POLYTECHNIQUE
MONTRÉAL

UNIVERSITÉ
D'INGÉNIERIE

PHS2223 – Introduction à l'optique moderne

Équipe : 04

Expérience 1

Microscopie confocale

Présenté à
Guillaume Sheehy
Esmat Zamani

Par :
Émile **Guertin-Picard** (2208363)
Laura-Li **Gilbert** (2204234)
Tom **Dessauvages** (???????)

25 septembre 2024
Département de Génie Physique
Polytechnique Montréal

Table des matières

1 Introduction

2 Théorie

2.1 Microsopie Confocale

La microscopie confocale est une technique d'imagerie avancée qui permet l'obtention d'images à hautes résolution, en éliminant la lumière hors-focus, particulièrement ceux à l'extérieur du plan focal. Le fonctionnement de cette technologie repose sur les principes d'illumination et de détection limités à un certain volume par la présence d'un simple trou, nommé sténopé (Source 1). En d'autres termes, ces microscopes fonctionnent de la même manière que ceux conventionnels à l'exception de ce trou. De cette manière, une source de lumière, telle qu'un laser, est émise à travers le système, et celle-ci est, ensuite, dirigée vers des miroirs dichroïques qui permettent la sélection de certaines longueurs d'onde réfléchies et la réjection des autres (Source 2). Une fois redirigée par ces miroirs, la lumière réfléchie passe à travers une première lentille, concentrant ainsi le faisceau en un point précis, soit l'objet. Arrivée à l'échantillon, une partie de ce faisceau est réfléchi par l'objet et, par le fait même, retransmis dans le système optique. La lumière transmise par l'échantillon se dirige, donc, à nouveau vers les miroirs qui la renvoie vers le sténopé, situé derrière une deuxième lentille permettant, elle-aussi, la convergence du rayon lumineux. Le rayon traverse, ensuite, le sténopé, et atteint le détecteur. Puisque l'oeil nu n'est pas en mesure de traiter l'image résultante, le détecteur reconstruit indirectement l'image à l'aide de balayages de l'échantillon. De ce fait, le faisceau se déplace, selon les différentes positions axiales, sur l'objet afin d'obtenir une image complète (Source 3).

2.2 Sténopé

Le sténopé correspond à une petite ouverture permettant d'éliminer les faisceaux ne provenant pas du plan focal. En d'autres termes, la lumière entrante est filtrée par l'ouverture, de sorte que l'aberration causée par la divergence des rayons du système, et ne laisse passer qu'un rayon pour chaque point de l'objet. Ainsi, l'ajout de cet outil permet d'améliorer la résolution axiale, en augmentant la netteté des images obtenues.

2.3 Résolution axiale

La résolution axiale, en microscopie confocale, correspond à la capacité d'un système à distinguer deux points situés à différentes profondeurs sur l'axe optique. En d'autres termes, elle correspond à la distance minimale à laquelle ces deux points peuvent être distingués comme distincts (Source 4). Cette donnée dépend, généralement, de quelques paramètres tels que la longueur d'onde de la lumière utilisée, de l'ouverture numérique du sténopé, et du système optique (Source 5). Pour cette expérience, les paramètres d'ouverture et de système optique sont considérés pour déterminer la résolution.

2.4 Lentilles convergentes

Les lentilles optiques ont pour fonction de concentrer ou de disperser les rayons qui les traversent à l'aide du principe de réfraction, soit la redirection des faisceaux lumineux. Dans le cas des lentilles convergentes, les rayons de lumière sont redirigés de sorte que ceux-ci se concentrent en un point spécifique, appelé le point focal (Source 6). Ce point est, généralement, donné par un paramètre : la distance focale. En effet, la distance focale correspond à la longueur entre le centre de la lentille et le point focal, soit l'endroit où convergent les faisceaux. Ainsi, une courte distance, comme dans le cas de cette expérience, place l'image à proximité de la lentille elle-même.

En microscopie confocale, les lentilles convergentes sont utilisées afin de concentrer la lumière et d'agrandir l'image des objets petits. Ainsi, le point focal est utilisé pour sélectionner un plan spécifique de l'échantillon à observer, et la lumière réfléchie par l'objet, soit le point focal, est filtrée par le sténopé, ce qui permet la formation nette des images (Source 7).

2.5 Approximation des rayons

L'approximation des rayons consiste à supposer que la lumière se déplace, dans un milieu homogène, en ligne droite perpendiculairement au front d'ondes (Source 8). En d'autres termes, avec cette approximation, l'onde, se déplaçant à travers une interface, est considérée comme une ligne droite se propageant dans la direction de ses rayons. De cette manière, cette théorie permet à la lumière d'effectuer des réflexions et des réfractions lorsqu'elle entre en contact avec des surfaces (Source 9). Cette supposition permet, ainsi, d'utiliser l'optique géométrique pour déterminer le parcours de ces dits faisceaux lumineux.

2.6 Modèle mathématique de calcul de résolution

Une méthode essentielle pour la résolution de problèmes optiques consiste à la méthode des matrices. Celle-ci repose sur la représentation visuelle des rayons à travers le système optique, qui, dans le cas de cette expérience, provient de l'énoncé du laboratoire (Source 3).

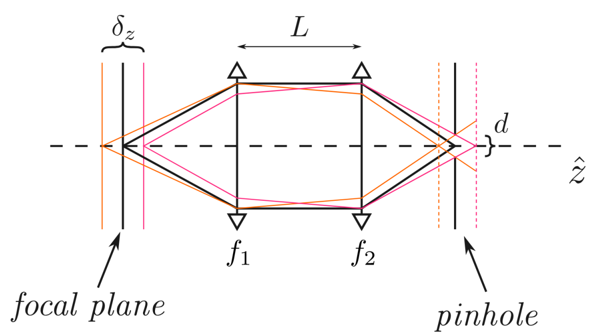

Figure 1 : Schéma illustrant les rayons passant aux extrémités du sténopé

À partir de ce schéma, il est possible de mettre le *input plane* (IP), qui correspond à l'entrée des rayons lumineux, au plan focal, et le *output plane* (OP), soit la sortie de ceux-ci, au sténopé. Ainsi, il est possible de poser une matrice de transfert M au système de la façon suivante :

$$M = M_{t3} M_{f2} M_{t2} M_{f1} M_{t1} \tag{1}$$

où les matrices sont les suivantes :

- M_{t1} est la matrice de translation entre le plan focal et la lentille 1
- M_{f1} est la matrice de lentille mince de la lentille 2
- M_{t2} est la matrice de translation entre les deux lentilles
- M_{f2} est la matrice de lentille mince de la lentille 2
- M_{t3} est la matrice de translation entre le plan focal et la lentille 1

Une matrice de translation se décrit par :

$$M_t = \begin{bmatrix} 1 & x \\ 0 & 1 \end{bmatrix} \tag{2}$$

où x est la longueur de la translation selon l'axe optique. Un matrice de lentille mince se décrit par :

$$M_f = \begin{bmatrix} 1 & 0 \\ -\frac{1}{f} & 1 \end{bmatrix} \tag{3}$$

où f est la distance focale de la lentille. Soit r_i, un rayon de lumière initial au plan focal. Ce dernier se décrit par sa hauteur et par son angle par rapport à l'axe optique $\hat{z}$:

$$r_i = \begin{bmatrix} y_i \\ \alpha_i \end{bmatrix} = \begin{bmatrix} 0 \\ \alpha_i \end{bmatrix} \tag{4}$$

La hauteur y_i est nulle car le rayon de lumière commence sur l'axe optique. Soit aussi r_f, le rayon final qui arrive au sténopé :

$$r_f = \begin{bmatrix} y_f \\ \alpha_f \end{bmatrix} \tag{5}$$

C'est avec la définition de ce rayon final qu'il est possible de développer le modèle pour la résolution axiale δ_z. En effet, tel que visible à la figure 1, deux rayons finaux peuvent être dessinés à partir des deux extrémités du sténopé. Ces deux derniers convergent à des plans focaux qui ont une distance différente avec la première lentille. La différence entre ces distances est la résolution axiale recherchée.

Soit le cas 1, où l'on dénote la hauteur du rayon final au sténopé $y_{f1} = d/2$. Ce rayon est illustré en rouge. Il est possible de développer l'équation matricielle suivante :

$$r_{f1} = M r_{i1} \tag{6}$$

$$\Rightarrow \begin{bmatrix} \frac{d}{2} \\ \alpha_{f1} \end{bmatrix} = M \begin{bmatrix} y_{i1} \\ \alpha_{i1} \end{bmatrix} \tag{7}$$

L'angle α_{f1} se trouve par trigonométrie :

$$\alpha_{f1} = \arctan\left(\frac{\phi_{f_2} - d}{2f_2}\right) \tag{8}$$

où ϕ dénote le diamètre d'une lentille. Il est possible de dénoter la distance inconnue entre le plan focal du cas 1 et la lentille 1 par z_1. Cette dernière se trouve dans M_{t1}, et donc dans M. Du coup, à l'aide de la librairie de calcul symbolique *sympy*, il est possible de calculer aisément la matrice de transfert et de résoudre l'équation (6) avec la méthode `LUSolve`. Le résultat a donc la forme suivante :

$$r_{i1} = \begin{bmatrix} y_{i1}(z_1) \\ \alpha_{i1}(z_1) \end{bmatrix} \tag{9}$$

Or, sachant que :

$$y_{i1}(z_1) = 0 \tag{10}$$

La fonction sympy `solve` permet de résoudre cette équation symboliquement pour z_1, et après substitution, sa valeur numérique peut être connue.

Ces étapes de résolution se répètent pour le cas 2, où l'on dénote la hauteur du rayon final au sténopé $y_{f2} = -d/2$. Ce rayon est illustré en orange. Son angle se trouve également par trigonométrie :

$$\alpha_{f1} = \arctan\left(\frac{\phi_{f_2} + d}{2f_2}\right) \tag{11}$$

Il est donc possible en résolvant les mêmes équations mais pour le cas 2 d'obtenir z_2. Enfin, la résolution axiale se trouve ainsi :

$$\delta_z = |z_2 - z_1| \tag{12}$$

Cette valeur dépend donc de z, qui, à son tour, dépend des paramètres du montage physique.

3 Méthodologie

4 Hypothèses

Afin de prédire le comportement de la résolution axiale en fonction des différents paramètres physiques du système, un modèle de calcul ainsi qu'un programme python a été utilisé. Ces derniers sont présentés en annexe. Le programme python utilise d'abord la librairie numpy afin de définir des fonctions qui génèrent des matrices de translation ou de lentille mince. Ensuite, une fonction est créée pour calculer la résolution axiale en fonction de la largeur du sténopé, de la distance focale ainsi que de la distance séparant les lentilles, tel que décrit dans le modèle de calcul. Dans cette fonction, sympy est utilisé pour effectuer la résolution symbolique de la multiplication matricielle et des équations autant algébriques que matricielle. À la fin de cette fonction, les valeurs numériques sont substituées pour avoir une valeur numérique de résolution. Le programme utilise enfin matplotlib afin de de visualiser le résultat de cette fonction pour des plages de paramètres.

Le graphique présenté en figure 2 démontre la variation de la résolution axiale pour une plage de diamètre de sténopé et ce, pour cinq valeurs différentes de longueur focale.

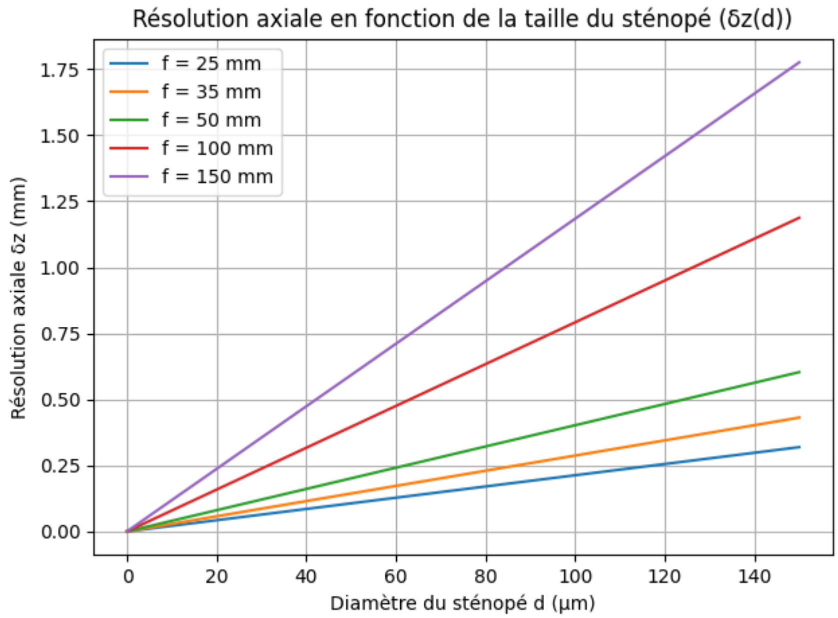

Figure 2 : Graphique de δ_z en fonction de d pour différentes valeurs de f.

" L'hypotèse présentée par le modèle de calcul est que la variation de d fait augmenter linéairement la résolution axiale. Ensuite, la figure 3 montre la variation de la résolution axiale pour une plage de longueur focale des deux lentilles, pour quatres valeurs différentes de diamètre de sténopé.

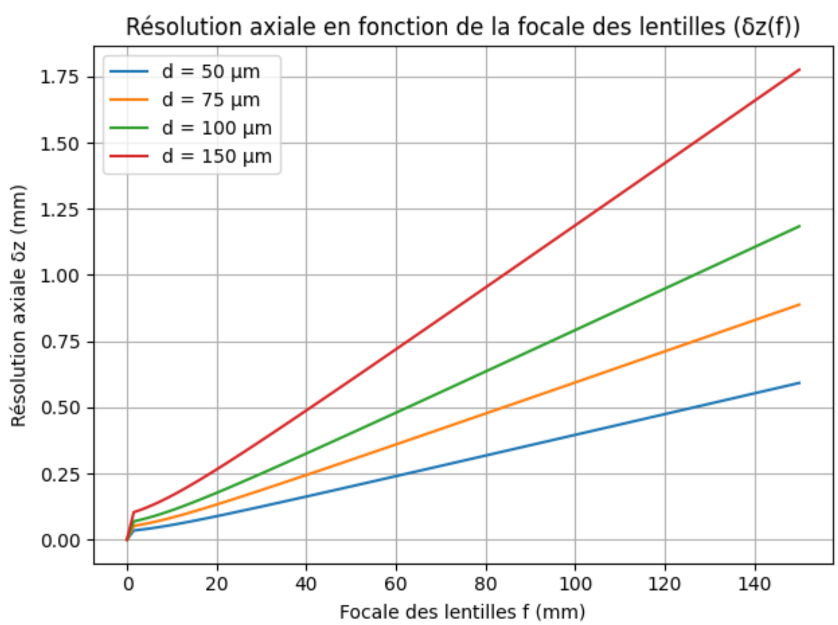

Figure 3 : Graphique de δ_z en fonction de f pour différentes valeurs de d.

Il est donc possible de prédire un comportement quasi-linéaire pour la résolution axiale en fonction de la focale. Ce comportement ne semble toutefois pas valide pour des valeurs de f très petites. Enfin, la figure

4 présente la résolution axiale en fonction de la distance entre les lentilles pour cinq valeurs différents de focales.

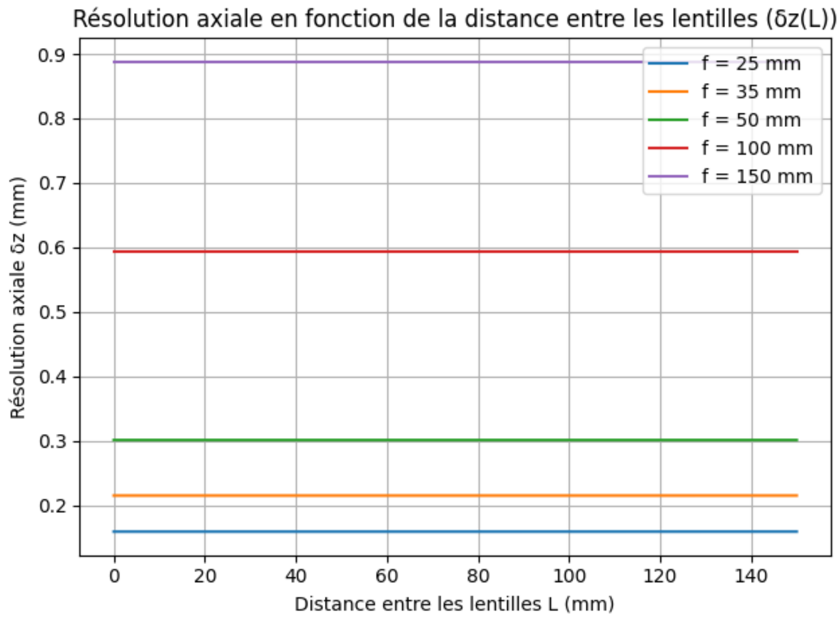

Figure 4 : Graphique de δ_z en fonction de L pour différentes valeurs de f.

L'hypothèse finale tirée du modèle est donc que la variation de L n'a pas d'impact sur la résolution.

5 Annexes

5.1 Code source

```
import numpy as np
import matplotlib.pyplot as plt
import sympy as sp

def Mt(d):
    '''Translation'''
    M = np.array([[1, d],
                  [0, 1]])
    return M

def Ml(f):
    '''Thin Lens'''
    M = np.array([[1, 0],
                  [-1/f, 1]])
    return M

def res_axial(pinhole, focal, lens_distance):
    """
    Calcule la resolution axiale en fonction du diametre du trou, de la focale, et de la
     distance entre les lentilles.

    Parametres :
    ------------
```

```
    pinhole : float
        Diametre du trou en metres.
    focal : float
        Longueur focale de la lentille en metres.
    lens_distance : float
        Distance entre les lentilles en metres.

    Retour :
    --------
    res : expression symbolique
        La resolution axiale en metres.
    """
    f = sp.symbols('f')
    phi = sp.symbols('phi')
    d = sp.symbols('d')
    L = sp.symbols('L')
    z = sp.symbols('z')

    # formation de la matrice de transfert
    M = Mt(f)@Ml(f)@Mt(L)@Ml(f)@Mt(z)
    M_sp = sp.Matrix(M)

    # definition des hauteurs et angles de chaque cote du pinhole
    yf1 = d/2
    alphaf1 = -sp.atan((phi-d)/(2*f))

    yf2 = -d/2
    alphaf2 = -sp.atan((phi+d)/(2*f))

    # vecteur decrivant le rayon final au pinhole
    vf1 = sp.Matrix([yf1, alphaf1])
    vf2 = sp.Matrix([yf2, alphaf2])

    # resolution des systemes matriciels puis de leur 1re
    # equation = 0 pour isoler z
    solve1 = M_sp.LUsolve(vf1)
    z1 = sp.solve(solve1[0], z)

    solve2 = M_sp.LUsolve(vf2)
    z2 = sp.solve(solve2[0], z)

    # substitution des valeurs numeriques
    values = {f:focal, phi:25.4e-3, d:pinhole, L:lens_distance}
    res = z2[0].subs(values) - z1[0].subs(values)

    return res

# set de courbes 1
focals = [25e-3, 35e-3, 50e-3, 100e-3, 150e-3]
pinhole_sizes = np.linspace(0, 150e-6, 100)
L = lambda f: f

plt.figure()

# Tracer une courbe pour chaque focale
for f in focals:
    res_values = [res_axial(d, f, L(f)) * 1e3 for d in pinhole_sizes]
    plt.plot(pinhole_sizes * 1e6, res_values, label=f'f = {f*1e3:.0f} mm')

```

```python
# Parametres d'affichage
plt.xlabel("Diametre du stenope d (micro m)")
plt.ylabel("Resolution axiale delta z (mm)")
plt.legend()
plt.grid(True)
plt.title("Resolution axiale en fonction de la taille du stenope (dz(d))")
plt.tight_layout()
plt.savefig("res_vs_pinhole.png")
plt.show()
plt.clf()

# set de courbes 2
pinhole_sizes = [50e-6, 75e-6, 100e-6, 150e-6]
focals = np.linspace(1e-6, 150e-3, 100)
L = lambda f: f  # si L(f) appele, L = f

plt.figure()

# Tracer une courbe pour chaque taille de stenope
for pinhole in pinhole_sizes:
    res_values = [res_axial(pinhole, f, L(f)) * 1e3 for f in focals]
    plt.plot(focals * 1e3, res_values, label=f'd = {pinhole*1e6:.0f} micro m')

# Parametres d'affichage
plt.xlabel("Focale des lentilles f (mm)")
plt.ylabel("Resolution axiale dz (mm)")
plt.legend()
plt.grid(True)
plt.title("Resolution axiale en fonction de la focale des lentilles (dz(f))")
plt.tight_layout()
plt.savefig("res_vs_focal.png")
plt.show()
plt.clf()

# set de courbes 3
pinhole = 75e-6  # 75 micro m
focals = [25e-3, 35e-3, 50e-3, 100e-3, 150e-3]  # en metres
lens_distances = np.linspace(0, 150e-3, 100)  # en metres

plt.figure()

# Tracer une courbe pour chaque taille de focale
for f in focals:
    res_values = [res_axial(pinhole, f, L) * 1e3 for L in lens_distances]
    plt.plot(lens_distances * 1e3, res_values, label=f'f = {f*1e3:.0f} mm')

# Parametres d'affichage
plt.xlabel("Distance entre les lentilles L (mm)")
plt.ylabel("Resolution axiale dz (mm)")
plt.legend()
plt.grid(True)
plt.title("Resolution axiale en fonction de la distance entre les lentilles (dz(L))")
plt.tight_layout()
plt.savefig("res_vs_L.png")
plt.show()
plt.clf()
```