

TP12 : Goniomètre à prisme

ENONCÉ

En utilisant un goniomètre, un prisme et les lampes spectrales mises à disposition, vérifier la loi de Cauchy.

1 Réglage du Goniomètre

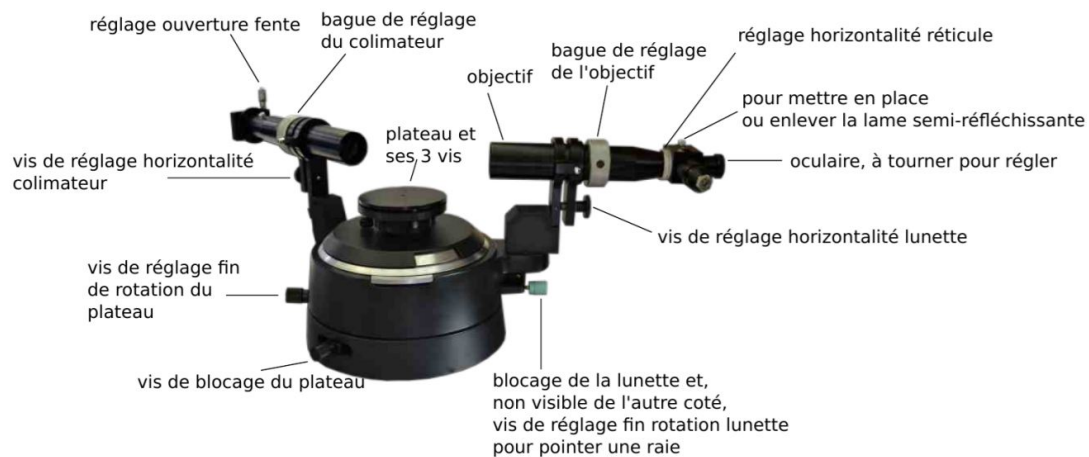


Figure 1 – Présentation goniomètre

Le Goniomètre est constitué :

- ✗ D'une lunette, comprenant un oculaire et un objectif,
- ✗ D'un collimateur devant lequel on place une source lumineuse,
- ✗ D'un plateau sur lequel on place soit un réseau soit un prisme.

1.1 Réglage de l'oculaire

- ✗ L'oculaire est modélisé par une lentille convergente : il joue le rôle d'une loupe.
- ✗ Pour un œil normal¹, on règle l'oculaire en plaçant le plan du réticule² dans le plan focal objet de l'oculaire. Pour cela, il suffit d'éclairer le réticule grâce à l'éclairage de l'autocollimation³ et de l'observer directement dans le réticule comme représenté figure 2.

1. Le principe du réglage de l'oculaire pour un œil quelconque est l'observation au repos afin de ne pas fatiguer les yeux de l'observateur. Pour un œil normal (œil emmétrope), cette observation se fait à l'infini

2. Le réticule est un croix gravée dans une lame transparente, lame perpendiculaire à l'axe optique de la lunette

3. Le réticule est éclairé par une lampe placée sur le coté du tube et grâce à une lame semi-réfléchissante : il faut penser à la retirer une fois le réglage du goniomètre fini (tirette)

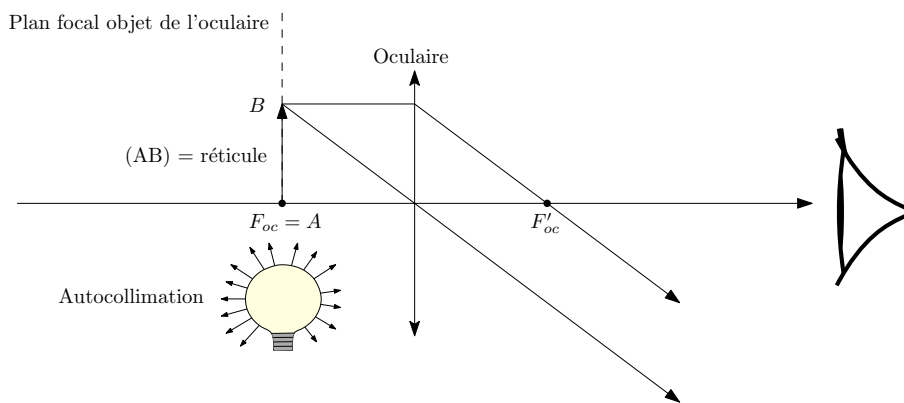


Figure 2 – Principe d'utilisation d'une loupe ou d'un oculaire (œil emmétrope)

✘ Principe du réglage :

- On tire l'oculaire jusqu'à ce que l'image du réticule soit floue : la distance entre le plan du réticule et le plan de l'oculaire étant alors supérieure à la distance focale de l'oculaire, l'image du réticule est au niveau de l'œil de l'observateur, voire au delà , elle ne peut donc pas être vue nettement comme représenté figure 3.

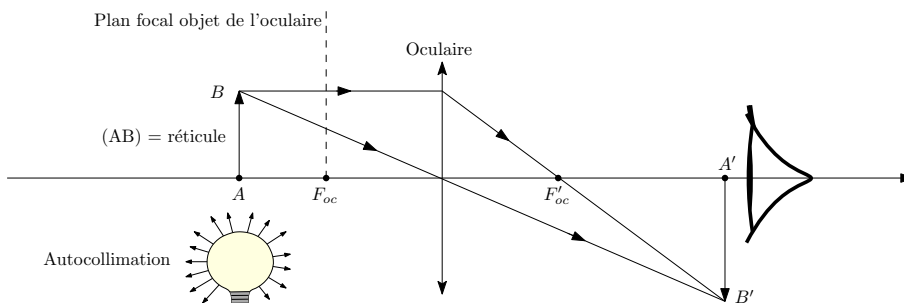


Figure 3 – Cas où l'oculaire est trop « tiré »

- On repousse doucement l'oculaire. La position de l'oculaire pour laquelle l'image du réticule redevient nette correspond au cas où le plan focal objet de l'oculaire est dans le plan du réticule comme représenté figure 2.
- Si l'on continue à repousser l'oculaire, le réticule paraît net (voir figure 4), mais l'œil accommode ce qui induit une fatigue de l'œil.

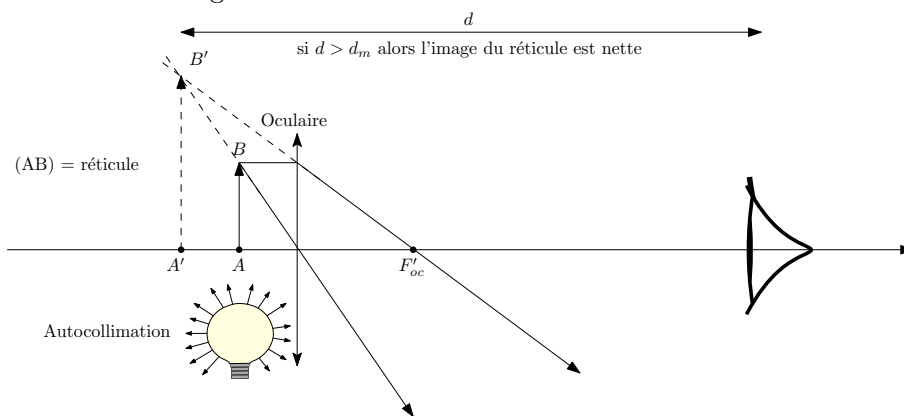


Figure 4 – Cas où l'oculaire est trop « poussé »

1.2 Réglage de l'objectif

- ✘ L'objectif est modélisé par une lentille convergente.

- ✘ Lorsque le système objectif-oculaire est réglé il forme une lunette astronomique et donc un système afocal. De fait, pour régler l'objectif, il faut placer le plan focal image de l'objectif dans le plan focal objet de l'oculaire soit le plan du réticule, comme représenté figure 5.

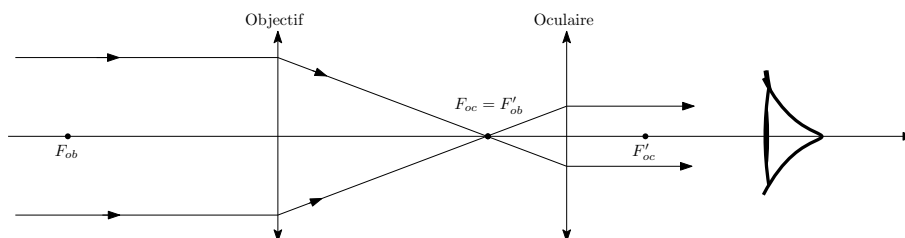


Figure 5 – Modélisation de la lunette

- ✘ Pour régler l'objectif, on utilise l'autocollimation : le réticule éclairé envoie des rayons lumineux en direction de l'oculaire (quand l'observateur regarde dans l'oculaire, il voit un premier réticule issu du réglage de l'oculaire), mais aussi en direction de l'objectif. Ainsi, le réticule joue le rôle d'objet pour l'objectif. Pour le placer dans le plan focal (objet) de l'objectif, il suffit de procéder comme pour autocollimation classique : on place à la sortie de la lunette un miroir plan qui renvoie les rayons lumineux émis par le réticule éclairé. Lorsque le réticule est dans le plan focal objet de l'objectif, un deuxième réticule apparaît net pour l'observateur, comme représenté figure 6.

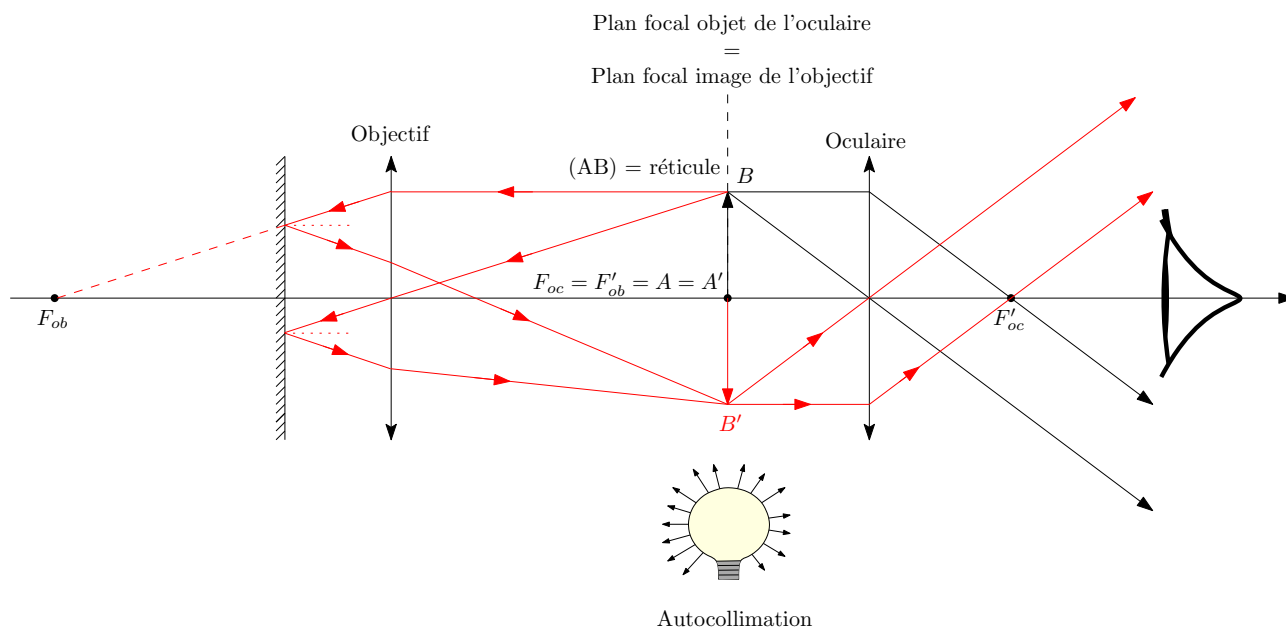


Figure 6 – Réglage de l'objectif

1.3 Réglage du collimateur

- ✘ Le collimateur est constitué d'une fente réglable et d'une lentille convergente. On utilise le collimateur en plaçant une lampe devant la fente.
- ✘ Le collimateur est réglé lorsque que le plan de la fente coïncide avec le plan focal objet de la lentille du collimateur, comme représenté figure 7.

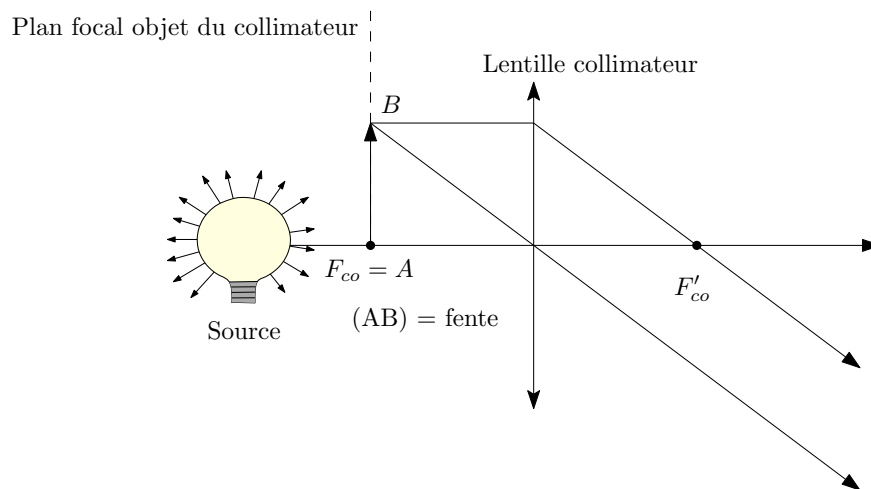


Figure 7 – Modélisation du collimateur

- ✘ Principe du réglage : on observe l'image de la fente à travers la lunette réglée. L'image de la fente est nette lorsque celle-ci envoie des rayons lumineux parallèles entre eux dans la lunette et donc lorsque la fente est placée dans le plan focal objet de la lentille du collimateur, comme représenté figure 8.

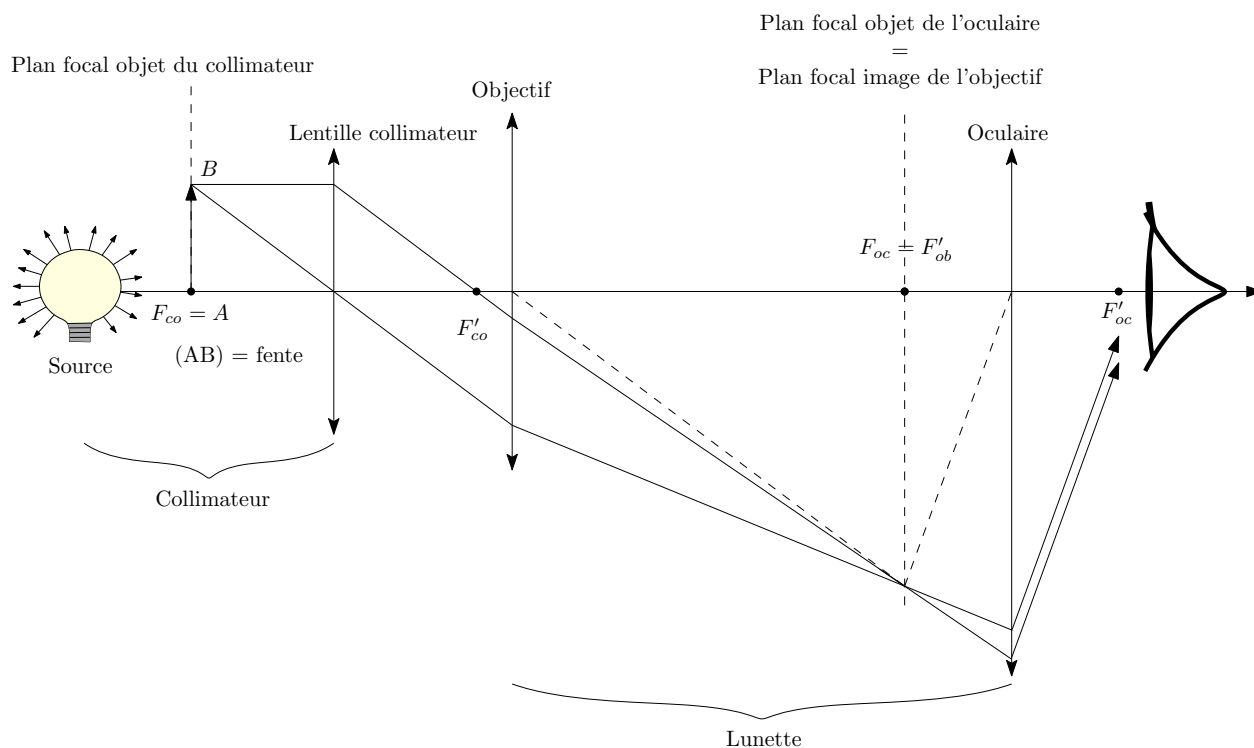


Figure 8 – Réglage du collimateur

2 Goniomètre à prisme, approche théorique

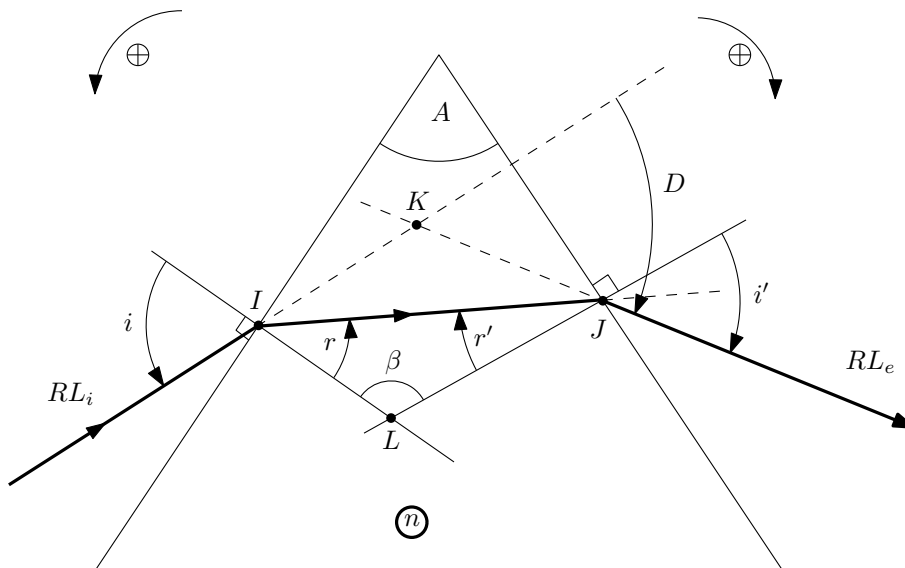


Figure 9 – Trajet d'un rayon lumineux dans le prisme

2.1 Les quatre relations du prisme

- ✘ On considère le prisme représenté figure 9.
- ✘ Les lois de Descartes en I et en J donnent :

$$\sin i = n \sin r$$

$$\sin i' = n \sin r'$$

- ✘ La somme des angles dans le quadrilatère $(AILJ)$ étant de 2π , on a :

$$A + \beta + \frac{\pi}{2} + \frac{\pi}{2} = 2\pi$$

D'où :

$$\beta = \pi - A$$

La somme des angles dans le triangle (IJL) étant de π , on a :

$$r + r' + \beta = \pi$$

D'où :

$$A = r + r'$$

- ✘ La somme des angles dans le triangle (IJK) étant égale à π , on a :

$$(\pi - D) + (i - r) + (i' - r') = \pi$$

D'où :

$$D = i + i' - r - r' = i + i' - A$$

Notons que pour les prismes à spectroscopie, l'angle au sommet A est standard et vaut :

$$A = 60^\circ$$

2.2 Réflexion totale

✘ Le phénomène de réflexion totale peut se poser au niveau du dioptre de sortie (passage d'un milieu plus réfringent à un milieu moins réfringent).

✘ L'angle limite de réflexion totale dans le cas de ce prisme vérifie :

$$\sin \Gamma = \frac{1}{n}$$

✘ Il y a réflexion totale en J , si :

$$r' > \Gamma$$

Soit :

$$r < A - \Gamma$$

Soit :

$$i < \arcsin(n(\sin(A - \Gamma)))$$

Pour $n = 1,5$ et $A = 60^\circ$, on trouve $i < i_0 = 28^\circ$.

Ainsi, pour éviter le phénomène de réflexion totale, on veillera à ne pas éclairer le prisme "normalement". A défaut de connaître précisément l'angle limite de réflexion totale, on préférera attaquer le prisme sous incidence rasante.

2.3 Déviation

2.3.1 Évolution de la déviation

✘ D'après les quatre relations du prisme, on établit facilement l'expression de la fonction $D(i)$:

$$D(i) = i + \arcsin(n \sin(A - \arcsin(\frac{1}{n} \sin(i)))) - A$$

✘ Cette fonction est définie pour i variant de i_0 à $\pi/2$ (d'après le principe de retour inverse de la lumière, i' prend alors les valeurs $\pi/2$ et i_0)

✘ Le tracé de D en fonction de i met en évidence un minimum de déviation :

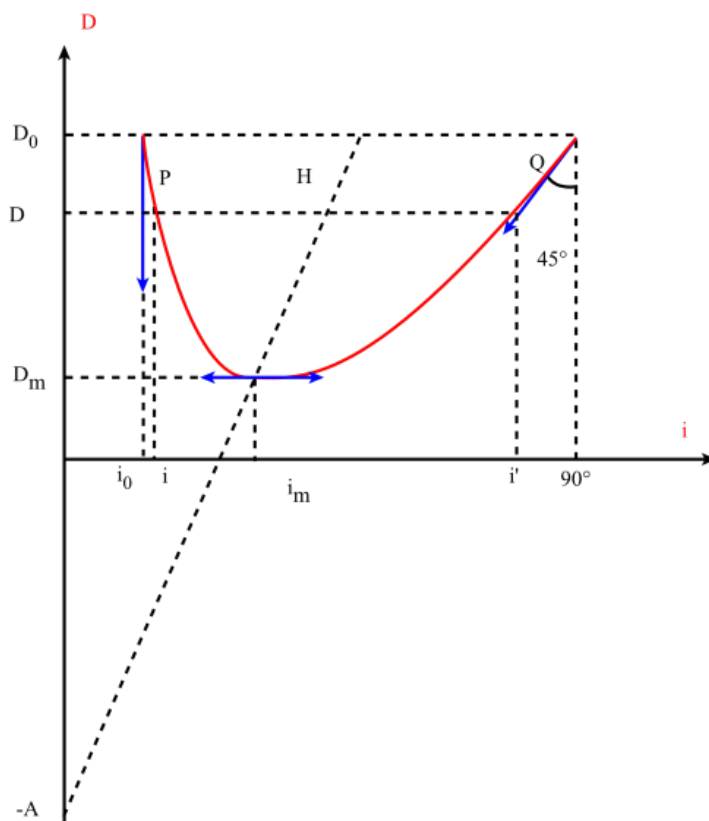


Figure 10 – Variation de la déviation en fonction de i

2.3.2 Minimum de déviation

✘ Le minimum de déviation correspond à :

$$dD = 0 = di + di'$$

$$\cos i di = \cos r dr$$

$$\cos i' di' = \cos r' dr'$$

$$0 = dr + dr'$$

✘ On obtient donc :

$$\frac{\cos i}{\cos i'} = \frac{\cos r}{\cos r'}$$

Soit, en passant au carré et en utilisant les lois de Descartes :

$$(1 - n^2(1 - \cos^2 r)) \cos^2 r' = (1 - n^2(1 - \cos^2 r')) \cos^2 r$$

Soit, après simplifications, sachant que r et r' sont compris entre 0 et $\pi/2$ et que n est différent de 1 :

$$r = r'$$

✘ Au minimum de déviation, on a donc une symétrie "entrée-sortie" :

$$i_m = i'_m$$

$$r_m = r'_m$$

$$D_m = 2i_m - A$$

D'où l'expression de l'indice n du prisme :

$$n = \frac{\sin\left(\frac{D_m + A}{2}\right)}{\sin\left(\frac{A}{2}\right)}$$

2.4 Loi de Cauchy

La loi de Cauchy s'énonce comme suit :

$$n(\lambda) = A + \frac{B}{\lambda^2}$$

Où :

$$A = 1 + \frac{\lambda_0^2}{2\lambda_p^2}$$

$$B = \frac{\lambda_0^4}{\lambda_p^2}$$

λ_0 et λ_p étant respectivement les longueurs d'onde propre et de plasma du milieu. Pour le verre étudié en TP (Flint Dense⁴), les ordres de grandeur sont les suivants :

$$A \approx 1.70$$

$$B \approx 13.2 \times 10^{-15} \text{ m}^2$$

Ce qui donne :

$$\lambda_0 = 138 \text{ nm}$$

$$\lambda_p = 116 \text{ nm}$$

Ces longueurs d'onde appartiennent bien aux UV.

3 Mesures de A , D_m et n

3.1 Mesure de A

3.1.1 Protocole

✘ On utilise la réflexion des rayons lumineux sur les deux faces du prisme :

4. SF : flint dense (Schwerflint), contenant 33% de SiO_2 ; 5% de K_2O et 62% de PbO

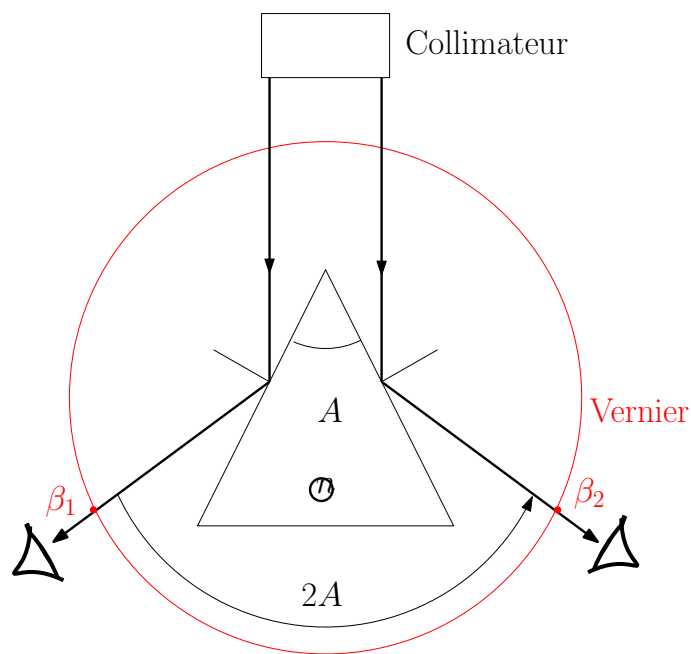


Figure 11 – Réflexion sur les deux faces du prisme

- ✘ On commence par repérer à l'œil nu les rayons réfléchis puis ensuite on utilise la lunette pour déterminer leur direction et mesurer β_1 et β_2 .
- ✘ On détermine β_1 et β_2 par encadrement :

$$\beta_1 = \frac{\beta_{1,max} + \beta_{1,min}}{2}$$

$$\beta_2 = \frac{\beta_{2,max} + \beta_{2,min}}{2}$$

- ✘ On en déduit les incertitudes sur β_1 et β_2 :

$$u_{\beta_1} = \frac{\beta_{1,max} - \beta_{1,min}}{2\sqrt{3}}$$

$$u_{\beta_2} = \frac{\beta_{2,max} - \beta_{2,min}}{2\sqrt{3}}$$

- ✘ Si ces incertitudes par encadrement sont modérées⁵, il faut rajouter les incertitudes de lecture sur le vernier :

$$u_{\beta_1} = \sqrt{\left(\frac{\beta_{1,max} - \beta_{1,min}}{2\sqrt{3}}\right)^2 + \left(\frac{q}{2\sqrt{3}}\right)^2}$$

$$u_{\beta_2} = \sqrt{\left(\frac{\beta_{2,max} - \beta_{2,min}}{2\sqrt{3}}\right)^2 + \left(\frac{q}{2\sqrt{3}}\right)^2}$$

- ✘ On en déduit A et u_A :

$$A = \frac{\beta_2 - \beta_1}{2}$$

$$u_A = \sqrt{\left(\frac{u_{\beta_1}}{2}\right)^2 + \left(\frac{u_{\beta_2}}{2}\right)^2}$$

5. du même ordre de grandeur que l'incertitude de lecture

3.1.2 Exemples de résultats - ODG de A

$$\times \beta_{1,min} = 109.6^\circ \text{ et } \beta_{1,max} = 110.2^\circ$$

$$\times \beta_{2,min} = 229.5^\circ \text{ et } \beta_{2,max} = 230.1^\circ$$

$$\times \beta_1 = 109.8^\circ \text{ et } u_{\beta_1} = \sqrt{\left(\frac{\beta_{1,max} - \beta_{1,min}}{2\sqrt{3}}\right)^2 + \left(\frac{q}{2\sqrt{3}}\right)^2} = 0.2^\circ$$

$$\times \beta_2 = 229.8^\circ \text{ et } u_{\beta_2} = \sqrt{\left(\frac{\beta_{2,max} - \beta_{2,min}}{2\sqrt{3}}\right)^2 + \left(\frac{q}{2\sqrt{3}}\right)^2} = 0.2^\circ$$

× D'où :

$$A = 59.95^\circ$$

$$u_A = 0.14^\circ$$

3.2 Mesure de D_m pour différentes longueurs d'ondes

3.2.1 Protocole

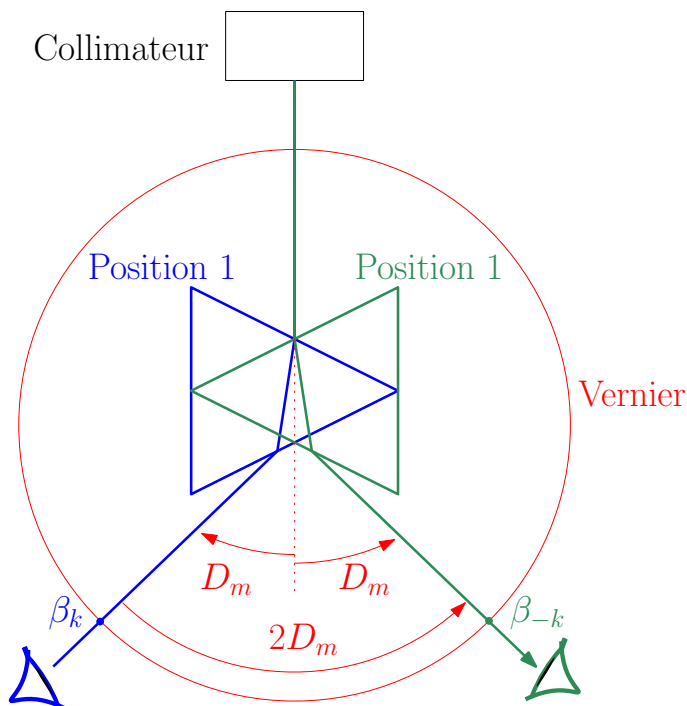


Figure 12 – Méthode du minimum de déviation

- × On éclaire une des faces du prisme sous incidence rasante et on repère, sur la face opposée, à l'œil, le spectre de la lumière utilisée.
- × On fait tourner le prisme avec le plateau en repérant, toujours à l'œil, le mouvement des raies. En tournant le plateau toujours dans le même sens on constate à un moment que les raies font demi-tour : on est alors au minimum de déviation. On utilise alors la lunette pour repérer précisément les positions des minima de déviation correspondant aux différentes longueurs d'onde λ_k . On note β_k l'angle lu sur le vernier.
- × On procède de la même manière en éclairant l'autre face du prisme et on note β_{-k} l'angle lu sur le vernier correspondant à la longueur d'onde λ_k .

3.2.2 Mesure de D_m

On en déduit, pour une longueur d'onde λ_k , la déviation minimale $D_{m,k}$:

$$D_{m,k} = \frac{\beta_k - \beta_{-k}}{2}$$

$$u_{D_m} = \sqrt{\left(\frac{u_{\beta_k}}{2}\right)^2 + \left(\frac{u_{\beta_{-k}}}{2}\right)^2}$$

3.2.3 Exemples de résultats - ODG de D_m

- ✘ On mesure β_1 et β_2 par encadrement : leurs incertitudes seront les mêmes que celles trouvées pour déterminer A
- ✘ On peut déterminer l'incertitude sur D_m en utilisant le calcul précédent. On peut aussi utiliser Régressi qui fait automatiquement le calcul de l'incertitude sur D_m .
- ✘ On obtient le tableau donné figure 15

3.3 Mesure de n pour différentes longueurs d'ondes

3.4 Approche théorique

- ✘ On déduit n de la mesure de D_m pour chaque raie grâce à la formule :

$$n = \frac{\sin\left(\frac{A + D_m}{2}\right)}{\sin\left(\frac{A}{2}\right)}$$

- ✘ Incertitude sur n

On détermine l'incertitude sur n de plusieurs manières :

- ☛ On calcule directement cette incertitude par la méthode des propagations :

$$n = \frac{\sin\left(\frac{A + D_m}{2}\right)}{\sin\left(\frac{A}{2}\right)}$$

$$\ln n = \ln\left(\sin\left(\frac{A + D_m}{2}\right)\right) - \ln\left(\sin\left(\frac{A}{2}\right)\right)$$

$$\frac{dn}{n} = \frac{\cos\left(\frac{A + D_m}{2}\right)}{\sin\left(\frac{A + D_m}{2}\right)} \left(\frac{dD_m}{2} + \frac{dA}{2}\right) - \frac{\cos\left(\frac{A}{2}\right)}{\sin\left(\frac{A}{2}\right)}$$

$$\frac{dn}{n} = \cot\left(\frac{A + D_m}{2}\right) \frac{dD_m}{2} + \left[\cot\left(\frac{A + D_m}{2}\right) - \cot\left(\frac{A}{2}\right)\right] \frac{dA}{2}$$

$$u_n = n \sqrt{\left[\cot\left(\frac{A + D_m}{2}\right)\right]^2 \left[\frac{dD_m}{2}\right]^2 + \left[\cot\left(\frac{A + D_m}{2}\right) - \cot\left(\frac{A}{2}\right)\right]^2 \left[\frac{dA}{2}\right]^2}$$

- ☛ On utilise Régressi pour calculer automatiquement cette incertitude.
- ☛ On utilise la méthode dite de Monte-Carlo pour faire ce calcul.

3.4.1 Exemple de résultats - ODG de n

Ci dessous un exemple de calcul de n et u_n pour $\lambda = 667.8 \text{ nm}$: les étendues ont été définies grâce aux incertitudes absolues à 95%.

```
#Méthode Monté-Carlo
import numpy as np
import matplotlib.pyplot as plt

#probabilités uniformes
N=10000
A = np.random.uniform(59.67 , 60.23 , size=N)
Dm=np.random.uniform(61.47,62.03,size=N)
n=(np.sin((A+Dm)*0.0174/2))/(np.sin(A*0.0174/2))
nmoy=np.average(n)
un=np.std(n,ddof=1)
print('valeur moyenne de n:',{nmoy})
print('ecart type : ', {un})

#histogramme des valeurs de n
plt.figure(1)
plt.hist(n, bins=50)
plt.title("histogramme indice")
plt.xlabel("indice")
plt.ylabel("Fréquence")
plt.show()
```

Figure 13 – Méthode Monté-Carlo : code

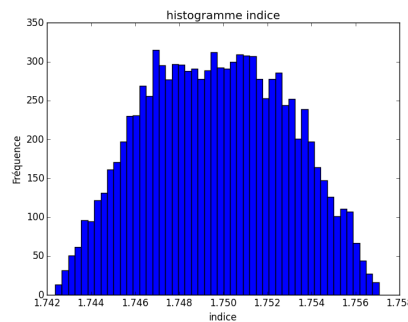


Figure 14 – Méthode Monté-Carlo : histogramme

La valeur donnée avec cette méthode est :

$$n = 1.750 \pm 0.003$$

3.5 Exemple de tableau récapitulatif - ODG

Les résultats sont récapitulés figure 15.

beta1	u(beta1)	beta2	u(beta2)	lambda	u(lambda)	A	u(A)	Dm	u(Dm)	X	$\frac{u(X)}{X}$	u(X)	n	u(n)
				m	m					m ⁻²		m ⁻²		
7,000	0,20	130,5	0,20	6,678·10 ⁻⁷	0,000000000010	59,95	0,14	61,75	0,14	2,242·10 ¹²		6,7·10 ⁷	1,748	0,0028
6,500	0,20	131,0	0,20	6,154·10 ⁻⁷	0,000000000010	59,95	0,14	62,25	0,14	2,640·10 ¹²		8,6·10 ⁷	1,752	0,0028
6,500	0,20	131,5	0,20	5,890·10 ⁻⁷	0,000000000010	59,95	0,14	62,50	0,14	2,883·10 ¹²		9,8·10 ⁷	1,754	0,0028
6,000	0,20	131,5	0,20	5,875·10 ⁻⁷	0,000000000010	59,95	0,14	62,75	0,14	2,897·10 ¹²		9,9·10 ⁷	1,756	0,0028
5,500	0,20	132,0	0,20	5,414·10 ⁻⁷	0,000000000010	59,95	0,14	63,25	0,14	3,412·10 ¹²		1,3·10 ⁸	1,761	0,0028
4,000	0,20	133,0	0,20	5,105·10 ⁻⁷	0,000000000010	59,95	0,14	64,50	0,14	3,837·10 ¹²		1,5·10 ⁸	1,771	0,0029
3,000	0,20	135,0	0,20	4,497·10 ⁻⁷	0,000000000010	59,95	0,14	66,00	0,14	4,945·10 ¹²		2,2·10 ⁸	1,783	0,0029
2,000	0,20	135,5	0,20	4,120·10 ⁻⁷	0,000000000010	59,95	0,14	66,75	0,14	5,891·10 ¹²		2,9·10 ⁸	1,789	0,0029

Figure 15 – Tableau récapitulatif des mesures de D_m et de n

Notons que pour l'incertitude sur λ , on a pris $u_\lambda \approx 0.01 \text{ nm}$ ce qui est l'ordre de grandeur classique des lampes spectrales basse pression.

4 Vérification de la loi de Cauchy

4.1 Avec le logiciel régressi sans incertitudes

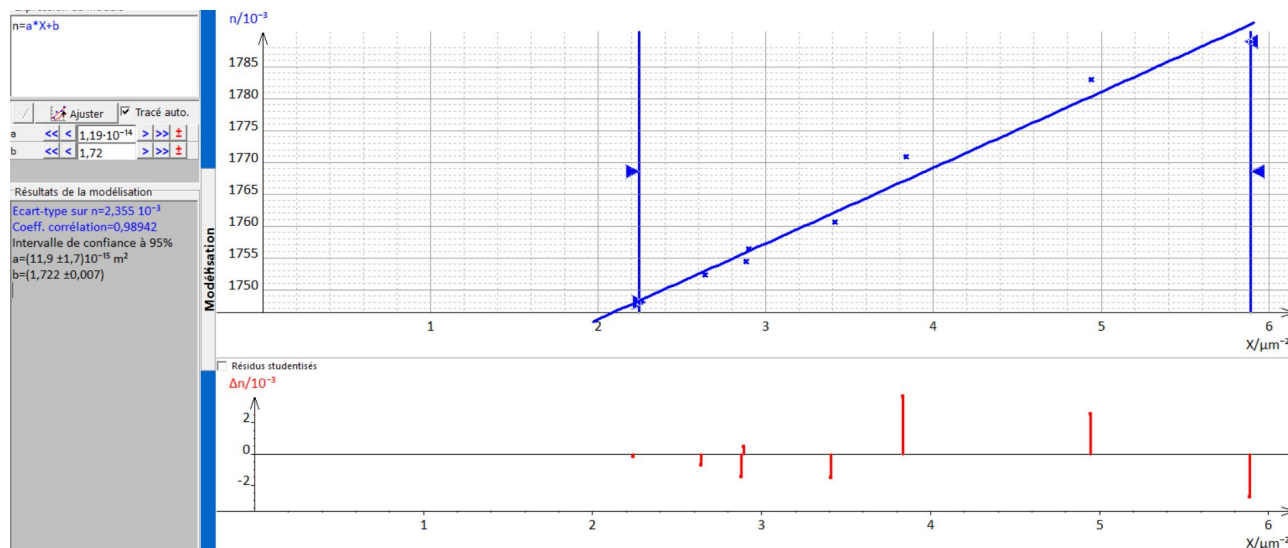


Figure 16 – Vérification de la loi de Cauchy : utilisation des résidus

Les résidus sont bien répartis de part et d'autre de la droite : la loi est vérifiée.

4.2 Avec le logiciel Régressi avec incertitudes

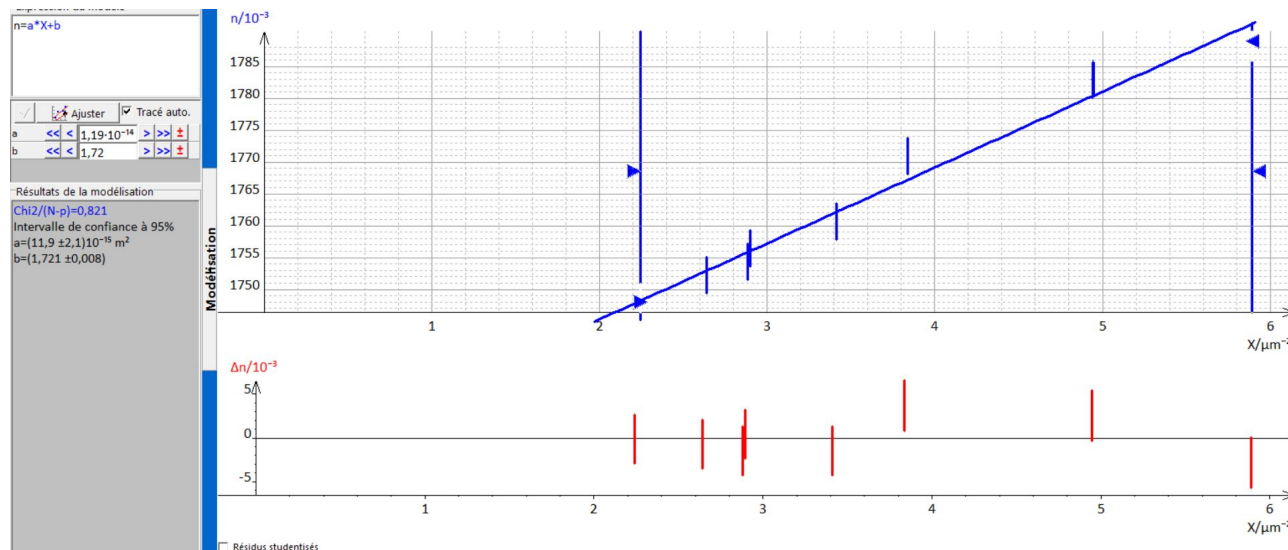


Figure 17 – Vérification de la loi de Cauchy : utilisation du χ^2

Le χ^2 est un peu faible, témoin du fait que les incertitudes sont légèrement sur-estimées. On peut néanmoins considérer que la loi est vérifiée.

4.3 Avec Python (Monté Carlo)

✘ Le code est donné figure18.

```

# -*- coding: utf-8 -*-
"""Régression linéaire."""

# Importation des bibliothèques
import numpy as np
import numpy.random as rd
import matplotlib.pyplot as plt

# Données
# Dans cet exemple, on connaît :
# * les incertitudes-types pour X ;
# * les incertitudes-types pour n.
X =
np.array([2.242*10**(12),2.640*10**(12),2.883*10**(12),2.897*10**(12),3.412*10**(12),
3.837*10**(12),4.945*10**(12),5.891*10**(12)])
uX = np.array([6.7*10**(7),8.6*10**(7), 9.8*10**(7), 9.9*10**(7), 1.3*10**(8),
1.5*10**(8), 2.2*10**(8), 2.9*10**(8)])
n = np.array([1.748,1.752,1.754,1.756,1.761,1.771,1.783,1.789])
un = np.array([0.0028,0.0028,0.0028,0.0028,0.0028,0.0029,0.0029,0.0029])

# Nombre de simulations
N = 100
# Détermination des pentes et ordonnées à l'origine
asim, bsim = [], []
for i in range(N):
    Xsim = X + uX*rd.uniform(-1, 1, len(X))
    nsim = n + un*rd.normal(-1, 1, len(n))
    p = np.polyfit(Xsim, nsim, 1)
    asim.append(p[0])
    bsim.append(p[1])
ma, ua = np.mean(asim), np.std(asim, ddof=1)
mb, ub = np.mean(bsim), np.std(bsim, ddof=1)
print("a =", ma)
print("u(a) =", ua)
print("b =", mb)
print("u(b) =", ub)

# Tracé des points et du modèle
nmod = ma*X + mb
plt.figure(figsize=(10, 5), dpi=200)
plt.subplot(1, 2, 1)
plt.errorbar(X, n, xerr=uX, yerr=un, fmt='bo', label="Points expérimentaux")
plt.plot(X, nmod, 'c--', label="Modèle affine")
plt.xlabel(r"$X$")
plt.ylabel(r"$n$")
plt.grid(), plt.legend(loc='best')

# Tracé des écarts normalisés algébriques
z = (n-nmod)/un
plt.subplot(1, 2, 2)
plt.plot(X, z, 'bo')
plt.fill_between([min(X)-1, max(X)+1], y1=-2, y2=2, color='cyan', alpha=0.1)
plt.xlabel(r"$X$"), plt.xlim(min(X)-1, max(X)+1)
plt.ylabel(r"$z$"), plt.ylim(-3, 3)
plt.grid()
plt.show()

```

Figure 18 – Code pour la régression linéaire par la méthode Monté-Carlo

✘ Les résultats donnés par la méthode Monté Carlo sont donnés difure 19

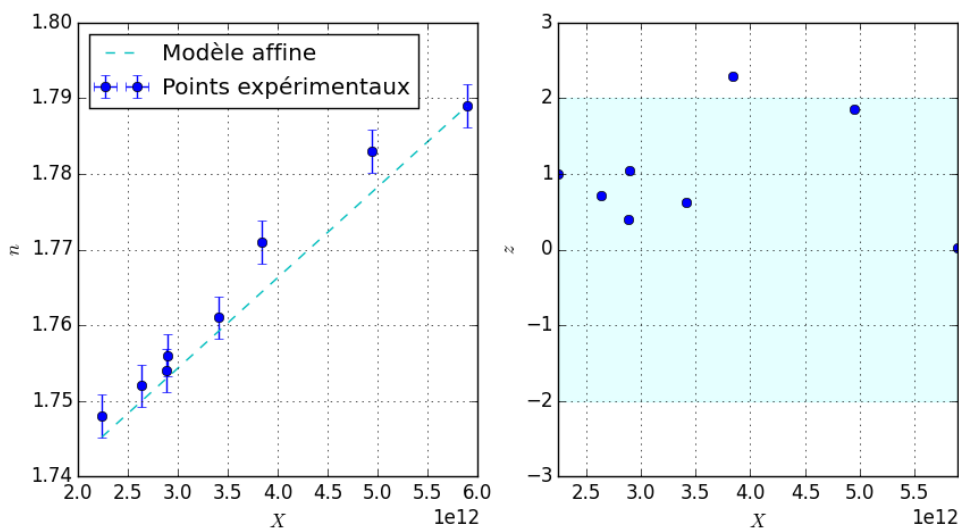


Figure 19 – Résultats de la méthode Monté Carlo

On peut considérer que les résidus sont corrects : la loi est vérifiée.