Problèmes inverses et imagerie médicale

Master 2 Math En Action - Université de Lyon 1 - Ecole Centrale de Lyon

E. Bretin & L. Seppecher

24 septembre 2025

Chapitre 1

Introduction

1 Problème inverse, définition et premiers exemples

D'un point de vue abstrait, le problème direct permet de déduire les effets lorsque les causes sont connues, tandis qu'un problème inverse consiste à déterminer les causes à partir de la connaissance des effets. Deux problèmes sont dits inverses si la formulation de l'un fait intervenir l'autre.

Par exemple, la prédiction de l'état futur d'un système physique connaissant son état actuel est considérée comme un problème direct, alors que la reconstitution de l'état passé connaissant le présent ou encore la détermination de paramètres physiques à partir d'observations sont des caractéristiques de problèmes inverses.

1.1 Problème de conductivité

Dans un premier exemple, on peut s'intéresser à la solution u du problème elliptique suivant :

$$\begin{cases} -(a(x)u'(x))' = f(x), & \text{pour } x \in]-1,1[, \\ u(-1) = u(1) = 0, \end{cases}$$

où

- le coefficient de diffusion a est supposé suffisamment régulier et strictement positif,
- la source f est supposée également suffisamment régulière.

Dans cet exemple, le problème direct consiste à déterminer la solution u lorsque a et f sont connus. En particulier, dans le cas $a = 1$ et $f = 1$, u s'identifie à la parabole $u(x) = \frac{1}{2}(1 - x^2)$.

Le problème inverse consiste à reconstruire le coefficient de diffusion a à partir d'une mesure de la solution u pour un choix de f donné. Il est alors possible de montrer que le coefficient de diffusion a est de la forme

$$a(x) = \frac{c}{u'(x)} - \frac{1}{u'(x)} \int_0^x f(s)\, ds,$$

où $c \in \mathbb{R}$. Il y a donc un problème d'unicité dans la reconstruction de a, ce qui nécessite d'introduire des hypothèses supplémentaires sur a. Par exemple, dans le cas simplifié où $a = 1$ et $f = 1$, on trouve la formule

$$a(x) = -\frac{c}{x} + \frac{1}{x} \int_0^x 1\, ds = 1 - \frac{c}{x},$$

qui montre que 1 est bien la seule solution bornée du problème inverse. En revanche, cette formule est inexploitable numériquement à cause de problèmes d'arrondi, notamment en $x = 0$, et soulève la notion de problème inverse bien posé.

1.2 Problème inverse bien posé

Définition 1

Un problème inverse est dit bien posé au sens de Hadamard si

- *la solution existe,*
- *elle est unique,*
- *elle dépend continûment des données.*

1.3 Résolution d'un système linéaire en dimension finie

Nous nous intéressons maintenant au cas d'un **problème inverse linéaire en dimension finie**. Le problème direct revient alors à déterminer le vecteur $b = Ax \in \mathbb{R}^m$ à partir de la connaissance du vecteur $x \in \mathbb{R}^n$ et de la matrice $A \in \mathbb{R}^{m\times n}$. Quant au problème inverse, il consiste à résoudre ce système linéaire en déterminant x à partir de A et b.

En particulier, une solution existe si b appartient à l'image de A ($b \in \mathrm{Im}(A)$). L'unicité nécessite que le noyau de A soit trivial ($\ker(A) = \{0\}$). Enfin, la continuité de la solution par rapport aux données est liée au conditionnement de la matrice A.

En effet, si x et $x + \delta x$ sont les solutions des systèmes linéaires

$$\begin{cases} Ax = b, \\ (A + \delta A)(x + \delta x) = b + \delta b, \end{cases}$$

alors il est possible de montrer que

$$\frac{\|\delta x\|}{\|x\|} \leqslant \frac{\mathrm{cond}(A)}{1 - \|A\|\,\|\delta A\|} \left(\frac{\|\delta b\|}{\|b\|} + \frac{\|\delta A\|}{\|A\|} \right),$$

où $\mathrm{cond}(A) = \|A\|\,\|A^{-1}\|$.

En particulier, la résolution numérique d'un système linéaire dont le conditionnement dépasse 10^{16} pose des difficultés lorsque l'on utilise une arithmétique en double précision.

1.4 Algorithme de déconvolution

Un troisième exemple est celui des algorithmes de déconvolution, qui entrent dans la classe des **problèmes inverses linéaires en dimension infinie**. On s'intéresse en particulier à la solution u de l'équation de la chaleur

$$\begin{cases} u_t = \Delta u, \quad \text{pour } (x,t) \in \mathbb{R} \times [0,T], \\ u(x,0) = f(x) \in L^2(\mathbb{R}). \end{cases}$$

Le problème inverse consiste à retrouver la condition initiale f à partir de la connaissance de la solution u à l'instant $t = T$. En particulier, il est possible de montrer, en utilisant la transformée de Fourier, que

$$u(x,T) = K_T * f(x),$$

où K_t est le noyau de la chaleur défini en dimension 1 par

$$K_t(x) = \frac{1}{\sqrt{4\pi t}} \exp\left(-\frac{x^2}{4t} \right) = \mathcal{F}^{-1}\left[e^{-4\pi^2|\xi|^2 t} \right](x).$$

```
clf; N = 2^8;
x = linspace(0,1,N);
f = 1 + cos(4*pi*x) ; T = 0.0001;
plot(x,f,'g');
xi = [0:N/2,-N/2+1:-1];
u_T = real(ifft(exp(-4*pi^2*T*abs(xi).^2).*fft(f)));
u_T = u_T.*(1 + 0.001*randn(size(u_T)));
hold on; plot(x,u_T);
f_recons = real(ifft(exp(+4*pi^2*T*abs(xi).^2).*fft(u_T)));
plot(x,f_recons,'r');
```

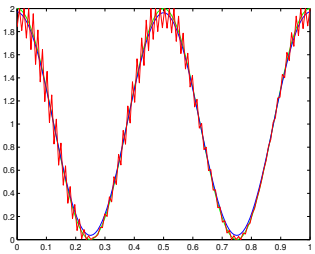

FIGURE 1.1 – Algorithme de déconvolution : code **Matlab** et image de la reconstruction de la source f (en vert f, en bleu $u(x,T)$ et en rouge la reconstruction de f).

Le problème inverse revient alors à utiliser un algorithme de déconvolution. On peut en effet obtenir une formule de reconstruction en utilisant la transformée de Fourier spatiale de $u(x,T)$, que l'on multiplie par le symbole $e^{4\pi^2|\xi|^2 T}$:

$$f_{\text{reconstr}}(x) = \mathcal{F}^{-1}\left[e^{4\pi^2|\xi|^2 T}\,\hat{u}(\xi,T)\right].$$

Le problème est que, lorsqu'on teste numériquement cette formule de reconstruction (voir figure 1.4), la solution présente de fortes instabilités et nécessite l'introduction de techniques de régularisation.

Une première idée consiste à utiliser une approche de type Tikhonov, qui revient à minimiser la fonctionnelle

$$J_1(f) = \|K_T * f - u(\cdot,T)\|^2 + \varepsilon^2 \|f\|_2^2,$$

où ε est un paramètre de régularisation. Le minimiser de cette fonctionnelle est explicite et s'écrit

$$f_{\varepsilon,1}(x) = \mathcal{F}^{-1}\left[\frac{e^{-4\pi^2|\xi|^2 T}}{\varepsilon^2 + e^{-8\pi^2|\xi|^2 T}}\,\hat{u}(\xi,T)\right].$$

Une autre idée est d'exploiter la régularité attendue de la source f que l'on souhaite reconstruire, en minimisant une fonctionnelle de la forme

$$J_2(f) = \|K_T * f - u(\cdot,T)\|^2 + \varepsilon^2 \|\nabla f\|_2^2,$$

dont la solution est également explicite et vérifie

$$f_{\varepsilon,2}(x) = \mathcal{F}^{-1}\left[\frac{e^{-4\pi^2|\xi|^2 T}}{\varepsilon^2\, 4\pi^2|\xi|^2 + e^{-8\pi^2|\xi|^2 T}}\,\hat{u}(\xi,T)\right].$$

La figure 1.2 présente des exemples de reconstruction obtenus en appliquant l'expression de $f_{\varepsilon,2}$.

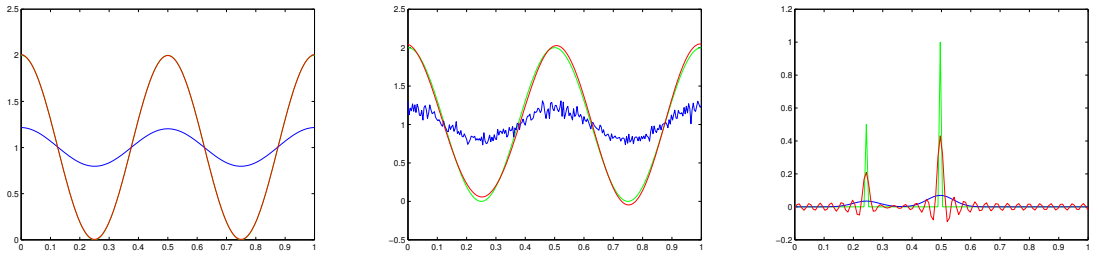

FIGURE 1.2 – Algorithme de déconvolution : exemples de reconstructions obtenues avec des techniques de régularisation à partir de données bruitées (en vert f, en bleu $u(x,T)$ et en rouge $f_{\varepsilon,2}$).

2 Plan du cours

- **Présentation de quelques problèmes inverses en traitement d'image et en imagerie médicale :**
  Il s'agira de présenter quelques exemples classiques de problèmes inverses linéaires et non linéaires rencontrés en imagerie, et plus précisément en imagerie médicale. Certains problèmes seront linéaires, avec des formules de reconstruction explicites, d'autres seront non linéaires et nécessiteront l'utilisation de formulations variationnelles ainsi que d'algorithmes de gradient pour approcher leur solution. La plupart de ces problèmes soulèvent des difficultés de stabilité qui nécessitent l'introduction de techniques de régularisation, comme nous l'avons déjà vu dans un premier exemple.

- **Formulation variationnelle, méthode du gradient et méthode de l'état adjoint :**
  Comme indiqué précédemment, la solution des problèmes inverses est rarement explicite. Un cadre général consiste à rechercher la solution sous une forme variationnelle, c'est-à-dire comme le minimum d'une fonctionnelle composée d'un terme d'attache aux données, couplé à un terme de régularisation. Ces approches nécessitent l'utilisation d'algorithmes de descente de type gradient et donc le calcul du gradient de l'énergie. Les formules de gradient ne sont pas explicites, notamment dans le cas de problèmes inverses non linéaires, et nécessitent alors en pratique la méthode dite de l'état adjoint.

- **Exemples de régularisation d'opérateurs : SVD / Tikhonov**
  Dans le cas de problèmes inverses linéaires, une première technique de régularisation consiste à utiliser une troncation spectrale de l'opérateur linéaire associé, à partir de sa décomposition en valeurs singulières (SVD). L'objectif est de montrer le lien entre cette approche et les techniques de régularisation de type Tikhonov, ainsi qu'avec les itérés de la méthode du gradient.

- **Régularisation H^1 (et variantes) sur la solution :**
  Les régularisations de type Tikhonov ne sont pas toujours efficaces, et il est nécessaire d'exploiter davantage les propriétés de régularité de la fonction f que l'on souhaite reconstruire. Une première idée consiste à ajouter la norme H^1 de f au terme d'attache aux données, afin de sélectionner uniquement des solutions régulières à partir des données. Le cas de fonctions régulières par morceaux soulèvera cependant d'autres difficultés, car la norme H^1 tend à lisser les discontinuités. Nous introduirons alors d'autres variantes, notamment l'approche de Perona–Malik ou encore la fonctionnelle de Mumford–Shah, qui sont classiquement utilisées en traitement d'image.

- **Régularisation TV (variation totale) sur la solution :**
  Les approches précédentes conduisent naturellement à s'intéresser à une minimisation de type variation totale, qui consiste à introduire la norme L^1 du gradient de f (au lieu de la norme L^2 dans le cas H^1). L'intérêt de cette méthode est de rechercher des solutions régulières tout en autorisant des discontinuités. D'un point de vue numérique, cette approche soulève des difficultés, car le terme de ré-

gularisation n'est plus lisse, et nécessite l'utilisation de techniques duales de résolution. L'objectif sera ici de présenter et d'étudier la convergence de l'algorithme de Chambolle pour le débruitage d'images.

- **Algorithmes ISTA–FISTA : techniques d'accélération pour les énergies non lisses :**
  Comme expliqué précédemment, les reconstructions s'obtiennent par la minimisation d'une énergie composée

  ▷ d'un terme d'attache aux données, généralement lisse et pouvant être traité explicitement,

  ▷ et d'un terme de régularisation non lisse, qui nécessite un traitement implicite (par une approche duale).

  Les algorithmes de gradient combinant de manière alternée ces deux étapes de minimisation (ISTA) ne sont pas optimaux en pratique. Nous verrons comment il est possible d'accélérer la convergence de ce type d'algorithmes de descente, notamment grâce à une approche de type FISTA.

Chapitre 2

Exemples de problèmes inverses appliqués à l'imagerie médicale

1 Imagerie par rayons X et transformée de Radon

Un exemple classique de problème inverse rencontré en imagerie médicale provient de l'imagerie par rayons X. L'idée consiste à mesurer la carte d'atténuation $A(x)$ d'un tissu biologique en émettant un rayonnement X dans des directions données. Le faisceau de rayons X traversant le tissu est atténué, et la quantité de photons n'ayant pas traversé le milieu est proportionnelle à l'intégrale du coefficient d'absorption $A(x)$ le long de la droite empruntée par le rayon X (voir figure 2.1).

On introduit alors la transformée de Radon de A, notée $\mathcal{R}[A]$, définie pour tout $(\theta, t) \in [0, \pi] \times \mathbb{R}$ par

$$\mathcal{R}[A](\theta, t) = \int_{\Delta_{\theta,t}} A \, ds = \int_{\mathbb{R}} A(t\mathbf{e}_\theta^\perp + s\mathbf{e}_\theta) \, ds,$$

où

$$\mathbf{e}_\theta = (\cos(\theta), \sin(\theta)), \quad \mathbf{e}_\theta^\perp = (-\sin(\theta), \cos(\theta)),$$

et $\Delta_{\theta,t} = \{ t\mathbf{e}_\theta^\perp + s\mathbf{e}_\theta \ ; \ s \in \mathbb{R} \}$.

Le problème inverse rencontré en imagerie par rayons X consiste donc à reconstruire l'atténuation A à partir des données complètes ou partielles de $\mathcal{R}[A]$. Il s'agit donc d'un **problème inverse linéaire en dimension finie**.

Dans la suite, nous allons montrer comment obtenir des formules de reconstruction de manière assez simple dans le cas de données complètes et régulières. On suppose que $f \in \mathcal{S}(\mathbb{R}^2)$ et que l'opérateur $\mathcal{R}_\theta : \mathcal{S}(\mathbb{R}^2) \mapsto \mathcal{S}(\mathbb{R})$ est défini par

$$\mathcal{R}_\theta[f](t) = \mathcal{R}[f](\theta, t) = \int_{\mathbb{R}} f(t\mathbf{e}_\theta^\perp + s\mathbf{e}_\theta) \, ds, \quad \forall t \in \mathbb{R}.$$

En particulier, la transformée de Fourier de $\mathcal{R}_\theta[f]$ vérifie, pour tout $\sigma \in \mathbb{R}$:

$$\begin{aligned} \mathcal{F}[\mathcal{R}_\theta[f]](\sigma) &= \int_{\mathbb{R}} \mathcal{R}[f](\theta, t) e^{-2i\pi\sigma t} \, dt = \int_{\mathbb{R}} \int_{\mathbb{R}} f(t\mathbf{e}_\theta^\perp + s\mathbf{e}_\theta) e^{-2i\pi\sigma t} \, ds \, dt, \\ &= \int_{\mathbb{R}^2} f(x) e^{-2i\pi\sigma e_\theta \cdot x} \, dx = \mathcal{F}[f](\sigma e_\theta). \end{aligned}$$

Il est également possible d'identifier simplement la transformée de Radon adjointe. En effet, pour tout

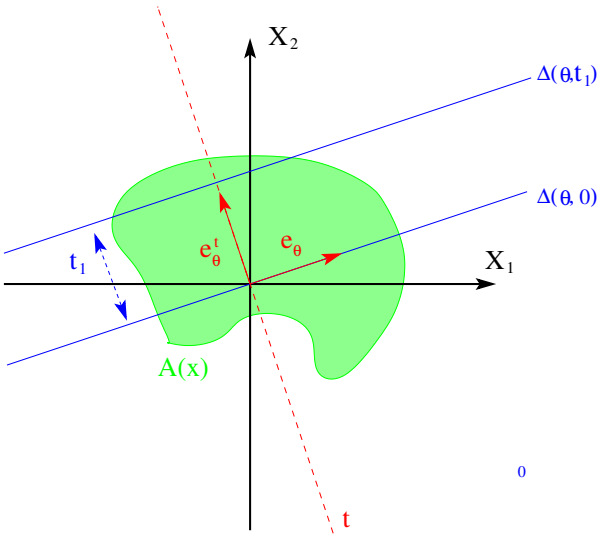

FIGURE 2.1 – Imagerie par rayons X et transformée de Radon

$f \in \mathcal{S}(\mathbb{R}^2)$ et $g \in \mathcal{S}([0,\pi] \times \mathbb{R})$,

$$\begin{aligned}
\langle \mathcal{R}[f], g \rangle_{L^2([0,\pi]\times\mathbb{R})} &= \int_{\mathbb{R}} \int_0^{\pi} \mathcal{R}[f](\theta,t) g(\theta,t)\, d\theta\, dt \\
&= \int_0^{\pi} \int_{\mathbb{R}} \int_{\mathbb{R}} f(t\mathbf{e}_\theta^{\perp} + s\mathbf{e}_\theta) g(\theta,t)\, ds\, dt\, d\theta \\
&= \int_{\mathbb{R}^2} f(x) \mathcal{R}^*[g](x)\, dx = \langle f, \mathcal{R}^*[g] \rangle_{L^2(\mathbb{R}^2)},
\end{aligned}$$

où

$$\mathcal{R}^*[g](x) = \int_0^{\pi} g(\theta, x \cdot e_\theta)\, d\theta.$$

Au final, en remarquant que

$$\begin{aligned}
f(x) &= \int_{\mathbb{R}^2} \mathcal{F}[f](\xi) e^{2i\pi\xi\cdot x}\, d\xi = \int_{\mathbb{R}} \int_0^{\pi} \mathcal{F}[f](\sigma e_\theta) e^{2i\pi(e_\theta\sigma)\cdot x} |\sigma|\, d\theta\, d\sigma, \\
&= \int_0^{\pi} \int_{\mathbb{R}} |\sigma| \mathcal{F}[\mathcal{R}_\theta[f]](\sigma) e^{2i\pi\sigma e_\theta \cdot x}\, d\sigma\, d\theta, \\
&= \int_0^{\pi} \mathcal{F}^{-1}\left[|\sigma| \mathcal{F}[\mathcal{R}_\theta[f]]\right](e_\theta \cdot x)\, d\theta = \mathcal{R}^*\left[\mathcal{F}^{-1}\left[|\sigma| \mathcal{F}[\mathcal{R}_\theta[f]]\right]\right](x),
\end{aligned}$$

on en déduit une formule d'inversion de la forme

$$f = \mathcal{R}^*\left[\mathcal{F}^{-1}\left[|\sigma| \mathcal{F}[\mathcal{R}_\theta[f]]\right]\right].$$

Cette formule montre donc l'unicité du problème inverse lié à la transformée de Radon. Toutefois, la multiplication par le symbole $|\sigma|$ met aussi en évidence le caractère mal posé du problème, puisque le bruit est fortement amplifié aux hautes fréquences.

La figure 1 illustre un exemple de source f et de sa transformée de Radon échantillonnée avec 2^8 points en θ et 2^8 points en translation t. La figure 1 présente des exemples de reconstructions qui montrent l'instabilité de la formule d'inversion en présence de bruit et dans le cas de données tronquées. Nous verrons par la suite que des algorithmes itératifs basés sur des approches variationnelles avec terme de régularisation permettent d'améliorer significativement la reconstruction.

```
function I_radon = radon_tranform(Im,N_t,N_theta)
I_radon = zeros(N_theta,N_t);
[N1,N2] = size(Im);
Theta = linspace(0,pi,N_theta +1);
T = linspace(-1,1,N_t);
Nd = max(N1,N2);
delta_t = linspace(-1,1,Nd);
for i_theta = 1:N_theta,
   for j_t = 1:N_t,
      ppx_theta = cos(Theta(i_theta))*delta_t - sin(Theta(i_theta))*T(j_t);
      ppy_theta = sin(Theta(i_theta))*delta_t + cos(Theta(i_theta))*T(j_t);
      sum=0;
      for k=1:Nd,
        if (ppx_theta(k)^2 + ppy_theta(k)^2<0.98),
     indice_x = (ppx_theta(k)+1)*(N1-1)/2+1;
     indice_y = (ppy_theta(k)+1)*(N2-1)/2+1;
     lambda_x = indice_x -floor(indice_x);
     lambda_y = indice_y -floor(indice_y);
     sum=sum + (1-lambda_x)*(1-lambda_y)*Im(floor(indice_x),floor(indice_y)) ...
        +(lambda_x)*(1-lambda_y)*Im(min(floor(indice_x)+1,N1),floor(indice_y)) ...
        +(1-lambda_x)*(lambda_y)*Im(floor(indice_x),min(floor(indice_y)+1,N2)) ...
        + (lambda_x)*(lambda_y)*Im(min(floor(indice_x)+1,N1),min(floor(indice_y)+1,N2)) ;
        end
      end
       I_radon(i_theta,j_t) = sum*2/Nd;
   end
end
end
```

FIGURE 2.2 – Exemple de code **Matlab** pour calculer la transformée de Radon d'une image

```
function Im = radon_tranform_adjoint(I_radon,N1,N2)
Im =zeros(N1,N2);
[X2,X1] = meshgrid(linspace(-1,1,N1),linspace(-1,1,N2));
[N_theta,N_t] = size(I_radon);
Theta = linspace(0,pi,N_theta +1);
T = linspace(-1,1,N_t);
for i_theta = 1:N_theta,
    t_theta = X2*cos(Theta(i_theta)) - X1*sin(Theta(i_theta));
    indice_t = (t_theta+1)/2*(N_t-1)+1;
    lambda_t = indice_t - floor(indice_t);
    for i=1:N1,
       for j=1:N2,
          if (sqrt(X1(i,j).^2 + X2(i,j).^2)<0.98)
          if t_theta(i,j)>-0.98 && t_theta(i,j)<0.98,
         Im(i,j) = Im(i,j) + ((1-lambda_t(i,j))*I_radon(i_theta,floor(indice_t(i,j))) ...
                        + lambda_t(i,j)*I_radon(i_theta,min(floor(indice_t(i,j))+1,N_t)))*pi/N_theta;
          end
          end
       end
    end
end
```

FIGURE 2.3 – Exemple de code **Matlab** pour calculer la transformée de Radon adjointe

1.1 Problème d'inpainting

Le problème d'inpainting est un problème inverse classique en traitement d'image. L'idée consiste à restaurer une image $f \in L^2(\Omega, \mathbb{R})$ qui est altérée dans une zone $K \subset \Omega$, comme illustré à la figure 2.7.

Ce problème inverse est linéaire mais naturellement mal posé, car la solution n'est pas unique. Il est donc nécessaire d'introduire un *a priori* sur l'image que l'on souhaite reconstruire, en particulier en introduisant une norme $\|\cdot\|_{reg,K}$ et en cherchant à résoudre le problème de minimisation

$$I^* = \arg\min_I \{\|I\|^2_{reg,K}\}$$

sous la contrainte $I = f$ à l'extérieur de K.

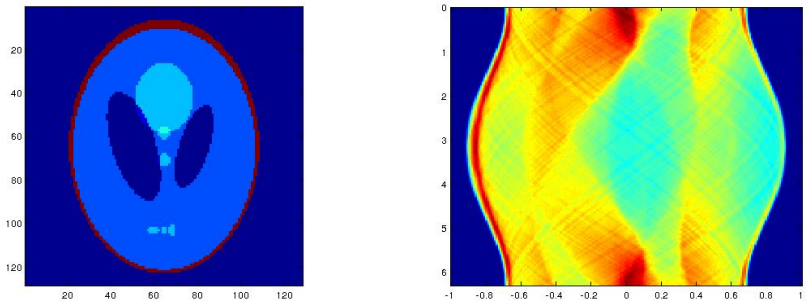

FIGURE 2.4 – Exemple numérique de source f et d'une approximation numérique de sa transformée de Radon

```
clear;colormap('jet');
N = 2^7;
x1 = linspace(-1,1,N);
x2 = linspace(-1,1,N);
[X1,X2] = meshgrid(x1,x2);
Domaine = (X1.^2 + X2.^2 < 0.99^2);
A = phantom(N);imagesc(A)
N_t = 2^9;N_theta = 2^9;
I_data = radon_tranform(A,N_t,N_theta);
I_data = I_data + 0*0.05*randn(size(I_data));
sigma = [0:N_t/2,-N_t/2+1:-1];
for i=1:N_theta,
    I_data_filter(i,1:N_t) = real(ifft( abs(sigma).*fft(I_data(i,:))));
end
imagesc(I_data_filter)
A_rec = radon_tranform_adjoint(I_data_filter,N,N);
imagesc(x1,x2,A_rec)
axis square;
```

FIGURE 2.5 – Exemple de code **Matlab** qui permet d'appliquer la transformée de Radon inverse

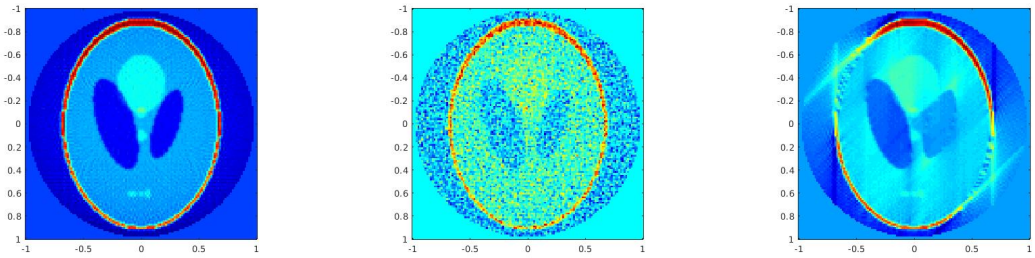

FIGURE 2.6 – Exemple de reconstructions obtenues en utilisant la transformée de Radon inverse. A gauche sans bruit ; Au milieu avec des données bruitées ; A droite avec des données incomplètes où les orientations se limitent à l'intervalle $\theta \in [0, \pi/2]$

On peut par exemple considérer :

- une régularisation harmonique :

$$\|I\|^2_{reg,\Omega} = \int_\Omega |\nabla I|^2 \, dx,$$

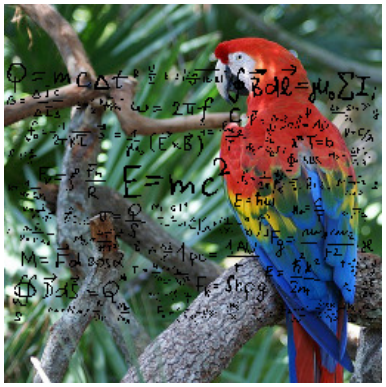

FIGURE 2.7 – Exemple d'image altérée dans une zone K

- une régularisation TV :

$$\|I\|^2_{reg,\Omega} = \int_\Omega |DI|,$$

- une régularisation de type Mumford–Shah :

$$\|I\|^2_{reg,\Omega} = \lambda \int_{\Omega\setminus A} |\nabla I|^2 \, dx + \mathcal{H}^1(A).$$

En pratique, il est souvent plus simple de considérer le problème de minimisation approchée

$$I^* = \arg\min_I \left\{ \|I\|^2_{reg,\Omega} + \frac{1}{2\varepsilon}\|I - f\|^2_{L^2(\Omega\setminus K)} \right\},$$

où le paramètre ε est supposé suffisamment petit pour forcer la solution I^* à coïncider avec f en dehors de K.

Cet exemple permet notamment d'illustrer l'influence du choix du terme de régularisation sur la reconstruction de l'image dans la zone altérée K. La figure 2.8 montre par exemple des reconstructions obtenues avec l'approche harmonique, qui se révèlent assez efficaces pour des masques K relativement fins.

1.2 Imagerie par élastographie

L'imagerie par élastographie consiste à reconstruire les propriétés élastiques d'un milieu biologique à l'aide d'échographies de ce milieu, que l'on déforme au cours du temps (voir la figure 2.9).

Dans le cas simple de l'élasticité linéaire et isotrope, le champ de déformation $\mathbf{u}$ d'un milieu élastique Ω soumis à la pression d'une force $\mathbf{f}$ est supposé solution de l'équation élastique suivante :

$$\begin{cases} \operatorname{div}\big(2\mu\nabla^s\mathbf{u}\big) + \nabla\big(\lambda \operatorname{div}(\mathbf{u})\big) = 0 & \text{dans } \Omega, \\ \mathbf{u} = \mathbf{f} & \text{sur } \partial\Omega. \end{cases}$$

où (λ, μ) sont les coefficients de Lamé du milieu que l'on cherche à reconstruire.

Les données du problème inverse sont alors des images échographiques

$$I_{\mathbf{u}}(x) = I(x + \mathbf{u}(x))$$

du milieu, comme illustré sur la figure 2.10.

La reconstruction des propriétés élastiques s'effectue alors en deux étapes :

FIGURE 2.8 – Exemples de reconstructions en utilisant une régularisation harmonique. À gauche : masque K ; colonnes 2 et 3 : images reconstruites.

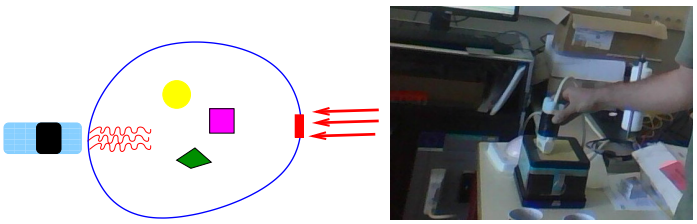

FIGURE 2.9 – Principe de l'imagerie par élastographie

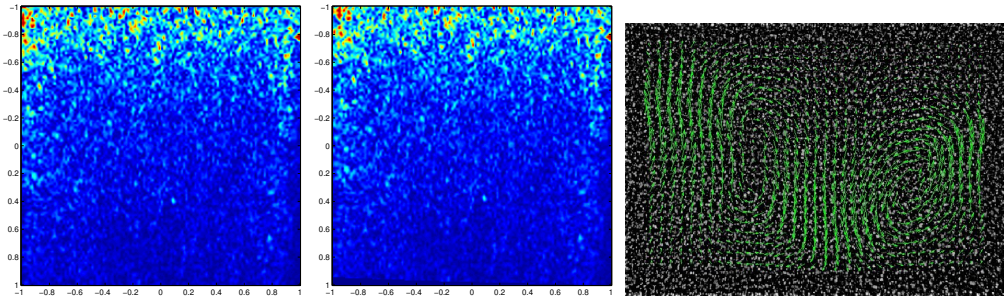

FIGURE 2.10 – Images échographiques d'un milieu en cours de déformation

1) reconstruire le champ de déformation $\mathbf{u}$ à partir d'un jeu de deux images : I et $I_{\mathbf{u}}$;

2) reconstruire les coefficients de Lamé (λ, μ) à partir de $\mathbf{u}$.

Le problème inverse lié à la reconstruction du champ de déformation $\mathbf{u}$ à partir des images I et $I_{\mathbf{u}}$ est généralement mal posé en raison d'un manque d'unicité. En pratique, on peut chercher à minimiser la fonctionnelle

$$J(\mathbf{u}) = \int (I(x + \mathbf{u}(x)) - I_{\mathbf{u}}(x))^2 \, dx.$$

Il existe néanmoins plusieurs algorithmes efficaces pour les petites déformations, notamment les méthodes de flot optique telles que celles de Lucas–Kanade ou de Horn–Schunck. L'idée est alors d'utiliser l'approximation

$$I(x + \mathbf{u}(x)) \simeq I(x) + \nabla I(x) \cdot \mathbf{u}(x),$$

couplée à des termes de régularisation H^1 sur $\mathbf{u}$. La figure 2.11 présente un exemple de reconstruction de champ de déformation à partir de données expérimentales.

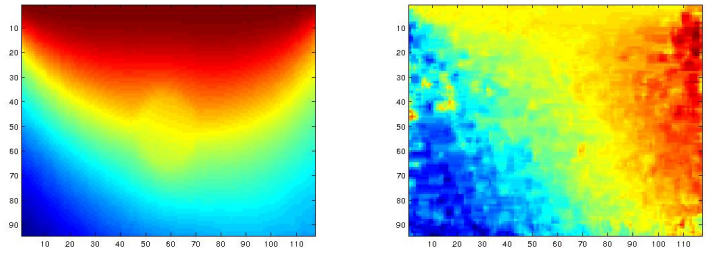

FIGURE 2.11 – Exemple de champs de déformation reconstruits à partir de données expérimentales

Concernant la reconstruction des coefficients de Lamé à partir de plusieurs champs de déformation $\mathbf{u}$, on rappelle que le problème direct consiste à déterminer la solution $\mathbf{u}[\lambda, \mu]$ de l'EDP

$$\begin{cases} \operatorname{div}\big(2\mu \nabla^s \mathbf{u}\big) + \nabla\big(\lambda \operatorname{div}(\mathbf{u})\big) = 0 & \text{dans } \Omega, \\ \mathbf{u} = \mathbf{f} & \text{sur } \partial\Omega. \end{cases}$$

Une difficulté vient ici de la dépendance non linéaire de $\mathbf{u}[\lambda, \mu]$ par rapport aux coefficients de Lamé (λ, μ). On peut cependant chercher à minimiser une énergie de la forme

$$J(\lambda, \mu) = \int_{\Omega} \big(\mathbf{u}[\lambda, \mu](x) - \mathbf{u}_{\text{data}}(x)\big)^2 \, dx.$$

Nous verrons ultérieurement comment la méthode de l'état adjoint permet d'obtenir facilement le gradient de J à l'aide de la résolution d'une deuxième EDP. La figure 2.12 présente un exemple de reconstruction du coefficient μ à partir de données expérimentales.

1.3 Imagerie photoacoustique

L'imagerie photoacoustique est un deuxième exemple d'imagerie hybride, où l'on cherche à reconstruire les propriétés d'absorption d'un milieu biologique Ω. Comme illustré à la figure 2.13, l'idée consiste à chauffer un milieu à l'aide d'un laser, provoquant une dilatation de la matière qui émet ainsi une onde acoustique se propageant dans le milieu. Des capteurs positionnés autour de la zone à imager permettent alors de mesurer l'onde acoustique évoluant au cours du temps.

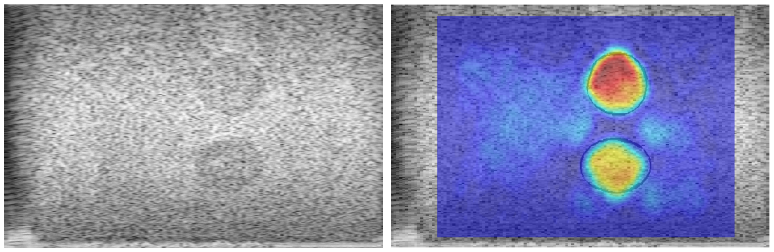

FIGURE 2.12 – Exemple de reconstruction du coefficient de Lamé μ

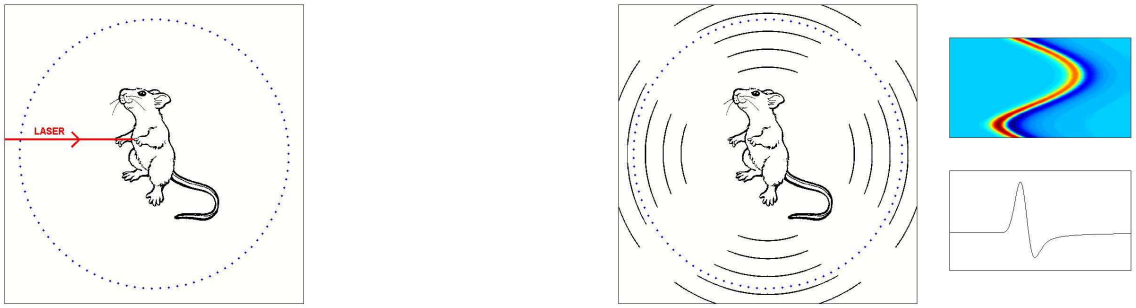

FIGURE 2.13 – Principe de l'imagerie photoacoustique

On suppose ici que la propagation de la lumière vérifie l'équation

$$\begin{cases} -\mathrm{div}(D\nabla u) + \sigma u = 0 & \text{dans } \Omega, \\ u = g & \text{sur } \partial\Omega, \end{cases}$$

où (D, σ) représentent respectivement les coefficients de diffusion et d'absorption.

La dilatation des tissus génère une onde acoustique p dont l'équation vérifie

$$\frac{1}{c_s^2}\partial_{tt}^2 p(x,t) - \Delta p(x,t) = \beta \partial_t \big(S(t)\big) H(x),$$

où c_s est la vitesse de propagation de l'onde, β un coefficient de couplage, $H = \sigma u$ la source proportionnelle à u, et S le profil temporel du pulse laser.

Les données mesurées sur l'intervalle de temps $[0, T]$ par les capteurs autour de Ω correspondent mathématiquement à

$$d(y,t) = p(y,t), \quad \forall (y,t) \in \partial\Omega \times [0,T].$$

Le problème inverse pour reconstruire les coefficients (D, σ) se déroule en deux étapes :

1) reconstruction de la source H à partir de d ;

2) reconstruction des coefficients (D, σ) à partir de H.

Pour la première étape, comme la durée du pulse laser est très courte par rapport au temps de propagation de l'onde acoustique, on peut approximer S par une masse de Dirac, ce qui conduit à l'EDP équivalente suivante :

$$\begin{cases} \frac{1}{c_s^2}\partial_{tt}^2 p(x,t) - \Delta p(x,t) = 0, \\ p(x,0) = \beta H(x), \quad \partial_t p(x,0) = 0. \end{cases}$$

Il s'agit donc de reconstruire la condition initiale $H = \sigma u(x)$ à partir des mesures de pression d. Ce problème inverse linéaire peut être résolu à l'aide de techniques de renversement temporel, basées sur l'irréversibilité de l'équation des ondes par inversion du temps, ou par l'utilisation de transformées de Radon sphériques.

La deuxième étape, dite partie quantitative, consiste à reconstruire (D, σ) à partir d'une collection d'illuminations (g_i, H_i) :

$$\begin{cases} -\mathrm{div}(D\nabla u_i) + \sigma u_i = 0 & \text{dans } \Omega, \\ u_i = g_i & \text{sur } \partial\Omega. \end{cases}$$

Ce problème inverse est bien posé, comme montré par Bal et Uhlmann (2009), avec des résultats de stabilité nécessitant $2d$ illuminations :

$$\|D - \tilde{D}\|_{C^k} + \|\sigma - \tilde{\sigma}\|_{C^k} \leqslant C \sum_{i=1}^{2d} \|H_i - \tilde{H}_i\|_{C^{k+1}}.$$

En pratique, on peut chercher à minimiser une fonctionnelle de la forme

$$J(D, \sigma) = \sum_i \int_\Omega (H_i(D, \sigma) - H_{i,\mathrm{mes}})^2 \, dx,$$

où l'utilisation de la méthode de l'état adjoint permet d'expliciter facilement les gradients de J.

2 Formulation variationnelle des problèmes inverses

Dans cette partie, nous nous intéressons à une formulation variationnelle des problèmes inverses dans un cadre général. L'idée est alors d'introduire :

- **L'espace des paramètres** M qui regroupe l'ensemble des inconnues du problème inverse.
- **L'espace d'état** U correspondant à l'espace des solutions du problème direct.
- **L'équation d'état** F qui relie implicitement les paramètres à la solution du problème direct :

  $$F(a, u) = 0.$$

  Pour un paramètre $a \in M$ donné, cette équation permet de déterminer l'état du système $u \in U$, noté par la suite $u[a] \in U$.
- **L'espace des observations** D qui regroupe les données du problème inverse.
- **L'équation d'observation** qui extrait de l'état la partie correspondant aux mesures :

  $$d_{\mathrm{obs}} = H(u[a]) = \phi(a).$$

La formulation variationnelle du problème inverse consiste alors à rechercher le paramètre a^* en minimisant l'énergie J d'écart aux données :

$$J(a) = \frac{1}{2} \|\phi(a) - d_{\mathrm{obs}}\|_D^2.$$

L'intérêt de cette approche est de pouvoir utiliser un schéma itératif pour minimiser J, et d'ajouter facilement à J des termes de régularisation sur a.

Cette approche présente cependant plusieurs difficultés :

- La non-linéarité de ϕ conduit à des fonctionnelles J non convexes, dont la minimisation peut converger vers des minimums locaux.

- Le problème peut être sous-déterminé, avec plusieurs solutions possibles pour un même jeu d'observations, entraînant une forte dépendance à l'initialisation des algorithmes.
- Le manque de continuité de J peut engendrer des instabilités dans la reconstruction des paramètres optimaux a^*, surtout lorsque les données d_{obs} sont bruitées.
- Même dans le cas linéaire, les gradients de J sont rarement explicites et nécessitent l'utilisation de la méthode dite de l'état adjoint.

3 Algorithmes du gradient et variantes

La minimisation d'une fonctionnelle J peut s'effectuer à l'aide d'un algorithme du gradient, consistant à construire la séquence $\{x_k\}_{k\in\mathbb{N}}$ par

$$x_{k+1} = x_k + \alpha d_k,$$

où la direction d_k s'identifie à $d_k = -\nabla J(x_k)$, α est le pas de descente, et l'initialisation x_0 vérifie $x_0 \in D$.

3.1 Stabilité

Si J est à gradient Lipschitz de module L,

$$\|\nabla J(x) - \nabla J(y)\| \leqslant L\|x - y\|,$$

alors, pour un pas de descente α vérifiant $\alpha \leqslant \frac{1}{L}$, il est possible de montrer que la suite $\{J(x_k)\}$ décroît au cours des itérations :

$$J(x_{k+1}) \leqslant J(x_k) - \frac{\alpha}{2}\|\nabla J(x_k)\|^2.$$

3.2 Choix du pas de descente α

Le choix du pas de descente α est crucial pour l'efficacité de l'optimisation de J. Un pas trop grand peut provoquer des instabilités numériques, tandis qu'un pas trop petit ralentit la convergence.

Un bon pas α peut être choisi en respectant les conditions de Wolfe, avec $0 < \omega_1 < \omega_2 < 1$:

- Condition d'Armijo :
$$J(x_k + \alpha d_k) \leqslant J(x_k) + \omega_1 \alpha \nabla J(x_k) \cdot d_k,$$
- Condition de courbure :
$$\nabla J(x_k + \alpha d_k) \cdot d_k \geqslant \omega_2 \nabla J(x_k) \cdot d_k.$$

On pourra appliquer l'algorithme de Fletcher-Lemaréchal pour déterminer un tel pas α, à l'aide d'une approche de type dichotomie.

3.3 Variantes

Il existe de nombreuses variantes de l'algorithme du gradient, qui introduisent une notion d'inertie et permettent d'accélérer la convergence de la séquence $\{x_k\}_k$. Parmi celles-ci, on peut citer :

- **Méthode des moments** :
$$\begin{cases} x_{k+1} &= x_k - \alpha d_k, \\ d_k &= \nabla J(x_k) + \beta d_{k-1}. \end{cases}$$

- **Algorithme du gradient conjugué** :

$$d_k = \begin{cases} -\nabla J(x_k) & \text{si } k = 1, \\ -\nabla J(x_k) + \beta_k d_{k-1} & \text{si } k \geqslant 2, \end{cases}$$

avec β_k donné par l'approche Fletcher-Reeves :

$$\beta_k = \frac{\|\nabla J(x_k)\|^2}{\|\nabla J(x_{k-1})\|^2}.$$

- **Accélération de Nesterov** :

$$\begin{cases} x_{k+1} &= x_k - \alpha \nabla J(y_k), \\ y_{k+1} &= x_{k+1} + \beta(x_{k+1} - x_k). \end{cases}$$

La figure 2.14 présente une comparaison de ces algorithmes pour la fonction de Rosenbrock J définie par :

$$J(x) = (x_1 - 1)^2 + p(x_1^2 - x_2)^2, \quad x \in \mathbb{R}^2.$$

Avec le même critère d'arrêt, l'algorithme du gradient nécessite environ 350 itérations, tandis que la version Polak-Ribière du gradient conjugué n'en utilise que 25.

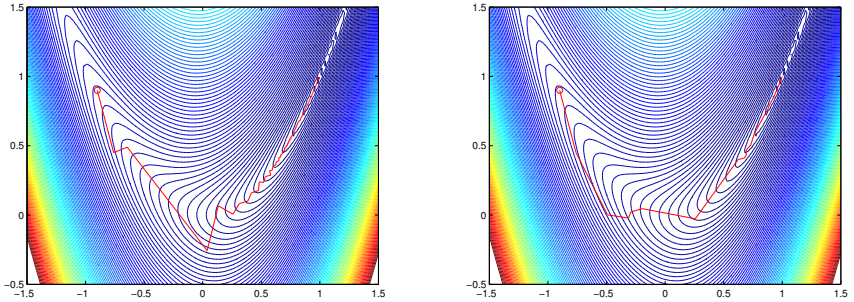

FIGURE 2.14 – Minimisation de la fonction de Rosenbrock : comparaison de la méthode du gradient et du gradient conjugué.

4 Estimation du gradient : cas linéaire

Une manière efficace de déterminer le gradient d'une fonctionnelle J consiste à l'identifier par l'intermédiaire de la dérivée directionnelle :

$$J'(u)(\delta_u) \quad = \quad \lim_{h \to 0} \frac{J(u + h\delta_u) - J(u)}{h} = \langle \nabla J(u), \delta_u \rangle.$$

Le cas linéaire conduit à une fonctionnelle quadratique de la forme

$$J(u) = \frac{1}{2}\|Lu - b\|^2,$$

où le gradient s'exprime comme

$$\nabla J(u) = L^*(Lu - b),$$

et nécessite donc d'expliciter l'adjoint de L.

4.1 Cas de la méthode d'inpainting

Dans le cas de la méthode d'inpainting présentée en introduction, le terme d'attache aux données est de la forme

$$J(u) = \frac{1}{2}\int_\Omega \mathbf{1}_{\Omega\setminus K}(u-I)^2 dx,$$

et l'opérateur A s'identifie à $Au = \mathbf{1}_{\Omega\setminus K}u$. Son adjoint vérifie également $A^*u = \mathbf{1}_{\Omega\setminus K}u$, ce qui conduit à l'expression suivante du gradient :

$$\nabla J(u) = (u-I)\mathbf{1}_{\Omega\setminus K}.$$

4.2 Régularisation H^1

Dans le cas d'une régularisation H^1 sur un domaine Ω avec des conditions de Dirichlet homogènes, la fonctionnelle à minimiser s'écrit

$$J(u) = \frac{1}{2}\int_\Omega |\nabla u|^2 dx.$$

L'opérateur associé s'identifie à $Au = \nabla u$, son adjoint à $A^*u = -\operatorname{div}(u)$, ce qui conduit à l'expression du gradient

$$\nabla J(u) = -\Delta u.$$

4.3 Transformée de Radon tronquée

Considérons maintenant la transformée de Radon tronquée pour $a \in L^2(B_1)$, définie par

$$\mathcal{R}[a](\theta,t) = \int_{\mathbb{R}} a(t\mathbf{e}_\theta + s\mathbf{e}_\theta^\perp)ds,$$

et observée seulement pour $(\theta,t) \in [0,\pi/2]\times[-1,1]$. Le problème inverse consiste à reconstruire a à partir de ces données tronquées.

Les espaces fondamentaux s'identifient à

$$M = L^2(B_1), \quad U = L^2([0,\pi]\times[-1,1]), \quad D = L^2([0,\pi/2]\times[-1,1]),$$

avec l'équation d'état

$$F(a,u) = \mathcal{R}[a] - u$$

et l'équation d'observation

$$d_{\text{obs}} = \phi(a) = u[a]_{|[0,\pi/2]\times[-1,1]}.$$

La fonctionnelle d'écart aux données s'écrit alors

$$J(a) = \frac{1}{2}\|\phi(a) - d_{\text{obs}}\|_D^2 = \frac{1}{2}\|\mathcal{R}[a](\theta,t) - d_{\text{obs}}\|^2_{L^2([0,\pi/2]\times[-1,1])}.$$

L'opérateur adjoint $\mathcal{R}^* : L^2([0,\pi/2]\times[-1,1]) \to L^2(B_1)$ est donné par

$$\mathcal{R}^*[g](x) = \int_0^{\pi/2} g(x\cdot\mathbf{e}_\theta,\theta)d\theta, \quad \forall g \in L^2([0,\pi/2]\times[-1,1]).$$

Ainsi, le gradient de J s'exprime comme

$$\nabla J(a) = \mathcal{R}^*\big[\mathcal{R}[a](\theta,t) - d_{\text{obs}}\big](x).$$

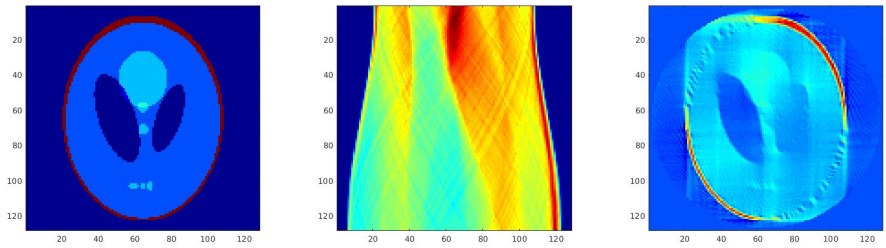

FIGURE 2.15 – Transformée de Radon : exemple de reconstruction avec l'algorithme du gradient

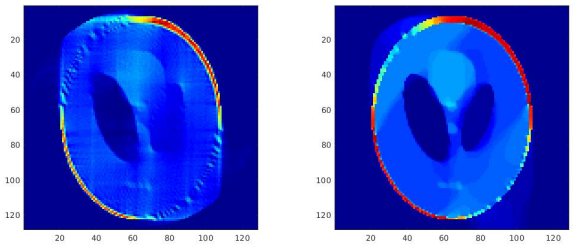

FIGURE 2.16 – Transformée de Radon : exemples de reconstruction en ajoutant un a priori de positivité (gauche) ou un terme de régularisation TV (droite)

5 Estimation du gradient : cas non linéaire et méthode de l'état adjoint

Dans cette partie, on s'intéresse aux problèmes inverses non linéaires pour lesquels le calcul des gradients des fonctionnelles variationnelles n'est plus explicite et nécessite l'utilisation de la méthode dite de l'état adjoint.

5.1 Cas d'une équation elliptique

On considère $u[a]$ solution du problème elliptique

$$\begin{cases} -(a(x)u'(x))' = f(x), & x \in (-1,1), \\ u(-1) = u(1) = 0. \end{cases}$$

Le problème inverse consiste à reconstruire le coefficient a à partir d'une mesure de u en $x = 0$. On pose les espaces fondamentaux [U+202F] : $M = C^0([-1,1],\mathbb{R})$, $U = H_0^1([-1,1],\mathbb{R})$ et $D = \mathbb{R}$. L'équation d'état est

$$F(a,u) = f(x) + (a(x)u'(x))',$$

et l'équation d'observation correspond à

$$\phi(a) = u,$$

avec la fonctionnelle à minimiser

$$J(a) = \frac{1}{2}(u - d_{\text{obs}})^2.$$

Cas d'une observation totale

Si l'on observe $u[a]$ sur l'ensemble du domaine, la fonctionnelle s'écrit

$$J(a) = \frac{1}{2}\|u[a] - d_{\text{obs}}\|^2_{L^2([-1,1])} = \frac{1}{2}\int_{-1}^{1} (u[a](x) - d_{\text{obs}}(x))^2 dx.$$

La dérivée directionnelle dans la direction δ_a est

$$J'(a)(\delta_a) = \int_{-1}^{1} (u[a] - d_{\text{obs}})\, \dot{u}[a]\, dx,$$

où $\dot{u}[a]$ est solution de l'équation linéarisée

$$\begin{cases} -(a(x)\dot{u}[a]'(x))' = (\delta_a(x)u[a]'(x))', \quad x \in (-1,1), \\ \dot{u} = 0. \end{cases}$$

Pour expliciter le gradient sans calculer $\dot{u}[a]$ à chaque itération, on introduit l'état adjoint p solution de

$$\begin{cases} -(a(x)p'(x))' = u[a](x) - d_{\text{obs}}(x), \quad x \in (-1,1), \\ p(-1) = p(1) = 0. \end{cases}$$

On obtient alors

$$J'(a)(\delta_a) = -\int_{-1}^{1} p'(x)u[a]'(x)\, \delta_a(x)\, dx,$$

ce qui fournit l'expression du gradient

$$\nabla J(a)(x) = -p'(x)\, u[a]'(x).$$

La figure 2.17 présente un exemple de script **Matlab** permettant de reconstruire le coefficient de conductivité a en utilisant un algorithme du gradient. À chaque itération, le calcul du gradient de J nécessite la résolution du problème direct pour $u[a]$ ainsi que celui de l'état adjoint p.

La figure 2.18 illustre un exemple de reconstruction de a, mettant en évidence la difficulté de reconstituer le paramètre dans les zones où $u[a]'$ s'annule. Une solution consiste à utiliser deux jeux de données distincts pour contourner ce problème.

Observation en un point

Si l'on ne dispose que de l'observation u, on pose

$$J(a) = \frac{1}{2}(u - d_{\text{obs}})^2.$$

On introduit l'état adjoint p solution de

$$\begin{cases} -(a(x)p'(x))' = (u - d_{\text{obs}})\delta_0, \quad x \in (-1,1), \\ p(-1) = p(1) = 0, \end{cases}$$

avec δ_0 la distribution de Dirac en $x = 0$, ce qui conduit à la même expression

$$\nabla J(a)(x) = -p'(x)\, u[a]'(x).$$

```
N = 2^7;
x = linspace(-1,1,N+1)';
h = 2/N;
a = 1+ 0.1*sin(pi*x);
a_exact = a;
f = 1 + 0*sin(4*pi*x(2:N));
M = -(-diag(a(2:N) + a(1:N-1)) + diag(a(2:N-1),+1) +  diag(a(2:N-1),-1))/h^2;
%%%%%%%%%%%%%%%%%%%%%% problème direct %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
u_a = M\f;
plot(x , [0;u_a;0]);
d_obs = u_a;
clf;
plot(x,a_exact,'g');
hold on;
plot(x,[0;u_a;0]);
plot(x(2:N),f,'k')
legend('a','u[a]','f');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
a = ones(size(x));
a_init = a;
%%%%%%%%%%%%%%%%%%%% reconstruction
for i=1:100000,
M = -(-diag(a(2:N) + a(1:N-1)) + diag(a(2:N-1),+1) +  diag(a(2:N-1),-1))/h^2;
u_a = M\f;
p = M'\(u_a - d_obs);
nablaJ_a =   -(diff([0;u_a])/h).*(diff([p;0])/h);
a = a - 1*[nablaJ_a(1);nablaJ_a ;nablaJ_a(N-1)];
if(mod(i,10)==1)
clf
plot(x,a);
hold on;
plot(x,a_exact,'g')
plot(x,a_init,'r')
legend('a^k','a_{exact}','a_{init}');
pause(0.01)
end
end
```

FIGURE 2.17 – Exemple de script **Matlab** pour reconstruire le coefficient a à l'aide d'une méthode de gradient.

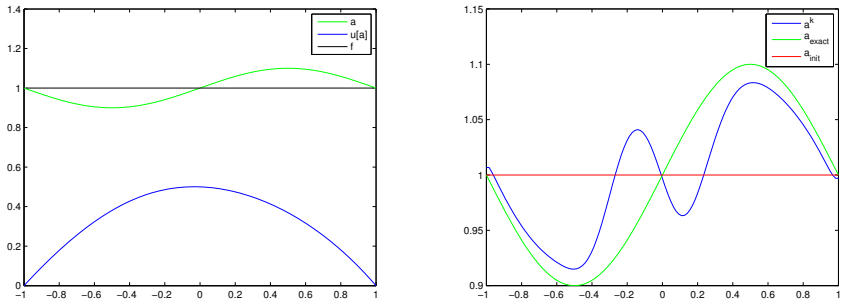

FIGURE 2.18 – Reconstruction de la conductivité a : à gauche, a, f et $u[a]$; à droite, a initial, a exact et a reconstruit après un certain nombre d'itérations.

5.2 Cas de la reconstruction d'un paramètre d'une EDO

On considère la solution $u[a]$ d'une EDO d'ordre 1 de la forme

$$\begin{cases} u'(t) = f(u(t), a), & t \in [0, T], \\ u(0) = u_0, \end{cases}$$

où $f \in C^1(\mathbb{R} \times \mathbb{R}^p, \mathbb{R})$. Le problème inverse consiste à reconstruire le paramètre $a \in \mathbb{R}^p$ à partir d'observations de $u[a]$ sur l'intervalle $[0, T]$.

Les espaces fondamentaux sont

$$M = \mathbb{R}^p, \quad U = H^1([0, T], \mathbb{R}), \quad D = H^1([0, T], \mathbb{R}),$$

l'équation d'état correspondant au problème direct :

$$F_1(a, u) = u'(t) - f(u(t), a), \quad F_2(a, u) = u(0) - u_0,$$

et l'équation d'observation est simplement l'identité :

$$\phi(a) = u[a].$$

La formulation variationnelle du problème inverse consiste à minimiser la fonctionnelle

$$J(a) = \frac{1}{2}\int_0^T (u[a](t) - d_{\text{obs}}(t))^2\, dt.$$

Sa dérivée directionnelle dans la direction δ_a s'écrit

$$J'(a)(\delta_a) = \int_0^T (u[a](t) - d_{\text{obs}}(t))\, \dot{u}[a](t)\, dt,$$

où $\dot{u}[a]$ satisfait l'équation linéaire

$$\dot{u}[a]'(t) - \partial_u f(u[a](t), a)\, \dot{u}[a](t) = \nabla_a f(u[a](t), a) \cdot \delta_a, \quad \dot{u} = 0.$$

On introduit alors l'état adjoint p solution de

$$-p'(t) - \partial_u f(u[a](t), a)\, p(t) = u[a](t) - d_{\text{obs}}(t), \quad p(T) = 0,$$

qui s'obtient en pratique par intégration rétrograde (retournement temporel).

En utilisant l'état adjoint, on obtient

$$\begin{aligned} J'(a)(\delta_a) &= \int_0^T (u[a](t) - d_{\text{obs}}(t))\, \dot{u}[a](t)\, dt \\ &= \int_0^T \big(- p'(t) - \partial_u f(u[a](t), a)\, p(t)\big)\, \dot{u}[a](t)\, dt \\ &= \int_0^T \big(\dot{u}[a]'(t) - \partial_u f(u[a](t), a)\, \dot{u}[a](t)\big)\, p(t)\, dt \\ &= \int_0^T \nabla_a f(u[a](t), a) \cdot \delta_a\, p(t)\, dt, \end{aligned}$$

ce qui conduit à l'expression du gradient de J :

$$\nabla J(a) = p(t)\, \nabla_a f(u[a](t), a).$$

5.3 Formulation abstraite de la méthode de l'état adjoint

On considère maintenant le cas général où l'équation d'état relie de façon implicite le paramètre a à reconstruire à la solution $u[a]$ du problème direct :

$$F(a, u) = 0 \quad \text{dans } H, \quad a \in M,\ u \in U.$$

L'équation d'observation permet d'extraire de l'état la partie correspondant aux mesures :

$$d_{\text{obs}} = H(u[a]) = \phi(a).$$

La reconstruction de a s'effectue par la minimisation de la fonctionnelle

$$J(a) = \frac{1}{2}\|H(u[a]) - d_{\text{obs}}\|_D^2,$$

qui nécessite de déterminer une expression de son gradient.

Pour cela, on introduit le lagrangien

$$\mathcal{L}(a,u,q) = \frac{1}{2}\|H(u) - d_{\text{obs}}\|^2 + \langle q, F(a,u)\rangle,$$

défini de sorte que

$$J(a) = \mathcal{L}(a, u[a], q), \quad \forall q \in H.$$

La dérivée directionnelle de J s'explicite alors sous la forme

$$J'(a)(\delta_a) = \langle \nabla_a \mathcal{L}(a, u[a], q), \delta_a \rangle_M + \langle \nabla_u \mathcal{L}(a, u[a], q), \dot{u}[a] \rangle_U.$$

L'idée est de choisir q de manière à ce que cette expression ne dépende plus de $\dot{u}[a]$. En particulier, on prend $q = p$ où p est l'état adjoint solution du problème

$$\nabla_u \mathcal{L}(a, u[a], p) = 0.$$

De manière équivalente, p est solution de l'équation linéaire

$$\nabla_u F(a, u[a])^*[p] = -H^*[H(u[a]) - d_{\text{obs}}],$$

ce qui permet enfin d'obtenir l'expression du gradient de J sous la forme

$$\nabla J(a) = (\nabla_a F(a, u[a]))^* \, p.$$

Chapitre 3

Décomposition en valeurs singulières (SVD) et régularisation de type Tikhonov

Dans le cas d'un problème inverse linéaire mal posé, une première technique de régularisation consiste à utiliser une **troncature spectrale** de l'opérateur linéaire associé, à partir de sa décomposition en valeurs singulières, que l'on appelle SVD. L'objectif de cette partie est de montrer le lien entre cette approche et les techniques de régularisation de type **Tikhonov**, lesquelles peuvent également s'appliquer dans le cas non linéaire.

Soient E et F deux espaces de Hilbert, et $A \in \mathcal{L}(E,F)$ un opérateur linéaire et continu. Pour un $\hat{z} \in F$, on s'intéresse au problème de moindres carrés suivant :

$$\min_{x \in E} \frac{1}{2}\|Ax - \hat{z}\|_F^2.$$

La première question concerne l'existence et l'unicité d'une solution. L'unicité dépend de l'injectivité de A, tandis que l'étude de l'existence conduit à l'équation d'Euler :

$$A^*(Ax - \hat{z}) = 0,$$

qui admet une solution si et seulement si $\hat{z} \in \mathrm{Im}(A) \oplus \mathrm{Im}(A)^\perp$.

En effet, si $\hat{z} \in \mathrm{Im}(A) \oplus \mathrm{Im}(A)^\perp$, alors il existe $(z_1, z_2) \in \mathrm{Im}(A) \times \mathrm{Im}(A)^\perp$ tels que $\hat{z} = z_1 + z_2$. En particulier, il existe $x_1 \in E$ tel que $Ax_1 = z_1$ et $A^*z_2 = 0$, ce qui montre que

$$A^*(Ax_1 - \hat{z}) = 0,$$

et donc x_1 est une solution du problème.

Inversement, si $A^*(Ax - \hat{z}) = 0$ admet une solution, alors il existe $x \in E$ tel que $Ax - \hat{z} = r \in \ker(A^*)$, et donc

$$\hat{z} = Ax - r.$$

Ce résultat permet en particulier de vérifier la propriété suivante :

Proposition 3.1 (*Problème des moindres carrés*)

Si $\hat{z} \in \mathrm{Im}(A) \oplus \mathrm{Im}(A)^\perp$, alors le problème des moindres carrés admet au moins une solution. De plus, il existe une solution unique de norme minimale.

On rappelle par ailleurs que $\ker(A^*) = \overline{\mathrm{Im}(A)}^\perp = \mathrm{Im}(A)^\perp$, et donc :

$$F = \overline{\mathrm{Im}(A)} \oplus \ker(A^*) = \overline{\mathrm{Im}(A)} \oplus \mathrm{Im}(A)^\perp.$$

Dans la section suivante, nous verrons que l'image de A doit être fermée si l'on souhaite que ce problème inverse soit bien posé.

1 Continuité de l'opérateur A^{-1} et image de A

L'opérateur A^{-1} est alors bien défini sur l'image de A si A est injectif :

$$A^{-1} : \mathrm{Im}(A) \longrightarrow E, \quad v \mapsto A^{-1}(v).$$

Proposition 3.2

L'opérateur A^{-1} est continu si et seulement si l'image de A est fermée dans F.

La démonstration repose sur le théorème de l'application ouverte.

Théorème 3.3 (*Application ouverte*)

Soient E et F deux espaces de Banach, et soit $f : E \to F$ une application linéaire continue. Si f est surjective, alors f est ouverte, c'est-à-dire que l'image par f de tout ouvert de E est un ouvert de F. En particulier, si f est surjective, son inverse bijectif est continu.

Ainsi, l'inverse d'un opérateur linéaire bijectif est continu.
Preuve de la proposition :

- A^{-1} *continu* $\Rightarrow$ $\mathrm{Im}(A)$ *fermée dans* F *:*
  Comme A est injectif, $\mathrm{Im}(A^{-1}) = E$ et est donc fermée. De plus, A^{-1} étant continu, on en déduit que $\mathrm{Im}(A) = (A^{-1})^{-1}(E)$ est fermée.
- $\mathrm{Im}(A)$ *fermée dans* $F \Rightarrow A^{-1}$ *continu :*
  Comme $W = \mathrm{Im}(A)$ est fermé, W est un espace de Hilbert. On introduit alors l'opérateur $\tilde{A} : E \to W$, défini par $\tilde{A}(u) = A(u)$ pour tout $u \in E$. Cet opérateur est linéaire, continu et bijectif. D'après le théorème de l'application ouverte, $\tilde{A}^{-1}$ est continu.

Exemple 2 : Soit $A : E \to E$ un opérateur auto-adjoint, et soit $\sigma(A)$ son spectre, défini comme l'ensemble des valeurs propres de A :

$$\sigma(A) = \{\lambda_n \in \mathbb{C} \mid \exists u_n \in E \text{ tel que } Au_n = \lambda_n u_n\}.$$

Alors :

- La continuité de A implique que le spectre $\sigma(A)$ est borné.
- L'injectivité de A implique que $0 \notin \sigma(A)$.
- La continuité de A^{-1} implique qu'il n'existe pas de suite $\{\lambda_n\}_{n\in\mathbb{N}} \subset \sigma(A)$ telle que $\lim_{n\to\infty} \lambda_n = 0$.

En particulier, le dernier point peut se démontrer par l'absurde. Supposons qu'il existe une suite $\{\lambda_n\}_{n\in\mathbb{N}} \subset \sigma(A)$ telle que $\lambda_n \to 0$. On peut alors introduire les deux suites u_N et v_N définies par

$$u_N = \sum_{n=1}^{N} \alpha_n e_n, \quad v_N = \sum_{n=1}^{N} \frac{\alpha_n}{\lambda_n} e_n,$$

où $\{e_n\}_n$ est telle que $Ae_n = \lambda_n e_n$.

En remarquant que $u_N \in \mathrm{Im}(A)$ car $Av_N = u_N$, il suffit de choisir la suite $\{\alpha_n\}_n$ de sorte que $\sum \alpha_n^2$ converge mais $\sum (\frac{\alpha_n}{\lambda_n})^2$ diverge. On en déduit alors que l'image de A n'est pas fermée.

Exemple 3 : Soit $\Omega \subset \mathbb{R}^d$ un ouvert borné et régulier, et soit A l'injection canonique de $H_0^1(\Omega)$ dans $L^2(\Omega)$:

$$A : H_0^1(\Omega) \longrightarrow L^2(\Omega), \quad u \mapsto u.$$

Cet exemple est intéressant car il est contre-intuitif. En effet, l'opérateur A est continu et injectif, ce qui résulte de l'inégalité de Poincaré :

$$\|u\|_{L^2(\Omega)} \leqslant C\|u\|_{H_0^1(\Omega)}.$$

En revanche, l'opérateur A^{-1} n'est pas continu car l'image de A, $\mathrm{Im}(A) = H_0^1(\Omega)$, n'est pas fermée dans $L^2(\Omega)$.

La solution du problème des moindres carrés existe et est unique si $\hat{z} \in \mathrm{Im}(A) = H_0^1(\Omega)$.

En revanche, si $\hat{z} \in L^2(\Omega) \backslash H_0^1(\Omega)$, une telle solution n'existe pas. En effet, on peut construire une suite $\{z_\varepsilon\} \subset H_0^1(\Omega)$ telle que $z_\varepsilon \to \hat{z}$ dans $L^2(\Omega)$, puisque $H_0^1(\Omega)$ est dense dans $L^2(\Omega)$.

2 Décomposition en valeurs singulières

On suppose tout d'abord que $A : E \to F$ est une application linéaire continue, avec $E = \mathbb{R}^N$ et $F = \mathbb{R}^M$, et $N < M$.

L'opérateur A^*A étant symétrique, il existe une base orthonormée $\{u_n\}_{n=1}^N$ de E telle que

$$A^*Au_n = \lambda_n^2 u_n, \quad \text{avec } \lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_N \geqslant 0.$$

La famille $\{v_n = \frac{1}{\lambda_n} Au_n\}_{n=1}^N$ forme une famille orthonormée de $\mathbb{R}^M$ et vérifie

$$Au_n = \lambda_n v_n, \quad A^*v_n = \lambda_n u_n, \quad \text{et} \quad AA^*v_n = \lambda_n^2 v_n.$$

On en déduit alors la décomposition en valeurs singulières suivante :

$$A = \sum_{n=1}^N \lambda_n \, v_n \otimes u_n, \quad \text{et} \quad A^* = \sum_{n=1}^N \lambda_n \, u_n \otimes v_n.$$

2.1 Solution des moindres carrés

La solution x^* du problème des moindres carrés de norme minimale s'exprime sous la forme :

$$x^* = \sum_{n=1}^N \frac{\langle \hat{z}, v_n \rangle}{\lambda_n} u_n.$$

En effet, on a :

$$A^*\hat{z} = \sum_{n=1}^N \lambda_n \langle \hat{z}, v_n \rangle u_n, \quad \text{et} \quad A^*Ax = \sum_{n=1}^N \lambda_n^2 \langle x, u_n \rangle u_n.$$

On peut également en déduire la pseudo-inverse de A, notée $A^\dagger$, donnée par :

$$A^\dagger = \sum_{n=1}^N \frac{1}{\lambda_n} u_n \otimes v_n, \quad \text{et} \quad A^\dagger \hat{z} = \sum_{n=1}^N \frac{\langle \hat{z}, v_n \rangle}{\lambda_n} u_n.$$

2.2 Point de vue matricielle

D'un point de vue matricielle, la décomposition SVD s'explicite sous la forme

$$A = U\Lambda V^T \quad A^* = V\Lambda U^T \quad \text{et } A^\perp = V\Lambda^{-1}U^T,$$

avec

$$U = [u_1, u_2, \cdots, u_N], V = [v_1, v_2, \cdots, v_N], \text{ et } \Lambda = diag(\lambda_n)$$

Exemple 4 : Dans le cas de la matrice A,

$$A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix},$$

on vérifie

$$U = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, V = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \text{et} \quad \Lambda = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}.$$

2.3 Cas opérateur compact en dimension infinie

La généralisation s'effectue facilement dans le cas d'opérateurs compacts, où :

- la famille $\{u_n\}_n$ forme une base de $\ker(A)^\perp$,
- la famille $\{v_n\}_n$ forme une base de $\overline{\mathrm{Im}(A)}$.

De plus, si $\hat{z} \in \overline{\mathrm{Im}(A)}$ et si

$$\sum_n \frac{|\langle \hat{z}, v_n \rangle|^2}{\lambda_n^2} < \infty,$$

alors le problème des moindres carrés admet les solutions :

$$x = \sum_n \frac{\langle \hat{z}, v_n \rangle}{\lambda_n} u_n + \ker(A).$$

Exemple 5 :[Cas de l'injection canonique de $H_0^1([0,1])$ dans $L^2([0,1])$]
L'opérateur $A : H_0^1([0,1]) \to L^2([0,1])$ est défini par

$$Au = u, \quad \forall u \in H_0^1([0,1]).$$

Concernant l'opérateur adjoint A^*, on remarque que pour tout $(u,v) \in H_0^1([0,1]) \cap L^2([0,1])$:

$$\int_0^1 (Au)(x)v(x)\,dx = \langle Au, v\rangle_{L^2} = \langle u, A^*v\rangle_{H_0^1} = \int_0^1 u'(x)(A^*v)'(x)\,dx.$$

On en déduit que $w = A^*v$ est défini comme la solution du problème suivant :

$$\begin{cases} -w''(x) = u(x), & \forall x \in [0,1], \\ w(0) = w(1) = 0. \end{cases}$$

La décomposition SVD de A s'identifie à :

$$u_n(x) = \frac{\sin(\pi n x)}{\pi n}, \quad v_n(x) = \sin(\pi n x), \quad \text{et} \quad \lambda_n = \frac{1}{\pi n}.$$

Elle correspond à la série de Fourier de u :

$$Au = \sum_{n=1}^{\infty} \langle u, u_n\rangle_{H_0^1} \lambda_n v_n = \sum_{n=1}^{\infty} \left\langle u, \frac{\sin(\pi n x)}{\pi n} \right\rangle_{H_0^1} \frac{1}{\pi n} \sin(\pi n x).$$

3 Régularisation de Tikhonov

On rappelle que la *régularisation de Tikhonov* consiste à ajouter un terme de régularisation au problème des moindres carrés, en considérant la minimisation de :

$$J_\varepsilon(x) = \frac{1}{2}\|Ax - \hat{z}\|_F^2 + \frac{\varepsilon^2}{2}\|x\|_E^2, \quad \varepsilon > 0.$$

En particulier, l'équation d'Euler associée au problème de minimisation conduit à :

$$(A^*A + \varepsilon^2 I)x = A^*\hat{z}.$$

L'avantage de cette approche est que l'existence et l'unicité de la solution sont toujours garanties. La solution s'exprime explicitement sous la forme :

$$x_\varepsilon = (A^*A + \varepsilon^2 I)^{-1}A^*\hat{z}.$$

Cette expression est bien définie car :

$$\|(A^*A + \varepsilon^2 I)^{-1}\| \leqslant \frac{1}{\varepsilon^2},$$

et la continuité de A permet d'obtenir le contrôle suivant :

$$\|x_\varepsilon\| \leqslant \frac{1}{\varepsilon^2}\|A^*\|\,\|\hat{z}\|.$$

3.1 Point de vue SVD

L'intérêt de la *décomposition en valeurs singulières* est qu'elle permet de mieux comprendre l'effet de cette régularisation, interprétable comme une troncature spectrale.

On note $\hat{x}$ la solution de norme minimale du problème des moindres carrés sans régularisation :

$$\hat{x} = \sum_n \frac{1}{\lambda_n}\langle \hat{z}, v_n\rangle u_n.$$

L'effet de la régularisation conduit à l'équation :

$$(A^*A + \varepsilon^2 I)\hat{x}_\varepsilon = A^*\hat{z},$$

et montre que :

$$\hat{x}_\varepsilon = \sum_n \frac{\lambda_n}{\lambda_n^2 + \varepsilon^2}\langle \hat{z}, v_n\rangle u_n = \sum_n h_\varepsilon(\lambda_n)\langle \hat{z}, v_n\rangle u_n,$$

où

$$h_\varepsilon(s) = \frac{s}{s^2 + \varepsilon^2} \simeq \begin{cases} 1/s, & \text{si } s \gg \varepsilon, \\ 0, & \text{si } s \to 0. \end{cases}$$

Exemple 6 :[Injection canonique de $H_0^1([0,1])$ dans $L^2([0,1])$]

On rappelle que dans ce cas :

$$u_n = \frac{\sin(\pi n x)}{\pi n}, \quad v_n = \sin(\pi n x), \quad \lambda_n = \frac{1}{\pi n}.$$

Alors, si $\hat{z} = u \in H_0^1([0,1])$, on retrouve la série de Fourier en sinus de u :

$$\hat{x} = \sum_n \frac{1}{\lambda_n}\langle u, \sin(\pi n x)\rangle \frac{\sin(\pi n x)}{\pi n} = \sum_n c_n \sin(\pi n x).$$

D'autre part, la régularisation de Tikhonov conduit à l'expression suivante :

$$\hat{x}_\varepsilon = \sum_n \frac{c_n}{1+\pi^2 n^2 \varepsilon^2} \sin(\pi n x),$$

ce qui correspond bien à une régularisation par troncature des hautes fréquences de $\hat{z}$.

3.2 Choix du paramètre de régularisation

La question du choix du paramètre ε est toujours cruciale dans le cas de données perturbées $\hat{z}_n$. La proposition suivante permet d'optimiser la valeur de ε en contrôlant l'erreur entre la solution exacte x^* et la solution régularisée $\hat{x}_{\varepsilon,n}$.

En effet, en minimisant

$$\alpha(\varepsilon) = \|A^*\| \frac{\delta_n}{\varepsilon^2} + \frac{\varepsilon}{2},$$

on trouve que

$$\varepsilon_n^* = \left(\frac{4\|A^*\|\,\delta_n}{\|\omega\|} \right)^{1/3},$$

et que l'erreur satisfait

$$\|\hat{x}_{\varepsilon,n} - \hat{x}\| \simeq O(\delta_n^{1/3}) \simeq O(\varepsilon_n^*).$$

Proposition 3.4 (*Résultat de stabilité*)

Soient $\hat{x}_{\varepsilon,n}$ la solution du problème régularisé associé à $\hat{z}_n$, et $\hat{x}$ la solution du problème initial associé à $\hat{z}$.

On suppose que

$$\delta_n = \|\hat{z}_n - \hat{z}\|_F \to 0 \quad \textit{quand } n \to \infty,$$

et que $\hat{x} \in \mathrm{Im}(A^*)$, ce qui implique l'existence de $\omega \in F$ tel que

$$\hat{x} = A^*\omega.$$

Alors :

$$\|\hat{x}_{\varepsilon,n} - \hat{x}\| \leqslant \|A^*\| \frac{\delta_n}{\varepsilon^2} + \frac{\varepsilon}{2}\|\omega\|_F.$$

La démonstration de cette propriété consiste tout d'abord à séparer les différents types d'erreurs :

$$\|\hat{x}_{\varepsilon,n} - \hat{x}\| \leqslant \|\hat{x}_{\varepsilon,n} - \hat{x}_\varepsilon\| + \|\hat{x}_\varepsilon - \hat{x}\|.$$

L'erreur liée à l'incertitude des données est contrôlée par :

$$\|\hat{x}_{\varepsilon,n} - \hat{x}_\varepsilon\| \leqslant \frac{1}{\varepsilon^2}\|A^*\|\,\|\hat{z}_n - \hat{z}\| \leqslant \|A^*\| \frac{\delta_n}{\varepsilon^2}.$$

L'erreur de régularisation $\|\hat{x}_\varepsilon - \hat{x}\|$ s'explicite à l'aide de la décomposition en valeurs singulières :

$$\hat{x} = \sum_n \frac{1}{\lambda_n} \langle \hat{z}, v_n \rangle u_n, \quad \hat{x}_\varepsilon = \sum_n \frac{\lambda_n}{\lambda_n^2 + \varepsilon^2} \langle \hat{z}, v_n \rangle u_n.$$

En utilisant l'hypothèse $\hat{x} = A^*\omega$, on remarque que :

$$\sum_n \frac{1}{\lambda_n} \langle \hat{z}, v_n \rangle u_n = \hat{x} = A^*\omega = \sum_n \lambda_n \langle \omega, v_n \rangle u_n.$$

On en déduit donc :

$$\langle \hat{z}, v_n \rangle = \lambda_n^2 \langle \omega, v_n \rangle,$$

et ainsi :

$$\hat{x} = \sum_n \lambda_n \langle \omega, v_n \rangle u_n, \quad \hat{x}_\varepsilon = \sum_n \frac{\lambda_n^3}{\lambda_n^2 + \varepsilon^2} \langle \omega, v_n \rangle u_n.$$

Au final, on obtient :

$$\begin{aligned} \|\hat{x}_\varepsilon - \hat{x}\|^2 &= \sum_n \left(\lambda_n - \frac{\lambda_n^3}{\lambda_n^2 + \varepsilon^2} \right)^2 \langle \omega, v_n \rangle^2 \\ &= \sum_n \left(\frac{\varepsilon^2 \lambda_n}{\lambda_n^2 + \varepsilon^2} \right)^2 \langle \omega, v_n \rangle^2 \leqslant \frac{\varepsilon^2}{4} \|\omega\|_F^2. \end{aligned}$$

4 Régularisation spectrale et méthode du gradient

On a vu précédemment que la régularisation de Tikhonov revient à utiliser le filtre

$$h_\varepsilon(s) = \frac{s}{s^2 + \varepsilon^2}$$

au lieu de $1/s$ via la décomposition en valeurs singulières de A. La conséquence est de négliger le spectre de A trop proche de 0.

4.1 Troncature spectrale

Une autre idée similaire consiste à introduire une *troncature spectrale* en considérant le filtre suivant :

$$h_\varepsilon(s) = \begin{cases} \frac{1}{s}, & \text{si } s \geqslant \varepsilon, \\ 0, & \text{sinon.} \end{cases}$$

Les nouvelles solutions régularisées $\hat{x}_\varepsilon$ s'écrivent alors :

$$\hat{x}_\varepsilon = \sum_{n \leqslant N_\varepsilon} \frac{1}{\lambda_n} \langle \hat{z}, v_n \rangle u_n,$$

où

$$N_\varepsilon = \min\{n \in \mathbb{N} \mid \lambda_n < \varepsilon\}.$$

En reprenant la preuve précédente, on peut montrer un résultat analogue pour l'erreur :

$$\|\hat{x}_\varepsilon - \hat{x}\|^2 = \sum_{n > N_\varepsilon} \lambda_n^2 \langle \omega, v_n \rangle^2 \leqslant \varepsilon^2 \|\omega\|_F^2.$$

4.2 Algorithme du gradient et régularisation

On rappelle que l'*algorithme du gradient à pas de descente fixe* s'explicite sous la forme :

$$\begin{cases} x_{k+1} = x_k - \alpha\, A^*[Ax_k - \hat{z}], \\ x_0 = 0, \end{cases}$$

soit encore :

$$x_k = \alpha \sum_{j=0}^{k-1} (I - \alpha A^* A)^j A^* \hat{z}.$$

En utilisant la décomposition en valeurs singulières (SVD) de A, on en déduit :

$$\begin{aligned} x_k &= \alpha \sum_{j=0}^{k-1} \sum_n (1-\alpha\lambda_n^2)^j \lambda_n \langle \hat{z}, v_n \rangle u_n \\ &= \sum_n \frac{1-(1-\alpha\lambda_n^2)^k}{\lambda_n} \langle \hat{z}, v_n \rangle u_n \\ &= \sum_n h_{\alpha,k}(\lambda_n) \langle \hat{z}, v_n \rangle u_n, \end{aligned}$$

où

$$h_{\alpha,k}(s) = \frac{1-(1-\alpha s^2)^k}{s}.$$

En particulier, la convergence de l'algorithme du gradient est assurée si $\alpha < 1/\lambda_1^2$.
On peut également exprimer l'erreur par rapport à la solution de norme minimale $\hat{x}$:

$$\|x_k - \hat{x}\|^2 = \sum_n \left[\frac{(1-\alpha\lambda_n^2)^k}{\lambda_n} \right]^2 \langle \hat{z}, v_n \rangle^2 = \sum_n \lambda_n^2 (1-\alpha\lambda_n^2)^{2k} \langle \omega, v_n \rangle^2.$$

Enfin, si pour tout $s \in [0,1]$, $(1-s)^k \leqslant 1/(ks)$, on obtient :

$$\|x_k - \hat{x}\|^2 \leqslant \sum_n \frac{\lambda_n^2}{k\alpha\lambda_n^2} \langle \omega, v_n \rangle^2 \leqslant \frac{1}{k\alpha} \|\omega\|^2.$$

Remarque 7 : Le cas d'une *régularisation implicite*

$$x_{k+1} = x_k - \alpha \left[A^*[Ax_k - \hat{z}] - \epsilon^2 x_{k+1} \right]$$

donne un résultat analogue.

Chapitre 4

Régularisation H^1 et variante

Comme nous l'avons vu précédemment, l'utilisation d'une régularisation de Tikhonov n'est pas toujours efficace en pratique. Il peut alors être intéressant d'ajouter un *a priori de régularit* sur la solution que l'on souhaite reconstruire.

Dans le cas d'une régularité harmonique, cette approche conduit à minimiser l'énergie suivante :

$$J_\varepsilon(u) = \frac{1}{2}\|Au - \hat{z}\|_F^2 + \frac{\varepsilon^2}{2}\int_\Omega |\nabla u|^2\, dx.$$

On s'intéressera à l'existence et à l'unicité d'une telle solution ainsi qu'à ses approximations numériques.

Dans un deuxième temps, nous introduirons des variantes de la norme H^1 permettant de traiter le cas de données *régulières par morceaux*.

1 Existence de solution

On suppose ici que X est un *espace de Banach* et que $J : X \to \mathbb{R}$. On introduit également X', le dual de X :

$$X' = \{\ell : X \to \mathbb{R} \mid \ell \text{ linéaire et continue}\}.$$

On rappelle les notions de convergence dans ce contexte :

- *Convergence forte* : $x_n \to x$ si $\|x_n - x\| \to 0$,
- *Convergence faible* : $x_n \rightharpoonup x$ si $\ell(x_n) \to \ell(x)$ pour tout $\ell \in X'$,
- *Convergence faiblement étoilée* : $\ell_n \overset{*}{\rightharpoonup} \ell$ si $\ell_n(x) \to \ell(x)$ pour tout $x \in X$.

1.1 Notion de semi-continuité inférieur (sci)

On rappelle qu'une fonctionnelle J est *semi-continue inférieure* (sci) si

$$\forall x_n \to x, \quad \Longrightarrow \quad \liminf J(x_n) \geqslant J(x).$$

Elle est de plus *convexe* si, pour tout $\alpha \in [0, 1]$,

$$J(\alpha u_1 + (1-\alpha)u_2) \leqslant \alpha J(u_1) + (1-\alpha)J(u_2).$$

En particulier, une propriété intéressante est que si J est convexe et sci, alors elle est *sci faible*.

L'étude de la semi-continuité inférieure est assez technique. Par exemple, pour une fonctionnelle $J : H^1(\Omega) \to \mathbb{R}$ de la forme

$$J(u) = \int_\Omega f(x, u, \nabla u(x))\, dx,$$

la *semi-continuité inférieure faiblement* dans $H^1(\Omega)$ s'obtient si :

- Ω est un ouvert borné de $\mathbb{R}^d$,
- $f(x,z,\cdot)$ est continue pour presque tout $z \in \mathbb{R}^d$ et presque tout $x \in \Omega$,
- $f(x,z,\cdot)$ est convexe pour tout $z \in \mathbb{R}^d$ et presque tout $x \in \Omega$,
- Il existe $a \in L^1(\Omega)$ et $b \in L^2(\Omega;\mathbb{R}^{d\times N})$ tels que

$$f(x,z,\xi) \geqslant a(x) + b(x)\cdot\xi, \quad \text{pour presque tout } x \in \Omega, \text{ et pour tout } (z,\xi) \in \mathbb{R}^d \times \mathbb{R}^{d\times N}.$$

1.2 Méthode directe du calcul des variations.

Concernant le problème de minimisation

$$\inf_{u\in X} J(u),$$

où X est un *espace de Banach*, l'existence d'un minimiseur s'obtient facilement dans le cas où la fonctionnelle J est *coercive* ($\lim_{\|x\|\to\infty} J(x) = \infty$) et *semi-continue inférieurement faiblement*.

En effet, il suffit de considérer une *suite minimisante* $\{x_n\}_n$ de J et de remarquer :

- Comme il s'agit d'une suite minimisante, il existe un réel $M > 0$ tel que $\forall n \in \mathbb{N},\ J(x_n) < M$.
- Comme J est coercive, on en déduit que $\{x_n\}_n$ est bornée dans X.
- À une sous-séquence près, on obtient l'existence de $x^* \in X$ telle que $x_n \rightharpoonup x^*$.
- La propriété de *semi-continuité inférieure faible* implique alors :

$$J(x^*) \leqslant \inf_n J(x_n).$$

1.3 Application à la régularisation H^1

On considère une fonctionnelle de la forme

$$J_\varepsilon(u) = \frac{1}{2}\int_\Omega (Au(x) - \hat{z}(x))^2\,dx + \frac{\varepsilon^2}{2}\int_\Omega |\nabla u|^2\,dx,$$

où A est un opérateur *continu* de $H^1(\Omega)$ dans $L^2(\Omega)$.

Existence

Dans ce cas précis, J_ε est *convexe* et *semi-continue inférieurement* dans $H^1(\Omega)$. Elle est de plus *coercive* si le noyau de A ne contient pas les constantes. On obtient alors l'existence en appliquant directement la *méthode directe du calcul des variations.*

Unicité

Concernant l'unicité, il suffit d'examiner l'équation d'Euler :

$$A^*(Au - \hat{z}) - \varepsilon^2\Delta u = 0.$$

En supposant l'existence de deux solutions u_1 et u_2, l'équation d'Euler implique :

$$A^*A(u_1 - u_2) = \varepsilon^2\Delta(u_1 - u_2),$$

et donc

$$\|A(u_1 - u_2)\|^2 = -\varepsilon^2\|\nabla(u_1 - u_2)\|^2.$$

On en déduit que $A(u_1 - u_2) = 0$ et $\nabla(u_1 - u_2) = 0$, ce qui implique l'existence d'une constante $c \in \mathbb{R}$ telle que

$$u_1 = u_2 + c.$$

Ainsi, l'unicité est vérifiée si le *noyau de A ne contient pas les constantes.*

2 Flot de gradient et discrétisation numérique

On considère dans cette section le *flot de gradient* de J_ε :

$$u_t = \varepsilon^2 \Delta u - A^*(Au - \hat{z}),$$

et différentes *discrétisations numériques* de cette EDP.

Méthode d'Euler implicite

La *méthode d'Euler implicite* consiste à traiter le gradient de J_ε de manière implicite :

$$\frac{u^{n+1} - u^n}{\delta_t} = \varepsilon^2 \Delta u^{n+1} - A^*(Au^{n+1} - \hat{z}),$$

et revient, d'un point de vue variationnel, à minimiser la fonctionnelle suivante :

$$u^{n+1} = \arg\min_u \left\{ \frac{1}{2\delta_t} \|u - u^n\|^2 + J_\varepsilon(u) \right\}.$$

En particulier, on en déduit la *décroissance de* J_ε avec :

$$J_\varepsilon(u^{n+1}) - J_\varepsilon(u^n) \leqslant -\frac{1}{2\delta_t} \|u^{n+1} - u^n\|^2,$$

sans condition de stabilité sur le pas de temps δ_t.

Méthode d'Euler explicite

La *méthode d'Euler explicite* consiste à traiter le gradient de J_ε de manière explicite :

$$\frac{u^{n+1} - u^n}{\delta_t} = \varepsilon^2 \Delta u^n - A^*(Au^n - \hat{z}),$$

et revient, d'un point de vue variationnel, à minimiser la fonctionnelle suivante :

$$u^{n+1} = \arg\min_u \left\{ \frac{1}{2\delta_t} \|u - u^n\|^2 + J_\varepsilon(u^n) + \langle \nabla J_\varepsilon(u^n), u - u^n \rangle \right\}.$$

En particulier, cette approche conduit à une *condition de stabilité conditionnelle* de la forme :

$$\delta_t < \frac{2}{\|A\|^2} \quad \text{et} \quad \delta_t \leqslant \frac{2\varepsilon^2}{\delta_x^2},$$

pour assurer la décroissance de J_ε.

Méthode d'Euler semi-implicite

Lorsque l'opérateur A est relativement *lisse* et que la contrainte $\delta_t < \frac{2}{\|A\|^2}$ n'est pas très restrictive par rapport à la norme issue du laplacien ($\delta_t \leqslant \frac{2\varepsilon^2}{\delta_x^2}$), il peut être intéressant de traiter A de manière *explicite* tout en gardant un traitement *implicite* pour le terme H^1.

Cela conduit au schéma numérique suivant :

$$\frac{u^{n+1} - u^n}{\delta_t} = \varepsilon^2 \Delta u^{n+1} - A^*(Au^n - \hat{z}),$$

qui s'écrit, d'un point de vue variationnel, sous la forme :

$$u^{n+1} = \arg\min_u \left\{ \frac{1}{2\delta_t}\|u-u^n\|^2 + \frac{\varepsilon^2}{2}\int_\Omega |\nabla u|^2\,dx + \langle A^*(Au^n-\hat{z}), u-u^n\rangle \right\}.$$

La *décroissance de J* s'obtient alors sous la condition $\delta_t < \frac{2}{\|A\|^2}$. Il s'agit en fait d'une méthode de *splitting* des deux opérateurs :

- *Flot de A de manière explicite* :

$$u^{n+1/2} = u^n - \delta_t A^*(Au^n - \hat{z})$$

- *Flot de la chaleur de manière implicite* :

$$u^{n+1} = [I_d - \delta_t \varepsilon^2 \Delta]^{-1} u^{n+1/2},$$

dont l'expression s'obtient facilement avec la *transformée de Fourier* :

$$\mathcal{F}[u^{n+1}](\xi) = \frac{1}{1+4\pi^2|\xi|^2\delta_t}\,\mathcal{F}[u^n](\xi).$$

En particulier, la figure 4.1 présente un code *Matlab* permettant d'appliquer un pas de descente de l'équation de la chaleur afin de *débruiter* une image donnée. La figure 4.2 montre le résultat numérique obtenu avec cette technique.

```
cam = im2double(imread('cameraman.tif'));
imagesc(cam);
N = size(cam,1)

%%%%%%%%%Ajout de bruit %%%%%%%%%%%%%%ù
I = cam.*(1 + 0.1*randn(size(cam)));
I_fourier = fft2(I);
imagesc(I);
caxis([0,1]);
colormap('gray');

%%%%%%%% Reg H1 // equation d'Euler Lagrange %%%%%%
epsilon = 0.005;
k = [0:N/2,-N/2+1:-1];
[K1,K2] = meshgrid(k,k);
Delta = -4*pi^2*(K1.^2 + K2.^2);
u_star = real(ifft2(I_fourier./(-epsilon^2*Delta + 1)));

subplot(1,2,1);
imagesc(I);
caxis([0,1]);
subplot(1,2,2);
imagesc(u_star)
caxis([0,1]);
```

FIGURE 4.1 – Code **Matlab** pour minimiser la norme H^1

3 Autres modèles de régularisation

Comme nous l'avons vu précédemment, une *régularisation* H^1 correspondant à la minimisation de l'énergie

$$J_\varepsilon(u) = \frac{1}{2}\|Au-\hat{z}\|_F^2 + \frac{\varepsilon^2}{2}\int_\Omega |\nabla u|^2\,dx,$$

s'obtient naturellement via l'approximation numérique du *flot de gradient* associé :

$$u_t = \varepsilon^2\Delta u - A^*(Au-\hat{z}).$$

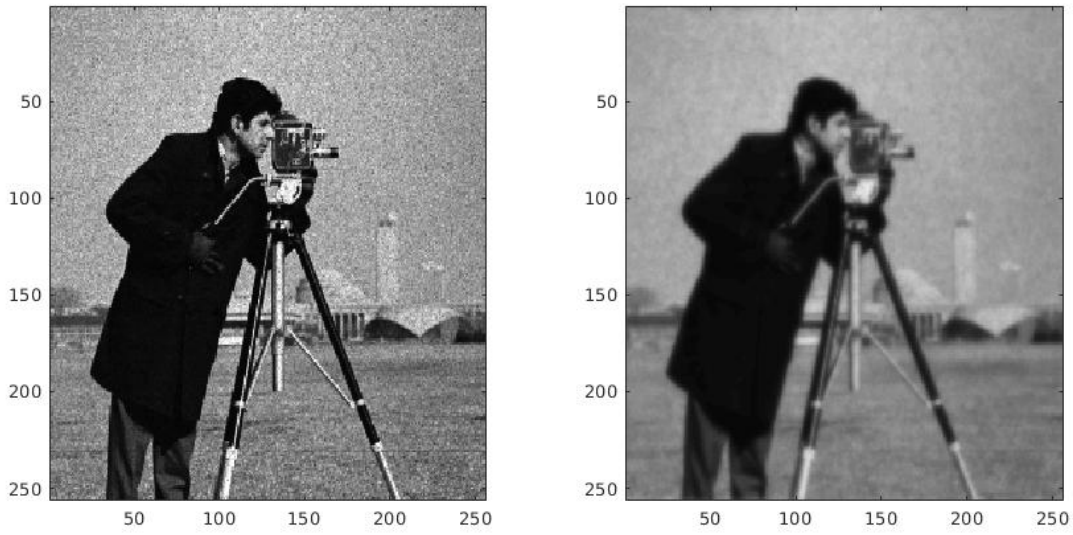

FIGURE 4.2 – Exemple de dé-bruitage d'image en utilisant $A = I_d$

Cette approche permet ainsi de retrouver la solution u du problème inverse lorsque celle-ci est suffisamment *régulière*. Cependant, lorsque la solution u est *régulière par morceaux*, une telle méthode tend à *lisser les contours* de u.

Dans cette section, nous allons introduire deux variantes qui permettent de *régulariser la solution tout en autorisant des discontinuités*.

3.1 Approche de Perona-Malick

L'approche de *Perona-Malik* consiste à généraliser le *flot* précédent en considérant l'équation

$$u_t = \varepsilon^2 \text{ div } \big(c(|\nabla u|^2)\nabla u\big) - A^*(Au - \hat{z}),$$

où le *coefficient de diffusion* c est choisi afin de lisser uniquement les zones *lisses* et de préserver les *contours*.

Choix du coefficient de diffusion c

En pratique, les *contours* sont localisés dans les zones où $|\nabla u| \simeq \infty$, et l'on souhaite donc fixer $c \simeq 0$ pour ces zones, avec

$$\lim_{s\to\infty} c(s) = 0.$$

Par ailleurs, les zones lisses correspondent à $\|\nabla u\| \ll 1$, ce qui conduit à imposer

$$c(0) = 1.$$

Plus précisément, en introduisant la *base locale* (η, ξ) définie par $\eta = \frac{\nabla u}{|\nabla u|}$ et $\xi = \frac{\nabla^\perp u}{|\nabla u|}$, l'équation précédente s'exprime sous la forme :

$$\begin{aligned} \text{div}(c(|\nabla u|^2)\nabla u) &= c(|\nabla u|^2)\Delta u + 2c'(|\nabla u|^2)|\nabla u|^2\langle\nabla^2 u\frac{\nabla u}{|\nabla u|^2}, \nabla u\rangle \\ &= c(|\nabla u|^2)\,\partial^2_{\xi\xi}u + \big(c(|\nabla u|^2) + 2c'(|\nabla u|^2)|\nabla u|^2\big)\,\partial^2_{\eta\eta}u. \end{aligned}$$

En particulier, chercher à imposer

$$c(|\nabla u|^2) + 2c'(|\nabla u|^2)|\nabla u|^2 = 0$$

permet de diffuser la solution u uniquement dans la direction *orthogonale à son gradient*, conservant ainsi les contours. Cette idée conduit à rechercher c comme solution de l'équation :

$$c(s^2) + 2s^2c'(s^2) = 0,$$

qui s'explicite sous la forme $c(s) = 1/\sqrt{s}$. Malheureusement, cette solution n'est pas compatible avec les conditions aux limites $c(0) = 1$ et $c(\infty) = 0$. En pratique, on choisit :

$$c(s) = \frac{1}{\sqrt{1+s}}.$$

Au final, l'équation de *Perona-Malik* s'écrit :

$$u_t = \varepsilon^2 \text{ div } \left(\frac{\nabla u}{\sqrt{1+|\nabla u|^2}} \right) - A^*(Au - \hat{z}).$$

Remarque : L'étude d'existence de ce type d'équation nécessite l'utilisation de la *théorie des opérateurs monotones*. La discrétisation numérique de ces équations n'est pas triviale, mais peut s'effectuer via des *schémas semi-implicites*.

Résolution numérique

L'objectif de cette partie est de proposer un *schéma numérique* simple et stable. Pour cela, on suppose pour l'instant que $c : \Omega \to [0, 1]$. On s'intéresse ainsi à l'équation

$$u_t = \varepsilon^2 \text{ div } (c(x)\nabla u) - (u - \hat{z}), \quad \text{avec } 0 < c \leqslant 1.$$

Cette équation est *variationnelle* car elle peut être obtenue comme le *flot de gradient* de l'énergie

$$J_\varepsilon(u) = \frac{\varepsilon^2}{2} \int_\Omega c(x)|\nabla u|^2 dx + \frac{1}{2}\|u - \hat{z}\|^2.$$

À noter qu'une approche *implicite* conduira à la résolution d'un *système linéaire* avec un coût algorithmique élevé, alors qu'une approche *totalement explicite* soulèvera des problèmes de *stabilité*.

Un compromis consiste à introduire une approche *semi-implicite* basée sur le *splitting convexe-concave* suivant de J_ε :

$$\begin{aligned} J_\varepsilon(u) &= \underbrace{\frac{\varepsilon^2}{2} \int_\Omega |\nabla u|^2 dx + \frac{1}{2}\|u - \hat{z}\|^2}_{\text{Partie quadratique convexe}} \\ &+ \underbrace{\frac{\varepsilon^2}{2} \int_\Omega (c(x) - 1)|\nabla u|^2 dx}_{\text{Partie concave}}. \end{aligned}$$

En utilisant ce *splitting* et en traitant explicitement la partie concave, on obtient le schéma :

$$\begin{aligned} \frac{u^{n+1} - u^n}{\delta_t} &= -\nabla_u J_{\text{convexe}}(u^{n+1}) - \nabla_u J_{\text{concave}}(u^n) \\ &= \Delta u^{n+1} + \frac{1}{\varepsilon^2}(u^{n+1} - \hat{z}) + \text{div } ((c(x) - 1)\nabla u^n). \end{aligned}$$

Ce schéma est équivalent à la *minimisation implicite* de la fonctionnelle

$$\widetilde{J}_\varepsilon(u) = J_{\text{convexe}}(u) + J_{\text{concave}}(u^n) + \langle DJ_{\text{concave}}(u^n), u - u^n \rangle,$$

où la partie concave a été linéarisée. On vérifie ainsi naturellement que

$$\widetilde{J}_\varepsilon(u^{n+1}) \leqslant \widetilde{J}_\varepsilon(u^n), \quad \text{et donc } J_\varepsilon(u^{n+1}) \leqslant J_\varepsilon(u^n).$$

Dans le cas de l'équation de *Perona-Malik*, on utilise la même approche avec un traitement explicite du coefficient de diffusion c :

$$\frac{u^{n+1} - u^n}{\delta_t} = \Delta u^{n+1} + \frac{1}{\varepsilon^2}(u^{n+1} - \hat{z}) + \operatorname{div}\left((c(|\nabla u^n|^2) - 1)\nabla u^n\right).$$

La figure 4.3 présente un exemple d'implémentation *Matlab* d'un tel *schéma numérique* utilisé pour le *dé-bruitage d'image*, comme illustré à la figure 4.4. On peut ainsi remarquer le succès de la méthode pour dé-bruiter l'image tout en conservant des *contours nets*.

```
23 -   c = @(s)  1./sqrt(1 + s.^2/10);
24 -  for i=1:T/dt,
25 -    nabla_u1 = real(ifft2(2*pi*1i*K1.*u_fourier));
26 -    nabla_u2 = real(ifft2(2*pi*1i*K2.*u_fourier));
27 -    c_poids = c(sqrt(nabla_u1.^2 + nabla_u2.^2));
28      %%%%%%%%%% div((c-1) \nabla u)
29 -    Terme_explicite = -2*pi*1i*K1.*fft2((c_poids-1).*nabla_u1)...
30          - 2*pi*1i*K2.*fft2((c_poids-1).*nabla_u2);
31      %%%%%%%%%% flot semi-implicite
32 -    u_fourier =(M.*(u_fourier + dt*I_fourier - epsilon^2*dt*Terme_explicite ));
33 -    u = real(ifft2(u_fourier));
34 -  end
```

FIGURE 4.3 – Code **matlab** pour le dé-bruitage d'une image avec l'approche de Perona-Malick

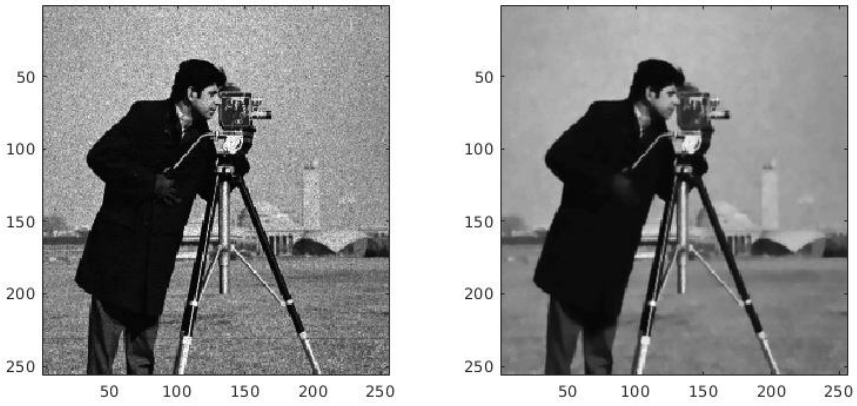

FIGURE 4.4 – Exemple de débruitage d'une image via l'approche de Perona-Malick

3.2 Fonctionnelle de Mumford-Shah

Une des difficultés de la méthode de *Perona-Malik* est que l'approche n'est pas *variationnelle*. L'idée de la fonctionnelle de *Mumford-Shah* est d'introduire une variante *variationnelle* utilisant une régularisation isotrope mais autorisant un *ensemble de discontinuités* K dans l'image.

(a) Input image (b) Approximation

FIGURE 4.5 – Exemple de régularisation obtenue avec l'approche de Mumford Shah

Plus précisément, la fonctionnelle de *Mumford-Shah* s'écrit :

$$F(u,K) = \frac{1}{2}\|u-\hat{z}\|_F^2 + \frac{\alpha}{2}\int_{\Omega\setminus K}|\nabla u|^2 dx + \beta\int_K 1\, d\sigma,$$

où l'ensemble K est aussi une variable de la fonctionnelle.

De nombreux travaux portent sur l'analyse de cette fonctionnelle, et certaines questions restent ouvertes, comme la conjecture sur la *régularité des minimisateurs* qui suggère que si (u,K) est un minimiseur de F, alors K est la réunion d'un nombre fini d'ensembles de classe $C^{1,1}$.

D'un point de vue *numérique*, cette approche soulève des difficultés, notamment sur la *discrétisation* de la variable K. La méthode de *champ de phase* consiste par exemple à approximer F à l'aide du modèle elliptique suivant :

$$\begin{aligned} F_\varepsilon(u,v) &= \frac{1}{2}\|u-\hat{z}\|_F^2 + \frac{\alpha}{2}\int_\Omega v^2\|\nabla u\|^2 dx \\ &\quad +\beta\int_\Omega\left(\frac{\varepsilon}{2}|\nabla v|^2 + \frac{1}{4\varepsilon}(v-1)^2\right)dx, \end{aligned}$$

où $\varepsilon > 0$ est un *paramètre d'approximation*.

Cette approximation est interprétée au sens de la Γ-*convergence*, définie comme suit :

$$F_\varepsilon \to F \quad \text{si} \quad \begin{cases} \forall u_\varepsilon \to u, & \liminf F_\varepsilon(u_\varepsilon) \geqslant F(u), \\ \exists u_\varepsilon \to u, & \limsup F_\varepsilon(u_\varepsilon) \leqslant F(u). \end{cases}$$

L'intérêt de cette notion est d'assurer qu'une suite minimisante de F_ε *convergera vers un minimiseur de* F. En pratique, il suffit alors de déterminer un minimiseur de F_ε pour ε suffisamment petit pour obtenir une bonne approximation d'un minimiseur de F.

De plus, trouver un minimiseur de F_ε est plus simple, car la variable v correspond à une *approximation fonctionnelle* de l'ensemble K et peut être discrétisée classiquement.

Les figures 4.5 et 4.6 montrent des exemples de *débruitage d'images* obtenus dans les travaux de Pock et al. à l'aide de cette fonctionnelle.

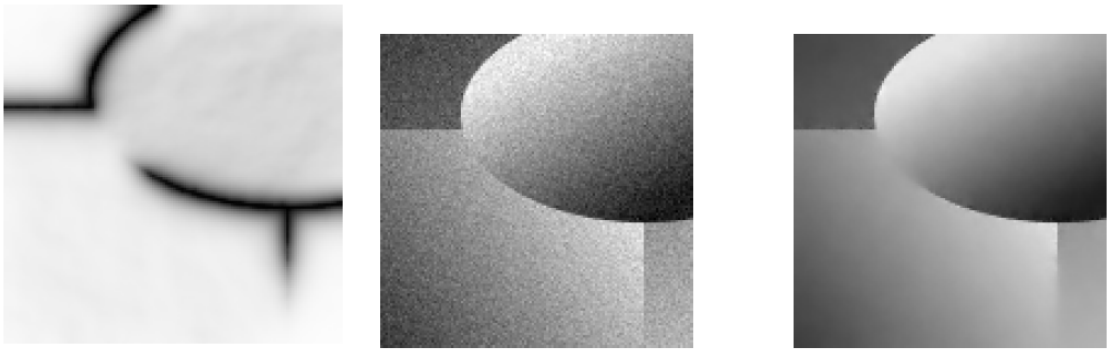

FIGURE 4.6 – Exemple de régularisation obtenue avec l'approche de Mumford Shah avec à gauche l'approximation v de K

Chapitre 5

Régularisation TV

Dans cette partie, nous nous intéressons à l'utilisation d'une régularisation TV (*variation totale*) en minimisant une énergie de la forme

$$J_\varepsilon(u) = \frac{1}{2}\|Au - \hat{z}\|_F^2 + \varepsilon \int_\Omega |\nabla u| dx,$$

où le paramètre ε est un *paramètre de régularisation*. L'intérêt est de *régulariser la solution tout en autorisant des ensembles de discontinuités*, comme le font déjà les méthodes de *Perona-Malik* et de *Mumford-Shah*, mais en proposant ici un *cadre variationnel simple*.

La minimisation pourrait naturellement s'effectuer dans l'espace $\{u\,;\, Au \in F,\, \nabla u \in L^1(\Omega)\}$, mais l'inconvénient est que l'espace de *Sobolev*

$$W^{1,1}(\Omega) = \{u \in L^1(\Omega)\,;\, \nabla u \in L^1(\Omega)\}$$

n'est pas *réflexif* et ne contient pas les fonctions qui admettent des *sauts*.

L'idée est donc de considérer l'espace BV dans lequel les gradients sont des *mesures*.

1 Espace de minimisation BV

L'idée est tout d'abord d'introduire la *norme* TV comme l'extension de la norme $\int_\Omega |\nabla u| dx$ dans le cas de fonctions u dont le gradient n'est pas dans $L^1(\Omega)$:

1.1 Variation totale

$$TV(u) = \int_\Omega |Du| = \sup_p \left\{ \int_\Omega u \operatorname{div}(p)\, dx\,;\, p \in C_c^1(\Omega; \mathbb{R}^N),\, \|p\|_\infty \leqslant 1 \right\}.$$

En particulier, si $u \in W^{1,1}(\Omega)$ alors

$$\int_\Omega |Du| = \int_\Omega |\nabla u| dx.$$

Il suffit d'utiliser $p_\varepsilon = -\frac{\nabla u}{|\nabla u|} * \rho_\varepsilon$, où ρ_ε est un *noyau régularisant*, pour obtenir

$$\int_\Omega u \operatorname{div}(p_\varepsilon)\, dx = -\int_\Omega \nabla u \cdot p_\varepsilon\, dx \to \int_\Omega |\nabla u| dx.$$

D'un autre côté, si $u = \chi_E$ est la *fonction caractéristique* d'un ensemble régulier E, alors la norme TV de u correspond au *périmètre* de E :

$$\int_\Omega |Du| = \mathrm{Per}(E) = \int_{\partial E} 1 \, d\sigma(x).$$

En effet, on a

$$\int_\Omega |Du| = \sup_p \left\{ \int_\Omega u \, \mathrm{div}(p) \, dx \, ; \, p \in C_c^1(\Omega; \mathbb{R}^N), \, \|p\|_\infty \leqslant 1 \right\},$$

avec

$$\int_\Omega u \, \mathrm{div}(p) \, dx = \int_E \mathrm{div}(p) \, dx = \int_{\partial E} p \cdot n \, d\sigma(x).$$

En choisissant pour p un *prolongement régulier de la normale* n à l'ensemble E, on en déduit

$$\int_{\partial E} p \cdot n \, d\sigma(x) = \mathrm{Per}(E).$$

Par exemple, en dimension $1D$, avec $u = \mathrm{sign}(x)$, alors $Du = 2\delta$ et $\int |Du| = 2$.

Proposition 5.1 (*Semi-continuité inférieure*)

L'application $u \mapsto \int |Du|$ est semi-continue inférieure pour la topologie $L^1(\Omega)$.

Démonstration : Soit $u_k \to u$ dans $L^1(\Omega)$. Alors, pour tout $\varphi \in C_c^1(\Omega, \mathbb{R})$ avec $\|\varphi\|_\infty \leqslant 1$, on a

$$\int_\Omega u_k \, \mathrm{div}(\varphi) \, dx \to \int_\Omega u \, \mathrm{div}(\varphi) \, dx, \quad \text{et} \quad \int_\Omega u_k \, \mathrm{div}(\varphi) \, dx \leqslant \int_\Omega |Du_k|.$$

En passant à la limite inférieure, on obtient

$$\int_\Omega u \, \mathrm{div}(\varphi) \, dx \leqslant \liminf_{k \to \infty} \int_\Omega |Du_k|.$$

En prenant le supremum sur tous les φ admissibles, on en déduit

$$\int_\Omega |Du| \leqslant \liminf_{k \to \infty} \int_\Omega |Du_k|,$$

ce qui conclut la preuve. ■

1.2 Espace BV et propriétés

L'espace $BV(\Omega)$ est l'espace de Banach composé de fonctions à variation bornée et muni de la norme

$$\|u\|_{BV} = \|u\|_{L^1} + \int_\Omega |Du|.$$

Proposition 5.2 (*Approximation régulière*)

Pour tout $u \in BV(\Omega)$, il existe une suite $(u_h) \subset C^\infty(\Omega)$ telle que

$$u_h \to u \text{ dans } L^1(\Omega) \quad \text{et} \quad \int_\Omega |\nabla u_h| dx \to \int_\Omega |Du|.$$

Proposition 5.3 (*Compacité*)

Soit Ω un ouvert borné à frontière Lipschitzienne et $\{u_k\}_k \subset BV(\Omega)$ une suite telle que $\sup_k \int_\Omega |Du_k| < \infty$.

Alors, il existe une sous-suite (non relabelée) et $u \in L^1(\Omega)$ tels que

$$u_k \to u \text{ dans } L^1(\Omega).$$

Proposition 5.4 (*Formule de la coaire*)

Soit $u \in BV(\Omega)$. Alors,

$$\int_\Omega |Du| = \int_{\mathbb{R}} Per(\{u > s\})\, ds,$$

et plus généralement, pour toute fonction g,

$$\int_\Omega g(x)|Du|dx = \int_{\mathbb{R}} \left[\int_{u=s} g(x)\, d\sigma(x) \right] ds.$$

Démonstration : Supposons $u \geqslant 0$ pour simplifier. Pour $t \geqslant 0$, posons

$$E_t = \{x \in \Omega : u(x) \geqslant t\}, \quad \chi_{E_t}(x) = \begin{cases} 1 & \text{si } u(x) \geqslant t, \\ 0 & \text{sinon.} \end{cases}$$

On peut écrire

$$u(x) = \int_0^{u(x)} 1\, dt = \int_0^\infty \chi_{E_t}(x)\, dt.$$

Ainsi,

$$\begin{aligned} \int_\Omega |Du| &= \sup_\varphi \left\{ \int_\Omega u \operatorname{div}(\varphi)\, dx \right\} \\ &= \sup_\varphi \left\{ \int_\Omega \int_0^\infty \chi_{E_t}(x) \operatorname{div}(\varphi)\, dt\, dx \right\} \\ &= \sup_\varphi \left\{ \int_0^\infty \int_{E_t} \operatorname{div}(\varphi)\, dx\, dt \right\} \\ &= \int_0^\infty \operatorname{Per}(E_t)\, dt, \end{aligned}$$

ce qui conclut la preuve. ■

1.3 Régularisation TV : existence et unicité de solution

Comme nous l'avons expliqué dans l'introduction, la régularisation TV de notre problème inverse revient à minimiser l'énergie

$$J_\varepsilon(u) = \frac{1}{2}\|Au - \hat{z}\|_F^2 + \varepsilon \int_\Omega |\nabla u|\, dx.$$

Existence de solution

L'existence s'obtient naturellement en utilisant la méthode directe du calcul des variations. Soit $(u_k)_k$ une suite minimisante de J_ε. D'après le résultat de compacité dans $BV(\Omega)$, il existe une sous-suite (non relabelée) et $u^* \in L^1(\Omega)$ telle que

$$u_k \to u^* \quad \text{dans } L^1(\Omega).$$

De plus, la semi-continuité inférieure de J_ε pour la topologie L^1 implique alors que

$$J_\varepsilon(u^*) \leqslant \liminf_{k\to\infty} J_\varepsilon(u_k),$$

et donc u^* est bien solution du problème de minimisation.

Unicité

L'équation d'Euler associée à J_ε s'écrit

$$A^* A(u - \hat{z}) - \varepsilon \operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right) = 0,$$

sous l'hypothèse que $|\nabla u| \neq 0$.

Comme pour la régularisation H^1, l'unicité de la solution nécessite que le noyau de A ne contienne pas les constantes.

2 Discrétisations numériques : premières méthodes

La discrétisation numérique de ce problème de minimisation pose un certain nombre de difficultés, car le terme TV n'est pas différentiable. Une approche explicite peut donc s'avérer délicate, et il est souvent plus efficace d'adopter une approche duale ou primal-dual.

Régularisation du terme TV

La première idée consiste à régulariser le terme TV [U+202F] :

$$\int_\Omega |Du| \simeq \int_\Omega \sqrt{|\nabla u|^2 + \sigma^2}\, dx = TV^\sigma(u),$$

dont le gradient est donné par

$$\nabla TV^\sigma(u) = -\operatorname{div}\left(\frac{\nabla u}{\sqrt{|\nabla u|^2 + \sigma^2}}\right).$$

L'avantage est que ce terme est maintenant bien défini et correspond à une régularisation de type Perona-Malick avec un coefficient de diffusion

$$c(s) = \frac{1}{\sqrt{s^2 + \sigma^2}}.$$

La figure 5.1 présente un code **Matlab** d'une telle implémentation appliquée au débruitage d'une image, et les résultats numériques sont illustrés à la figure 2.

```
%%%%%%%%%%%%% Estimation du terme div( c(|nabla u|) - 1) nabla
u_fourier = fft2(u);
gradu1 = real(ifft2(2*1i*pi*K1.*u_fourier));
gradu2 = real(ifft2(2*1i*pi*K2.*u_fourier));
Norm_grad = (gradu1.^2 + gradu2.^2);
poids = 1./sqrt(Norm_grad + sigma^2) - 1/sigma;
coef = real(ifft2(2*1i*pi*K1.*fft2(poids.*gradu1) + 2*1i*pi*K2.*fft2(poids.*gradu2)));
u =  ifft2( 1./(1 - dt*Delta*1/sigma + dt/epsilon^2).*(fft2(u + dt/epsilon^2*I + dt*(coef))));
```

FIGURE 5.1 – Exemple de script **Matlab** qui implémente une approche Perona-Malick du terme TV régularisé.

2.1 Approche de Rudin-Osher-Fatemi

Une autre technique de discrétisation, proposée par Rudin-Osher-Fatemi, consiste à introduire le schéma explicite suivant [U+202F] :

$$u_{ij}^{n+1} = u_{ij}^n + \delta_t \left[\Delta_-^1 \left(\frac{\Delta_+^1 u_{ij}^n}{\sqrt{(\Delta_+^1 u_{ij}^n)^2 + m(\Delta_+^2 u_{ij}^n, \Delta_-^2 u_{ij}^n)^2}} \right) + \Delta_-^2 \left(\frac{\Delta_+^2 u_{ij}^n}{\sqrt{(\Delta_+^2 u_{ij}^n)^2 + m(\Delta_+^1 u_{ij}^n, \Delta_-^1 u_{ij}^n)^2}} \right) \right],$$

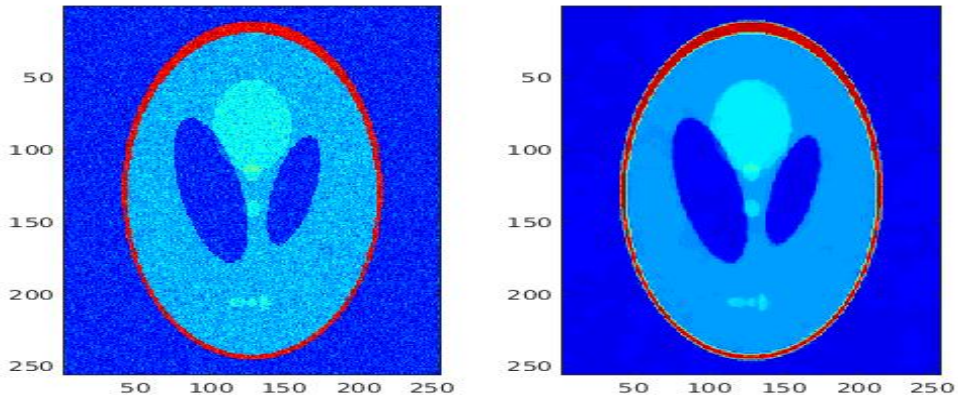

FIGURE 5.2 – Illustration numérique du débruitage obtenu à partir d'une régularisation du terme TV

où

$$m(a,b) = \min(|a|,|b|)\,\frac{\operatorname{sign}(a)+\operatorname{sign}(b)}{2}, \quad \Delta^1_+ u_{ij} = \frac{u_{i+1,j}-u_{i,j}}{h}, \quad \Delta^1_- u_{ij} = \frac{u_{i,j}-u_{i-1,j}}{h},$$

$$\Delta^2_+ u_{ij} = \frac{u_{i,j+1}-u_{i,j}}{h}, \quad \Delta^2_- u_{ij} = \frac{u_{i,j}-u_{i,j-1}}{h}.$$

L'intérêt du terme m est de choisir le bon sens de diffusion, ce qui permet d'obtenir un schéma numérique efficace et stable sous la condition naturelle de stabilité $\delta_t \leqslant C/h^2$.

Les figures 5.3 et 5.4 présentent un exemple d'implémentation **Matlab** de la discrétisation de Rudin-Osher-Fatemi et son utilisation pour débruiter une image bruitée.

```
I_p = [2:N,1]; I_m = [N,1:N-1];
m =  @(x,y) ((sign(x) + sign(y))/2).*min(abs(x),abs(y));
for i=1:T/dt,
 grad1_u_p =  (u(I_p,:) - u(:,:))/h; grad2_u_p =  (u(:,I_p) - u(:,:))/h;
 grad1_u_m =  (u(:,:) - u(I_m,:))/h; grad2_u_m =  (u(:,:) - u(:,I_m))/h;

 temp1 = grad1_u_p./sqrt(grad1_u_p.^2 + m(grad2_u_p,grad2_u_m).^2);
 temp2 = grad2_u_p./sqrt(grad2_u_p.^2 + m(grad1_u_p,grad1_u_m).^2);

 u = u + dt*( (temp1 - temp1(I_m,:))/h +  (temp2 - temp2(:,I_m))/h) - dt/epsilon^2*(u - I);
```

FIGURE 5.3 – Exemple de script **Matlab** qui implémente la discrétisation de Rudin-Osher-Fatemi.

2.2 Relaxation du terme TV

On rappelle que l'on souhaite ici minimiser la fonctionnelle

$$J(u) = \frac{1}{2\varepsilon}\int_\Omega (u-\hat{z})^2\,dx + \int_\Omega |Du|.$$

Le terme TV posant des difficultés, l'idée de la méthode de relaxation consiste à introduire la fonctionnelle $\tilde{J}$ définie par

$$\tilde{J}(u,\varphi) = \frac{1}{2\varepsilon}\int_\Omega (u-\hat{z})^2\,dx + \int_\Omega u\,\operatorname{div}(\varphi)\,dx,$$

où φ est un champ de vecteurs supposé C^1.

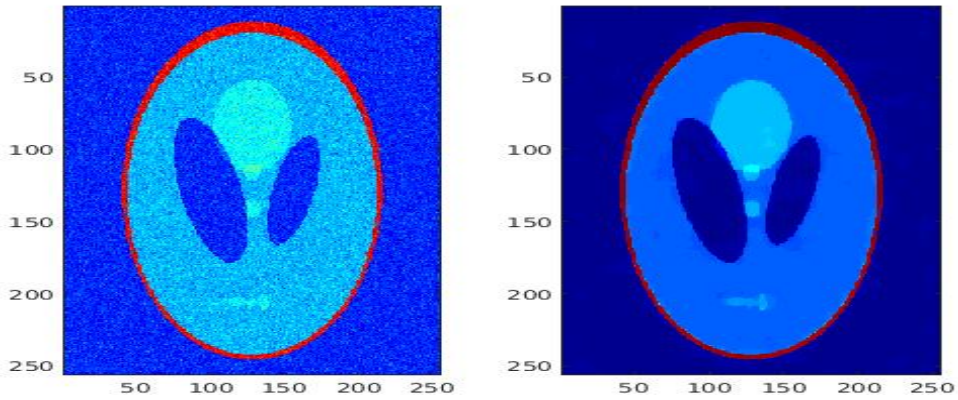

FIGURE 5.4 – Illustration numérique du débruitage obtenu à partir de la discrétisation de Rudin-Osher-Fatemi.

Le terme de relaxation vient du fait que

$$J(u) = \sup_{\varphi \in C_c^1(\Omega,\mathbb{R}^d),\, |\varphi|_\infty \leqslant 1} \tilde{J}(u,\varphi).$$

L'estimation du gradient de $\tilde{J}$ montre que

$$\begin{cases} \nabla_u \tilde{J}(u,\varphi) = \frac{1}{\varepsilon}(u-\hat{z}) + \operatorname{div}(\varphi), \\ \nabla_\varphi \tilde{J}(u,\varphi) = -\nabla u, \end{cases}$$

et conduit ainsi à l'équation d'Euler suivante :

$$u^* = \hat{z} - \varepsilon \operatorname{div}(\varphi^*).$$

Avec une initialisation

$$u^0 = \hat{z}, \quad \varphi^0 = -\frac{\nabla \hat{z}}{|\nabla \hat{z}|},$$

on peut utiliser une approche par splitting alternée :

$$\begin{cases} u^{n+1} = \hat{z} - \varepsilon \operatorname{div}(\varphi^n), \\ \varphi^{n+1} = P_{\{|\varphi|_\infty \leqslant 1\}}\big(\varphi^n - \tau \nabla u^{n+1}\big) = \frac{\varphi^n - \tau \nabla u^{n+1}}{\max(|\varphi^n - \tau \nabla u^{n+1}|, 1)}. \end{cases}$$

Ce schéma est très proche de l'approche duale proposée par Chambolle, bien que la preuve de convergence et la condition de stabilité ne soient pas triviales.

Les figures 5.5 et 5.6 présentent un exemple d'implémentation **Matlab** de ce flot de gradient relaxé et son utilisation pour débruiter une image bruitée.

Pour la discrétisation spatiale, on utilise :

$$(\nabla u)_{ij} = \begin{pmatrix} (u_{i+1,j} - u_{i,j})/h \\ (u_{i,j+1} - u_{i,j})/h \end{pmatrix}, \quad \operatorname{div}(q)_{ij} = \frac{q_1(i,j) - q_1(i-1,j)}{h} + \frac{q_2(i,j) - q_2(i,j-1)}{h}.$$

3 Approche duale et algorithme de Chambolle

L'objectif de cette section est d'introduire l'algorithme de Chambolle en effectuant le lien entre la technique de relaxation et la minimisation duale de la variation totale.

```
clear all;clf;
 I = phantom(2^8);colormap('jet');
 I = I + 0.05*randn(size(I));
 %%%%%%%%%%%%%%%%%%%%%% Approche relaxation explicite Dual %%%%%%%%%%%%%%%%%
 I_p = [2:N,1]; I_m = [N,1:N-1]; h = 1/N; epsilon = 0.02; dt = (h/2/epsilon)^2;
  u = I; q1 = zeros(N,N); q2 = zeros(N,N);
  for i=1:20,
   q1 = q1 - dt*(u(I_p,:) - u)/h;
   q2 = q2 - dt*(u(:,I_p) - u)/h;
   norm_q = sqrt(q1.^2 + q2.^2);
   q1 = q1./max(1,norm_q);
   q2 = q2./max(1,norm_q);
   u = I - epsilon^2*((q1 - q1(I_m,:))/h + (q2 - q2(:,I_m))/h);
   if (mod(i,10)==1)
       subplot(1,2,1); imagesc(I); axis square
       subplot(1,2,2); imagesc(u); axis square
       pause(0.01)
   end
  end
```

FIGURE 5.5 – Exemple de script **Matlab** qui implémente la discrétisation de la version relaxée du terme TV.

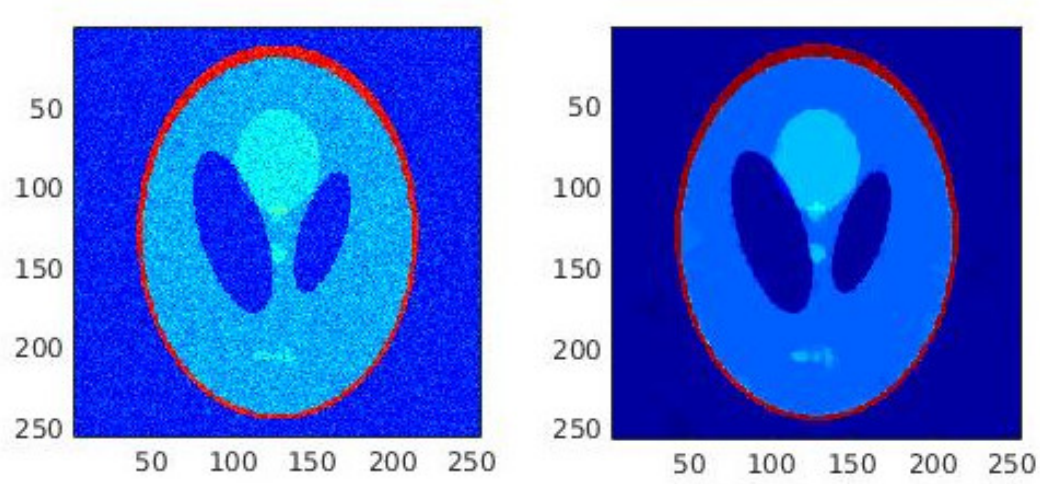

FIGURE 5.6 – Illustration numérique du débruitage obtenu à partir de la discrétisation de la version relaxée. du terme TV

3.1 Transformée de Legendre-Fenchel

Soit X un espace de Hilbert et $f : X \to \mathbb{R} \cup \{+\infty\}$. La transformée de Legendre-Fenchel de f, notée f^*, est définie pour tout $q \in X$ par

$$f^*(q) = \sup_{x \in X} \{\langle q, x\rangle - f(x)\}.$$

En particulier, f^* est convexe et semi-continue-inférieurement car, d'après sa définition, elle s'obtient comme le supremum d'applications linéaires. Les fonctions f et f^* vérifient de plus l'inégalité de Fenchel :

$$f(x) + f^*(q) \geqslant \langle x, q\rangle,$$

où l'inégalité est atteinte pour $q \in \partial f(x)$.

La transformée de Legendre-Fenchel de f^*, notée f^{**}, vérifie également :

$$f^{**}(x) = \sup_{\alpha} \{\alpha(x)\,;\, \alpha \text{ est un minorant affine de } f\}.$$

En particulier, $f^{**} \leqslant f$ et $f^{**} = f$ si f est convexe.

Exemple 8 : Dans le cas où f est une parabole

$$f(u) = \frac{1}{2\lambda}\|u - \hat{z}\|^2,$$

la transformée f^* est définie par

$$f^*(q) = \sup_u \left\{\langle u, q\rangle - \frac{1}{2\lambda}\|u - \hat{z}\|^2\right\}.$$

L'équation d'Euler de ce problème d'optimisation montre que

$$q - \frac{1}{\lambda}(u^* - \hat{z}) = 0 \quad \Longrightarrow \quad u^* = \lambda q + \hat{z}.$$

On en déduit alors :

$$f^*(q) = \frac{\lambda}{2}\|q\|^2 + \langle q, \hat{z}\rangle.$$

Exemple 9 : Pour la norme TV, $f(u) = \int_\Omega |Du|$, l'expression est également explicite. En utilisant la formulation relaxée :

$$f(u) = \sup_{\xi \in C_c^1(\Omega, \mathbb{R}^d),\, \|\xi\|_\infty \leqslant 1} \int_\Omega u \operatorname{div}(\xi)\, dx = \sup_{v \in K_{TV}} \langle u, v\rangle,$$

où

$$K_{TV} = \left\{\operatorname{div}(\xi)\,;\, \xi \in C_c^1(\Omega, \mathbb{R}^d), \|\xi\|_\infty \leqslant 1\right\}.$$

On montre alors que

$$f^*(q) = \sup_u \left\{\langle u, q\rangle - \sup_{v \in K_{TV}} \langle u, v\rangle\right\} = \begin{cases} 0 & \text{si } q \in K_{TV}, \\ +\infty & \text{sinon.} \end{cases}$$

3.2 Problème dual de la minimisation TV

Le problème de minimisation de J, où

$$J(u) = \frac{1}{2\lambda}\|u - \hat{z}\|^2 + \int |Du|,$$

s'exprime sous la forme

$$\begin{aligned}\arg\min_u J(u) &= \arg\min_u \left\{ \frac{1}{2\lambda}\|u - \hat{z}\|^2 + \int |Du| \right\} \\ &= \arg\min_u \left\{ \sup_q \left(\frac{1}{2\lambda}\|u - \hat{z}\|^2 + \langle u, q\rangle - TV^*(q) \right) \right\},\end{aligned}$$

où $TV^*(q)$ est la transformée de Legendre-Fenchel de TV.

L'équation d'Euler montre que

$$\frac{1}{\lambda}(u^* - \hat{z}) + q^* = 0 \quad \Longrightarrow \quad u^* = \hat{z} - \lambda q^*.$$

La solution duale q^* s'obtient alors en résolvant

$$q^* = \arg\max_q \left\{ \frac{1}{2\lambda}\|\lambda q\|^2 + \langle \hat{z} - \lambda q, q\rangle - TV^*(q) \right\} = \arg\max_q \left\{ -\frac{1}{2\lambda}\|\hat{z} - \lambda q\|^2 - \frac{1}{2\lambda}\|\hat{z}\|^2 - TV^*(q) \right\}.$$

En pratique, cette solution correspond à la projection de $\hat{z}/\lambda$ sur l'ensemble K_{TV}, c'est-à-dire

$$q^* = P_{K_{TV}}(\hat{z}/\lambda),$$

où $K_{TV} = \{\mathrm{div}(\xi)\,;\, \xi \in C_c^1(\Omega, \mathbb{R}^d), \|\xi\|_\infty \leqslant 1\}$.

3.3 Algorithme de Chambolle

L'algorithme de Chambolle permet de déterminer la solution u^* du problème de minimisation TV en résolvant le problème dual suivant :

$$\varphi^* = \arg\min_\varphi \left\{ \left\| \mathrm{div}(\varphi) - \frac{\hat{z}}{\lambda} \right\|^2 \;;\; |\varphi| \leqslant 1 \right\}.$$

La solution u^* s'obtient ensuite via

$$u^* = \hat{z} - \lambda q^*, \quad \text{avec } q^* = \mathrm{div}(\varphi^*).$$

L'équation d'Euler associée à ce problème d'optimisation en φ s'écrit

$$-\nabla\big(\,\mathrm{div}(\varphi) - \hat{z}/\lambda\big) + \alpha\varphi = 0,$$

où α est le multiplicateur de Lagrange associé à la contrainte $|\varphi| \leqslant 1$, défini par

$$\begin{cases} \alpha(x) > 0, & \text{si la contrainte est active en } x : |\varphi(x)| = 1, \\ \alpha(x) = 0, & \text{si la contrainte est inactive en } x : |\varphi(x)| < 1. \end{cases}$$

En pratique, α peut s'exprimer comme

$$\alpha(x) = \big|\nabla\big(\,\mathrm{div}(\varphi) - \hat{z}/\lambda\big)\big|.$$

L'idée de l'algorithme de Chambolle est alors d'utiliser le schéma d'Euler semi-implicite :

$$\varphi^{n+1} = \varphi^n + \tau \left(\nabla\big(\operatorname{div}(\varphi^n) - \hat{z}/\lambda \big) - \big| \nabla\big(\operatorname{div}(\varphi^n) - \hat{z}/\lambda \big) \big| \, \varphi^{n+1} \right),$$

qui se réécrit de manière explicite :

$$\varphi^{n+1} = \frac{\varphi^n + \tau \nabla\big(\operatorname{div}(\varphi^n) - \hat{z}/\lambda \big)}{1 + \tau \big| \nabla\big(\operatorname{div}(\varphi^n) - \hat{z}/\lambda \big) \big|} = \frac{\varphi^n - \tau \nabla u^n/\lambda}{1 - \tau |\nabla u^n|/\lambda},$$

où

$$u^n = \hat{z} - \lambda \operatorname{div}(\varphi^n).$$

Proposition 5.5

Si $\tau \leqslant \frac{1}{\kappa^2}$, alors $\lambda \operatorname{div}(\varphi^n)$ converge vers $P_{K_{TV}}(\hat{z}/\lambda)$, où κ correspond à la norme de l'opérateur divergence, bornée en dimension finie :

$$\| \operatorname{div}(\eta) \| \leqslant \kappa \|\eta\|.$$

Démonstration : La preuve repose sur la décroissance de l'énergie

$$\tilde{J}(\varphi) = \left\| \operatorname{div}(\varphi) - \frac{\hat{z}}{\lambda} \right\|^2$$

le long des itérations, c'est-à-dire $\tilde{J}(\varphi^{n+1}) \leqslant \tilde{J}(\varphi^n)$.

Posons

$$\eta = \frac{\varphi^{n+1} - \varphi^n}{\tau}.$$

On a alors

$$\begin{aligned} \tilde{J}(\varphi^{n+1}) &= \tilde{J}(\varphi^n) + 2\tau \langle \operatorname{div}(\eta), \operatorname{div}(\varphi^n) - \hat{z}/\lambda \rangle + \tau^2 \| \operatorname{div}(\eta) \|^2 \\ &\leqslant \tilde{J}(\varphi^n) - \tau \left(2\langle \eta, \nabla(\operatorname{div}(\varphi^n) - \hat{z}/\lambda) \rangle - \kappa^2 \tau \|\eta\|^2 \right), \end{aligned}$$

avec

$$\eta = \nabla(\operatorname{div}(\varphi^n) - \hat{z}/\lambda) - \rho, \quad \rho = |\nabla(\operatorname{div}(\varphi^n) - \hat{z}/\lambda)| \, \varphi^{n+1}.$$

On en déduit que

$$2\langle \eta, \nabla(\operatorname{div}(\varphi^n) - \hat{z}/\lambda) \rangle - \kappa^2 \tau \|\eta\|^2 = (1 - \kappa^2 \tau) \|\eta\|^2 + (\|\nabla(\operatorname{div}(\varphi^n) - \hat{z}/\lambda)\|^2 - \|\rho\|^2).$$

Comme

$$\varphi^{n+1} = \frac{\varphi^n + \tau \nabla(\operatorname{div}(\varphi^n) - \hat{z}/\lambda)}{1 + \tau |\nabla(\operatorname{div}(\varphi^n) - \hat{z}/\lambda)|},$$

on obtient $|\varphi^{n+1}| \leqslant 1$ et $|\rho| \leqslant |\nabla(\operatorname{div}(\varphi^n) - \hat{z}/\lambda)|$, ce qui implique finalement

$$\tilde{J}(\varphi^{n+1}) \leqslant \tilde{J}(\varphi^n) - (1 - \kappa^2 \tau) \|\eta\|^2.$$

L'énergie $\tilde{J}$ décroît donc jusqu'à $\eta = 0$, c'est-à-dire lorsque $\varphi^{n+1} - \varphi^n = 0$. On en déduit l'existence de φ^* tel que $\varphi^n \to \varphi^*$, qui satisfait

$$\varphi^* = \frac{\varphi^* + \tau \nabla(\operatorname{div}(\varphi^*) - \hat{z}/\lambda)}{1 + \tau |\nabla(\operatorname{div}(\varphi^*) - \hat{z}/\lambda)|},$$

ou encore

$$\nabla(\operatorname{div}(\varphi^*) - \hat{z}/\lambda) - |\nabla(\operatorname{div}(\varphi^*) - \hat{z}/\lambda)| \, \varphi^* = 0,$$

ce qui correspond à l'équation d'Euler du problème dual. ■

Chapitre 6

Approche ISTA-FISTA

Dans ce chapitre, nous nous intéressons à la minimisation d'une fonctionnelle

$$F(x) = f(x) + g(x),$$

où f est convexe et de classe $C^{1,1}$, c'est-à-dire $\|\nabla f(x) - \nabla f(y)\| \leqslant L(f)\|x - y\|$, et g est convexe mais non lisse.

Pour résoudre ce problème, on utilise une méthode de descente avec un traitement explicite pour f et implicite pour g :

$$x_{k+1} = \mathrm{Prox}_{\tau g}\big(x_k - \tau \nabla f(x_k)\big), \quad \text{où } \mathrm{Prox}_g(y) = \arg\min_x \left\{ \frac{1}{2}\|x - y\|^2 + g(x) \right\}.$$

Cet algorithme, appelé *ISTA*, converge relativement lentement en pratique. L'objectif de ce chapitre est de démontrer sa convergence et de proposer une version accélérée en introduisant un terme d'inertie.

Exemple 10 : Le cas du problème inverse avec la transformée de Radon et une régularisation TV correspond à utiliser pour les fonctions f et g :

$$f(u) = \frac{1}{2}\|\mathcal{R}[u] - g_{\mathrm{data}}\|^2, \quad g(u) = \int |Du|.$$

L'estimation du gradient de f s'effectue avec la transformée de Radon adjointe, et le proximal de la norme TV s'approche à l'aide de l'algorithme de Chambolle.

Exemple 11 : Une régularisation par parcimonie ℓ^1 correspond à utiliser $g(x) = \varepsilon\|x\|_1$, dont le proximal est explicite :

$$\mathrm{Prox}_{\varepsilon\|x\|_1}(y) = \mathrm{sign}(y) \cdot \max(0, |y| - \varepsilon).$$

1 Algorithme ISTA

L'algorithme de gradient semi-implicite présenté ci-dessus peut également s'exprimer sous la forme

$$x_{k+1} = P_L(x_k) = \arg\min_x \{Q_L(x, x_k)\},$$

où

$$Q_L(x, y) = f(y) + \langle x - y, \nabla f(y)\rangle + \frac{L}{2}\|x - y\|^2 + g(x).$$

Nous allons montrer la proposition suivante, qui établit la décroissance de F au cours des itérations.

Remarque 12 : On rappelle qu'un traitement explicite d'une partie de F revient à linéariser cette partie autour de x_k.

Proposition 6.1

Soit x^* une solution du problème de minimisation. Alors

$$F(x_k) - F(x^*) \leqslant \frac{L\|x_0 - x^*\|^2}{2k}.$$

Démonstration : Tout d'abord, on rappelle que les hypothèses de convexité et de régularité sur f et g impliquent

$$\begin{cases} f(x) & \geqslant f(y) + \langle \nabla f(y), x - y\rangle, \\ f(x) & \leqslant f(y) + \langle \nabla f(y), x - y\rangle + \frac{L}{2}\|x - y\|^2, \\ g(x) & \geqslant g(y) + \langle \gamma(y), x - y\rangle, \quad \gamma(y) \in \partial g(y). \end{cases}$$

Ainsi, on montre facilement que

$$\begin{aligned} F(P_L(y)) &= f(P_L(y)) + g(P_L(y)) \\ &\leqslant f(y) + \langle \nabla f(y), P_L(y) - y\rangle + \frac{L}{2}\|P_L(y) - y\|^2 + g(P_L(y)) \\ &\leqslant Q_L(P_L(y), y). \end{aligned}$$

L'équation d'Euler du problème d'optimalité $z = P_L(y)$ montre de plus que

$$\nabla f(y) + L(z - y) + \gamma(y) = 0,$$

où $\gamma(y) \in \partial g(z)$ est un sous-gradient de g en $z = P_L(y)$. Avec la convexité de g, on a

$$g(x) \geqslant g(P_L(y)) + \langle \gamma(y), x - P_L(y)\rangle.$$

Montrons que si $F(P_L(y)) \leqslant Q_L(P_L(y), y)$, alors

$$F(x) - F(P_L(y)) \geqslant \frac{L}{2}\|P_L(y) - y\|^2 + L\langle P_L(y) - y, y - x\rangle.$$

En effet,

$$\begin{aligned} A &= F(x) - F(P_L(y)) \geqslant F(x) - Q_L(P_L(y), y) \\ &\geqslant f(x) + g(x) - \left(f(y) + \langle \nabla f(y), P_L(y) - y\rangle + \frac{L}{2}\|P_L(y) - y\|^2 + g(P_L(y))\right) \\ &\geqslant \langle \nabla f(y) + \gamma(y), x - P_L(y)\rangle - \frac{L}{2}\|P_L(y) - y\|^2 \\ &\geqslant L\langle y - P_L(y), x - P_L(y)\rangle - \frac{L}{2}\|P_L(y) - y\|^2 \\ &\geqslant L\langle y - P_L(y), x - y\rangle + \frac{L}{2}\|P_L(y) - y\|^2. \end{aligned}$$

On applique cette inégalité avec $x = x^*$ et $y = x_n$:

$$\begin{aligned} F(x^*) - F(x_{n+1}) &\geqslant \frac{L}{2}\|x_{n+1} - x_n\|^2 + L\langle x_{n+1} - x_n, x_n - x^*\rangle \\ &\geqslant \frac{L}{2}\left(\|x_{n+1} - x^*\|^2 - \|x_n - x^*\|^2\right). \end{aligned}$$

On peut de plus montrer que

$$F(x_n) - F(x_{n+1}) \geqslant \frac{L}{2}\|x_{n+1} - x_n\|^2,$$

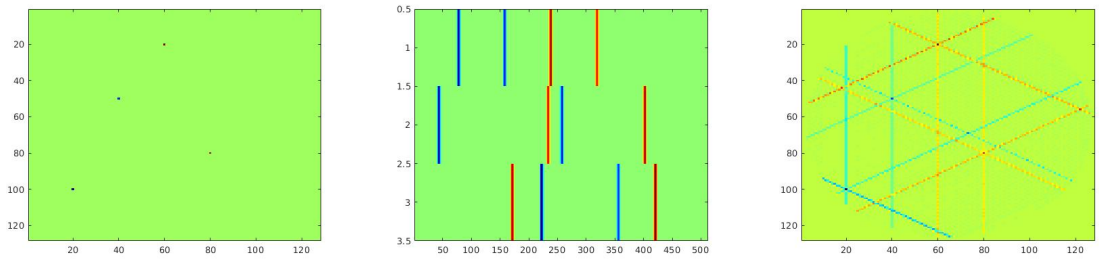

FIGURE 6.1 – Exemple de transformée de Radon d'une image sparse ; à gauche l'image, au centre, la transformée de Radon limité à trois orientation seulement, à droite, la reconstruction obtenue avec la méthode des moindres carrés sans régularisation

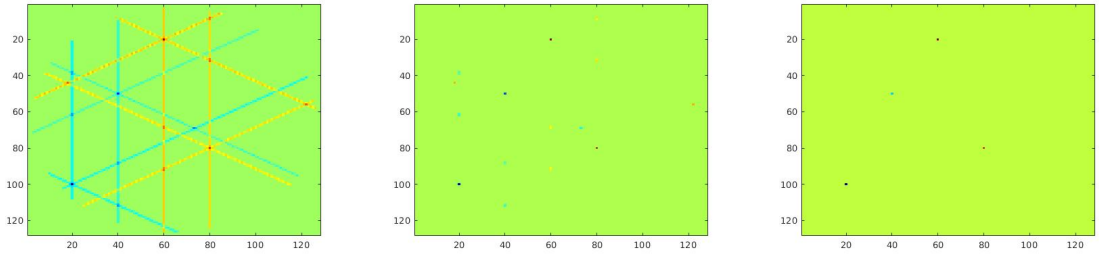

FIGURE 6.2 – Exemple de transformée de Radon d'une image sparse avec une régularisation pénalisant la norme ℓ^1 de l'image : recontrcution obtenue après 1, 10 et 100 itérations

et par sommation sur n, on obtient

$$\begin{aligned}\frac{2k}{L}\left(F(x_k)-F(x^*)\right) &\leqslant \|x^*-x_0\|^2-\|x^*-x_k\|^2-\sum_{n=0}^{k-1} n\|x_{n+1}-x_n\|^2\\ &\leqslant \|x^*-x_0\|^2.\end{aligned}$$

On obtient donc la borne désirée :

$$F(x_k)-F(x^*)\leqslant \frac{L\|x_0-x^*\|^2}{2k}.$$

■

2 Applications numériques

2.1 Cas d'une image sparse

Le premier exemple présenté à la figure 6.1 montre une reconstruction obtenue avec la méthode des moindres carrés sans régularisation, dans le cas d'une image sparse et d'une transformée de Radon limitée à seulement 3 angles. On remarquera que la reconstruction obtenue ne permet pas d'identifier correctement l'image de départ. La figure 6.2 montre, de plus, les reconstructions obtenues en appliquant l'algorithme ISTA avec une pénalisation ℓ^1 :

$$F(x)=\frac{1}{2}\|R[x]-I\|^2+\varepsilon\|x\|_1.$$

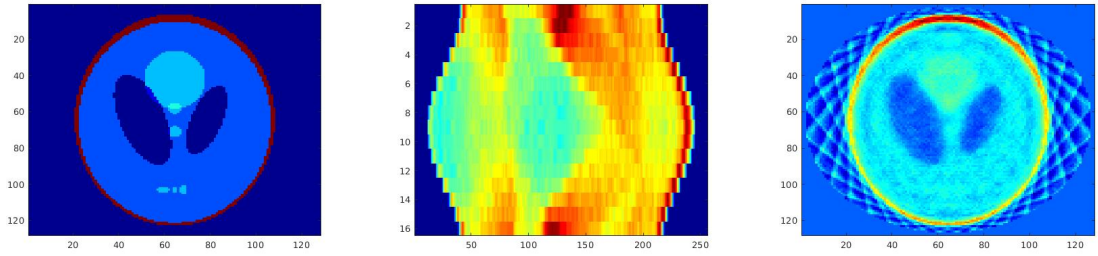

FIGURE 6.3 – Exemple de transformée de Radon d'une image constante par morceaux dans le cas d'un échantillonage grossier de la variable d'orientation θ ; à gauche l'image, au centre, la transformée de Radon , à droite, la reconstruction obtenue avec la méthode des moindres carrés sans régularisation

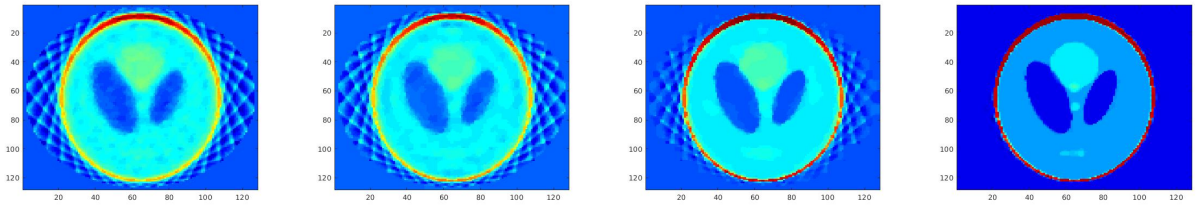

FIGURE 6.4 – Cas de la transformée de Radon d'une image constante morceaux dans le cas d'une discrétisation grossière de la variable d'orientation ; reconstructions obtenues avec une régularisation TV en appliquant l'algorithme ISTA après 50, 100,1000 et 5000 itérations

2.2 Cas d'une image régulière par morceaux

Un deuxième exemple concerne le cas d'une image constante par morceaux où, là encore, la transformée de Radon est discrétisée avec un échantillonnage limité à 2^5 orientations. La figure 6.3 présente l'image, sa transformée de Radon et la reconstruction obtenue par la méthode des moindres carrés sans régularisation. La figure 6.4 montre des exemples de reconstructions obtenues en appliquant l'algorithme ISTA avec une régularisation TV. En particulier, on remarque que la régularisation TV permet de retrouver une bonne estimation de l'image initiale, mais nécessite cependant un grand nombre d'itérations.

2.3 Algorithme FISTA

On rappelle que l'algorithme ISTA, qui s'explicite sous la forme

$$x_{k+1} = P_L(x_k),$$

n'a une vitesse de convergence en $1/k$ seulement :

$$F(x_k) - F(x^*) = O(1/k).$$

Beck et Teboulle ont alors proposé d'utiliser l'accélération de Nesterov pour gagner en efficacité, en considérant l'algorithme suivant :

$$\begin{cases} x_k = P_L(y_k), \\ t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}, \\ y_{k+1} = x_k + \frac{t_k - 1}{t_{k+1}}(x_k - x_{k-1}). \end{cases}$$

Cette stratégie leur a permis d'améliorer la vitesse de convergence, en démontrant que

$$F(x_k) - F(x^*) = O(1/k^2).$$

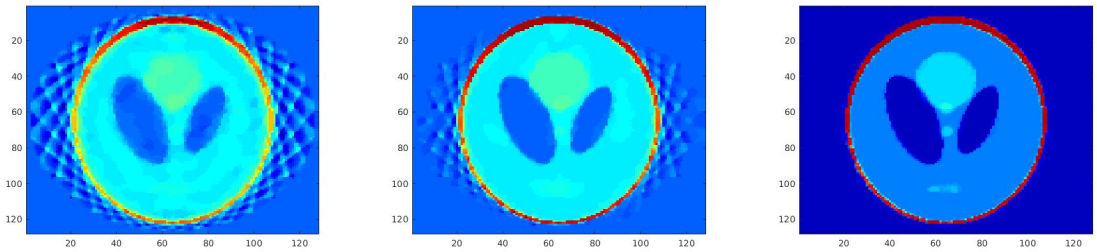

FIGURE 6.5 – Cas de la transformée de Radon d'une image constante morceaux dans le cas d'une discrétisation grossière de la variable d'orientation ; reconstructions obtenues avec une régularisation TV en appliquant l'algorithme FISTA avec respectivement 50, 100 et 200 itérations

Ce résultat est également très visible numériquement. En reprenant le cadre de l'expérience présentée à la figure 6.4, l'utilisation de l'algorithme FISTA permet d'obtenir une bonne reconstruction avec seulement 200 itérations, au lieu de 5000 pour l'algorithme ISTA. La figure 6.5 présente les reconstructions obtenues après 50, 100 et 200 itérations.