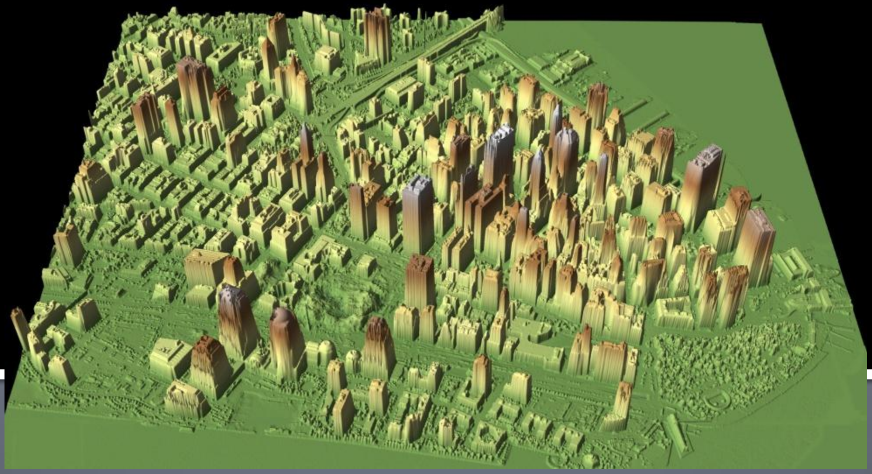
Génie Mathématique et Modélisation
4ème année Méthodes et Modèles Numériques

Modélisation, simulation et restitution d'images 3D d'objets : radar laser imageur

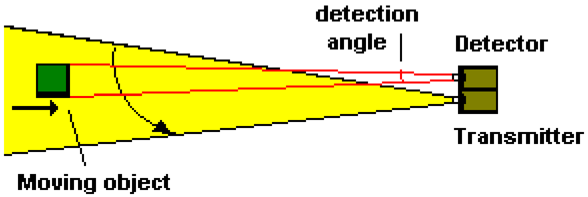


Marie Bonnasse-Gahot
Rémi Chauvin

Date de soutenance :
Mercredi 8 juin 2011

Jacques Isbert
Pierre Weiss

Ladar



Plan

I. Bilan radiométrique

II. Analyse temporelle

III. Construction du simulateur

IV. Reconstruction de l'image 3D

V. Tests numériques

I. Bilan radiométrique

- Mesure l’énergie transportée par les rayonnements du lidar.
- Aide à définir les performances du capteur .

  ⇨ Distance maximale d’émission du laser.

I. Bilan radiométrique

1) Fonction de transfert radiométrique

Calcule le nombre de photons qui revient lorsqu'on éclaire une cible.

- Puissance reçue pour une cible résolue : ($A_{cible} \geq (\theta_T R + d_{TA})$)

$$P_{reçue} = P_{laser} \rho T_A^2 T_R T_T \frac{4 A_R \cos(\theta_S)}{\pi R^2}$$

- Puissance reçue pour une cible non résolue : ($A_{cible} \leq (\theta_T R + d_{TA})$)

$$P_{reçue} = P_{laser} \rho T_A^2 T_R T_T \frac{4 A_R A_{cible} \cos(\theta_S)}{\pi^2 R^4 \theta_T^2}$$

I. Bilan radiométrique

2) Rapport signal à bruit :SNR

Mesure la qualité de la transmission d'une information par rapport aux bruits.

Détermine la précision de la mesure de distance.

$$SNR = \frac{P_{reçue}}{NEP}$$

NEP, noise equivalent power : puissance reçue par le détecteur correspondant aux bruits.

I. Bilan radiométrique

3) Calcul de la distance maximale en fonction de la visibilité

On cherche:

- Pour une cible résolue:

$$\min_{R \in \Re} SNR - \left(P_{laser} \rho T_R T_T \frac{4 A_R \cos(\theta_S)}{\pi}\right) \frac{e^{-2\gamma(V)R}}{R^2}$$

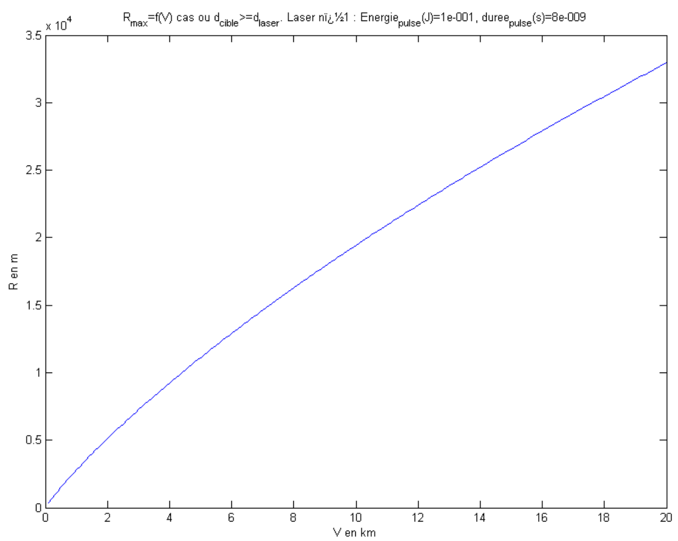
- Pour une cible non résolue:

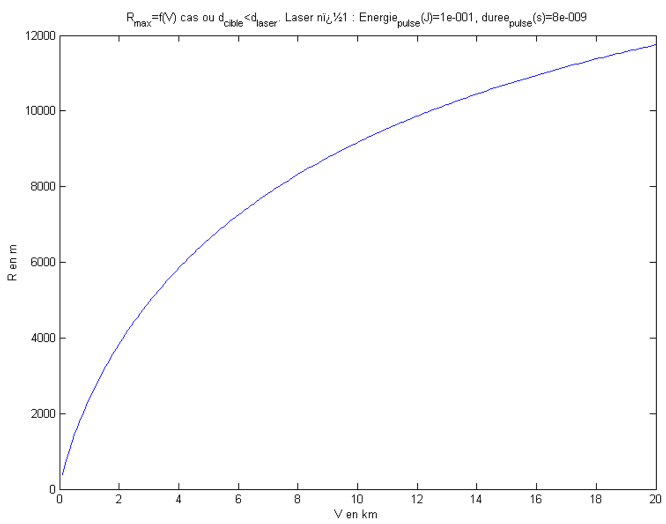
$$\min_{R \in \Re} SNR - \left(P_{laser} \rho T_R T_T \frac{4 A_R A_{cible} \cos(\theta_S)}{\pi^2 {\theta_T}^2}\right) \frac{e^{-2\gamma(V)R}}{R^4}$$

I. Bilan radiométrique

3) <u>Calcul de la distance maximale en fonction de la visibilité</u>

Exemple de résultats obtenus pour un type de laser:





Cible résolue
<u>Visibilité maximale</u>: 20 km
<u>Distance d'émission maximale</u>: 33 km

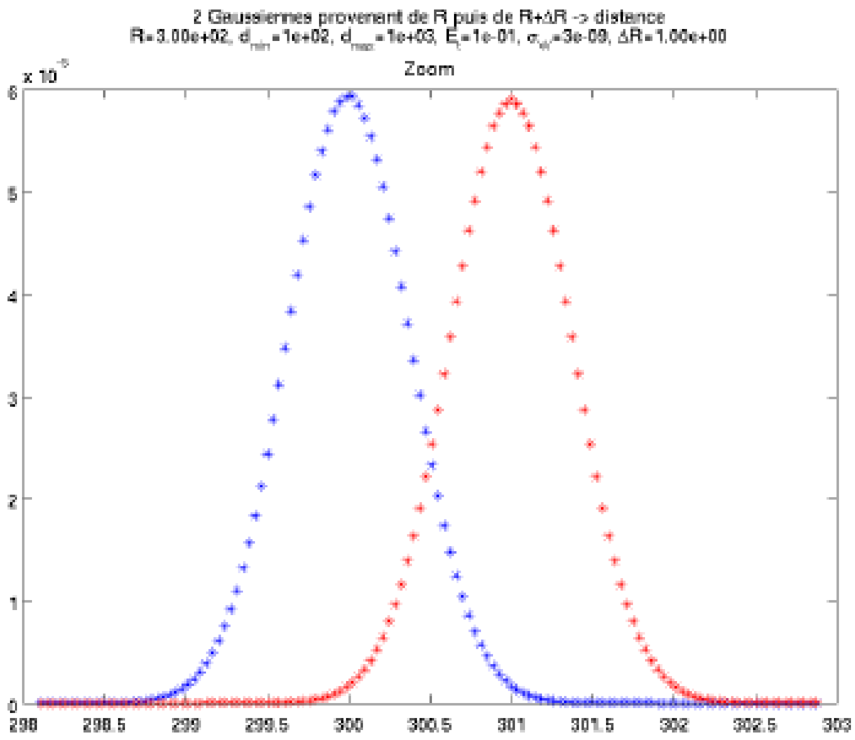
Cible non résolue
<u>Visibilité maximale:</u> 20 km
<u>Distance d'émission maximale:</u> 12 km

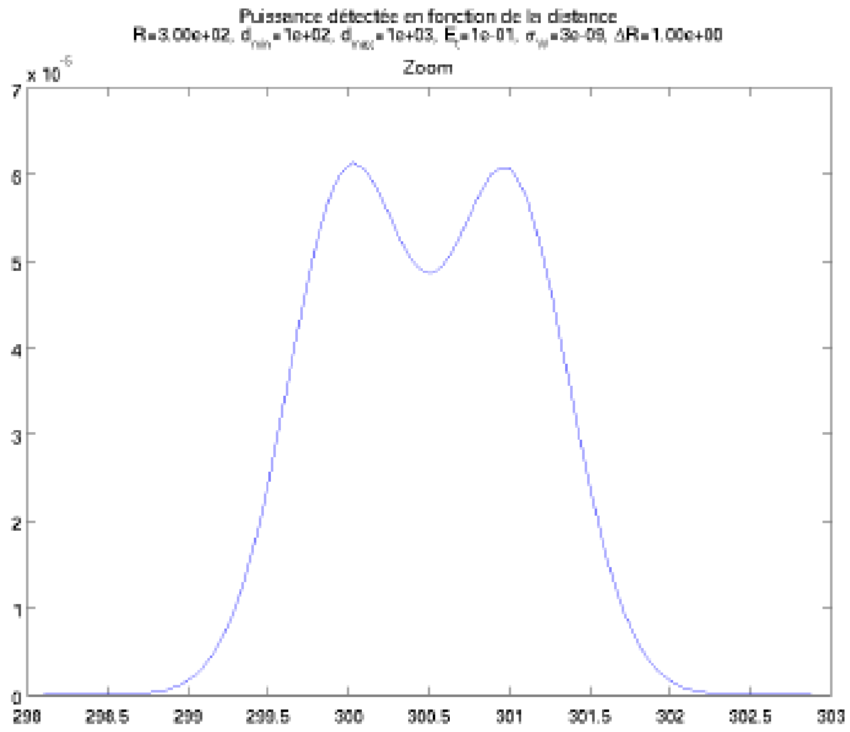
II. Analyse temporelle

- Modélisation : plans d'équations
  - z=R
  - z=R+ ΔR
- Impulsion gaussienne de variance σ_W
- Retour de l'impulsion atténuée (=> Radiométrie)
- Echantillonnage du retour de signal : $\Delta t = \sigma_W / 10$

II. Analyse temporelle

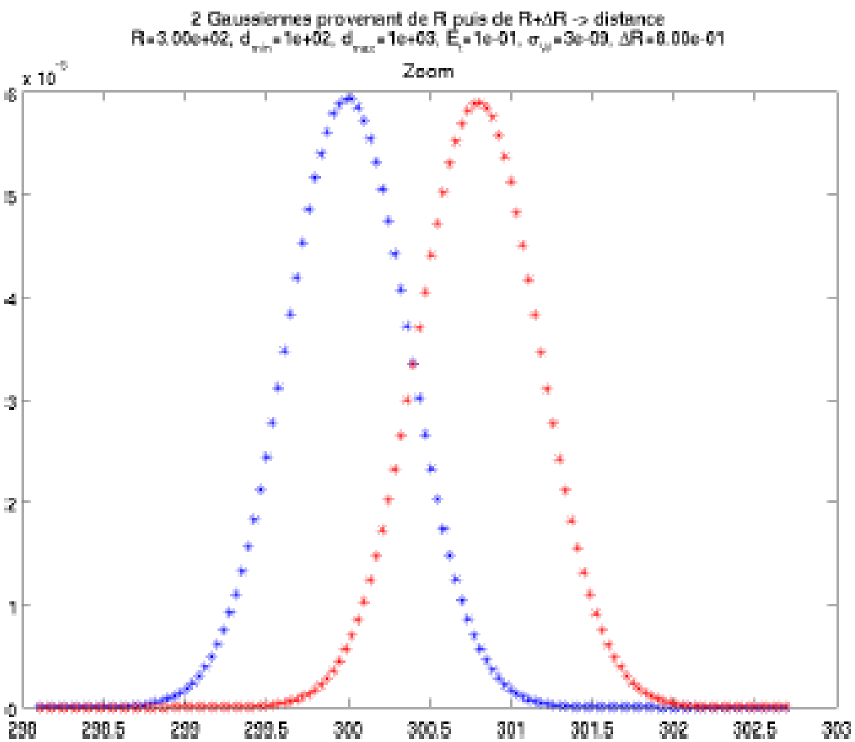
- ΔR = 1 m

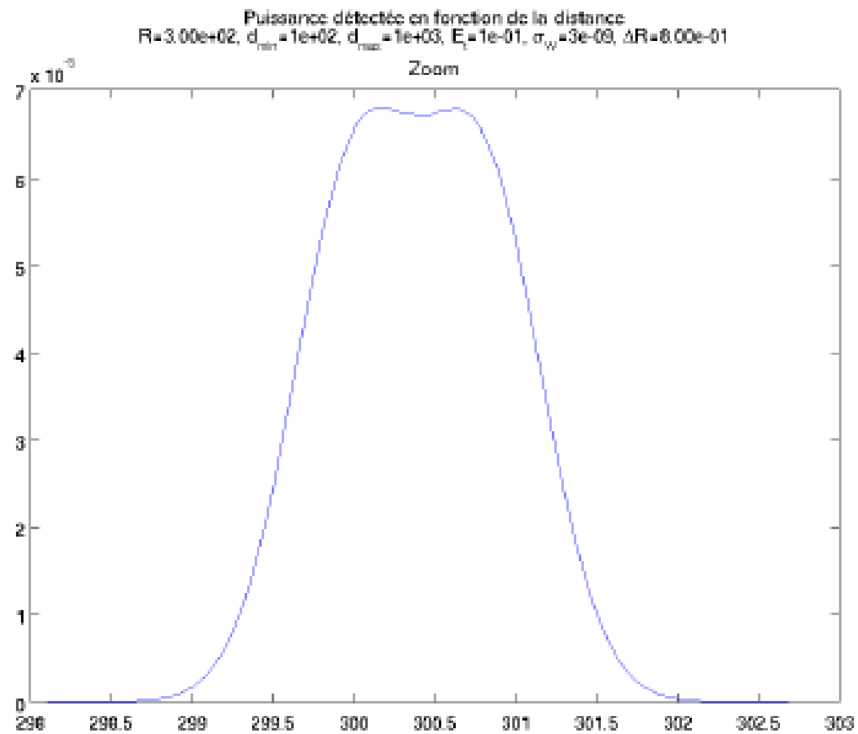




II. Analyse temporelle

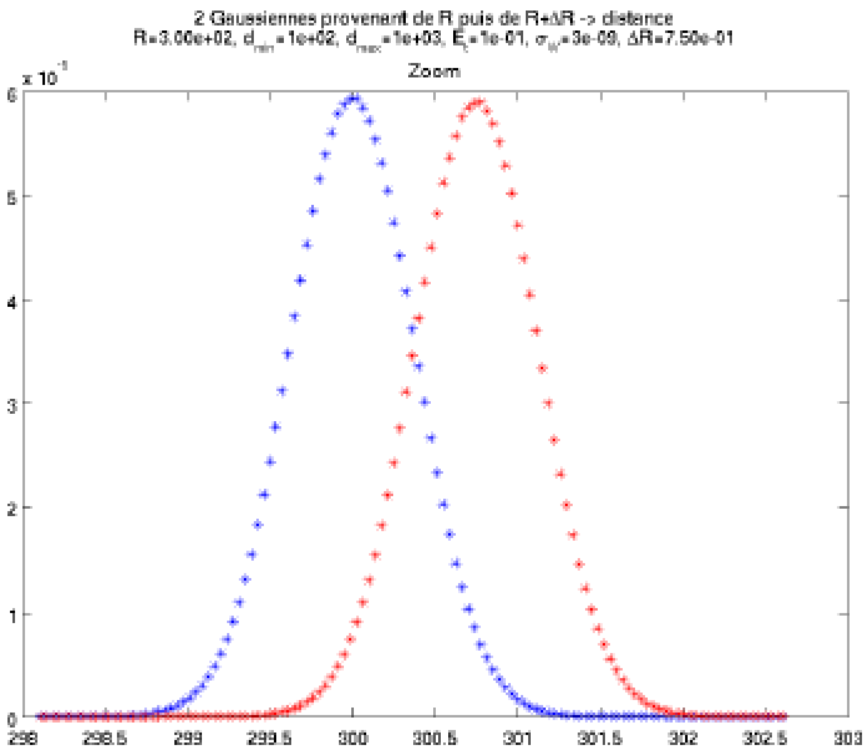
- ΔR = 0.8 m

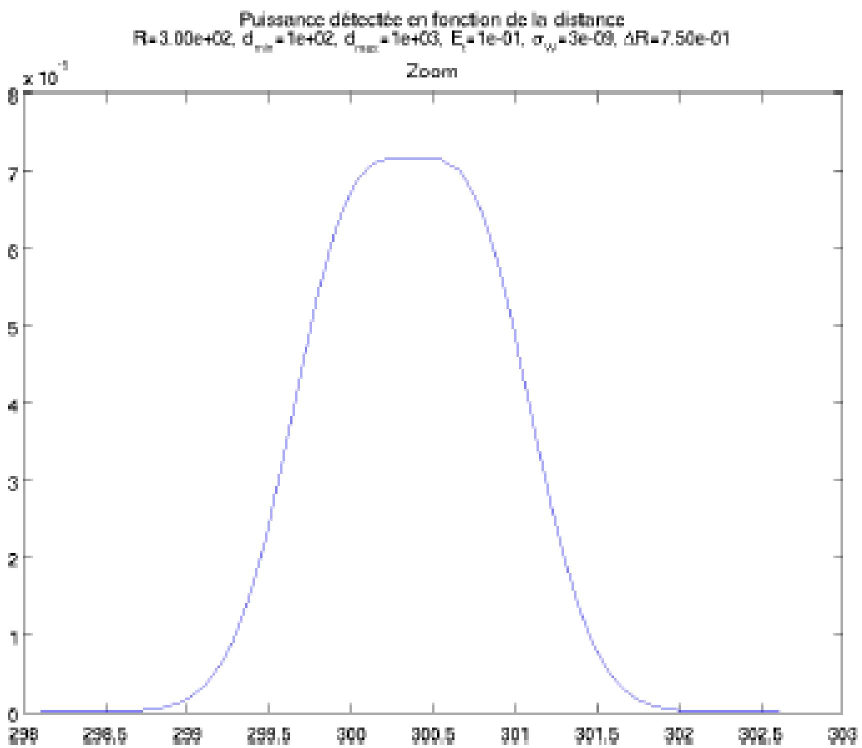




II. Analyse temporelle

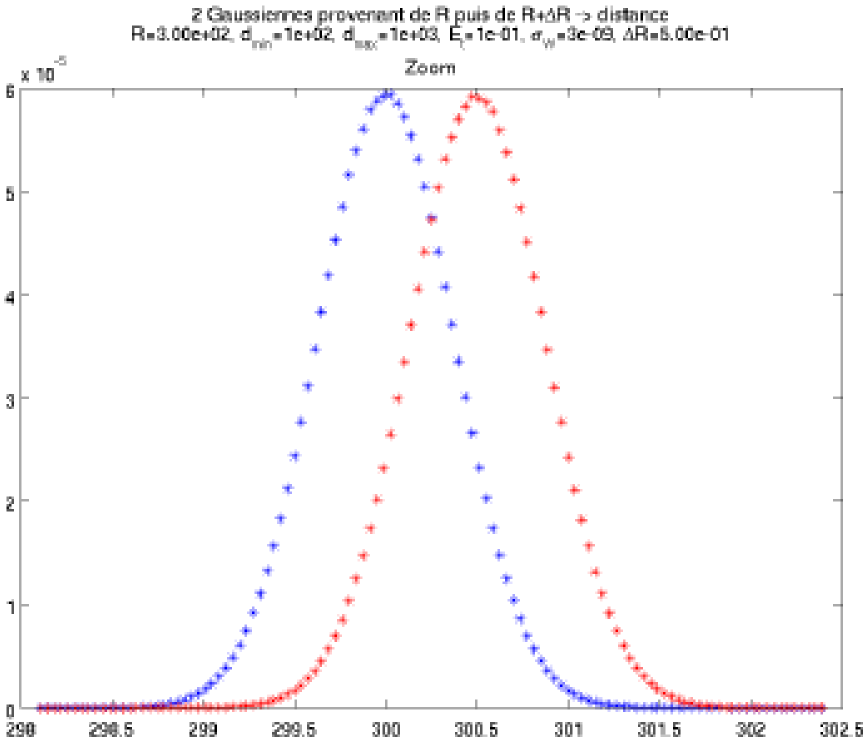
- ΔR = 0.75 m

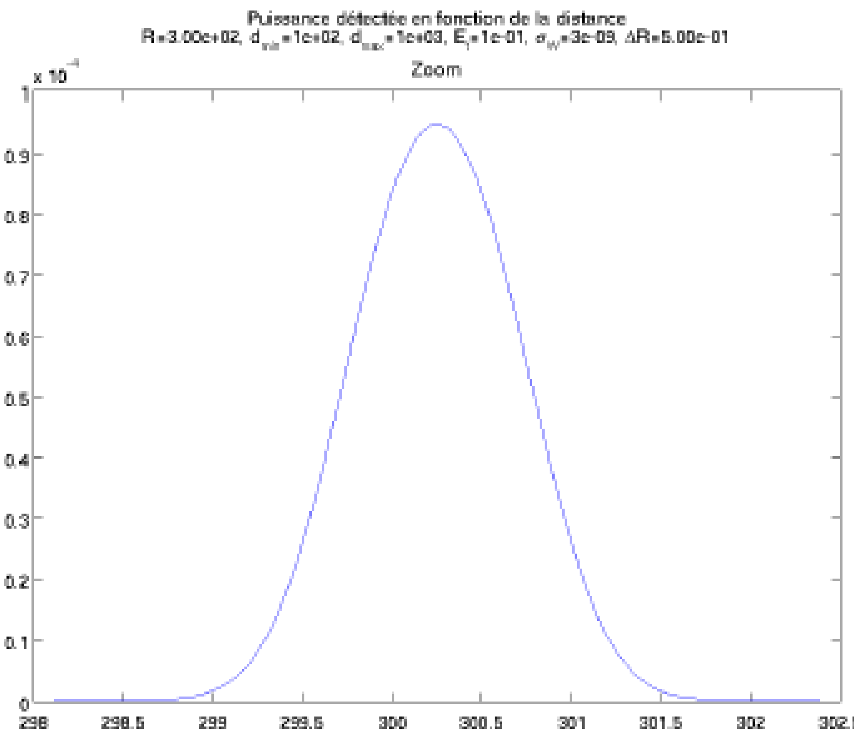




II. Analyse temporelle

- ΔR = 0.5 m





III. Simulateur

1) Signal utile

- Conversion puissance -> photons   P=hν
- Arrivée de photons => conversion électrons
- Rendement η : 1 photons = η électrons
- Ici, η =1

$$E(k) = \int_{t_k - \frac{\Delta t}{2}}^{t_k + \frac{\Delta t}{2}} P_{det}(t)\, dt \approx P(t_k)\Delta t$$

$$\mathbb{E}[K] = \frac{P_{det}(k)\Delta t}{h\nu}$$

III. Simulateur

2) Les bruits

- Bruit quantique : arrivée aléatoire des photons

  => Loi de Poisson

$$\mathbb{P}[n,k] = \frac{(\mathbb{E}[K])^n \, e^{-\mathbb{E}[K]}}{n!}$$

- Bruit d'éclairement + bruit d'obscurité

  => Loi de Poisson

- Bruit thermique : gaussienne centrée de variance

$$Q_n^2 = k_B T C / q_e^2$$

III. Simulateur

3) Algorithme

Fonction simulation1prise(R,dmin,dmax,cosVerticaleNormale) :

K : **Nombre d'échantillons**

E_{detect} : **Vecteur échantillon indicé de 1 à** K

Pour k **de** 1 **à** K **faire**

$$K(i) \leftarrow \int_{t_k-\Delta t/2}^{t_k+\Delta t/2} P_{detect}(t)dt$$

[P_{detect} calculé en combinant formules de radiométrie et temporelles]

[dmin et dmax permettent de définir la fenêtre d'analyse temporelle]

Fin Pour

Retourner E_{detect}

Fin

III. Simulateur

3) Algorithme

Fonction simulateur(X,Y,Z,nbPixels,sigmaXY) :

[(X, Y, Z) Paysage cartésien "réel", sigmaXY écart type du noyau de convolution de la lentille]

Se donner dmin, dmax, K

Signal, u_0,Signal_capté : **matrices nbPixels** $\times K$

Image_detectée : **vecteur de dimension nbPixels**

Pour i **de** 1 **à** nbPixels **faire**

Signal(i, :)$\leftarrow$ simulation1prise(Z(i),dmin,dmax,cosVerticaleNormale)

Fin Pour

Pour k **de** 1 **à** K **faire**

$u_0(:, k) = h*$Signal$(:, k)$

[h gaussienne de variance sigmaXY, $*$ définit le produit de convolution]

Signal_capté$(:, k) = \mathscr{P}($Signal$(:, k)) + \mathscr{P}(N_b) + \mathscr{N}(0, Q_n^2)$

Fin Pour

[$\mathscr{P}(\lambda)$ désigne une loi de Poisson de moyenne λ, $\mathscr{N}$ la loi normale]

Pour i **de** 1 **à** nbPixels **faire**

Image_detectée(i) = max (Signal_capté$(i, :)$)

Fin Pour

Retourner Image_detectée

Fin

IV. Reconstitution de l'image 3D

1) Position générale du problème

$$u_0 = h * u + b$$

u image réelle
u_o l'image réelle convoluée par la lentille de l'appareil
h noyau de convolution du système optique, supposé connu et gaussien
b bruits

On a alors:

$$u_0 = Hu + b$$

IV. Reconstitution de l'image 3D

1) <u>Position générale du problème</u>

On cherche donc:

$$u^* = \arg\min_{u \in \mathbb{R}^n} \|\nabla u\|_p^p + \lambda \|Hu - u_0\|_2^2$$

$p \in \{1, 2\}$ et $\lambda > 0$

On peut raffiner en posant:

$$u^* = \arg\min_{u \in \mathbb{R}^n} \|\nabla u\|_p^p + \frac{\lambda}{2} \|D(Hu - u_0)\|_2^2$$

D est une matrice diagonale visant à favoriser certains pixels par rapport à d'autres.

IV. Reconstitution de l'image 3D

2) Algorithme de Chambolle-Pock

- L'algorithme de Chambolle-Pock consiste à résoudre le problème dual d'un problème de minimisation:

$$\underbrace{\min_{x\in X} F(Kx)+G(x)}_{problème\, primal} = \underbrace{\max_{y\in Y} -(G^*(-K^*y)+F^*(y))}_{problème\, dual}$$

IV. Reconstitution de l'image 3D

2) Algorithme de Chambolle-Pock

- <u>Initialisation:</u> τ,σ >0, θ∈ [0;1], $(x^0, y^0) \in X \times Y$ et $\overline{x}^0 = x^0$
- Pour n≥0,

$$\begin{cases} y^{n+1} = (I + \sigma\partial F^*)^{-1}(y^n + \sigma K \overline{x}^n) \\ x^{n+1} = (I + \tau\partial G)^{-1}(x^n - \tau K^* \overline{y}^{n+1}) \\ \overline{x}^{n+1} = x^{n+1} + \theta(x^{n+1} - x^n) \end{cases}$$

IV. Reconstitution de l'image 3D

2) Algorithme de Chambolle-Pock

- Opérateur proximal:

$$f : \Re^n \to \Re, \text{ convexe s.c.i}$$

$$(I + \partial f)^{-1}(x) = \arg\min_{x' \in \Re^n} f(x') + \frac{1}{2}\|x - x'\|_2^2$$

- Fonction duale:

$$f : \Re^n \to \Re, \text{ fonction convexe s.c.i}$$

$$f^*(x) = \sup_{x' \in \Re^n} < x, x' > - f(x')$$

IV. Reconstitution de l'image 3D

2) Algorithme de Chambolle-Pock

- Pour notre problème, nous cherchons:

$$\min_{x \in \Re^{3n}} \left\|\nabla x\right\|_1 + \frac{\lambda}{2}\left\|D(Hx - x_0)\right\|_2^2$$

- On pose alors:

$$F : \Re^{3n} \to \Re,\ F\left(x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}\right) = \left\|\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}\right\|_1 + \frac{\lambda}{2}\left\|x_3 - x_0\right\|_2^2$$

$$G : \Re^{3n} \to \Re,\ G(x) = 0$$

$$K : \Re^{n} \to \Re^{3n},\ K(x) = \begin{pmatrix} \nabla x \\ DHx \end{pmatrix}$$

IV. Reconstitution de l'image 3D

2) Algorithme de Chambolle-Pock

- Fonction duale de *F*:

$$F^*(y)=\frac{1}{2\lambda}\|y_3\|_2^2+<y_3,x_0>+\begin{cases}0 & \text{si } \forall \mathrm{i}=1,\dots,\mathrm{n} \left\|\begin{pmatrix}(y)_1^i\\(y)_2^i\end{pmatrix}\right\|_2\leq 1\\+\infty & \text{sinon}\end{cases}$$

- Opérateur proximal de *F*:*

$$(I+\sigma\partial F^*)^{-1}(y)=\begin{pmatrix}\Pi_\kappa\begin{pmatrix}y_1\\y_2\end{pmatrix}\\ \dfrac{y_3-\sigma\, x_0}{1+\dfrac{\sigma}{\lambda}}\end{pmatrix}$$

où Π_κ est la projection de $\begin{pmatrix}\mathrm{y}_1\\\mathrm{y}_2\end{pmatrix}$ sur $\kappa=\left\{\begin{pmatrix}y_1\\y_2\end{pmatrix}\in\Re^2,\left\|\begin{pmatrix}y_1\\y_2\end{pmatrix}\right\|_2\leq 1\right\}$

V. Tests numériques

1) Tests sur des images 2D

- $\sigma = 0.02, \sigma_b = 10^{-3}$

FIGURE 5.1 – Image réelle

FIGURE 5.2 – Image convolée et bruitée

FIGURE 5.3 – Image traitée

V. Tests numériques

1) Tests sur des images 2D

- $\sigma = 0.02, \sigma_b = 0.05$

FIGURE 5.1 – Image réelle

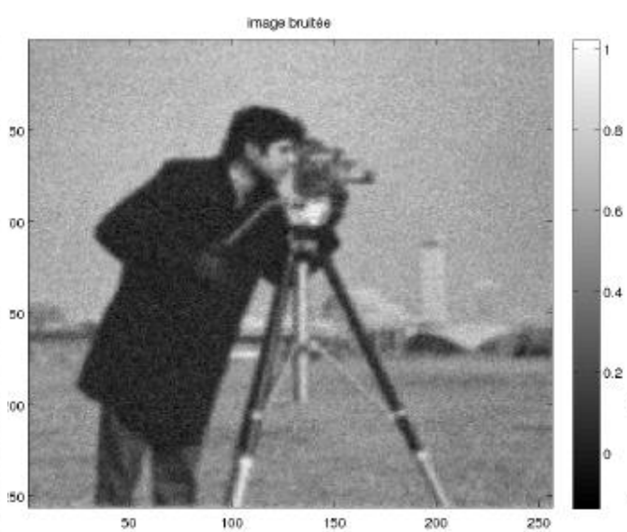


FIGURE 5.5 – Image convolée et bruitée

FIGURE 5.6 – Image traitée

V. Tests numériques

1) Tests sur des images 2D

$\sigma = 0.02, \sigma_b = 10^{-3}$, 10% de pixels

FIGURE 5.1 – Image réelle

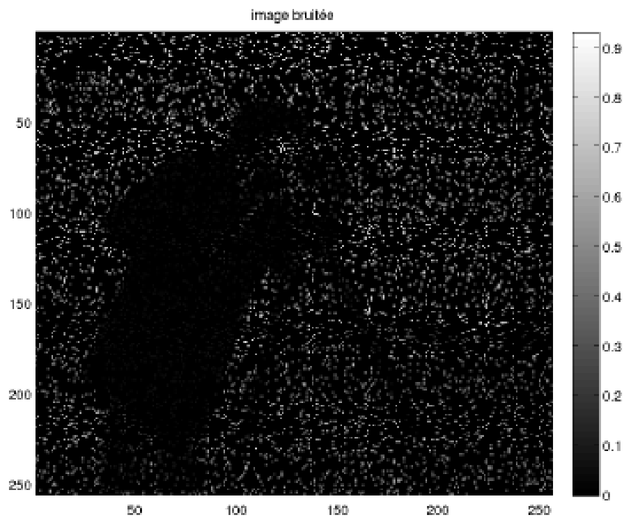


FIGURE 5.7 – Image convolée et bruitée

FIGURE 5.8 – Image traitée

V. Tests numériques

2) Images 3D

- **Cochon**

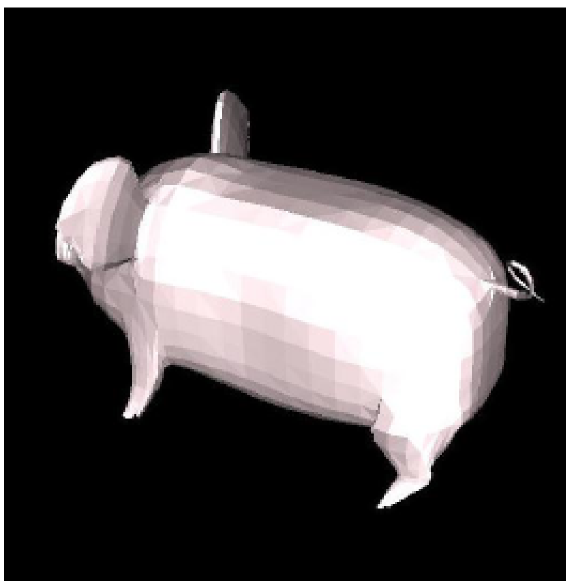


FIGURE 5.9 – Image maillée

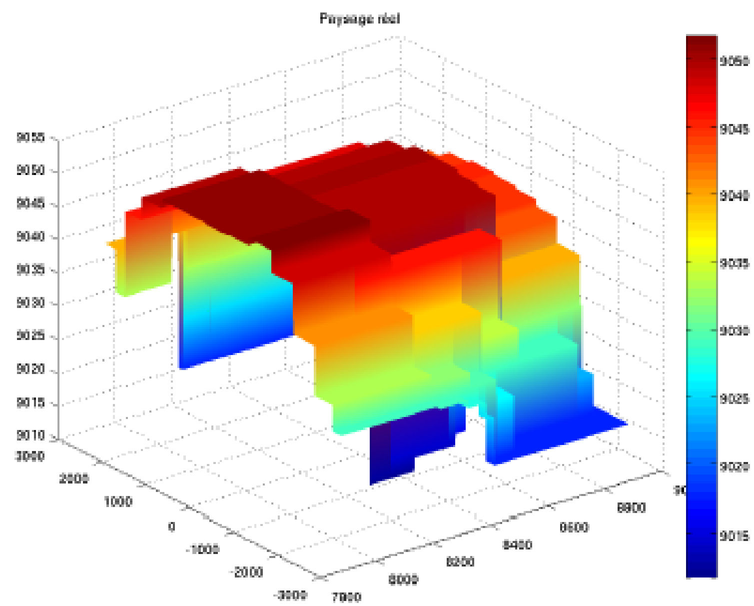


FIGURE 5.10 – Image cartésienne

V. Tests numériques

2) Images 3D

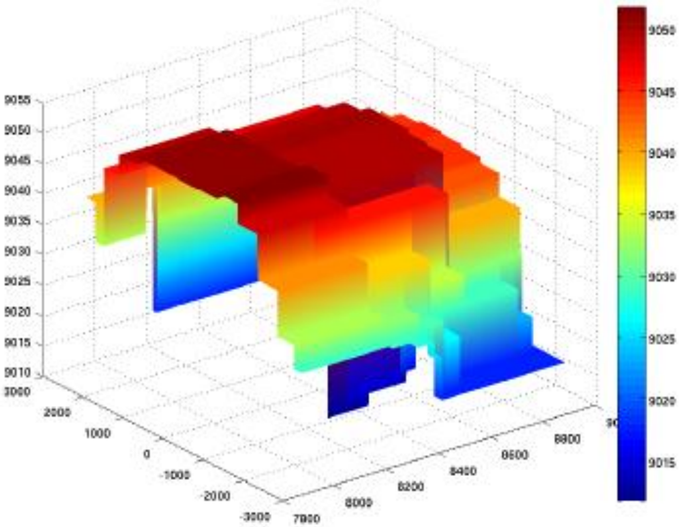


FIGURE 5.10 – Image cartésienne

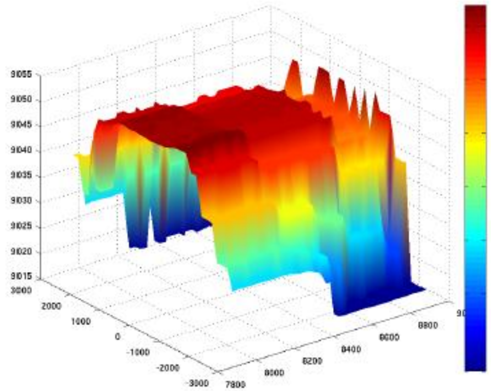


FIGURE 5.11 – Image convolée

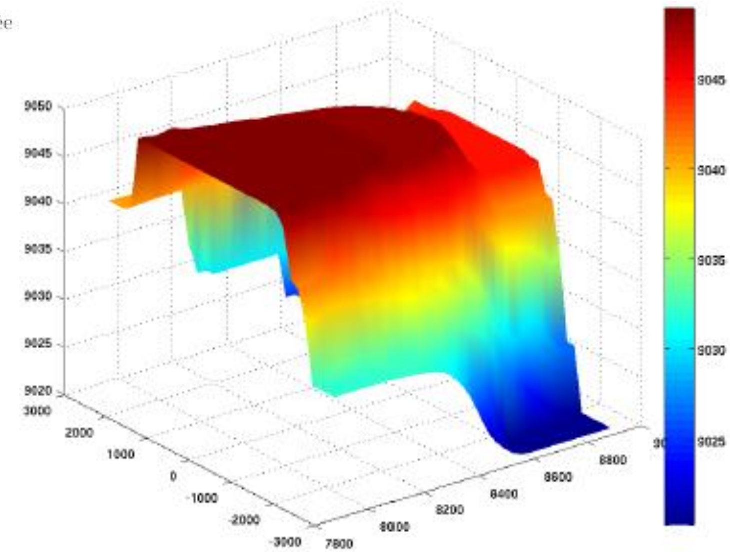


FIGURE 5.12 – Image traitée, $\lambda = 0.1$

V. Tests numériques

2) Images 3D

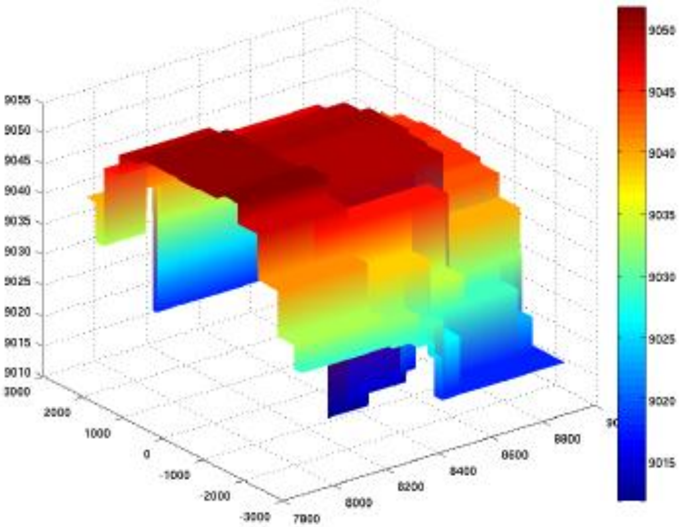


FIGURE 5.10 – Image cartésienne

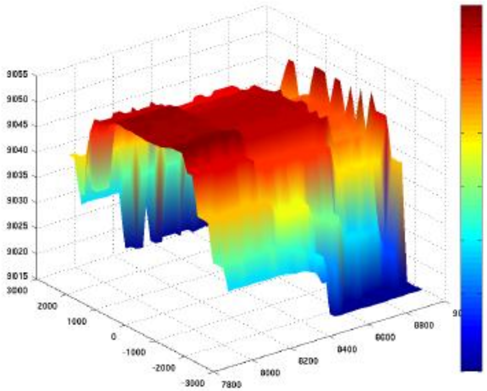


FIGURE 5.11 – Image convolée et bruit

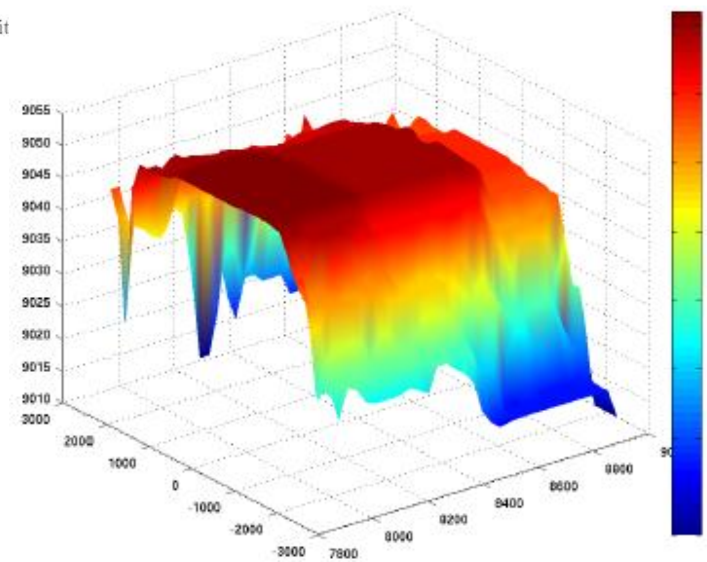


FIGURE 5.13 – Image traitée, $\lambda = 1$

V. Tests numériques

2) Images 3D

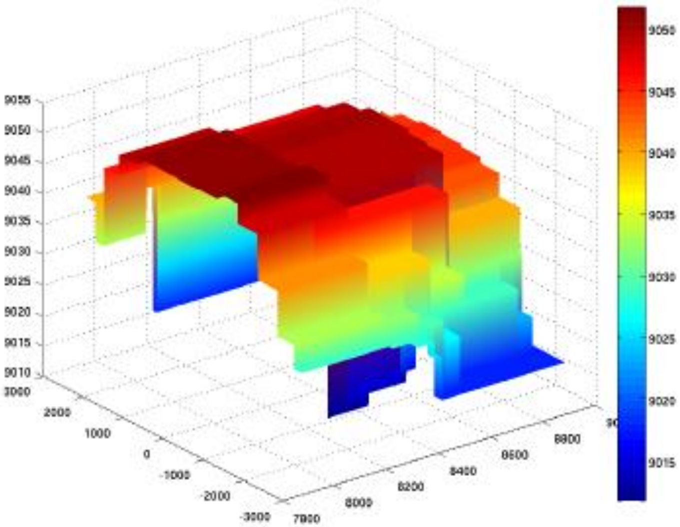


FIGURE 5.10 – Image cartésienne

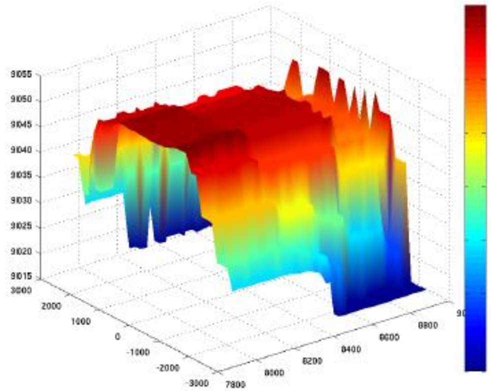


FIGURE 5.11 – Image convolée et bru

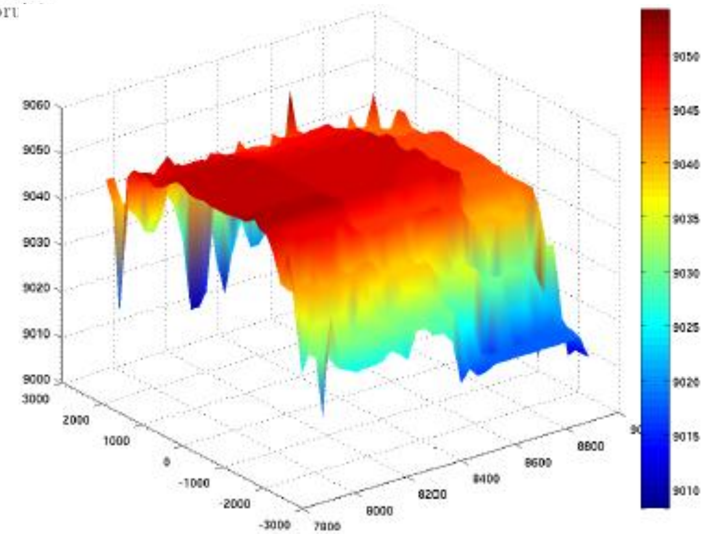


FIGURE 5.14 – Image traitée, $\lambda = 2$

V. Tests numériques

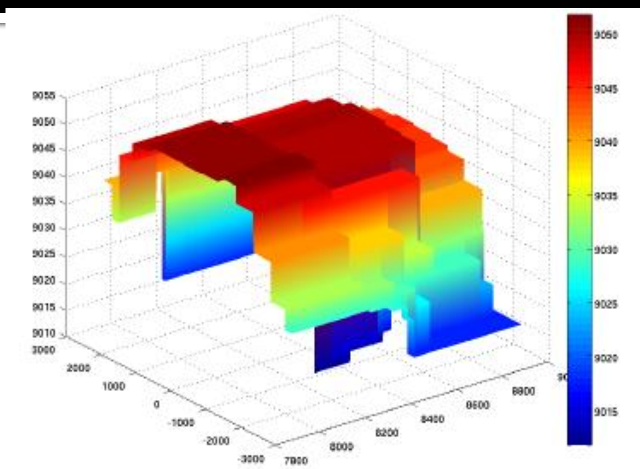


FIGURE 5.10 – Image cartésienne

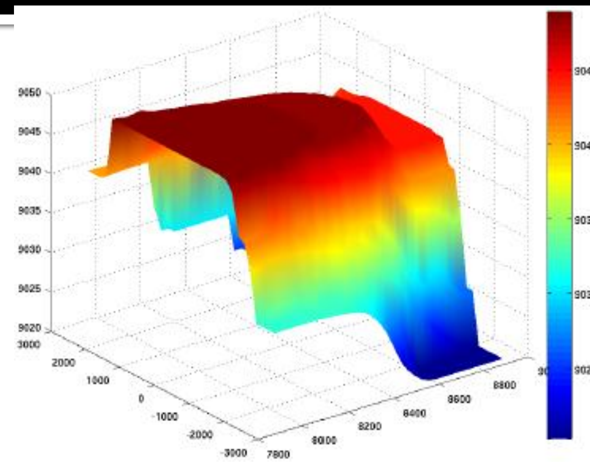


FIGURE 5.12 – Image traitée, $\lambda = 0.1$

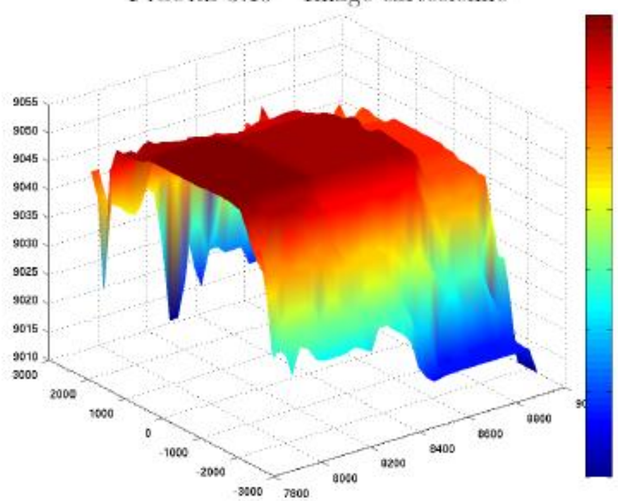


FIGURE 5.13 – Image traitée, $\lambda = 1$

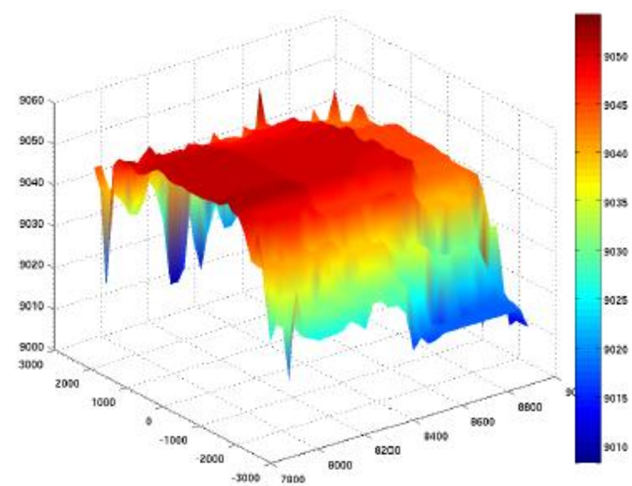


FIGURE 5.14 – Image traitée, $\lambda = 2$

V. Tests numériques

2) Images 3D

- **Paysage**

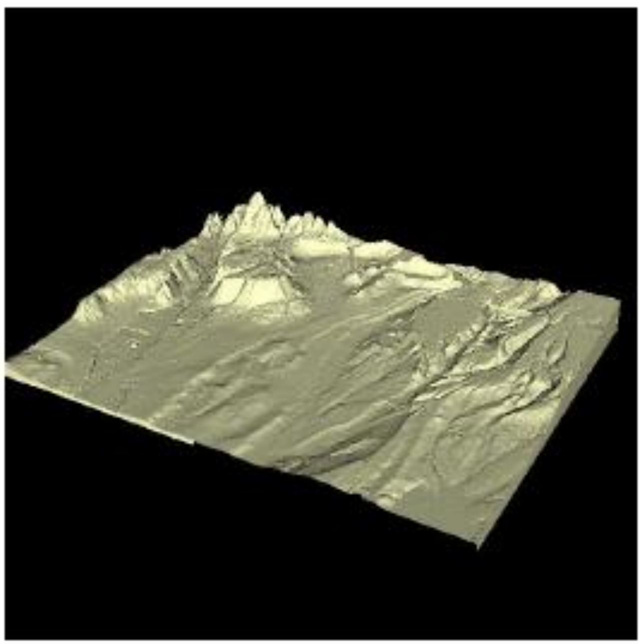


FIGURE 5.15 – Image maillée

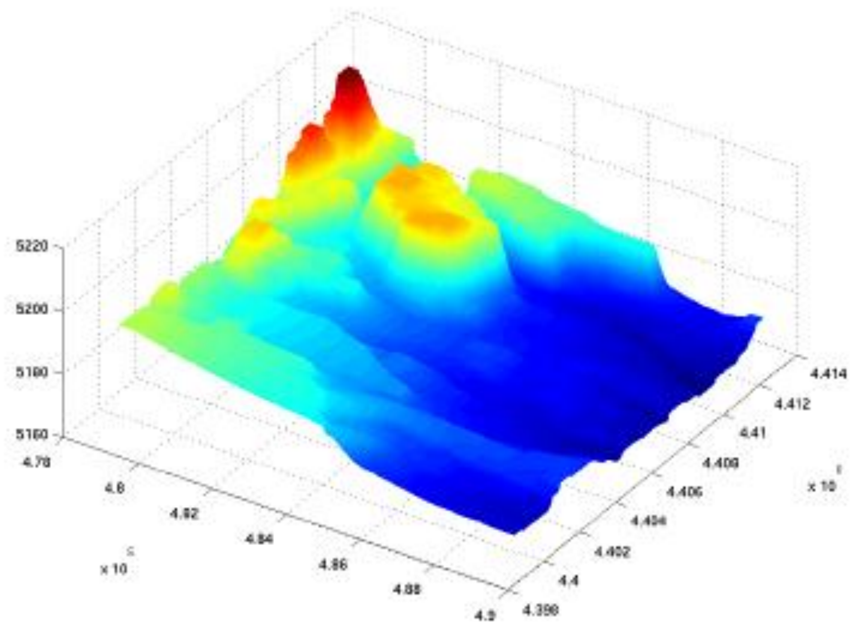


FIGURE 5.16 – Image cartésienne

V. Tests numériques

2) Images 3D

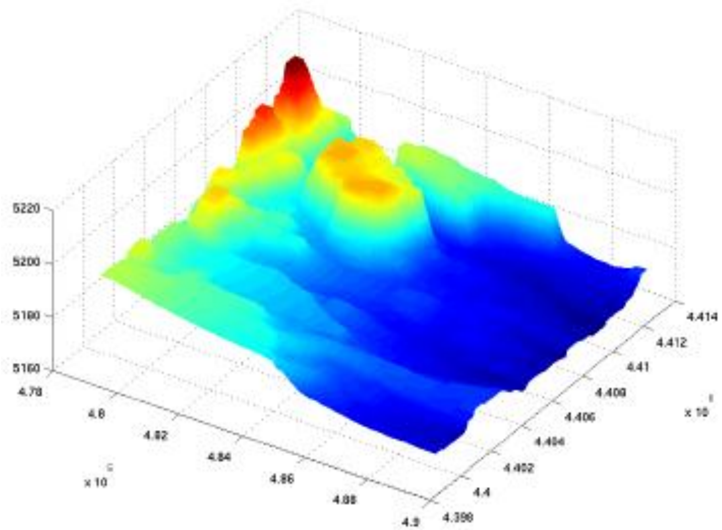
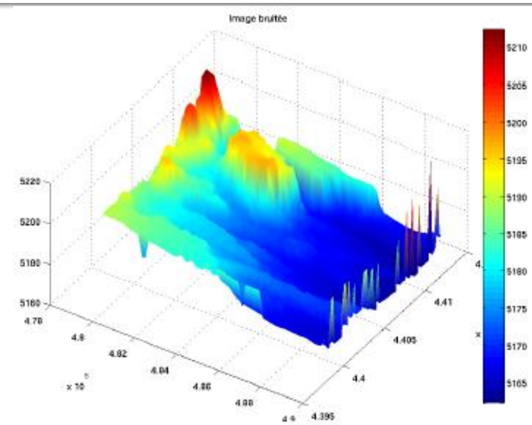


FIGURE 5.16 – Image cartésienne



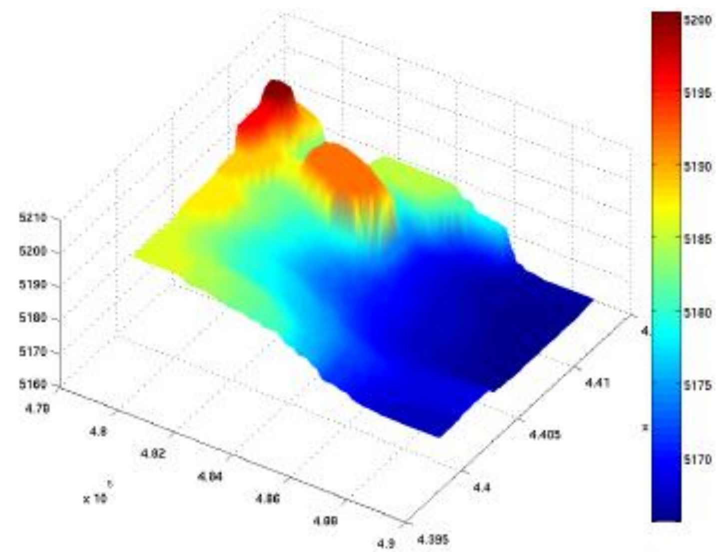


FIGURE 5.18 – $\lambda = 0.1$

V. Tests numériques

2) Images 3D

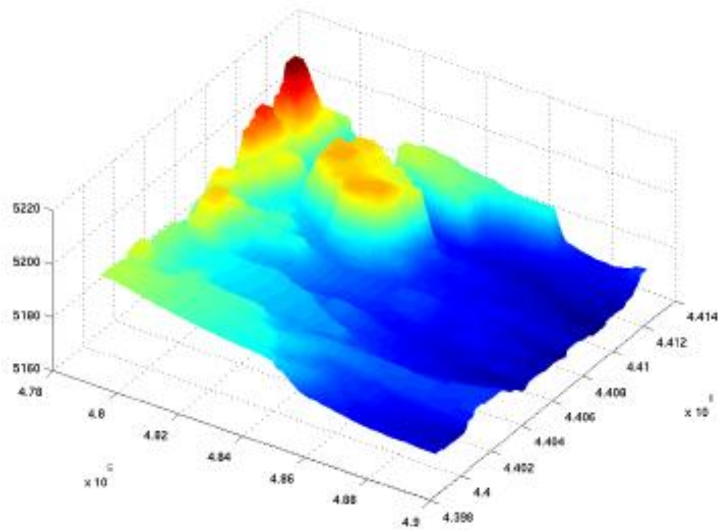
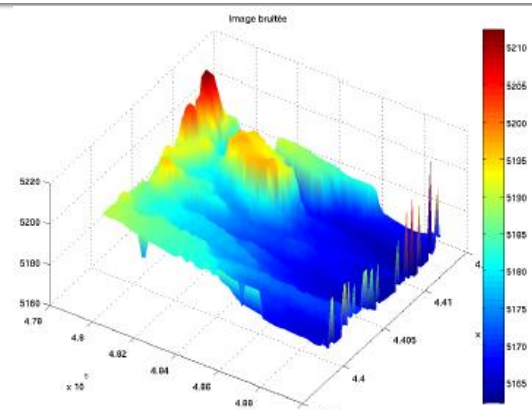


FIGURE 5.16 – Image cartésienne



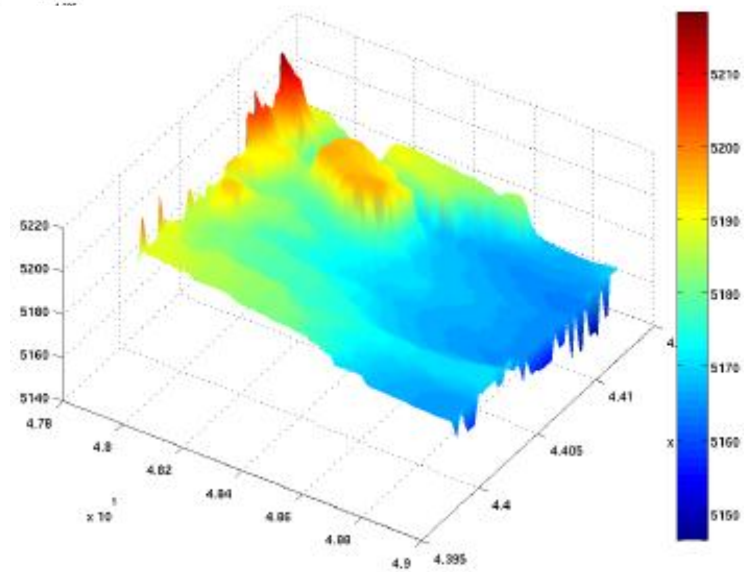


FIGURE 5.19 – $\lambda = 1$

V. Tests numériques

2) Images 3D

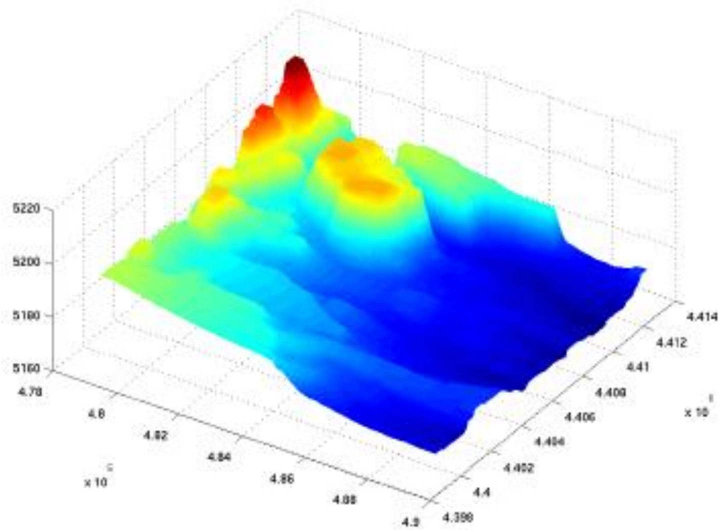
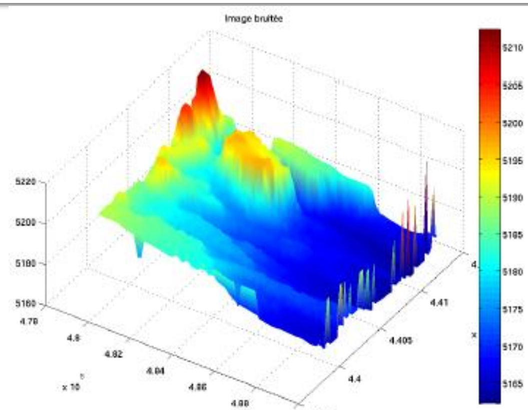


FIGURE 5.16 – Image cartésienne



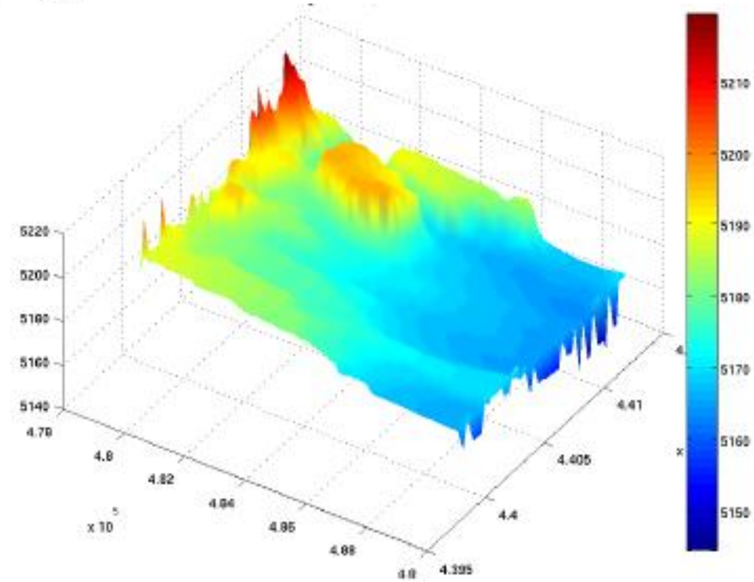


FIGURE 5.20 – $\lambda = 1.2$

V. Tests numériques

2) Images 3D

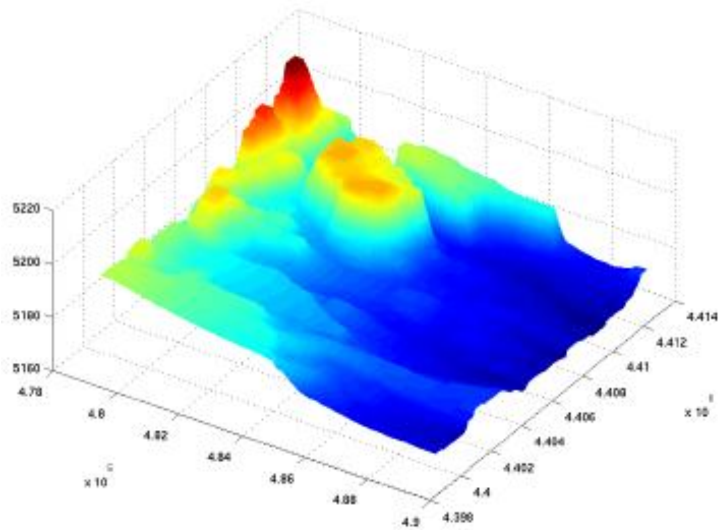
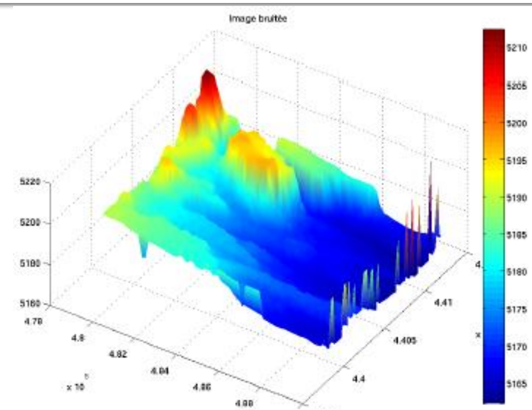


FIGURE 5.16 – Image cartésienne



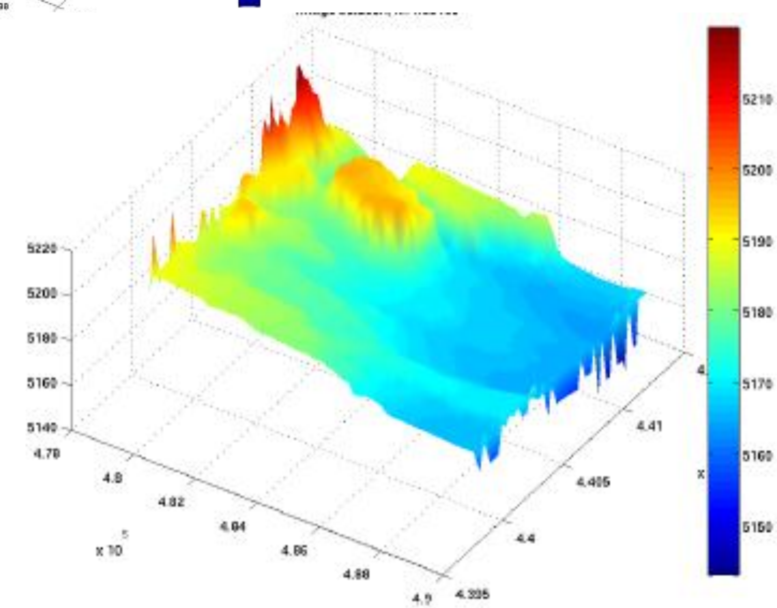


FIGURE 5.21 – $\lambda = 1.33$

V. Tests numériques

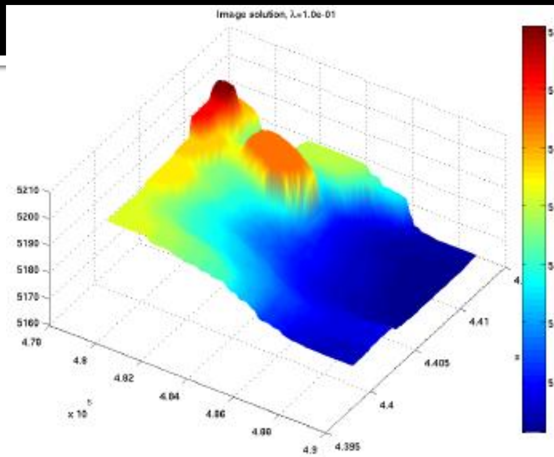


FIGURE 5.18 – $\lambda = 0.1$

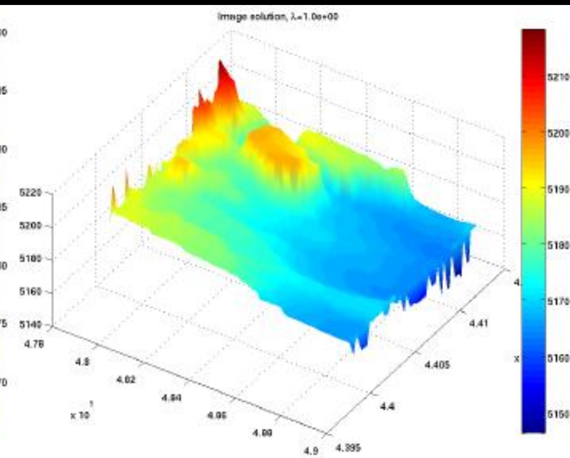


FIGURE 5.19 – $\lambda = 1$

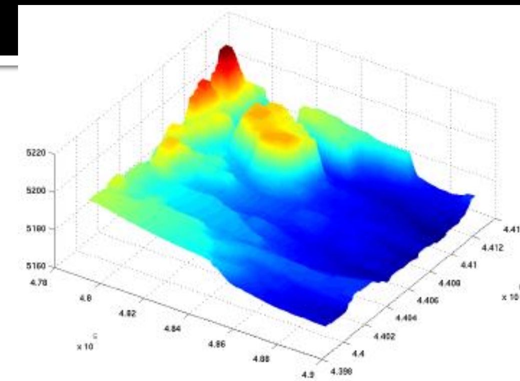


FIGURE 5.16 – Image cartésienne

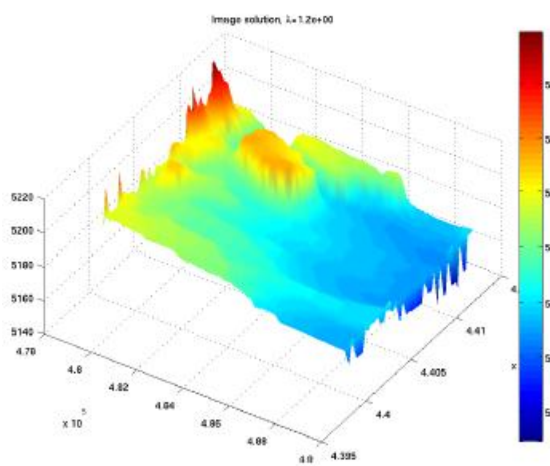


FIGURE 5.20 – $\lambda = 1.2$

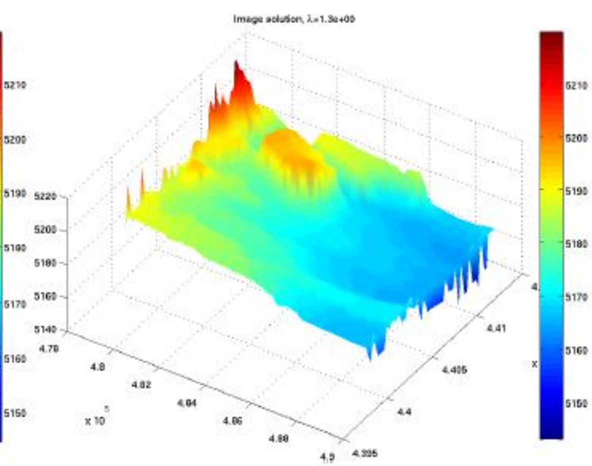
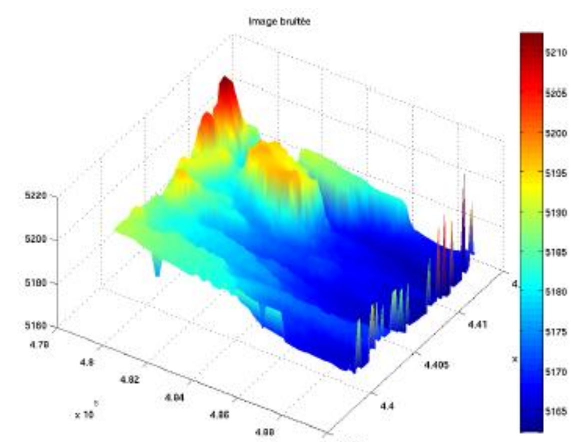


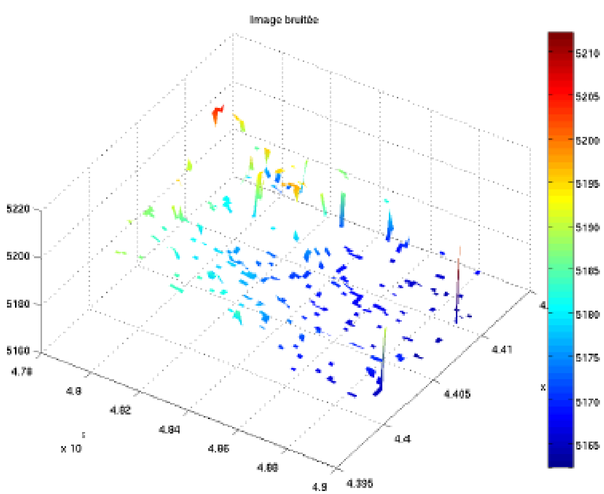
FIGURE 5.21 – $\lambda = 1.33$

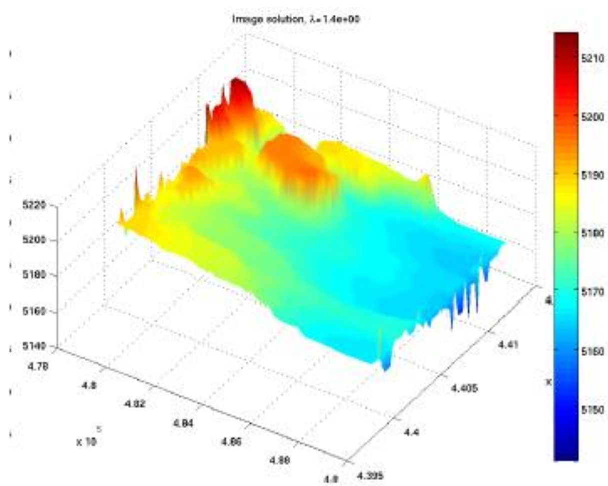


V. Tests numériques

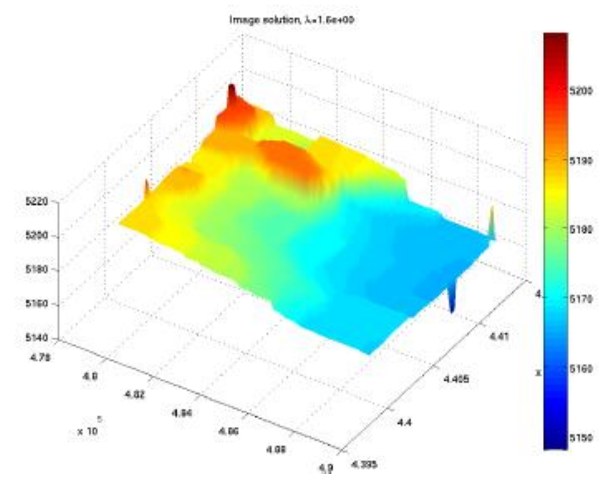
2) Images 3D

- **Perte d'information : pixels manquants**





50% de perte



90% de perte

Conclusion

- Projet intéressant
- Reconstitution d’images correcte mais difficile à mettre en œuvre.
- Limites du modèle : résultats mitigés
- **Merci à Jacques et Pierre ☺**