

UE MÉTHODES NUMÉRIQUES

M1 2A EEEA OPTION PHYSIQUE NUMÉRIQUE

Optimisation : Déconvolution d'images

Étudiants :

Mohamed Reda YACOUBI
Walid MOUJAHID

Enseignants :

Emmanuel SOUBIES

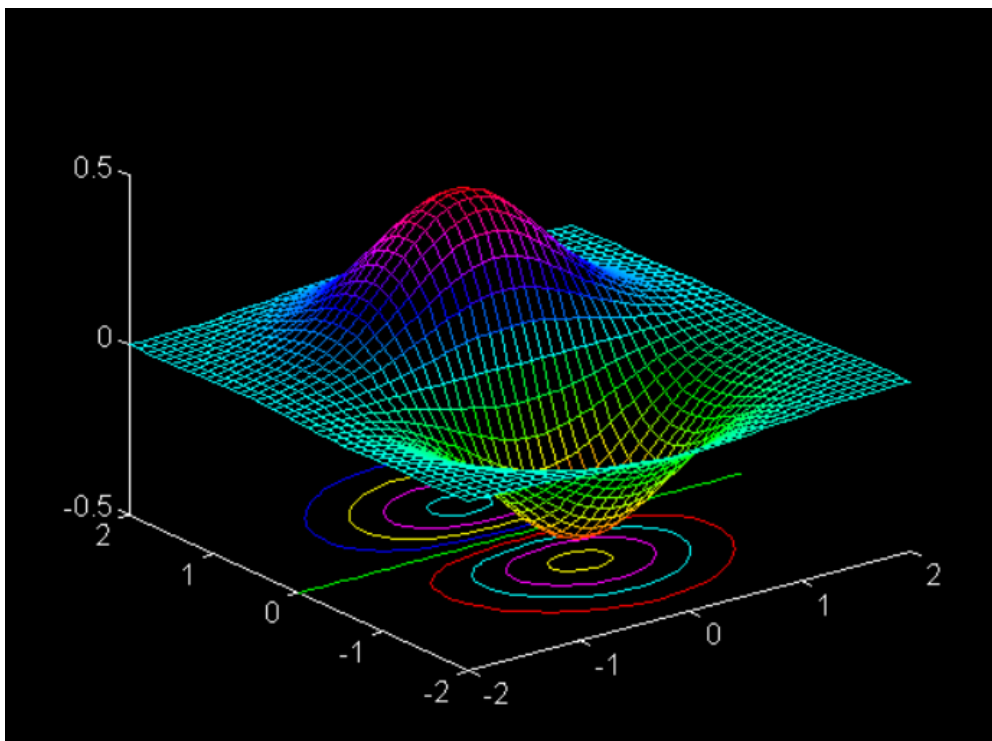




Table des matières

1	Introduction	2
2	Partie 1 : Préliminaire	2
3	Partie 2 : Solution des moindres carré	3
3.1	Problème moindre carrés	3
4	Partie 3 : Solution moindres carrés régularisés	6
4.1	Problème moindres carrés régularisés	6
4.2	Algorithme ADMM	8
5	Conclusion	12

1 Introduction

Actuellement, l'optimisation est un outil indispensable pour effectuer le traitement et la déconvolution des images. Cela fera donc l'objet de notre bureau d'étude, et on s'intéressera plus particulièrement au traitement d'une image qu'on note $y \in R^N$ et qui contient un certain nombre de pixels, et qui représentera la solution de convolution d'une image noté par $x \in R^N$ et le noyau qu'on note h .

On essayera dans ce projet de trouver l'image noté x à partir de l'image y et le noyau h .

Cela est modélisé par l'équation suivante :

$$h = y * x^* \quad (1.1)$$

2 Partie 1 : Préliminaire

Prenons l'exemple d'une image qu'on appelle Xstar et un noyau qu'on appelle h :

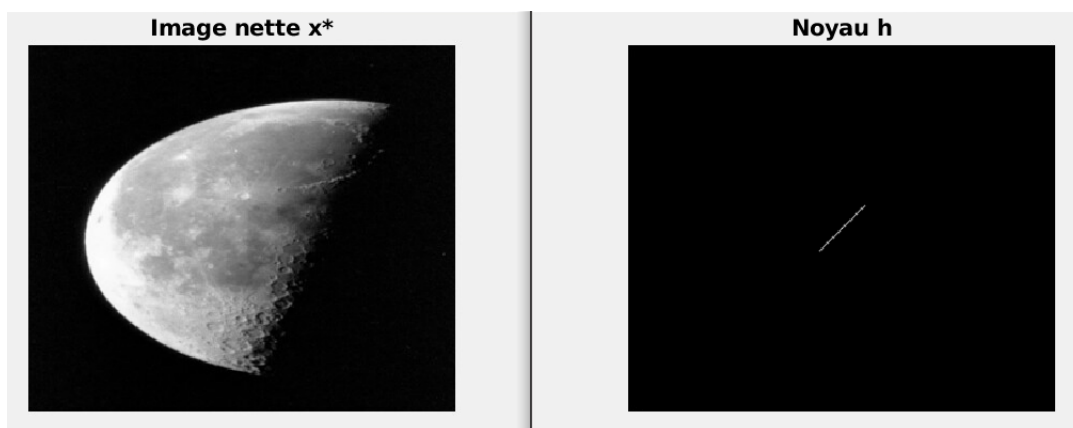


FIGURE 1 – L'image Xstar (à gauche) et le noyau h (à droite)

Nous obtiendrons le résultat de l'image floue y à travers de l'image x et le noyau h . Après avoir appliqué Fourier à l'équation précédente, on aura :

$$F(y) = F(x) \odot F(h) \quad (2.1)$$

Sous Matlab, on remarque qu'il est mieux de faire le calcul des produits élément par élément au lieu d'appliquer la convolution.

Après avoir calculé la fft (transformée de Fourier inverse), on obtient le résultat de l'image y .

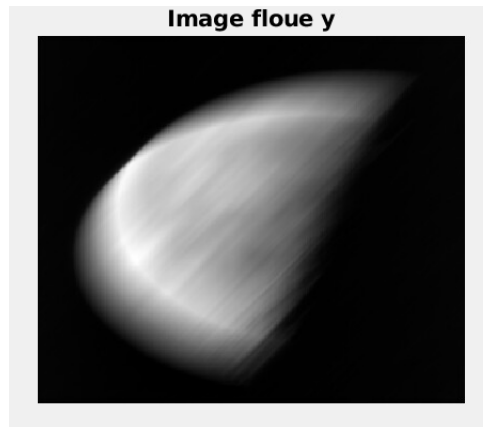


FIGURE 2 – Image floue y

3 Partie 2 : Solution des moindres carré

3.1 Problème moindre carrés

Dans cette partie, on doit retrouver l'image x^* à partir de y en résolvant le problème des moindres carrés :

$$\hat{x} \in \{\operatorname{argmin}_{x \in \mathbb{R}^N} \frac{1}{2} \|Hx - y\|^2\} \quad (3.1)$$

Avec H est telle que : $Hx = h * x$ la matrice associé à la convolution.

Une propriété importante de la convolution est que la transformée de Fourier d'une convolution correspond au produit des transformées de Fourier de chacun des termes. On a donc

$$Hx = h * x \Leftrightarrow Fy = Fh \odot Fx$$

Important : En observant que $\forall (w, x) \in \mathbb{R}^N \times \mathbb{R}^N$ on a : $w \odot x = \operatorname{diag}(w) \times x$ où $\operatorname{diag}(w)$ représente une matrice diagonale de taille $\mathbb{N} \times \mathbb{N}$ dont les éléments diagonaux sont les éléments du vecteur w).

$\forall (w, x) \in \mathbb{R}^N \times \mathbb{R}^N$ on a :

$$Fy = Fh \odot Fx = \operatorname{diag}(Fh) \times Fx$$

Les éléments diagonaux de $\operatorname{diag}(Fh)$ sont les éléments du vecteur Fh.

$$y = F^T (\operatorname{diag} Fh \times Fx)$$

$$y = F^T D F x$$

La diagonalisation de H est telle que :

$$H = F^T \times \text{diag}(Fh) \times F$$

Avec $D = \text{diag}(Fh)$.

Comme on a vu dans le TD, pour qu'un problème de moindres carrés admette une unique solution, il suffit que $\ker(H) = \{0\}$

Or, la matrice H est inversible car $\det(H) \neq 0$ (Tous les éléments de Fh sont non nuls).

Et donc la condition d'optimalité est donnée par : $H^T H \hat{x} = H^T y$.

On a l'expression de $H = F^T \times \text{diag}(Fh) \times F$ Et :

$$\begin{aligned} H^T H &= F^T \text{diag}(Fh)^T F F^T \text{diag}(Fh) F = F^T \text{diag}(Fh)^T \text{diag}(Fh) F \\ H^T y &= F^T \text{diag}(Fh)^T F F^T y = F^T \text{diag}(Fh)^T y \end{aligned}$$

Avec la condition d'optimalité : $H^T H \hat{x} = H^T y$, on trouve donc :

$$\hat{x} = F^T [\text{diag}(Fh)]^{-1} F y \quad (3.2)$$

Sous Matlab, on trouve une image identique à celle de l'image nette \hat{x} :

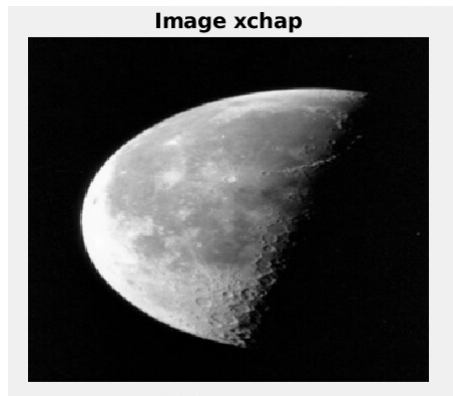


FIGURE 3 – Image \hat{x}

En pratique, l'image x n'est pas seulement dégradée par un flou, mais aussi par un bruit additif $n \in \mathbb{R}^N$. Le modèle (1.1) devient donc :

$$y = h * x^* + n \quad (3.3)$$

L'image dans le fichier "image2bruit" est le résultat du modèle ci-dessus avec un x^* différent du précédent (et qui ne vous est pas donné ici) et un bruit Gaussien.

Sous matlab, l'affichage de l'image "image2 bruit" donne :

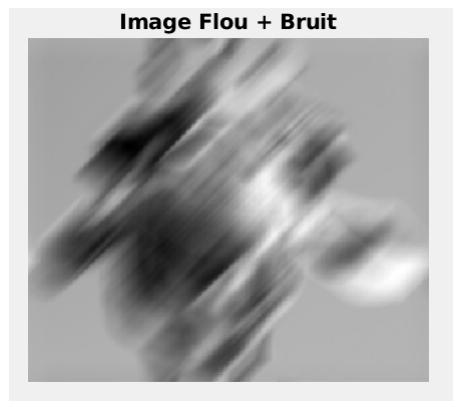


FIGURE 4 – Image flou + Bruit

L'image est donc bruitée et floutée.

La méthode des moindres carrés donne :

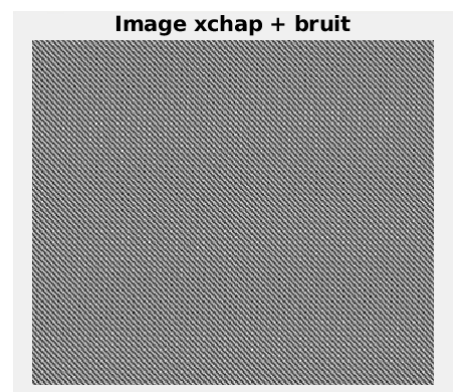


FIGURE 5 – Image \hat{x} + Bruit

On remarque que cette méthode des moindres carrées n'est plus valable . Et donc pour résoudre un tel genre de problème avec des images bruitées on va appliquer la méthode des moindres carrés régularisés.

4 Partie 3 : Solution moindres carrés régularisés

On va s'intéresser dorénavant au problème des moindres carrés afin de fixer le problème précédent

4.1 Problème moindres carrés régularisés

$$\hat{x} \in \{\operatorname{argmin}_{x \in R^N} [\frac{1}{2} \|Hx - y\|^2 + \frac{\lambda}{2} \|\nabla x\|^2]\} \quad (4.1)$$

où $\lambda \succ 0$ est un paramètre de pondérant l'importance du terme de régularisation et $\nabla : R^N$ dans R^{2N} désigne l'opérateur gradient.

On visualise l'effet de l'opérateur ∇ et on l'applique à l'image x , on obtient :

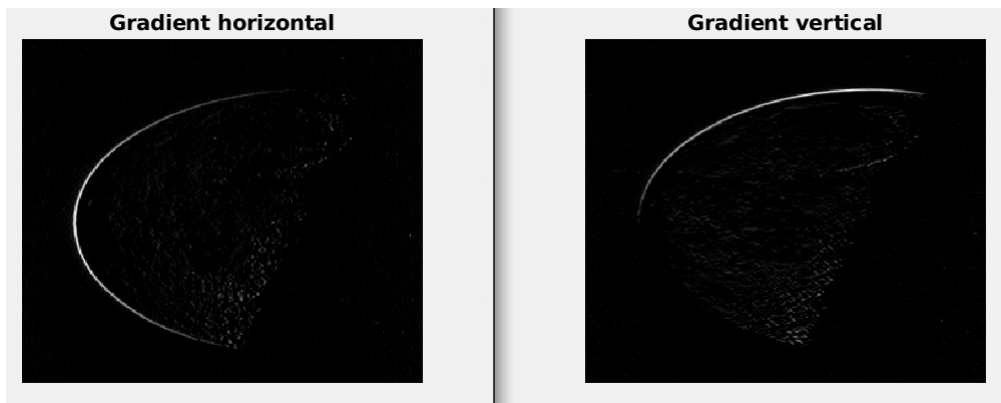


FIGURE 6 – l'effet de l'opérateur ∇

Fixons $\lambda = 10^{-3}$

On va utiliser le gradient accéléré afin de trouver la valeur du minimum de la fonction mise en oeuvre, et on affichera sous matlab la fonction $f(x_k)$ qui dépend de k , et on aura donc :

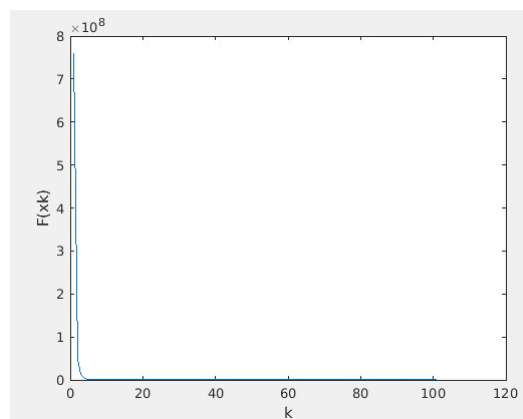


FIGURE 7 – l'effet de l'opérateur ∇

Remarques :

L'algorithme de gradient descendant converge rapidement (moins de 10 itération) d'où son nom. Même s'il converge rapidement, l'algorithme en question prend beaucoup de temps pour faire les calcul (beaucoup cher).

On remarque une convergence rapide de l'algorithme gradient descendant, par par contre, le temps d'exécution est grand pour le calcul.

Dans cette partie on va modifier le facteur et on observe l'influence de ce facteur sur l'image

On essayera dans cette partie de changer le facteur afin de voir son impact dans le résultat de l'image

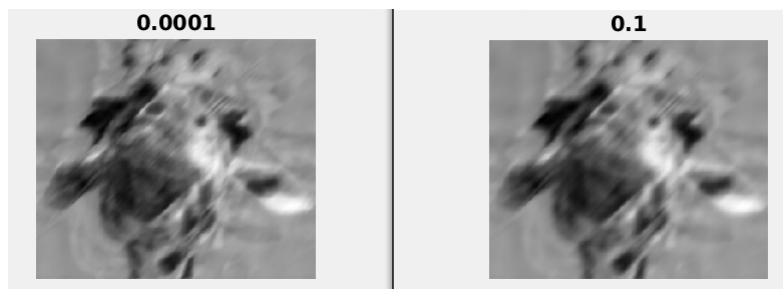


FIGURE 8 – $\lambda = 0.0001$ et $\lambda = 0.1$



FIGURE 9 – $\lambda = 1$ et $\lambda = 0.001$

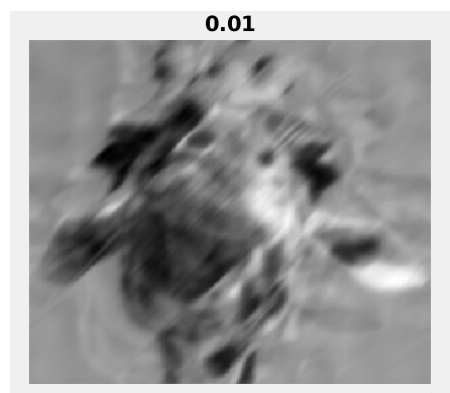


FIGURE 10 – $\lambda = 0.01$

L'augmentation de la valeur de λ implique la diminution de la qualité.

Pour la Variation totale, si on prend une régularisation autre que celle qu'on a vu, on trouve :

$$\hat{x} \in \{\operatorname{argmin}_{x \in \mathbb{R}^N} [\frac{1}{2} \|Hx - y\|^2 + \frac{\lambda}{2} \|\nabla x\|_1^2]\} \quad (4.2)$$

$$\|x\|_1 = \left| \sum_{i=1}^N x_i \right| \quad (4.3)$$

Il est impossible d'utiliser l'algorithme de descente de gradient comme dans la partie précédente puisque il y a une non-différentiabilité de la fonction.

Il est possible d'utiliser l'algorithme de descente de gradient proximale puisque la somme d'une fonction différentiable, un gradient Lipchitzien et un convexe est une fonction convexe.

— Il est possible d'utiliser l'ADMM puisque on a une somme de deux fonctions continues convexes qui est aussi convexe

4.2 Algorithme ADMM

On a :

$$\begin{aligned} f : x &\rightarrow \frac{1}{2} \|Hx - y\|^2 + \frac{\lambda}{2} \|\nabla x\|^2 \\ f : x &\rightarrow f_0(x) + f_1(Ax) \end{aligned}$$

les fonction f_0 et f_1 sont convexe :

$$\begin{aligned} f_0(x) &= \frac{1}{2} \|Hx - y\|^2 \\ f_1 &= \frac{\lambda}{2} \|\cdot\| \end{aligned}$$

Le proximal de la fonction f_1 est donnée par :

$$\operatorname{prox}_{\gamma \|\cdot\|_1}(x_i) = \operatorname{signe}(x_i) \max(|x_i| - \gamma) \quad (4.4)$$

En effet, on a :

$$\operatorname{prox}_{\gamma \|\cdot\|_1}(x_i) = \{\operatorname{argmin}_{\nu} [\frac{1}{2} \|\nu - x_i\|^2 + \gamma |x_i|]\} \quad (4.5)$$

Pour trouver la solution de ce problème il suffit d'étudier la fonction réelle avec la séparation des cas $v > 0$ et $v < 0$ en utilisant l'algorithme de ADMM suivant :

Algorithm 3 ADMM

Require: $\mathbf{x}^0 \in \mathbb{R}^N$, $\rho > 0$,
1: $\mathbf{v}_q^0 = \mathbf{A}_q \mathbf{x}^0$, $\forall q \in \{1, \dots, Q\}$
2: $\lambda_q^0 = \mathbf{v}_q^0$, $\forall q \in \{1, \dots, Q\}$
3: $k = 0$
4: **while** (Critère d'arrêt pas satisfait) **do**
5: $\mathbf{v}_q^{k+1} = \text{prox}_{\frac{1}{\rho} f_q} \left(\mathbf{A}_q \mathbf{x}^k + \frac{\lambda_q^k}{\rho} \right)$, $\forall q \in \{1, \dots, Q\}$
6: $\mathbf{x}^{k+1} = \arg \min_{\mathbf{x} \in \mathbb{R}^N} f_0(\mathbf{x}) + \sum_{q=1}^Q \frac{\rho}{2} \left\| \mathbf{A}_q \mathbf{x} - \mathbf{v}_q^{k+1} + \frac{\lambda_q^k}{\rho} \right\|^2$
7: $\lambda_q^{k+1} = \lambda_q^k + \rho(\mathbf{A}_q \mathbf{x}^{k+1} - \mathbf{v}_q^{k+1})$, $\forall q \in \{1, \dots, Q\}$
8: $k \leftarrow k + 1$
9: **end while**

FIGURE 11 – Algorithme de ADMM

Pour implémenter cet algorithme sous Matlab on doit expliciter la ligne en colorée en jaune. Pour notre cas $Q=1$. On a le problème linéaire suivant :

$$\left\{ \underset{x \in \mathbb{R}^N}{\text{argmin}} \left[\frac{1}{2} \|Hx - y\|^2 + \frac{\rho}{2} \|Ax - \nu + \frac{\lambda}{\rho}\|^2 \right] \right\} \quad (4.6)$$

Avec : $v = \text{prox}$

La condition de optimalité du premier ordre : $\text{grad}(f(\hat{x})) = 0$

Ce qui equivalent à dire :

$$H^T(H\hat{x} - y) + \rho \nabla^T(\nabla \hat{x} - \nu + \frac{\lambda}{\rho}) = 0$$

$$H^T H \hat{x} - H^T y + \rho \nabla^T \nabla \hat{x} - \rho \nabla^T \nu + \rho \nabla^T \frac{\lambda}{\rho} = 0$$

$$(H^T H + \rho \nabla^T \nabla) \hat{x} = H^T y + \rho \nabla^T \nu - \rho \nabla^T \frac{\lambda}{\rho}$$

Explicitant le terme : $(H^T H + \rho \nabla^T \nabla)$ D'après l'hypothèse $\nabla^T \nabla = F^T D_{nabla} F$

F et d'après la première partie :

$$H^T H = F^T D_H F$$

Alors :

$$F^T (D_H + \rho D_{nabla}) = H^T y + \rho \nabla^T \nu - \rho \nabla^T \frac{\lambda}{\rho}$$

D'où :

$$\hat{x} = F^T (D_H + \rho D_{nabla})^{-1} F (H^T y + \rho \nabla^T (\nu - \frac{\lambda}{\rho}))$$

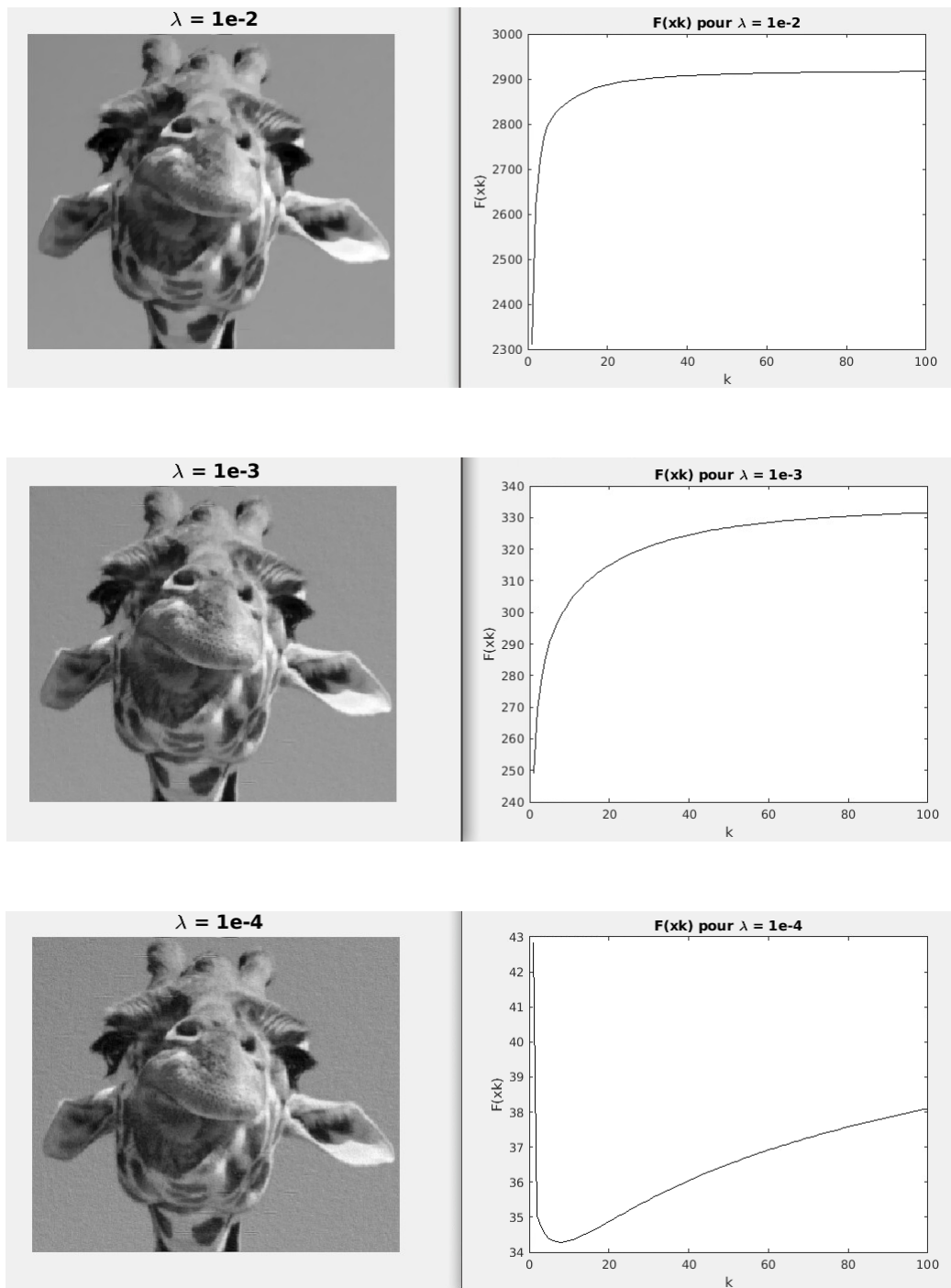


FIGURE 12 – L'image avec $\rho = 1e - 3$ fixe

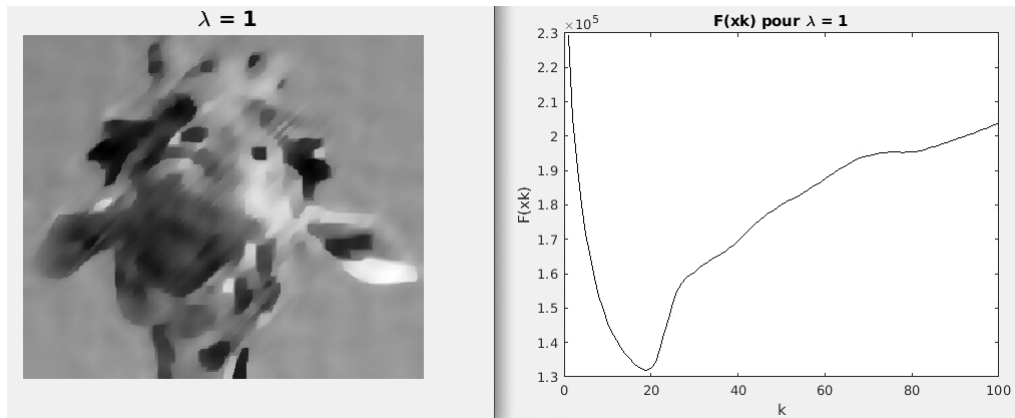


FIGURE 13 – L'image avec $\rho = 1e - 3$ fixe

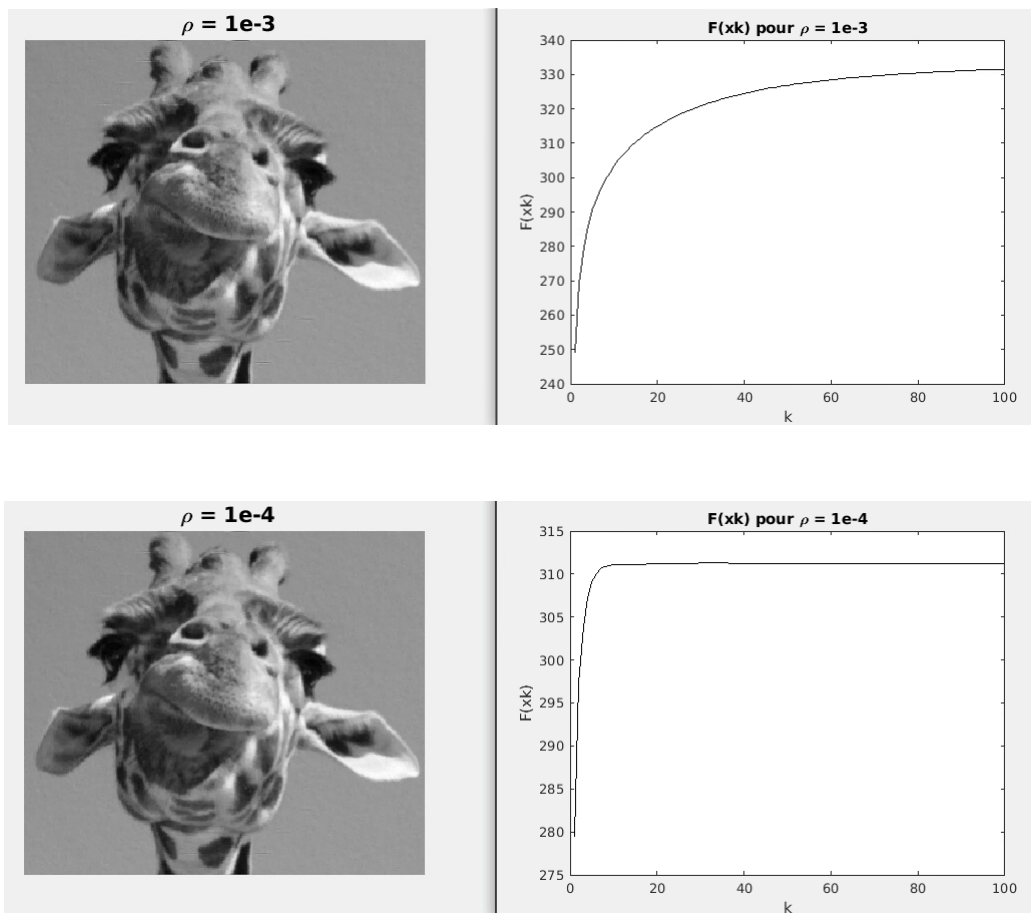


FIGURE 14 – L'image avec $\lambda = 1e - 3$

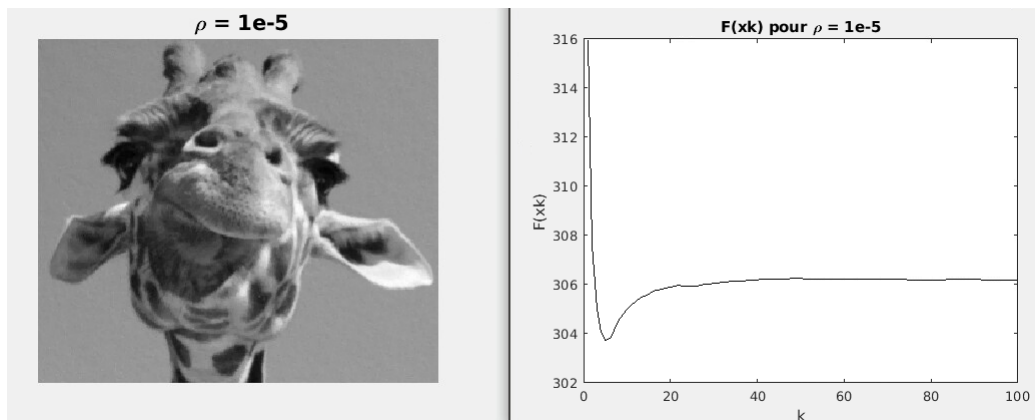


FIGURE 15 – L'image avec $\lambda = 1e - 3$

Contrairement à la partie précédente, ici si on augmente la qualité devient mieux, par contre au paramètre ρ .

Finalement la méthode de ADMM est la méthode qui donne le meilleur résultat en terme de qualité de l'image, pourtant la méthode nécessite beaucoup de temps pour implémenter et elle coûte cher en terme de nombre d'itérations et de stockage.

5 Conclusion

Ce bureau d'étude était pour nous une opportunité pour appliquer la déconvolution des images qui représente la base de l'optimisation, et on a réussi à mettre en pratique les méthodes de gradient descendant, proximale et ADMM afin d'avoir les résultats attendus des images, et les comparer à l'aide des différentes régularisations mise en oeuvre.

Fin du rapport