

TP 4 : Aberration de lentilles plans convexes

Question 1 / 2 :

On trace une lentille convergente avec l'axe optique

def lentilleconvergente(d,r):

```
y = np.linspace(-d / 2, d / 2,100)
e = r - r * np.sqrt(1-( d**2 )/ (4*r**2))
```

```
x = np.sqrt(r**2 - y**2) - r + e
```

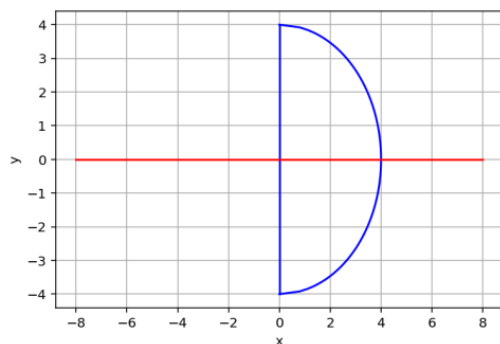
```
plt.plot(x, y, color="blue")
plt.plot (np.zeros(100), y, 'b')
```

```
plt.plot([-d, d], [0, 0], color="red", linestyle="-")
```

```
plt.xlabel("x")
plt.ylabel("y")
plt.grid()
```

```
plt.show()
```

lentilleconvergente(8,4)



Question 3/4

On ajoute à la lentille le tracé d'un rayon incident et du rayon émergent.

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

```
    i1 = np.arcsin(z/r)
```

```
    i2 = np.arcsin(nz/r)
```

```
    e = r - r * np.sqrt(1-( d**2 )/ (4r**2))
```

```
    xm = e - r + r * np.cos(i1)
```

```
    yf = z - (3r + xm)*np.tan(i2-i1)
```

```
    x1 = np.linspace((-3/2)*r,xm,100)
```

```
    x2 = np.linspace(xm,5*r,100)
```

```
    y1 = z*np.ones(100)
```

```
    y2 = (x2-xm)*(yf-z)/(3*r-xm)+z
```

```
    plt.plot(x1,y1,"-")
```

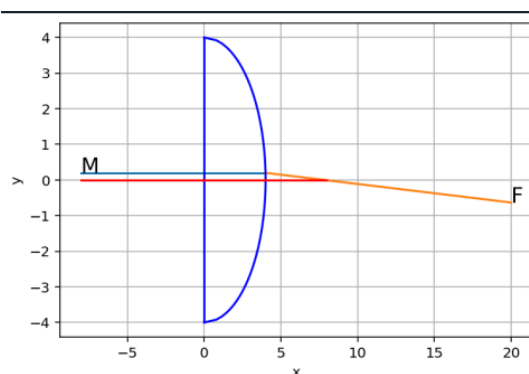
```
    plt.plot(x2,y2,"-")
```

```
    plt.text(x1[0],y1[0],"M", size = 15)
```

```
    plt.text(x2[-1],y2[-1],"F", size = 15)
```

```
    lentilleconvergente(d,r)
```

```
    plt.show()
```



Question 5

On trace un réseau de 10 rayons à hauteurs différentes.

```
hauteurs = np.linspace(-3, 3, 10)
```

```
for z in hauteurs :
```

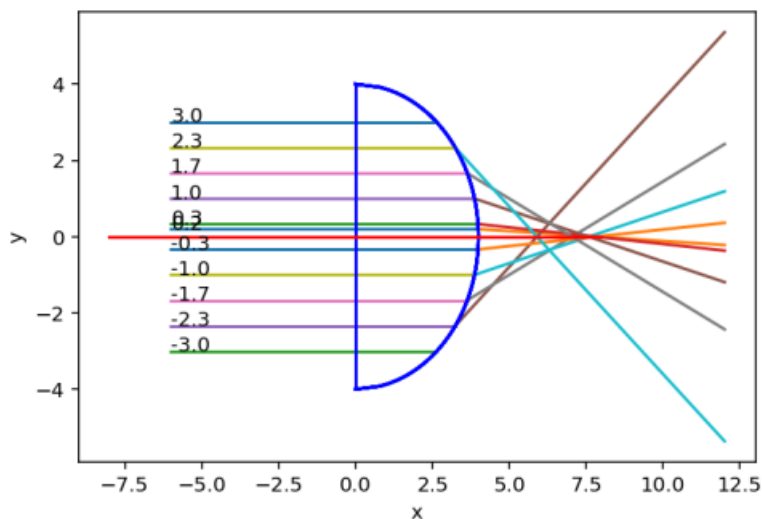
```
    tracerrayon(z,8,4,1.52)
```

```
plt.show()
```

On modifie légèrement la fonction tracerrayon pour afficher les différentes altitudes.

On ajoute :

```
plt.text(x1[0],y1[0],round(z,1), size = 10)
```



Question 6.

Pour avoir un stigmatisme rigoureux il faut un miroir plan comme système optique.

Question 7

On refait la même chose que précédemment pour une lentille divergente.

```
def lentilleconvergente(d,r):
```

```
    y = np.linspace(-d / 2, d / 2,100)
    e = r - r * np.sqrt(1-( d**2 )/ (4*r**2))
```

```
    x = np.sqrt(r**2 - y**2) - r + e
```

```
    plt.plot(x, y, color="blue")
    plt.plot (np.zeros(100), y, 'b')
```

```
    plt.plot([-d, d], [0, 0], color="red", linestyle="-")
```

```
    plt.xlabel("x")
    plt.ylabel("y")
    plt.grid()
```

```
lentilleconvergente(8,4)
```

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

```
    i1 = np.arcsin(z/r)
```

```
    i2 = np.arcsin(nz/r)
```

```
    xm = r - r * r * np.cos(i1)
```

```
    yf = (r - xm)np.tan(i2-i1) + z
```

$$xg = xm - z((r-xm)/(yf - z))$$

$$e = r - r * np.sqrt(1 - (d**2)/(4r**2))$$

$$x2 = np.linspace(r-e, 3*r, 100)$$

$$x1 = np.linspace((-3/2)*r, r-e, 100)$$

$$y1 = z * np.ones(100)$$

$$y2 = (x2-xm)*(yf-z)/(3*r-xm)$$

$$xg1 = np.linspace((-3/2)*r, r-e, 100)$$

$$yg = (xg1 - xm) * (yf - z) / (3*r - xm)$$

$$\text{plt.plot}(x1, y1, \text{"-"})$$

$$\text{plt.plot}(xg1, yg, \text{"--"})$$

$$\text{plt.plot}(x2, y2, \text{"-"})$$

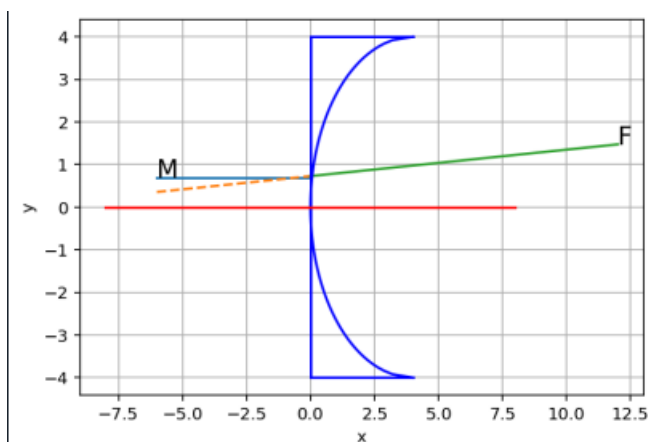
$$\text{plt.text}(x1[0], y1[0], \text{"M"}, \text{size} = 15)$$

$$\text{plt.text}(x2[-1], y2[-1], \text{"F"}, \text{size} = 15)$$

$$\text{lentilledivergente}(d, r)$$

$$\text{plt.show}()$$

tracerrayondiver(0.7,8,4,1.52)

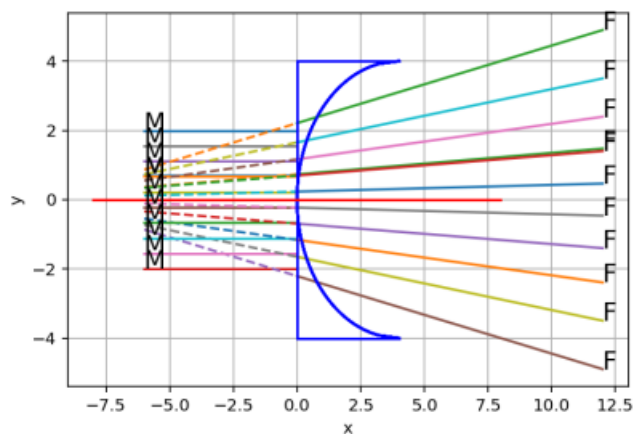


```
hauteurs = np.linspace(-2, 2, 10)
```

```
for z in hauteurs:
```

```
    tracerrayondiver(z,8,4,1.52)
```

```
plt.show()
```



Question 8

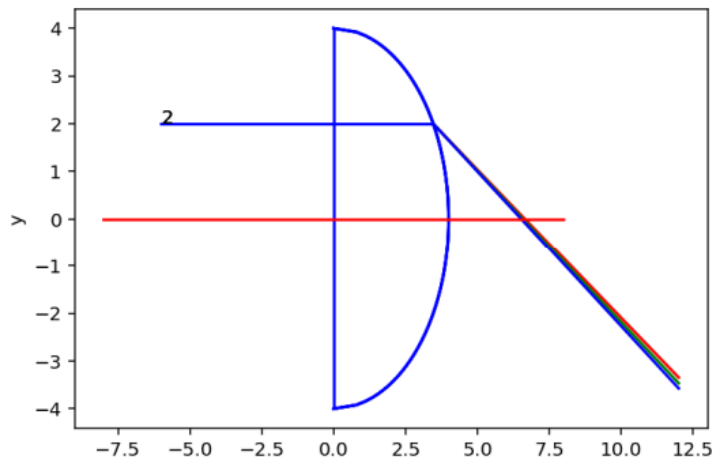
On trace les rayons lumineux de différentes couleurs (rouge, vert, bleu).

```
tracerrayon(2,8,4,1.511,"red")
```

```
tracerrayon(2,8,4,1.520,"green")
```

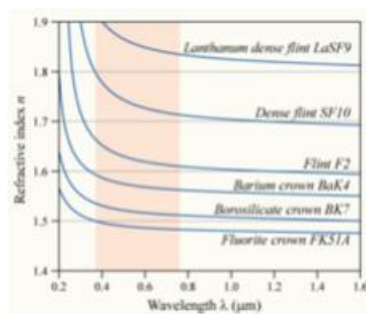
```
tracerrayon(2,8,4,1.528,"blue")
```

```
plt.show()
```



Question 9

En fonction du document on détermine le type de verre pour diminuer les aberrations chromatiques



Plus l'indice est élevé plus le rayon est réfracté. Le Fluorite crown est le type de verre diminuant le plus les aberrations chromatiques.