
TP-4 : Aberrations des lentilles plan-convexes

– Compte Rendu –

Arthur Le Guludec – MP2I
Lycée Louis Thuillier – 2026/2027

Ce compte rendu porte sur l'étude, à l'aide de Python (numpy et matplotlib), du stigmatisme approché d'une lentille plan-convexe (demi-boule) pour des rayons proches de l'axe optique, puis des aberrations chromatiques.

I) et II) Tracé et trajet des rayons

1) et 2) Tracé de la lentille convergente

On crée la fonction `lentille_convergente(d, r)` qui trace une lentille plan-convexe de diamètre d et de rayon de courbure r . Sa face plane est en $x = 0$ et sa face sphérique est le morceau de cercle d'équation $x(y) = \sqrt{r^2 - y^2} - r + e$, avec l'épaisseur $e = r - r\sqrt{1 - \frac{d^2}{4r^2}}$. On ajoute la ligne de l'axe optique en pointillés avec `plt.axhline`.

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 def lentille_convergente(d, r):
5     """
6     Trace une lentille convergente de diamètre d et de rayon r
7     """
8     e = r - r*np.sqrt(1- (d**2)/(4*r**2))
9     y = np.linspace(-d/2 , d/2 , 1000)
10    x = np.sqrt((r**2) - (y**2) ) - r + e
11
12    plt.plot([0,0], [-d/2 , d/2], color= 'b')
13    plt.plot(x, y, color= 'b')
14    plt.axhline(0, color='black', linestyle="--")
15
16    plt.axis("equal")
```

3) et 4) Trajet d'un rayon à travers la lentille

On considère un objet à l'infini sur l'axe : les rayons incidents sont parallèles à l'axe, à l'altitude z . Ils arrivent sous incidence normale sur la face plane, donc sans déviation, puis se réfractent sur la face sphérique en M . D'après les lois de Descartes :

- $i_1 = \arcsin(z/r)$ et $x_M = e - r + r \cos(i_1)$;
- un rayon émergent existe si $n \sin(i_1) < 1$, avec $i_2 = \arcsin(n \sin i_1)$;

— le rayon émergent, dévié de $i_2 - i_1$, est tracé jusqu'en $x = 3r$: $y_F = z - (3r - x_M) \tan(i_2 - i_1)$.

```
1 def tracer_rayon(z, d, r, n):
2     e = r - r*np.sqrt(1 - d**2/(4*r**2))
3     i1 = np.arcsin(z/r)
4     xM = e - r + r*np.cos(i1)
5
6     plt.plot([-3*r/2, xM], [z, z], color="red")
7     plt.text(-3*r/2, z, f"{round(z, 2)}", ha="right", va="center",
8             fontsize=7)
9
10    if n*np.sin(i1) < 1:
11        i2 = np.arcsin(n*z/r)
12        yF = z - (3*r - xM)*np.tan(i2 - i1)
13        plt.plot([xM, 3*r], [z, yF], color="red")
```

III) Stigmatisme

5) Réseau de 10 rayons

On trace 10 rayons d'altitudes régulièrement espacées entre $-0,3$ et $0,3$ pour une lentille de diamètre $d = 1$, de rayon $r = 1$ et d'indice $n = 1,5$.

```
1 lentille_convergente(1, 1)
2
3 tab_h = np.linspace(-0.3, 0.3, 10)
4
5 for i in tab_h:
6     tracer_rayon(i, 1, 1, 1.5)
7
8 plt.title("Lentille plan-convexe convergente : réseau de rayons (n =
9         1,5)")
10 plt.xlabel("x")
11 plt.ylabel("y")
12 plt.show()
```

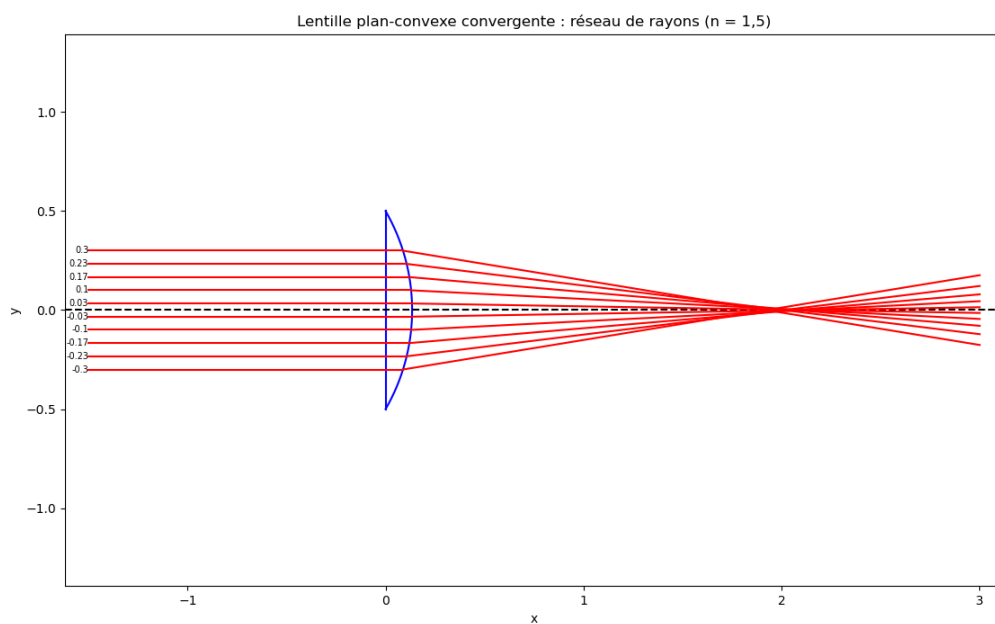


FIGURE 1 – Réseau de 10 rayons à travers la lentille plan-convexe ($n = 1,5$).

6) Stigmatisme rigoureux ?

Un système optique est rigoureusement stigmatique si l'image d'un point est un point. C'est le cas du miroir plan.

Pour notre lentille, le stigmatisme n'est qu'approché. Le zoom autour du point de convergence (figure 2) montre que les rayons ne se croisent pas tous exactement au même point.

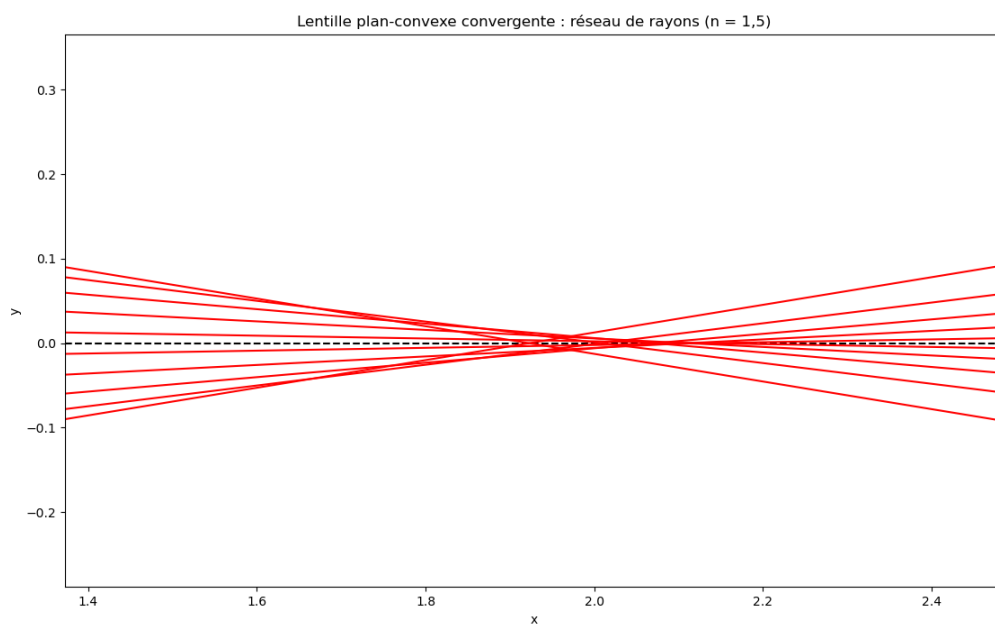


FIGURE 2 – Zoom sur la zone de convergence : les rayons ne passent pas tous par le même point.

IV) Lentille divergente

7) Lentille plan-concave

On refait l'étude avec une lentille divergente. La face plane est en $x = 0$, la face concave a pour centre $S(r; 0)$ et pour équation $x(y) = r - \sqrt{r^2 - y^2}$, avec la même épaisseur de bord e . Le rayon horizontal traverse la face plane, atteint la face courbée en M avec $x_M = r - r \cos(i_1)$, puis est dévié (il s'écarte de l'axe). Le rayon émergent est en deux parties :

- la partie **réelle**, de M jusqu'en $x = r$, d'ordonnée $y_F = (r - x_M) \tan(i_2 - i_1) + z$;
- le **prolongement virtuel** (en pointillés) de M jusqu'à l'axe en G , d'abscisse $x_G = x_M - z \frac{r - x_M}{y_F - z}$.

```
1 def lentille_divergente(d, r):
2     """
3     Trace une lentille divergente de diamètre d et de rayon r
4     """
5     e = r - r*np.sqrt(1- (d**2)/(4*r**2))
6     y = np.linspace(-d/2 , d/2 , 500)
7     x = r - np.sqrt((r**2) - (y**2))
8
9     plt.plot([0,0], [-d/2 , d/2], color= 'b')
10    plt.plot([0,e], [d/2, d/2], color= 'b')
11    plt.plot([0,e], [-d/2, -d/2], color= 'b')
12    plt.plot(x, y, color= 'b')
13    plt.axhline(0, color='black', linestyle="--")
14
15    plt.axis("equal")
16
17 def tracer_rayon_div(z, d, r, n):
18     i1 = np.arcsin(z/r)
19     xM = r - r*np.cos(i1)
20
21     plt.plot([-3*r/2, xM], [z, z], color="red")
22     plt.text(-3*r/2, z, f"{round(z, 2)}", ha="right", va="center",
23             fontsize=7)
24
25     if n*np.sin(i1) < 1:
26         i2 = np.arcsin(n*np.sin(i1))
27         yF = (r - xM)*np.tan(i2 - i1) + z
28         plt.plot([xM, r], [z, yF], color="red") #
29         rayons réels
30         xG = xM - z*(r - xM)/(yF - z)
31         plt.plot([xM, xG], [z, 0], color="red", linestyle=":") #
32         rayons virtuels
33
34     lentille_divergente(1, 1)
35     tab_h2 = np.linspace(-0.3, 0.3, 10)
36
37     for i in tab_h2:
38         tracer_rayon_div(i, 1, 1, 1.5)
39
40     plt.title("Lentille plan-concave divergente (pointillés : prolongements
41             virtuels)")
42     plt.xlabel("x")
43     plt.ylabel("y")
44     plt.show()
```

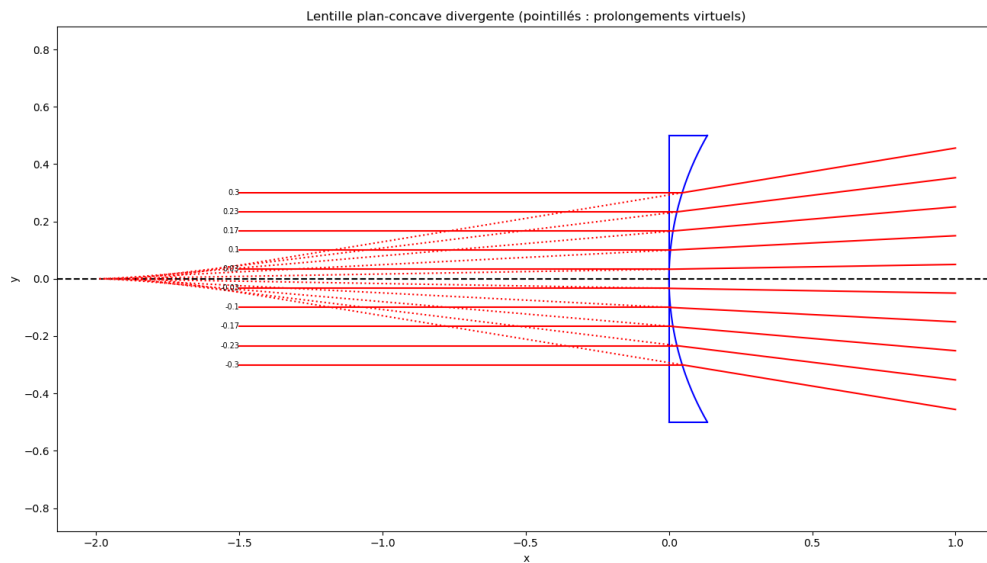


FIGURE 3 – Réseau de rayons à travers la lentille divergente : les rayons émergents s'écartent de l'axe et leurs prolongements (pointillés) se coupent à gauche de la lentille.

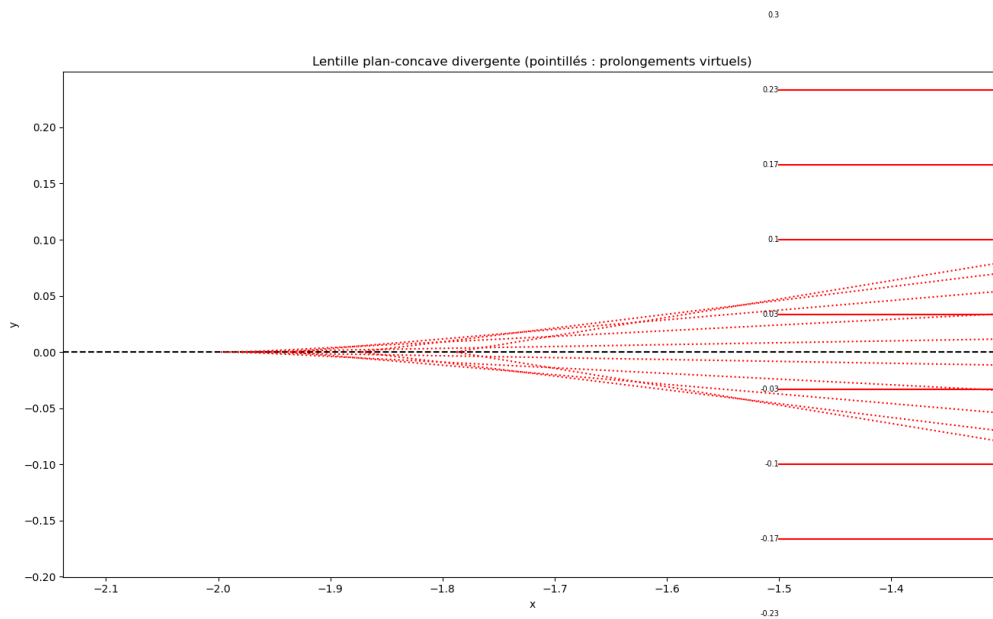


FIGURE 4 – Zoom sur la zone où les prolongements virtuels coupent l'axe.

La lentille est bien divergente : les rayons émergents s'écartent de l'axe et semblent provenir d'un foyer image virtuel situé avant la lentille, aux environs de $x \approx -2$.

V) Aberrations chromatiques

8) Rayons de trois longueurs d'onde

Le verre est dispersif : son indice dépend de la longueur d'onde. On éclaire la lentille convergente avec un rayon horizontal d'altitude $z = 0,3$ pour trois longueurs d'onde, avec les indices :

- $n_r = 1,511$ pour $\lambda = 768$ nm (rouge) ;
- $n_v = 1,520$ pour $\lambda = 532$ nm (vert) ;
- $n_b = 1,528$ pour $\lambda = 434$ nm (bleu).

La fonction `rayon_couleur` trace le rayon jusqu'à son intersection avec l'axe, d'abscisse $x_F = x_M + \frac{z}{\tan(i_2 - i_1)}$.

```
1 def rayon_couleur(z, d, r, n, couleur):
2     """Trace le rayon coloré jusqu'à son intersection avec l'axe"""
3     e = r - r*np.sqrt(1 - d**2/(4*r**2))
4     i1 = np.arcsin(z/r)
5     xM = e - r + r*np.cos(i1)
6     i2 = np.arcsin(n*np.sin(i1))
7     xF = xM + z/np.tan(i2 - i1)
8     plt.plot([-3*r/2, xM], [z, z], color=couleur)
9     plt.plot([xM, xF], [z, 0], color=couleur)
10    plt.plot(xF, 0, 'o', color=couleur, markersize=4)
11
12    lentille_convergente(1, 1)
13    tab_couleurs = [(1.511, "red"), (1.520, "green"), (1.528, "blue")]
14    for settings in tab_couleurs:
15        rayon_couleur(0.3, 1, 1, settings[0], settings[1])
16
17    plt.title("Aberrations chromatiques (z = 0,3)")
18    plt.xlabel("x")
19    plt.ylabel("y")
20    plt.show()
```

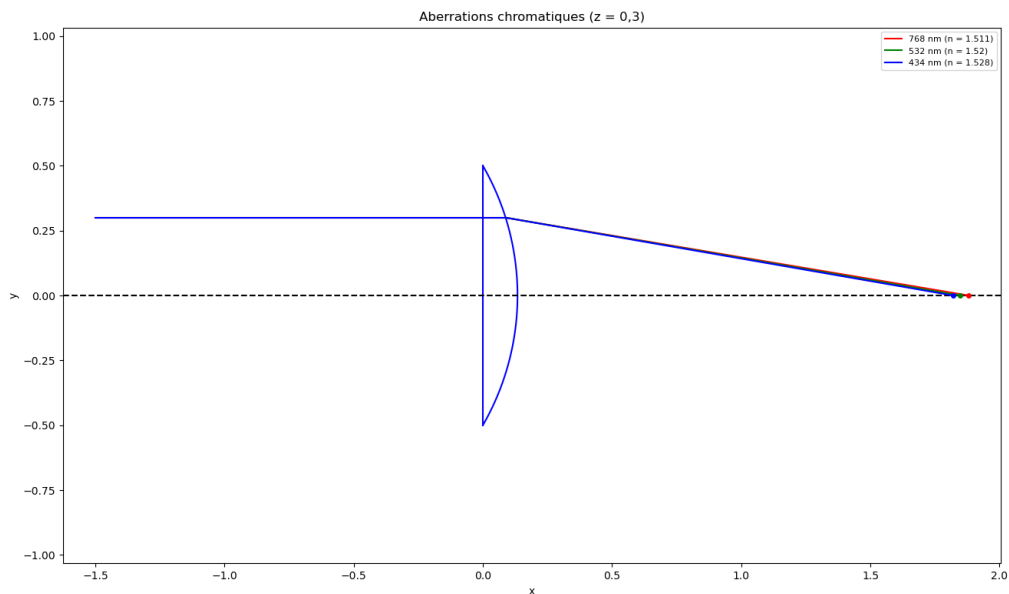


FIGURE 5 – Rayons rouge, vert et bleu à travers la lentille.

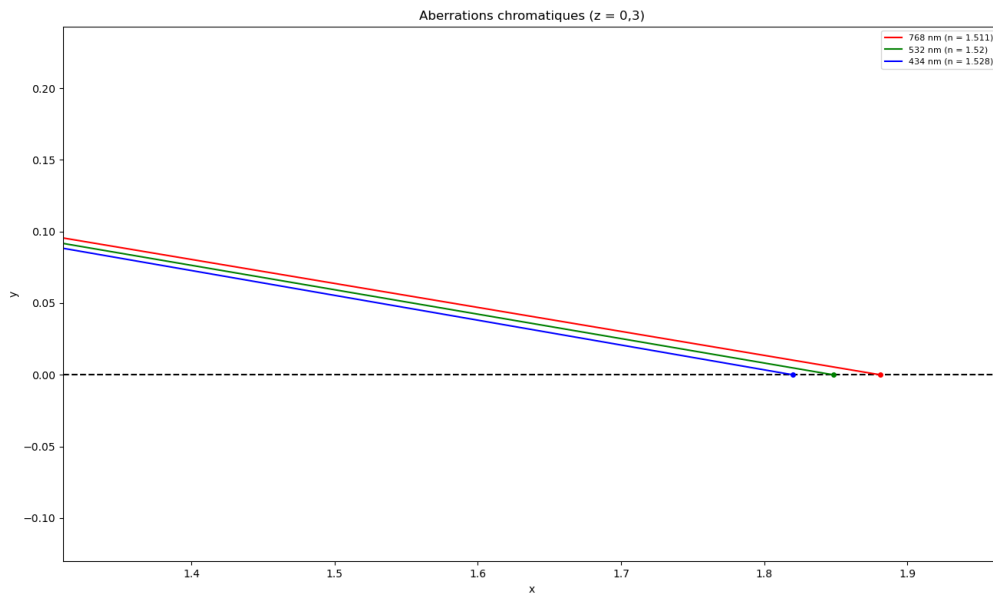


FIGURE 6 – Zoom sur les points d’intersection avec l’axe pour les trois couleurs.

Les trois rayons coupent l’axe en des points différents :

Couleur	rouge (768 nm)	vert (532 nm)	bleu (434 nm)
x_F	1,881	1,848	1,820

Plus l’indice est élevé, plus le rayon est dévié : le bleu converge le plus près de la lentille, le rouge le plus loin.

9) Choix du verre pour limiter les aberrations chromatiques

Pour avoir des aberrations chromatiques faibles, il faut que, sur le domaine visible (la zone colorée sur le schéma), la courbe $n(\lambda)$ soit la plus constante possible, c’est-à-dire que l’indice de réfraction change le moins possible. Pour cela, il faut se tourner du côté des verres crown : le meilleur est le Fluorite crown FK51A, car c’est le plus constant sur le domaine visible.