

Projet CLANU 2017

Transformée de Radon

A.HAJALI – A.HALIMI – Y.ZAHIDI

Introduction

Le projet CLANU est un programme pluridisciplinaire visant à envisager des solutions techniques et concrètes de problèmes réels. Par ses deux volets, mathématique et informatique ce projet constitue un véritable exercice nécessitant des connaissances et des capacités diverses. Nous sommes donc amenés à concevoir un logiciel Matlab/C++ illustrant une méthode mathématique.

Cette année, le projet CLANU a pour sujet 'La Transformée de Radon' qui est liée à de multiples techniques et problématiques de la vie réelle. En effet, la transformée de Radon apparaît dans des problèmes liés à l'imagerie 3D par tomographie ou tomodensitométrie, c'est le cas des scanners X, la tomographie à émission de positons ou la microscopie électronique en transmission.

Organisation du rapport

Ce rapport constitue une synthèse du travail réalisé afin de comprendre, implémenter et améliorer le programme associé à la transformée de Radon et son inversion. Il se structure en trois parties : une première partie visant à expliquer le principe de la transformée de Radon, son inversion ainsi que son intérêt du point de vue technique. Une deuxième partie abordera l'analyse numérique du code matlab associé ainsi que les méthodes mathématiques utilisées à cette fin. Une troisième partie porte sur le programme informatique en C++ ainsi que quelques tests de filtrage sur des données bruitées.

Principe de la transformée de Radon et son inversion

1.1 Contexte :

Un faisceau monochromatique de photons traversant des tissus biologiques (par exemple) voit son énergie varier en fonction des caractéristiques du volume physique traversé. C'est le phénomène de l'atténuation. La fonction d'atténuation dépend alors des variables spatiales et constitue une image en 3 dimensions des tissus biologiques.

Notre objectif dans cette étude serait donc de déterminer cette fonction d'atténuation à partir d'un ensemble de projections selon des droites concourantes. Cette relation entre une image et ses projections a été formalisée dès 1917 par le mathématicien Johann Radon qui donna une première

formulation mathématique d'une projection selon une direction φ sous forme d'une double intégrale :

$$p_{\varphi}(x') = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(x \cos(\varphi) + y \sin(\varphi) - x') dx dy$$

Où δ
(x)

représente l'impulsion de Dirac et $f(x,y)$ l'atténuation au point (x,y) du volume traversé.

La transformée de Radon P_{φ} correspond donc à la valeur mesurée par le détecteur qui est l'intégrale du coefficient d'atténuation le long d'une droite de direction φ . En faisant varier φ , on acquiert des projections tout autour du patient ou de l'objet scanné.

1.2 L'atténuation :

Nous allons définir la fonction d'atténuation A qui est nulle en dehors de la boule unité B_1 :

$$A : \mathbb{R}^2 \rightarrow \mathbb{R}$$

$$B_1 = \{x \in \mathbb{R}^2; |x| \leq 1\}.$$

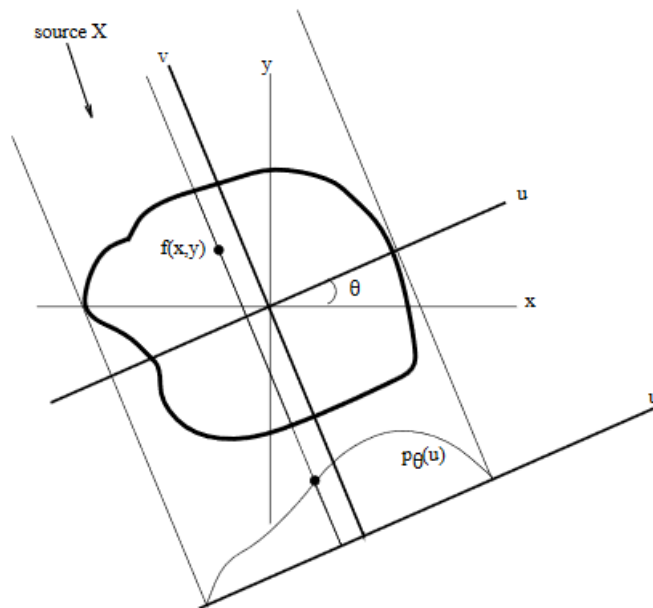


Fig1: principe de projection, $P_{\theta}(u)$ est l'intégrale de l'atténuation sur une droite orthogonale à l'axe u

1.3 Définition de la transformée de Radon:

On procède ensuite à un changement de variable qui va nous permettre de simplifier notre fonction P_{θ} et cela en considérant un repère polaire (Fig 1) :

$$(\theta, t) \in [0, \pi] \times [-1, 1],$$

Avec les vecteurs : $e_{\theta} = (\cos(\theta), \sin(\theta))$ et $e_{\theta}^{\perp} = (-\sin(\theta), \cos(\theta))$,

On a donc les droites $\Delta_{\theta,t}$ définies par :

$$\Delta_{\theta,t} = \{te_{\theta}^{\perp} + se_{\theta} ; s \in \mathbb{R}\}$$

Ainsi, On définit la transformée de Radon comme l'application linéaire suivante :

$$\mathcal{R} : A \rightarrow \mathcal{R}[A]$$

$$\mathcal{R}[A](\theta, t) = \int_{\Delta_{\theta,t}} A ds = \int_{[-1,1]} A(te_{\theta}^{\perp} + se_{\theta}) ds.$$

Où

$$\mathcal{R}[A] : [0, \pi] \times [-1, 1] \rightarrow \mathbb{R}$$

1.4 L'acquisition :

De manière plus concrète, acquisition des projections se fait à l'aide d'un scanner inventé bien plus tard que la transformée de Radon. Sur la figure 2, on peut voir l'évolution des différents dispositifs d'acquisition jusqu'aux scanners de quatrième génération où les détecteurs sont disposés en couronne et où seulement la source qui effectue la rotation.

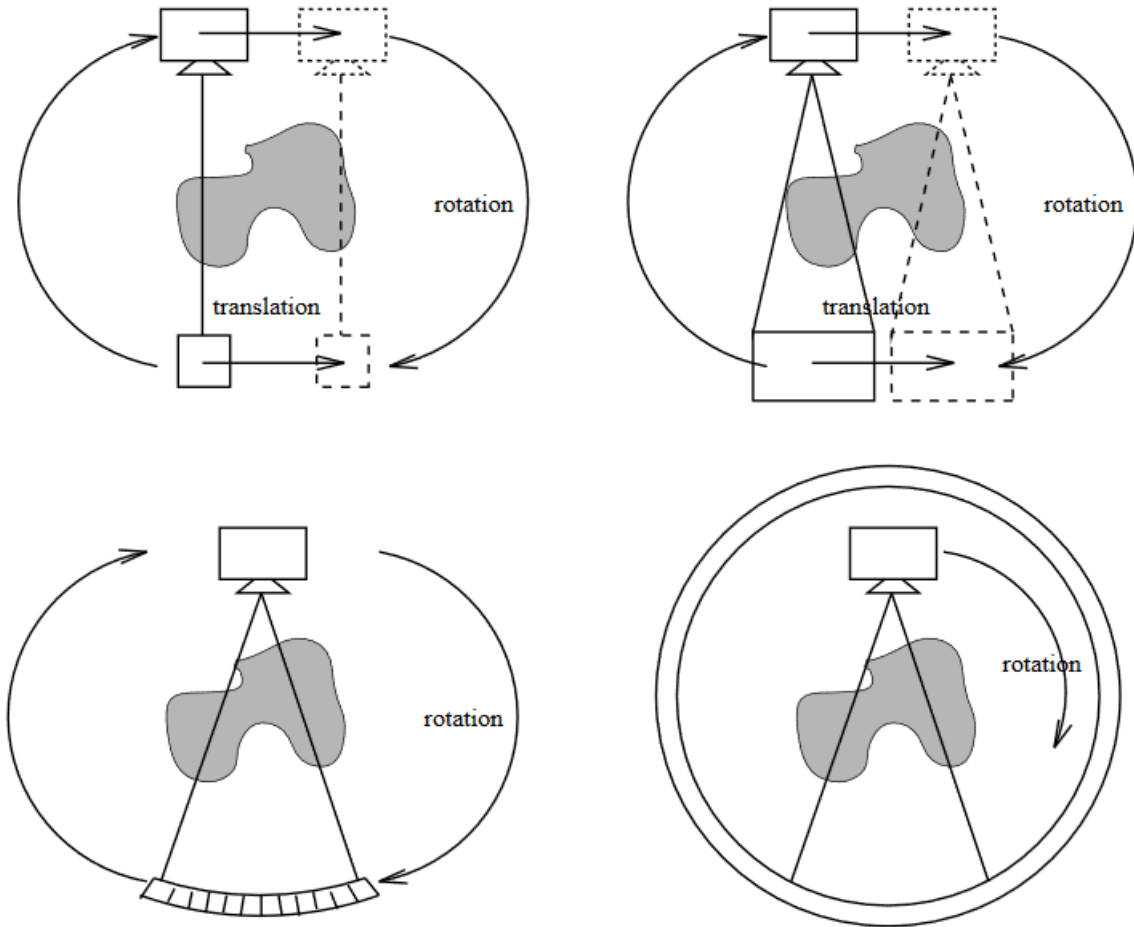


FIG. 2 – Les 4 générations de scanners X.

1.5 L'inversion de la transformée de Radon :

Dans notre étude, nous allons reconstruire notre fonction d'atténuation à partir de données I_{data} . Ces données sont les projections mesurées par les dispositifs d'acquisition.

Une façon simple pour remonter à $R[A]$ consiste à minimiser l'écart aux données J :

$$J(A) = \|\mathcal{R}[A] - I_{data}\|_{L^2([0,\pi] \times [0,1])}^2 = \int_{[-1,1]} \int_{[0,\pi]} (\mathcal{R}[A](\theta, t) - I_{data}(\theta, t))^2 d\theta dt.$$

Nous utiliserons dans ce cas l'algorithme itératif suivant :

$$A^{n+1} = A^n - \tau \mathcal{R}^*[\mathcal{R}[A^n] - I_{data}]$$

Ici τ représente le pas de descente de l'algorithme et \mathcal{R}^* la transformée adjointe de Radon. Après un nombre suffisant d'itérations on arrivera à approcher correctement la fonction d'atténuation A .

1.6 Transformée adjointe de Radon :

Pour une application linéaire, la transformée de Radon adjointe \mathcal{R}^* de \mathcal{R} est définie comme suit :

$$\langle \mathcal{R}[A], I \rangle_{L^2([0,\pi] \times [-1,1])} = \langle A, \mathcal{R}^*[I] \rangle_{L^2(B_1)}, \quad \forall A : B_1 \rightarrow \mathbb{R}, \text{ et } \forall I : [0, \pi] \times [-1, 1] \rightarrow \mathbb{R}$$

On peut remarquer que :

$$\begin{aligned} \langle \mathcal{R}[A], I \rangle_{L^2([0,\pi] \times [-1,1])} &= \int_{[0,\pi]} \int_{[-1,1]} \mathcal{R}[A](\theta, t) I(\theta, t) d\theta dt \\ &= \int_{[0,\pi]} \int_{[-1,1]} \int_{[-1,1]} A(te_\theta^\perp + se_\theta) I(\theta, t) d\theta dt ds \end{aligned}$$

Puis en utilisant le changement de variable $(t, s) \mapsto x = te_\theta^\perp + se_\theta$,

On a :

$$\begin{aligned} \langle \mathcal{R}[A], I \rangle_{L^2([0,\pi] \times [-1,1])} &= \int_{B_1} A(x) \int_{[0,\pi]} I(\theta, x \cdot e_\theta^\perp) d\theta dx \\ &= \int_{B_1} A(x) \mathcal{R}^*[I](x) dx. \end{aligned}$$

On peut alors déduire par identification une expression explicite de la transformée adjointe de Radon :

$$\mathcal{R}^*[I](x) = \int_{[0,\pi]} I(\theta, x \cdot e_\theta^\perp) d\theta$$

Analyse numérique

Réponses aux questions mathématiques

Question 1 : Commentaire du code Fig3 & 4

Le code de calcul de la transformée de radon et son adjointe se fait de la même manière. D'abord, on discrétise θ et t pour faire une projection sur (x,y) et retrouver l'ensemble des points de A discrétisée. Une fois les ppx et ppy retrouvés, on cherche les indices de ces points dans la matrice Im qui représente l'image d'atténuation.

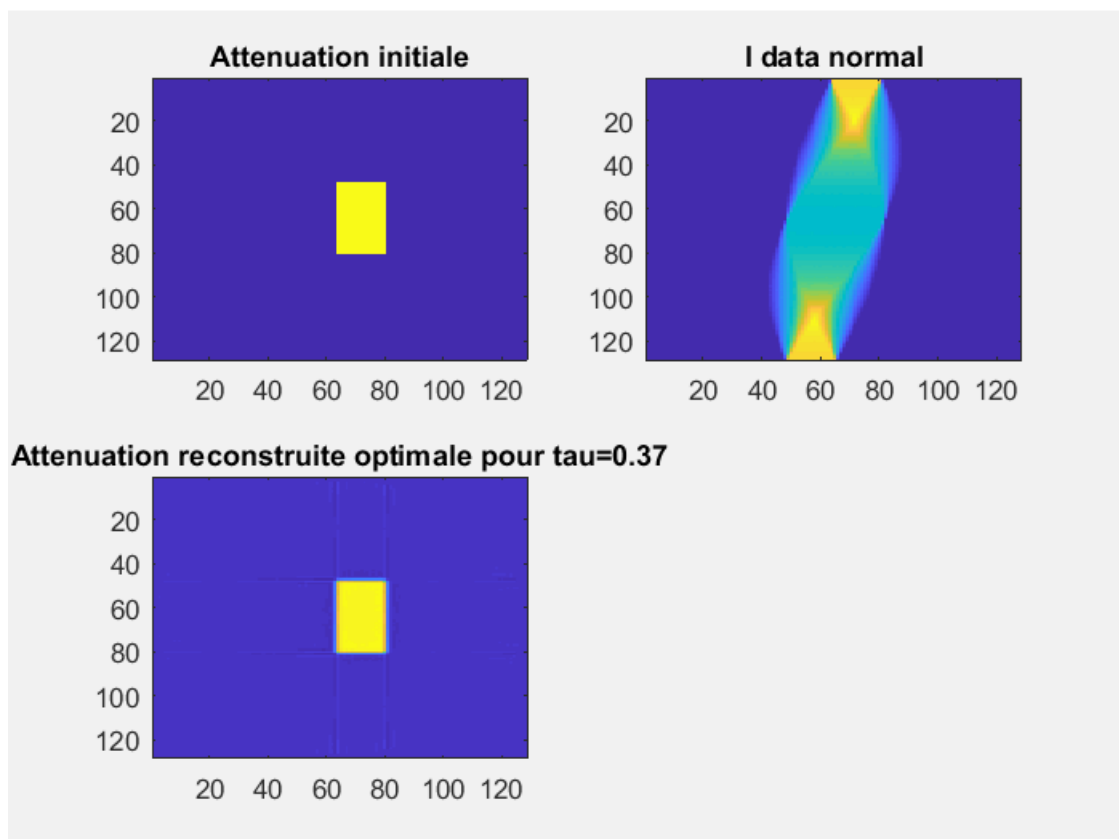
Comme la transformée de radon représente l'intégrale de A , on peut la calculer numériquement en calculant la surface de A point par point. Ceci se fait de la manière suivante : pour chaque point de A , on calcule les 4 petits fragments de surface autour et ça revient donc à calculer les 4 points de I les plus proches du "vrai" point puis, on calcule leur moyenne pour enfin retrouver I - radon du point de départ de A . Cette opération se fait pour tous les points de A et on arrive finalement à retrouver tous les points de I -radon.

Question 2 : Premier test de l'algorithme de reconstruction

On teste l'algorithme de reconstruction proposé sur un premier exemple où la fonction d'atténuation est égale à $A(x) = 1_{[1=4;1=4][0;1=4]}$ et avec les paramètres $N_1 = N_2 = N_t = N = 2^7$.

```
K=50;
tau=0.37; % 0.37 tau le plus grand à peu près
A=0;
for i=1:K
    A=A-tau*radon_tranform_adjoint(radon_tranform(A,N_t,N_theta)-Id,N1,N2);
end
```

D'où :

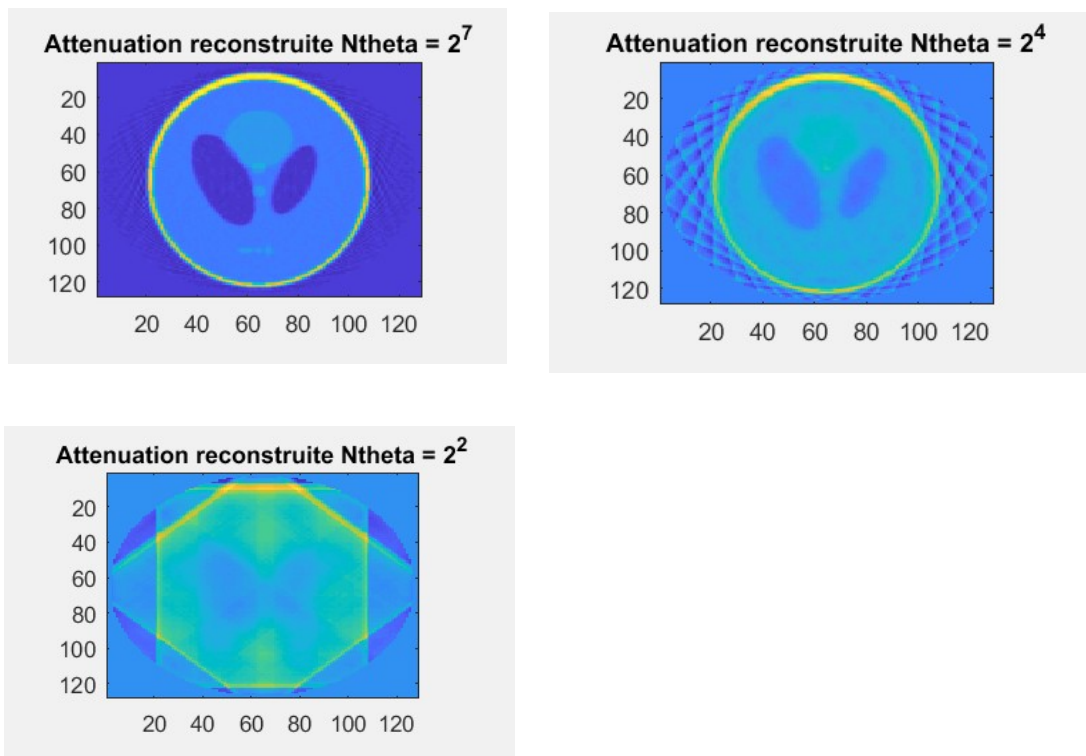


Après plusieurs essais pour des valeurs différentes de τ , la plus grande valeur à laquelle on a pas constaté de perte des données est : 0,37.

Question 3 : Reconstruction de $A(x)$ avec une atténuation plus compliquée

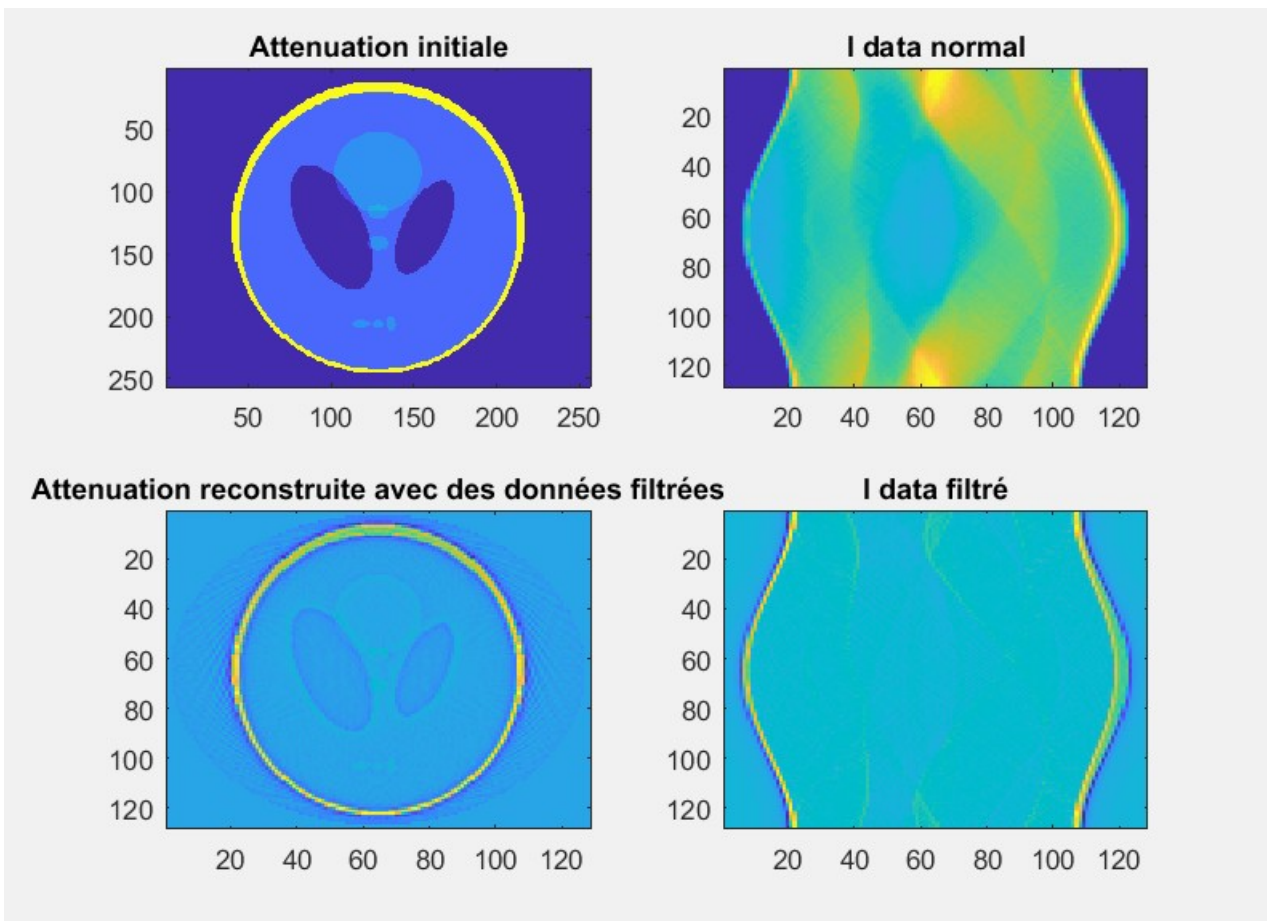
En utilisant la fonction `phantom` de Matlab pour générer une image un peu plus compliquée de l'atténuation, on teste l'algorithme de reconstruction.

En faisant diminuer le nombre d'orientations N -theta, on constate une perte de la précision avec l'apparition des droites de projections pour des angles donnés ($360^\circ/N$ -theta). L'augmentation de N -theta ne change pas significativement notre image en sortie.



Question 4 : Application d'un filtre

On applique maintenant le filtre h aux données I_{data} qui est défini dans l'espace de Fourier par $F'(f) = |f|$ et en utilisant l'algorithme proposé. La reconstruction donne les résultats suivants :



On constate une perte d'information due à l'application du filtre aux données.

Question 5: Bruit additif et filtrage de ce dernier par convolution avec une fonction gaussienne

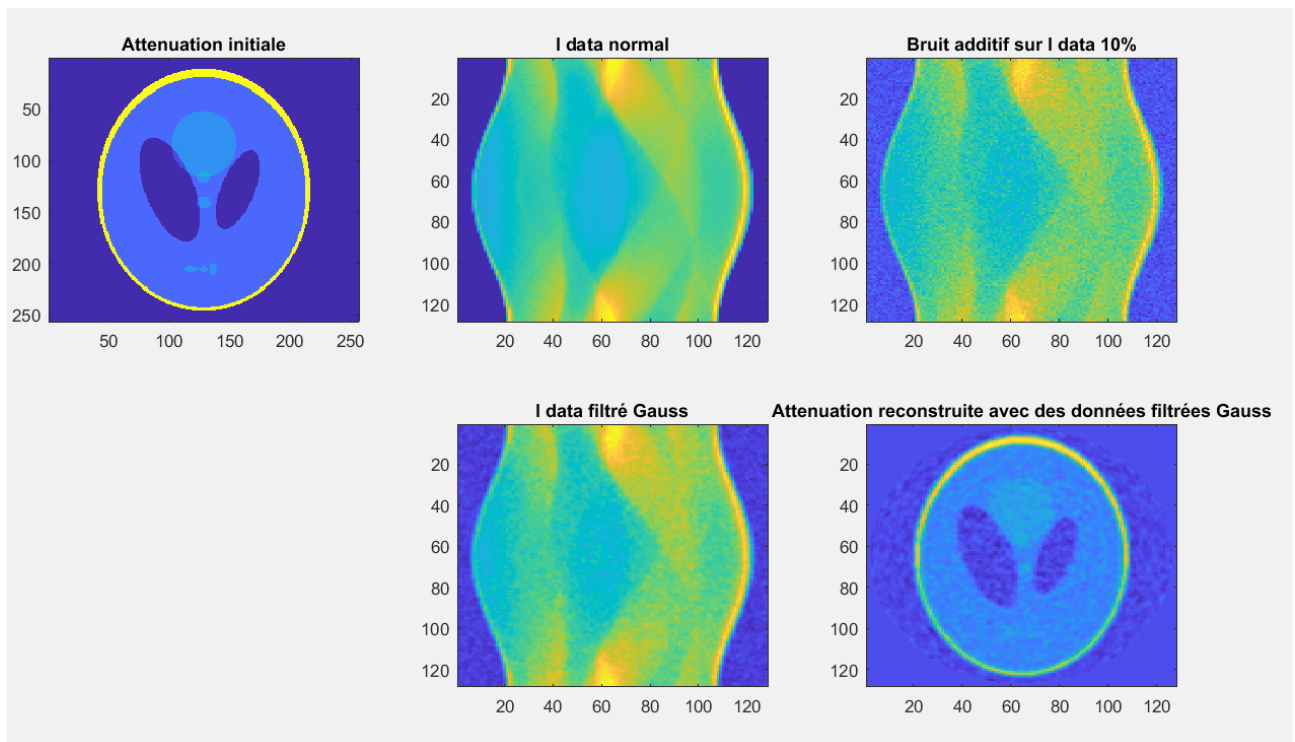
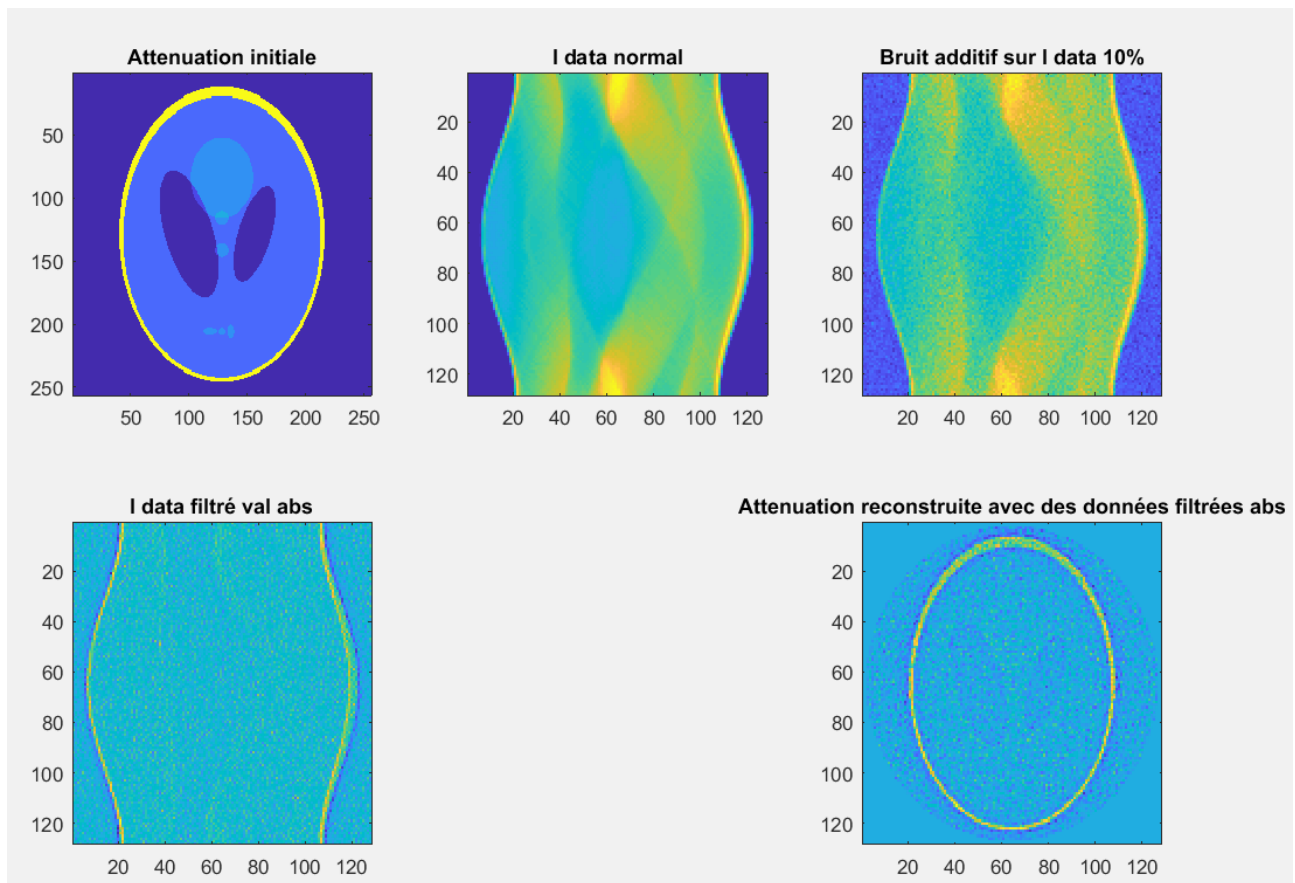
Dans un premier temps, nous allons comparer le filtre proposé à la question précédente avec le filtrage avec une gaussienne.

```

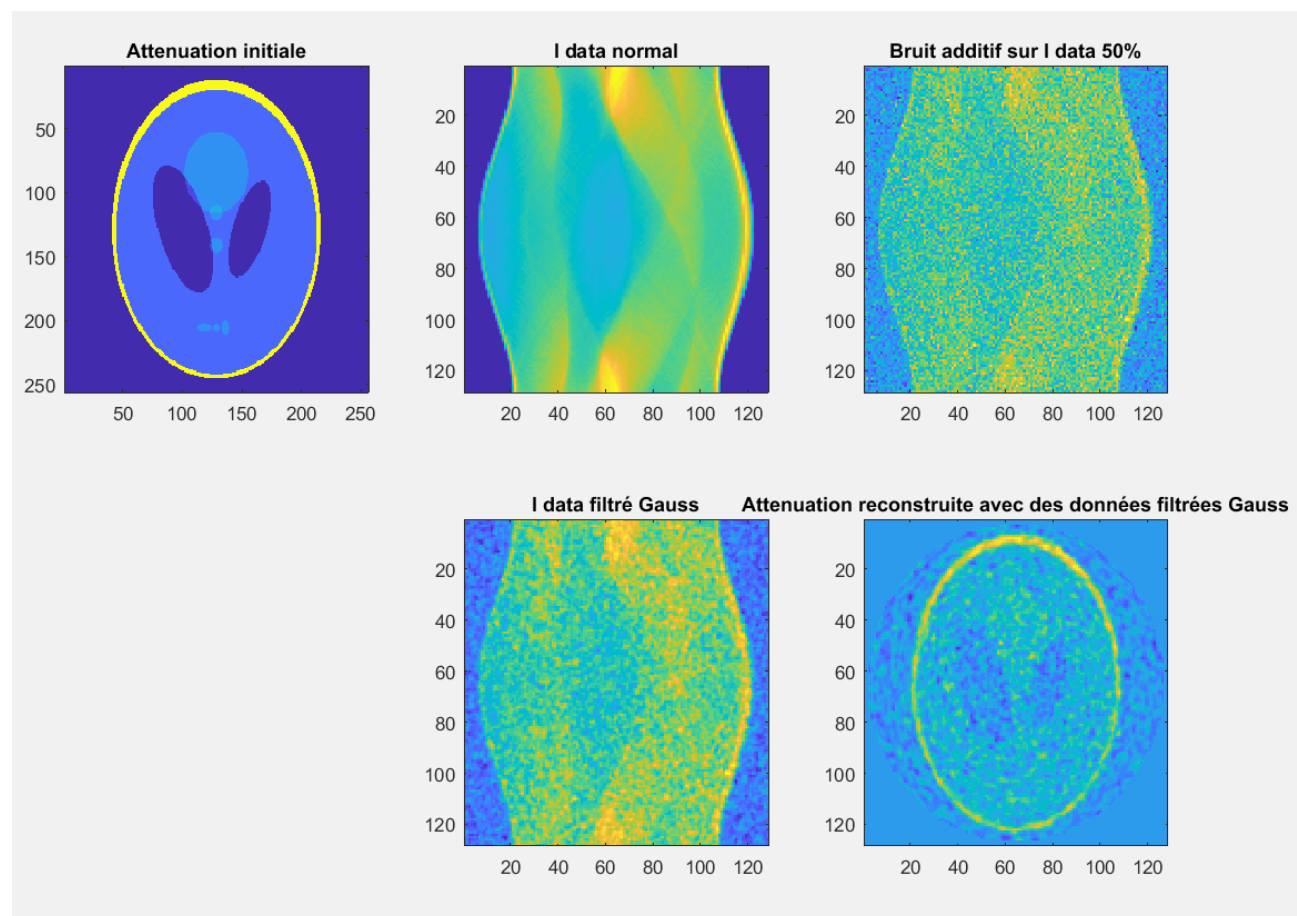
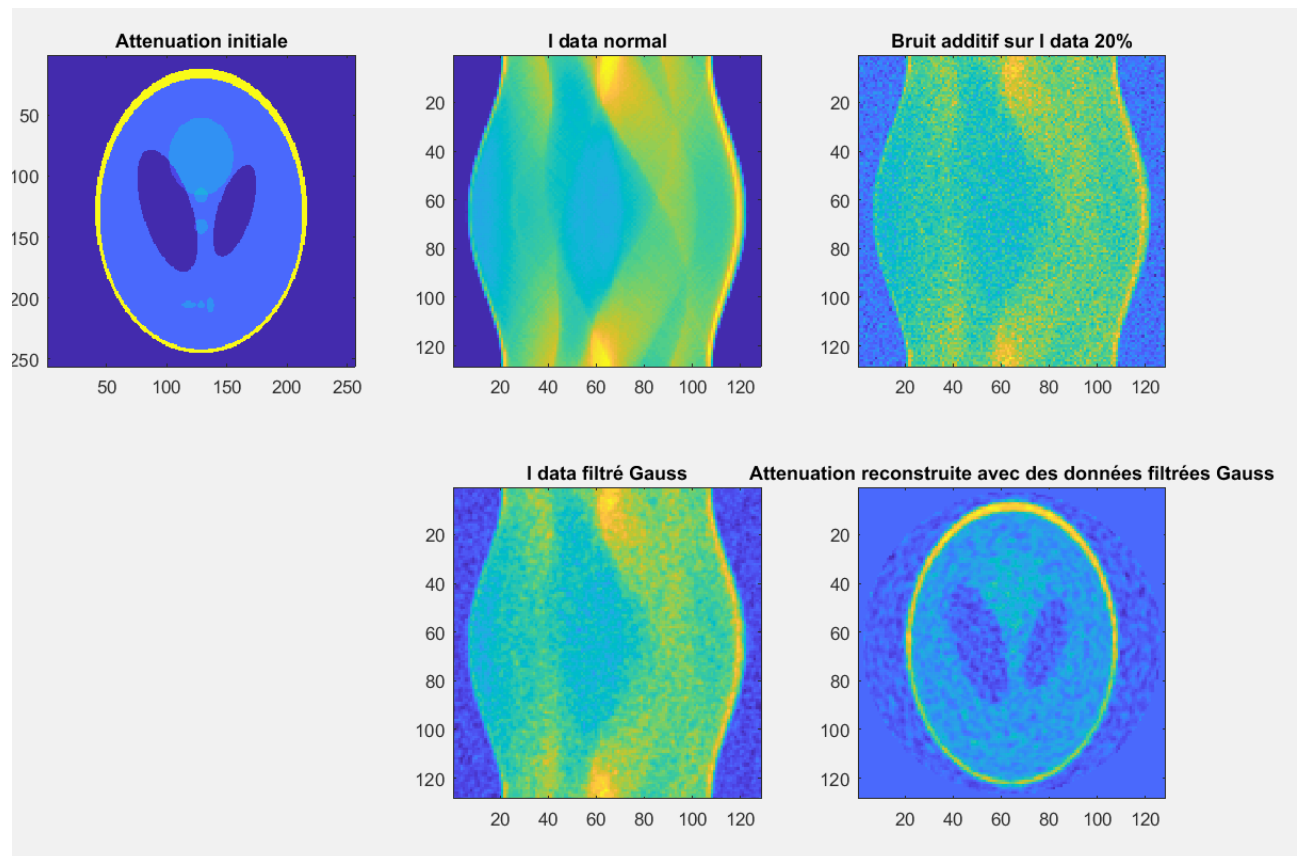
moyId = mean2(Id);
Br = moyId*0.05*randn(N);
Id = Id + Br;
subplot(2,3,3),imagesc(Id), title('Bruit additif sur I data 5%');

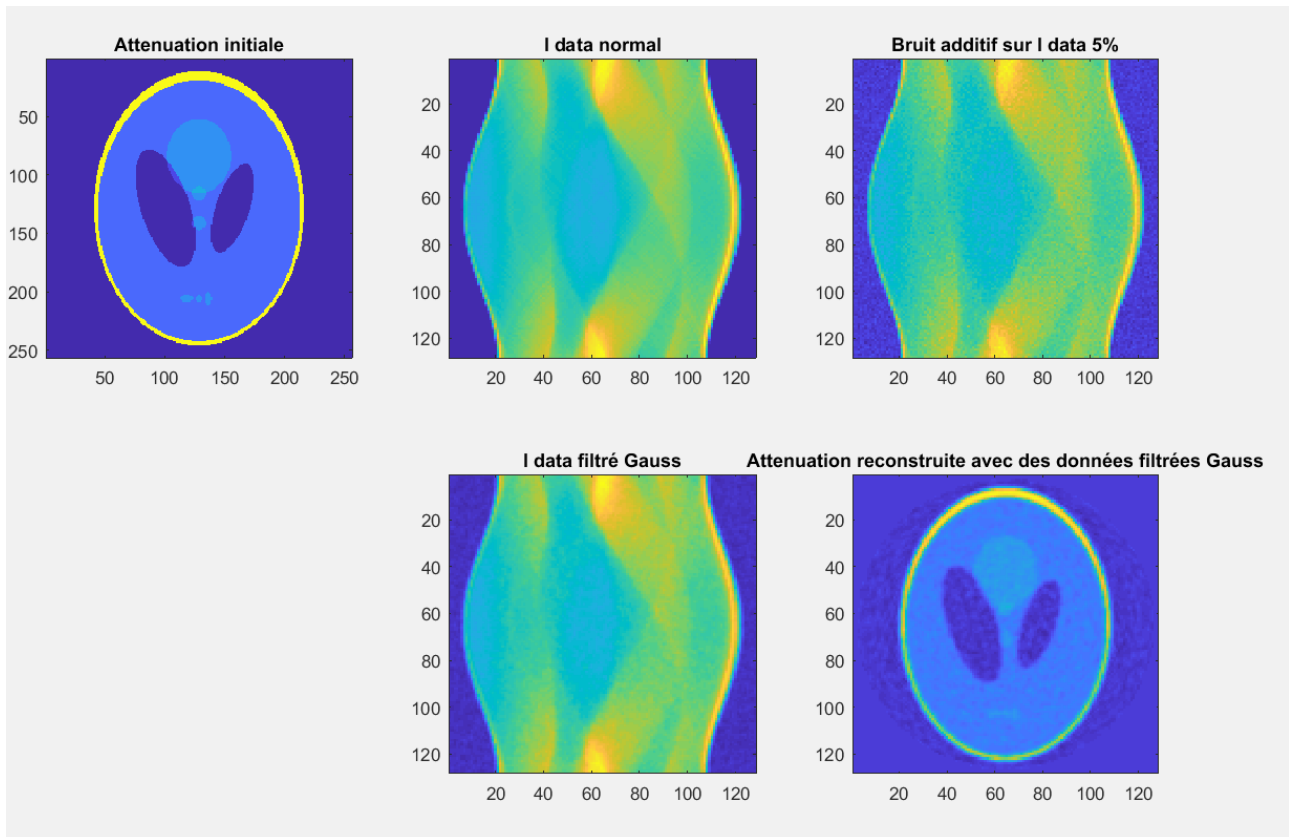
sig = 0.00001;|
k = [0:N/2,-N/2+1:-1];
[K1,K2] = meshgrid(k,k);
M = exp(-4*(pi^2)*sig*(K1.^2 + K2.^2));
Id=ifft2(M.*fft2(Id));
subplot(2,3,5),imagesc(Id), title('I data filtré Gauss');

```

Le résultat avec le filtre de la question quatre n'est pas convaincant, ce dernier filtre les basses fréquences. On décide par la suite de faire des tests avec le filtre gaussien en additionnant différents degrés de bruit.





D'après le résultat obtenu avec plusieurs valeurs de bruit. On peut conclure que notre reconstruction est globalement correcte avec un bruit inférieur à 5 %.

Programmation informatique

La partie informatique rédigée en c++ du sujet a consisté en deux projets distincts : Le premier est celui donné sur moodle contenant toutes les fonctions nécessaires à l'exécution d'une "exécutable" qui prend en entrée une image de format .png et donne en sortie sa transformée de radon en format .mhd

Ainsi pour répondre à la question 1 :

Pour générer un sinogramme de dimensions $N_t = 128$ et $N_{\theta} = 128$ pour un angles de départ de $eAngle = -90^\circ$ et un angle d'arrivée $sAngle = 90^\circ$ et sachant que l'image d'entrée test1.png et le testR.mhd de sortie se trouvent dans le répertoire C:/Users/Yacine/Desktop/CLANU/ du pc, la ligne de commande à utiliser est alors :

```
--input=C:/Users/Yacine/Desktop/CLANU/test1.png
--output=C:/Users/Yacine/Desktop/CLANU/testR.mhd --ntheta 128 --nt 128 --start -90 --end 90
```

Concernant la question 2 :

Étant donné que des fonctions telles que `meshgrid` ou encore les fonctions effectuant les opérations entre matrices et scalaires, ou encore entre matrices et matrices n'existent pas naturellement dans le langage c++, il a fallu donc les programmer une à une dans le fichier source `fonctions.cpp`

On retrouve ainsi les fonctions suivantes qui n'existaient pas au préalable dans le code fourni sur moodle, mais qui ont dûes être créées dans le but d'effectuer les calculs nécessaires pour la transformée adjointe (définie aussi dans `fonctions.cpp`) et pour l'algorithme de reconstruction (implémenté dans notre `main`) :

```
void displayVect(float *v, int tailleVect);  
void displayMatrice(float **m, int sizeX, int sizeY);
```

qui ont servi à afficher les matrices et vecteurs (pour les phases de test et de débogage)

```
float **meshgrid(float *x, float *y, int tailleX, int tailleY, bool isX);
```

qui correspond à la version c++ de la fonction `meshgrid` de matlab (le booleen `isX` permet d'avoir X en sortie lorsqu'il est vrai et Y sinon)

ainsi que les fonctions :

```
float ** MultiplyMatriceByScalar(float **Iin, int tailleX, int tailleY, float scalar);  
float ** SubstractMatrice(float **m1, float **m2, int tailleX, int tailleY);  
float ** AddScalarToMatrice(float **Iin, int tailleX, int tailleY, float scalar);  
float ** floorMatrice(float **Iin, int tailleX, int tailleY);
```

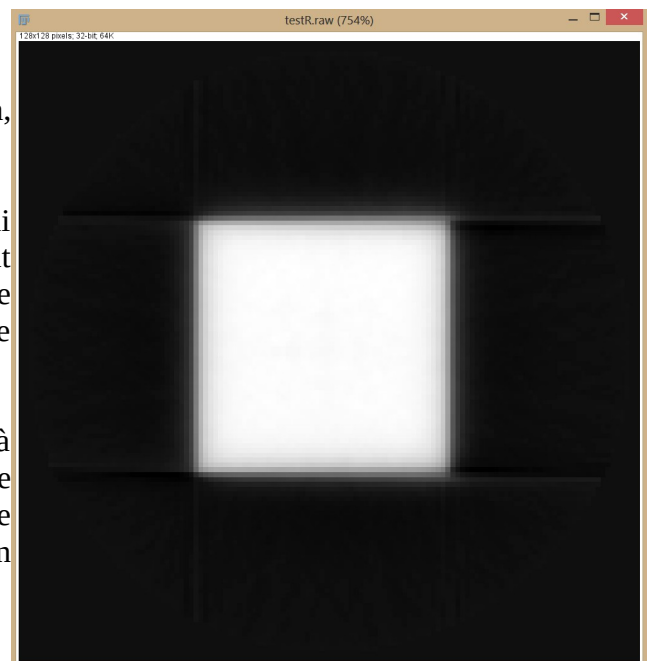
pour les calculs de soustraction entre deux matrices, d'addition de scalaire sur chaque case d'une matrice, de multiplication d'une matrice par un scalaire et d'arrondissement à défaut de chaque élément d'une matrice.

et enfin :

```
void RadonAdjointTransform(float **Iout, float **Iin,  
int N1, int N2, int Ntheta, int Nt, int sA, int eA);
```

qui représente le "coeur" de notre programme et qui permet de nous donner en sortie le tableau de float correspondant à la transformée de radon adjointe calculée à partir du tableau de float de la transformée de radon donnée en entrée.

Enfin, le `main` est programmé de façon très similaire à celle du `main` donné sur moodle, de sorte à ce que chaque paramètre choisi par l'utilisateur de l'exécutable soit passé correctement en argument de la fonction (notamment à l'aide du parser). Cf.figure.



Algorithmes implémentés

Cf. Fichier .zip

Conclusions

Dans cette étude, nous avons analysé et testé une application de la transformée de radon sur des images dans divers cas : bruitées/non bruitées, avec ou sans filtres. Cela a été réalisé sur deux langages différents de programmation à savoir Matlab et C++, ce qui a permis de les comparer et aussi de distinguer l'utilité de chacun.

On peut dire après la réalisation de ces deux phases : mathématique & informatique, que la transformée de Radon peut être implémentée en Matlab ou en C++. Or, pour un langage de programmation comme le C++, l'implémentation de certaines fonctions mathématiques (déjà intégrées en Matlab) rend ce langage peu pratique pour la rédaction de programmes impliquant des fonctions mathématiques.

L'étude mathématique a permis de tester différents filtres et de les comparer. Ainsi, la fonction gaussienne demeure efficace pour filtrer un bruit additif. Ce bruit ne doit tout de même pas dépasser un certain seuil, au risque d'avoir des images moins exploitables.

Répartition du travail dans le trinôme

Voici un tableau récapitulatif de la répartition des différentes tâches qui a permis la réalisation de ce rapport :

	Ahmad HAJALI	Assil HALIMI	Yacine ZAHIDI
Introduction et principe de la transformée de Radon	X	X	
Partie mathématique			
Question 1	X		
Question 2		X	
Question 3			X
Question 4	X	X	
Question 5		X	X
Partie informatique			
Question 1			X
Question 2	X	X	X
Question 3		X	X
Conclusion		X	

Sources & bibliographie

DIP Lecture 18: Reconstruction from parallel projections and the Radon transform :
<https://www.youtube.com/watch?v=ZgcD4C-4u0Q>

Théorème de Radon, Wikipédia https://fr.wikipedia.org/wiki/Th%C3%A9or%C3%A8me_de_Radon

Reconstruction d'images de tomographie -Isabelle BLOCH Telecom ParisTech, d'épartement TSI,
CNRS UMR 5141, 46 rue Barrault, 75634
Paris Cedex 13

Fascicule-sujet du CLANU 2017 : La transformée de Radon.