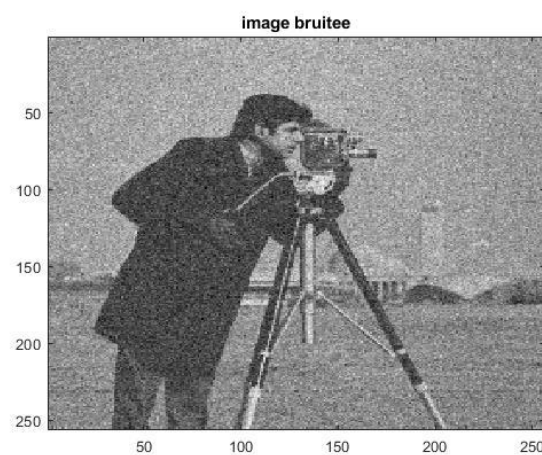


# TP n°2 : Optimisation convexe non-lisse



# Sommaire :

Introduction

Influence du paramètre  $\lambda$

Démonstration du problème dual

Résolution du problème dual à l'aide de l'algorithme explicite-implicite

Tracer du critère au cours des itérations

Tracer de l'évolution de l'erreur quadratique

Le but de ce TP n'est pas de reproduire ce que nous avons réalisé auparavant, mais d'améliorer la qualité de reconstruction par une approche de type Tikhonov en incorporant une pénalisation non-différentiable (contrairement au TP n°1), c'est-à-dire en résolvant un problème dual à l'aide de l'algorithme explicite-implicite.

On considère  $\bar{x} \in \mathbb{R}^N$  la représentation vectorielle de l'image originale, et  $y = \bar{x} + \varepsilon$  la représentation vectorielle de l'image dégradée avec  $\varepsilon$  un bruit additif Gaussien tel que  $\varepsilon \sim N(0, \sigma^2) = N(0, 1)$ .

Le but est d'estimer une image  $\hat{x}$  au plus proche de  $\bar{x}$  à partir des observations de  $y$ . Cela revient donc à minimiser le critère suivant :

$$\hat{x}_\lambda = \arg \min_{x \in \mathbb{R}^N} \frac{1}{2} \|x - y\|_2^2 + \lambda \|Lx\| \quad (1)$$

Où  $L = [H^T, V^T]^T$  avec  $H$  et  $V$  les opérateurs de différences finies horizontal et vertical et  $\lambda$  le paramètre de régularisation.

Dans ce TP des codes de fonctions nous sont déjà fournis. Si on observe la fonction `op_reg.m` appliqué à notre image *Camerman\_bruit*, on s'aperçoit que celle-ci nous retourne le gradient de l'image et permet d'isoler les contours grâce aux variations de pixels.

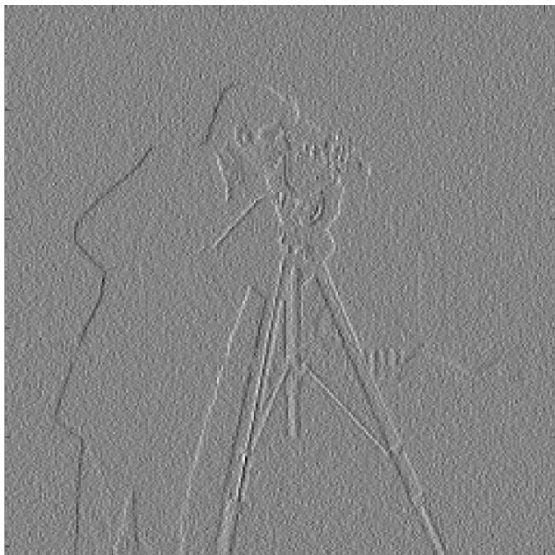


Figure 1 `op_reg` sur l'image l : z1

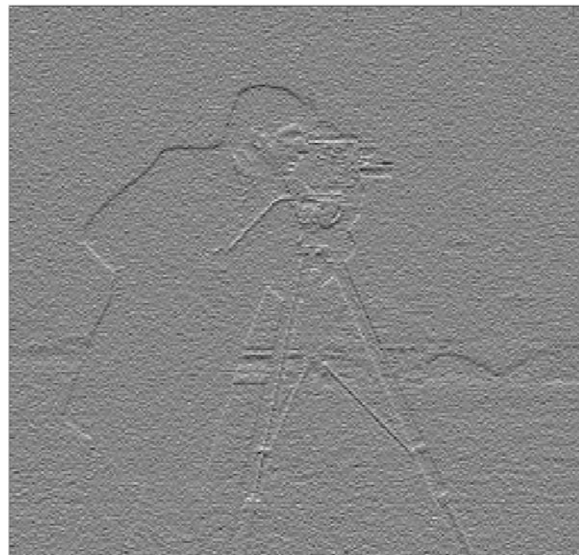


Figure 2 `op_reg` sur l'image l : z2

Le code de la fonction `op_reg` permet de faire une transformé dans un autre domaine.

### Influence du paramètre $\lambda$

Si  $\lambda$  est proche de 0, alors  $\hat{x}_\lambda = \arg \min_{x \in \mathbb{R}^N} \frac{1}{2} \|x - y\|_2^2$ . On constate alors que l'estimation converge vers l'image bruitée.

Cette fois-ci, si  $\lambda \gg 0$ , alors  $\hat{x}_\lambda = \arg \min_{x \in \mathbb{R}^N} \lambda \|Lx\|$ . L'estimation converge donc vers une image d'autant plus floue que  $\lambda$  est grand. Toute l'information contenue dans  $y$  est alors perdue. Par ailleurs, la représentation vectorielle de l'image dégradée  $y$  dépend également du bruit gaussien, plus précisément de sa variance. Ainsi, plus celle-ci est grande, plus  $\lambda$  devra être important.

## Dualité

Si on pose :

$$\begin{aligned} f: \mathbb{R}^N &\rightarrow ]-\infty, +\infty] & \text{et} & & g: \mathbb{R}^M &\rightarrow ]-\infty, +\infty \\ x &\rightarrow \frac{1}{2} \|x - y\|_2^2 & & & x &\rightarrow \lambda \|x\| \end{aligned}$$

Nous cherchons à exprimer le problème dual associé à l'équation (1).

Or,  $f \in \Gamma_0(\mathbb{R}^N)$ ,  $g \in \Gamma_0(\mathbb{R}^M)$  et  $L \in \mathbb{R}^{M \times N}$ . Donc d'après le théorème de Fenchel-Rockafellar, nous avons :

$$\min_{x \in \mathbb{R}^N} f(x) + g(Lx) = - \min_{v \in \mathbb{R}^N} f^*(L^*v) + g^*(v)$$

Nous devons alors déterminer  $f^*$  et  $g^*$ .

Par définition, nous avons  $(x, v \in \mathbb{R}^N), f^*(v) = \sup(\langle x|u \rangle - f(x - y))$

$$\begin{aligned} &= \sup\left(\frac{1}{2} \|x\|_2^2 + \frac{1}{2} \|v\|_2^2 - \frac{1}{2} \|x - v\|_2^2 - \frac{1}{2} \|x\|_2^2 - \frac{1}{2} \|y\|_2^2 + \langle x|y \rangle\right) \\ &= \sup\left(\frac{1}{2} \|v\|_2^2 - \frac{1}{2} \|x - v\|_2^2 - \frac{1}{2} \|y\|_2^2 + \langle x|y \rangle\right) \\ &= \sup\left(\frac{1}{2} \|v\|_2^2 - \frac{1}{2} \|x - v\|_2^2 - \frac{1}{2} \|y\|_2^2 + \frac{1}{2} \|x + y\|_2^2 - \frac{1}{2} \|x\|_2^2 - \frac{1}{2} \|y\|_2^2\right) \\ &= \sup\left(\frac{1}{2} \|v\|_2^2 - \frac{1}{2} \|x - v\|_2^2 + \frac{1}{2} \|x + y\|_2^2 - \frac{1}{2} \|x\|_2^2\right) \end{aligned}$$

On remarque alors que le sup est atteint pour  $x = v$  donc  $f^*(v) = \frac{1}{2} \|v + y\|_2^2$ .

Par définition de la fonction d'appui de  $C = \{x \in \mathbb{R}^N \mid \|x\|_\infty < \lambda\} \subset \mathbb{R}^N$ ,  $g = \sigma_C = i_C^*$ .

Finalement, d'après le théorème de Moreau-Fenchel, comme  $g \in \Gamma_0(\mathbb{R}^M) \Leftrightarrow g^{**} = g$ , nous avons alors  $g^* = i_C$ .

On a donc le problème dual suivant :

$$\hat{v}_\lambda = \arg \min_{v \in \mathbb{R}^N} \frac{1}{2} \|L^*v + y\|_2^2 + i_C(v) \text{ s.t. } \|v\|_\infty < \lambda$$

Par ailleurs, comme  $\min(i_C) = 0$ , le problème dual devient :

$$\hat{v}_\lambda = \arg \min_{v \in \mathbb{R}^N} \frac{1}{2} \|L^*v + y\|_2^2 \text{ s.t. } \|v\|_\infty < \lambda \quad (2)$$

Finalement,  $f^*$  étant différentiable en  $L^*\hat{v}_\lambda$ , les solutions du problème primal (1) et du problème dual (2) sont reliées par la relation :  $\hat{x}_\lambda = \nabla f^*(L^*\hat{v}_\lambda)$  avec  $\nabla f^*(v) = (v + y)$ .

Donc on a bien la relation :  $\hat{x}_\lambda = L^*\hat{v}_\lambda + y$ .

## Résolution du problème dual à l'aide de l'algorithme explicite-implicite

Soit le cas du problème dual :  $f^* : \mathbb{R}^N \rightarrow [-\infty, +\infty]$ ,  $\nabla f^*(v) = L(L^*v + y)$ .  

$$v \rightarrow \frac{1}{2} \|L^*v + y\|_2^2$$

On a,  $\forall u, v \in \mathbb{R}^N$ ,  $\|\nabla f^*(u) - \nabla f^*(v)\| = \|LL^*\| \|u - v\|$ . Ainsi,  $f^*$  est donc de gradient Lipchitz de constante  $\beta = \|LL^*\|$ . Par ailleurs, comme  $L$  et  $L^*$  sont normalisées,  $\beta = 1$ .

Soit  $f \in \Gamma_0(\mathbb{R}^N)$ , et  $f^*$  son conjugué. Alors l'opérateur proximal relie  $f$  et  $f^*$  est défini par la relation suivante :  $(\forall x \in \mathbb{R}^N)$ ,  $prox_{\gamma f^*}(x) = x - \gamma prox_{\gamma^{-1}f}(\gamma^{-1}x)$  (3).

On obtient ensuite le proximal de la norme infinie :  $prox_{\gamma \|\cdot\|_\infty}(x) = x - \gamma prox_{\gamma^{-1}\|\cdot\|}(\gamma^{-1}x)$

On peut donc implémenter l'algorithme explicite-implicite, d'après le cours, comme :

$$x^{(k+1)} = (1 - \text{lambda})x^{(k)} + \text{lambda} * prox_{\gamma f^*}(x^{(k)} - \gamma \nabla g^*(x^{(k)}))$$

Avec :

$(1 - \text{lambda})$  et  $\text{lambda}$  le point  $\lambda_k$  sur le segment  $[x_k, prox_{\gamma f^*}(x^{(k)} - \gamma \nabla g^*(x^{(k)}))]$

$prox_{\gamma f^*}$  la descente de grad de  $F$  de pas  $\gamma$

$(x^{(k)} - \gamma \nabla g^*(x^{(k)}))$  la descente de grad de  $g$  de pas  $\gamma$

## Implémentation de l'algorithme explicite-implicite

Pour implémenter l'algorithme nous allons avoir besoin de l'expression de la fonction différentiable de gradient Lipschitz qui s'écrit :  $\frac{1}{2} \|y + L^*u\|^2$ .

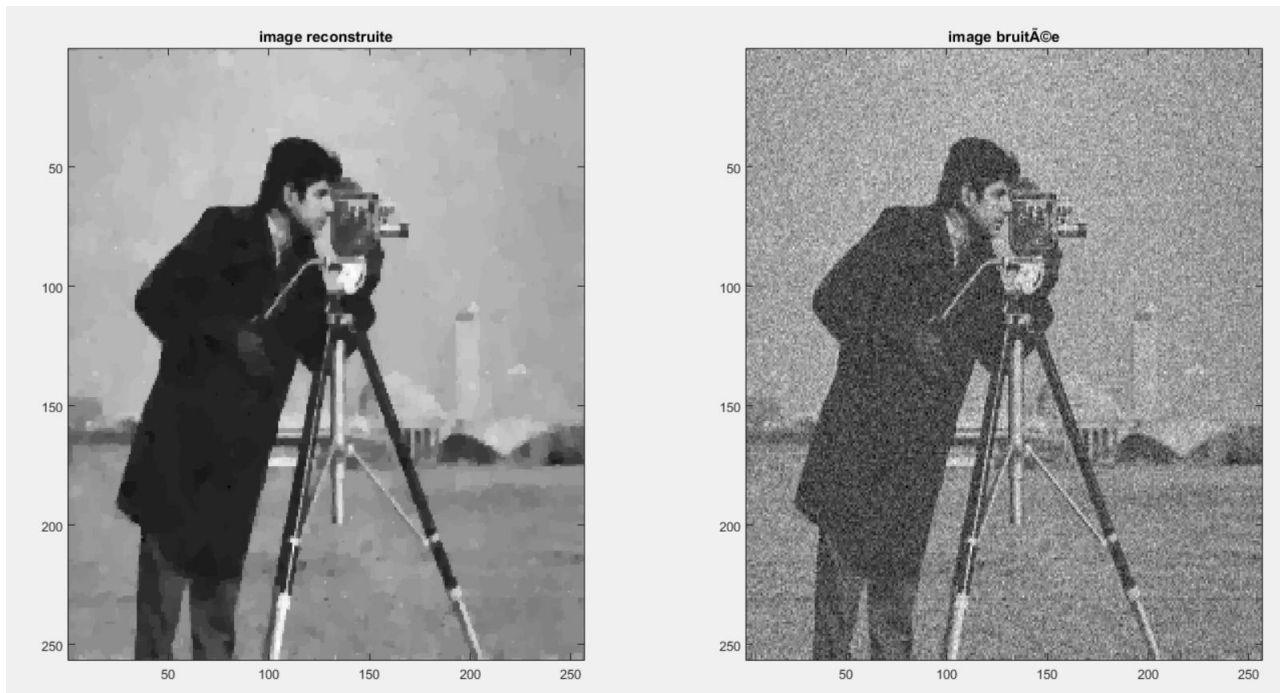
De cette expression nous pouvons en déduire son gradient en la dérivant :  $L(L^*(u) + y)$

Pour calculer  $prox_{\gamma f^*}$  nous utiliserons le code `prox_L1` qui permet le calcul de la fonction  $prox_{\gamma^{-1}f}$ . Ainsi nous pouvons écrire la fonction (3)

En appliquant cette formule avec les fonctions `op_reg` et `op_reg_adj` nous allons obtenir deux gradients. Ces fonctions permettent de simuler l'opérateur de différence finie et l'opérateur de différence finie adjoint.

Il faudra ensuite appliquer l'algorithme pour chaque gradient et ensuite appliquer la fonction `op_reg_adj` qui nous donne une variable en sortie qui correspondra à notre image traitée. Il faudra ensuite transformer l'image obtenue dans le domaine dual en se basant sur la règle de Fermat et de la démonstration ci-dessus :  $\hat{x}_\lambda = L^*\hat{v}_\lambda + y$ .

En appliquant cet algorithme, on obtient des résultats plutôt satisfaisant comme le montre les images ci-dessous :



On remarque que l'image est plutôt bien reconstruite, en agissant sur les paramètres  $\lambda$  et  $\gamma$  nous pouvons obtenir des résultats différents mais si  $\lambda$  est trop grand l'image devient plus floue et s'il est trop petit il reste trop de bruit.

Code Matlab associé :

```
N=100;
lambda=40;
nu=lambda;
gamma=1.1/nu;

prox_F = @(x) x - gamma.*prox_L1(x/gamma,lambda/gamma);
gammaX=1;
lambdaX=1;
X1=I;
X2=I;

for k=1:N

    [grad1,grad2] = op_reg(op_reg_adj(X1,X2)+I);

    X1 = (1-lambdaX) * X1 + lambdaX*prox_F(X1-gammaX*grad1);
    X2 = (1-lambdaX) * X2 + lambdaX*prox_F(X2-gammaX*grad2);
end

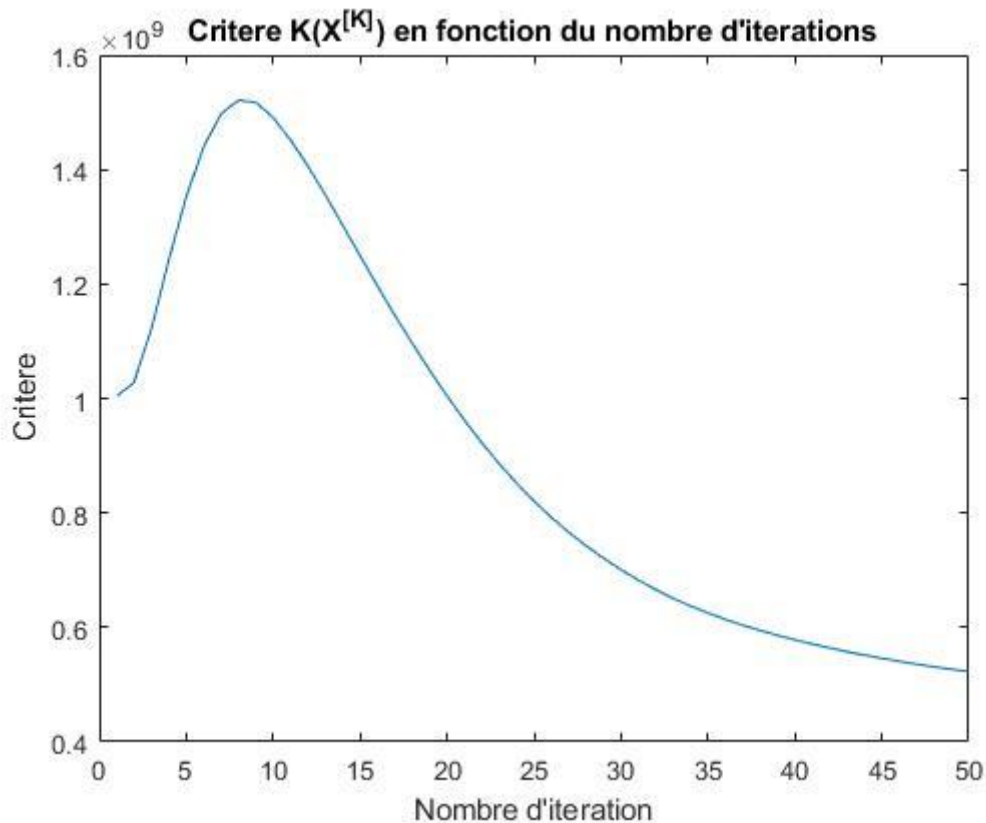
X= op_reg_adj(X1,X2);
u = I + X;

figure();
subplot(1,2,1)
imagesc(u);
colormap('gray');
title('image reconstruite');

subplot(1,2,2)
imagesc(I);
colormap('gray');
title('image bruitée');
```

### Vérification du critère au cours des itérations

Nous décidons maintenant d'afficher le critère  $K^{[x]}$  permettant de vérifier la validité de notre implémentation. Nous obtenons la courbe suivante :



Avec le code suivant :

```
K=[];
lambdaX=0,2 ;
for k=1:N

    [grad1,grad2] = op_reg(op_reg_adj(X1,X2)+I);

    X1 = (1-lambdaX) * X1 + lambdaX*prox_F(X1-gammaX*grad1);
    X2 = (1-lambdaX) * X2 + lambdaX*prox_F(X2-gammaX*grad2);
    K=[K 0.5*norm(op_reg_adj(X1,X2)).^2+lambda.*norm(op_reg(I+op_reg_adj(X1,X2)),1).^2];
end

figure()
plot([1:N],K),title('Critère K(X^[K]) en fonction du nombre d''itérations')
```

Nous constatons alors bien que pour un lambdaX de 0,2 notre critère est maximal à la 8<sup>ième</sup> itérations et au fur et à mesure des itérations le critère décroît. Ceci est normal de par sa définition par la formule

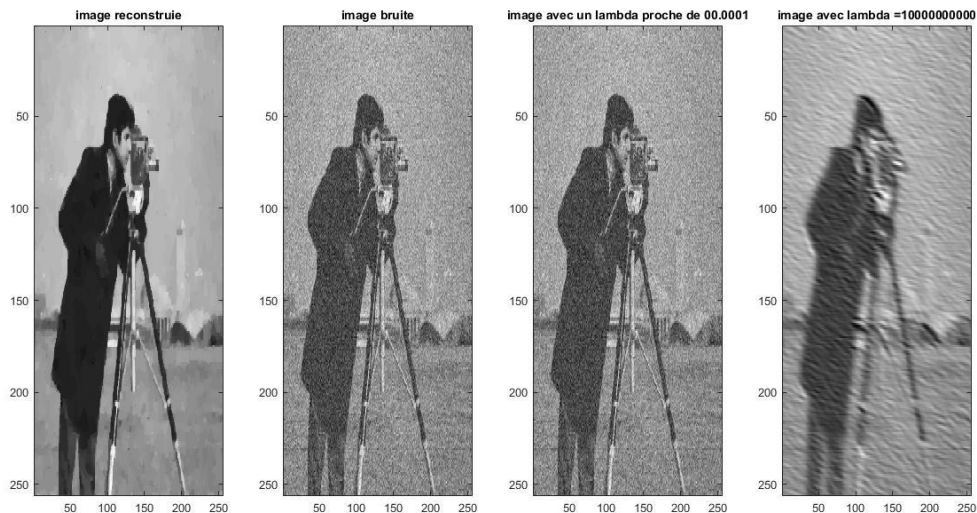
$$K(x) = \frac{1}{2} \|x - y\|_2^2 + \lambda \|Lx\|_1.$$

Ce critère montre alors bien la validité de l'implémentation utilisée.



## Influence de $\lambda$ sur l'erreur quadratique

Nous allons dorénavant montrer l'influence de  $\lambda$  sur l'évolution de l'erreur quadratique entre  $\bar{x}$  et  $\hat{x}_\lambda$ . Afin de s'assurer que l'image n'évolue plus, nous travaillons sur 1000 itérations. Cherchons d'abord à visualiser l'influence du paramètre  $\lambda$ .



Avec le code suivant :

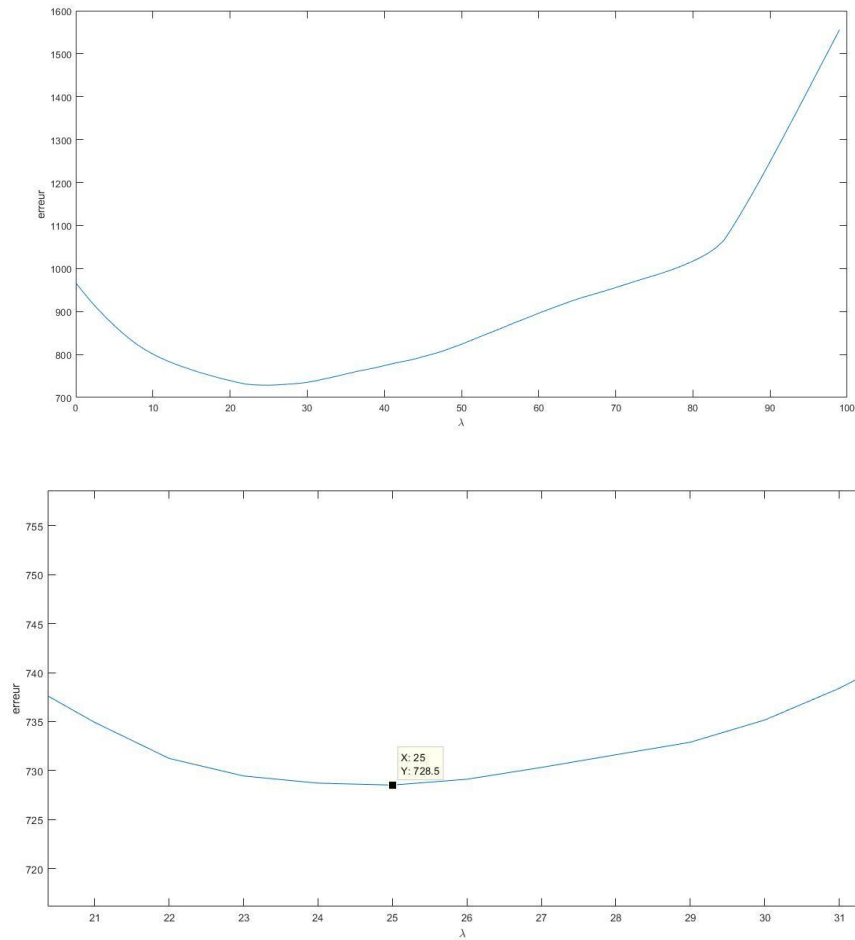
```
%% Influence du parametre lambda
lambdaFaible=0.0001;
lambdaFort=1e10;
uFaible=zeros(256,256);
uFort=zeros(256,256);
nbrIteration=1000;
for i=1:nbrIteration
    gradFaible=uFaible-gamma*op_reg(op_reg_adj(uFaible, uFaible)+I);
    uFaible=gradFaible-gamma*prox_L1(gradFaible/gamma,lambdaFaible/gamma);
    gradFort=uFort-gamma*op_reg(op_reg_adj(uFort, uFort)+I);
    uFort=gradFort-gamma*prox_L1(gradFort/gamma,lambdaFort/gamma);
end

figure;
hold on
colormap('gray');
title('Mise en évidence du paramètre lambda');
subplot(1,4,1);
imagesc(u);
title('image reconstruite');
subplot(1,4,2);
imagesc(I);
title('image bruitée');
subplot(1,4,3);
imagesc(I+op_reg_adj(uFaible, uFaible));
title(['image avec un lambda proche de 0',num2str(lambdaFaible)]);
subplot(1,4,4);
imagesc(I+op_reg_adj(uFort, uFort));
title(['image avec lambda =',num2str(lambdaFort)]);
```

Ainsi, on peut constater ce qui a été dit précédemment sur le paramètre  $\lambda$ , c'est-à-dire que plus  $\lambda$  est proche de 0, plus l'image estimée est confondue avec l'image bruitée. Et plus  $\lambda$  est grand, plus l'image devient floue, perdant tous les détails de l'image, cependant le bruit n'a plus d'influence.

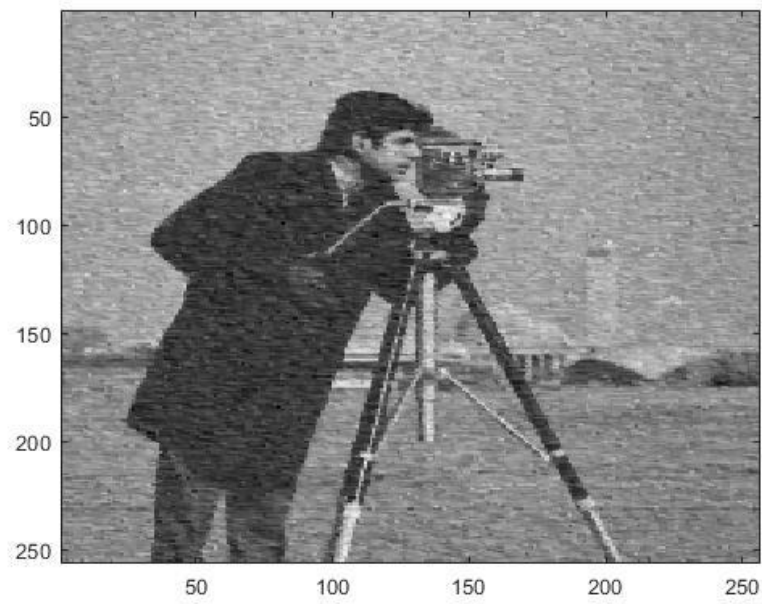


On cherche alors la valeur optimale de  $\lambda$  pour trouver un compromis. Pour ce faire, nous traçons l'erreur quadratique. On obtient la figure suivante :



On constate que la courbe représentant l'erreur quadratique n'est pas lisse. Néanmoins, on peut remarquer que l'erreur quadratique passe par un minimum. En effectuant un zoom sur ce minimum, on peut déterminer la valeur  $\lambda_{opt}$ . Dans notre cas, nous obtenons  $\lambda_{opt} = 25$ .

Pour cette valeur, on obtient l'image reconstruite suivante :



L'image reconstruite ne se confond pas totalement avec l'image originale mais se rapproche au mieux.  $\lambda_{opt} = 25$  permet de récupérer un maximum d'informations à partir de l'image bruitée.