dl^2} &= \frac{1}{2} \frac{\partial n^2}{\partial x}, \\ \frac{d^2y}{dl^2} &= \frac{1}{2} \frac{\partial n^2}{\partial y}. \end{aligned}$$

Since MATLAB can only solve ODE of the first order, the previous system of second order has to be rewritten:

$$\begin{aligned} \frac{dp_x}{dl} &= \frac{1}{2} \frac{\partial n^2}{\partial x}, & \frac{dx}{dl} &= p_x, \\ \frac{dp_y}{dl} &= \frac{1}{2} \frac{\partial n^2}{\partial y}, & \frac{dy}{dl} &= p_y. \end{aligned} \quad \text{with}$$

In this study, the refractive index of the non-homogeneous media satisfies the law [8]:

$$n^2(\rho) = 1 + \frac{\rho_0^2}{\rho^2} \quad \text{with} \quad \rho = \sqrt{x^2 + y^2}.$$

The region where the media is non-homogeneous will be restricted to a disk with radius $R > \rho_0$. Outside this disk, n will be supposed to be uniform and equal to $n_i = \sqrt{1 + \rho_0^2/R^2}$, so that no discontinuity occurs at the boundary. If this were not the case, SNELL-DESCARTES laws would be applied at the boundary where the refractive index may vary.

Within the non-homogeneous media, the gradient of n^2 can be written also in cartesian coordinates:

$$\begin{aligned} \frac{\partial n^2}{\partial x} &= -\frac{2x\rho_0^2}{(x^2 + y^2)^2}, \\ \frac{\partial n^2}{\partial y} &= -\frac{2y\rho_0^2}{(x^2 + y^2)^2}. \end{aligned}$$

The function `ode2D` contains the final definition of the problem under study: in vector `f` are stored the values of x , y , p_x and p_y for a given value of l , whereas the vector `df` returns the values of p_x , p_y , dp_x/dl and dp_y/dl for the same value of l .

```
function df=ode2D(l,f,ro)
x = f(1); px = f(3);
y = f(2); py = f(4);
dpxdl = 0.5*(-2*x*ro^2/(x^2+y^2)^2);
dpydl = 0.5*(-2*y*ro^2/(x^2+y^2)^2);
df = [px; py; dpxdl; dpydl];
```

Before solving the ODE described in function `ode2D` it is necessary to precise the initial conditions of the problem. We consider here an incident ray hitting the boundary of the disk at a point $M_i(x_i, y_i)$ on the circle with radius R making an angle α with Ox axis. In the neighbourhood of M_i , we have: $dx = ds \cos \alpha$ and $dy = ds \sin \alpha$. Since at any point on the ray path $dl = ds/n$, we therefore have:

$$(dx/dl)_i = n_i \cos \alpha \quad \text{and} \quad (dy/dl)_i = n_i \sin \alpha.$$

The radius of the disk is set, for example, to $R = 4\rho_o$ with $\rho_o = 1$. For clarity reasons, we will restrict the study to incidents rays parallel to Ox axis, at a distance h from this axis, so that $\alpha = \pi$.

```
>> ro = 1;
>> R = 4*ro;
>> h = 2*ro;
>> Xi = sqrt(R^2 - h^2);
>> Yi = h;
>> Ni = sqrt(1 + (ro/R)^2);
>> alpha = pi;
>> dXidl = Ni*cos(alpha);
>> dYidl = Ni*sin(alpha);
```

After point M_i , the light propagates in the non-homogeneous media and the resulting path is the solution of the ODE described in the function `ode2D`. This equation is solved here with the aid of the function `ode45` whose integration scheme is based on a fifth order RUNGE-KUTTA approach:

```
>> options = odeset('Stats','on','RelTol',1E-12,'AbsTol',1E-12);
>> [l,f] = ode45('ode2D',[0:0.1:10],[Xi Yi dXidl dYidl],options,ro);
>> X = f(:,1); Y = f(:,2); dXd1 = f(:,3); dYd1 = f(:,4);
```

The values of the solutions x , y , $p_x = dx/dl$ et $p_y = dy/dl$ along the ray path in the disk are returned in the array `f`. For convenience reasons, they are written in separates vectors `X`, `Y`, `dXd1` and `dYd1`.

The ray emerges from the disk at a point $M_e(x_e, y_e)$ on the circle with radius R with an inclination angle β with the Ox axis. The coordinates (x_e, y_e) are computed with the aid of a simple linear interpolation between two points lying on each side of the boundary. The tangent of β is equal to the ratio $(dy/dx)_e$ which is computed from the derivatives $(dy/dl)_e$ and $(dx/dl)_e$ obtained again after a linear interpolation.

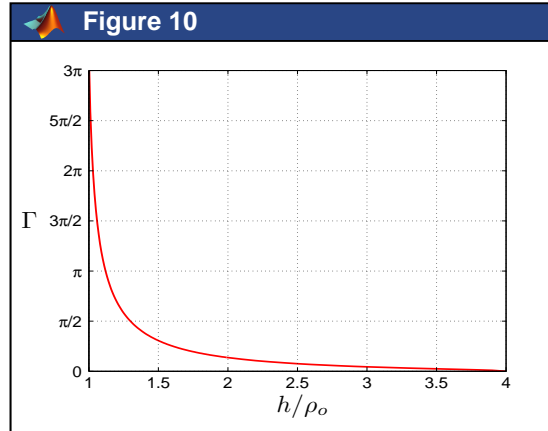
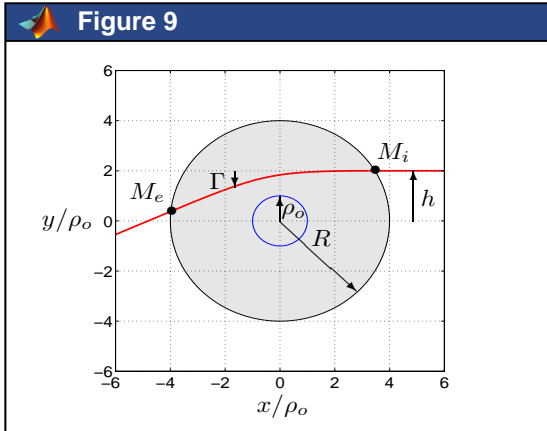
```
>> k = max(find((X.^2+Y.^2) < R^2));
>> le = interp1([X(k)^2+Y(k)^2,X(k+1)^2+Y(k+1)^2],[l(k),l(k+1)],R^2,'linear');
>> Xe = interp1([l(k),l(k+1)],[X(k),X(k+1)],le,'linear');
>> Ye = interp1([l(k),l(k+1)],[Y(k),Y(k+1)],le,'linear');
>> dXed1 = interp1([l(k),l(k+1)],[dXd1(k),dXd1(k+1)],le,'linear');
>> dYed1 = interp1([l(k),l(k+1)],[dYd1(k),dYd1(k+1)],le,'linear');
>> beta = atan2(dYed1,dXed1);
```

The complete path of the ray is represented on Fig. 9:

```
>> figure(9); plot([X(1:k); Xe]/ro,[Y(1:k); Ye]/ro);
```

Up to M_i and beyond M_e the light is travelling in straight line since the media is homogeneous with uniform refractive index n_i . On the contrary, from M_i to M_e , in the disk of radius R , the path is curved since the light is here travelling in a non-homogeneous media.

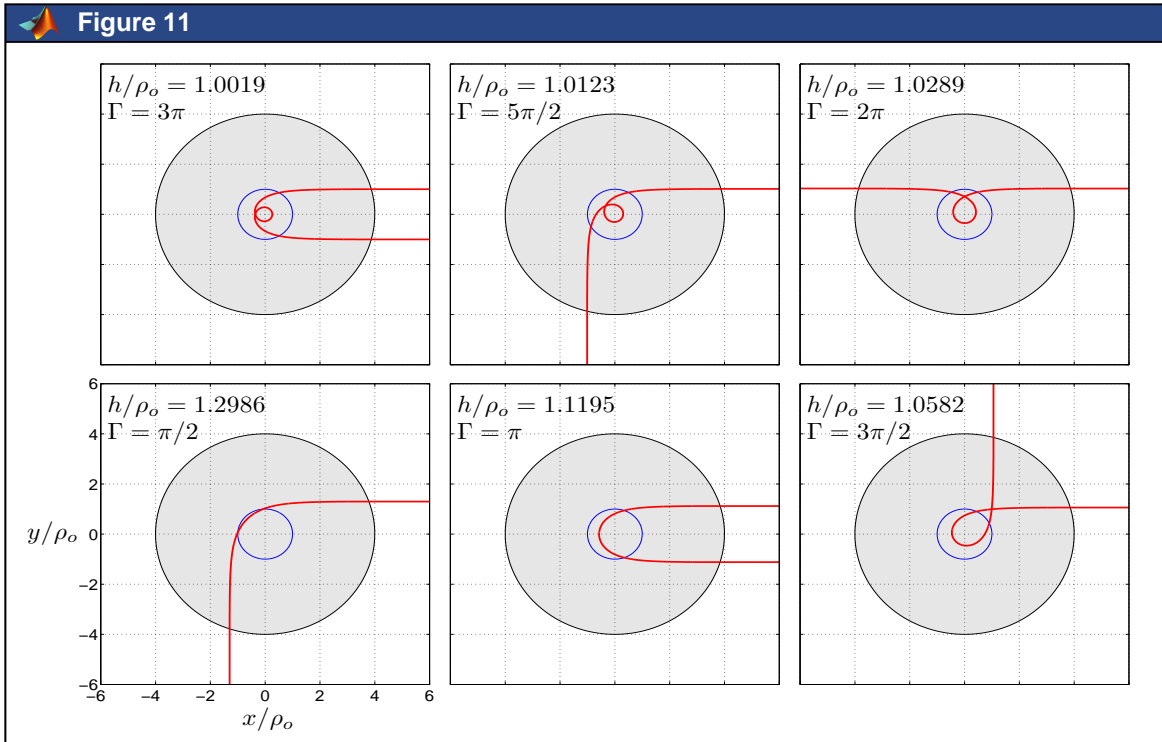
The curvature of the path is oriented towards the center of the disk, that is to say towards the direction of $\mathbf{grad} n$. The ray continuously tends towards the center without reaching it and moves away in a symmetric manner with respect to the location where the distance from the center of the disk was minimal.



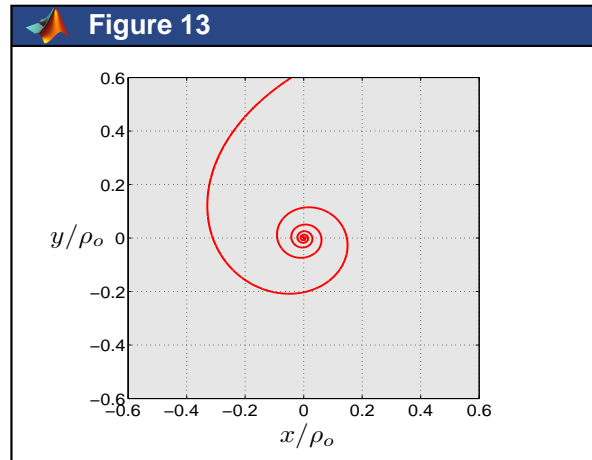
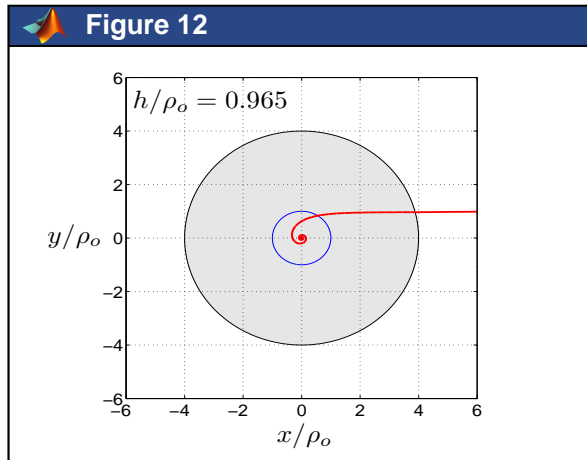
According to Fig. 9, the deviation angle Γ after travelling through the disk is equal to $\beta + \pi$:

```
>> gamma = beta+pi
gamma = 0.4288
```

Here, for $h = 2\rho_o$, the deviation is about 25° with respect to the incident direction. As shown in Fig. 10, Γ varies in a non-linear manner provided that h does not go below a limit h_ℓ which depends on ρ_o , R and α . Values of Γ greater than $\pi/2$ correspond to rays which make a half turn, or even more than a complete turn, before leaving the disk. Some particular situations are shown in Fig. 11 for different values of the ratio h/ρ_o .



One can think that the light can make a large number of turns before leaving the disk. On the contrary, below the limit h_ℓ , the path of the light identifies to that of a spiral: the ray seems to be attracted by the discontinuity of the refractive index of the non-homogeneous media for $\rho = 0$ and does not emerge from the disk. Such a situation is illustrated in Figs. 12 and 13 for $h = 0.965\rho_o$:



These simulations illustrate the results, sometimes amazing, of the propagation of the light in a non-homogeneous medium.

5. Conclusion

This contribution has described a MATLAB toolbox for the study of optical systems. Thanks to the Object Oriented Programming (OOP) capabilities of MATLAB, this toolbox provides a quick and easy way to manipulate both dioptric and catadioptric systems, to study them interactively in the paraxial approximation (whether they are focal or non-focal) and also with attention paid to marginal rays (whether they propagate in homogeneous media or not).

Two illustrations in both homogeneous and non-homogeneous media have demonstrated the capabilities of this toolbox for the education and training of students from college to university and for professionals like those in the medical field for example.

References

1. J.-P. Pérez, *Optique: fondements et applications*, Dunod (Paris), 2004.
2. M. Born and E. Wolf, *Principles of Optics*, Pergamon Press (Oxford), 1964.
3. E. Anterrieu and J.-P. Pérez, *Comparizon of matrix method and ray tracing in the study of complex optical systems*, in proc. 6th international topical meeting on Education and Training in Optics and Photonics (ETOP'99), Vol. 3831, pp. 268-279, 1999.
4. *Using Matlab*, The MathWorks Inc., 6th ed. revised for Matlab 6.5, 2002.
5. G. Smith and D.A. Atchison, *The eye and visual optical instruments*, Cambridge University Press (Cambridge), 1997.
6. J.-P. Colliac and J.-P. Pérez, *Gaussian optics for photorefractive keratectomy*, *Ophthalmology*, Vol. 103, No. 11, pp. 1956-1961, 1996.
7. M.K. Smolek and S.D. Klyce, *Zernike polynomial fitting fails to represent all visually significant corneal aberrations*, *Investigative Ophthalmology & Visual Science*, Vol. 44, No. 11, pp. 4676-4681, 2003.
8. T.H. Boyer, *Unfamiliar trajectories for a relativistic particle in a Kepler or Coulomb potential*, *American Journal of Physics*, Vol. 72, No. 8, pp. 992-997, 2004.