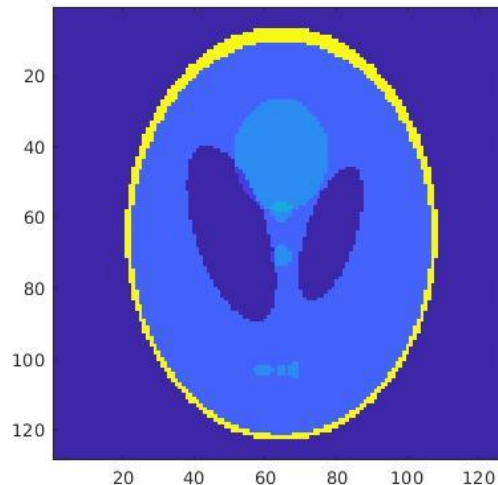


TP Reconstruction d'images

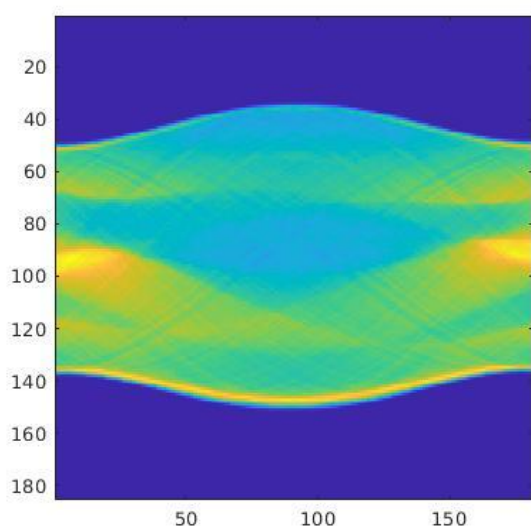
I. Reconstruction TDM

Question 1 : Instantialisation des variables.

Question 2 : Visualisation du "fantôme" utilisé : *Shep Logan Phantom*



Question 3 : Ci-dessous le sinogramme généré par la transformée de Radon appliquée sur le "vrai" fantôme \mathbf{x}_{true} .



Calculées à partir du sinogramme, le **tableau 1** présente :

a : La rétroprojection filtrée, qui pour chaque point correspond à l'intégrale des lignes passant par ce point (a)

b + c : La reconstruction de l'image d'origine, en ajoutant un filtre $|\sigma|$ à la rétroprojection, d'après le théorème d'inversion de la transformée de Radon :

$$f(\mathbf{r}) = \frac{1}{2} \int_0^{2\pi} \int_{-\infty}^{+\infty} |\sigma| \mathcal{F}_1(g_\theta)(\sigma) e^{2i\pi\sigma\mathbf{r}\cdot\boldsymbol{\omega}(\theta)} d\sigma d\theta$$

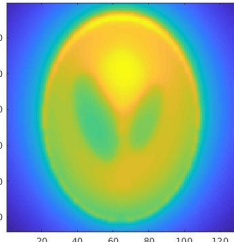
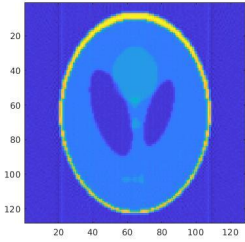
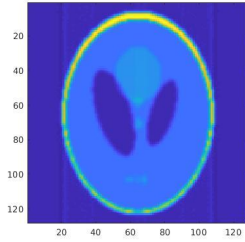
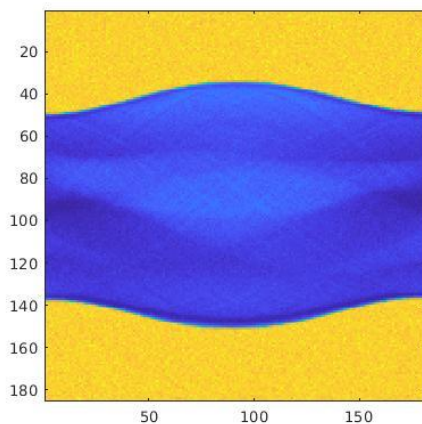
a. Non Filter	b. Filtre Ram-Lak	c. Filtre Hann
		

Tableau n°1 : Comparaison des reconstructions à partir du fantôme x_{true} .

Bien que le filtre de Hann soit connu pour atténuer les hautes fréquences de l'image qui sont sources de bruit, la reconstruction correspondante (c) semble assez équivalente à la reconstruction (b) réalisée avec le filtre rampe, en termes de résolution d'image (ou réponse impulsionnelle).

Question 4 :



Les **données du scanner y_i** , représentent la mesure du nombre de photons reçus au cours de la durée d'acquisition τ , après avoir traversé la carte d'atténuation x , pour chaque détecteur i . Elle peuvent être modélisées par une loi de Poisson, dans la mesure où chaque y_i prend la valeur d'une variable de comptage positive, indépendante du temps. L'espérance mathématique de cette loi de Poisson peut être retrouvée grâce à **la loi de Beer-Lambert** :

$$\bar{y}(x^{\text{true}}) = \tau I_0 e^{-Ax^{\text{true}}}$$

qui représente le nombre de photons moyen reçus, étant donné la carte d'atténuation x_{true} . En connaissant la valeur de l'espérance de la loi de Poisson, on peut générer n réalisations aléatoires tombant dans la distribution. Cette opération consiste à ajouter un "bruit de Poisson", visible au sinogramme ci-dessus. Le bruit est visible dans l'aspect granuleux de l'image. **L'enjeu est maintenant de parvenir à reconstruire l'image à partir de ce sinogramme, donc sans connaître ni x_{true} , ni y_{bar} .**

Question 5 :

La grandeur b_i , représente le **logarithme de la probabilité que y_i photons soit reçus par le détecteur i , au bout de τ secondes**, d'après la formule de Beer-Lambert (question 4). Elle correspond donc à une approximation de Ax_{true} par substitution. On calcule donc son image par l'opérateur de rétroprojection filtrée.

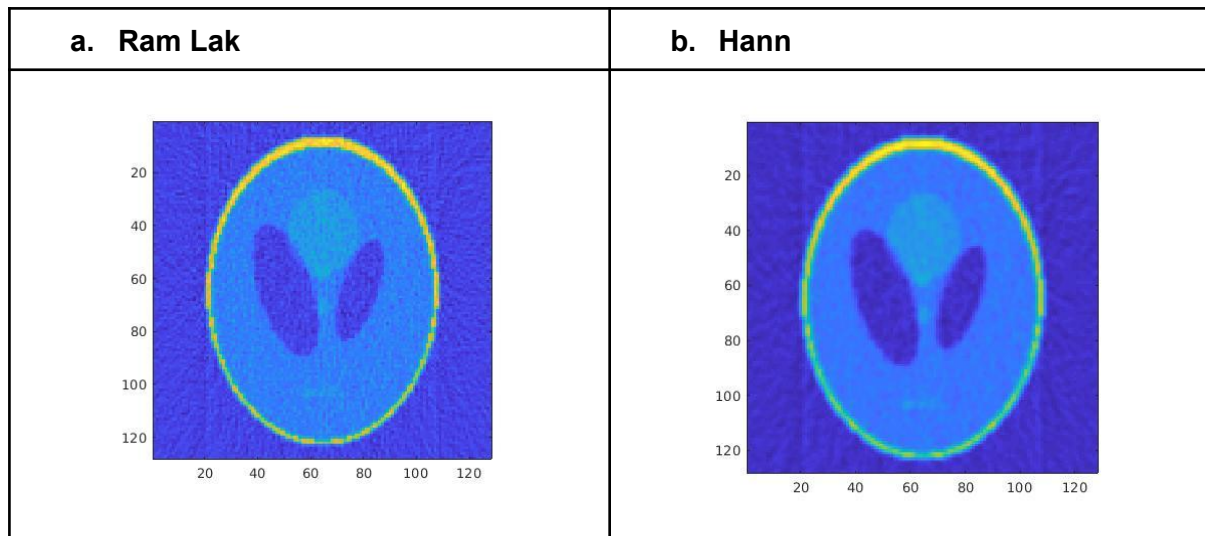
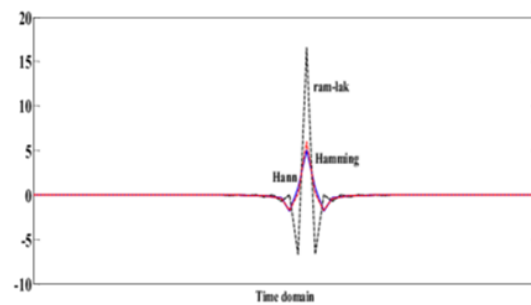
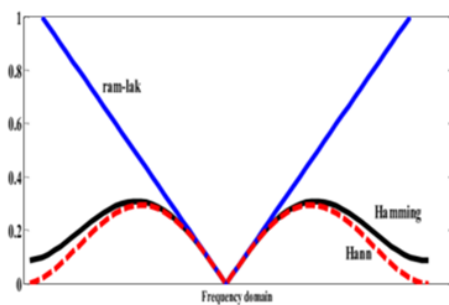


Tableau n°2 : Rétroprojection filtrée

En raison du bruit de Poisson, la reconstruction de Hann (b) apparaît cette fois-ci plus lissée que la reconstruction Ram-Lak (a).



Cela est logique puisqu'un tel filtre laisse passer moins de hautes fréquences.

Question 6 :

On peut également chercher à remonter à x_{true} en retrouvant le paramètre y_{bar} qui paramétrise la loi de Poisson (cf. question 4). En effet, on peut chercher la valeur de x la plus à même d'expliquer les observations de valeurs des y_i . Mathématiquement, cela revient à maximiser la vraisemblance L de x , sachant les données y . Cela revient à minimiser la fonction quadratique :

$$f(x) = \frac{1}{2} \|Ax - b\|_W^2$$

(Démonstration pag suivante)

Malheureusement, il est numériquement impossible de calculer directement la solution de ce problème de minimisation en utilisant la formule démontrée ci-dessus, en raison de la **complexité calculatoire requise pour inverser la matrice $[A^TWA]$** .

$$f(x) = \frac{1}{2} \|Ax - b\|_W^2$$

$$= \frac{1}{2} (Ax - b)^T \cdot W \cdot (Ax - b)$$

$\left. \begin{array}{l} \parallel \cdot \parallel = x^T B x \quad \forall x \in \mathbb{R}^n \end{array} \right\}$

$$\nabla f(x) = \frac{1}{2} \nabla \left((Ax - b)^T, W \cdot (Ax - b) \right)$$

$$\nabla(AB) = \nabla A \cdot B + A \nabla B$$

$$= \frac{1}{2} \left(\nabla(Ax - b)^T, W \cdot (Ax - b) + (Ax - b)^T \cdot \left[\underbrace{\nabla W}_{0} (Ax - b) + W \cdot \nabla(Ax - b) \right] \right)$$

$$= \frac{1}{2} \left(A^T W (Ax - b) + (Ax - b)^T W A \right)$$

$$= \frac{1}{2} \left(\underbrace{A^T W A x}_{A^T W (Ax - b)} - \underbrace{A^T W b}_{(Ax)^T (WA)} + \underbrace{A^T x^T W A - b^T (WA)^T}_{-(WA)^T b} \right)$$

$$(AB)^T = B^T A^T$$

$$(WA)^T Ax$$

$$A^T W^T A x - A^T W^T b$$

$$A^T W A x - A^T W b$$

$$A^T W (Ax - b)$$

$$\downarrow W = W^T \text{ (diagonal)}$$

$$= \frac{1}{2} (2 A^T W (Ax - b))$$

$$\boxed{\nabla f(x) = A^T W (Ax - b)} \quad \square$$

$$\arg \min_{x \in \mathbb{R}^n} f(x) \Leftrightarrow \nabla f(x) = 0 \Leftrightarrow A^T W (Ax - b) = 0$$

$$A^T W A x = A^T W b$$

$$\boxed{x = [A^T W A]^{-1} A^T W b} \quad \square$$

Question 7 :

Nous cherchons donc un minimum de f en utilisant un algorithme itératif de **descente de gradient**, donc nous pouvons observer qualitativement la vitesse de convergence après un nombre n d'itérations (Tableau 3), après avoir réglé manuellement **le pas de descente** à 10^{-5} .

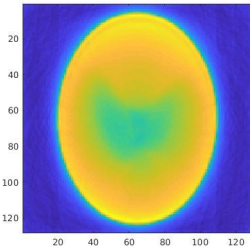
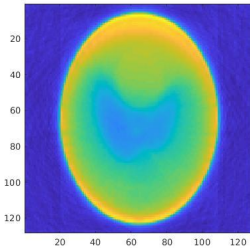
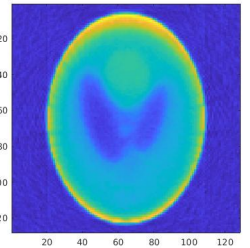
n = 10	n = 20	n = 50
		

Tableau 3 : Vitesse de convergence qualitative avec $\alpha = 10^{-5}$

Question 8 :

Pour accélérer la convergence de l'algorithme d'optimisation utilisé, nous avons recours à un transfert d'optimisation, qui consiste à trouver les minima d'une suite de fonctions polynomiales, définies dans le voisinage de la valeur x actuelle.

D'après les hypothèses, $\forall q \in \mathbb{N}$

$$\begin{cases} x^{(q+1)} = \arg \min_{x \in \mathbb{R}^m} Q(x | x^{(q)}) & (1) \end{cases}$$

$$\begin{cases} Q(x | z) - Q(z | z) \geq f(x) - f(z) & (2) \end{cases}$$

$$(1) \Rightarrow Q(x^{(q+1)} | x^{(q)}) \leq Q(x^{(q)} | x^{(q)}) \quad \forall q < q+1$$

$$\Rightarrow Q(x^{(q+1)} | x^{(q)}) \leq Q(x^{(q)} | x^{(q)})$$

$$\Rightarrow Q(x^{(q+1)} | x^{(q)}) - Q(x^{(q)} | x^{(q)}) \leq 0$$

$$(2) \Rightarrow f(x^{(q+1)}) - f(x^{(q)}) \leq 0$$

$$\Rightarrow \boxed{f(x^{(q+1)}) \leq f(x^{(q)})} \quad \forall q$$

Question 9 :

En s'appuyant sur la définition de x^* :

$$\exists x^* \in \mathbb{R}^m, \quad \exists P > 0, \quad \forall q > P, \quad x^{(q)} = x^*.$$

A partir du rang P , les valeurs de q deviennent constantes et égales à x^* . Or d'après la question 8, nous avons montré que les valeurs de Q et donc de f continuaient de décroître au fil des itérations. **Ainsi, x^* est le point stationnaire qui donne le minimum de f .**

From x^* , the minimum is reached since there is no change in the iteration of $x(q)$ to $x(q+1)$, so $f(x(q)) = f(x(q+1))$, meaning using $f(x(q+1)) < f(x(q))$

Question 10 :

$$f(x) \leq f(x^{(q)}) + \nabla f(x^{(q)}) (x - x^{(q)}) + \frac{1}{2} (x - x^{(q)})^T D (x - x^{(q)}) \\ \triangleq Q(x | x^{(q)}). \quad (5)$$

Q définie en (5) est un TO pour f car il vérifie

$$Q(x | z) - Q(z | z) \geq f(x) - f(z) \quad (2)$$

$$\nabla_1 Q(z | z) = \nabla f(z) \quad (3)$$

$$Q(x | z) \geq f(x) \quad \text{par (5)}$$

$$Q(z | z) \geq f(z) \quad "$$

$$\Rightarrow Q(x | z) - Q(z | z) \geq f(x) - f(z)$$

$$\nabla_1 Q(z | z) = \nabla f(z) - \underbrace{\nabla(\nabla f(z)(z-z))}_0 - \underbrace{\frac{1}{2} \nabla(z-z)^T D (z-z)}_0 = \nabla f(z)$$

Question 11 :

Exprimons le minimum de Q tel qu'exprimé par la relation (5).

$$Q(x|x^{(q)}) = f(x^{(q)}) + \nabla f(x^{(q)})(x - x^{(q)}) + \frac{1}{2}(x - x^{(q)})^T D (x - x^{(q)})$$

$$\underset{x \in \mathbb{R}^m}{\operatorname{argmin}} (Q(x|x^{(q)})) ?$$

cf. question 6

$$\text{Calculons } \nabla Q(x|x^{(q)}) = \nabla f(x^{(q)}) + D (x - x^{(q)})$$

$$\text{Donc } \nabla Q(x|x^{(q)}) = 0$$

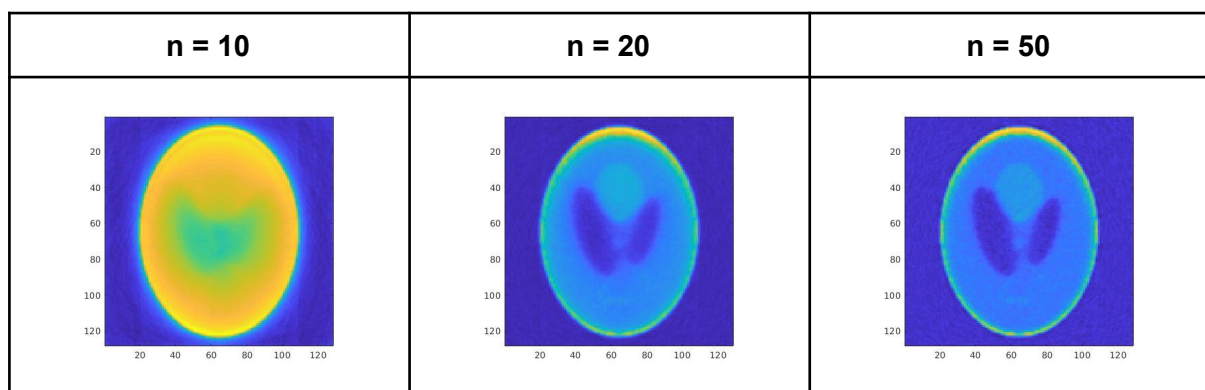
$$\Leftrightarrow D(x - x^{(q)}) = -\nabla f(x^{(q)})$$

$$\Leftrightarrow x - x^{(q)} = -D^{-1} \nabla f(x^{(q)})$$

$$\Leftrightarrow x = x^{(q)} - D^{-1} \nabla f(x^{(q)})$$

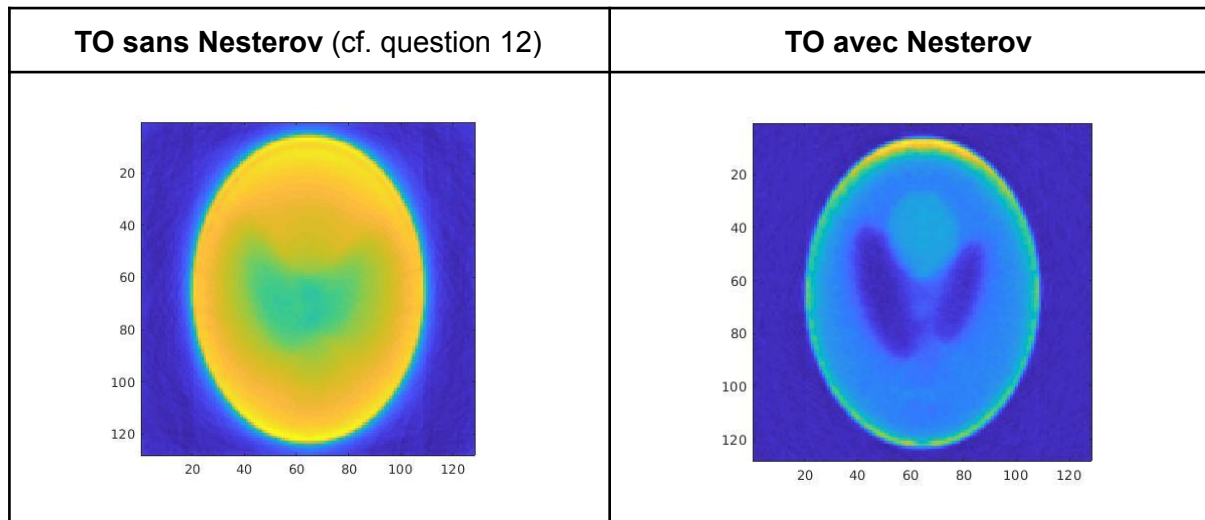
Question 12 :

De même qu'à la question 7, nous pouvons visualiser la convergence de l'algorithme de minimisation à n itérations. Il apparaît clair ici que le transfert d'optimisation implémenté à partir de la formule démontrée précédemment, est **bien plus rapide que la descente de gradient**, si on compare les itérations respectives.



Question 13 :

En utilisant l'accélération de Nesterov, à l'itération $n=10$.



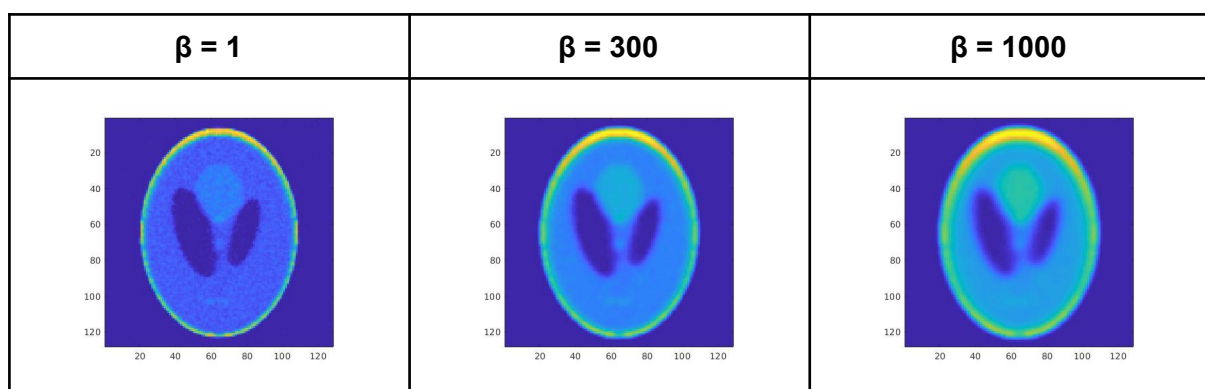
L'accélération semble très efficace, étant donné qu'à partir seulement de la 10^{ème} itération, l'image semble déjà reconstruite de manière convaincante.

Question 14:

On peut quantifier la rapidité de convergence de l'optimisation du transfert d'optimisation par calcul de coût à chaque itération. Le graphique obtenu ci-dessous permet en effet de confirmer l'efficacité de la méthode de Nesterov.

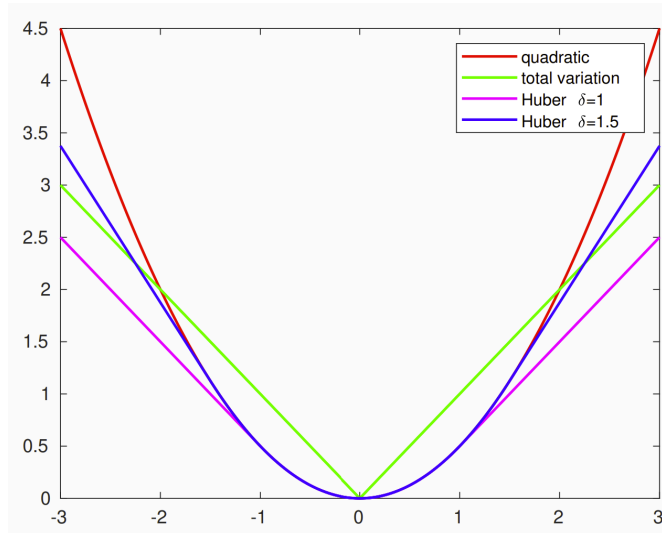
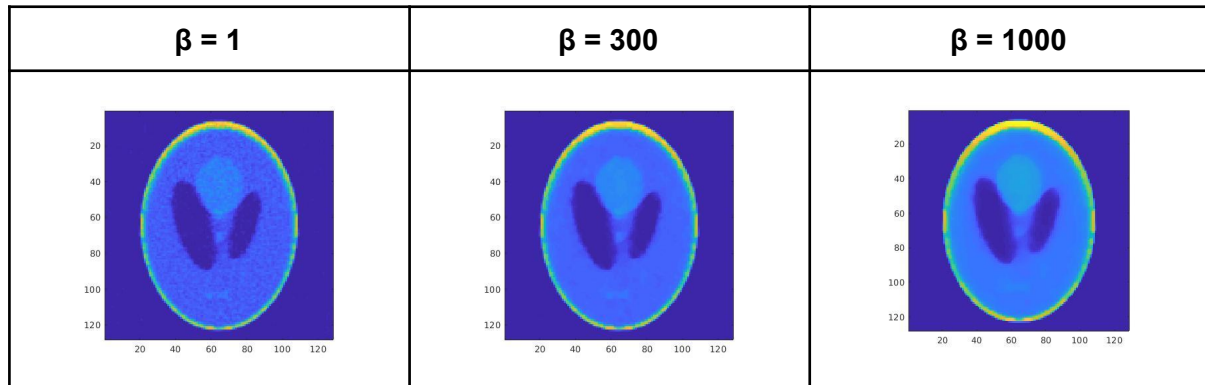
Question 15 :

Le tableau suivant montre l'effet de la régularisation quadratique sur la reconstruction de l'image, en faisant varier la prépondérance de cette régularisation avec le paramètre β .



Plus β est élevé, plus l'image tend à être uniformisée, et les contours deviennent moins nets. Cependant une valeur trop faible induit un bruit résiduel trop important.

Question 16 :

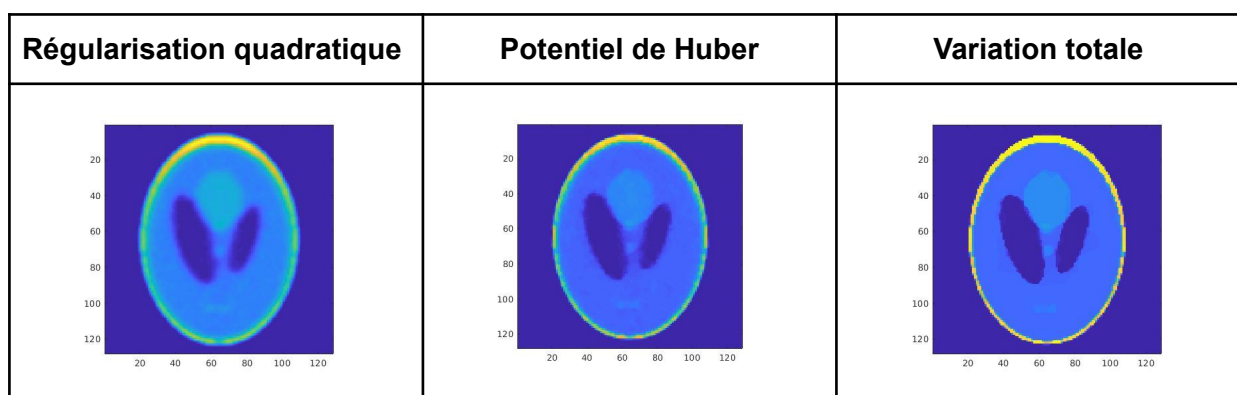


D'après le graph suivant permettant de comparer les différentes fonction de régularisation qui existent, on peut remarquer que le **potentiel de Huber** défini par morceaux, **pénalise moins les écarts de valeurs importants**, comparé à la fonction quadratique (variation linéaire à partir d'un certain seuil).

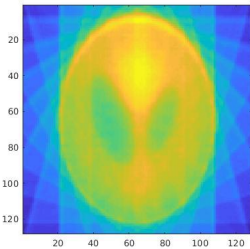
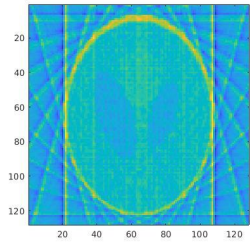
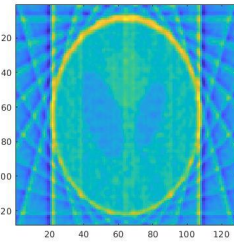
Ainsi les pixels de valeurs très différentes seront moins "pénalisées" et tendront moins à être uniformisées. Ainsi les formes correspondant à des zones de couleurs sont préservées.

Question 17 :

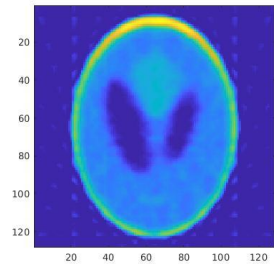
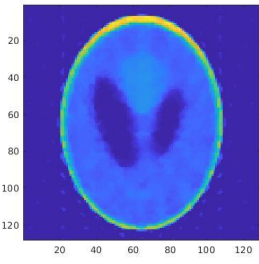
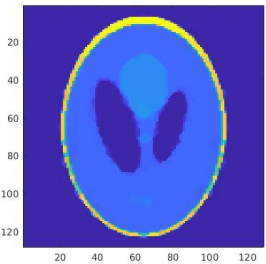
La parcimonie favorise les aplats de couleur, ce qui a pour effet de préserver les contours (gradient présent uniquement dans les zones de contours). En effet la pénalité est forte pour les petites variations, ce qui a pour effet de les uniformiser, et faible au-delà.



Question 18 :

Non Filter	Filtre Ram-Lak	Filtre Hann
		

Les reconstructions d'images obtenues sans régularisation présentent de très mauvaises définitions sans introduire de terme de régularisation (angles de vue apparents).

Régularisation quadratique	Potentiel de Huber	Variation totale
		

On peut observer que le manque d'information provoque des bruits avec les 3 méthodes de régularisation, mais ils sont moins exagérés avec total variation et plus exagérés avec la régularisation quadratique. **Ainsi la variation totale apparaît comme la meilleure fonction de régularisation dans le cas d'un système sous-déterminé.**