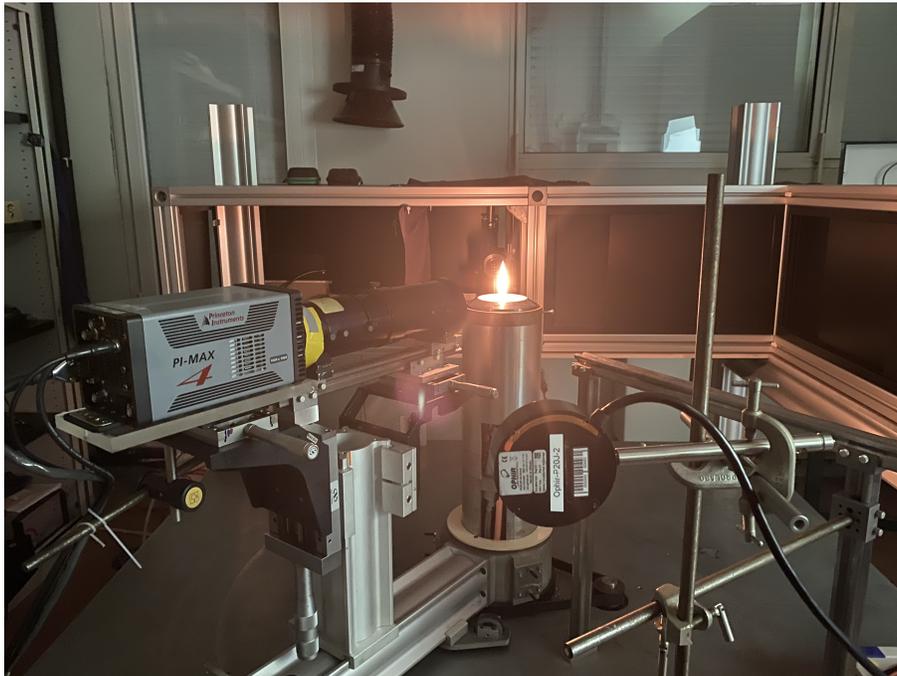


Projet de Physique P6 :
LA DECONVOLUTION D'ABEL



Justin BENOIST - Naya CASTAGNOS
Chloé MECHELINCK- Charlotte PIQUET
Guillaume RIAUD-SIMON

Enseignant-reponsable du projet : M. Jérôme YON
STPI/P6/2020 - 21

Date de remise du rapport : 15/06/2020

Référence du projet : STPI/P6/2020 21

Intitulé du projet : La déconvolution d'Abel

Type de projet : analyse de méthodes numériques

Objectifs du projet :

- travailler efficacement en groupe et à distance (répartition des tâches, organisation ...)
- s'adapter aux sources disponibles (documentation, anglais ...)
- utiliser un programme complet et existant : PyAbel
- analyser données, en tirer des conclusions

Mots-clés du projet : déconvolution, PyAbel, analyse

INSTITUT NATIONAL DES SCIENCES APPLIQUÉES DE ROUEN
DÉPARTEMENT SCIENCES ET TECHNIQUES POUR L'INGÉNIEUR
685 AVENUE DE L'UNIVERSITÉ BP 08 - 76801
SAINT-ÉTIENNE-DU-ROUVRAY
TÉL : +33 2 32 95 66 21 - FAX : +33 2 32 95 66 31

Remerciements

Nous tenons à remercier tout particulièrement Jérôme YON pour sa présence et son aide durant ce projet. En effet, il a pu nous apporter des pistes tout au long du projet, répondre à nos questions et nous a aidé à nous adapter pendant les derniers mois du projet, malgré les événements.

Nous souhaiterions aussi remercier José Moran, pour son aide à la prise en main du logiciel et pour nous avoir donné des axes de recherches.

Table des matières

1	Partie Historique	7
1.1	Jusqu'à 1825 ...	7
1.2	... à partir de 1825	7
2	Applications	9
2.1	Le principe de la déconvolution et comment est elle appliquée . . .	9
2.2	Nos applications : Soleil et données expérimentales	11
2.2.1	Le Soleil	11
2.2.2	Les données expérimentales	13
3	Partie Mathématiques	16
4	Partie Informatique	18
4.1	La convolution à la main	18
4.2	Description des méthodes de PyAbel	19
4.2.1	Présentation des familles	19
4.2.2	Famille 1	20
4.2.3	La famille 2 et 3	21
5	Analyse des résultats et programme	23
6	Sources	27
6.1	Sources Chloé	28
6.2	Source partie historique :	28
6.3	Sources Charlotte	28

Introduction

Dans le cadre de notre projet de P6 traitant de la déconvolution d'Abel, l'ensemble de l'équipe s'est intéressée aux différentes techniques de déconvolution et de convolution d'un signal donné. Nous nous sommes aussi penchés sur l'utilité, le fonctionnement, les applications et l'histoire de ce principe. Enfin, nous avons essayé de mieux comprendre les mathématiques qui se cachent derrière nos analyses de la déconvolution.

En effet, dès le départ, l'objectif de ce projet était d'être en mesure d'analyser le signal que l'enseignant allait nous transmettre. Il devait l'obtenir dans le cadre de ses recherches aux laboratoires du Coria avec son doctorant. Notre but étant de lui faire gagner du temps en lui permettant d'analyser les résultats expérimentaux obtenus et d'apporter un complément d'informations précis, la documentation de ce programme étant très complet. Nous avons ainsi cherché le moyen le plus précis et efficace d'analyser les signaux qui allaient nous être donnés.

Organisation du travail

Le but du projet étant de comprendre la déconvolution d'Abel et de la mettre en œuvre afin de s'en servir pour analyser les courbes obtenues lors des expériences de Jérôme Yon et de son doctorant. Pour cela nous nous sommes tout d'abord renseigné sur ce qu'était la déconvolution et avons également cherché à comprendre le fonctionnement de PyAbel et des différentes techniques de déconvolution qui y sont répertoriées. A la suite de cela, nous avons décidé de nous répartir le travail en 4 parties :

- une partie historique
- une partie mathématiques
- une partie informatique
- une partie analyse des résultats et programme

Nous nous sommes par la suite distribuer le travail afin que tous les membres du groupe travail dans un domaine qui leur convient.

Cependant, il nous semble juste de préciser le contexte dans lequel nous avons évolué. Effectivement après 4 semaines de projet, le covid 19 nous a empêché de nous réunir et nous avons dû travailler chacun de notre côté. Cela ne nous a pas empêcher de nous rassembler toutes les semaines, mais il vrai que le partage de l'information et la quantité de ressources à notre disposition s'en est trouvé touché. Surtout, nous n'avons pas été en mesure de recevoir la courbe expérimentale attendue et avons donc dû finalement se contenter de déconvoluer des courbes théoriques. Nous nous tenions également toujours informé des avancées de chacun avant et pendant le confinement lors des séances avec M.Yon. Ainsi nous étions en capacité d'apporter des modifications et de redéfinir régulièrement le travail de chacun chaque semaine.

Justin Benoist	Recherches, partie informatique, codage d'une méthode de convolution, analyse des résultats et programme, prise en main du logiciel PyAbel
Chloé Mechelinck	Recherches, compréhension et description des méthodes de PyAbel, partie mathématique, mise en page et réalisation du rapport
Charlotte Piquet	Recherches, compréhension et description des méthodes de PyAbel, analyse des résultats, conclusion
Naya Castagnos	Recherches, partie historique et recherche au niveau des applications de la déconvolution
Guillaume Riaud-Simon	Recherches, partie historique et application de la déconvolution, introduction et répartition du travail

Chapitre 1

Partie Historique

1.1 Jusqu'à 1825 ...

Niels Abel est un mathématicien d'origine norvégienne qui est né le 5 août 1802. Ce futur génie provient d'un milieu marqué par la pauvreté. En effet, bien que son père fut un homme politique norvégien de grande envergure, il tomba en disgrâce lors de la fin de sa vie. Ainsi lors de sa mort en 1820, c'est à Abel de s'occuper et de parvenir aux besoins de sa famille. Abel qui était jusqu'à lors un élève assez moyen, va faire la rencontre de Bernt Michael Holmboe. Ce dernier lui donne le goût des mathématiques et décèle en Abel un véritable génie. Il le pousse à poursuivre ses études et arrive à lui obtenir une bourse au sein de la meilleure université norvégienne. Durant cette période, Abel apporte de nombreuses contributions, il s'agit d'une phase prolifique pour lui. Il démontre notamment que l'équation quelconque de degré cinq n'est pas résoluble à partir de combinaisons de racines des coefficients.

1.2 ... à partir de 1825

A partir de 1825, Abel démarre un voyage en Europe. Il ira dans tous les centres intellectuels de l'époque comme Paris, Berlin, Göttingen... Il y fera la rencontre de ses contemporains mathématiciens comme Gauss, Cauchy. A Paris il veut publier ses mémoires et les confie à Cauchy mais ce dernier les perd, de plus elles seront mésestimées par Gauss. Après son doctorat, Abel ne parvient pas à trouver un poste, ses conditions de vie sont de plus en plus précaires et sa santé se fait fragile : il est atteint de la tuberculose. Malgré des déplacements à Paris et à Berlin, ses travaux ne sont toujours pas perçus à leur juste valeur.

Dans ses dernières semaines, il n'a plus assez de force pour quitter son lit. Il décède le 5 avril 1829, à même pas 27 ans, alors qu'un ami venait juste de lui trouver un poste à Berlin. C'est Jacobi qui comprendra tout le génie de

ce jeune mathématicien. Abel avait notamment démontré, à l'âge de 19 ans, l'impossibilité de résoudre par radicaux les équations algébriques de degré 5, ce que son contemporain Galois généralisera à tout $n > 4$. A titre posthume, Abel recevra en 1830 le grand prix de Mathématiques de l'Institut de France.

Chapitre 2

Applications

2.1 Le principe de la déconvolution et comment est elle appliquée

La déconvolution d'Abel est un outil mathématique, qui trouve son utilité dans de nombreux domaines scientifiques et techniques. Il est vrai que la transformée d'Abel est un phénomène complexe, c'est pour cela que son application est nécessaire dans des spécialités à la pointe de la recherche.

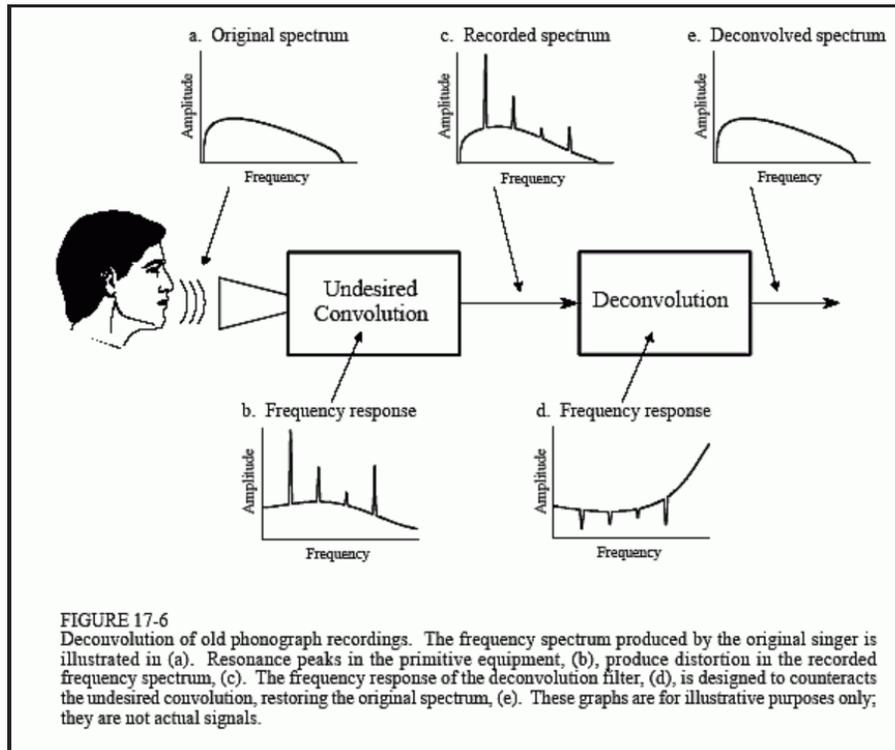
On retrouve ainsi la déconvolution en microscopie optique notamment dans l'analyse d'image/photographie, en spectroscopie d'absorption de flammes ou panaches cylindriques mais aussi appliquée aux signaux codés dans le domaine fréquentiel . Enfin au cours des dernières années la transformé inverse d'Abel est devenue primordiale pour l'analyse des données en imagerie par ions photo fragmenté et en imagerie photoélectronique.

La déconvolution est donc utile en analyse d'image. Effectivement, il est possible de louper une photo à cause d'un manque de lumière. Ainsi la photo est floue, on dit qu'elle est bougée. Il est possible d'enlever ce bougé en déconvoluant notre image. En effet, le bougé n'est dû qu'à la convolution d'une image nette. Il est donc possible grâce aux mathématiques de remplacer les données de l'image par des fréquences. Ainsi à la suite de nombreux calculs il est possible d'obtenir une image moins floue. Cependant, il faut faire très attention à notre estimation du mouvement car un léger manque de précision peut avoir de graves conséquences sur notre image finale.



FIGURE 2.1 – Déconvolution + traitement de l'image

On retrouve également l'utilisation de la transformée d'Abel dans le domaine de la combustion. Lors de l'étude de la propagation d'une flamme de prémélange, le paramètre le plus important est la chaleur de cette dernière. Cependant, sa est est complexe. Il existe donc une technique appelée chimiluminescence qui permet d'obtenir des informations liées à la chaleur. Pour cette méthode, l'information tomographique est obtenue grâce à une inversion du procédé d'Abel. La déconvolution est donc également utilisée pour décoder des signaux dans le domaine fréquentiel qui ont été convolués. Un exemple connu est la restauration d'anciens enregistrements du célèbre chanteur d'opéra, Enrico Caruso (1873-1921). Ces enregistrements ont été réalisés avec un vieil équipement que les appareils d'aujourd'hui ont du mal à retransmettre. Le problème est la résonance du long pavillon d'enregistrement tubulaire utilisé pour recueillir le son. Chaque fois que le chanteur frappe une de ces fréquences de résonance, le volume de l'enregistrement augmente brusquement. La déconvolution numérique a réussi à améliorer la qualité subjective de ces enregistrements.



La figure 17-6 montre l'approche générale de ce décodage . Le spectre de fréquences du signal audio d'origine est illustré en (a). La figure (b) montre la réponse en fréquence de l'équipement d'enregistrement, une courbe relativement lisse à l'exception de plusieurs pics de résonance nets. Le spectre du signal enregistré, montré en (c), est égal au vrai spectre, (a), multiplié par la réponse en fréquence inégale, (b). Le but de la déconvolution est de contrer la convolution indésirable. En d'autres termes, la réponse en fréquence du filtre de déconvolution, (d), doit être l'inverse de (b). C'est-à-dire que chaque pic en (b) est annulé par un creux correspondant en (d). Si ce filtre était parfaitement conçu, le signal résultant aurait un spectre (e) identique à celui de l'original.

2.2 Nos applications : Soleil et données expérimentales

2.2.1 Le Soleil

Nous avons souhaité découvrir une autre application de la déconvolution d'Abel. Nous nous sommes donc intéressés à un objet diffusant de la lumière et possédant une symétrie sphérique tel que le Soleil. Pour créer une telle image, nous avons utilisé le logiciel Blender. Ainsi, la diffusion est parfaite.

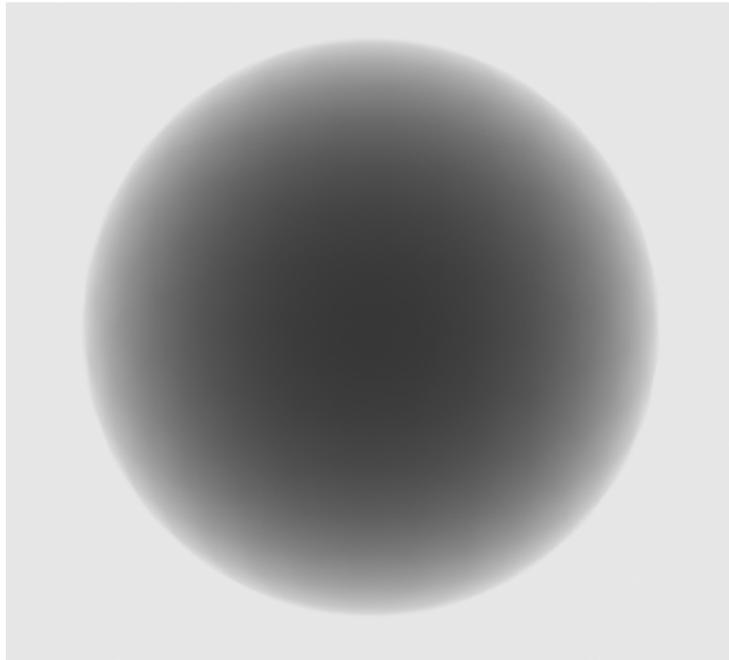


FIGURE 2.2 – Soleil absorbant sous Blender

A partir de l'image de notre "Soleil absorbant", nous avons procédé à une déconvolution afin d'étudier le profil. Voici ce que nous obtenons :

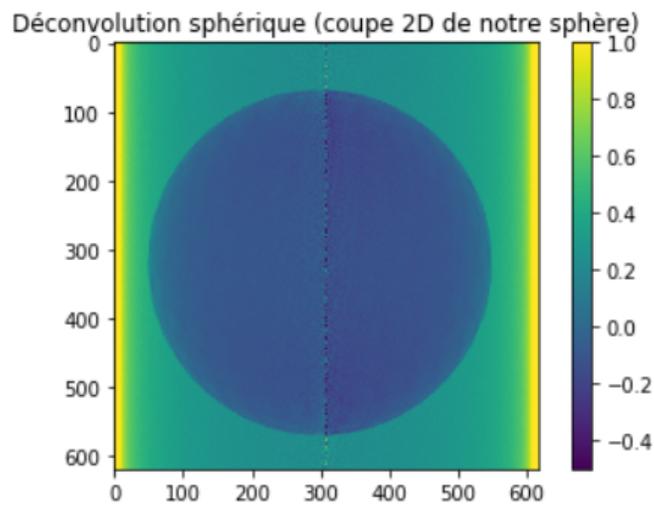


FIGURE 2.3 – Signal observé

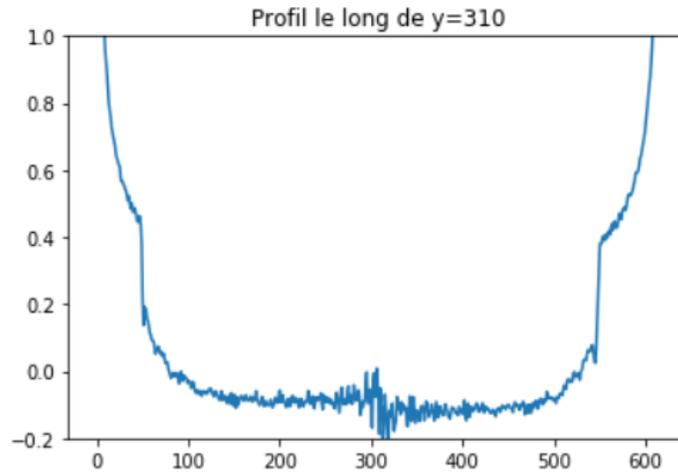


FIGURE 2.4 – Signal observé

On peut voir que l’absorption forme bien un plateau constant le long du rayon de notre “soleil” entre 50 et 550 environ. Ce qui est normal puisque l’absorption de notre soleil ne dépend est uniforme. Remarquons que le bruit semble élevé au centre de notre image.

2.2.2 Les données expérimentales

Maintenant que nous avons comparé les différentes méthodes de déconvolution offertes par PyAbel, nous pouvons appliquer la déconvolution d’Abel sur des cas concrets. Nous allons donc traiter des mesures faites au CORIA, situé à côté de l’INSA. Le but est d’étudier la répartition des particules de suie produites par une flamme à l’aide d’une nappe laser qui la traverse. Les mesures en sortie forment donc une projection car la nappe laser traverse toute la flamme avant d’être traitée (voir ci-dessous).

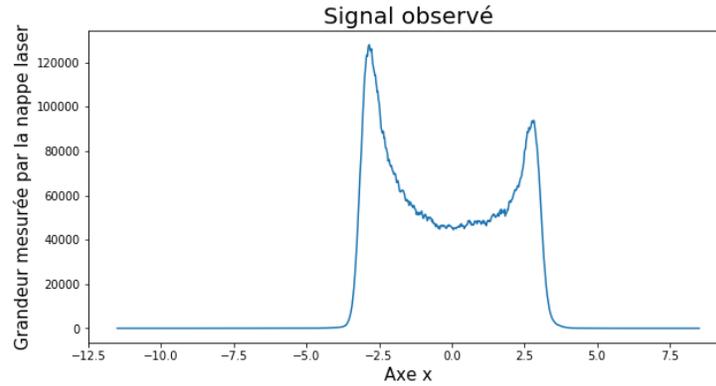


FIGURE 2.5 – **Signal observé**

Nous utiliserons la méthode “direct” pour cette déconvolution, cette dernière ayant déjà montré de bons résultats. La déconvolution est faite sur une seule moitié de l’image puisque cette dernière est censé être symétrique.

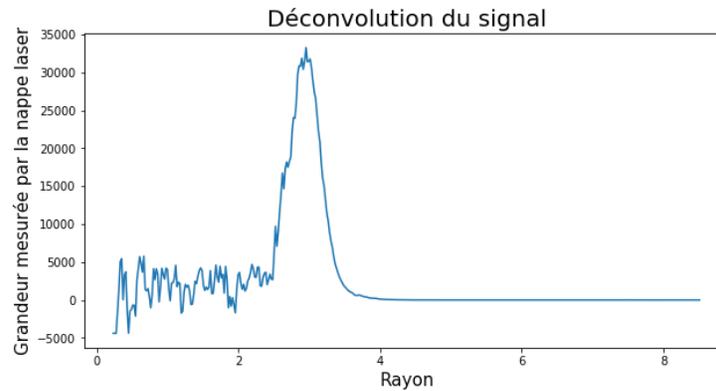


FIGURE 2.6 – **Déconvolution du signal**

Même si le signal semble un peu bruité au début, on peut remarquer un pic de la grandeur mesurée. Nous pouvons désormais vérifier notre déconvolution en la convoluant et comparant la convoluée avec le signal de base.

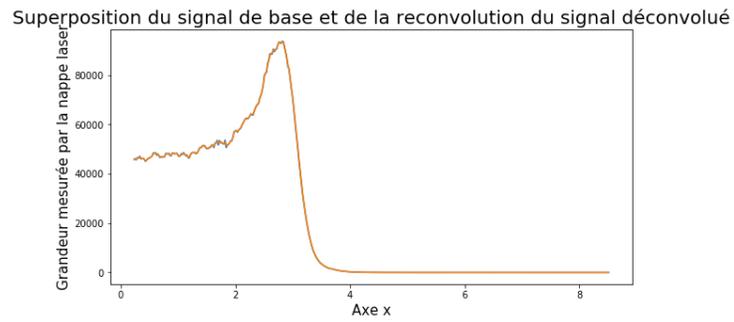


FIGURE 2.7 – **Superposition du signal de base et de la reconvolution du signal déconvolué**

Voilà qui montre que le bruit de la déconvolution vient du bruit causé par la mesure et non de la méthode de déconvolution.

Chapitre 3

Partie Mathématiques

Le but de ce projet était d'obtenir une fonction dite de répartition des suies à l'intérieur d'une flamme. En effet, une nappe de fuseau laser est projetée à travers la flamme et est déviée lorsqu'elle rencontre des suies. Les rayonnements sont en suite capté et forme la fonction $f(r)$. En effet, par expérimentation une valeur pour chaque y est déterminée.

r est le rayon de la flamme

x et y sont les deux coordonnées de l'espace

le schéma ci dessous explicite la manière dont l'on obtient la courbe. (à noter que la courbe dans notre cas est inversée à celle du schéma).

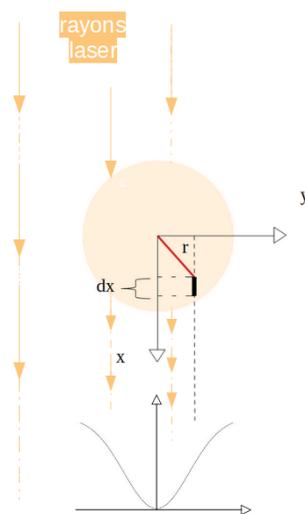


FIGURE 3.1 – scéma explicitant l'obtention des données

$$I(y) = 2 \int_{x=0}^{+\infty} f(r) dx$$

$$I(x + dx) = I(x) - k_{ext}(r) dx I(x)$$

$$\Rightarrow d(I) = -k_{ext}(r) dx I(x)$$

$$\begin{aligned} \Rightarrow \int_{I_0}^{I(y)} \frac{dI}{I} &= \int_{-\infty}^{+\infty} -k_{ext}(r) dx \\ &= 2 \int_{x=0}^{+\infty} -k_{ext}(r) dx \end{aligned}$$

car la fonction I est symétrique (selon l'axe y voir schéma)

$$\text{or } r^2 = x^2 + y^2 \iff x^2 = r^2 - y^2$$

$$\Rightarrow dx = \frac{r dr}{x}$$

$$dx = \frac{r dr}{\sqrt{r^2 - y^2}}$$

$$\begin{aligned} \text{donc } \int_{I_0}^{I(y)} \frac{dI}{I} &= 2 \int_y^{+\infty} \frac{-k_{ext}(r) r dr}{\sqrt{r^2 - y^2}} \\ \Rightarrow \int_{I_0}^{I(y)} d(\ln(I)) &= 2 \int_y^{+\infty} \frac{-k_{ext}(r) r}{\sqrt{r^2 - y^2}} dr \\ \Rightarrow \text{Log}\left(\frac{I(y)}{I_0}\right) &= -2 \underbrace{\int_y^{+\infty} \frac{-k_{ext}(r) r}{\sqrt{r^2 - y^2}} dr}_{f(r, y, k)} \end{aligned}$$

$$I(y) = I_0 \times e^{f(r, y, k)}$$

Chapitre 4

Partie Informatique

4.1 La convolution à la main

Avant d'étudier les différentes méthodes de déconvolution, nous pouvons nous pencher sur la convolution. En effet, le processus de convolution est plus simple que son opposé et implique donc moins d'erreurs. Il suffit de calculer l'intégrale suivante :

$$F(y) = 2 \int_y^{+\infty} \frac{f(r)r}{\sqrt{r^2 - y^2}} dr$$

Si on code cette formule sous Python et que nous l'appliquons à ce signal par exemple :

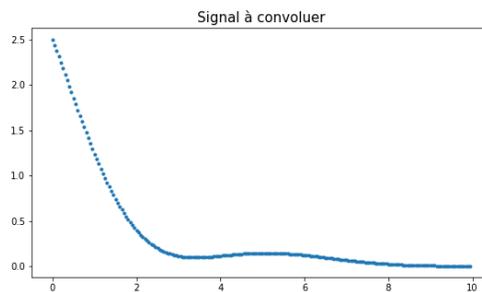


FIGURE 4.1 – **Signal à convoluer**

Application :

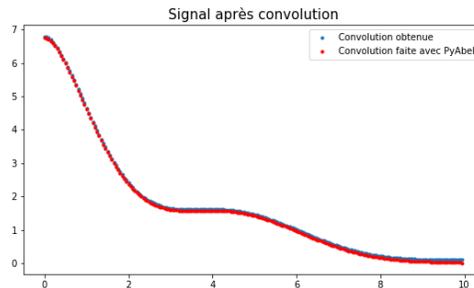


FIGURE 4.2 – Signal après convolution

On peut voir que la courbe obtenue directement avec la formule de convolution coïncide bien avec la courbe obtenue grâce à PyAbel (librairie Python, voir suite). Il est important que cette méthode de convolution dépend fortement de la précision du signal original. Un nombre de points faible donnera comme résultat une convolution médiocre.

La méthode d’intégration a aussi son poids dans la balance. Une méthode simple comme celle des rectangles donnera également une moins bonne convolution. Pour la méthode de convolution que nous avons réalisée, nous utilisons donc la fonction quad de la librairie SciPy, ce qui permet de réduire l’erreur et le temps de calcul.

4.2 Description des méthodes de PyAbel

PyAbel est une librairie du langage Python qui fournit des implémentations efficaces de plusieurs algorithmes permettant d’effectuer des transformations d’Abel. Celles-ci peuvent s’exécuter dans deux sens, inverses l’un de l’autre : le sens dit “direct” et celui dit “inverse”.

4.2.1 Présentation des familles

Présentation des familles PyAbel offre plusieurs méthodes d’approximations c’est-à-dire plusieurs implémentations d’algorithmes à la transformation exacte d’Abel. Ainsi, toutes les méthodes devraient produire des résultats identiques, mais selon le niveau et le type de bruit que l’on trouve dans l’image (c’est-à-dire le signal que l’on cherche à convoluer ou déconvoluer), certaines sont plus performantes que d’autres. Les différentes méthodes proposées par PyAbel peuvent se découper en trois familles :

Famille 1	Famille 2	Famille 3
Basex	Two_point	Onion peeling
Hansenlaw	three_point	Onion bordas
Direct		
+ linbasex		

Nous avons ici détaillé les différentes méthodes sans prendre en compte le fait que nous allons plus utiliser certaines méthodes que d'autres. En effet, certaines possèdent des caractéristiques propres correspondant mieux à notre étude.

4.2.2 Famille 1

Méthode BASEX (BASis SEt EXpansion)

L'algorithme découpe les données en plusieurs fonctions dont les transformées sont déjà connues analytiquement par l'algorithme. Ces fonctions connues sont des fonctions gaussiennes. Ainsi, les fonctions par défaut se concentrent sur chaque pixel, sans se préoccuper de la forme globale de l'image.

Cette méthode est l'une des plus utilisées en imagerie photoélectronique/photoion. Le processus d'évaluation de ces fonctions présente de nombreux calculs, et le processus peut prendre plusieurs secondes voire quelques minutes pour des images plus grandes (plus grandes que 1000x1000 pixels). Cependant, une fois calculés, ces ensembles de base peuvent être réutilisés, et sont donc stockés en mémoire et chargés rapidement pour une utilisation future. La transformation se poursuit alors très rapidement, puisque chaque transformée inverse d'Abel d'une image est une simple multiplication de matrice. Cette méthode peut donc être très utile si on doit étudier plusieurs images semblables car elle permet d'économiser du temps. Pour les projections sans bruit, BASEX reconstruit une image exacte, évitant ainsi de devoir introduire des procédures d'interpolation qui pourraient être à l'origine d'erreurs. En outre, la procédure de reconstruction ne contribue pas au bruit dans l'image reconstruite ; le bruit n'apparaît dans l'image que lorsqu'il existe dans la projection. Cette méthode n'ajoute donc pas de bruit inexistant et donc pas d'informations fausses.

Méthode Hansen-Law

Il s'agit ici d'un algorithme récursif qui ne s'appuie que sur une seule hypothèse : celle de la symétrie cylindrique. Cette méthode permet d'obtenir une transformation extrêmement rapide avec un faible bruit. Elle effectue un changement de variable pour modéliser la transformée sous la forme d'un ensemble d'équations différentielles linéaires avec le gradient de l'image de projection $g'(R)$ (pour la transformée inverse) ou avec l'image source $f(r)$ (pour la transformée directe) où :

$$F(R) = 2 \int_y^{+\infty} \frac{f(r)r}{\sqrt{r^2 - R^2}} dr$$

$$f(r) = \frac{-1}{\pi} \int_r^{+\infty} \frac{g'(R)dR}{\sqrt{R^2 - r^2}}$$

Cet algorithme intègre le long de chaque colonne de l'image (de l'extérieur vers le centre) et fonctionne très bien avec les grandes images. Pour le bon fonctionnement de cette méthode, un échantillonnage fin de données est nécessaire.

Méthode direct

Comme son nom l'indique, cette méthode passe par une intégration directe de l'intégrale de la transformation d'Abel. Elle ne fait aucune hypothèse particulière sur les données (à l'exception de celle concernant la symétrie cylindrique), mais elle nécessite généralement un échantillonnage fin pour une bonne approximation. De telles méthodes sont généralement inefficaces, mais celle-ci est implémentée en Cython - il s'agit d'une bibliothèque Python permettant de permuter avec du code écrit en C. Ainsi cette méthode Directe se hisse au même niveau que les autres.

Cette méthode fonctionne très bien pour les transformations dans le sens direct. Cependant elle est moins recommandée pour les transformations dans le sens inverse, à moins que la fonction en question soit dépourvue de bruit et bien échantillonnée, comme vu précédemment.

4.2.3 La famille 2 et 3

La méthode Onion peeling et la méthode two_points sont similaires, elles sont en fait deux implémentations d'une même méthode (celle de la méthode de la pelure d'oignon). Il en va de même pour les méthodes onion bordas et three_points lorsque la méthode onion bordas est appliquée à une seule dimension. Elles sont très semblables. Nous nous sommes donc centrés sur les méthodes onion peeling et onion bordas dans cette description, les deux couples de méthodes fonctionnant sur les mêmes principes.

Méthode onion peeling

Cette méthode est traduite en français comme la méthode pelure d'oignon. En effet, on peut associer cette méthode à l'image d'un oignon qu'on pellerait au fur et à mesure de ses différentes couches de peau. Cette méthode est simple et très efficace par ordinateur. Elle est également facile à se représenter mentalement. En revanche, si elle est faite à la main, cette méthode est très longue car il s'agit de la répétition d'opérations simples un très grand nombre de fois (une couche est simple à traiter et il existe un très grand nombre de couches élémentaires). Cette méthode (comme la méthode two_points) lisse moins les résultats que les autres méthodes : les fonctions en sortie peuvent être moins régulières mais cela peut être utile lorsque les fonctions étudiées ne le sont pas d'origine car on garde plus d'informations (et donc un résultat plus exact).

Méthode onion bordas

Il s'agit d'une autre implémentation de la méthode onion peeling. Cet algorithme calcule les effets des particules, à une énergie cinétique donnée et d'un signal dans un pixel donné (le signal étant découpé en un tableau composé de colonnes et lignes infinitésimales). Ce signal est ensuite soustrait du pixel projeté (expérimental) et également ajouté au pixel d'image rétro-projeté. La procédure est répétée jusqu'à ce que le centre de l'image soit atteint. L'ensemble de la procédure est effectué pour chaque ligne de pixels de l'image.

Chapitre 5

Analyse des résultats et programme

PyAbel propose 7 méthodes différentes qui effectuent des transformées et des transformées inverses d'Abel. Nous avons réalisé un programme permettant de les comparer et ainsi de déterminer les plus efficaces.

Pour notre projet, nous nous intéressons à un signal convolué bruité (c'est ce qui se rapproche le plus des signaux que nous pouvons obtenir expérimentalement).

C'est pourquoi nous commençons par créer un signal quelconque que nous convoluons et auquel nous ajoutons du bruit de façon aléatoire. Ensuite, une fois le signal convolué bruité créé, nous le déconvoluons en utilisant chacune des sept méthodes. On constate pour chacune des sept déconvolutions obtenues que plus on s'éloigne du centre de l'image, mieux la déconvolution correspond au signal initial et ce, pour tous les signaux différents que nous avons testés.

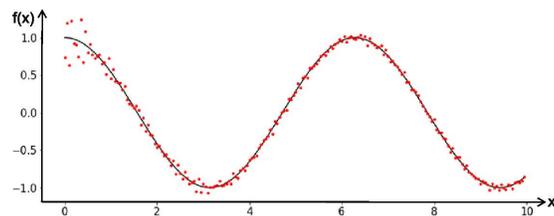


FIGURE 5.1 – Signal non convolué et signal déconvolué (Basex)

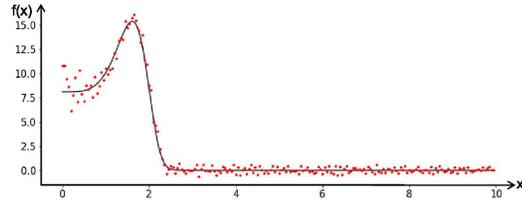


FIGURE 5.2 – Un autre signal non convolué et déconvolué (Hansen Law)

Comparons, à présent, l'efficacité de ces méthodes. On calcule l'écart-type entre le signal initial et la déconvolution pour chaque méthode (nous nous intéressons à la déconvolution donc les convolutions ont toutes été effectuées par la même méthode : nous avons choisi Basex. Puis nous avons ajouté du bruit). On montre grâce à un diagramme en bâtons que les trois méthodes les plus fiables sont Direct, Three-Point et Onion-Bordas.

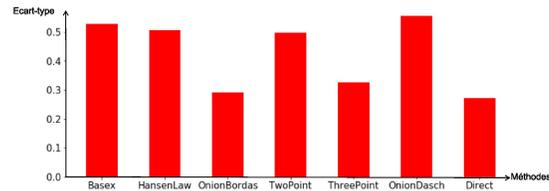


FIGURE 5.3 – Diagramme en bâtons représentant l'écart-type de la déconvolution pour chaque méthode

Il est important de prendre en compte la valeur maximale du bruit par rapport au maximum du signal convolué. Nous représentons alors l'écart-type en fonction du bruit ajouté. Les résultats sont obtenus ici avec la méthode Hansen Law mais on retrouve la même chose avec les autres méthodes, à savoir que l'écart-type augmente de façon linéaire avec la quantité de bruit ajouté.

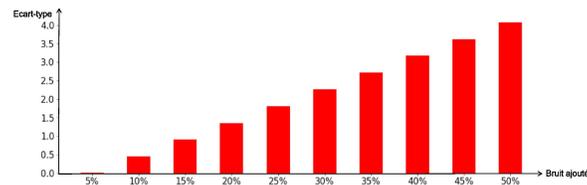


FIGURE 5.4 – Diagramme en bâtons représentant l'écart-type de la déconvolution pour la méthode Hansen Law en fonction du bruit ajouté

Il est désormais possible de rassembler l'ensemble de ces données pour représenter, pour chaque méthode, l'écart-type en fonction du bruit ajouté. Comme précédemment, les méthodes Direct, Three-Point et Onion-Bordas se démarquent par leur fiabilité.

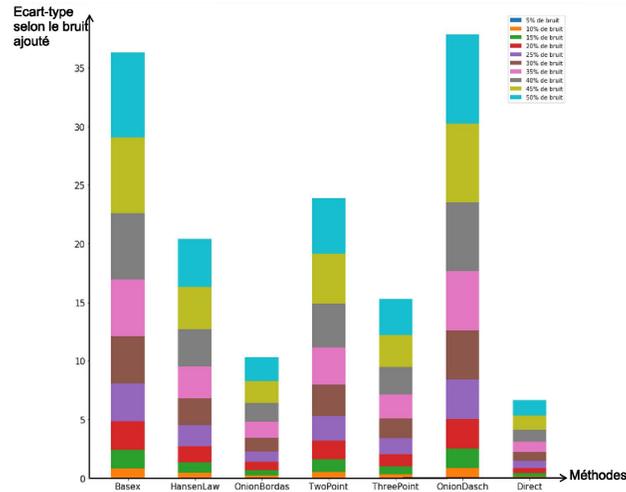


FIGURE 5.5 – Diagramme en bâtons représentant l'écart-type selon le bruit ajouté pour la déconvolution de chacune des méthodes

Enfin, nous avons utilisé un dernier critère pour comparer les méthodes : celui de la durée d'exécution. Nous avons donc calculé ce paramètre pour les déconvolutions d'un même signal, réalisées avec chaque méthode. La plus chronophage est Onion-Bordas, suivie par Basex puis Direct.

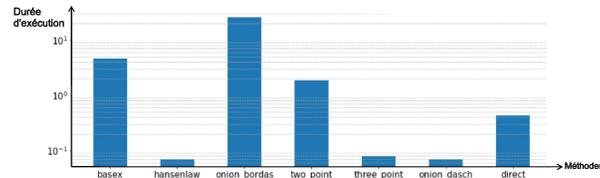


FIGURE 5.6 – Diagramme en bâtons représentant la durée d'exécution de la déconvolution pour chaque méthode

Conclusion et perspectives

Nous pouvons à présent clore ce projet dans lequel nous nous sommes investis pendant notre quatrième semestre à l'INSA. Il nous a permis d'aborder les sciences dans un cadre expérimental, de recherche. Cette approche est bien différente de l'enseignement théorique que nous avons suivi jusque-là. En effet, notre projet sur la déconvolution d'Abel était directement relié au réel projet de recherche de M. Yon, notre enseignant responsable.

L'étude que nous avons menée était à la fois historique, mathématique, et principalement informatique. Ce dernier domaine a modifié et approfondi la vision que nous avons pu en avoir au cours des trois premiers semestres d'enseignement à l'INSA. Effectivement, pour la plupart, nous ne connaissions que le Pascal tandis que pour notre projet nous avons utilisé le langage Python. La maîtrise de cet outil, plus répandu dans le monde professionnel, nous sera forcément utile plus tard.

Enfin, un autre atout non-négligeable de ce projet est le travail en groupe. Savoir se répartir les tâches, savoir travailler en temps et en heure pour ne pas bloquer l'avancée des autres membres, se mettre d'accord sur le choix de l'approche du sujet... Autant de compétences qui sauront faire de nous des ingénieurs. Par ailleurs, la difficulté en plus que présentait ce projet était de travailler à cinq - dans notre cas - sur un unique sujet alors que nous sommes tous destinés, à priori, à choisir des départements différents. Il a donc fallu s'investir avec des connaissances différentes, sans étudier dans un autre cadre le domaine en question.

26/05/20

Lien du résumé des différentes methode

pyabel https://pyabel.readthedocs.io/en/latest/transform_methods.html

12/04/20

comparaison des méthodes

<https://arxiv.org/pdf/1902.09007.pdf>

04/04/20

6.1 Sources Chloé

https://pyabel.readthedocs.io/en/latest/transform_methods/direct.html

04/03/20

https://github.com/PyAbel/PyAbel/blob/master/doc/transform_methods.rst#id9

25/03/20

https://pyabel.readthedocs.io/en/latest/example_onion_bordas.html

08/05/20

https://pyabel.readthedocs.io/en/latest/transform_methods/direct.html

23/04/20

6.2 Source partie historique :

Biographie de Niels Abel

<http://www.bibmath.net/bios/index.php?action=affiche&quoi=abel>

07/05/2020

Abel Niels Henrik Abel

<https://www.universalis.fr/encyclopedie/niels-henrik-abel/>

07/05/2020

Niels Henrik Abel [https](https://fr.wikipedia.org/wiki/Niels_Henrik_Abel)

[://fr.wikipedia.org/wiki/Niels_Henrik_Abel](https://fr.wikipedia.org/wiki/Niels_Henrik_Abel)

07/05/2020

6.3 Sources Charlotte

https://pyabel.readthedocs.io/en/latest/readme_link.html

02/2020

https://pyabel.readthedocs.io/en/latest/transform_methods/hansenlaw.html
10/03/2020

<https://www.osapublishing.org/ao/abstract.cfm?uri=ao-31-8-1146>
10/03/2020

https://pyabel.readthedocs.io/en/latest/example_dasch_methods.html
22/03/2020

Annexe (non obligatoire)

```
import random

def convolutionBasex (I):
    return(abel.basex.basex_transform(I,dr=0.1 , direction='forward'))

J1=convolutionBasex(I)

for i in range(200):
    J1[i]=J1[i]+random.uniform(-0.1,0.1)

def deconvolutionBasex (J1):
    return(abel.basex.basex_transform(J1,dr=0.1, direction='inverse'))

K1=deconvolutionBasex(J1)

plt.figure(1,figsize=(13,5))
plt.subplot()
plt.title("Signal non convolué et signal déconvolué (basex)")
plt.plot(x,I,'-',color='black',label='4')
plt.plot(x,K1,'.',color='red',label='6')

plt.figure(2,figsize=(10,3))
plt.subplot()
plt.title("Signal bruité convolué (basex)")
plt.plot(x,J1,'.',color='green',label='5')

plt.plot(x,I,'-',color='black',label="10")
plt.plot(x,deconvolutionHansenLaw(J1),'.',color='red',label="12")

fig = plt.figure(figsize=(13,5))

abcisse = ['ecartBasex','ecartHansenLaw','ecartOnionBordas','ecartTwoPoint','ecartTwoPoint']
height11 = [calculEcart(K1,I),calculEcart(K2,I),calculEcart(K3,I),calculEcart(K4,I)]
width = 0.5

plt.bar(abcisse, height11, width, color='r' )

plt.show()
```

```

height10=np.zeros(10)

def signal2bruit(bruit):
    J2=np.zeros(200)
    for i in range(200):
        J2[i]=random.uniform(-bruit,bruit)
    return(J2)

J1= signal2(x,a,b,c)
bruits = ['bruit5%', 'bruit10%', 'bruit15%', 'bruit20%', 'bruit25%']
ecartDirect']
(K6,I),calculEcart(K7,I)]for i in range(10):
    height10[i]=calculEcart(deconvolutionHansenLaw(i*J2+J1,bruits[i]))

plt.figure(figsize=(15,5))
plt.bar(bruits, height10, width, color='r' )
plt.show()

```

```

M=np.arange(0.0,10.0,0.005)
M1=signal2(M,a,b,c)
deconvolutions = {
    "basex": deconvolutionBasex,
    "hansenlaw": deconvolutionHansenLaw,
    "onion_bordas": deconvolutionOnionBordas,
    "two_point" : deconvolutionTwoPoint,
    "three_point": deconvolutionThreePoint,
    "onion_dasch": deconvolutionOnionDasch,
    "direct": deconvolutionDirect,
}
t=np.zeros(7)
for i,methode in enumerate (deconvolutions.keys()):
    J1=convolutionBasex(M1)
    t0=time()
    L1=deconvolutions[methode](J1)
    t[i]=time()-t0
    print ("\n----- {:s} inverse ...".format(methode))
    print ("                {:.1f} sec".format(time()-t0))

plt.figure(figsize=(15,5))
plt.bar(deconvolutions.keys(),t,width=0.5)

```

Table des figures

2.1	Déconvolution +traitement de l'image	10
2.2	Soleil absorbant sous Blender	12
2.3	Signal observé	12
2.4	Signal observé	13
2.5	Signal observé	14
2.6	Déconvolution du signal	14
2.7	Superposition du signal de base et de la reconvolution du signal déconvolué	15
3.1	scéma explicitant l'obtention des données	16
4.1	Signal à convoluer	18
4.2	Signal après convolution	19
5.1	Signal non convolué et signal déconvolué (Basex)	23
5.2	Un autre signal non convolué et déconvolué (Hansen Law)	24
5.3	Diagramme en bâtons représentant l'écart-type de la déconvolution pour chaque méthode	24
5.4	Diagramme en bâtons représentant l'écart-type de la déconvolution pour la méthode Hansen Law en fonction du bruit ajouté	24
5.5	Diagramme en bâtons représentant l'écart-type selon le bruit ajouté pour la déconvolution de chacune des méthodes	25
5.6	Diagramme en bâtons représentant la durée d'exécution de la déconvolution pour chaque méthode	25