

Serial Block Face Imaging

Kratoscope

Orson Vermare

Supervisé par :

Idier Jérôme¹

En collaboration avec :

P. Paul-Gilloteaux², P. De Villemagne³ et P. Dorval³

1. LS2N, CNRS

2. Plateforme MicroPICell (Univ. Nantes), CNRS

3. Kaer Labs

31 mars 2023



Table des matières

1	Introduction	2
2	Présentation du système	2
3	Modélisation Analytique	2
3.1	Obtention d'un modèle 3D	2
3.2	Cas d'un modèle 2D gaussien	4
4	Déconvolution	5
4.1	Critère de minimisation	5
4.2	Solution Algorithmique	5
5	Simulations numériques	5
5.1	Problème direct	5
5.2	Problème Inverse	6
6	Application aux données réelles	6
6.1	Identification du noyau	6
6.2	Déconvolution d'un échantillon réel	8
7	Améliorations et suite	8

1 Introduction

Ce projet est une collaboration entre MicroPICell, la société Kaer Labs et l'école Centrale de Nantes. MicroPICell est une plateforme d'imagerie biologique (basée au CHU de Nantes) qui développe un dispositif de Serial Block Face Imaging en partenariat avec Kaer Labs, une société commercialisant des instruments d'imagerie par fluorescence.

2 Présentation du système

Le principe de l'acquisition par Serial Block Face Imaging est simple, il s'agit de photographier la surface d'un échantillon subissant des coupes successives. Un traitement numérique permet ensuite de reconstruire le volume par empilement des images obtenues. Les échantillons sont placés dans une matrice de paraffine ayant un certain degré d'opacité mais aucune coloration n'est utilisée, seule la fluorescence naturelle des échantillons produit le signal lumineux acquis [Mallocci et al., 2022].

Cependant, les images acquises sont dégradées par la fluorescence sub-surfacique. En effet, la paraffine ne pouvant être complètement opaque, les éléments de l'échantillon sous la surface émettent eux-aussi par fluorescence. L'objectif de ce projet est de proposer un traitement numérique permettant de supprimer cette contribution sub-surfacique des images. Dans un premier temps une approche par déconvolution est étudiée comme proposé par [Walsh et al., 2020].

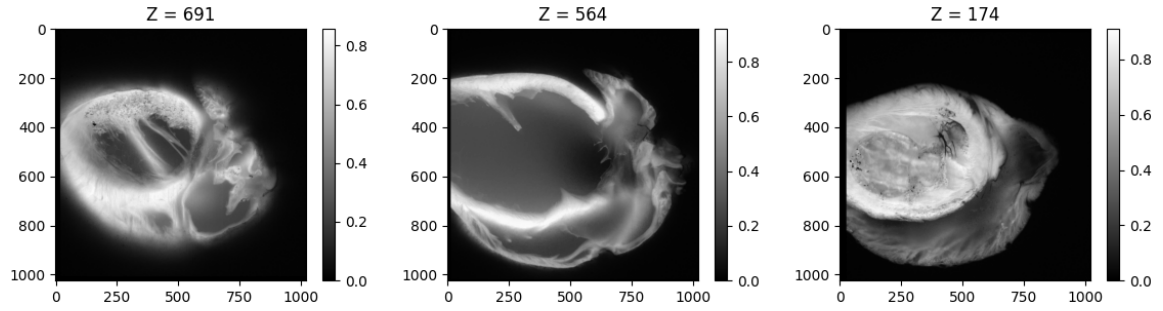


FIGURE 1 – Images obtenues sur un échantillon de coeur de souris

3 Modélisation Analytique

On cherche à réduire l'influence de la fluorescence sub-surfacique lors de l'acquisition par Serial Block Face imaging. Lors de l'acquisition, la surface de l'échantillon est observée sans flou (la résolution de l'instrument de mesure est suffisante) mais les éléments sous la surface sont également observés par transparence de la paraffine. L'acquisition étant faite coupe par coupe, on peut modéliser l'influence des couches sub-surfaciques par une approche discrète. Si on considère que la paraffine est homogène et isotrope (à notre échelle d'observation) alors la diffusion de la lumière est identique pour chaque couche (la profondeur de coupe étant constante). Autrement dit, l'image acquise est la somme de l'image de la surface et des images des coupes suivantes de plus en plus floues. Ainsi on peut modéliser le processus de mesure par une convolution.

3.1 Obtention d'un modèle 3D

Dans un repère $\mathcal{R} = (O; \vec{x}, \vec{y}, \vec{z})$ avec \vec{z} la profondeur, on considère :

- $O_z = O(., ., z)$ la coupe réelle à la profondeur z
- $I_z = I(., ., z)$ l'image de cette coupe
- h le noyau de convolution 3D

On considère :

- Un échantillon O de taille $L_x \times L_y \times D$
- Une image I de taille $P_x \times P_y \times M$
- Un noyau de convolution h de taille $2k+1 \times 2k+1 \times d$

Ainsi $x \in [0, L_x]$, $y \in [0, L_y]$ et $z \in [0, D]$. On notera N le nombre total de coupes et δz l'épaisseur de coupe ($\delta z \ll D$).

Cas Discret : Mathématiquement, le flou peut être défini à partir d'un produit de convolution 2D, selon les dimensions x et y , entre l'image réelle de la coupe à la profondeur z et le noyau $h_z = h(.,., z)$.

$$I_z(x, y) = \sum_{w=0}^{(N-1)} (O_{z-w} * h_w)(x, y) = \sum_{w=0}^{(N-1)} \iint_{u,v} O_{z-w}(x-u, y-v) h_w(u, v) du dv \quad (1)$$

Cas Continu : On peut re-écrire l'équation 1 dans le cas continu :

$$I(x, y, z) = (O * h)(x, y, z) = \iiint O(x-u, y-v, z-w) h(u, v, w) du dv dw \quad (2)$$

On cherche maintenant une modélisation adéquate du noyau de convolution $h(x, y, z)$. On sait que, pour une coupe O_z fixée, seules les coupes ayant un z plus petit ont une influence sur $I(x, y, z)$. On peut donc en déduire que :

$$\forall x, y, z / z < 0, h(x, y, z) = 0 \quad (3)$$

On sait que lors de l'acquisition, la surface du block est vue avec une modification faible, que l'on note g_0 , et la couche suivante avec une modification plus importante, que l'on note g . On obtient alors :

$$\forall x, y \quad h(x, y, 0) = g_0(x, y) \text{ et } h(x, y, \delta z) = (g_0 * g)(x, y) \quad (4)$$

En raisonnant par récurrence couche par couche on peut modéliser h comme ceci :

$$\forall x, y, n \in [0, N] \quad h(x, y, n\delta z) = (g_0 * g * \dots * g)(x, y) = (g_0 * (g)^{(n)})(x, y) \quad (5)$$

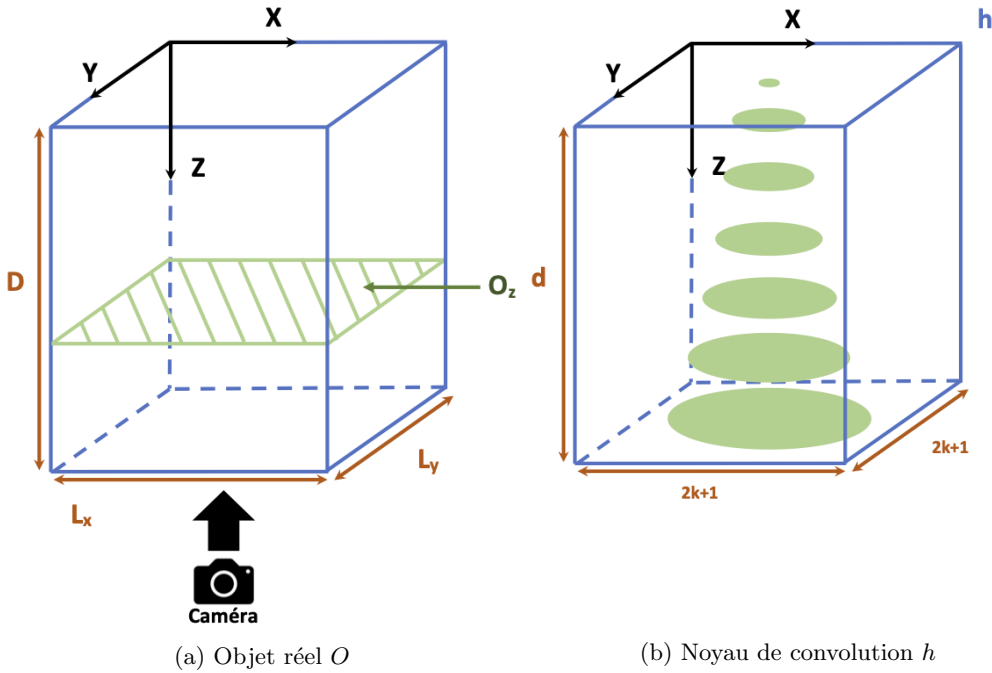


FIGURE 2 – Représentation schématique

3.2 Cas d'un modèle 2D gaussien

En première approximation on peut définir le noyau de convolution à partir de formes gaussiennes 2D selon x et y . On peut supposer que h_0 correspond alors à un flou gaussien avec un écart-type σ_0 faible. Dans les parties suivantes, $\sigma_0 = 0$ et $A_0 = 1$, ce qui amène que h_0 est un dirac.

$$\forall x, y \quad h(x, y, 0) = g_0(x, y) = g_{\sigma_0}(x, y) = \frac{A_0}{2\pi\sigma_0^2} \exp\left(-\frac{x^2 + y^2}{2\sigma_0^2}\right) \quad (6)$$

En raisonnant couche par couche, comme précédemment, on obtient un noyau de convolution ayant une forme gaussienne en (x, y) avec un écart type et une amplitude dépendants de la profondeur z . En effet, on sait que le résultat de la convolution d'une gaussienne par une gaussienne reste une gaussienne, plus précisément la convolution d'une gaussienne d'écart type σ_0 par une gaussienne d'écart type σ_1 est une gaussienne d'écart type $\sqrt{\sigma_0^2 + \sigma_1^2}$. Ce résultat se généralise pour une "puissance" de convolution d'une gaussienne pour la n -ième convolution :

$$(g_{\sigma_0} * (g_{\sigma_1})^{(n)})(x, y) = \frac{A_0 A_1^n}{2\pi(\sigma_0^2 + n\sigma_1^2)} \exp\left(-\frac{x^2 + y^2}{2(\sigma_0^2 + n\sigma_1^2)}\right) \quad (7)$$

Pour s'en convaincre, il suffit d'écrire l'intégrale puis développer et faire un changement de variable en sachant que :

$$\int_{-\infty}^{\infty} e^{-au^2} du = \sqrt{\frac{\pi}{a}}$$

Soit δz l'épaisseur de chaque coupe et n le numéro de la coupe considérée, on a alors un noyau gaussien continu de la forme suivante :

$$\begin{aligned} \forall x, y, z / z > 0, \quad h(x, y, n\delta z) &= (g_{\sigma_0} * g_{\delta\sigma})^{(n)}(x, y, n\delta z) \\ &= \frac{A_0(\delta A)^n}{2\pi(\sigma_0^2 + n(\delta\sigma)^2)} \exp\left(-\frac{x^2 + y^2}{2(\sigma_0^2 + n(\delta\sigma)^2)}\right) \\ &= \frac{A_0(\delta A')^z}{2\pi(\sigma_0^2 + z(\delta\sigma'))} \exp\left(-\frac{x^2 + y^2}{2(\sigma_0^2 + z(\delta\sigma'))}\right) \\ &= h(x, y, z) \end{aligned}$$

avec $z = n\delta z$, $\delta A' = \delta A^{\frac{1}{\delta z}}$ et $\delta\sigma' = \frac{\delta\sigma^2}{\delta z}$.

L'intensité des pixels ne devant pas augmenter, il est nécessaire de choisir les amplitudes δA et A_0 telles que :

$$\delta A \leq 1 \text{ et } A_0 \leq 1 \quad (8)$$

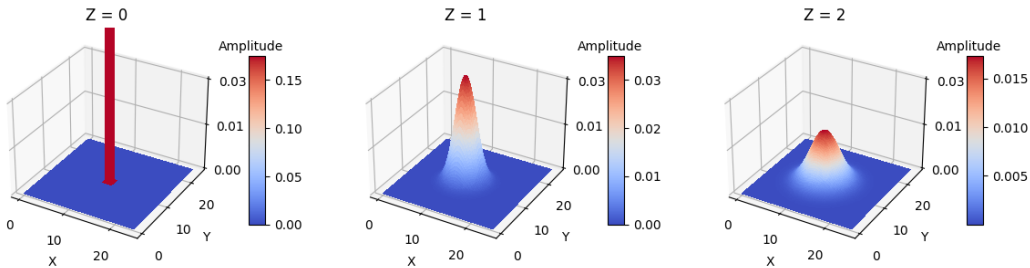


FIGURE 3 – Représentation du noyau gaussien $h(.,., z)$ avec $\sigma_0 = 0$ et $A_0 = 1$

4 Déconvolution

4.1 Critère de minimisation

En considérant une perturbation ϵ , le problème s'écrit :

$$I(x, y, z) = (O * h)(x, y, z) + \epsilon(x, y, z) \quad (9)$$

On suppose que ϵ est un bruit blanc gaussien car malgré la nature photonique des signaux, le bruit dans les images ne semble pas suivre une loi de poisson. Cette hypothèse est importante pour l'expression du critère de minimisation suivant [Lantéri et al., 2002].

$$\hat{O} = \arg \min_X \|I - (X * h)\|_2^2 + \gamma \|X\|_2^2 \quad (10)$$

avec $X \geq 0$

Le critère de minimisation proposé ici est composé d'un premier terme d'erreur suivis d'un terme de régularisation. Ces deux termes sont adaptés à l'hypothèse faites sur la nature du bruit [Lantéri et al., 2002]. La contrainte de positivité assure que la solution estimée soit cohérente avec la nature des données, les intensités lumineuses sont toujours positives.

4.2 Solution Algorithmique

Pour résoudre ce problème de déconvolution sous contrainte de positivité, on utilise une version relaxée d'un algorithme multiplicatif nommé *Image Space Reconstruction Algorithm (ISRA)*. Sa structure est détaillée dans [Lantéri et al., 2002]. Cet algorithme de descente itératif a été choisi car sa formulation relativement simple permet de l'implémenter soit-même et qu'il assure la positivité de la solution. Néanmoins, sa convergence requiert un nombre important d'itérations.

5 Simulations numériques

5.1 Problème direct

Grâce à la modélisation analytique précédente, il est possible de simuler le problème direct. On retrouve alors, sur les données simulées, la fluorescence sub-surfacique, ce qui permet de valider la cohérence entre la modélisation analytique et les observations réelles.

Lors de la simulation, une attention particulière doit être portée aux hypothèses de bords utilisées pour la convolution. En effet, dans le cas réel, l'échantillon peut ne pas être complètement entouré de paraffin, ce qui empêche l'utilisation de zéro-padding. De plus, l'acquisition peut être interrompue volontairement à tout instant. Pour ces raisons, la convolution sera faite en mode "Valid", sans dépassement du noyau. Il en résulte que pour un échantillon O de taille $L_x \times L_y \times D$ et un noyau de convolution h de taille $2k + 1 \times 2k + 1 \times d$, l'image obtenue est de taille $(P_x = L_x - 2k) \times (P_y = L_y - 2k) \times (M = D - d + 1)$.

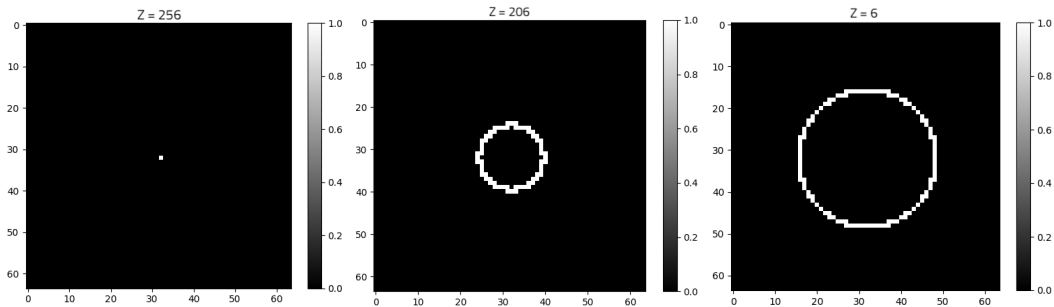


FIGURE 4 – *Objet réel simulé O_z*

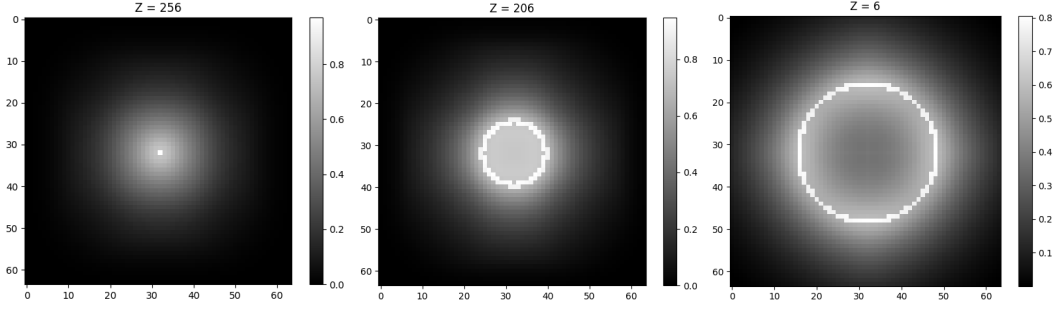


FIGURE 5 – Image simulée $I_z(x, y) = (O * h)(x, y, z)$

Cette simulation est implémentée dans le fichier *Krato-Simulation.py*. L'ensemble des paramètres de la simulation sont à choisir dans la partie *Main Script*, sauf les paramètres de la recherche de pas optimal (paramètres β et σ de la fonction *armijo*). Ces paramètres sont réglables dans la fonction *ISRA* à la ligne 144.

5.2 Problème Inverse

On peut également vérifier le fonctionnement de la méthode de déconvolution proposée et de l'implémentation numérique grâce à la simulation. Notamment il est possible d'étudier les performances obtenues selon le niveau de bruit. En utilisant les paramètres du noyau suivants : $\delta\sigma = 2.116$ et $\delta A = 0.989$ et avec un rapport signal sur bruit de 20 dB, on obtient des résultats qualitativement bons. On notera l'importance du paramètre de régularisation γ et le nombre d'itérations très élevé. Le niveau de bruit utilisé ici est néanmoins bien supérieur à celui observé dans les données réelles.

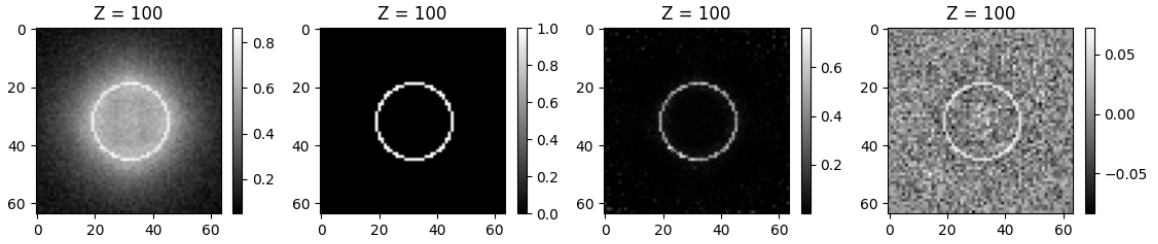


FIGURE 6 – $SNR = 20$ dB, $\gamma = 10^{-6}$ et 500 itérations

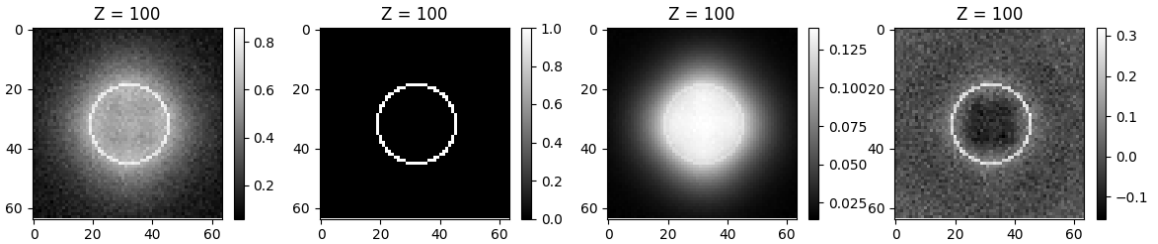


FIGURE 7 – $SNR = 20$ dB, $\gamma = 10^{-3}$ et 500 itérations

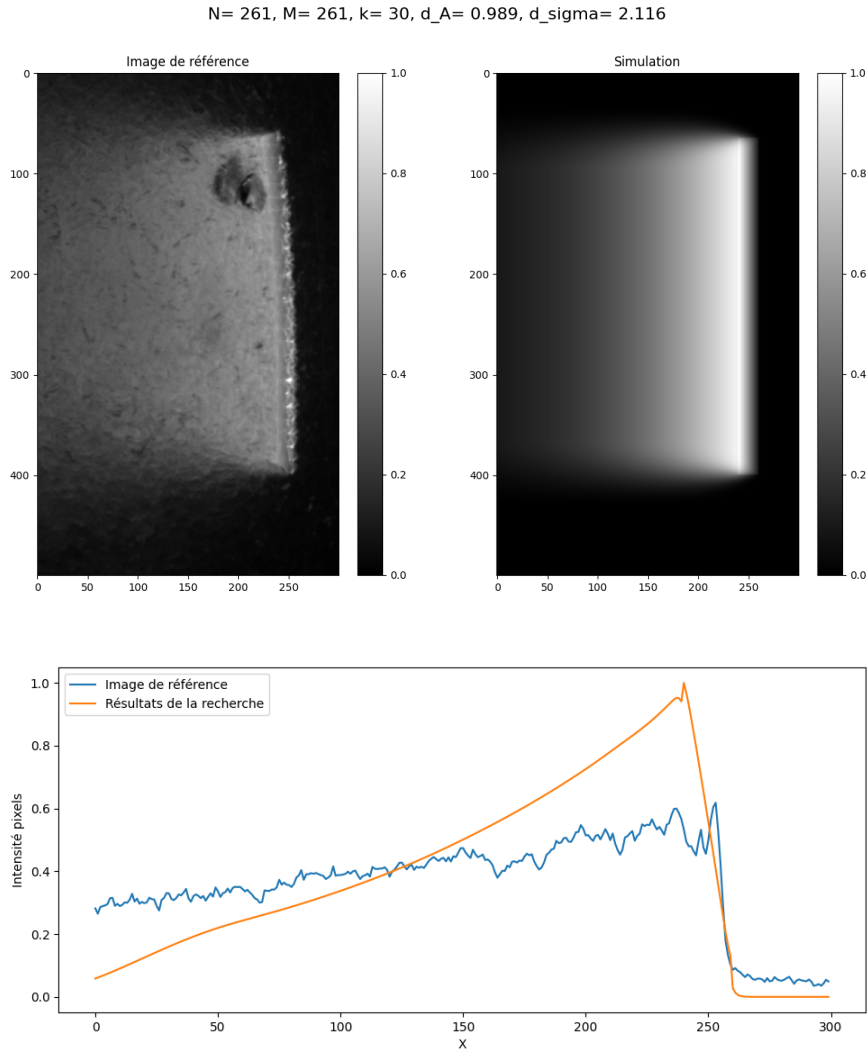
6 Application aux données réelles

6.1 Identification du noyau

Pour pouvoir appliquer la déconvolution aux données réelles il est primordial d'avoir une bonne estimation du noyau de convolution. Plusieurs méthodes existent pour déterminer h [Walsh et al., 2020]. Ayant accès à une image d'un objet connu, il est possible d'utiliser une simulation numérique pour

identifier les paramètres du noyau h . Ici, l'objet connu est un plan fixé dans un bloc de paraffin dont on ne connaît pas l'inclinaison. Ainsi on construit une modélisation à trois paramètres ($\delta\sigma$, δA et l'angle d'inclinaison du plan) et on cherche la combinaison permettant d'obtenir la simulation la plus fidèle à l'image de référence. La ressemblance est mesurée par la somme des différences élevée au carré entre le résultat de la simulation et la référence. La recherche de la meilleure combinaison est faite avec une méthode ne nécessitant pas le calcul analytique du gradient nommée méthode du Simplex de Nelder-Mead [Gao and Han, 2012].

Après recherche, on trouve les paramètres $\delta\sigma = 2.116$ et $\delta A = 0.989$, qui permettent d'obtenir un profil ayant globalement le même comportement que celui de la référence mais qui décroît plus rapidement. Cet écart peut s'expliquer par la présence de minimums locaux multiples ou l'incohérence du modèle gaussien de h .



L'identification est implémentée dans le fichier *Krato-Calibration.py*. Le chemin d'accès au fichier utilisé comme référence se trouve dans la fonction *load data*. Dans cette fonction l'image de référence est rognée, cela permet de supprimer un paramètre dans la modélisation. En effet, on sait que le plan incliné va d'un bord à l'autre de l'échantillon, si ce n'était pas le cas il faudrait aussi estimer la position du plan. Une implémentation scipy de la méthode de Nelder-Mead est utilisée (<https://docs.scipy.org/doc/scipy/reference/optimize.minimize-neldermead.html>).

6.2 Déconvolution d'un échantillon réel

Après avoir obtenu une estimation des paramètres du noyau de convolution, on peut appliquer la méthode de déconvolution à des données réelles fournies par MicroPICell.

Les résultats obtenus permettent de confirmer que le traitement numérique fonctionne, on supprime effectivement la fluorescence non souhaitée. Cependant, la solution obtenue est clairement sur-régularisée. En effet, l'amplitude du résidu est du même ordre de grandeur que le résultat, ce qui n'était pas vrai lors de la déconvolution sur des données simulées. Qualitativement, la netteté de l'image obtenue est fortement dégradée, c'est d'autant plus flagrant lorsqu'on remarque que les détails sont présents seulement dans le résidu. On arrive aux mêmes conclusions en étudiant la figure 9.

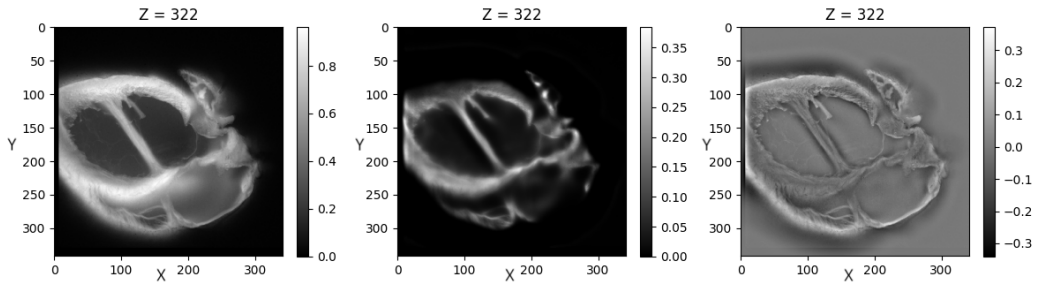


FIGURE 8 – Résultats avec $\gamma = 10^{-1}$

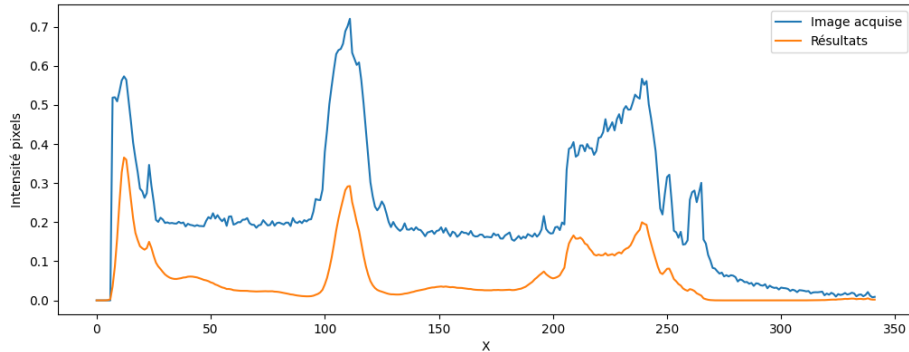


FIGURE 9 – Intensité des pixels pour $(y, z) = (150, 322)$

La déconvolution appliquée aux données réelles est implémentée dans le fichier *Krato-Real-Data.py*. Le chemin d'accès au fichier à traiter se trouve dans la fonction *load data*. Il y a également deux autres chemins d'accès, pour sauvegarder dans un fichier *.TIF* les résultats intermédiaires et le résultat final, ils se trouvent respectivement dans la fonction *ISRA* ligne 117 et dans la partie *Main Script* ligne 315. De la même manière que pour la simulation, l'ensemble des paramètres sont à choisir dans la partie *Main Script*, sauf les paramètres de la recherche de pas optimal (paramètres β et σ de la fonction *armijo*). Ces paramètres sont réglables dans la fonction *ISRA* à la ligne 105.

7 Améliorations et suite

- Implémentation d'une régularisation plus sophistiquée et réglage du γ
- Définition d'un critère d'arrêt pertinent pour l'algorithme de déconvolution
- Réglage des paramètres β et σ de la recherche de pas
- Utilisation d'un noyau de convolution plus réaliste issu d'un modèle optique
- Optimisation de l'implémentation numérique

Références

- [Gao and Han, 2012] Gao, F. and Han, L. (2012). Implementing the nelder-mead simplex algorithm with adaptive parameters. *Computational Optimization and Applications*, 51(1) :259–277.
- [Lantéri et al., 2002] Lantéri, H., Roche, M., and Aime, C. (2002). Penalized maximum likelihood image restoration with positivity constraints : multiplicative algorithms. *Inverse Problems*, 18 :1397 – 1419.
- [Mallocci et al., 2022] Mallocci, M., de Villemagne, P., Dorval, P., Feyeux, M., Blandin, S., Schmid, G., Hulin, P., and Gilloteaux, P. P. (2022). Label-free imaging of large samples : 3d rendering and morphological analysis within histological workflows using serial block face imaging. *bioRxiv*, <https://doi.org/10.1101/2022.05.03.488912>.
- [Walsh et al., 2020] Walsh, C., Holroyd, N., Shipley, R., and Walker-Samuel, S. (2020). Asymmetric point spread function estimation and deconvolution for serial-sectioning block-face imaging. In Papież, B. W., Namburete, A. I. L., Yaqub, M., and Noble, J. A., editors, *Medical Image Understanding and Analysis*, pages 235–249, Cham. Springer International Publishing.