

Chapitre 6

Régularisation

La régularisation d'un problème **mal-posé** consiste à le transformer en un problème **bien-posé**, c'est-à-dire à définir **une solution unique** pour **toutes les mesures possibles** dans l'espace des observations, et à **assurer la stabilité** de cette solution vis-à-vis des **erreurs** sur ces mesures.

6.1 Définition d'un opérateur de régularisation

Un **régulariseur** de $\mathbf{g} = \mathbf{H}\mathbf{f}$ est une famille d'opérateurs $\{\mathbf{R}_\lambda; \lambda \in \Lambda \subset \mathbf{R}^+\}$ tels que :

- $\forall \mathbf{f} \in F, \lim_{\lambda \rightarrow 0} \|\mathbf{R}_\lambda \mathbf{H}\mathbf{f} - \mathbf{f}\|^2 = 0$
- $\forall \lambda \in \Lambda, \mathbf{R}_\lambda$ est un opérateur continu de G dans F

\Downarrow
 Solution Approchée

$$\mathbf{f}_\epsilon = \mathbf{R}_\lambda \mathbf{g}_\epsilon, \quad \text{avec} \quad \mathbf{g}_\epsilon = \mathbf{H}\mathbf{f} + \mathbf{b}$$

$$\|\mathbf{f}_\epsilon - \mathbf{H}^{-1}\mathbf{g}\| \leq \underbrace{\|\mathbf{R}_\lambda \mathbf{g} - \mathbf{H}^{-1}\mathbf{g}\|}_{\text{erreur de régularisation}} + \underbrace{\|\mathbf{R}_\lambda(\mathbf{g}_\epsilon - \mathbf{g})\|}_{\text{erreur due au bruit}}$$

- Utilisation d'E.H.N.R.
- Décomposition TSVD
- Régularisation P.T.T.

6.2 Décomposition tronquée en valeurs singulières

Considérons l'équation

$$\mathbf{g} = \mathbf{H}\mathbf{f}$$

et notons par $\{\mathbf{v}_j, j = 1, \dots, K\}$ les vecteurs propres de $\mathbf{H}^t\mathbf{H}$ et par $\{\mathbf{u}_i, i = 1, \dots, K\}$ les vecteurs propres de $\mathbf{H}\mathbf{H}^t$ et par λ_j^2 les valeurs propres correspondantes. Nous avons alors :

$$\mathbf{H}^t\mathbf{H}\mathbf{v}_j = \lambda_j^2\mathbf{v}_j, \quad j = 1, \dots, K$$

$$\mathbf{H}\mathbf{H}^t\mathbf{u}_i = \lambda_i^2\mathbf{u}_i, \quad i = 1, \dots, K$$

$$\mathbf{H}\mathbf{v}_j = \lambda_j\mathbf{u}_j, \quad j = 1, \dots, K$$

$$\mathbf{H}^t\mathbf{u}_i = \lambda_i\mathbf{v}_i, \quad i = 1, \dots, K$$

$$\mathbf{H} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^t$$

$$\mathbf{U} = (\mathbf{u}_1 : \mathbf{u}_2 : \dots : \mathbf{u}_K), \quad \mathbf{V} = (\mathbf{v}_1 : \mathbf{v}_2 : \dots : \mathbf{v}_K), \quad \mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_K)$$

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_K, \quad K \leq \inf(M, N)$$

- On décide qu'à partir du rang $r < K$, les valeurs singulières ordonnées ne sont plus significatives.
- Dans l'inversion, on utilise la pseudo-inverse \mathbf{H}^+ .

$$\mathbf{H}^+ = \mathbf{V}\mathbf{\Lambda}^+\mathbf{U}^t$$

$$\mathbf{\Lambda}^+ = \text{diag}\{\alpha_i\}, \quad \begin{cases} \alpha_i = \frac{1}{\lambda_i} & \text{si } \lambda_i \neq 0 \text{ (significatif)} \\ \alpha_i = 0 & \text{si } \lambda_i \simeq 0 \text{ (non-significatif)} \end{cases}$$

$$\mathbf{g} + \delta\mathbf{g} = \mathbf{H}(\mathbf{f} + \delta\mathbf{f}) \longrightarrow \frac{\|\delta\mathbf{f}\|}{\|\mathbf{f}\|} \leq \left(\frac{\lambda_{max}}{\lambda_{min}}\right)^2 \frac{\|\delta\mathbf{g}\|}{\|\mathbf{g}\|}$$

6.3 Régularisation par méthodes itératives

- Méthodes itératives du type inversion généralisée (BIALY, LANDWEBER,...)
- établir une règle d'arrêt fondée sur l'évolution du résidu

$$\|\mathbf{H}\hat{\mathbf{f}}^k - \mathbf{g}_\epsilon\| > \delta \text{ pour } i < k$$

$$\|\mathbf{H}\hat{\mathbf{f}}^k - \mathbf{g}_\epsilon\| < \delta \text{ pour } i > k$$

- $\frac{1}{k}$ joue le rôle du paramètre de régularisation.

6.4 Introduction de contraintes supplémentaires

\mathbf{C} un opérateur de contrainte

$$\mathbf{f} = \mathbf{C}\mathbf{f} \text{ ssi } \mathbf{f} \text{ satisfait la contrainte}$$

Exemple: Positivité

$$x_j = \begin{cases} x_j & \text{si } x_j > 0 \\ 0 & \text{sinon} \end{cases}, \quad j = 1, \dots, n$$

Le problème initial $\mathbf{g} = \mathbf{H}\mathbf{f}$ devient alors

$$\mathbf{g} = \mathbf{H}\mathbf{C}\mathbf{f}$$

L'itération générale est modifiée en remplaçant l'opérateur \mathbf{H} par $\mathbf{H}\mathbf{C}$.

6.5 Régularisation avec *a priori* de douceur

Tikhonov (régularisation continue) :

$$g_i = \int f(t)h_i(t) dt + b_i$$

$$\begin{aligned} J(f) &= Q(f) + \lambda\Omega(f) \\ &= \sum_i \left| g_i - \int f(t)h_i(t) dt \right|^2 + \lambda \int \sum_k p_k(t) \left| \frac{\partial^k f}{\partial t^k} \right|^2 dt, \quad \lambda > 0, \end{aligned}$$

où les $p_k(t)$ sont des fonctions positives, continues, et continûment dérivables.

PHILIPPS et TWOMEY (régularisation discrète) :

$$J_\lambda(\mathbf{f}) = \|\mathbf{g} - \mathbf{H}\mathbf{f}\|^2 + \lambda\Omega(\mathbf{f}) \quad \lambda > 0$$

Exemple : $k = 1, p_k(t) = 1$

$$J_\lambda(f) = \sum_i \left| g_i - \int f(t)h_i(t) dt \right|^2 + \lambda \int |f'(t)|^2 dt$$

Cas discret :

$$J_\lambda(\mathbf{f}) = \|\mathbf{g} - \mathbf{H}\mathbf{f}\|^2 + \lambda\|\mathbf{D}_k\mathbf{f}\|^2 = [\mathbf{g} - \mathbf{H}\mathbf{f}]^t[\mathbf{g} - \mathbf{H}\mathbf{f}] + \lambda[\mathbf{D}_k\mathbf{f}]^t[\mathbf{D}_k\mathbf{f}]$$

avec

$$\mathbf{D}_0 = \mathbf{I}, \quad \mathbf{D}_1 = \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ -1 & 1 & \ddots & & \vdots \\ 0 & -1 & 1 & \ddots & \vdots \\ \vdots & \ddots & & & 0 \\ 0 & \cdots & 0 & -1 & 1 \end{pmatrix}, \quad \mathbf{D}_2 = \begin{pmatrix} 1 & 0 & \cdots & \cdots & \cdots & 0 \\ -2 & 1 & \ddots & & & \vdots \\ 1 & -2 & 1 & \ddots & & \vdots \\ 0 & 1 & -2 & 1 & & \vdots \\ \vdots & \ddots & & & & \vdots \\ & & & \ddots & & \vdots \\ 0 & \cdots & 0 & 1 & -2 & 1 \end{pmatrix}$$

$$\begin{aligned} \nabla J_\lambda &= 2\mathbf{H}^t[\mathbf{g} - \mathbf{H}\mathbf{f}] + 2\lambda\mathbf{D}^t\mathbf{D}\mathbf{f} = 0 \longrightarrow \\ [\mathbf{H}^t\mathbf{H} + \lambda\mathbf{D}^t\mathbf{D}]\hat{\mathbf{f}} &= \mathbf{H}^t\mathbf{g} \longrightarrow \hat{\mathbf{f}} = [\mathbf{H}^t\mathbf{H} + \lambda\mathbf{D}^t\mathbf{D}]^{-1}\mathbf{H}^t\mathbf{g} \end{aligned}$$

6.6 Algorithmes de régularisation

minimiser $J(\mathbf{f}) = Q(\mathbf{f}) + \lambda\Omega(\mathbf{f})$

avec $Q(\mathbf{f}) = [\mathbf{g} - \mathbf{H}\mathbf{f}]^t[\mathbf{g} - \mathbf{H}\mathbf{f}]$

=

minimiser $\Omega(\mathbf{f})$ sous la contrainte

$$e = [\mathbf{g} - \mathbf{H}\mathbf{f}]^t[\mathbf{g} - \mathbf{H}\mathbf{f}] < \epsilon$$

Information a priori:

- Douceur

$$\Omega(\mathbf{f}) = [\mathbf{D}\mathbf{f}]^t[\mathbf{D}\mathbf{f}] = \mathbf{f}^t \mathbf{D}^t \mathbf{D} \mathbf{f}$$

$$\hat{\mathbf{f}} = [\mathbf{H}^t \mathbf{H} + \lambda \mathbf{P}]^{-1} \mathbf{H}^t \mathbf{g} \quad \text{avec} \quad \mathbf{P} = \mathbf{D}^t \mathbf{D}$$

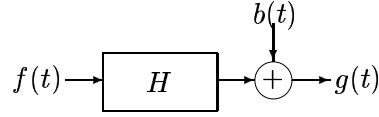
- Positivité $\Omega(\mathbf{f}) =$ fonction non quadratique de \mathbf{f}
Pas de solution explicite

Difficulté: Le calcul de l'expression $[\mathbf{H}^t \mathbf{H} + \lambda \mathbf{D}^t \mathbf{D}]^{-1}$

- approximation circulante
- méthodes récursives
- méthodes itératives

6.6.1 Méthode de HUNT (approximation circulante)

Pour illustrer cette méthode considérons le problème de la déconvolution des signaux :



$$g(t) = \int f(t')h(t-t') dt' + b(t) = \int h(t')f(t-t') dt' + b(t)$$

Nous avons vu que lorsqu'on discrétise cette équation on obtient

$$g(n) = \sum_{i=0}^p h(i)f(n-i) + b(n)$$

ou sous forme matricielle :

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \mathbf{b}$$

où \mathbf{H} est une matrice de TOEPLITZ. Notons que si on définit $\mathbf{h} = [h_0, \dots, h_p]$ on peut noter que $\mathbf{H}\mathbf{f} = \text{conv}(\mathbf{h}, \mathbf{f})$.

La solution régularisée au sens de PTT est définie par

$$\hat{\mathbf{f}} = \arg \min_{\mathbf{f}} \{J(\mathbf{f})\}$$

avec

$$\begin{aligned} J(\mathbf{f}) &= Q(\mathbf{f}) + \lambda\Omega(\mathbf{f}) \\ Q(\mathbf{f}) &= [\mathbf{g} - \mathbf{H}\mathbf{f}]^t [\mathbf{g} - \mathbf{H}\mathbf{f}] = \|\mathbf{g} - \mathbf{H}\mathbf{f}\|^2 \\ \Omega(\mathbf{f}) &= [\mathbf{D}\mathbf{f}]^t [\mathbf{D}\mathbf{f}] = \|\mathbf{D}\mathbf{f}\|^2 \end{aligned}$$

où \mathbf{D} est une matrice correspondant à une opération de filtrage linéaire, donc de convolution par une réponse impulsionnelle, par exemple :

$$\mathbf{d} = [1, -2, 1], \quad \mathbf{D}\mathbf{f} = \text{conv}(\mathbf{d}, \mathbf{f})$$

et on a

$$\Omega(\mathbf{f}) = \sum_{j=1}^N (f(i+1) - 2f(i) + f(i-1))^2 = \mathbf{f}^t \mathbf{D}^t \mathbf{D} \mathbf{f}.$$

La solution régularisée a pour expression $\hat{\mathbf{f}} = [\mathbf{H}^t \mathbf{H} + \lambda \mathbf{D}^t \mathbf{D}]^{-1} \mathbf{H}^t \mathbf{g}$ et la principale difficulté est le coût de calcul de l'inverse de la matrice $[\mathbf{H}^t \mathbf{H} + \lambda \mathbf{D}^t \mathbf{D}]$. Cette matrice est symétrique, mais, dans le cas général n'a pas d'autres propriétés remarquables. Dans le cas particulier où \mathbf{H} et \mathbf{D} sont des matrices de TOEPLITZ, cette matrice est aussi TOEPLITZ.

L'idée de base est d'étendre les vecteurs \mathbf{f} , \mathbf{h} et \mathbf{g} par des zéros :

$$\mathbf{f}_e(i) = \begin{cases} f(i) & i = 1, \dots, Nf \\ 0 & i = Nf + 1, \dots, P \geq Nf + Nh - 1 \end{cases}$$

$$g_e(i) = \begin{cases} g(i) & i = 1, \dots, Ng \\ 0 & i = Ng + 1, \dots, P \end{cases}$$

$$h_e(i) = \begin{cases} h(i) & i = 1, \dots, Nh \\ 0 & i = Nh + 1, \dots, P \end{cases}$$

pour obtenir une relation $\mathbf{g}_e = \mathbf{H}_e \mathbf{f}_e$ avec \mathbf{H}_e une matrice circulante que l'on peut diagonaliser par transformée de FOURIER discrète (TFD). En effet, pour une matrice circulante \mathbf{H}_e on a

$$\mathbf{H}_e = \mathbf{F} \mathbf{\Lambda} \mathbf{F}^{-1}, \quad \mathbf{\Lambda} = \text{diag}[\lambda_1, \dots, \lambda_P]$$

avec

$$\mathbf{F}[k, l] = \exp \left[j2\pi \frac{kl}{P} \right] \quad \mathbf{F}^{-1} = \frac{1}{P} \exp \left[-j2\pi \frac{kl}{P} \right]$$

$$[\lambda_1, \dots, \lambda_P] = \text{TFD} [h_1, \dots, h_{Nh}, 0, \dots, 0]$$

De même, on peut étendre le vecteur $\mathbf{d} = [1, -2, 1]$

$$d_e(i) = \begin{cases} d(i) & i = 1, \dots, 3 \\ 0 & i = 4, \dots, P \end{cases}$$

de telle sorte que l'on puisse écrire les relations suivantes :

$$\widehat{\mathbf{f}} = [\mathbf{H}^t \mathbf{H} + \lambda \mathbf{D}^t \mathbf{D}]^{-1} \mathbf{H}^t \mathbf{g}$$

$$\mathbf{F} \widehat{\mathbf{f}}_e = [\Lambda_h^t \Lambda_h + \lambda \Lambda_c^t \Lambda_c]^{-1} \Lambda_h^t \mathbf{F} \mathbf{g}$$

$$\text{TFD} \{ \mathbf{f}_e \} = [\Lambda_h^t \Lambda_h + \lambda \Lambda_c^t \Lambda_c]^{-1} \Lambda_h^t \text{TFD} \{ \mathbf{g} \}$$

$$\widehat{F}(\omega) = \frac{1}{H(\omega)} \frac{|H(\omega)|^2}{|H(\omega)|^2 + \lambda |C(\omega)|^2} G(\omega)$$

Lien avec le filtre de WIENER :

$$C(\omega) = S_{bb}(\omega) / S_{ff}(\omega)$$

Restauration d'image :

D matrice de convolution par une réponse impulsionnelle

$$H_1 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

$$\Omega(\mathbf{f}) = \sum \sum (f(i+1, j) + f(i-1, j) + f(i, j+1) + f(i, j-1) - 4f(i, j))^2$$

$$f_e(k, l) = \begin{cases} f(k, l) & k = 1, \dots, K \quad l = 1, \dots, L \\ 0 & k = K+1, \dots, P \quad l = L+1, \dots, P \end{cases}$$

$$F(\omega_x, \omega_y) = \frac{1}{H(\omega_x, \omega_y)} \frac{|H(\omega_x, \omega_y)|^2}{|H(\omega_x, \omega_y)|^2 + \lambda |C(\omega_x, \omega_y)|^2} G(\omega_x, \omega_y)$$

6.6.2 Méthodes itératives

Nous avons vu que la solution régularisée est définie comme l'argument qui minimise un critère $J(\mathbf{f})$:

$$\hat{\mathbf{f}} = \arg \min_{\mathbf{f}} \{J(\mathbf{f}) = Q(\mathbf{f}) + \lambda\Omega(\mathbf{f})\}$$

Lorsque le critère est unimodal, on peut alors utiliser les algorithmes d'optimisation classique du type gradient pour calculer la solution. Notant par :

$$\mathbf{g} = \nabla J(\mathbf{f}) = \left[\frac{\partial J}{\partial f_1}, \dots, \frac{\partial J}{\partial f_n} \right]^t \quad \text{le gradient de } J, \quad \mathbf{g}_k = \nabla J(\mathbf{f}_k)$$

et par

$$\mathbf{H} = \nabla^2 J(\mathbf{f}) = \left\{ \frac{\partial^2 J}{\partial f_i \partial f_j} \right\} \quad \text{la matrice Hessienne de } J, \quad \mathbf{H}_k = \nabla^2 J(\mathbf{f}_k)$$

La majorité des méthodes d'optimisation locale calcule la solution d'une manière itérative suivante :

$$\begin{aligned} \mathbf{f}_{k+1} &= \mathbf{f}_k - \alpha_k \mathbf{D}_k \nabla J(\mathbf{f}_k) \\ &= \mathbf{f}_k - \alpha_k \mathbf{D}_k \mathbf{g}_k \\ &= \mathbf{f}_k + \alpha_k \mathbf{d}_k \end{aligned}$$

Suivant les différents choix pour α_k et \mathbf{D}_k on peut distinguer les algorithmes suivants :

1. Méthodes de premier ordre ou de gradient

- gradient à pas fixe : $\mathbf{D}_k = \mathbf{I}$, $\alpha_k = \alpha$ et $\mathbf{d}_k = -\mathbf{g}_k$
- gradient à pas variable : $\mathbf{D}_k = \mathbf{I}$ et

$$\alpha_k = -\frac{\mathbf{g}^{(k)t} \mathbf{g}_k}{\mathbf{g}^{(k)t} \mathbf{H}_k \mathbf{g}_k} = \frac{\|\mathbf{g}_k\|^2}{\|\mathbf{g}_k\|_H^2}$$

- gradient conjugué :

$$\mathbf{f}_{k+1} = \mathbf{f}_k + \alpha_k \mathbf{d}_k \quad \alpha_k = -\frac{\mathbf{d}^{(k)t} \mathbf{g}_k}{\mathbf{d}^{(k)t} \mathbf{H}_k \mathbf{d}_k}$$

$$\mathbf{d}_{k+1} = \mathbf{d}_k + \beta_k \mathbf{g}_k \quad \beta_k = -\frac{\mathbf{g}^{(k)t} \mathbf{g}_k}{\mathbf{g}^{(k-1)t} \mathbf{g}^{(k-1)}}$$

2. Méthodes de second ordre

– Méthode de Newton : $\mathbf{D}_k = [\mathbf{H}_k]^{-1}$

$$\mathbf{f}_{k+1} = \mathbf{f}_k - \alpha_k [\mathbf{H}_k]^{-1} \mathbf{g}_k$$

– Méthode de Newton approchée : $\mathbf{D}_k = \text{diag}\{d_{1k}, \dots, d_{nk}\}$ avec

$$d_{i_k} = \mathbf{D}_{ii} = \left(\frac{\partial^2 J}{\partial f_i^2} \right)^{-1}$$

les éléments diagonaux de la matrice Hessienne.

– Méthodes de Newton modifiées :

$$\mathbf{D}_k = \mathbf{D}_0 = [\mathbf{H}_0]^{-1} = \mathbf{D}$$

ou

$$\mathbf{D}_k = \mathbf{D}_0 = \text{diag}\{d_{1_0}, \dots, d_{n_0}\} \quad \text{avec} \quad d_{i_0} = \left(\frac{\partial^2 J}{\partial f_i^2}(\mathbf{f}_0) \right)^{-1}$$

- Avantage : $\Omega(\mathbf{f})$ peut être quelconque
- Inconvénient : Coût de calcul

6.6.3 Méthodes récursives

Considérons l'équation $\mathbf{g} = \mathbf{H}\mathbf{f} + \mathbf{b}$ et sa solution régularisée :

$$\hat{\mathbf{f}} = [\mathbf{H}^t \mathbf{H} + \lambda \mathbf{D}]^{-1} \mathbf{H}^t \mathbf{g}$$

Notons par \mathbf{H}_i la matrice \mathbf{H} lorsqu'on a observé les données jusqu'à l'indice i ; \mathbf{g}_i et par \mathbf{H}_{i+1} la matrice \mathbf{H} lorsqu'on a observé les données jusqu'à l'indice i ; \mathbf{g}_{i+1} . On a alors

$$\mathbf{g}_i = \begin{pmatrix} g_1 \\ \vdots \\ g_i \end{pmatrix} = \mathbf{H}_i \mathbf{f} + \mathbf{b},$$

$$\mathbf{g}_{i+1} = \begin{pmatrix} g_1 \\ \vdots \\ g_i \\ g_{i+1} \end{pmatrix} = \begin{pmatrix} \mathbf{g}_i \\ \dots \\ g_{i+1} \end{pmatrix} = \begin{pmatrix} \mathbf{H}_i \\ \dots \\ \mathbf{h}_{i+1}^t \end{pmatrix} \mathbf{f} + \mathbf{b}$$

L'idée de base consiste à exprimer la solution \mathbf{f}_{i+1}

$$\begin{aligned} \mathbf{f}_{i+1} &= (\mathbf{H}_{i+1}^t \mathbf{H}_{i+1} + \lambda \mathbf{D})^{-1} \mathbf{H}_{i+1}^t \mathbf{g}_{i+1} \\ &= \left[\begin{pmatrix} \mathbf{H}_i^t & \vdots & \mathbf{h}_{i+1}^t \end{pmatrix} \begin{pmatrix} \mathbf{H}_i \\ \dots \\ \mathbf{h}_{i+1}^t \end{pmatrix} + \lambda \mathbf{D} \right]^{-1} \begin{pmatrix} \mathbf{H}_i^t & \vdots & \mathbf{h}_{i+1}^t \end{pmatrix} \begin{pmatrix} \mathbf{g}_i \\ \dots \\ g_{i+1} \end{pmatrix} \end{aligned}$$

en fonction de \mathbf{f}_i :

$$\mathbf{f}_i = (\mathbf{H}_i^t \mathbf{H}_i + \lambda \mathbf{D})^{-1} \mathbf{H}_i^t \mathbf{g}_i$$

Utilisant la lemme d'inversion des matrices on obtient facilement :

$$\mathbf{f}_{i+1} = (\mathbf{H}_i^t \mathbf{H}_i + \mathbf{h}_{i+1}^t \mathbf{h}_{i+1} + \lambda \mathbf{D})^{-1} (\mathbf{H}_i^t \mathbf{g}_i + \mathbf{h}_{i+1} g_{i+1})$$

qui peut être réécrite d'une manière plus familière en notant :

$$\mathbf{P}_i = (\mathbf{H}_i^t \mathbf{H}_i + \lambda \mathbf{D})^{-1}$$

$$\mathbf{P}_{i+1}^t = \mathbf{P}_i^t + \mathbf{H}_{i+1} \mathbf{H}_{i+1}^t$$

↓

$$\mathbf{f}_{i+1} = \mathbf{f}_i + \mathbf{P}_{i+1} \mathbf{H}_{i+1} (g_{i+1} - \mathbf{H}_{i+1} \mathbf{f}_i)$$

$$\mathbf{P}_{i+1} = \mathbf{P}_i - \mathbf{P}_i \mathbf{H}_{i+1} (\mathbf{H}_{i+1}^t \mathbf{P}_i \mathbf{H}_{i+1} + \lambda)^{-1} \mathbf{H}_{i+1}^t \mathbf{P}_i$$

Filtre de Kalman

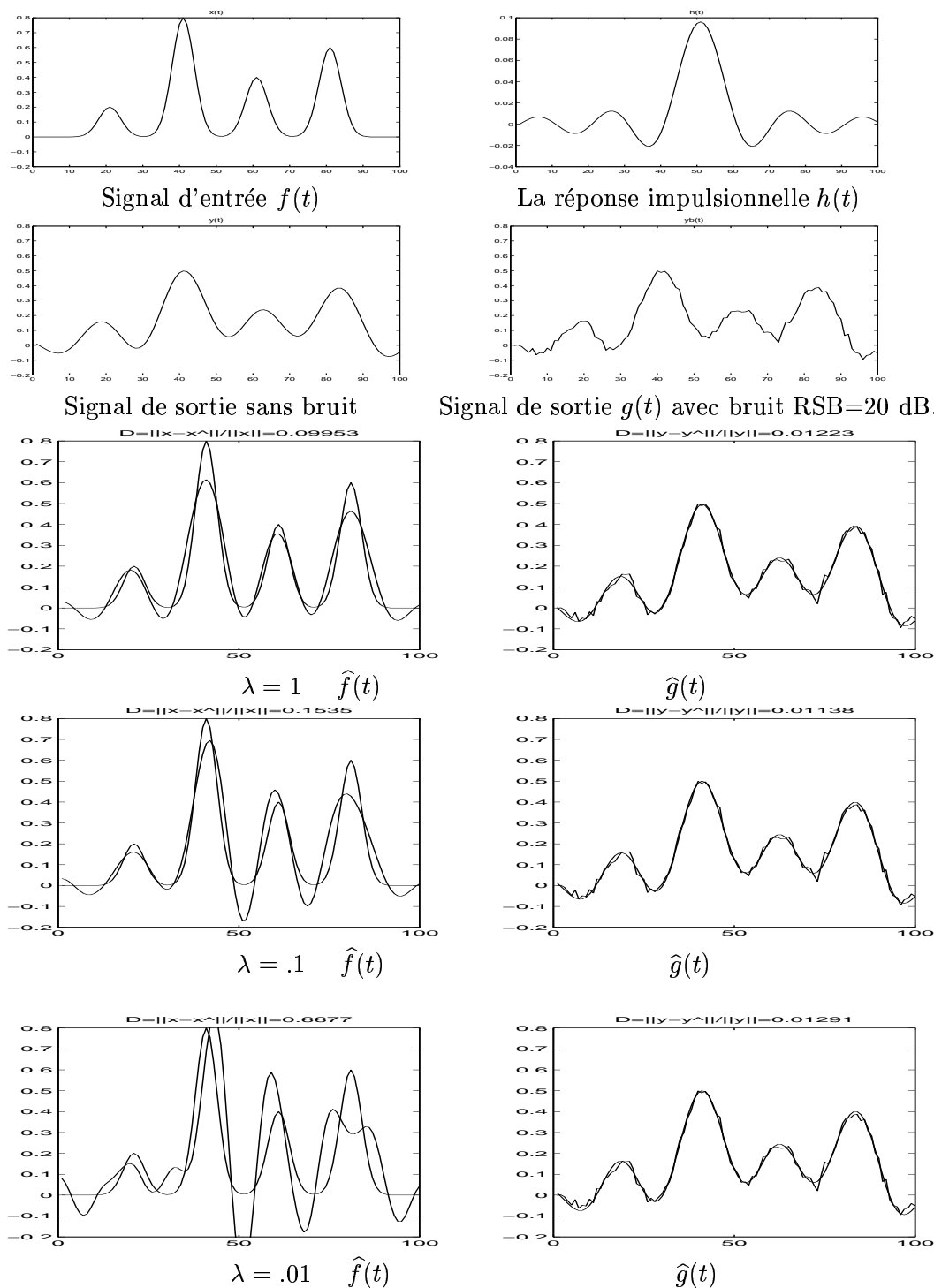


FIG. 6.1 – Déconvolution par la méthode de Hunt : Exemple 1

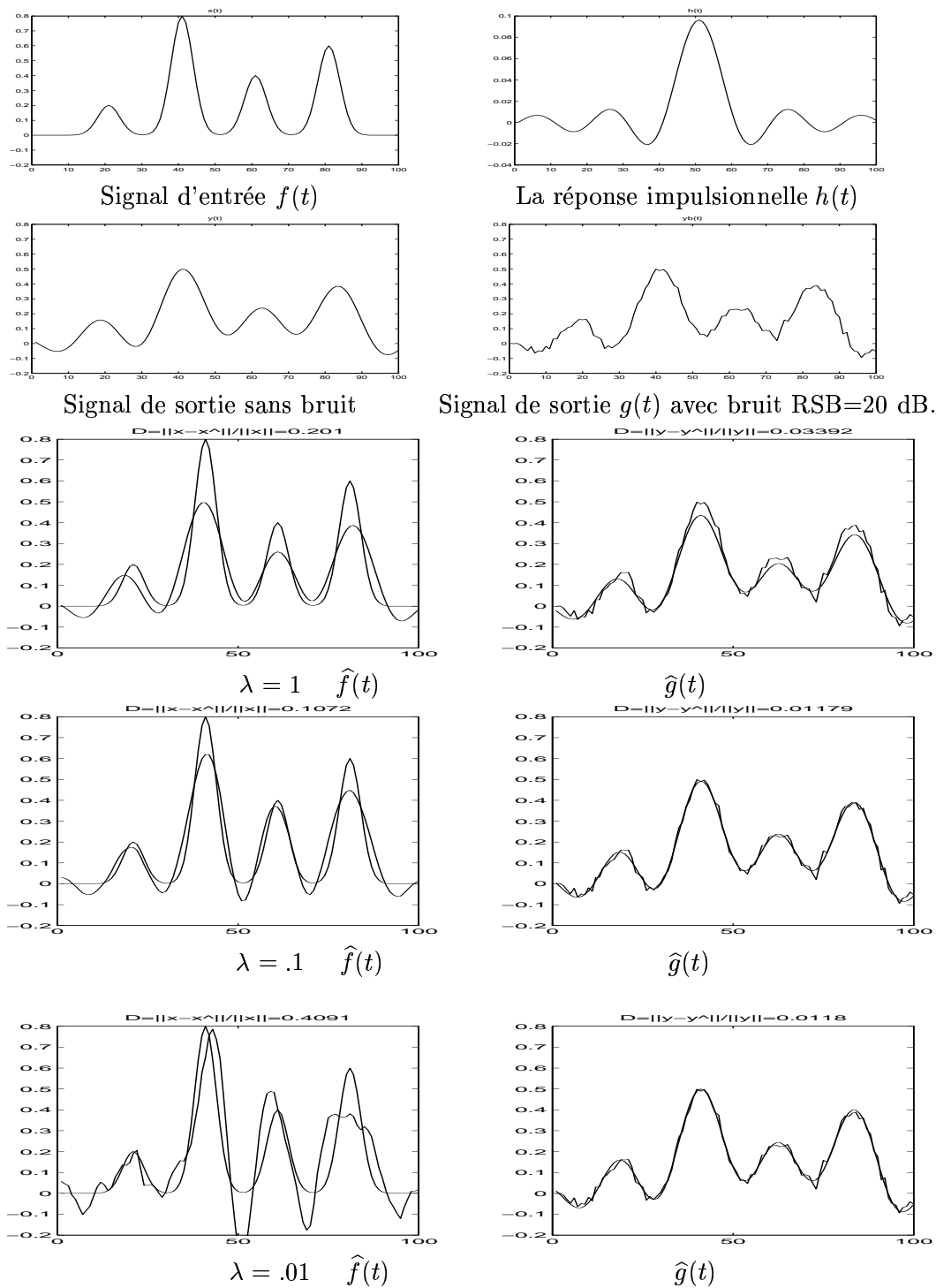


FIG. 6.2 – Déconvolution par filtre de Wiener : Exemple 1

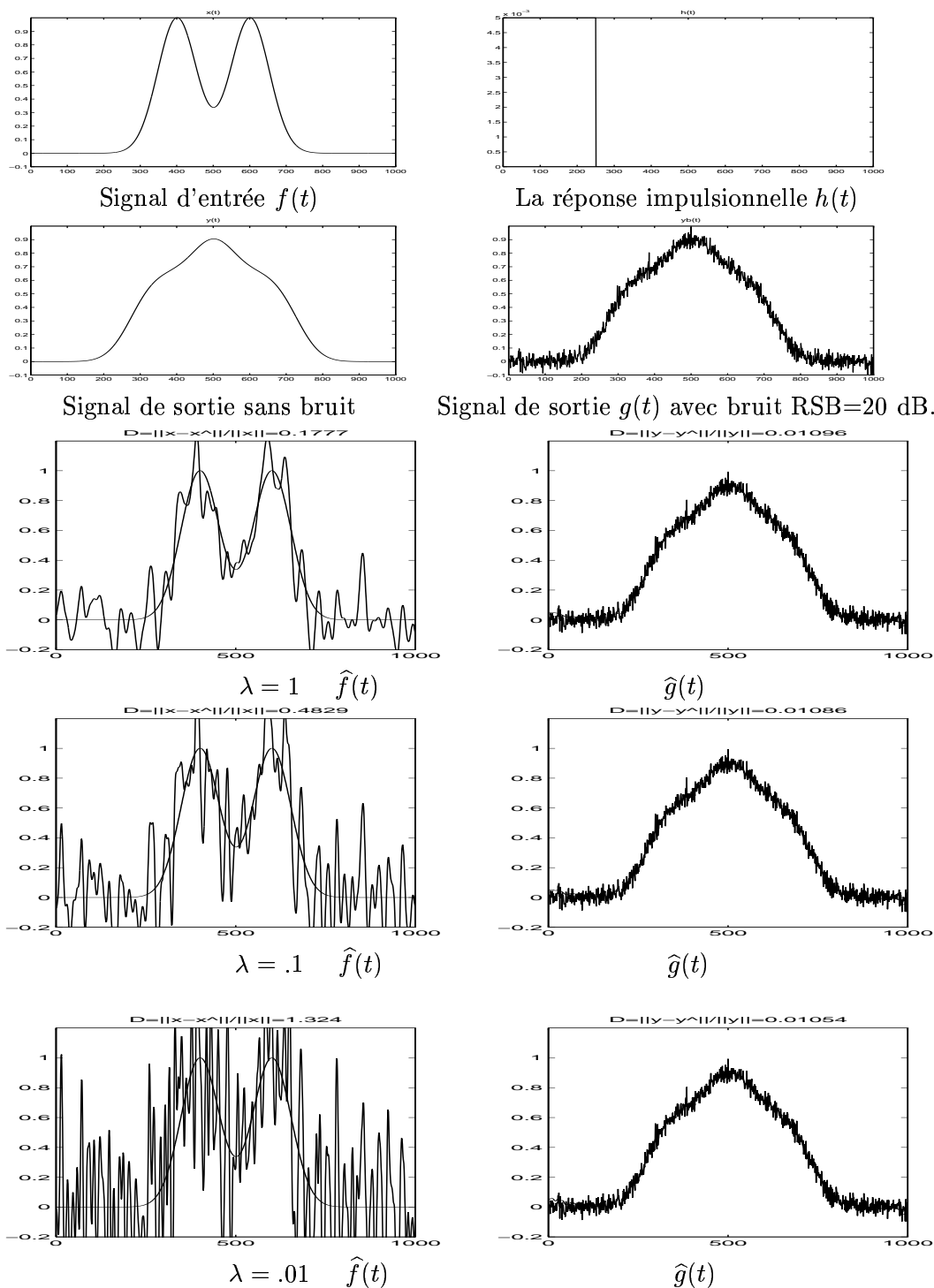


FIG. 6.3 – Déconvolution par la méthode de Hunt : Exemple 2

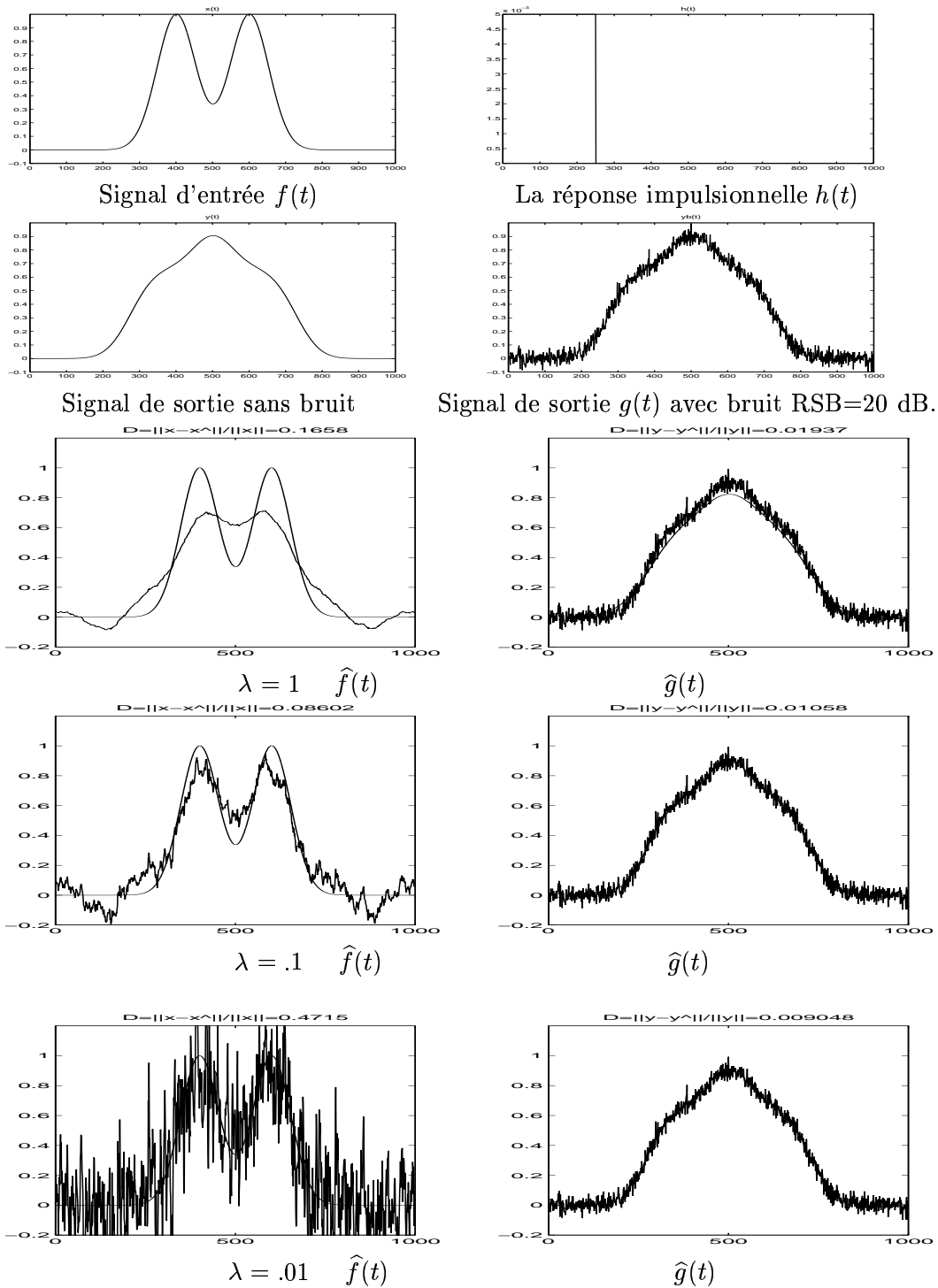


FIG. 6.4 – Déconvolution par filtre de Wiener : Exemple 2

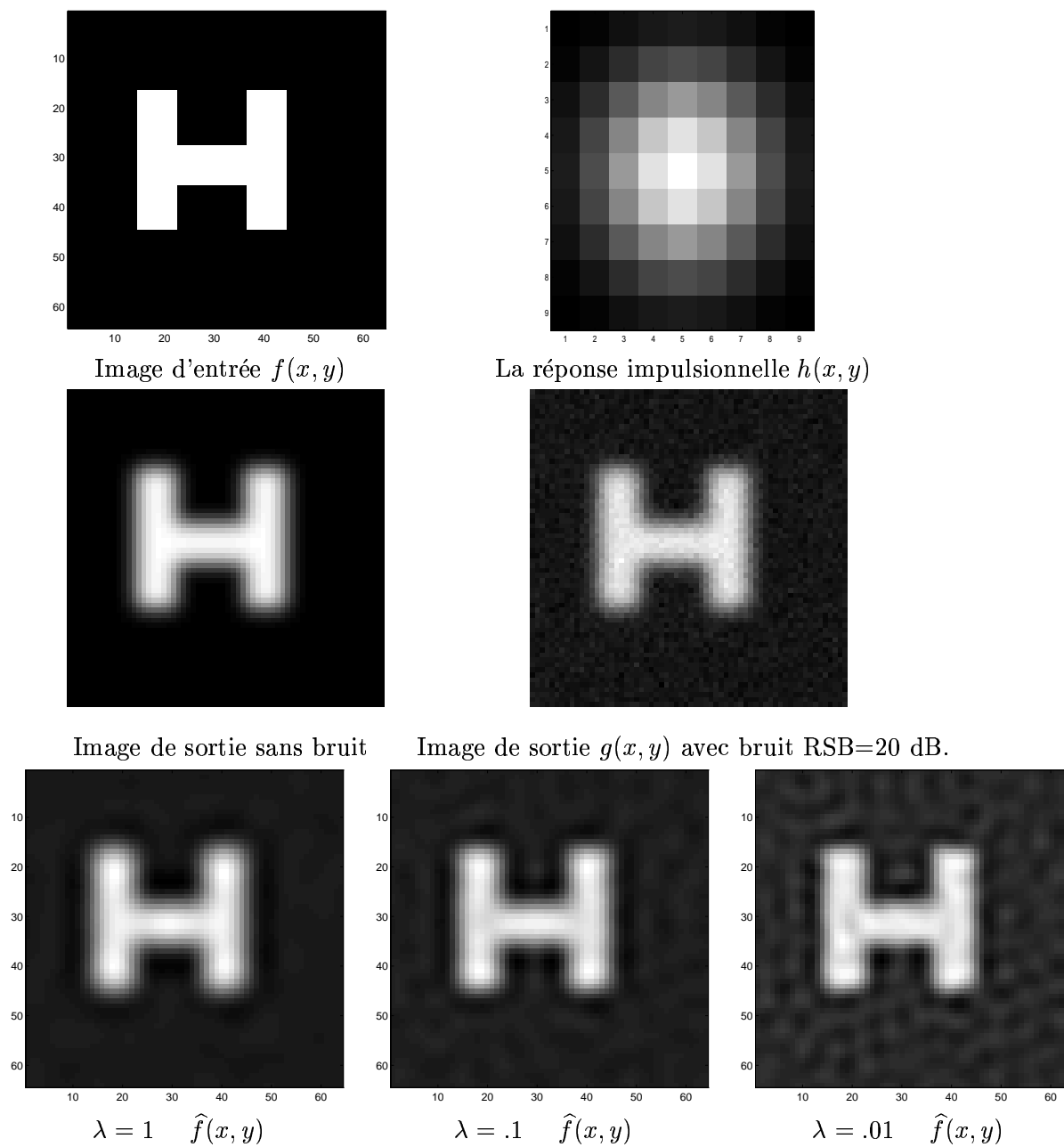


FIG. 6.5 – Restauration par la méthode de Hunt: Exemple 1

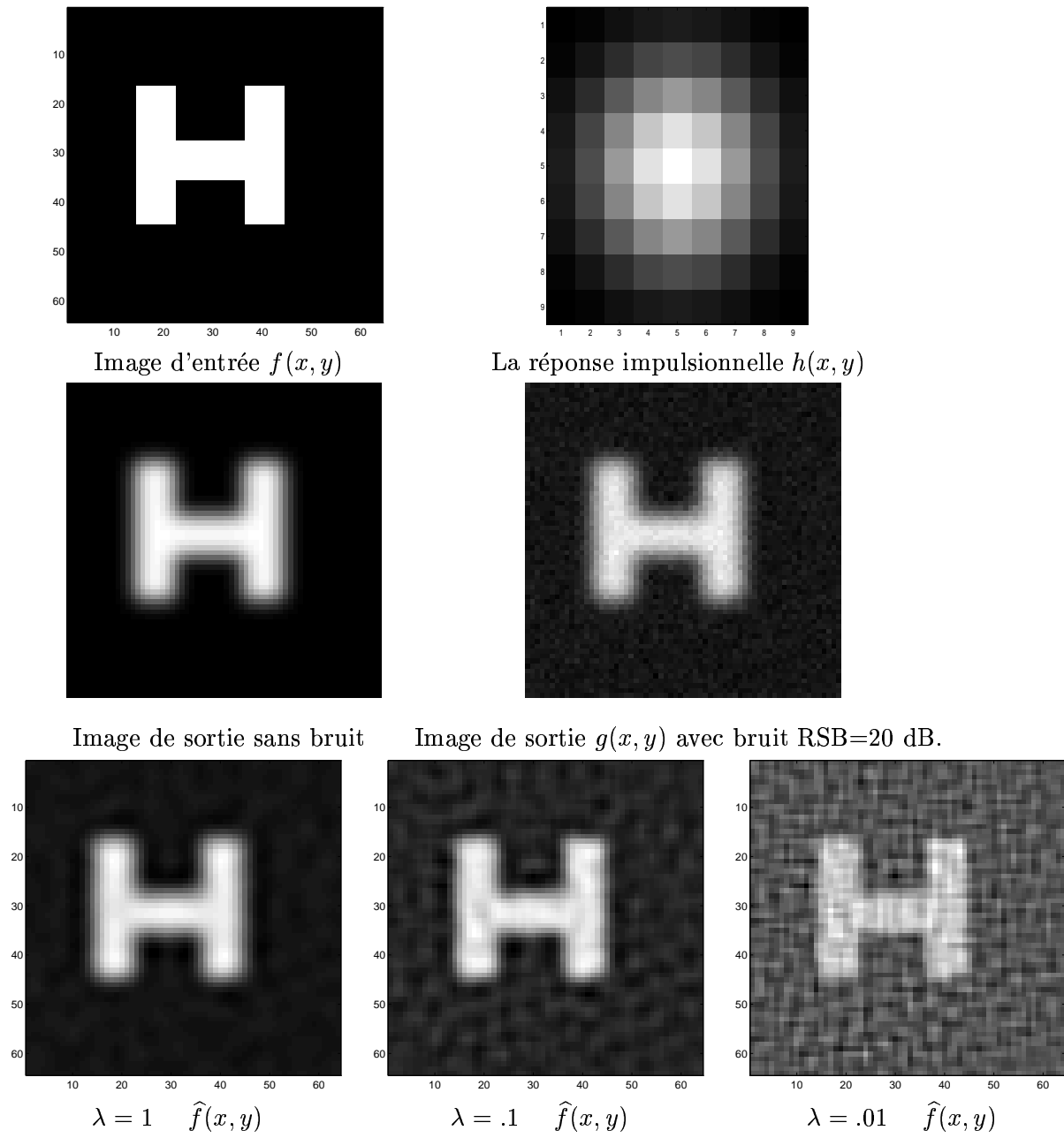


FIG. 6.6 – Restauration par filtre de Wiener: Exemple 1

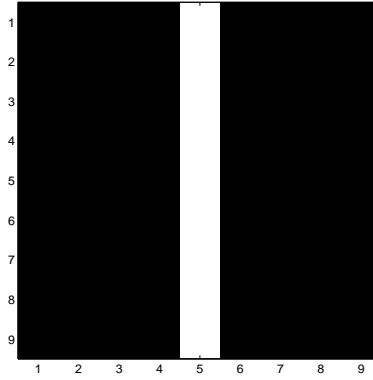
Image d'entrée $f(x, y)$ La réponse impulsionnelle $h(x, y)$ 

Image de sortie sans bruit

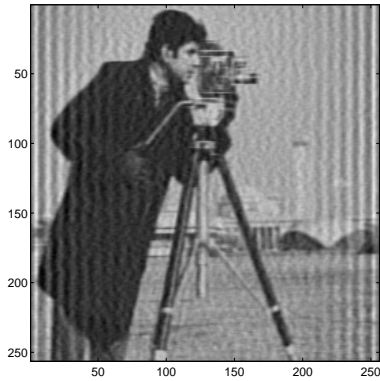
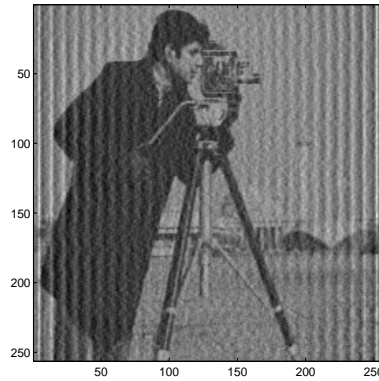
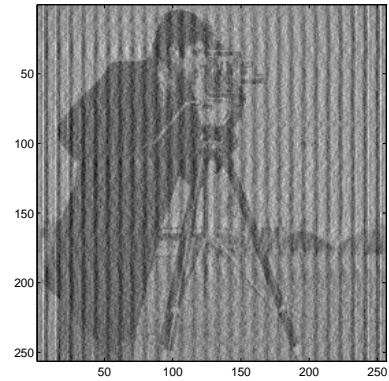
Image de sortie $g(x, y)$ avec bruit RSB=20 dB. $\lambda = 1 \quad \hat{f}(x, y)$  $\lambda = .1 \quad \hat{f}(x, y)$  $\lambda = .01 \quad \hat{f}(x, y)$

FIG. 6.7 - Restauration par la méthode de Hunt: Exemple 2

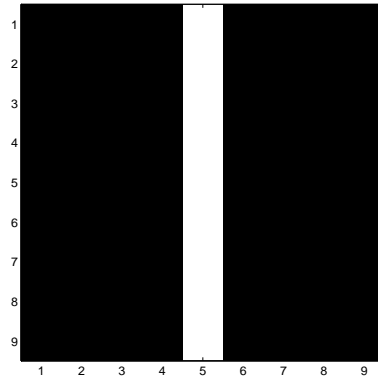
Image d'entrée $f(x, y)$ La réponse impulsionnelle $h(x, y)$ 

Image de sortie sans bruit

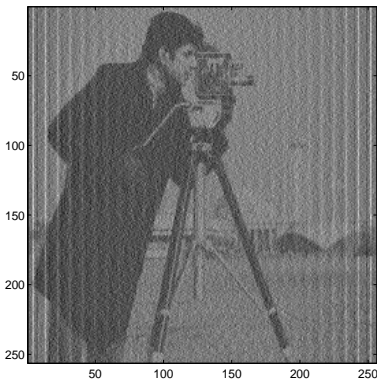
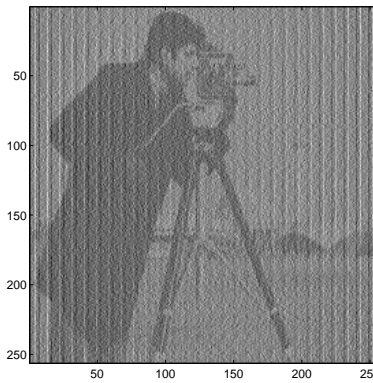
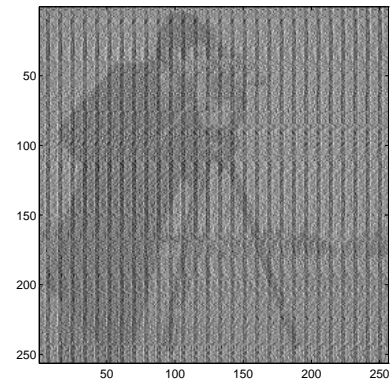
Image de sortie $g(x, y)$ avec bruit RSB=20 dB. $\lambda = 1 \quad \hat{f}(x, y)$  $\lambda = .1 \quad \hat{f}(x, y)$  $\lambda = .01 \quad \hat{f}(x, y)$

FIG. 6.8 – Restauration par filtre de Wiener : Exemple 2