Capacité numérique : Demi-boule

Proposition de script Python :

```
# Stigmatisme d'une lentille demi-boule

# Importation des bibliothèques utiles
# numpy pour la constitution des tableaux (colonnes, etc)
# et matplotlib.pyplot pour les graphes

import numpy as np
import matplotlib.pyplot as plt


# Les constantes du problème
n = 1.5 # C'est l'indice du milieu de la lentille
R = 5 # C'est le rayon en cm de la lentille


# Définition des fonctions utiles

# Définition de l'angle d'incidence angleInc (noté i dans l'étude théorique) en fonction de yI
def angleInc ( yI ) :
        return -np.arcsin ( yI / R )

# Définition de l'angle de réfraction angleRef (noté r dans l'étude théorique) en fonction de yI
ylim = R/n

def angleRef ( yI ) :
        if abs ( yI ) >  ylim:
                return None
        else:
                return np.arcsin ( n*np.sin ( angleInc ( yI ) ) )

# Définition de l'angle de déviation D
def angleD ( yI ) :
        if abs ( yI ) >  ylim:
                return None
        else:
                return angleRef ( yI ) – angleInc ( yI )

# Abscisse du point I
def xI (yI) :
        return np.sqrt ( R**2 - yI**2 )

# Partie graphique proprement dite

plt.plot ( [0,0] , [-R, R] , 'k-')     # on trace le dioptre d'entrée
                                       # On trace un trait qui va du point (0, -R) au point (0, R).

yS = np.linspace (-R, R, 500)
plt.plot ( xI(yS) , yS , 'k-')                 # on trace le dioptre de sortie
```

```
for yI in [k*R/10 for k in range(-10, 11)]:
        plt.plot ( [-1, xI(yI)] , [yI, yI] , 'b-')        #On trace les rayons à gauche du dioptre sphérique.
        if abs (yI) < ylim:
                y2 = yI + np.tan (angleD (yI) ) * (20 – xI (yI) )    #On trace les rayons à droite du
                                                                      dioptre sphérique.
                plt.plot ([xI(yI), 20], [yI, y2], 'b-')

plt.xlabel ("x (cm)")
plt.ylabel ("y (cm)")
plt.title ("Tracé des rayons lumineux pour une lentille demi-boule")
plt.axis ('scaled')          #pour avoir la même échelle sur les deux axes
plt.show ()
```