Lycée A. Schweitzer MULHOUSE

Document Annexe

OVER THE RAINBOW

Louis François , Luna Pearon, Agathe Pospiech,

Mathilde Roecklin-Mihé, Elise Westrich, Chloé Zirone

3.Angle d'observation de l'arc

3.1 Mesures dans le jardin

Calcul du débit d'eau :

Nous avons commencé par calculer le débit de l'eau. Il dépend de la vitesse d'écoulement et de la pression d'eau.

Combien de temps faut-il pour remplir 1L ?

5'12 ; 5'33 ; 5'34 : Moyenne = 5'26

Il faut savoir qu'un débit s'exprime en m^3/h

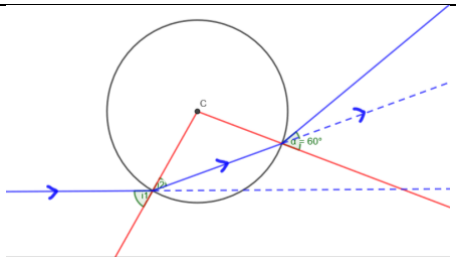
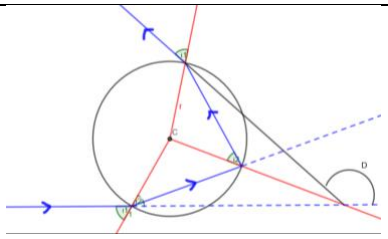
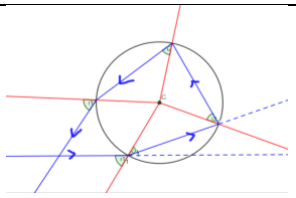
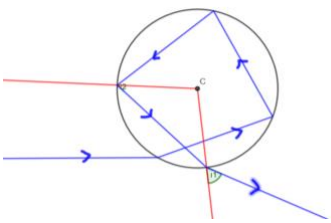
1L en 5'26 soit 60/5'26 = 11,41 L en 1 minute

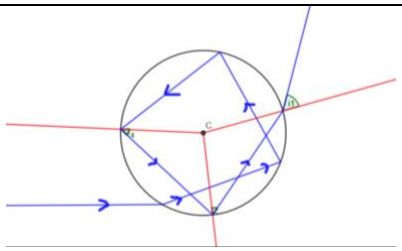
11,41 x 60 = 684,6 L en 1 h

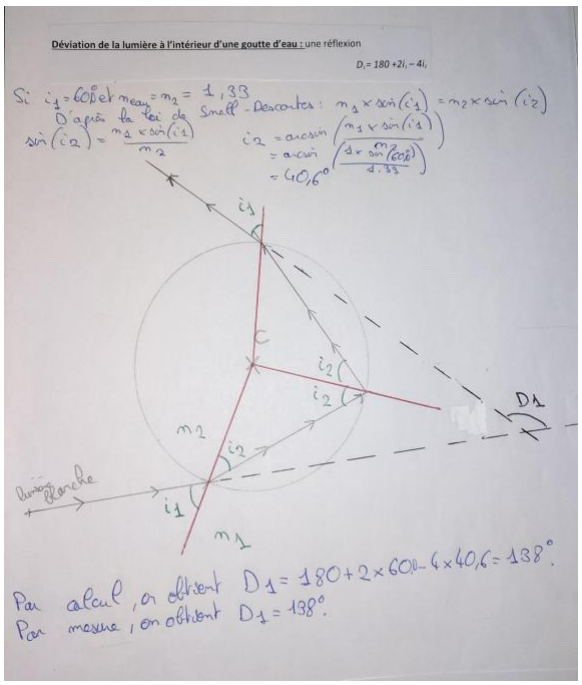
= 0,6846 m^3/h

3.3 Explications théoriques sur la déviation

Schémas et calculs théoriques :

Nombre de réflexions internes	Schéma des rayons	Expression de la déviation (°)
0		$D_0 = (i_1 - i_2) \times 2$
1		$D_1 = 2\,(i_1 - i_2) + (180 - 2i_2)$ $D1 = 180 + 2i_1 - 4i_2$
2		$D_2 = 360 - 6i_2 + 2i_1$
3		$D_3 = 540 - 8i_2 + 2i_1$

4		$D_4 = 720 - 10i_2 + 2i_1$

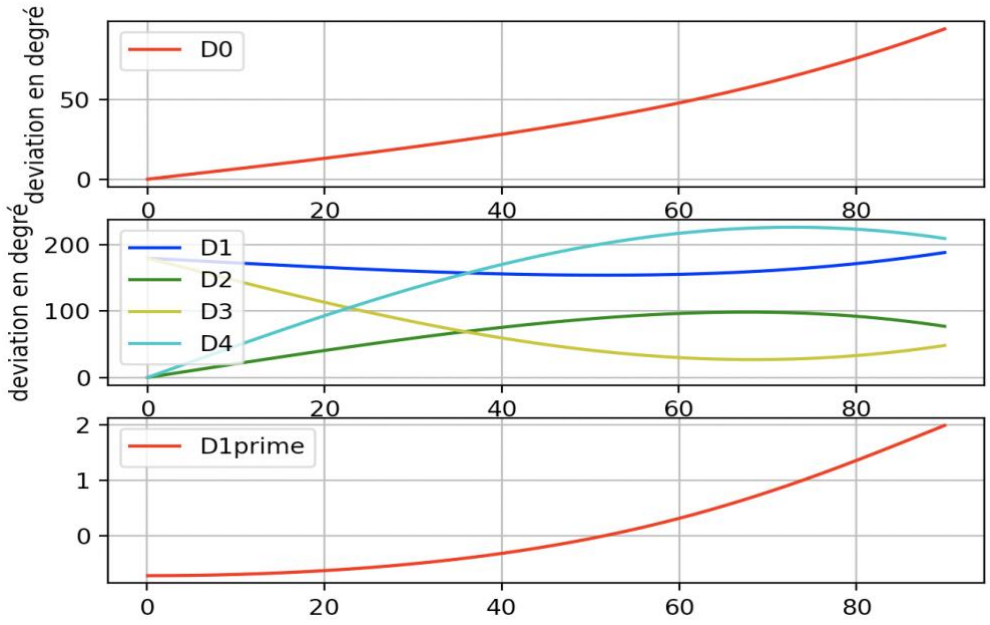


Représentation Python :

i1 minimum de déviation glycérine : 51,48

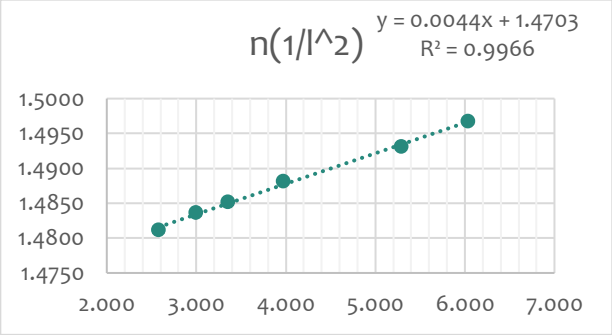
d1 minimum de déviation glycérine : 154,33

Représentation obtenue pour la glycérine



4.2. Etude de l'indice de la glycerine au goniomètre :

	Glycérine					Delta n	q0 =215°25'10"
		q				3,00E-04	
	l (mm)	degrés	minutes	secondes	Delta(")	n	1/l^2 (mm-2)
ROUGE	0,623	180	48	30		1,4811	2,576
JAUNE	0,578	180	36	10		1,4836	2,993
vert	0,546	180	28	30	30	1,4851	3,354
INDIGO	0,502	180	13	40		1,4881	3,968
BLEU	0,435	179	48	50		1,4931	5,285
VIOLET	0,407	179	30	40		1,4967	6,037



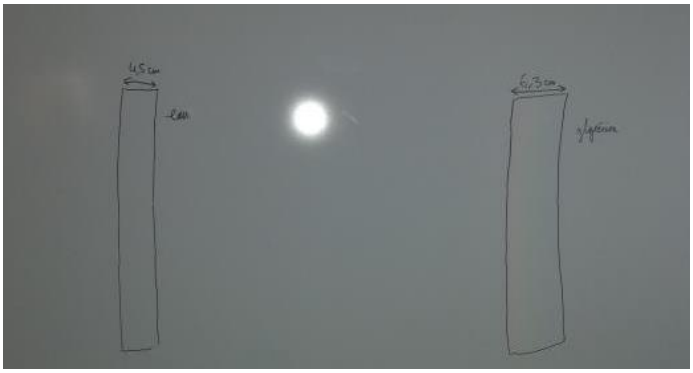
On obtient $n = A + B/l^2$

Avec A = 1,4703 et B = 4,4.e-3 mm^2 : on constate bien que la glycérine est plus dispersive que l'eau (B_{eau} = 3,2.e-3 mm^2).

4.1 Mise en évidence de la dispersion de la lumière et du minimum de déviation :

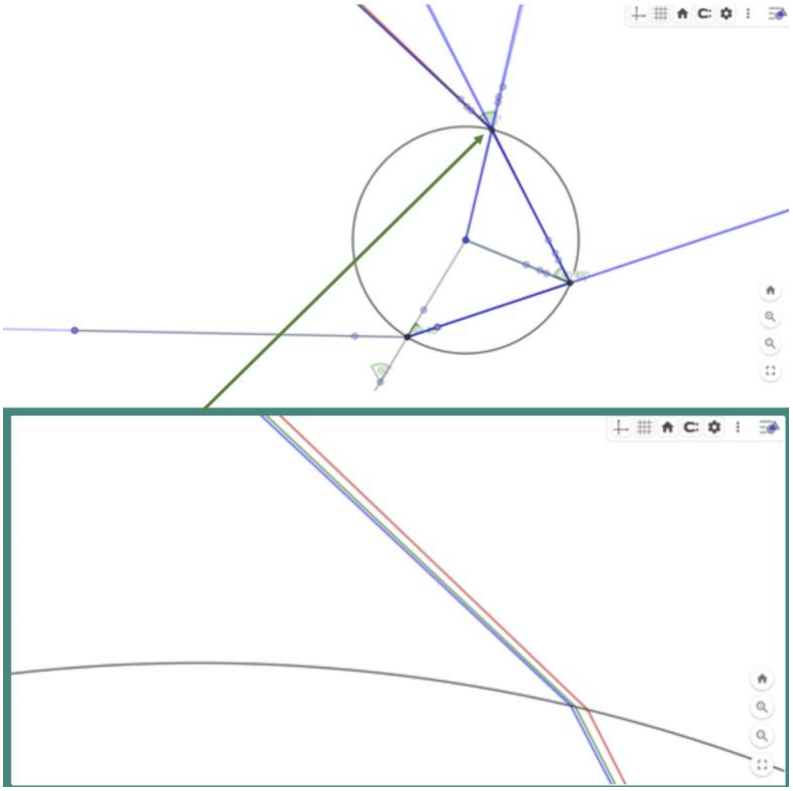
En plus des prismes à eau, Flint et Crown, nous avons utilisé un prisme à glycérine.

Spectre de la lampe avec un prisme à glycérine par rapport à un rayon non dévié (fente blanche) et au spectre de l'eau (rectangle dessiné)

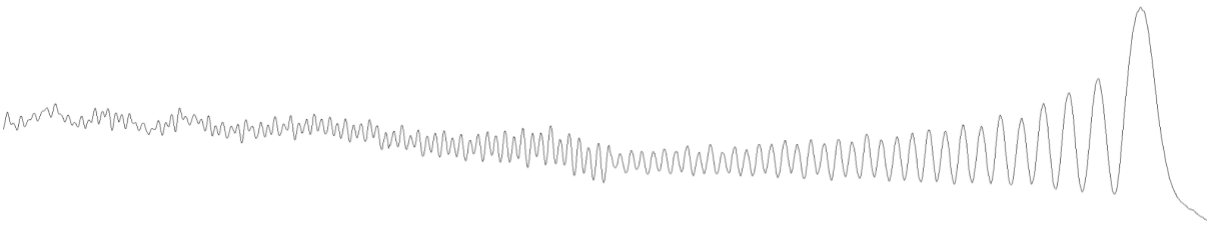


Le spectre est plus étalé avec le prisme à glycérine qu'avec le prisme à eau.

4.3 Visualisation avec Geogebra



5.3 Arcs surnuméraires : résultats obtenus avec la glycérine :



Courbe Caliens : intensité en fonction de la position sur le capteur.

Collage des franges d'interférence obtenues avec Caliens pour la goutte de glycérine afin de visualiser le nombre de franges. On peut en compter au moins 80.

La goutte est plus stable et les interférences sont bien plus faciles à obtenir. On voit bien le nombre de franges d'interférences qui est très élevé.

5.6. Script Python :

```
Tracé des déviations pour les minimas d'intensité théoriques

"""
import numpy as np
from scipy.optimize import fsolve
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
from matplotlib.offsetbox import TextArea, DrawingArea, OffsetImage, AnnotationBbox

#franges = mpimg.imread('/Users/Shared/Professionnel/INFO/Python/Optique/photo1.png')

n=1.33#indice de la goutte
a = 816#rayon de la goutte en um
print("loading...")
l=.632#longueur d'onde en um
i0=np.arcsin(np.sqrt((4-n**2)/3))#1.04#angle d'incidence à la déviation minimale
r0=np.arcsin(np.sin(i0)/n)#.70#angle de réfraction à la déviation minimale

i1=np.linspace(i0,73*np.pi/180,10000);#i1 en radians
r1=np.arcsin(np.sin(i1)/n)#r1 en radians

#recherche des valeurs de i2 qui donnent une même déviation que i1
#l'équation à résoudre est D(i2)=D(i1)
def myFunction(z):
   D2= (np.pi+2*i-4*np.arcsin(np.sin(i)/n))*180/np.pi
   F = D2-(np.pi+2*z-4*np.arcsin(np.sin(z)/n))*180/np.pi
   return F

i2=[]
r2=[]

#on parcourt i1 et on associe à chaque valeur le i2 qui donne la même dévia
for i in i1:
   x = fsolve(myFunction,2*i0-i)[0]
   i2.append(x)
   r2.append(np.arcsin(np.sin(x)/n))

delta=(-4*n*(np.cos(r2)-np.cos(r1))+2*(np.cos(i2)-np.cos(i1)))*a
ordre = delta/l
intensité = .5*(1+np.cos(2*np.pi*ordre))


#recherche des valeurs de i1 donnant un minimum d'intensité.
i=0
etat=0  # etat=0 : intensité est loin de 1, etat=1 : intensité proche de 1
        #vu la courbe, au depart on est proche de 1, donc on cherche localement
        # le max
m=1   #le minimum
i1minTh=[] #liste des i1 théoriques donnant les minimas locaux de l'intensité

while i<len(i1) :
    if etat==1 and intensité[i]<m :
        m=intensité[i]
        imin=i
    if etat==1 and intensité[i]<0.01 :
        etat=0
        i1minTh.append(i1[imin])
        m=1
    if etat==0 and intensité[i]>=0.01 : #on approche de nouveau d'un max
                                       #on recommence la recherche du max
        etat=1
    i+=1
```

```
#Liste des valeurs des déviations théoriques
DminTh=[]
for i in i1minTh :
    DminTh.append((np.pi+2*i-4*np.arcsin(np.sin(i)/n))*180/np.pi)

#Création des abcisses pour les déviations théoriques
b=np.linspace(1,len(DminTh),len(DminTh))

#Liste des valeurs des déviations expérimentales
DminExp=[138.450,138.839,139.147,139.361,139.653,139.894,140.125,
140.342,140.583,140.833,140.981,141.211,141.350,141.556,141.758,
141.969,142.125,142.311,142.483,142.675]
DminExpDeux = [137.647,138.367,138.772,139.103,139.372,139.625,
               139.878,140.092,140.336,140.542,140.733,140.911,
               141.114,141.311,141.492,141.675,141.842,142.011
]
#Création des abcisses pour les déviations théoriques
c=np.linspace(1,len(DminExpDeux),len(DminExpDeux))
e=np.linspace(1,len(DminExp),len(DminExp))

d = []
for i in range(len(DminExpDeux)) :
    d.append(0.09)

f = []

for i in range(len(DminExp)) :
    f.append(0.02)

plt.clf()
plt.plot(b,DminTh,"ob",label="DminTh(°)")
plt.plot(c,DminExpDeux,"xr",label="DminExpDeux(°)")
plt.errorbar(c,DminExpDeux,fmt= 'None',yerr=d,color='red')
plt.xlabel('Numéro de la frange sombre')
plt.ylabel('Déviation(rad)')
plt.legend()
plt.grid(which='major', color='#DDDDDD', linewidth=1)
plt.grid(which='minor', color='#EEEEEE', linestyle=':', linewidth=0.7)
plt.minorticks_on()
plt.show()
```