TP POLARISATION CORRIGE

1 Verification de la loi de Malus

Le but de cette partie est de vérifier la loi de Malus avec différentes méthodes et différents outils : Régressi, Python, papier millimétré.

1.1 Loi de Malus

On effectue le montage donné par la figure 1 :

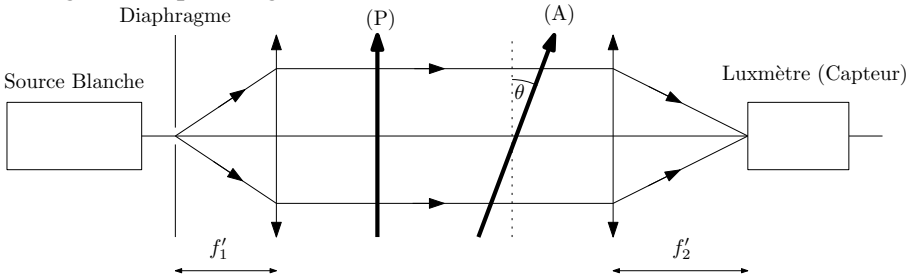


Figure 1

- ✘ La source utilisée est une lumière blanche, mais on peut utiliser une source monochromatique.
- ✘ L'objet, dont on forme l'image à l'entrée du capteur est un diaphragme.
- ✘ Sur le trajet de la lumière on place un polariseur (P) et un analyseur (A) faisant entre eux un angle θ.

Le champ électrique à la sortie du polariseur est de la forme :

$$\vec{E}_P(z,t) = E_0 \cos(\omega t - kz)\,\vec{u}_P$$

Où, $\vec{u}_P$ est le vecteur unitaire porté parla direction du polariseur.
Á la sortie de l'analyseur, le champ électrique est de la forme :

$$\vec{E}_A(z,t) = \vec{E}_P(z = z_A, t) \cdot \vec{u}_A = E_0 \cos(\omega t - kz_A) \times \cos\theta$$

Où, $\vec{u}_A$ est le vecteur unitaire porté parla direction de l'analyseur.
L'éclairement lumineux reçu par le capteur vaut donc :

$$I(\theta) = \frac{E_0^2}{2}\cos^2\theta = I_0 \cos^2\theta$$

⇒ C'est la loi de Malus.

1.2 Protocole expérimental

- ✘ On réalise le montage donné par la figure 1.
- ✘ On part de $\theta = 0\,°$ puis on augmente progressivement θ par pas de $10\,°$ par exemple.
- ✘ Pour chaque valeur de θ, on relève la valeur de l'éclairement I affiché par le luxmètre.
- ✘ On obtient une tableau à deux colonnes, l'une présentant les valeurs de θ et l'autre présentant les valeurs de I
- ✘ On retranche le flux minimal lorsque les deux polariseurs sont croisés. Cet éclirement résiduel est dû à l'environnement lumineux de la pièce et à la sensibilité spectrale du luxmètre qui déborde sur l'infrarouge. Si cet éclairement résiduel fluctue, on prend sa value moyenne et on adapte ensuite le nombre de chiffres significatifs dans les valeurs de I afin de ne pas avoir trop de chiffres incertains (maximum 2).

✘ Afin de vérifier la loi de Malus, on peut procéder de deux manières :

☛ On trace à l'aide d'un logiciel I en fonction de $\cos^2\theta$, sans entrer d'incertitudes expérimentales. Une régression linéaire nous permet d'obtenir l'équation d'une droite de la forme :

$$y = ax \text{ où } y = I; x = \cos^2\theta; a = I_0$$

Le logiciel affiche la valeur numérique de a avec son incertitude : cette incertitude est une incertitude d'origine statistique, elle est calculée à partir de la dispersion du nuage de points (extraite de la matrice, variance-covariance). Le seul critère pour vérifier la qualité de la regression linéaire est l'étude des résidus.

☛ On introduit les incertitudes expérimentales sur les mesures de I et θ :

— $u_I^2 = u_l^2 + u_c^2 + u_s^2$ avec $u_l = \dfrac{1digits}{2\sqrt{3}}$; u_c donnée dans la documentation constructeur et u_s est une incertitude type à rajouter éventuellement si les chiffres affichés sont instables (dans son cas là, on estime les valeurs maximales et minimales de I et on en déduit $u_s = \dfrac{1}{\sqrt{3}}\dfrac{I_{max} - I_{min}}{2}$. Notons que dans ce cas là, la valeur mesurée est $I = \dfrac{I_{max} + I_{min}}{2}$).

— Sachant que θ correspond à la différence des positions du polariseur et de l'analyseur, on a :

$$\theta = \theta_A - \theta_P$$

Et donc :

$$u_\theta^2 = u_{\theta_A}^2 + u_{\theta_P}^2$$

Avec :

$$u_{\theta_A}^2 = u_{\theta_P}^2 = \left(\frac{q}{2\sqrt{3}}\right)^2$$

On trace à l'aide d'un logiciel I en fonction de $\cos^2\theta$. Les incertitudes expérimentales peuvent être représentées sous forme de rectangles ou d'ellipses. Une régression linéaire nous permet d'obtenir l'équation d'une droite de la forme :

$$y = ax \text{ où } y = I; x = \cos^2\theta; a = I_0$$

Le logiciel affiche les valeurs numériques de a et de son incertitude type : cette incertitude est calculée à partir des incertitudes expérimentales sur I et θ. L'intérêt de cette méthode est que le logiciel affiche alors la valeur du χ^2 : Ce coefficient est le seul critère rigoureux permettant de valider la régression linéaire. Si χ^2 est très inférieur à 1, les incertitudes expérimentales sont sous estimées, si le χ^2 est très supérieur à 1, les incertitudes sont surestimées. De fait la régression linéaire peut être validée si le χ^2 est de l'ordre de 1. Notons que les résidus peuvent être affichés également et studentisés (ou normalisés).

1.3 Méthode 1 : sans incertitudes expérimentales

i	θ	I	$x = \cos^2\theta$
	°	Lux	
0	0.000	0.4000	1.000
1	10.00	0.3700	0.9698
2	2.00	0.3200	0.8830
3	30.00	0.2600	0.7500
4	40,00	0,1900	0,5868
5	50,00	0,1200	0,4132
6	60,00	0,0900	0,2500
7	70,00	0,0050	0,1170
8	80,00	0,0010	0,03015
9	90,00	0,000	$3{,}749\cdot10^{-}33$

1.3.1 Méthode 1, avec Régressi

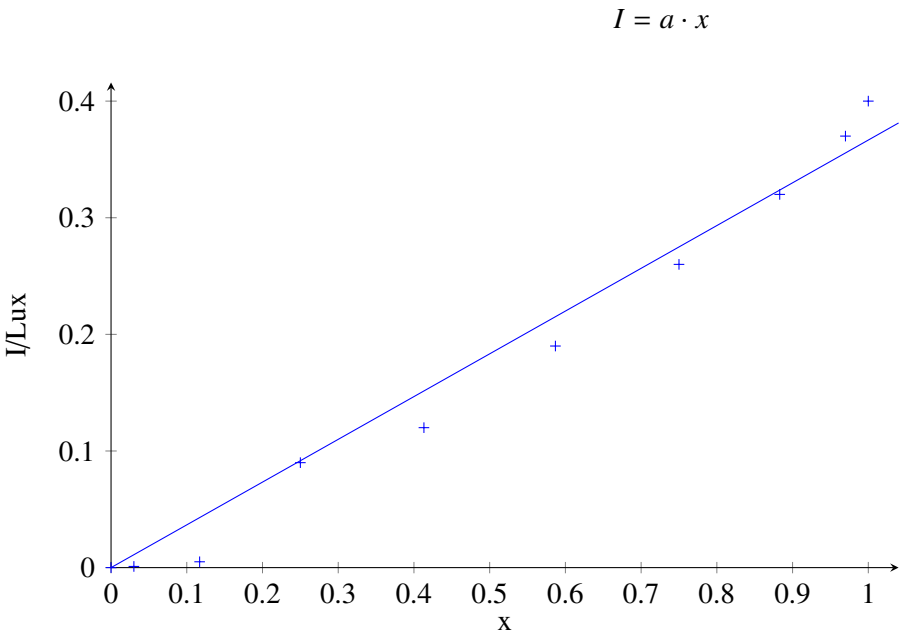


Figure 2 – Résultats méthode 1 avec Régressi

Résultats méthode 1 avec Régressi

Ecart-type sur I=22,91 $10^{-}3$ Lux
Coeff. corrélation=0,99295
Intervalle de confiance à 95%
a=(0.367 ±0.026) S.I.
Résidus paraboliques

1.3.2 Méthode 1, avec Python

Le code est fourni en annexe. Noter que l'on peut obtenir la variance sur la pente en utilisant curvefit. les résultats sont les suivants :

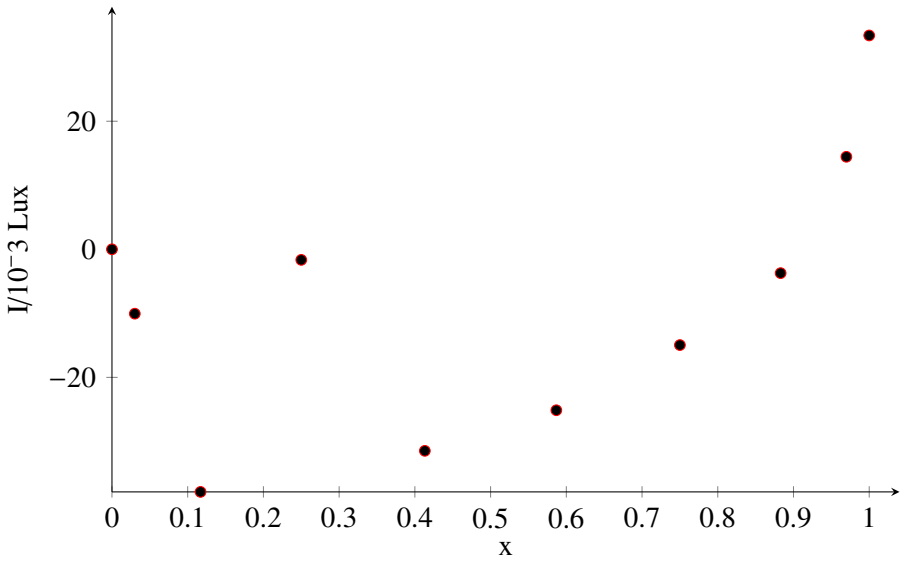


FIGURE 3 – Résidus méthode 1 avec Régressi

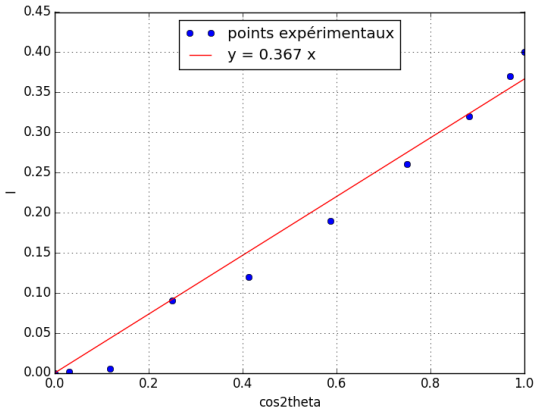


FIGURE 4 – $a = 0.37$ et $u_a = 0.01$

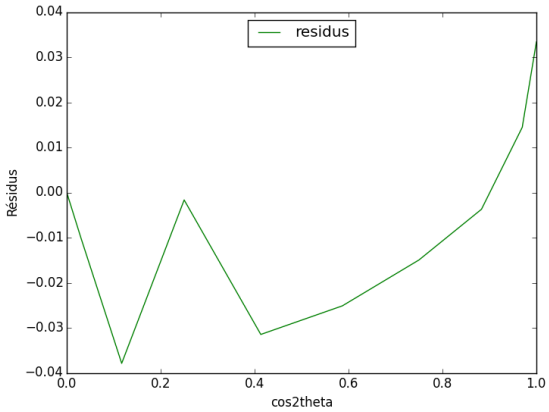


FIGURE 5 – Résidus sous python

Les résidus ont ainsi une tendance parabolique, la loi n'est pas trés bien vérifiée.

Résultats méthode 1 avec Python

Ecart-type sur I=19,53 $10^{-}3$ Lux
Coeff. corrélation=0,99295
Intervalle de confiance à 95%
a=(0.395 ±0.038) S.I.

1.4 Méthode 2 : avec incertitudes expérimentales

i	θ	u_θ	I	u_I	x	u_x
	°	°	Lux	Lux		
0	0,000	0,71	0,4000	0,013	1,000	0,00
1	10,00	0,71	0,3700	0,012	0,9698	0,0042
2	20,00	0,71	0,3200	0,011	0,8830	0,0080
3	30,00	0,71	0,2600	0,0090	0,7500	0,011
4	40,00	0,71	0,1900	0,0070	0,5868	0,012
5	50,00	0,71	0,1200	0,0050	0,4132	0,012
6	60,00	0,71	0,0900	0,0040	0,2500	0,011
7	70,00	0,71	0,0050	0,0020	0,1170	0,0080
8	80,00	0,71	0,0010	0,0010	0,03015	0,0042
9	90,00	0,71	0,000	0,0010	$3{,}749\cdot10^{-}33$	0,001

1.4.1 Méthode 2, avec Régressi

$$I = a \cdot x$$

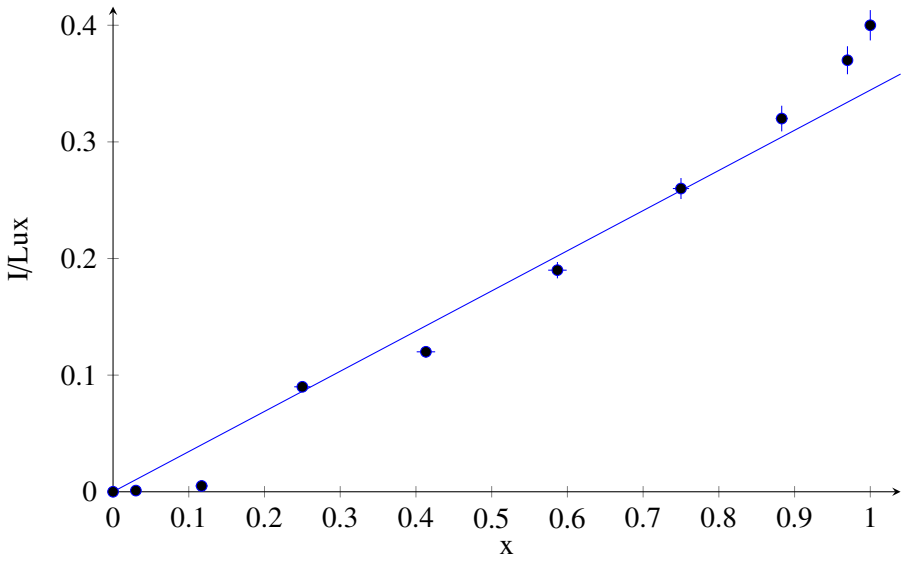


Figure 6 – Résultat méthode 2 avec Régressi

Résultats méthode 2 avec Régressi

Chi2/(N-p)=4,47
Intervalle de confiance à 95%
a=(0.344 ±0.012) S.I.
Résidus paraboliques

1.4.2 Méthode 2, avec python et calcul de χ^2

Le code est fourni en annexe, les résultats sont les suivants :

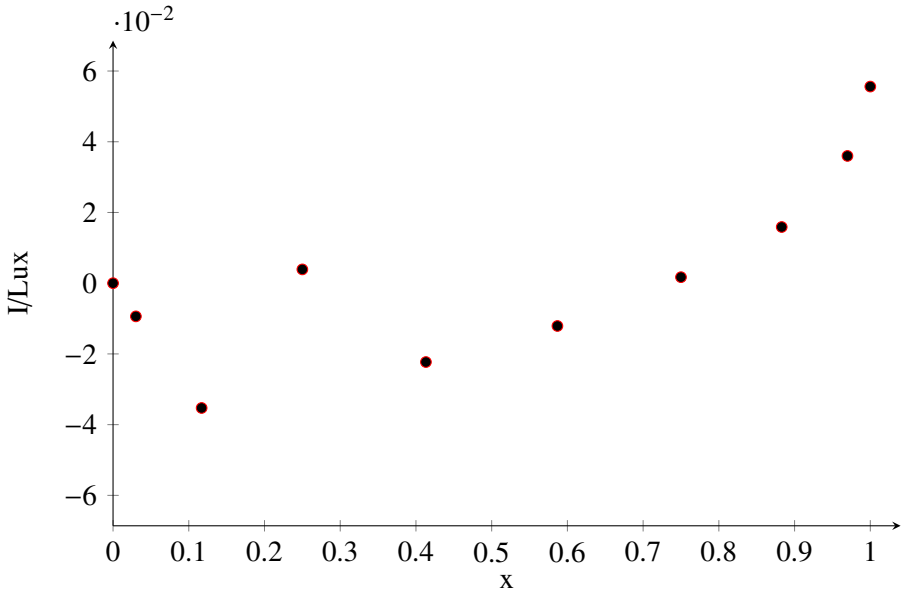


Figure 7 – Résidus non normalisés, méthode 2, sous Régressi

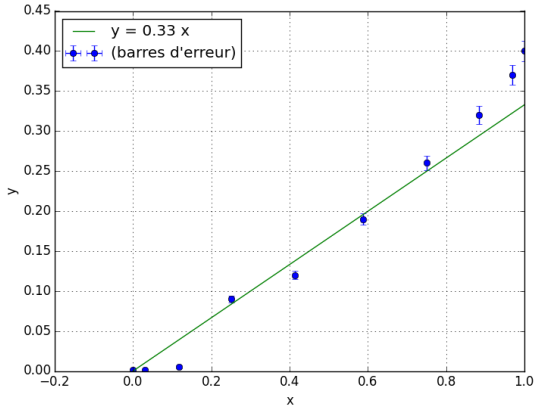


Figure 8 – Nuage de points et courbe de tendance sous python
(a=0.366 ± 0.003 SI)

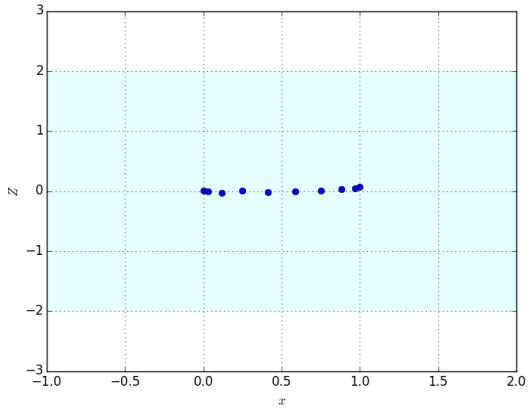


Figure 9 – Résidus sous python

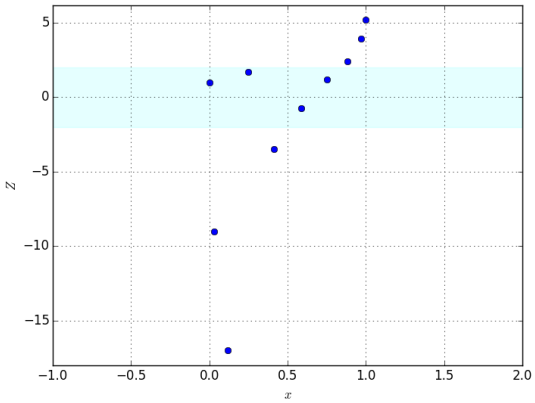


Figure 10 – Résidus normalisés sous python

Les résidus normalisés ne sont pas corrects : les incertitudes sont sous-évaluées. La valeur élevée du χ^2 le confirme.

Résultats méthode 2, avec Python et calcul du χ^2

Chi2/(N-p)=38.74
Intervalle de confiance à 95%
a=(0.333 ±0.005) S.I.
Résidus paraboliques

1.4.3 Méthode 2, avec python et Monté-Carlo

Le code est fourni en annexe, les résultats sont les suivants :

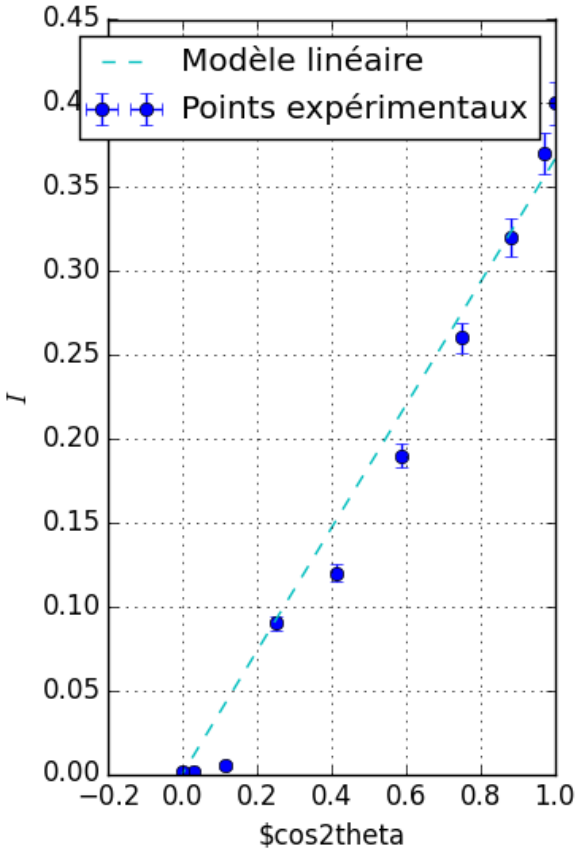


Figure 11 – Nuage de points et courbe de tendance sous python
(a=0.366 ± 0.003 SI)

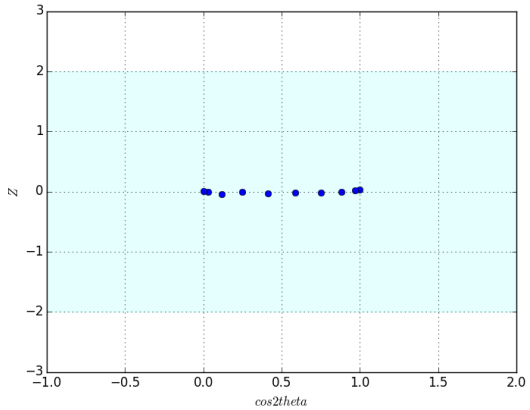


Figure 12 – Résidus sous python

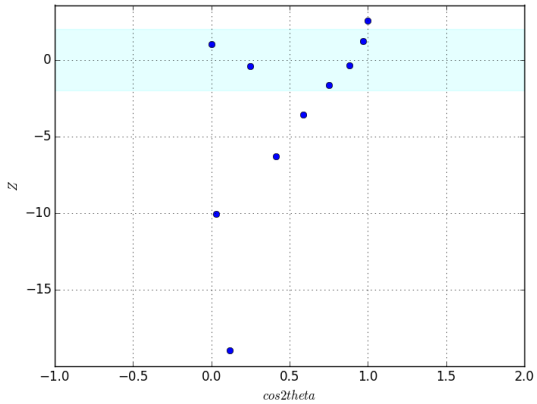


Figure 13 – Résidus normalisés sous python

Résultats méthode 2, avec Python et méthode Monté Carlo

Intervalle de confiance à 95%
a=(0.344 ±0.012)S.I.
Résidus paraboliques

1.5 Conclusion

Les deux méthodes donnent des résultats proches.

Les résidus sont très médiocres : ils ont une tendance parabolique.

Le χ^2 avec la deuxième méthode est médiocre : ceci vient de la sous-estimation des incertitudes (les ellipses ne coupent pas la droite de régression).

2 Détermination de la biréfringence d'une lame - application au scotch

2.1 Approche théorique

2.1.1 Expression générale de l'éclairement

On étudie le montage représenté figure 14.

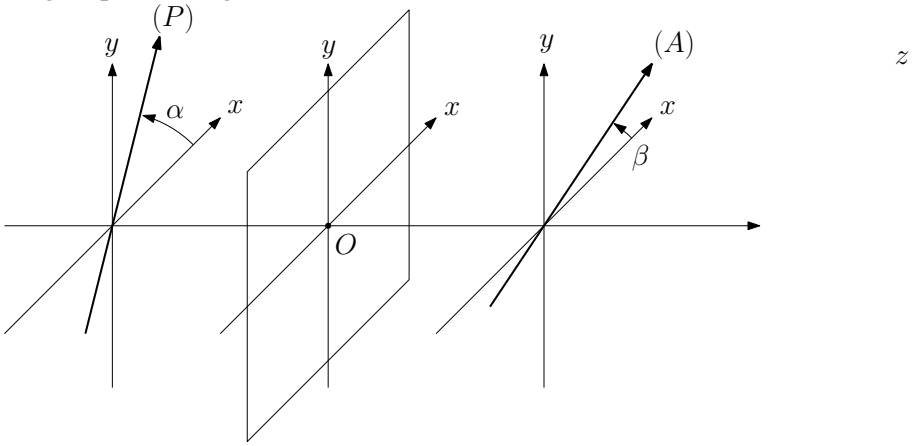


Figure 14

On travaille dans le repère de la lame : (Ox, Oy) sont les directions des lignes neutres.

Le retard de phase introduit par la lame entre les projections du champ suivant (Oy) et (Ox) est $\Delta\theta = k\Delta n = \dfrac{2\pi}{\lambda}\Delta n$.

Le champ issu du polariseur et arrivant au niveau de la lame est polarisé rectilignement :

$$\vec{E}(z=0,t) = E_0 \exp(j\omega t)\,\vec{u}_P$$

La projection de ce champ sur repère de la lame donne :

$$\vec{E}(z=0,t) = E_0\left(\cos\alpha\,\vec{u}_x + \sin\alpha\,\vec{u}_y\right)\exp(j\omega t)$$

Á la sortie de la lame, le champ est de la forme :

$$\vec{E}(e,t) = E_0\left(\cos\alpha\,\vec{u}_x + \sin\alpha\exp(-j\Delta\theta)\,\vec{u}_y\right)\exp\left(j\omega t - \frac{2\pi}{\lambda}n_x - kz\right)$$

En effectuant un changement dans la base de temps, on peu écrire le champ sous la forme :

$$\vec{E}(e,t) = E_0\left(\cos\alpha\,\vec{u}_x + \sin\alpha\exp(-j\Delta\theta)\,\vec{u}_y\right)\exp(j\omega t - kz)$$

Le champ à la sortie de l'analyseur est polarisé rectilignement : on trouve sa valeur en projetant le champ issu de la lame sur l'axe de l'analyseur :

$$E_A = \vec{E}(z,t)\cdot\vec{u}_A = \vec{E}(z,t)\cdot\left(\cos\beta\,\vec{u}_x + \sin\beta\,\vec{u}_y\right)$$

$$E_A = E_0\left(\cos\alpha\,\vec{u}_x + \sin\alpha\exp(-j\Delta\theta)\,\vec{u}_y\right)\cdot\left(\cos\beta\,\vec{u}_x + \sin\beta\,\vec{u}_y\right)\exp(j\omega t - kz)$$

$$E_A = E_0\left(\cos\alpha\cos\beta + \sin\alpha\sin\beta\exp(-j\Delta\theta)\right)\exp(j\omega t - kz)$$

D'où, l'expression de l'éclairement à la sortie de l'analyseur :

$$I_A = kE_0^2\left(\cos^2\alpha\cdot\cos^2\beta + \sin^2\alpha\cdot\sin^2\beta + 2\cos\alpha\cos\beta\sin\alpha\sin\beta\cos\Delta\theta\right)$$

$$I_A = kE_0^2\left(\cos^2\alpha\cdot\cos^2\beta + \sin^2\alpha\cdot\sin^2\beta + \frac{1}{2}\sin 2\alpha\sin 2\beta\cos\Delta\theta\right)$$

$$I_A = I_0\left(\cos^2(\alpha-\beta) - \sin 2\alpha\sin 2\beta\sin^2\left(\frac{\Delta\theta}{2}\right)\right)$$

Ainsi, l'intensité résultante n'est pas la somme des intensités des deux vibrations transmises par la lame : on peut dire que les deux vibrations déphasées de $\Delta\theta$ interfèrent grâce à l'analyseur qui rend leur directions parallèles.

2.1.2 Cas où le polariseur et l'analyseur sont croisés et font un angle de $45\,^\circ$ avec les axes de la lame

On se place dans le cas où (P) et (A) sont croisés et font un angle de $45\,^\circ$ avec les axes de la lame. On peut par exemple prendre :

$$\alpha = 45\,^\circ \text{ et } \beta = -45\,^\circ$$

On obtient alors :

$$I = I_0\sin^2\left(\frac{\Delta\theta}{2}\right)$$

2.1.3 Cas où le polariseur et l'analyseur sont parallèles et font un angle de $45\,^\circ$ avec les axes de la lame

On se place dans le cas où (P) et (A) sont parallèles et font un angle de $45\,^\circ$ avec les axes de la lame. On peut par exemple prendre :

$$\alpha = 45\,^\circ \text{ et } \beta = 45\,^\circ$$

On obtient alors :

$$I = I_0\cos^2\left(\frac{\Delta\theta}{2}\right)$$

2.1.4 Interprétation

Les interférences sont constructives ($I = I_0$) ou destructives ($I = 0$) suivant la valeur de $\Delta\theta$. Comme $\Delta\theta$ dépend de λ, pour une lame donnée, certaines longueurs d'onde donnent des interférences constructives et d'autres destructives. Si la lumière incidente est blanche et la lame est suffisamment fine, la lumière transmise sera colorée, d'une teinte caractéristique de l'épaisseur et de la biréfringence de la lame (échelle des teintes de Newton).

Pour la suite, on se placera dans le cas où le polariseur et l'analyseur font un angle de $45\,^\circ$ avec les axes de la lame. Pour cela, on déterminera les axes neutres de la lame et croisant le polariseur et l'analyseur et en tournant l'ensemble jusqu'à ce que l'éclairement sur l'écran soit nul.

2.2 Mesure avec un faible épaisseur de scotch : échelle de teintes

On utilise des petites épaisseurs de scotch de façon à pouvoir observer clairement des teintes. Connaissant l'épaisseur du scotch, on peut en déduit la biréfringence.

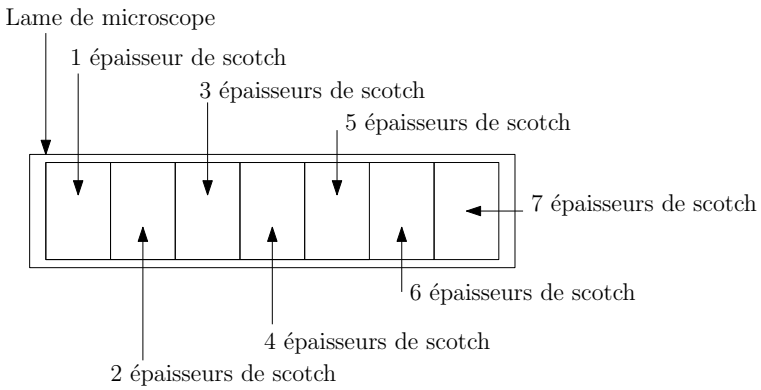


Figure 15 – Disposition des différentes épaisseurs de scotch sur la lame de microscope

2.2.1 Montage expérimental

✘ On réalise le montage suivant :

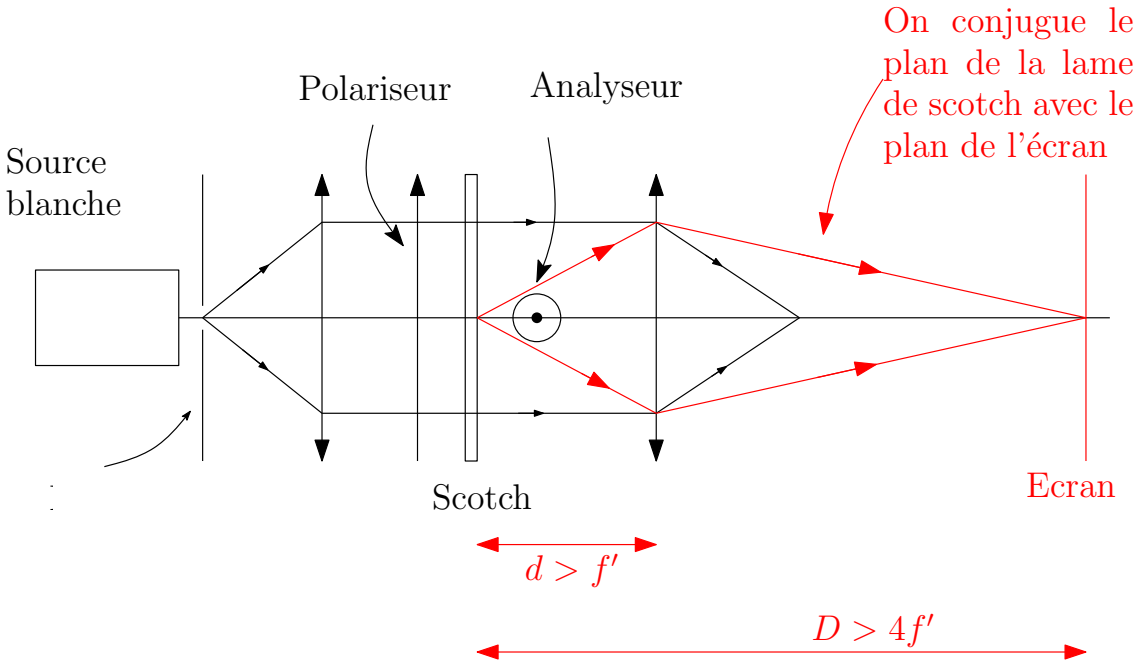


Figure 16 – Montage pour observer les teintes de Newton

✘ On obtient des teintes en fonction du nombre d'épaisseurs de scotch concernées :

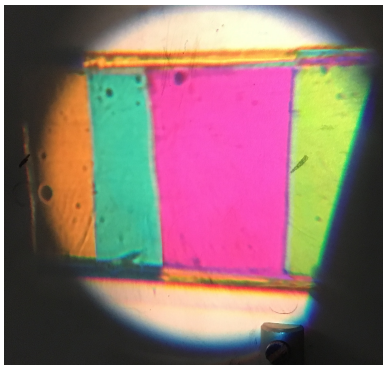


Figure 17 – Obtention des teintes de Newton

2.2.2 Résultats

✘ On estime un encadrement pour les teintes observées. Cet encadrement permet de déterminer deux différences de marche δ_{max} et δ_{min} et donc de déterminer une différence de marche mesurée : $\delta = \dfrac{\delta_{max} + \delta_{min}}{2}$ ainsi que son incertitude type : $u_\delta = \dfrac{\delta_{max} - \delta_{min}}{2\sqrt{3}}$.

✘ On en déduit, connaissant l'épaisseur de la lame de scotch ($e = 39\,\mu$m; $u_e = 1\,\mu$m), la biréfringence du scotch Δn et son incertitude type $u_{\Delta n}$ grâce aux formules :

$$\Delta n = \frac{\delta}{Ne}; \frac{u_{\Delta n}}{\Delta n} = \sqrt{\left(\frac{u_\delta}{\delta}\right)^2 + \left(\frac{u_e}{e}\right)^2}$$

nombre d'épaisseur (N)	teinte ($A \perp P$)	teinte ($A \| P$)	$\delta = Ne\Delta n$ (nm)	u_δ	Δn	$u_{\Delta n}$
1	jaune vif - jaune brun	bleu - bleu gris	381	28	0.0098	0.0008
2	vert - vert clair	rouge carmin - pourpre	787	23	0.010	0.0004
3	indigo - bleu verdâtre	jaune sale - chair	1204	31	0.010	0.0004

Proposition d'incertitude liée aux différentes mesures :
On fait la moyenne des valeurs mesurées :

$$\Delta n_m = 0.010$$

Incertitudes : On utilise les valeurs maximale et minimale de δn :

$$u_{\delta n} = \frac{(0.010 + 0.0004) - (0.0098 - 0.0008)}{2} = 0.0007$$

D'où :

$$\Delta n = 0.010 \pm 0.001$$

(à 95%)

2.3 Mesure avec une « grande épaisseur » de scotch : étude du spectre cannelé

On se place dans le cas où (P) et (A) sont croisés et font un angle de 45 ° avec les axes de la lame.

2.3.1 Principe

✘ Rappel : on se place dans le cas où (P) et (A) sont croisés et à 45 ° des lignes neutres de la lame. Une longueur d'onde est éteinte si :

$$\sin^2\left(\frac{\pi\delta}{\lambda}\right)^2 = 0 \Rightarrow \delta = p\lambda$$

✘ On dispose un grand nombre d'épaisseur de lame de scotch ($N \Rightarrow \delta = Ne\Delta n$) et on observe la lumière à la sortie de l'analyseur ((P) et (A) étant croisés par exemple à 45 ° des lignes neutres du scotch).

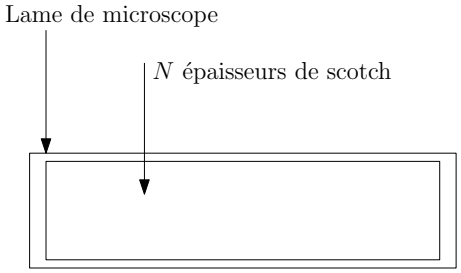


Figure 18 – Disposition des N épaisseurs de scotch sur la lame de microscope

✘ En pratique, on peut exploiter le spectre cannelé de différentes manières :

- ☛ On connait N et on cherche Δn,
- ☛ On connait δn et on déterminer N
- ☛ ...

✘ Estimation de Ne :

✘ A l'aide d'un système dispersif, on observe le spectre de la lumière issue de l'analyseur (A) : on doit obtenir un spectre cannelé. On peut alors soit compter le nombre de cannelures pour avoir une valeur approchée de δ, soit mesurer les longueurs d'ondes éteintes et effectuer une régression linéaire pour une mesure plus précise de δ.

2.3.2 Protocole 1 : observation directe du spectre cannelé

On mesure N à l'aide d'un Balmer et on en déduit δn

✘ Estimation de Ne :

- ☛ à l'aide d'une Palmer [1], on détermine l'épaisseur d'une lame de microscope :

$$e_{lame} = 1.09 \times 10^{-3}\ \text{m}$$

$$u_{e_{lame}} = \sqrt{2\left(\frac{q}{2\sqrt{2}}\right)^2} = 0.03 \times 10^{-3}\ \text{m}$$

- ☛ Toujours à l'aide du Palmer, on détermine l'épaisseur de la lame de microscope et des N couches de scotch :

$$e_{lame+Ne} = 2.65 \times 10^{-3}\ \text{m}$$

$$u_{e_{lame+Ne}} = \sqrt{2\left(\frac{q}{2\sqrt{2}}\right)^2} = 0.03 \times 10^{-3}\ \text{m}$$

- ☛ On en déduit la valeur de e_{Ne} :

$$e_{Ne} = e_{lame+Ne} - e_{lame} = 1.56 \times 10^{-3}\ \text{m}$$

$$u_{e_{Ne}} = \sqrt{(u_{e_{lame+Ne}})^2 + (u_{e_{lame+Ne}})^2} = 0.04 \times 10^{-3}\ \text{m}$$

- ☛ On en déduit la valeur de N (non obligatoire) :

$$N = \frac{e_{Ne}}{e} = 40$$

$$u_N = N\sqrt{\left(\frac{u_{e_{Ne}}}{e_{Ne}}\right)^2 + \left(\frac{u_e}{e}\right)^2} = 1$$

✘ On effectue le montage de la figure 19.

1. précision :$q = 0.01 \times 10^{-3}$ m

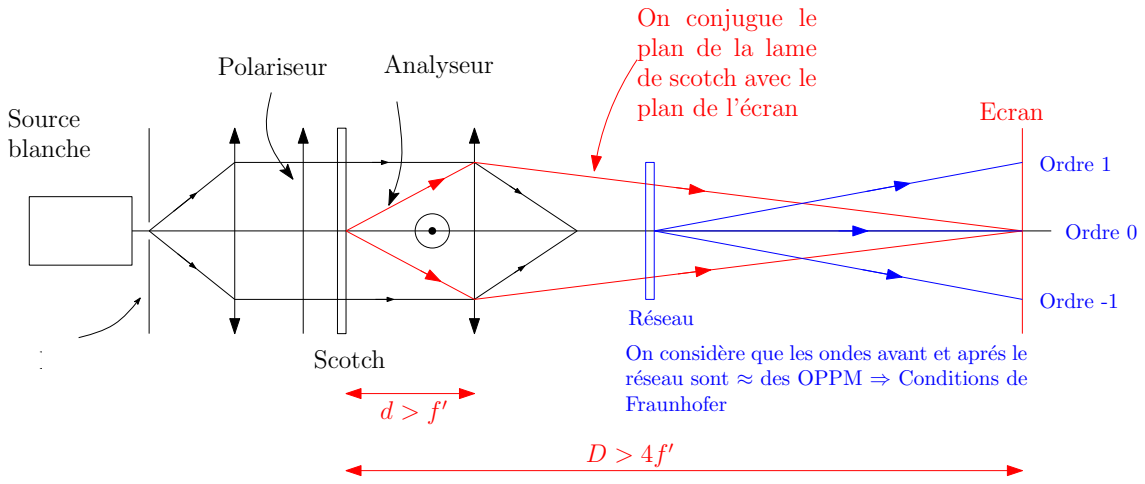


Figure 19 – Observation directe du spectre cannelé à l'aide d'un réseau

✘ Observation :

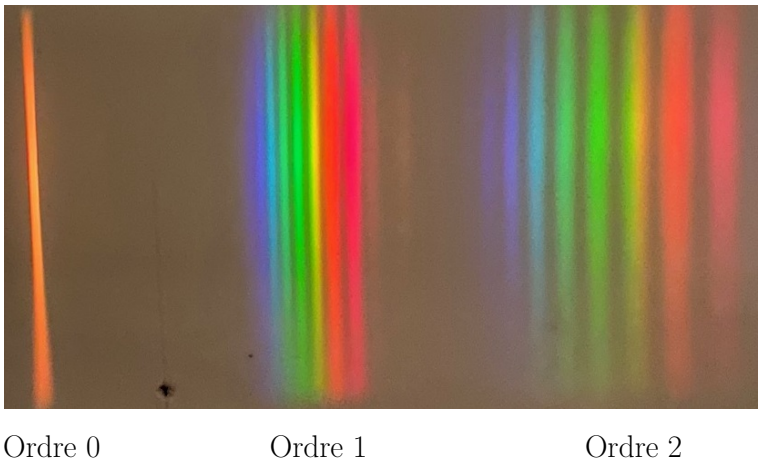


Figure 20 – Observation du spectre cannelé sur l'écran

✘ On compte le nombre de cannelures N' et on estime l'incertitude type sur N' : $u_{N'} = \dfrac{N'_{max} - N'_{min}}{2\sqrt{3}}$.

✘ Ce nombre de cannelures est liées à la taille du spectre de la source. Pour une lumière blanche, on a :

$$\lambda_V < \lambda < \lambda_R \Rightarrow \frac{1}{\lambda_R} < \frac{1}{\lambda} < \frac{1}{\lambda_V} \Rightarrow \frac{\delta}{\lambda_R} < \underbrace{\frac{\delta}{\lambda}}_{p} < \frac{\delta}{\lambda_V} \Rightarrow \underbrace{\frac{Ne\Delta n}{\lambda_R}}_{p_{min}} < p < \underbrace{\frac{Ne\Delta n}{\lambda_V}}_{p_{max}}$$

D'où, l'expression du nombre de cannelures :

$$N' = \left\lfloor \frac{Ne\Delta n}{\lambda_V} - \frac{Ne\Delta n}{\lambda_R} \right\rfloor = \left\lfloor Ne\Delta n \left(\frac{1}{\lambda_V} - \frac{1}{\lambda_R} \right) \right\rfloor$$

En assimilant la partie entière du second membre à sa valeur on obtient :

$$\Delta n = \frac{N'}{Ne\left(\dfrac{1}{\lambda_V} - \dfrac{1}{\lambda_R} \right)}$$

✘ Pour calculer l'incertitude liée à cette mesure, on peut utiliser la propagation des incertitudes ou utiliser la méthode Mont-Carlo.

☛ Mesure de δn avec propagation des incertitudes

— Connaissant les incertitudes sur N', N, e et l'intervalle de longueur d'onde, on en déduit la mesure de Δn :

$$\Delta n = \frac{N'}{Ne\left(\frac{1}{\lambda_V} - \frac{1}{\lambda_R}\right)}$$

$$\Delta n = \frac{N'\lambda_R}{Ne} \text{ et } \frac{u_{\Delta n}}{\Delta n} = \sqrt{\left(\frac{u_N}{N}\right)^2 + \left(\frac{u_e}{e}\right)^2 + \left(\frac{u_{N'}}{N'}\right)^2 + \left(\frac{\frac{u_{\lambda_V}}{\lambda_V^2}}{\frac{1}{\lambda_V} - \frac{1}{\lambda_R}}\right)^2 + \left(\frac{\frac{u_{\lambda_R}}{\lambda_R^2}}{\frac{1}{\lambda_V} - \frac{1}{\lambda_R}}\right)^2}$$

.

— Résultats :

— Le nombre de couche N a une épaisseur : $e_{Ne} = 1.56 \times 10^{-3}$ m avec $u_{e_{Ne}} = 0.04 \times 10^{-3}$ m

— On compte $N' = 11$ cannelures avec $u_{N'} = 1$ (partie entière)

— $\lambda_V = 400\,\text{nm}$, $\lambda_R = 700\,\text{nm}$ avec $u_{\lambda_R} = u_{\lambda_v} = 50\,\text{nm}$

— En utilisant :

$$\Delta n = \frac{N'}{Ne\left(\frac{1}{\lambda_V} - \frac{1}{\lambda_R}\right)}$$

$$\Delta n = \frac{N'\lambda_R}{Ne} \text{ et } \frac{u_{\Delta n}}{\Delta n} = \sqrt{\left(\frac{u_N}{N}\right)^2 + \left(\frac{u_e}{e}\right)^2 + \left(\frac{u_{N'}}{N'}\right)^2 + \left(\frac{\frac{u_{\lambda_V}}{\lambda_V^2}}{\frac{1}{\lambda_V} - \frac{1}{\lambda_R}}\right)^2 + \left(\frac{\frac{u_{\lambda_R}}{\lambda_R^2}}{\frac{1}{\lambda_V} - \frac{1}{\lambda_R}}\right)^2}$$

. On en déduit :

$$\Delta n = 0.0092$$

$$u_{\Delta n} = 0.0009$$

☛ Mesure de δn avec Monté-Carlo

— Le code utilisé est donné ci-dessous :

```
# -*- coding: utf-8 -*-

# Importation des bibliothèques


import numpy as np
from scipy import *
import matplotlib.pyplot as plt

#probabilités uniformes
n=10000
Np = np.random.randint(10,12)
N = np.random.randint(39,41)
e=np.random.uniform(38*10**(-6),40*10**(-6),size=n)
lambdar=np.random.uniform(700*10**(-9),750*10**(-9),size=n)
lambdav=np.random.uniform(400*10**(-9),450*10**(-9),size=n)
Dn=Np/(e*N*(1/lambdav-1/lambdar))
Dnmoy=np.average(Dn)
uDn=np.std(Dn,ddof=1)
print('valeur moyenne de delta n:',{Dnmoy})
print('ecart type : ', {uDn})

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

Figure 21

— On obtient :

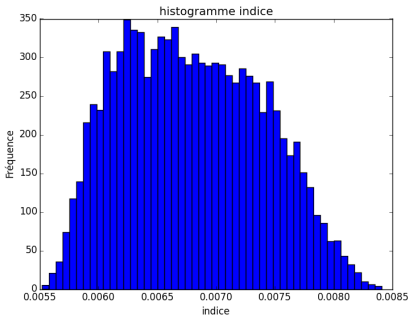


Figure 22 – Histogramme des valeurs de δn

Avec :

$$\delta n = 0.0076 \pm 0.0008$$

2.3.3 Protocole 2 : observation du spectre cannelé à l'aide d'un goniomètre

On utilise ici une autre lame ($e = 39\,\mu$m, $u_e = 1\mu m$, $N = 20 \pm 1$)**, on cherche** δn**.**

✘ On effectue le montage donné par la figure 23 : on peut, comme dans le protocole 1, compter les cannelures directement par observation dans la lunette.

✘ Toujours, avec le montage figure 23, on peut relever les longueurs d'ondes pour lesquelles on a extinction. Notons qu'au préalable, il faut avoir obtenu une courbe d'étalonnage avec une source connue (lampe spectrale au mercure ou à l'hélium).

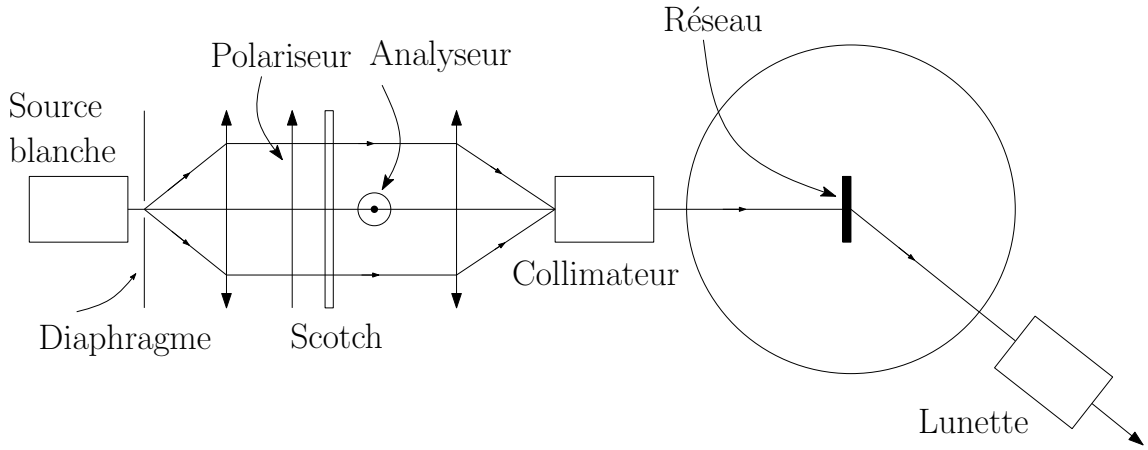


Figure 23

✘ Longueurs d'onde éteintes :

☛ On dispose un grand nombre d'épaisseur de lame de scotch ($N \Rightarrow \delta = Ne\Delta n$) et on observe la lumière à la sortie de l'analyseur ((P) et (A) étant croisés par exemple à 45 ° des lignes neutres du scotch) :

$$\delta = Ne\Delta n = p\lambda$$

$$\Rightarrow p = \frac{Ne\Delta n}{\lambda}$$

☛ Pour relier L'ordre p au numéro de la cannelure, noté k, on effectue le changement de variable : $p = -k + m_0$ [2]

2. Le signe moins est là car quand λ augmente p diminue et on va compter les extinctions en partant du bleu.

☛ On en déduit :

$$k = m_0 - \frac{Ne\Delta n}{\lambda}$$

☛ On trace alors k en fonction de $\frac{1}{\lambda}$: on doit obtenir une droite de pente $\alpha = -Ne\Delta n$.

☛ Connaissant les incertitudes sur α, N, e, on en déduit la mesure de Δn :

$$\Delta n = -\frac{\alpha}{Ne}$$

$$u_{\Delta n} = \Delta n \sqrt{\left(\frac{u_N}{N}\right)^2 + \left(\frac{u_e}{e}\right)^2 + \left(\frac{u_\alpha}{\alpha}\right)^2}$$

✘ Obtention de la courbe d'étalonnage :

☛ On règle le goniomètre et on place le réseau sur le plateau (incidence quasi-normale).

☛ On utilise une lampe spectrale à l'hélium pour tracer la courbe d'étalonnage : on se place à l'ordre 1 et on relève les angles correspondants aux différentes longueurs d'onde connues :

Angle θ lu sur le vernier °	Longueur d'onde nm
120.0	402.6
119.0	501.6
117.6	587.6
116.2	706.5

On trace alors $\sin\theta$ en fonction de λ, on est en effet supposé obtenir une droite [3]. C'est cette droite qui sera utilisée pour l'étalonnage (plutôt qu'une méthode graphique).

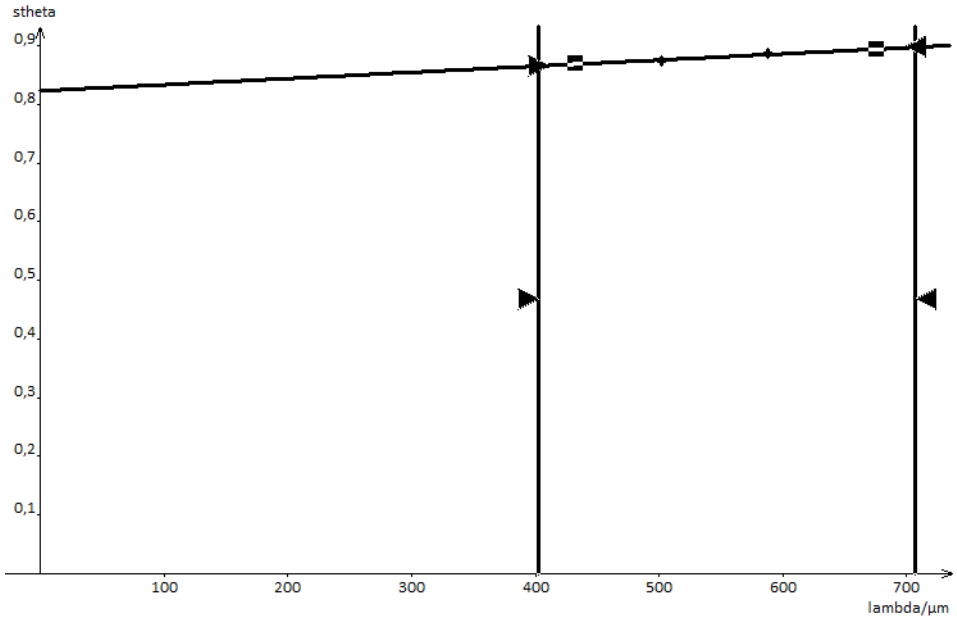


Figure 24 – Régression du type $y = ax + b$ avec $y = \sin\theta$, $r = 0.9968$, $x = \lambda$, $a = 0.105\,\mu\text{m}^{-1}$ et $b = 0.823$

✘ Résultats :

☛ On remplace la lampe spectrale par le montage décrit figure 23 : on observe à nouveau à l'ordre 1 un spectre cannelé.

3. En effet : $\sin\theta - \sin\theta_0 = \frac{\lambda}{a}$

☛ En partant du violet, on note le numéro de la cannelure (k) et la valeur de l'angle correspondant. A l'aide de la courbe d'étalonnage, on en déduit la longueur d'onde correspondante.

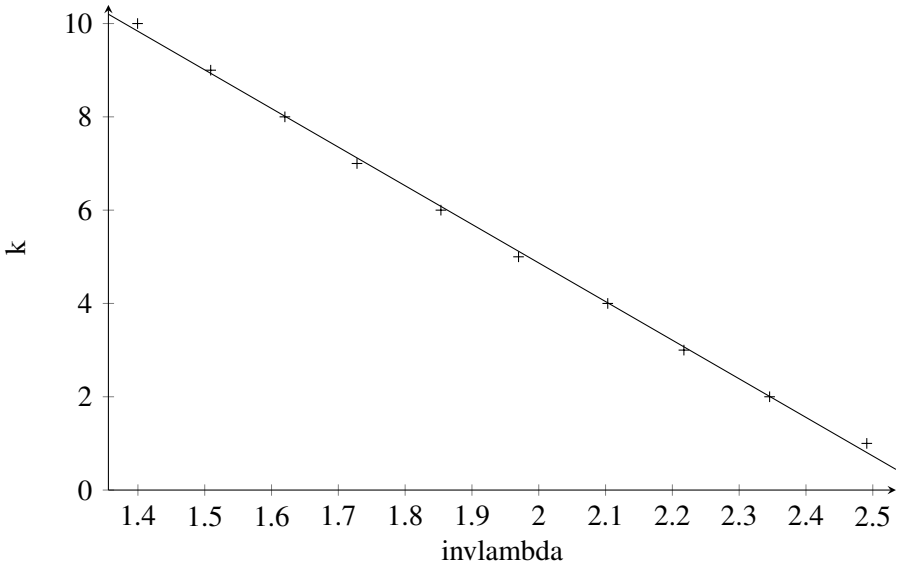
k	Angle lu sur le vernier °	Longueur d'onde : $\lambda = \dfrac{\sin\theta - b}{a}$ en nm	$\frac{1}{\lambda}$ en μm^{-1}
1	120.1	401.4	2.491
2	119.8	426.3	2.346
3	119.5	451.0	2.217
4	119.2	475.4	2.103
5	118.8	507.7	1.970
6	118.4	539.5	1.854
7	117.9	578.7	1.728
8	117.4	617.3	1.620
9	116.8	662.7	1.509
10	116.1	714.5	1.399

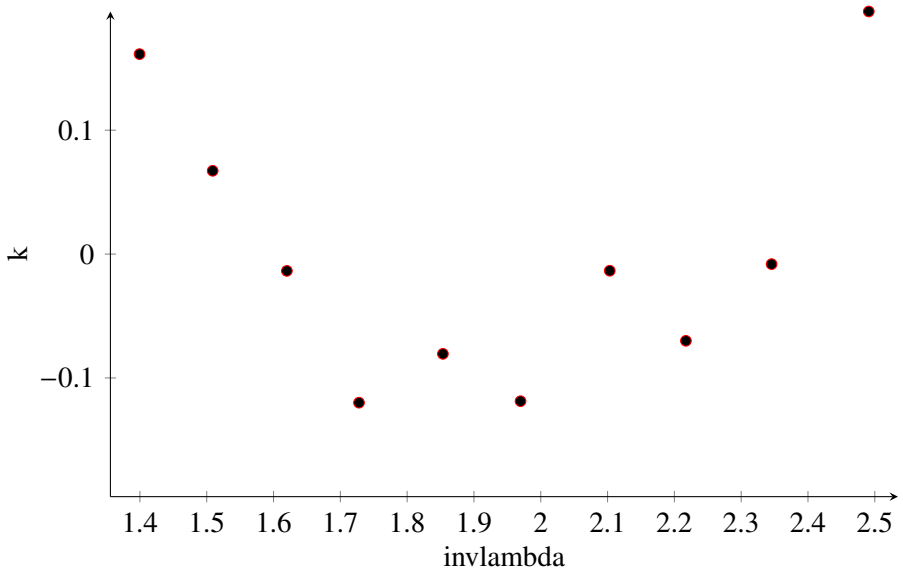
☛ On trace alors k en fonction de $\dfrac{1}{\lambda}$, et on procède comme indiqué dans la partie théorique.

$$lambda = \frac{\sin(theta) - 0.823}{0.105}$$

$$invlambda = \frac{1}{lambda}$$

$$k = a \cdot invlambda + b$$





Ecart-type sur k=116,8 10⁻3
Intervalle de confiance à 95%
a=(-8,28 ±0,25)
b=(21,4 ±0,5)

☛ En utilisant :

$$\Delta n = -\frac{a}{Ne}$$

$$u_{\Delta n} = \Delta n \sqrt{\left(\frac{u_N}{N}\right)^2 + \left(\frac{u_e}{e}\right)^2 + \left(\frac{u_\alpha}{\alpha}\right)^2}$$

Avec $N = 20$, on obtient :

$$\Delta n = 0.0106$$

$$u_{\Delta n} = 0.0003$$

2.3.4 Protocole 3 : observation du spectre à l'aide d'une fibre optique et d'un spectromètre

On utilise ici une autre lame ($e = 39\,\mu$m, $u_e = 1\mu m$, $N = 20 \pm 1$), on cherche δn.

✘ On peut effectuer le montage donné par la figure 25 : on observe le spectre directement sur l'écran de l'ordinateur.

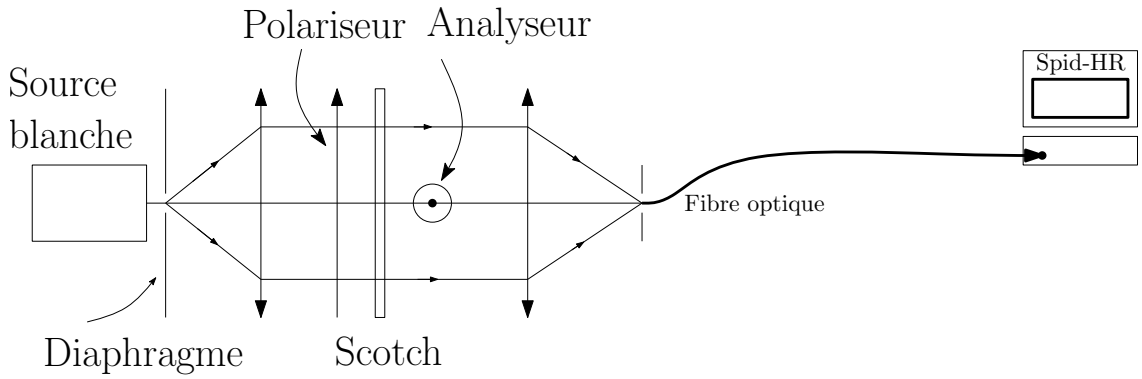


Figure 25

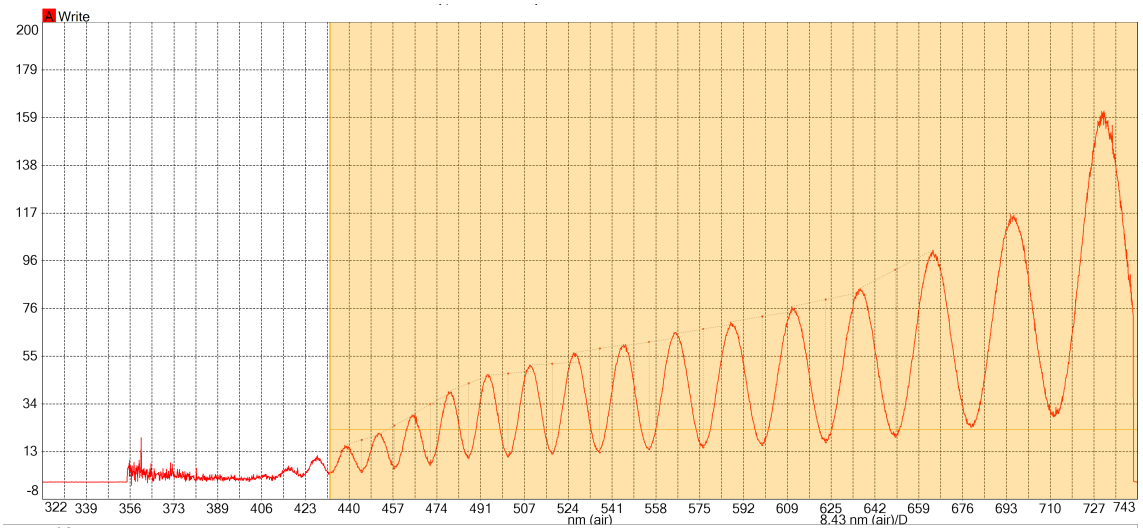


Figure 26 – Spectre cannelé observé avec le spectromètre Thorlab

✘ Le logiciel utilisé nous donne directement les longueurs d'onde éteintes :

Minimum Wavelength [nm (air)]
650.1423950
623.3731079
598.9344482
576.2343140
555.3877563
536.3858643
518.1671143
501.0783081
485.9197388
471.1771545
457.3075867
444.7584534

Figure 27 – Longueurs d'onde éteintes

✘ Longueurs d'onde éteintes :

☛ On dispose un grand nombre d'épaisseur de lame de scotch ($N \Rightarrow \delta = Ne\Delta n$) et on observe la lumière à la sortie de l'analyseur ((P) et (A) étant croisés par exemple à 45 ° des lignes neutres du scotch) :

$$\delta = Ne\Delta n = p\lambda$$

$$\Rightarrow p = \frac{Ne\Delta n}{\lambda}$$

☛ Pour relier L'ordre p au numéro de la cannelure, noté k, on effectue le changement de variable : $p = -k + m_0$ [4]

☛ On en déduit :

$$\frac{1}{\lambda} = \frac{1}{Ne\Delta n}(m_0 - k)$$

☛ On trace alors $\frac{1}{\lambda}$ en fonction de k : on doit obtenir une droite de pente $\alpha = -\frac{1}{Ne\Delta n}$.

☛ Connaissant les incertitudes sur α, N, e, on en déduit la mesure de Δn :

$$\Delta n = -\frac{1}{\alpha Ne}$$

4. Le signe moins est là car quand λ augmente p diminue et on va compter les extinctions en partant du bleu.

$$u_{\Delta n} = \Delta n \sqrt{\left(\frac{u_{Ne}}{Ne}\right)^2 + \left(\frac{u_\alpha}{\alpha}\right)^2}$$

✘ **Méthode 1 : Détermination de Δn sans les incertitudes sur λ**

lambda	k	$L = \frac{1}{\lambda}$
m		$m^{-}1$
$4{,}440 \cdot 10^{-}7$	1,000	$2{,}252 \cdot 10^{6}$
$4{,}570 \cdot 10^{-}7$	2,000	$2{,}188 \cdot 10^{6}$
$4{,}710 \cdot 10^{-}7$	3,000	$2{,}123 \cdot 10^{6}$
$4{,}850 \cdot 10^{-}7$	4,000	$2{,}062 \cdot 10^{6}$
$5{,}010 \cdot 10^{-}7$	5,000	$1{,}996 \cdot 10^{6}$
$5{,}180 \cdot 10^{-}7$	6,000	$1{,}931 \cdot 10^{6}$
$5{,}360 \cdot 10^{-}7$	7,000	$1{,}866 \cdot 10^{6}$
$5{,}550 \cdot 10^{-}7$	8,000	$1{,}802 \cdot 10^{6}$
$5{,}760 \cdot 10^{-}7$	9,000	$1{,}736 \cdot 10^{6}$
$5{,}980 \cdot 10^{-}7$	10,00	$1{,}672 \cdot 10^{6}$
$6{,}230 \cdot 10^{-}7$	11,00	$1{,}605 \cdot 10^{6}$
$6{,}500 \cdot 10^{-}7$	12,00	$1{,}538 \cdot 10^{6}$

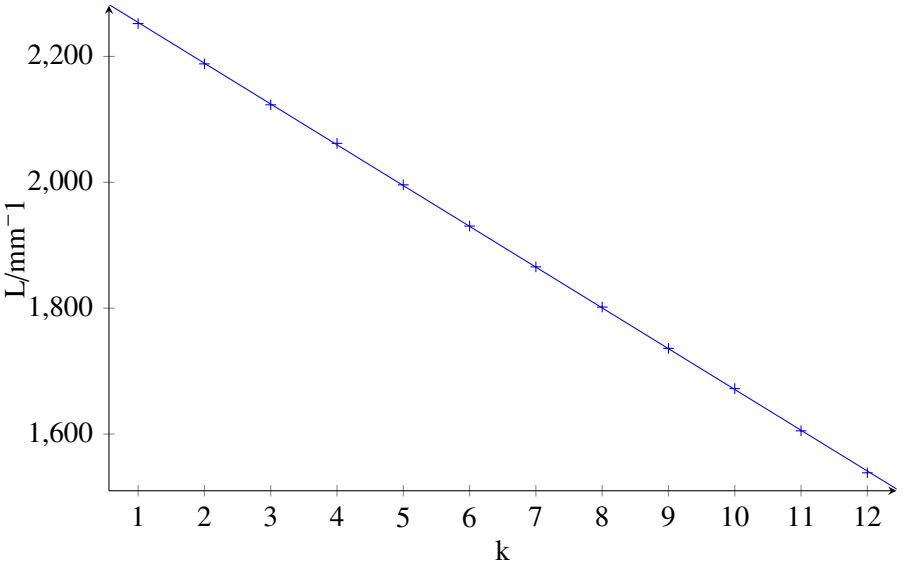


Figure 28 – Résultat de la régression linéaire sans incertitudes

Résultats méthode 1

Ecart-type sur L=1,588 $mm^{-}1$
Coeff. corrélation=-0,99998
Intervalle de confiance à 95%
a=(-64,80 ±0,30)mm^{-1}
b=(2,3188 ±0,0022)μm^{-1}

D'où :

$$\Delta n = -\frac{1}{aNe} = 9.9 \times 10^{-3}$$

$$u_{\Delta n} = 0.2 \times 10^{-3}$$

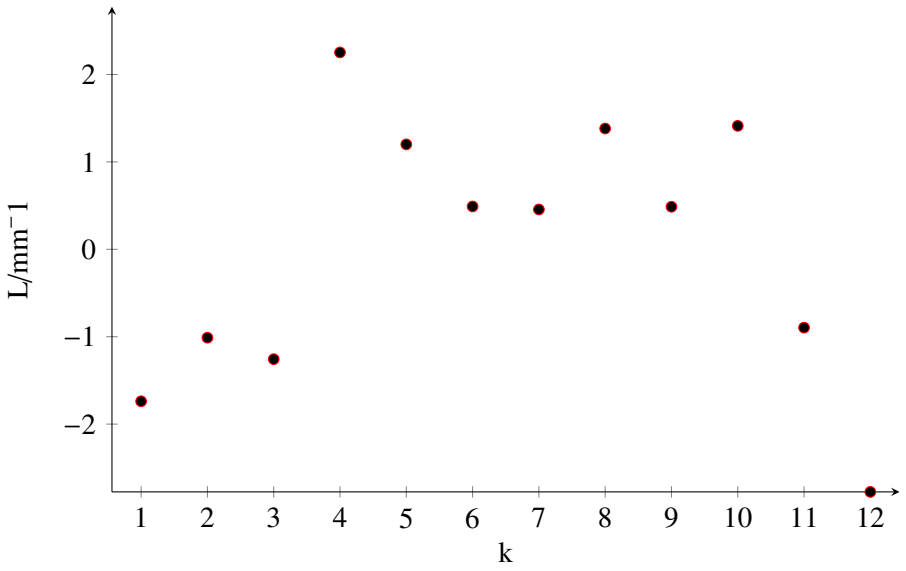


Figure 29 – Résidus avec la méthode 1

✘ **Méthode 2 : Détermination de Δn avec les incertitudes sur λ**

On estime l'incertitude sur λ en tenant compte de la résolution du spectromètre :

$$u_\lambda = 0.2\,\text{nm}$$

λ	u_λ	k	$L = \frac{1}{\lambda}$	u_L
m	m		m⁻1	m⁻1
4,440·10⁻7	0,00000000020	1,000	2,252·10⁶	1·10³
4,570·10⁻7	0,00000000020	2,000	2,188·10⁶	9,6·10²
4,710·10⁻7	0,00000000020	3,000	2,123·10⁶	9·10²
4,850·10⁻7	0,00000000020	4,000	2,062·10⁶	8,5·10²
5,010·10⁻7	0,00000000020	5,000	1,996·10⁶	8·10²
5,180·10⁻7	0,00000000020	6,000	1,931·10⁶	7,5·10²
5,360·10⁻7	0,00000000020	7,000	1,866·10⁶	7·10²
5,550·10⁻7	0,00000000020	8,000	1,802·10⁶	6,5·10²
5,760·10⁻7	0,00000000020	9,000	1,736·10⁶	6·10²
5,980·10⁻7	0,00000000020	10,00	1,672·10⁶	5,6·10²
6,230·10⁻7	0,00000000020	11,00	1,605·10⁶	5,2·10²
6,500·10⁻7	0,00000000020	12,00	1,538·10⁶	4,7·10²

Modélisation

$$L = a \cdot k + b$$

Résultats méthode 2

$\chi^2/(N-p)$=2,40
Intervalle de confiance à 95%
a=(-64,95 ±0,13)mm^{-1}
b=(2,3199 ±0,0012)μ m^{-1}

D'où :

$$\Delta n = -\frac{1}{aNe} = 9.9 \times 10^{-3}$$

$$u_{\Delta n} = 0.3 \times 10^{-3}$$

✘ Conclusion

Les deux méthodes donnent des résultats proches et cohérents, les résidus sont très bons et le χ^2 montre que l'incertitude sur λ a été légèrement sous estimée.

3 Annexes

```python
# Importation des bibliothèques
import numpy as np
import scipy.optimize as so
import matplotlib.pyplot as plt

# Données : on rentre theta et I (y) puis cos**2theta (x)

theta = np.pi*np.array([0,10,20,30,40,50,60,70,80,90])/180

I=np.array([0.4,0.37,0.32,0.26,0.19,0.12,0.09,0.005,0.001,0])

cos2theta=(np.cos(theta))**2


#forcer la regression linéaire (on utilise ensuite curfit)
def fct(x,k):
    return k*x
#regression avec curve_fit : la diagonale de la matrice covariance (pcov) nous donne
la variance des différents paramètres
popt , pcov = so.curve_fit(fct,cos2theta, I)
pente=popt[0]
u_pente = np.sqrt(np.diag(pcov)[0])
Imod = pente*cos2theta

print(pente)
print(u_pente)

# Tracé des points et du modèle
plt.figure("Loi de Malus")
plt.plot(cos2theta,I,'bo',label="points expérimentaux")
plt.plot(cos2theta, Imod, 'r-', label="Droite de régression")
plt.xlabel("cos2theta")
plt.ylabel("I")
plt.legend(loc="upper center")
plt.grid()

# graphique 2: les résidus
plt.figure("Résidus")
plt.plot(cos2theta, I-Imod, "g", label="residus")
plt.xlabel("cos2theta")
plt.ylabel("Résidus")
plt.legend(loc="upper center")
print("les résidus sont paraboliques")
```

Code python avec incertitudes et chi2

```
# -*- coding: utf-8 -*-

# Importation des bibliothèques

import scipy.optimize as spo
import scipy.stats as sstats

import numpy as np
from scipy import *
import matplotlib.pyplot as plt
# ce code permet de faire une régression linéaire sans utiliser la méthode Monté
Carlo et de calculer le chi^2

# Données : on rentre theta et I (y) puis cos**2theta (x) et leur incertitudes

## Code pour la régression linéaire permettant d'obtenir les incertitudes sur la
pente  et le chi^2
def reg_lin(x,y,ux,uy):
    """Effectue une regression lineaire y = ax+b.
      Affiche a, b, les incertitudes types et elargies, le chi2 reduit, etc.
      Produit un graphique avec barres d'erreurs.
       - x et y sont les donnees
       - ux et uy sont les incertitudes types sur x et y
       Ne permet pas de pendre en compte la covariance entre x et y.

    """

    # Fonction f decrivant la courbe a ajuster aux donnees
    def f(x,a):
        return a*x


    # Fonction d'ecart ponderee par les erreurs (ne prend pas en compte cov(x,y) car
variables indépendantes)
    def residual(a, y, x):
        return (y-f(x,a))/np.sqrt(uy**2 + (a*ux)**2)

    # Estimation initiale des parametres.
    # a changer si resultat aberrant.
    p0 = 0

    # Minimisation de la quantite residual.
    # On utilise l'algorithme des moindres carres non-lineaires
    # disponible dans la biliotheque scipy
    result = spo.leastsq(residual, p0, args=(y, x), full_output=True)

    # On obtient :
    # les parametres d'ajustement optimaux
    a = result[0];
    # la matrice de variance-covariance estimee des parametres
    acov = result[1];
    # les incertitudes-types sur ces parametres
    ua = np.sqrt(np.abs(np.diagonal(acov)))


    # Calcul de la valeur du "chi2 reduit" pour les parametres ajustes
    chi2 = np.sum(np.square(residual(a,y,x)))
    chi2r = chi2/(x.size-a.size)


    # Affichage des resultats
    print("---- regression lineaire ----")
    print("a = " + str("%.4f" % a) + ", u(a) = " + str("%.4f" % ua))
    print("chi2 = " + str("%.2f" % chi2))
    print("chi2 reduit = " + str("%.2f" % chi2r))
    print("----")
```

Code python avec incertitudes et chi2

```
    # Trace de x en fonction de y.
    plt.figure(1)
    plt.errorbar(x, y, xerr=ux, yerr=uy, fmt='o', label="(barres d'erreur)")
    plt.plot(x,a*x,label="y = " + str("%.2f" % a)+" x " )
    plt.xlabel(u'x')
    plt.ylabel(u'y')
    plt.grid()
    plt.legend(loc='best')
    plt.show()

    #Trace des résidus.
    z = (y-a*x)
    plt.figure(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()

    #Trace des résidus normalisés
    z_n = (y-a*x)/uy
    plt.figure(3)
    plt.errorbar(x, z_n, xerr=ux, yerr=uy, fmt='o', label="(barres d'erreur)")
    #plt.plot(x, z_n, '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(min(z_n)-1, max(z_n)+1)
    plt.grid()
    plt.show()

# ---- Application ----
theta = np.pi*np.array([0,10,20,30,40,50,60,70,80,90])/180
utheta=np.pi*1/(2*np.sqrt(3))*ones(len(theta))/180
I=np.array([0.4,0.37,0.32,0.26,0.19,0.12,0.09,0.005,0.001,0.001])
uI=np.array([0.013,0.012,0.011,0.0090,0.007,0.005,0.004,0.002,0.001,0.001])
cos2theta=(np.cos(theta))**2
ucos2=2*np.cos(theta)*np.sin(theta)*utheta

reg_lin(cos2theta,I,ucos2,uI)
```

Code python avec incertitudes et MC

```
# -*- coding: utf-8 -*-

"""Régression linéaire."""

# Importation des bibliothèques
import numpy as np
import numpy.random as rd
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
from scipy import *

# Données

theta = np.pi*np.array([0,10,20,30,40,50,60,70,80,90])/180
utheta=np.pi*1/(2*np.sqrt(3))*ones(len(theta))/180
I=np.array([0.4,0.37,0.32,0.26,0.19,0.12,0.09,0.005,0.001,0.001])
uI=np.array([0.013,0.012,0.011,0.0090,0.007,0.005,0.004,0.002,0.001,0.001])
cos2theta=(np.cos(theta))**2
ucos2=2*np.cos(theta)*np.sin(theta)*utheta


#forcer la regression linéaire (on utilise ensuite curfit)
def fct(x,k):
    return k*x

# Nombre de simulations
N = 10000
# Détermination des pentes et ordonnées à l'origine
asim = []
for i in range(N):
    Xsim = cos2theta + ucos2*rd.uniform(-1, 1, len(cos2theta))
    Ysim = I + uI*rd.uniform(-1, 1, len(I))
    pente = curve_fit(fct,Xsim, Ysim)
    asim.append(pente[0])

ma, ua = np.mean(asim), np.std(asim, ddof=1)

print("a =", ma)
print("u(a) =", ua)

# Tracé des points et du modèle
Imod = ma*cos2theta
plt.figure(1)
plt.subplot(1, 2, 1)
plt.errorbar(cos2theta, I, xerr=ucos2, yerr=uI, fmt='bo', label="Points
expérimentaux")
plt.plot(cos2theta, Imod, 'c--', label="Modèle linéaire")
plt.xlabel(r"$cos2theta")
plt.ylabel(r"$I$")
plt.grid(), plt.legend(loc='best')

    #Trace des résidus.
z= (I-ma*cos2theta)
plt.figure(2)
plt.plot(cos2theta, z, 'bo')
plt.fill_between([min(cos2theta)-1, max(cos2theta)+1], y1=-2, y2=2, color='cyan',
alpha=0.1)
plt.xlabel(r"$cos2theta$"), plt.xlim(min(cos2theta)-1, max(cos2theta)+1)
plt.ylabel(r"$Z$"), plt.ylim(-3, 3)
plt.grid()
plt.show()

    #Trace des résidus normalisés
z_n = (I-ma*cos2theta)/uI
plt.figure(3)
plt.errorbar(cos2theta, z_n, xerr=ucos2, yerr=uI, fmt='o', label="(barres d'erreur)")
plt.fill_between([min(cos2theta)-1, max(cos2theta)+1], y1=-2, y2=2, color='cyan',
```

Code python avec incertitudes et MC

```
alpha=0.1)
plt.xlabel(r"$cos2theta$"), plt.xlim(min(cos2theta)-1, max(cos2theta)+1)
plt.ylabel(r"$Z$"), plt.ylim(min(z_n)-1, max(z_n)+1)
plt.grid()
plt.show()
```

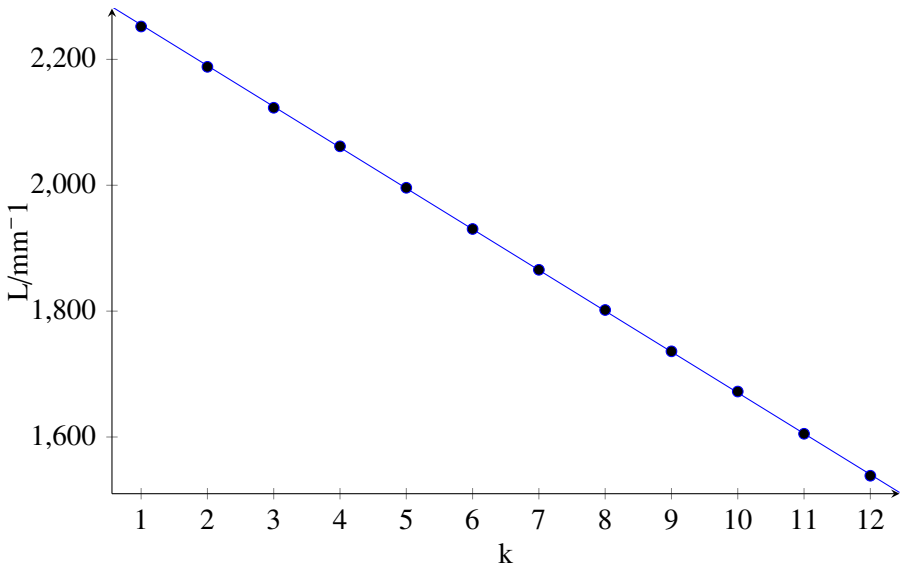


Figure 30 – Résultat de la régression linéaire avec la méthode 2

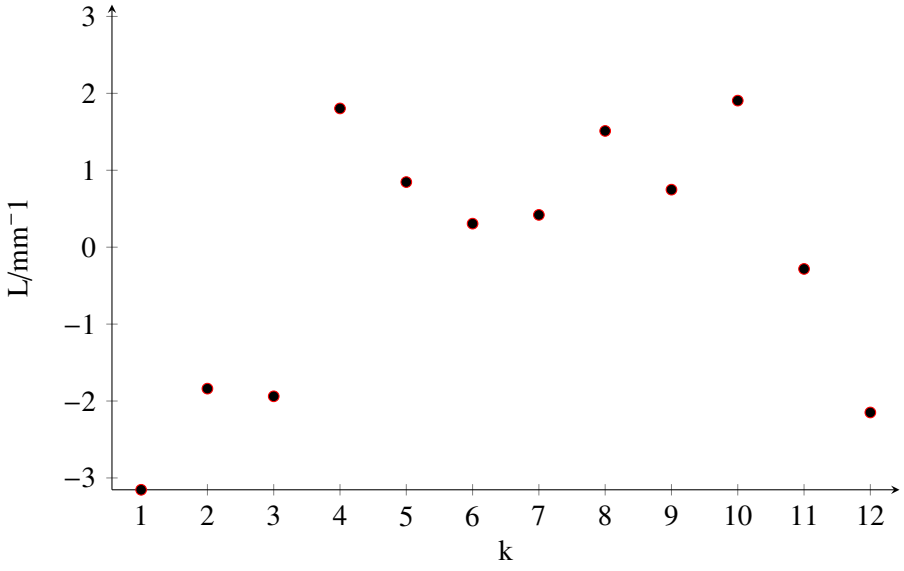


Figure 31 – Résidus avec la méthode 2