

CR TP4 : Aberrations des lentilles plans convexes

Objectif du TP :

L'objectif est de tester, à l'aide d'un langage de programmation comme python, le stigmatisme approché d'une lentille demi-boule pour des rayons proches de l'axe optique.

Lentille convergente :

1. Tracer de la lentille et du rayon :

Tout d'abord il faut tracer, en utilisant les packages numpy et matplotlib, l'axe optique et la lentille.

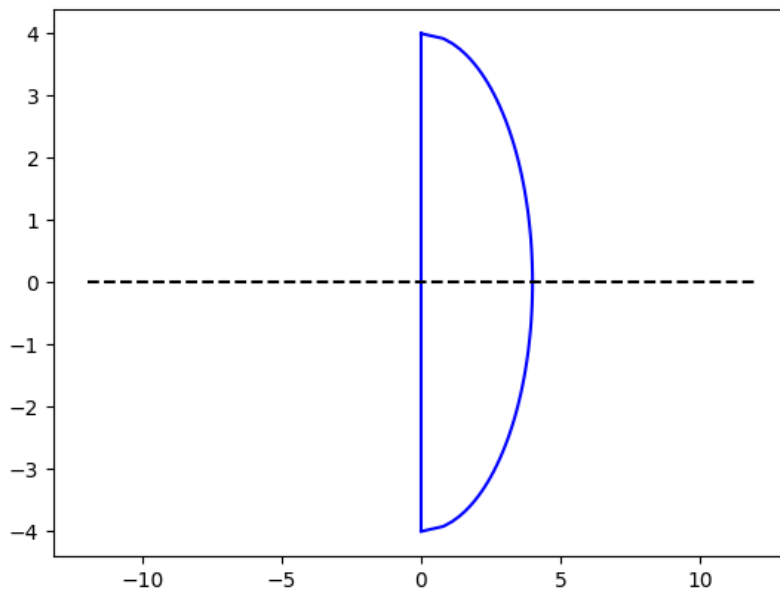
Pour cela, on crée la fonction *lentilleconvergente*(d, r) avec d un diamètre et r un rayon de courbure et avec les formules suivantes :

- L'épaisseur $e = r - r \sqrt{1 - \frac{d^2}{4r^2}}$
- L'équation décrivant la portion du cercle de la lentille $x(y) = \sqrt{r^2 - y^2} - r + e$

On obtient donc la fonction ci-dessous :

```
def lentilleconvergente(d,r):  
    e = r - r * np.sqrt(1 - (d**2 / (4 * r**2)))  
    y1 = np.linspace(-1/2*d, 1/2*d, 100)  
    x1 = np.sqrt(r**2 - y1**2) - r + e  
    y2= np.linspace(-3/2*d, 3/2 *d, 100)  
    x2 = np.zeros(100)  
  
    plt.plot(x1, y1, 'blue')  
    plt.plot(x2, y1, 'blue')  
    plt.plot(y2, x2, linestyle='--', color='black')
```

Ce qui donne ce tracer-là :



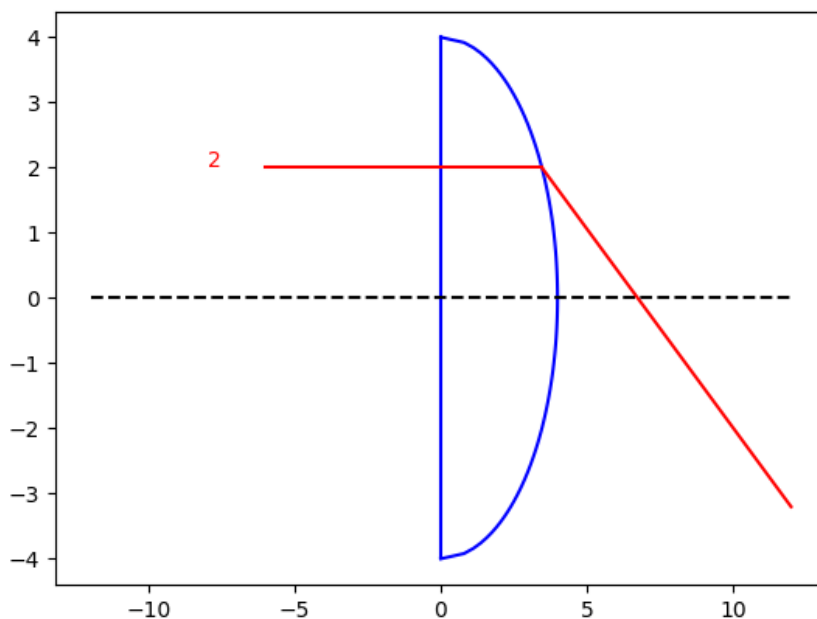
Ensuite, on crée la fonction $tracerrayon(z, d, r, n)$ avec z la hauteur du rayon par rapport à (O_x) , d un diamètre, r un rayon de courbure et n l'indice de réfraction ($n = 1.5$ pour le verre). On a pour un rayon lumineux horizontal d'altitude z :

- L'angle du rayon incident : $i_1 = \arcsin\left(\frac{z}{r}\right)$
- L'existence d'un rayon émergent si : $n \sin(i_1) < 1$
- L'angle du rayon émergent : $i_2 = \arcsin\left(\frac{nz}{r}\right)$
- Coordonnées du point M où se situe la réfraction : $x_M = e - r + r \cos(i_1)$ et $y_M = z$
- Coordonnées du point F qui est sur le rayon émergent : $x_F = 3r$ et $y_F = z - (3r + x_M) \tan(i_2 - i_1)$

On obtient donc la fonction ci-dessous :

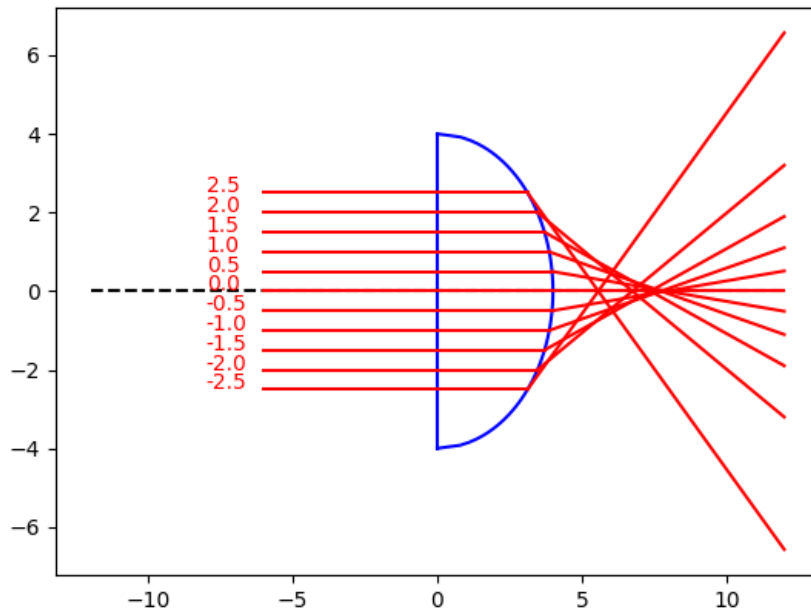
```
def tracerrayon(z,d,r,n):  
  
    e = r - r * np.sqrt(1 - (d**2 / (4 * r**2)))  
    i1 = np.arcsin(z/r)  
  
    if(np.sin(i1)<1):  
        i2 = np.arcsin(n*z/r)  
        xM = e - r + r*np.cos(i1)  
        yF = z - (3*r + xM) * np.tan(i2 - i1)  
  
        x1 = np.linspace(-3/2*r, xM, 100)  
        y1 = np.linspace(z , z, 100)  
        x2 = np.linspace(xM, 3*r, 100)  
        y2 = np.linspace(z, yF, 100)  
  
        plt.plot(x1, y1, color='red')  
        plt.plot(x2, y2, color='red')  
        plt.text(-2*r,z, str(z), color='red')
```

Ce qui donne, avec un rayon avec $z = 2$, ceci :



2. Etude du Stigmatisme :

On peut maintenant tracer plusieurs rayons d'un objet à l'infini afin d'étudier le stigmatisme de la lentille, pour cela on trace ici 10 rayons à hauteurs régulières de 0.5 en 0.5 de $z = -2.5$ à $z = 2.5$. Ce qui donne ceci :



Un système optique est rigoureusement stigmatique si l'image d'un point est un point. Ce qui signifie que chaque rayon émergeant converge en un unique point objet.

On peut donc remarquer ici que tout le rayon émergeant ne converge pas en un seul point, ce n'est donc pas un système optique rigoureusement stigmatique.

Lentille Divergente :

1. Tracer de la lentille et du rayon :

Dans cette partie nous allons refaire les mêmes manipulations pour une lentille divergente.

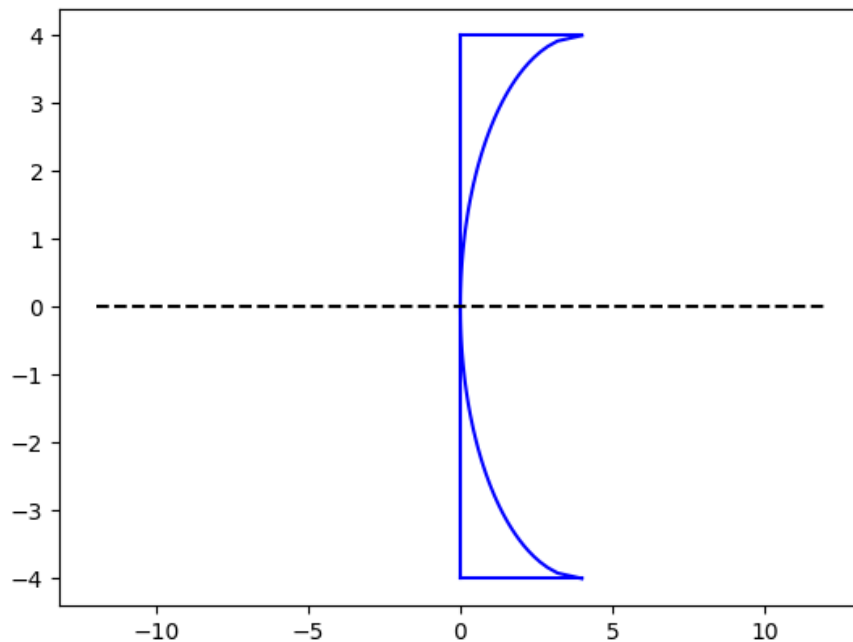
On crée donc une fonction *lentilledivergente*(d, r) avec les mêmes paramètres et la même formule pour l'épaisseur mais une formule différente pour décrire la portion du cercle de la lentille :

- $x(y) = -\sqrt{r^2 - y^2} - r + 2e$

```
def lentilledivergente(d,r):
    e = r - r * np.sqrt(1 - (d**2 / (4 * r**2)))
    y1 = np.linspace(-1/2*d, 1/2*d, 100)
    x1 = -np.sqrt(r**2 - y1**2) - r + 2*e
    y2= np.linspace(-3/2*d, 3/2 *d, 100)
    x2 = np.zeros(100)
    x3 = np.linspace(0, e, 100)
    y3 = np.zeros(100) + d/2
    y4 = np.zeros(100) - d/2

    plt.plot(x1, y1, 'blue')
    plt.plot(x2, y1, 'blue')
    plt.plot(x3, y3, 'blue')
    plt.plot(x3, y4, 'blue')
    plt.plot(y2, x2, linestyle='--', color='black')
```

Ce qui donne ceci :



Ensuite, on crée la fonction $tracerrayon(z, d, r, n)$ avec toujours les mêmes paramètres et les formules suivantes qui ont changés :

- Les coordonnées du point M : $x_M = r - r \cos(i_1)$ et $y_M = z$
- Les coordonnées du point F : $x_F = r$ et $y_F = (r - x_M) \tan(i_2 - i_1) + z$
- Les coordonnées du point G qui est le point d'intersection le prolongement du rayon émergent à gauche de la lentille et l'axe (O_x) : $x_G = x_M - z \frac{r - x_M}{y_F - z}$ et $y_G = 0$

On obtient donc la fonction ci-dessous :

```
def tracerrayon(z,d,r,n):

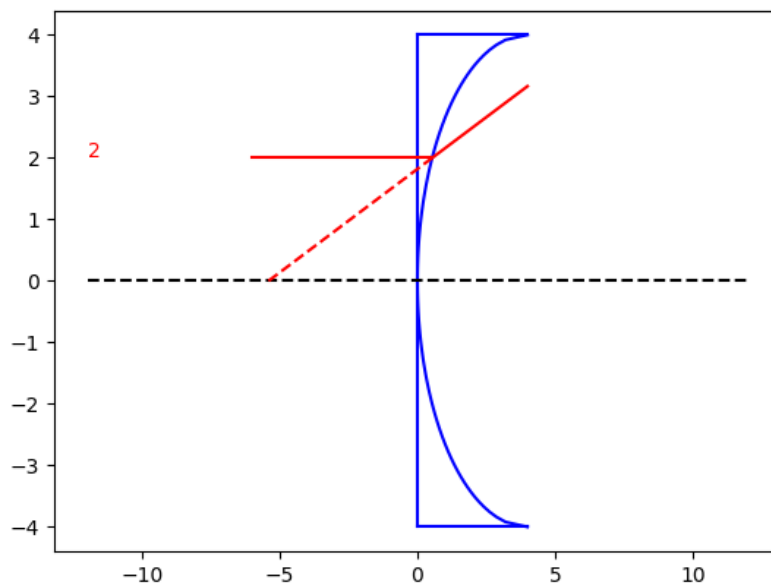
    e = r - r * np.sqrt(1 - (d**2 / (4 * r**2)))
    i1 = np.arcsin(z/r)

    if(np.sin(i1)<1):
        i2 = np.arcsin(n*z/r)
        xM = r - r*np.cos(i1)
        yF = (r-xM)*np.tan(i2-i1) + z
        xG = xM - z*(r-xM)/(yF-z)

        x1 = np.linspace(-3/2*r, xM, 100)
        y1 = np.linspace(z , z, 100)
        x2 = np.linspace(xM, r, 100)
        y2 = np.linspace(z, yF, 100)
        x3 = np.linspace(xG, xM, 100)
        y3 = np.linspace(0, z, 100)

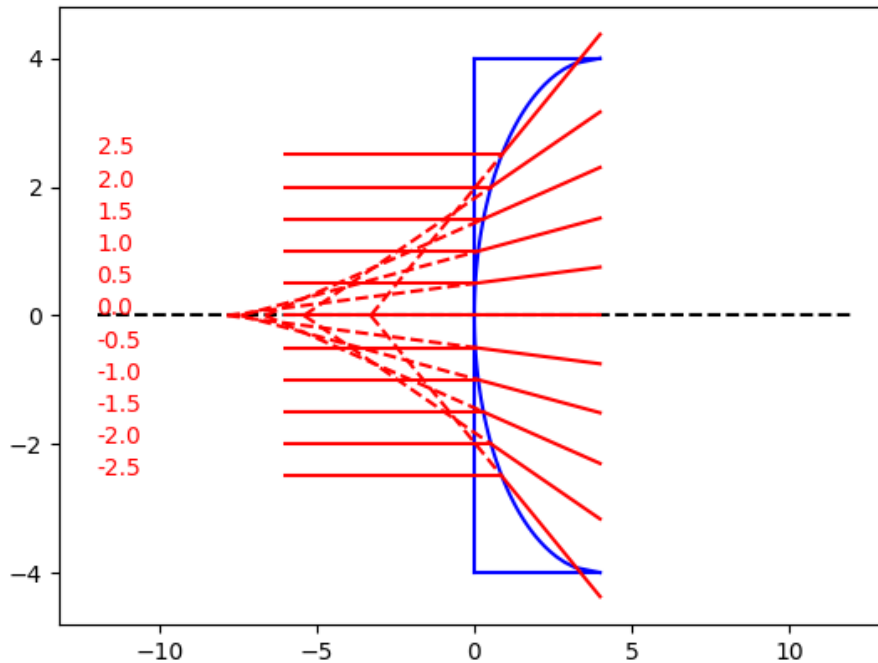
        plt.plot(x1, y1, color='red')
        plt.plot(x2, y2, color='red')
        plt.plot(x3, y3, color='red', linestyle='--')
        plt.text(-3*r,z, str(z), color='red')
```

Ce qui donne ceci :



2. Etude du Stigmatisme :

On peut maintenant tracer plusieurs rayons d'un objet à l'infini afin d'étudier le stigmatisme de la lentille, pour cela on trace les 10 mêmes rayons que pour la lentille convergente. Ce qui donne ceci :



On peut observer ici également que le prolongement des rayons émergeant ne convergent pas en un point unique, le système optique n'est donc pas rigoureusement stigmatique.

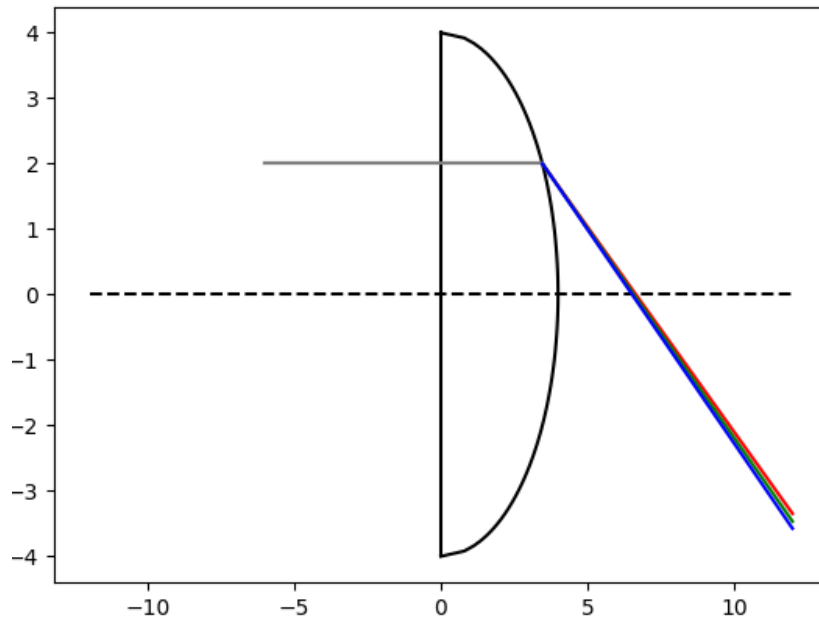
Aberrations chromatiques pour lentilles convergentes :

Le but de cette partie est d'étudier les aberrations chromatiques créées par une lentille. On éclaire la lentille avec un faisceau constitué de trois longueurs d'onde (rouge, vert, bleu). On cherche les images de l'objet à l'infini sur l'axe pour chaque rayon chromatique.

Pour cela on va donc tracer le faisceau lumineux ainsi que les rayons chromatiques émergeant avec ces données :

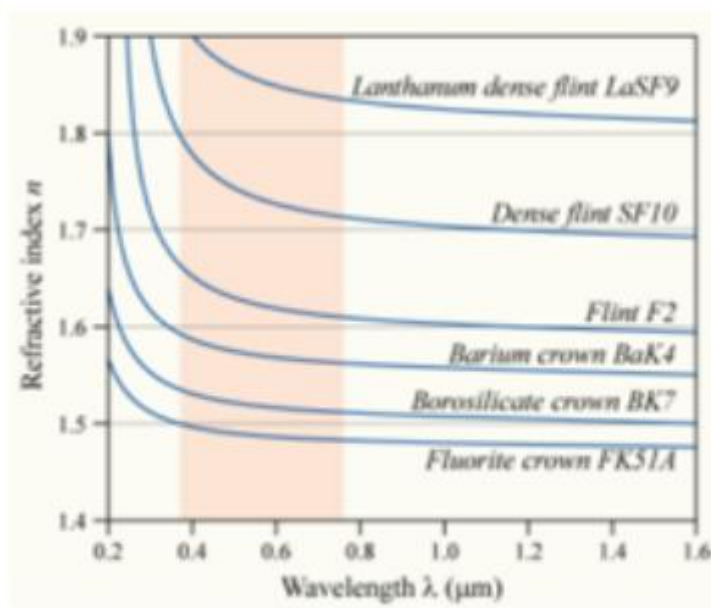
- $n_r = 1.511$ pour $\lambda = 768 \text{ nm}$
- $n_v = 1.520$ pour $\lambda = 532 \text{ nm}$
- $n_b = 1.528$ pour $\lambda = 434 \text{ nm}$

Ce qui donne donc ceci pour $z = 2$:



Alors quel type de verre doit on utiliser pour diminuer les aberrations chromatiques ?

On peut le déterminer avec ce schéma :



On peut observer que dans la zone indiquée, dans laquelle se trouvent les longueurs d'onde des rayons utilisés ici, la meilleure courbe est celle du Fluorite crown FR51A.