MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ ABDERRAHMANE MIRA BEJAIA
FACULTÉ DES SCIENCES EXACTES
DÉPARTEMENT DE MATHÉMATIQUES

Mémoire présenté pour l'obtention du diplôme de master en mathématiques

Spécialité : Analyse Mathématique

Par

HADDAD Djahida
OUHADDAD Samira

THÈME

PROBLÈMES MAL POSÉS ET MÉTHODES DE RÉSOLUTION

Soutenu, le 07/07/2019 devant le jury composé de :

M.F.BOUHMILA	Maitre de Conférences	U.A.M.Béjaia	*Président.*
Mme.H.BECHIR	Maitre de Conférences	U.A.M.Béjaia	*Promotrice.*
Mme.S.TAS	Professeur	U.A.M.Béjaia	*Examinatrice.*

Promotion 2018/2019

Remerciements

*Nous tenons à exprimer toute notre gratitude envers notre promotrice, Mme **H.BECHIR**, pour son soutien et la confiance qu'elle nous a accordé en nous proposant ce thème.*

*Nos remerciements sont aussi adressés à Mme **S.TAS** et M **F.BOUHMILA** qui nous font l'honneur de juger notre travail.*

Merci à tous les membres de la Faculté des Sciences Exactes en général, et aux membres du département de Mathématiques en particulier.

Enfin, nous n'oublions pas de remercier toute personne qui de près ou de loin, a contribué à la finalisation de ce travail.

Table des matières

Introduction

Un problème inverse est une situation, dans laquelle les valeurs de certains paramètres (ou inconnues) d'un modèle doivent être identifiés à partir d'observations (ou mesures) du phénomène. C'est en quelque sorte, le contraire d'un problème direct : supposons que l'on dispose d'un modèle dont on se fixe des valeurs pour les paramètres du modèle, on peut alors faire tourner le modèle, en déduire une trajectoire, et l'observer. Il s'agit du problème direct.

Le problème inverse consiste à remonter le schéma : connaissant les observations, le but est de retrouver les valeurs des paramètres.

La résolution du problème inverse passe donc en général par une étape initiale de modélisation du phénomène, dite problème direct, qui décrit comment les paramètres du modèle se traduisent en effets observables expérimentalement. Ensuite, à partir des mesures obtenues sur le phénomène réel, la démarche va consister à approximer au mieux les paramètres qui permettent de rendre compte de ces mesures.

Cette résolution peut se faire par simulation numérique ou de façon analytique. La résolution mathématique est rendue difficile par le fait que les problèmes inverses sont en général des problèmes mal posés, c'est à dire que les seules observations expérimentales ne suffisent pas à déterminer parfaitement tous les paramètres du modèle. Il est donc nécessaire d'ajouter des contraintes ou des *a priori* qui permettent de réduire l'espace des possibilités de façon à aboutir à une solution unique.

On retrouve des problèmes inverses dans de nombreux domaines scientifiques, en particulier dans l'étude de systèmes complexes pour lesquels on a accès qu'à un petit nombre de mesures, par exemple : la terre en géophysique, les tissus organiques en imagerie médicale, l'univers en cosmologie, une salle de concert en acoustique architecturale, résolution d'un système linéaire, ingénierie pétrolière, tomographie en médecine, dé convolution (en imagerie notamment), détermination des constantes d'une réaction chimique, détermination de la forme d'un obstacle par radar,...

Les problèmes inverses sont typiquement des problèmes mal posés. Cette notion de problèmes bien et mal posés a été introduite par le mathématicien français Jacques Hadamard en 1902 à propos des équations aux dérivées partielles et leurs interprétations physiques. Au sens de Hadamard, un problème est bien posé lorsque les trois conditions suivantes sont satisfaites :

1. La solution doit exister.
2. La solution doit être unique.
3. La solution doit être stable, c'est-à-dire qu'elle doit dépendre continûment des données initiales.

Le problème est mal posé "ill-posed" en anglais si l'une de ces propriétés n'est pas satisfaite.

Nous allons à présent donner de façon plus formelle l'expression d'un problème inverse linéaire. Celui-ci s'exprime sous la forme d'une équation :

$$Ax = u$$

Où A est un opérateur défini sur un espace métrique E à valeurs dans un autre espace métrique F. On cherche à trouver x à partir de u.

Ce mémoire s'intéresse aux problèmes mal posés et méthodes de résolution. Il est composé d'une introduction et de trois chapitres.

Dans le premier chapitre, nous présentons la notion d'un problème inverse , exemples et quelques méthodes de résolution des problèmes inverses.

Le second chapitre est consacré à la méthode de régularisation de Tikhonov, qui consiste à remplacer un problème mal posé par un autre bien posé de sorte que l'erreur commise soit compensée par le gain de stabilité. La principale difficulté de cette méthode est la détermination du paramètre de régularisation, ensuite nous nous intéressons à deux variantes de Tikhonov.

La première variante consiste à utiliser un contrôle différent pour la solution et la deuxième consiste à utiliser des itérations.

Dans le dernier chapitre, nous proposons une application reposant sur la Technique de débruitage d'image. Initialement, les première méthodes de débruitage consistaient à utiliser un filtre passe-bas (suppression de hautes fréquences) ce qui avait pour inconvénient d'atténuer les contours de l'image. Pour parer à ce problème, de nouvelles techniques plus performantes (qui dépendent de la direction) et utilisant des équations aux dérivées partielles, ont été élaborées.

Nous allons partir d'une « simple » équation. Puis nous baserons notre réflexion autour de celle-ci. Elle nous permettra d'en tirer plusieurs méthodes de restaurations que nous appliquerons.

Cette modeste contribution s'achèvera par une conclusion.

Chapitre 1

Problèmes inverses et méthodes de résolution

1.1 Problème inverse

Dans le livre **"Inverse problems"**[4], l'auteur introduit la définition d'un problème inverse. Il s'agit d'un problème qui consiste à déterminer les causes connaissant les effets.
Ce problème est l'inverse d'un autre appelé problème direct qui détermine les effets, les causes étant connues.

L'étude des problèmes inverses est compliquée et ça est dû à la possibilité d'avoir plusieurs solutions, car des causes différentes ménent parfois aux mêmes effets. Des informations supplémentaires sont néccessaires pour obtenir l'unicité de la solution.

Une autre difficulté pratique de l'étude des problèmes inverses est qu'elle demande souvent une bonne connaissance du problème direct, donc le succés dans la résolution d'un problème inverse repose en général sur des éléments spécifiques à ce problème. Il existe toutefois quelques techniques qui possédent un domaine d'application étendu.

Parmi les domaines dans lesquels les problèmes inverses jouent un rôle important nous pouvons citer : l'imagerie médical (échographie, scanners, rayous x,...), l'ingéniere pétrolière (prospection par des méthodes sismiques, magnétiques, identification des perméabilités dans un réservoir...), l'hydrogéologie(identification des perméabilités hydrauliques), la chimie(détermination des constantes de réaction), le radar (détermination de la forme d'un obstacle), la mécanique quantique (détermination du potentiel), le traitement d'image (restauration d'images floues)...etc

La plupart des problèmes inverses ne satisfait pas à la définition d'un problème **bien-posé**, on les appelle problèmes **mal-posés**.

Exemple de problème inverse

On s'intéresse à l'estimation de paramètres dans une équation aux dérivées partielles :

$$\begin{cases} \frac{\partial y}{\partial t} - \sum_{i=1}^{n} \frac{\partial}{\partial x_i}(a\frac{\partial y}{\partial x_i}) = f \ dans \ \Omega\times]0,T[\\ y(x,0) = y_0(x) \ dans \ \Omega \\ \frac{\partial y}{\partial n} = g \ sur \ \Gamma\times]0,T[\end{cases}$$

C'est l'équation de la chaleur, y est la température, f est un terme source, a est la conductivité thermique, et g est le flux de chaleur (entrant ou sortant). On peut utiliser la même équation pour modéliser un écoulement monophasique (comme du pétrole) : y est la pression, f représente les puits de pompage, a est la perméabilité du milieu et $g = 0$ pour un milieu fermé.

Le problème est le suivant : à partir de mesures de y en certains points et à certains instants, il faut identifier a. Le problème direct est évidemment trivial, mais le problème inverse peut être des plus compliqués.

1.2 Problème bien posé et mal posé

Le Mathématicien français **Jaques Hadamard** a défini le concept d'un problème bien posé. Il s'agit d'un problème dont :

- La solution existe.
- Est unique.
- Est stable (elle dépend continûment des données).

Un problème qui n'est pas bien-posé au sens de la définition précédente est dit **mal-posé**.

La non-existence et la non-unicité de la solution d'un problème mal-posé sont sans doute des difficultés sérieuses mais on peut les rectifier.

Cependant le problème de la continuité est plus problématique, en particulier en vue d'une résolution approchée ou numérique, c'est-à-dire il ne sera pas possible (indépendament de la méthode numérique) d'approcher d'une manière satisfaisante la solution du problème inverse (car les données disponibles seront bruitées donc proches, mais différentes des données réelles).

1.3 Exemples de problèmes mal posés

On présente ici quelques exemples simples de problèmes mal-posés

Exemple 1 :"problème polynômial"

Le nombre de racines réelles du polynôme $p(x) = -x^2 + 3x - m$ varie de façon discontinue quand m varie continûment sur la droite réelle. En effet,

$$\Delta = 9 - 4m$$

- Si $m > \frac{9}{4}$, $\Delta < 0 \implies$ pas de racines réelles.
- Si $m < \frac{9}{4}$, $\Delta > 0 \implies$ deux racines réelles distinctes.
- Si $m = \frac{9}{4}$, $\Delta = 0 \implies$ une racine double.

Exemple 2 :"problème de matrice mal conditionnée"

On souhaite résoudre le système linéaire $Ax = u$, où A est la matrice donnée par :

$$A = \begin{pmatrix} 11 & 8 & 9 & 8 \\ 8 & 6 & 7 & 6 \\ 9 & 7 & 11 & 10 \\ 8 & 6 & 10 & 11 \end{pmatrix} \text{ et } u = \begin{pmatrix} 36 \\ 27 \\ 37 \\ 35 \end{pmatrix} \text{ alors on trouve : } x = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Prenons maintenant un second membre u^* très légèrement différent de u,

$$\text{Soit : } u^* = \begin{pmatrix} 36.1 \\ 26.9 \\ 37.1 \\ 34.9 \end{pmatrix}$$

$$\text{On vérifie alors que la solution de } Ax = u^* \text{ est : } x = \begin{pmatrix} \frac{60}{7} \\ \frac{-79}{7} \\ \frac{32}{7} \\ \frac{-8}{7} \end{pmatrix} = \begin{pmatrix} 8.57 \\ -11.28 \\ 4.57 \\ -1.14 \end{pmatrix}$$

On voit que de très petites variations de u ont conduit à de grandes variations sur x.

Dans cet exemple, de façon précise, $cond(A) = 546.06$ (conditionnement de la matrice A qui est défini par :

$$cond(A) = \|A\|\|A^{-1}\|$$

Où la norme choisie est la norme matricielle associées à $\|.\|_2$ sur $\mathbb{R}^4$).
Ce phénomène de mauvais conditionnement explique en partie, la difficulté de prévoir certains phénomènes.
Les appareils de mesure ne sont jamais fiable et il est impossible de connaitre exactement u, cela peut entraîner une très grande imprécision sur la valeur de x.

Exemple 3 :

Pour tout fonction intégrable ψ l'équation

$$\int_0^1 (3x^2t + xt^2 + t^3)\psi(t)dt = sinx$$

n'admet pas de solution.

En effet, le membre de gauche de l'équation est un pôlynome de degré 2 et le membre de droite est un sinus.

Exemple 4 :

Soit $f \in \mathcal{C}^1([0,1])$, $\delta \in]0,1[$ et $n \in \mathbb{N}(n \geq 2)$, on définit :

$$f_n^\delta(t) = f(t) + \delta \sin \frac{nt}{\delta},\ t \in [0,1]$$

Alors

$$(f_n^\delta)'(t) = f'(t) + n \cos \frac{nt}{\delta},\ t \in [0,1]$$

$$\|f_n^\delta - f\|_\infty = \sup_{0\leq t\leq 1} |f_n^\delta(t) - f(t)| = \sup_{0\leq t\leq 1} |\delta \sin \frac{nt}{\delta}| = \delta$$

Et

$$\|(f_n^\delta)' - f'\|_\infty = \sup_{0\leq t\leq 1} |(f_n^\delta)'(t) - f'(t)| = \sup_{0\leq t\leq 1} |n \cos \frac{nt}{\delta}| = n$$

Si on considère f la donnée exacte et f_n^δ la donnée bruitée. Pour δ petit, l'erreur sur les dérivées est grande.

1.4 Méthodes de résolution des problèmes inverses

On rappelle que dans le cas des problèmes inverses linéaires, le problème discret s'écrit comme

$$Ax = u \tag{1.1}$$

Où A est une matrice réctangulaire.
Si on souhaite résoudre l'équation $Ax = u$ on a deux cas :

1. Si A est carré et inversible, la solution est unique et simple, elle s'exprime par $x = A^{-1}u$.
2. Si A n'est pas inversible, de plus si on suppose que le problème est mal posé, plusieurs méthodes de résolution existent.

On va brièvement décrire, dans un premier temps les méthodes basées sur la résolution du système par factorisation de matrices et dans un second temps les méthodes de régularisation.

1.4.1 Factorisation de matrices

Les deux méthodes (les plus utilisées) décrites dans ce paraghraphe et qui sont basées sur la factorisation de matrices, sont applicables lorsque le problème (1.1) est mal posé au sens de deux premières conditions de Hadamard (la non-existence et la non unicité), mais la matrice A est bien conditionnée.

Factorisation de Cholesky

On rappelle qu'une matrice S admet la décomposition de Cholesky s'il existe une matrice triangulaire inférieure B telle que $S = BB^t$ (B^t étant la transposée de B). Sachant que la matrice A du système (1.1) peut être réctangulaire, de taille $m \times n$ avec $m \geq n$, pour appliquer cette factorisation, on considère l'équation :

$$A^t Ax = A^t u \tag{1.2}$$

$$(A^t A \; matrice \; d'information)$$

Avec une matrice A^tA, symétrique, définie positive et donc admettant la décomposition de **Cholesky**, c'est à dire qu'il existe une matrice triangulaire inférieure, notée B, vérifiant

$$A^tA = BB^t.$$

La résolution du système triangulaire et qui est :

$$\begin{cases} Bb = A^tu \\ B^tx = b \end{cases}$$

Remarques 1.4.1 *1. Dans le cas où la matrice A^tA n'est pas définie positive, on utilise la décomposition $B\overline{B}$ (la matrice B est triangulaire inférieure, $\overline{B}$ est sa transposée avec des signes"$-$" sur sa diagonale)*

2. La solution du système (1.2) coïncide avec l'expression :

$$x = (A^tA)^{-1}A^tu$$

La matrice $N = (A^tA)^{-1}A^t$ est appelée l'inverse général de la matrice A.

Exemple 1.4.1 *Pour la régularisation du système $Ax = u$ avec*

$$u = \begin{pmatrix} -2 \\ 2 \\ 2 \end{pmatrix} \text{ et } A = \begin{pmatrix} -1 & -1 \\ 1 & 0 \\ 1 & 1 \end{pmatrix}$$

La normalisation, conduit à la factorisation de la matrice symétrique définie positive A^tA en

$$A^tA = \begin{pmatrix} 3 & 2 \\ 2 & 2 \end{pmatrix} = BB^t \text{ avec } B = \begin{pmatrix} \sqrt{3} & 0 \\ \frac{2}{\sqrt{3}} & \frac{\sqrt{2}}{\sqrt{3}} \end{pmatrix}$$

La résolution des deux systèmes triangulaires

$$\begin{cases} Bb = A^tu = (5,4)^t \\ B^tx = b \end{cases}$$

conduit à la solution exacte $x = (1,1)^t$

Factorisation de QB

On rappelle qu'une matrice S admet la décomposition QB s'il existe une matrice orthogonale Q ($QQ^t = I$) et une matrice triangulaire inférieure B telle que $S = QB$.

La matrice A du système (1.1) peut être réctangulaire de taille $m \times n$ avec $m \geq n$. La factorisation QB, va être appliquée à la matrice A complétée par des colonnes nulles de sorte que la matrice $(A, 0)$ soit d'ordre m. On obtient alors la factorisation $(A, 0) = Q(B, 0)$ où Q est une matrice orthogonale d'ordre m et B est une matrice d'ordre n, mais $(B, 0)$ est d'ordre m. La solution du système (1.1) est alors solution du système triangulaire

$$\begin{cases} Bx = Z_1 \\ Z_1 \ est \ donne \ par \ Q^t u = (Z_1, Z_2)^t \end{cases}$$

Exemple 1.4.2 *La matrice* $A = \begin{pmatrix} -1 & -1 \\ 1 & 0 \\ 1 & 1 \end{pmatrix}$

admet la factorisation $A = Q\overline{B}$, *où* $Q = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ *et* $\overline{B} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 1 \end{pmatrix} \left(B = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \right)$

La solution du système $Ax = u$, *avec* $u = (-2, 1, 2)$ *est donnée par* $Bx = Z_1$ *:*
Le vecteur Z_1 *est obtenu de l'équation :*

$$Q^t u = (1, 2, 2)^t$$

On pose $z_1 = (1, 2)$ *cela entraine que* $x = (1, 1)^t$

1.4.2 La régularisation

Régulariser un problème mal posé, c'est le remplacer par un autre bien posé, de sorte que "l'erreur commise" soit compensée par le gain de stabilité.

Méthodes directes :

La méthode de Tikhonov

Par exemple, pour résoudre le problème de l'instabilité du système linéaire du type : $Ax = u$, mal posé au sens de la troisième condition de **Hadamard**, on utilise la technique de régularisation développée par **Tikhonov** et **Arsenine**[8]. On se raméne à estimer la solution par une minimisation du problème régularisé défini par :

$$\min_x \frac{1}{2}\|Ax - u\|^2 + \frac{\alpha^2}{2}\|R(x)\|^2 \tag{1.3}$$

Avec α le coefficient de régularisation et $R(.)$ est l'opérateur de régularisation de la fonction coût (1.3).

Le choix du paramètre de régularisation α provient de connaissance d'information statistique sur le bruit de mesure.

Le choix de $R(.)$ revient à introduire un *a priori* dans la solution, il est donc important d'avoir une idée de la solution pour faire son choix.

Lorsque R est une matrice, la fonctionnelle à minimiser devient :

$$\min_x \frac{1}{2}\|Ax - u\|^2 + \frac{\alpha^2}{2}\|Rx\|^2 \tag{1.4}$$

Pour R matrice de **Tikhonov** (matrice de régularisation), les différentes formes sont :

- **Régularisation d'ordre 0 :**

$R = \mathbb{I}_d$ où $\mathbb{I}_d$ est la matrice identité.

La minimisation de la fonction coût (équation (1.4)) conduit à l'expression explicite de l'estimateur régularisé

$$x = (A^t A + \alpha^2 \mathbb{I}^t \mathbb{I})^{-1} A^t u$$

- **Régularisation d'ordre 1 :**

Exemple : $R \in M(5,5)$

$$R = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- **Régularisation d'ordre 2 :**

Définition 1.4.1 *On appelle* ***argument de minimum****, noté* ***arg min****, l'ensemble des points en lesquels une expression atteint sa valeur minimale.*

$$\arg\min_x f(x) := \{x/\forall y : f(y) \geq f(x)\}$$

Où encore, $\arg\min_x f(x)$ *est la valeur de* x *pour laquelle* $f(x)$ *atteint la plus petite valeur.*

Exemple : $R \in M(5,5)$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

La solution de problème (1.3) s'exprime comme :

$$x = \arg\min \frac{1}{2}\|Ax - u\|^2 + \frac{\alpha^2}{2}\|Rx\|^2$$

Le principe de Morozov

On donne ici un exemple de méthode de choix à posteriori du paramètre du régularisation. On expose la plus classique de celles ci, "the discrepancy principle" de **Morozov**[7], ou le principe de décalage de **Morozov**.

On présente un principe basé sur la méthode de régularisation de **Tikhonov**.(**kirch**).

On suppose que $A : E \longrightarrow F$ est un opérateur compact et injectif défini entre les deux espaces de Hilbert E et F avec une image dense $Im(A) \subset F$.

En étudie encore l'équation $Ax = u$, $u \in F$, on calcule maintenant le paramètre de régularisation $\alpha = \alpha(\delta) > 0$, tel que la solution de **Tikhonov** correspondante est la solution de l'équation :

$$\alpha x^{\alpha,\delta} + A^* A x^{\alpha,\delta} = A^* u^\delta$$

et elle est le minimum de :

$$J_\alpha(x) = \|Ax - u\|^2 + \alpha\|x\|^2$$

qui satisfait à l'équation :

$$\|Ax^{\alpha,\delta} - u^{\delta}\| = \delta$$

On note que le choix de α par "le principe de décalage" garantit d'une part que l'erreur est δ, d'autre part, α est très petit.

Définition 1.4.2 *Une famille d'opérateurs linéaires bornés $R_\alpha : F \longrightarrow E$ est une "stratégie de régularisation" si*

$$\forall x \in E \ , \ \lim_{\alpha \longrightarrow 0} R_\alpha Ax = x$$

i.e l'opérateur $R_\alpha A$ converge simplement vers l'identité.

Définition 1.4.3 *Une stratégie de régularisation $\delta \longrightarrow \alpha(\delta)$ est **admissible** si pour tout $x \in E$*

$$\lim_{\delta \longrightarrow 0} \alpha(\delta) = 0$$

Et

$$\lim_{\delta \longrightarrow 0} \sup_{u^\delta \in F} \{\|R_{\alpha(\delta)}u^\delta - x\|_F \ ; tel \ que \ \|Ax - u^\delta\|_F \leq \delta\} = 0$$

Théorème 1.4.1 *Soit $A : E \longrightarrow F$ est un opérateur linéaire et compact avec une image dense dans F*

Soit $Ax = u$, $x \in E$, $u \in F$, $u^\delta \in F$ tels que :

$$\|u - u^\delta\| \leq \delta < \|u^\delta\|$$

*Soit $x^{\alpha(\delta)}$ la solution de **Tikhonov** satisfaisant*

$$\|Ax^{\alpha(\delta),\delta} - u^\delta\| = \delta, \ pour \ tout \ \delta \in (0, \delta_0).$$

Alors :

1. *$x^{\alpha(\delta),\delta} \longrightarrow x$ pour $\delta \longrightarrow 0$. (Donc "le principe de décalage" est admissible).*
2. *Soit $x = A^*u \in A^*(F)$ avec $\|u\| \leq K$, alors*

$$\|x^{\alpha(\delta),\delta} - x\| \leq 2\sqrt{\delta K}$$

Pour cela "le principe de décalage" est une stratégie de régularisation optimale sous la condition

$$\|(A^*)^{-1}x\| \leq K$$

► La démonstration de ce théorème est dans [5] .p.48.

La détermination de $\alpha(\delta)$ est équivalente au problème qui consiste à trouver la racine de la fonction monotone :

$$\phi(\alpha) = \|Ax^{\alpha(\delta),\delta} - u^{\delta}\|^2 - \delta^2$$

pour δ fixé. Il n'est pas nécessaire de satisfaire l'équation $\|Ax^{\alpha,\delta} - u^{\delta}\| = \delta$, donc la double inégalité suivante :

$$c_1\delta \leq \|Ax^{\alpha(\delta),\delta} - u^{\delta}\| \leq c_2\delta$$

est suffisante pour prouver les assertions du théorème précédent.

La méthode de la décomposition en valeurs singulières (S.V.D)

Une des méthode de résolution d'un problème inverse est la décomposition en valeurs singulières, elle concerne les opérateurs compact, elle est donnée comme suit :

Si $A : E \longrightarrow F$ un opérateur compact, E et F étant des espaces de Hilbert, alors il existe deux bases orthonormées $(x_k)_{k\in\mathbb{N}}$ de E et $(u_k)_{k\in\mathbb{N}}$ de F et une suite strictement positive $b_k \longrightarrow 0$ quand k tend vers l'infini, telle que :

$$Ax_k = b_k u_k \ , \ A^*u_k = b_k x_k \ , \ A^*Ax_k = b_k^2 x_k \ , \ AA^*u_k = b_k^2 u_k$$

C'est ce qui est appelé décomposition en valeurs singulières de l'opérateur A^*A.

Le triplet (x_k, u_k, b_k) est appelé système singulier. Cette méthode donne une solution exprimée par :

$$x = \sum_{k=1}^{\infty} \frac{(Ax, u_k)}{b_k} x_k$$

De manière générale, la décomposition d'une matrice est une factorisation (méthodes factorielles) en un produit de matrices sous une forme donnée, ce qui facilite son étude. Le procédé de décomposition en valeurs singulières s'apparente en algèbre linéaire à un outil de factorisation des matrices rectangulaires $(m \times n)$ réelles ou complexes. La décomposition en valeurs propres par contre, ne s'applique que pour certaines matrices carrées. Néanmoins, dans certains cas, les deux décompositions restent liées. Par exemple pour une matrice hermitienne (ou auto-adjointe), semi-définie positive, les valeurs singulières et vecteurs singuliers correspondent aux valeurs propres et vecteurs propres de la matrice. D'un point de vue géométrique, la SVD de la matrice A, fournit deux bases de vecteurs orthonormées, à savoir les colonnes des matrices U et V, telles que la matrice devient diagonale une fois transformée dans ces bases. Ceci constitue la principale différence avec la diagonalisation, bien que toute matrice possède une décomposition en valeurs singulières, seules les matrices normales sont diagonalisables dans une base orthonormée.

Décomposition sur une base d'ondelettes

La méthode présentée dans cette partie repose sur une procédure de régularisation couplée avec une décomposition sur une base d'ondelettes.

• **Les ondelettes, la transformation et la synthèse du signal :**

Transformer un signal revient simplement à présenter différement ce signal sans perdre l'information qu'il contient. En traitement du signal, il est commun d'utiliser la transformée de "Fourier" du signal afin d'en obtenir une nouvelle représentation temps-fréquence.

Pour un signal donné f fonction du temps, la transformée de Fourier de f donnée par la relation suivante :

$$F(\omega) = \int_{-\infty}^{+\infty} f(t) exp(-j\omega t) dt \tag{1.5}$$

Les coefficients de la transformée de Fourier $F(\omega)$ sont alors multipliés par une exponentielle de pulsation ω afin de définir les composantes harmoniques du signal original à cette fréquence :

$$f(t) = \int_{-\infty}^{+\infty} F(\omega) exp(j\omega t) d\omega \tag{1.6}$$

Une extension de la transformé de *Fourier* pour l'étude locale d'un signal est la transformée de Fourier à fenêtre glissante (short-time Fourier transform (STFT) en anglais) qui utilise cette fois une fonction de fenêtrage g (fonction à support compact ou décroissance rapide). Ses coefficients, notés $C(\tau, \omega)$ sont donnés dans le cas continu par la relation suivante :

$$C(\tau, \omega) = \int_{-\infty}^{+\infty} f(t) g(t-\tau) exp(-j\omega t) dt \tag{1.7}$$

Plutôt que de présenter le signal par sa décomposition temps-fréquence, la transfomée en ondelettes (ou TO) utilise une représentation temps -échelle. La transformation repose sur la multiplication du signal initial f, non plus par une exponentielle complexe, mais par une ondelettes ψ_w translatée et mise à l'echelle) de a (paramètre d'échelle) et b (paramètre de position).

Les coefficients d'ondelettes, notés $C(a, b)$, associés au signal f sont alors donnés par la relation suivante :

$$C(a, b) = \int_{\mathbb{R}} f(t) \frac{1}{\sqrt{a}} \psi_w\left(\frac{t-b}{a}\right) dt \ , \ a \in \mathbb{R}_+^*, \ b \in \mathbb{R} \tag{1.8}$$

L'ondelette a pour avantage d'être une onde ($\int_{-\infty}^{+\infty} \psi_w(t)dt = 0$) localisée, dont l'energie finie ($\int_{-\infty}^{+\infty} (\psi_w)^2(t)dt < \infty$) est concentrée en temps afin d'étudier les signaux non stationnaires. La TO fût crée initialement pour supplanter la transformée de Fourier à temps court(STFT). En effet, alors que la SFTF utilise la même résolution à tous les niveaux de fréquence du signal, différentes échelles sont analysées à différentes résolutions par la TO. Pour les petites échelles (relatives aux hautes fréquences), la TO offre une bonne résolution temporelle mais une faible résolution frèquentielle alors que pour les grandes échelles (relatives aux basses fréquences), elle fournit une bonne résolution frèquentielle mais une faible résolution temporelle. Cette résolution adaptative à l'échelle sied parfaitement aux signaux qui présentent des variations rapides pour des laps de temps courts, d'où la haute résolution temporelle. Concernant les signaux avec épisodes de basses fréquences (souvent plus long), ils ne nécessitent pas une résolution temporelle élevée.

Les méthodes itératives :

Les méthodes itératives sont des méthodes qui construisent une suite de solutions approchées lesquelles (dans le cas non bruité) convergent vers la solution désirée. Dans le contexte des problèmes inverses la situation est plus compliquée : en présence de bruit, la suite construite par la méthode itérative ne converge pas, en général, vers une solution du problème de départ. Il est encore une fois, nécessaire de régulariser le processus itératif et c'est l'indice d'itération lui-même qui joue le rôle de paramètre de régularisation. En d'autres termes, il convient d'arrêter les itérations plus tôt qu'on ne le ferait dans un cas non bruité.

Nous examinous dans ce paraghraphe une des méthodes itératives : **la méthode de Landweber** [6], qui a pour principal avantage de se prêter à une analyse simple. Elle converge trop lentement pour être utilisable en pratique.

Landweber a proposé de réécrire l'équation $Ax = u$ sous la forme :

$$x = (I - aA^*A)x + aA^*u$$

Pour $a > 0$, le schéma itératif de cette équation est le suivant :

$$x^0 = 0 \; et \; x^m = (I - aA^*A)x^{m-1} + aA^*u \tag{1.9}$$

$$pour\ m = 1, 2, ...$$

Lemme 1.4.1 *Soit la suite* (x^m) *définie par* (1.9) *et on définit la fonctionnelle* $\psi : E \longrightarrow \mathbb{R}$ *par :*

$$\psi(x) = \frac{1}{2}\|Ax - u\|^2\ ,\ x \in E$$

Alors ψ *est différentiable au sens de fréchet pour tout* $y \in E$ *et*

$$\psi^{'}(y)x = Re(Ay - u, Ax) = Re(A^*(Ay - u), x)\ ,\ x \in E \tag{1.10}$$

La fonctionnelle linéaire $\psi^{'}(y)$ *peut être identifiée avec* $A^*(Ay - u) \in E$ *sur l'espace de Hilbert* E.

Remarque :

Il est facile d'avoir la forme explicite $x^m = R_m u$, *où l'opérateur* $R_m : F \longrightarrow E$ *est défini par :*

$$R_m = a\sum_{k=0}^{m-1}(\mathbb{I} - aA^*A)^k A^*\ ,\ m = 1, 2, ... \tag{1.11}$$

Théorème 1.4.2 *1. Soit* $A : E \longrightarrow F$, *un opérateur compact et soit* $0 < a < \frac{1}{\|A\|^2}$.

On définit les opérateurs linéaires et bornés $R_m : F \longrightarrow E$ *par* (1.11). *Ces opérateurs définissent une stratégie de régularisation de paramètre* $\alpha = \frac{1}{m}$, $m \in \mathbb{N}$ *et* $\|R_m\| \leq \sqrt{am}$.

La suite $x^{m,\,\delta} = R_m u^\delta$ *est calculée par les itérations suivantes*

$$x^{0,\delta} = 0\ et\ x^{m,\,\delta} = (\mathbb{I} - aA^*A)x^{m-1,\,\delta} + aA^*u^\delta\ pour\ m = 1, 2, ...$$

Tout stratégie $m(\delta) \longrightarrow \infty(\delta \longrightarrow 0)$ *avec* $\delta^2 m(\delta) \longrightarrow 0(\delta \longrightarrow 0)$ *est admissible.*

2. Soit $x = A^*y \in Im(A^*)$ *avec* $\|u\| \leq k$ *et* $0 < c_1 < c_2$, *pour chaque choix* $m(\delta)$, *avec* $c_1\frac{k}{\delta} \leq m(\delta) \leq c_2\frac{k}{\delta}$, *l'estimation suivante est vérifiée :*

$$\|x^{m,\,\delta} - x\| \leq c_3\sqrt{\delta k}.$$

Pour c_3 *qui dépend de* c_1 , c_2 *et* a, *alors l'itération de Landweber est optimale pour* $\|(A^*)^{-1}x\| \leq K$.

3. A présent, soit $x = A^*Ay \in Im(A^*A)$, $\|y\| \leq k$ *et* $0 < c_1 < c_2$, *pour chaque choix* $m(\delta)$ *avec* $c_1(\frac{k}{\delta})^{\frac{2}{3}} \leq m(\delta) \leq c_2(\frac{k}{\delta})^{\frac{k}{\delta}}$

On a :

$$\|x^{m,\,\delta} - x\| \leq c_3 k^{\frac{1}{3}}\delta^{\frac{2}{3}}$$

pour c_3 qui dépend de c_1, c_2 et a.

Pour cela, l'itération de Landweber est aussi optimale pour $\|(A^*A)^{-1}x\| \leq k$

Pour cette méthode, on remarque qu'une solution est plus précise lorsque le nombre d'itérations m est grand mais la stabilité nous oblige à garder m le plus petit possible.

- **Le cas des problèmes inverses non linéaires :**

Dans les cas des problèmes inverses non linéaires, le problème : $u = F(x)$, [où F est l'opérateur (ou l'ensemble des opérations) à effectuer sur x pour obtenir u], se transforme en un problème d'optimisation et qui consiste à chercher le minimum de la fonctionnelle définie par :

$$J(x) = \frac{1}{2}\|u - u_{ex}\|^2 \tag{1.12}$$

puis on applique une des méthodes d'optimisation pour résoudre ce type de problèmes. Des difficultés qui résident à différents niveaux d'existence, d'unicité et de stabilité se posent.
Donc on peut utiliser une technique classique permettant de trouver la solution en ajoutant une contrainte ou des contraintes à la fonction coût (1.12) et ceci en faisant intervenir les multiplicateurs de Lagrange.

Définition 1.4.4 *Les **multiplicateurs de Lagrange** est une méthode permettant de trouver les points stationnaires (maximum, minimum,...) d'une fonction dérivable d'une ou plusieurs variables.*
Cette technique est utile entre autres pour résoudre les problèmes d'optimisation sous contrainte(s) (linéaire(s)).

La méthode de gradient conjugué (rappels)

Théorème 1.4.3 *"Weierstrass"*

Si $J : E \subset \mathbb{R}^n \longrightarrow \mathbb{R}$, est continue sur un compact $I \subset E$, alors le minimum et le maximum sont atteints sur I.

Définition 1.4.5 *On dit que d est une direction admissible de descente en x pour la fonctionnelle $J(x)$ s'il existe un réel γ, tel que pour t dans l'intervalle $[0, \gamma]$, on a $J(x + td) \leq J(x)$.*
La méthode du gradient conjugué est une méthode de minimisation de descente basée sur la direction opposée à celle du gradient de la fonctionnelle J $(d = -\nabla J)$.

L'algorithme du gradient conjugué :

− On pose $k = 0$ on choisit $x^{(k)}$.

− On calcul $g^{(k)} = \nabla J(x^{(k)})$.

Si $g^{(k)} = 0 \longrightarrow$ stop (le minimum est $x^{(k)}$)......(*) Sinon on calcule :

− La direction $d^{(k)} = -\nabla J(x^{(k)})$.

− Le pas de déplacement $\beta^{(k)} = \arg\min(J(x^{(k)} - \beta^{(k)} d^{(k)}))$.

− La nouvelle itération $x^{(k+1)} = x^{(k)} - \beta^{(k)} d^{(k)}$.

− Le gradient de la fonction objective $g^{(k+1)} = \nabla J(x^{(k+1)})$.

− Le paramètre de descente $u^{(k)} = \frac{(g^{(k+1)})^t g^{(k+1)}}{(g^{(k)})^t g^{(k)}}$

− La nouvelle direction de descente :

$$d^{(k+1)} = -g^{(k+1)} + u^{(k)} d^{(k)}$$

On pose $k = k + 1$, aller à l'étape (*).

Chapitre 2

Convergence de la méthode de régularisation de Tikhonov

2.1 Notion d'opérateur régularisant

On considère l'équation $Ax = u$ où A est un opérateur d'un espace E dans un espace F.

Si l'opérateur inverse A^{-1} n'est pas continu et qu'on ne dispose pas de la valeur exacte u_{ex} mais de u_δ vérifiant $d(u_{ex}, u_\delta) < \delta$, la solution approchée $x_\delta = A^{-1}u_\delta$ ne peut évidemment pas être considérée comme une approximation de $x_{ex} = A^{-1}u_{ex}$.

Définition 2.1.1 *Un opérateur $R(u, \alpha)$ est appelé **"opérateur régularisant"** pour l'équation $Ax = u$ dans le voisinage de $u = u_{ex}$ s'il vérifie les propriétés suivantes :*

1. *$\exists \delta_1 > 0$ tq $R(u, \alpha)$ soit défini sur : $\{\alpha > 0\} \times \{u \in F / d(u, u_{ex}) < \delta_1\}$*

2. *Il existe une fonction $\alpha(\delta)$ telle que $\forall \epsilon > 0$, $\exists \delta \in]0, \delta_1]$ tel que : $d(u_{ex}, u_\delta) < \delta \Longrightarrow d(x_{ex}, x_\delta) < \epsilon$*

   avec $x_\delta = R(u_\delta, \alpha(\delta))$.

Cette définition ne suppose pas que l'opérateur $R(u, \alpha)$ soit injectif et il existe en fait une grande diversité d'opérateurs régularisants.

Un tel opérateur R est donc capable de fournir une approximation de x_δ aussi précise que l'on veut, pour peu que l'on dispose d'une précision suffisante sur u_{ex}.

Le problème de recherche d'une solution approchée de $Ax = u$, stable vis à vis de faibles variations du second membre se réduit donc à :

1. Recherche un opérateur régularisant ;
2. Définir le paramètre de régularisation α.

2.2 Régularisation de Tikhonov

On énonce et on démontre les résultats justifiant la méthode de régularisation de Tikhonov. Cette méthode consiste à prendre pour solution du problème $Ax = u_{ex}$, la solution de $\min \|Ax - u\|^2 + \alpha \|x\|^2$.

On suppose que l'équation $Ax = u_{ex}$ n'admet qu'une seule solution x_{ex}.

2.2.1 Fonction stabilisatrice

Soit $J(x)$ une fonctionnelle positive continue d'un sous ensemble $E_M \subset E$, dense dans E.

On dit que J est une fonctionnelle stabilisatrice si J vérifie :

• x_{ex} appartient au domaine de définition de J.

• $\forall d > 0$, l'ensemble des x tels que $J(x) \leq d$ est un compact de E_M.

Observons que l'opérateur $x \longrightarrow \|x\|^2$ est bien évidemment stabilisateur.

2.2.2 Existence d'une solution à $\min \|Ax - u\|^2 + \alpha J(x)$

Soit E l'ensemble des solutions possibles de $Ax = u$, et $J(x)$ une fonctionnelle stabilisante définie sur $E_M \subset E$.

Théorème 2.2.1 *Soit A un opérateur continu de E dans F .*

Quels que soient $u \in F$ et $\alpha > 0$, il existe un élément $x_\alpha \in E_M$ qui minimise la fonctionnelle

$$\psi^\alpha(x, u) = \|Ax - u\|^2 + \alpha J(x)$$

.

C'est-à-dire tel que $\min \psi^\alpha(x, u) = \psi^\alpha(x_\alpha, u)$

Démonstration

On a $\forall \alpha > 0$, $\psi^{\alpha}(E \times F) \subset \mathbb{R}_+$

C'est-à-dire que $\psi^{\alpha}(E \times F)$ est minoré dans $\mathbb{R}_+$

Par conséquent, cet ensemble admet une borne inférieure.

Soit

$$\psi^{\alpha,0} = \min_{x \in E} \psi^{\alpha}(x, u)$$

Ceci entraine que :

$\forall n \in \mathbb{N}$, $\exists x_n \in E$: $\psi^{\alpha,0} \leq \psi^{\alpha}(x_n, u) \leq \psi^{\alpha,0} + \frac{1}{n+1}$

D'où : $\lim_{n \to \infty} \psi^{\alpha}(x_n, u) = \psi^{\alpha,0}$

On pose : $\psi^{\alpha}(x_{n+1}, u) \leq \psi^{\alpha}(x_n, u) \leq \psi^{\alpha,0}$

Et on a : $\psi^{\alpha}(x_n, u) = \|A(x_n) - u\|^2 + \alpha J(x_n)$

Ce qui nous donne :

$$J(x_n) \leq \frac{1}{\alpha}\psi^{\alpha,0}$$

L'ensemble $\{x \in E : J(x) \leq \frac{1}{\alpha}\psi^{\alpha,0}\}$ étant compact, nous pouvons extraire de la suite $(x_n)_n$ une sous suite $(x_{nk})_k$ convergente.

Soit x_α sa limite.

Nous avons

$$\psi^{\alpha,0} = \min_{x \in E} \psi^{\alpha}(x, u) = \lim_{n \to \infty} \psi^{\alpha}(x_n, u) = \lim_{k \to \infty} \psi^{\alpha}(x_{nk}, u)$$

Comme A et J sont continus, alors

$$\psi^{\alpha,0} = \lim_{k \to \infty} \|A(x_{nk}) - u\|^2 + \alpha J(x_{nk}) = \|A(x_\alpha) - u\|^2 + \alpha J(x_\alpha).$$

2.2.3 Caractère régularisant de R

Théorème 2.2.1 *Soit x_{ex} solution de $Ax = u_{ex}$ $(Ax_{ex} = u_{ex})$*

Alors $\forall \epsilon > 0$

$\forall \beta_1, \beta_2$ deux fonctions continues, non négatives et non décroissantes sur $[0, ...\delta_1]$ telles que :

$$\frac{\delta^2}{\beta_1(\delta)} \leq \beta_2(\delta)$$

$\exists \delta_0 \leq \delta_1 \;\; tq \; \forall \delta \leq \delta_0,$

$$\forall \alpha \in [\frac{\delta^2}{\beta_1(\delta)} ... \beta_2(\delta)] \tag{2.1}$$

$$d(u_{ex}, u) \leq \delta \Longrightarrow d(x_{ex}, x_\alpha) \leq \epsilon$$

avec : $x_\alpha = R(u, \alpha)$

Remarque :

L'opérateur R est donc bien régularisant au voisinage de u_{ex}. En fait, R est régularisant pour tout $u \in F$.

En rappelant le caractère stabilisateur de $x \longrightarrow \|x\|^2$, on a bien la validation de la méthode de régularisation de Tikhonov.

Démonstration

La fonctionnelle lissante ψ^α atteint un minimum en $x = x_\alpha$.

On a :

$$\psi^\alpha(x_\alpha, u) \leq \psi^\alpha(x_{ex}, u), \forall u \in F$$

Ceci entraine :

$$\alpha J(x_\alpha) \leq \psi^\alpha(x_\alpha, u) \leq \psi^\alpha(x_{ex}, u)$$

$$= \|A(x_{ex}) - u\|^2 + \alpha J(x_{ex})$$

$$= \|u_{ex} - u\|^2 + \alpha J(x_{ex})$$

Comme

$$\|u - u_{ex}\|_F \leq \delta$$

Alors

$$\alpha J(x_\alpha) \leq \alpha \frac{\delta^2}{\alpha} + J(x_{ex})$$

De (2.1), on obtient :

$$\frac{\delta^2}{\alpha} \leq \beta_1(\delta) \leq \beta_1(\delta_1)$$

Ainsi que

$$\frac{\delta^2}{\alpha^2} + J(x_{ex}) \leq \beta_1(\delta_1) + J(x_{ex})$$

Posons

$$\beta_1(\delta_1) + J(x_{ex}) = N$$

$$J(x_\alpha) \leq N \; et \; J(x_{ex}) \leq N$$

Par conséquent les éléments x_α et x_{ex} appartiennent au compact K_N défini par :

$$K_N = \{x \in E : J(x) \leq N\}$$

Soit F_N l'image dans F de K_N par l'opérateur A

A^{-1} est continu sur F_N, donc en x_{ex}.

C'est-à-dire

$$\forall \epsilon > 0 \; \exists \; \gamma(\epsilon), \forall u_\alpha \in F_N \tag{2.2}$$

$$\|u_\alpha - u_{ex}\|_F \leq \gamma(\epsilon) \Longrightarrow \|x_\alpha - x_{ex}\| \leq \epsilon$$

Avec $A(x_\alpha) = u_\alpha$ et $A(x_{ex}) = u_{ex}$

$$\|u_\alpha - u\|^2 = \|A(x_\alpha) - u\|^2 \leq \psi^\alpha(x_\alpha, u) \leq \psi^\alpha(x_{ex}, u)$$

$$= \|A(x_{ex}) - u\|^2 + \alpha J(x_{ex})$$

$$= \|u_{ex} - u\|^2 + \alpha J(x_{ex})$$

$$\leq \delta^2 + \alpha J(x_{ex})$$

De l'inégalité (2.2) on obtient

$$\|u_\alpha - u\| \leq (\delta^2 + \beta_2(\delta) J(x_{ex}))^{\frac{1}{2}}$$

Posons $f(\delta) = (\delta^2 + \beta_2(\delta) J(x_{ex}))^{\frac{1}{2}}$

On a f une fonction croissante, continue sur $[0, \delta_1]$ et $f(0) = 0$

De l'inégalité triangulaire

$$\|u_\alpha - u_{ex}\| \leq \|u_\alpha - u\| + \|u - u_{ex}\|$$

Et comme

$$\|u - u_{ex}\|_F \leq \delta$$

On a

$$\|u_\alpha - u_{ex}\| \leq f(\delta) + \delta$$

Posons

$$g(\delta) = f(\delta) + \delta$$

On a aussi g une fonction croissante, continue sur $[0, \delta_1]$ donc inversible et $g(0) = 0$.

Posons

$$\delta_0 = g^{-1}(\gamma(\epsilon))$$

$$\forall \delta \leq \delta_0, \ on \ a \ g(\delta) \leq g(\delta_0) = \gamma(\epsilon)$$

$$\|u - u_{ex}\|_F \leq \delta \Longrightarrow \|u_\alpha - u_{ex}\| \leq \gamma(\epsilon)$$

D'après (2.2), on a

$$\|x_\alpha - x_{ex}\| \leq \epsilon.$$

2.3 Méthode simple pour la détermination du coefficient de régularisation

Une difficulté des méthodes de régularisation réside dans le choix de la valeur du paramètre α.

Il existe une grande variété de méthodes permettant d'optimiser ce choix reposant sur une étude attentive du problème posé.

Il existe toutefois une méthode empirique très simple à mettre en œuvre, appelée (en anglais) **Generalized Cross-Validation(GCV)**.

L'idée consiste à mettre de côté l'une des données u_i du problème et à considérer que la valeur optimale de α conduit à une bonne approximation de cet u_i. Le α choisi sera alors celui qui optimise cette approximation.

2.4 Variantes de la méthode de Tikhonov

Dans ce paragraphe, nous présentons des variantes de la méthode de Tikhonov. Elles ont été développées dans le but de pouvoir estimer une gamme de fonctions beaucoup plus large que ne le permet la méthode classique de Tikhonov.

2.4.1 Première variante de la méthode de Tikhonov

Pour tout entier a, introduisons $L^a = (A^*A)^{-a}$.

Sachant que la méthode classique de Tikhonov n'est parfois pas suffisante pour garantir une qualité d'estimation convenable de la solution, même avec un bon choix de paramètre, une nouvelle donnée est donc introduite : la quantité a. L'opérateur L^a est en quelque sorte un opérateur différentiel et la quantité a peut être vue comme un *"a priori"* que l'on placerait sur la régularité de la solution. Ainsi, en utilisant la même méthodologie que pour la méthode de Tikhonov classique, il est établir que l'expression de la solution approchée de $Ax = u$ donnée par :

$$x_\alpha = (A^*A + \alpha(L^a)^*L^a)^{-1}A^*u$$

Cette expression peut être qualifiée de solution approchée de Tikhonov avec régularité *"a priori"*.

2.4.2 Méthode de Tikhonov itérée

Une deuxième variante de Tikhonov utilise des itérations.

Fixons $x_\alpha^0 = 0$. Pour tout entier $k > 1$, et étant donné x_α^{k-1}, l'estimateur x_α^k est défini comme la solution de l'équation

$$(A^*A + \alpha I)^{-1}x_\alpha^k = A^*u + \alpha x_\alpha^{k-1}$$

.

Dans cette situation, α joue toujours le rôle du paramètre de régularisation. Le nombre d'itérations k permet d'élargir le champ d'action de l'estimateur (de Tikhonov). Plus le nombre d'itérations sera important, plus il sera possible d'estimer des fonctions très régulières. D'un point de vue pratique cette méthode ne requiert pas plus de temps de calcul que la procédure classique. L'opérateur inverse $(A^*A + \alpha I)^{-1}$ peut être réutilisé à chaque étape sans calcul supplémentaire.

Chapitre 3

Application : Technique de débruitage d'image

3.1 Introduction

Un problème intéressant en traitement numérique d'image est la restauration d'images dégradées. Il arrive souvent, lors de l'acquisition d'une image (par photographie notamment) que l'image obtenue soit différente de l'image espérée.

En effet, certains phénomènes peuvent apparaître. Notamment de bruit (par exemple lors d'une mauvaise réception de données), de flou (dû à une mauvaise mise au point), ou encore des pertes de qualité (dues à une mauvaise luminosité).

Les dégradations que subissent les images sont en général dans les catégories que l'on vient d'énoncer.

Notre problème consistera donc à récupérer une image proche de l'image originale à partir d'une image de mauvaise qualité. Dans ce chapitre, nous ne nous focaliserons que sur le problème du bruit.

3.2 Théorie

Dans cette partie, nous allons mettre en place les outils nécessaires à un certain type de restauration par équations aux dérivées partielles.

Le type d'équations aux dérivées partielles que nous allons découvrir a été introduit en traitement numérique d'image il y a une quinzaine d'années, les premières étaient basées sur les équations physiques de la diffusion de la chaleur (méthode de Malik et Perona), puis d'autres, plus générales, sont apparues.

Nous présenterons une classe « simple » et performante de ces équations basées sur une méthode variationnelle.

Dans cette partie, nous utiliserons certaines notations.

- Ω désignera un ouvert de $\mathbb{R}^2$, tel que $\partial\Omega$ soit une courbe fermée de classe C^1 par morceaux.

- α désignera un réel strictement positif.

- $[[n, m]]$: l'ensemble des entiers compris entre n et m, (donc $[[n, m]] = [n, m] \cap \mathbb{Z}$).

- f_r, b, f et f_o désigneront des applications de l'ouvert Ω dans $\mathbb{R}$ de classe C^2 bornées et à variations bornées dans Ω, c'est à dire $(f_r, b, f, f_o) \in BV(\Omega)_2^4$.

De plus, $\nabla(f)$ aura une limite sur $\partial\Omega$ qui vaudra 0. On pourra donc éventuellement prolonger par continuité $\nabla(f)$ sur la frontière.

- H désignera une application de $\mathbb{R}$ dans $\mathbb{R}$ de classe C^2. Soit f_o l'image observée et f_r l'image réelle non dégradée. Afin de chercher l'image réelle f_r à partir de f_o, nous allons chercher à minimiser l'expression suivante :

$$\psi(f) = \iint_\Omega (f(x,y) - f_o(x,y))^2 dxdy + \alpha \iint_\Omega H(\|\nabla(f)(x,y)\|^2) dxdy \tag{3.1}$$

Que l'on peut noter :

$$\begin{aligned} \psi(f) &= \textstyle\iint_\Omega (f - f_o)^2 + \alpha \iint_\Omega H(\|\nabla(f)\|^2). \\ &= \|f - f_o\|_2 + \alpha \textstyle\iint_\Omega H(\|\nabla(f)\|^2). \end{aligned}$$

Notons que la norme $\|.\|$ n'est pas une norme de fonction mais la norme euclidienne.

La solution qui minimise cette expression s'appelle la solution régularisée. La première partie de l'expression est bien définie car f et f_o sont bornées, la deuxième partie est quant à elle bien définie car H est continue et f est à variation bornée.

Interprétations

Dans l'expression (3.1), le premier terme sert à rapprocher la solution de l'image observée et le deuxième terme est un terme de régularisation qui est différent selon la valeur donnée à H. Lorsque l'image est bruitée, la valeur$\|\nabla(f)\|$ aura tendance à avoir des pics au point où il y a du bruit, ce terme sert ainsi à réduire ces pics. De plus, l'utilisation de ce terme permet de réaliser un processus anisotrope (qui dépend de la direction et de la valeur du gradient), ce qui est important pour permettre de garder les contours mais d'éliminer le bruit.

Dans la littérature classique (pour ce domaine), on ne fait en général pas intervenir le carré de la norme du gradient, mais simplement sa norme. Mais ceci peut poser de très gros problèmes de division par 0 que beaucoup ne semblent pas remarquer.

3.2.1 Simplification

Théorème fondamental (Équation d'Euler-Lagrange)

Afin d'utiliser la propriété précédente pour reconstituer l'image, nous utiliserons un théorème très important permettant "d'éliminer" l'intégrale double.

Théorème 3.2.1 *Si* $\forall (x,y) \in \partial\Omega$, $\frac{\partial f}{\partial x}(x,y) = \frac{\partial f}{\partial y} = 0$, *alors,* f *minimise l'équation* (3.1) *si et seulement si* f *vérifie l'équation :*

$$\alpha.div(H'(\|\nabla(f)\|^2.\nabla(f)) - 2.(f - f_o)) = 0 \tag{3.2}$$

Démonstration

Si f minimise l'équation (3.1), alors pour tout $g \in BV(\Omega)_2$, les applications ψ_g de $\mathbb{R}$ dans $\mathbb{R}$ définies par $\psi_g(\lambda) = \psi(f + \lambda g)$ vérifient : $\psi'_g(0) = 0$.

Ainsi, comme toutes les fonctions f, f_o et g sont dans $BV(\Omega)$, on peut permuter l'intégrale et la dérivée et écrire :

$$\begin{aligned}\psi'_g(\lambda) &= \frac{\partial}{\partial\lambda}\iint_\Omega (f + \lambda g - f_o)^2 + \alpha.\frac{\partial}{\partial\lambda}\iint_\Omega H(\|\nabla(f + \lambda g)\|^2) \\ &= \iint_\Omega \frac{\partial(f+\lambda g-f_o)^2}{\partial\lambda} + \alpha\iint_\Omega \frac{\partial H(\|\nabla(f+\lambda g)\|^2)}{\partial\lambda}\end{aligned}$$

Posons $\psi_1(\lambda) = \iint_\Omega (f + \lambda g - f_o)^2$ et $\psi_2(\lambda) = \iint_\Omega H(\|\nabla(f + \lambda g)\|^2)$, on a donc :
$\psi_g = \psi_1 + \alpha\psi_2$.
Calculons $\psi'_1(0)$ et $\psi'_2(0)$:
$\forall\lambda \in \mathbb{R}, \psi'_1(\lambda) = \iint_\Omega \frac{\partial(f+\lambda g-f_o)^2}{\partial\lambda} = \iint_\Omega 2g(f + \lambda g - f_o)$, d'où :

$$\psi'_1(0) = \iint_\Omega 2g(f - f_o) \tag{3.3}$$

On a aussi $\forall\lambda \in \mathbb{R}, \psi_2'(\lambda) = \iint_\Omega \frac{\partial H(\|\nabla(f+\lambda g)\|^2)}{\partial\lambda}(\lambda) = \iint_\Omega \frac{\partial\|\nabla(f+\lambda g)\|^2}{\partial\lambda}(\lambda).H'(\|\nabla(f+\lambda g)\|^2)$.

Or $\|\nabla(f+\lambda g)\|^2 = (\frac{\partial f}{\partial x}+\lambda\frac{\partial g}{\partial x})^2+(\frac{\partial f}{\partial y}+\lambda\frac{\partial g}{\partial y})^2$, d'où $\frac{\partial\|\nabla(f+\lambda g)\|^2}{\partial\lambda}(\lambda) = 2.\frac{\partial g}{\partial x}(\frac{\partial f}{\partial x}+\lambda\frac{\partial g}{\partial x})+2.\frac{\partial g}{\partial y}(\frac{\partial f}{\partial y}+\lambda\frac{\partial g}{\partial y})$

Ainsi, on trouve $\psi_2'(0) = \iint_\Omega(2.\frac{\partial g}{\partial x}\frac{\partial f}{\partial x} + 2.\frac{\partial g}{\partial y}\frac{\partial f}{\partial y}).H'(\|\nabla(f)\|)^2$,

D'où : $\psi_2'(0) = \iint_\Omega 2\langle\nabla(g), \nabla(f)\rangle.H'(\|\nabla(f)\|^2)$.

Nous réalisons une intégration par partie en utilisant les deux égalités suivantes :

$\frac{\partial}{x}(g\frac{\partial f}{\partial x}H'(\|\nabla(f)\|^2)) = \frac{\partial g}{\partial x}\frac{\partial f}{\partial x}H'(\|\nabla(f)\|^2) + g\frac{\partial^2 f}{\partial x^2}H'(\|\nabla(f)\|^2) + g\frac{\partial f}{\partial x}\frac{\partial}{\partial x}(H'(\|\nabla(f)\|^2))$

Et

$\frac{\partial}{y}(g\frac{\partial f}{\partial y}H'(\|\nabla(f)\|^2)) = \frac{\partial g}{\partial y}\frac{\partial f}{\partial y}H'(\|\nabla(f)\|^2) + g\frac{\partial^2 f}{\partial y^2}H'(\|\nabla(f)\|^2) + g\frac{\partial f}{\partial y}\frac{\partial}{\partial y}(H'(\|\nabla(f)\|^2))$

Et en utilisant la décomposition de la divergence :

$div(H'(\|\nabla(f)\|^2).\nabla(f)) = \frac{\partial}{\partial x}(H'(\|\nabla(f)\|^2).\frac{\partial f}{\partial x}) + \frac{\partial}{\partial y}(H'(\|\nabla(f)\|^2).\frac{\partial f}{\partial y})$

On en déduit l'égalité :

$\frac{\partial}{\partial x}(g\frac{\partial f}{\partial x}H'(\|\nabla(f)\|^2)) + \frac{\partial}{\partial y}(g\frac{\partial f}{\partial y}H'(\|\nabla(f)\|^2))$

$= \langle\nabla(f), \nabla(g)\rangle.H'(\|\nabla(f)\|^2) + g.div(H'(\|\nabla(f)\|^2).\nabla(f))$.

D'où : $\psi_2'(0) = \iint_\Omega \frac{\partial}{\partial x}(g\frac{\partial f}{\partial x}H'(\|\nabla(f)\|^2)) + \frac{\partial}{\partial y}(g\frac{\partial f}{\partial y}H'(\|\nabla(f)\|^2)) - g.div(H'(\|\nabla(f)\|^2).\nabla(f))$

On utilise le théorème de Green-Riemann, comme :

- Ω est un ouvert,

- $\partial\Omega$ est une courbe fermée de classe C^1,

- $g, \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}$ et $\|\nabla(f)\|^2$ sont C^1,

- H est C^2.

Alors : $\iint_\Omega \frac{\partial}{\partial x}(g\frac{\partial f}{\partial x}H'(\|\nabla(f)\|^2)) + \frac{\partial}{\partial y}(g\frac{\partial f}{\partial y}H'(\|\nabla(f)\|^2))dxdy$

$= \int_{\partial\Omega} -g(x,y)\frac{\partial f}{\partial y}(x,y)H'(\|\nabla(f)(x,y)\|^2)dx + g(x,y)\frac{\partial f}{\partial x}(x,y)H'(\|\nabla(f)(x,y)\|^2)dy = 0.$

Ce dernier terme est nul à causes des conditions des dérivées partielles de f sur les bords.

D'où :

$$\psi_2'(0) = \iint_\Omega -g.div(H'(\|\nabla(f)\|^2).\nabla(f)) \tag{3.4}$$

Puisque $\psi_g'(0) = 0 \Leftrightarrow \iint_\Omega 2g(f-f_o) - \alpha.g.div(H'(\|\nabla(f)\|^2).\nabla(f)) = 0$

Comme cette propriété est vraie pour tout g bornée et à variation bornée, on en déduit :

$$\alpha div(H'(\|\nabla(f)\|^2).\nabla(f)) - 2(f-f_o) = 0$$

3.2.2 Réécriture

A fin de programmer une méthode de restauration, on réécrit souvent d'une autre manière l'équation (3.2).

Théorème 3.2.2 *On suppose que $\nabla(f)$ ne s'annule en aucun point (sauf sur la frontière $\partial\Omega$, mais la fonction n'est pas à priori pas définie dans cet ensemble).*

On peut donc poser $n = \frac{\nabla(f)}{\|\nabla(f)\|}$ et $n^{\perp}$ son vecteur orthogonal.

En posant $\forall x \in \mathbb{R}$, $\phi(x) = H(x^2)$, on dispose du théorème suivant :

$$2.div(H'(\|\nabla(f)\|^2).\nabla(f)) = div(\tfrac{\phi'(\|\nabla(f)\|)}{\|\nabla(f)\|}.\nabla(f))$$
$$= \phi''(\|\nabla(f)\|).\tfrac{\partial^2 f}{\partial n^2} + \tfrac{\phi'(\|\nabla(f)\|)}{\|\nabla(f)\|}.\tfrac{\partial^2 f}{\partial (n^{\perp})^2}.$$

On pose :

$$\tfrac{1}{2}\phi''(\|\nabla(f)\|) = C_n$$

Et

$$\tfrac{1}{2}\tfrac{\phi'(\|\nabla(f)\|)}{\|\nabla(f)\|} = C_{n^{\perp}}$$

Ce qui permet d'écrire :

$$\alpha(C_n \tfrac{\partial^2 f}{\partial n^2} + C_{n^{\perp}} \tfrac{\partial^2 f}{\partial (n^{\perp})^2}) - 2.(f - f_o) = 0$$

3.2.3 Résolution de l'équation

Nous proposons une méthode permettant de résoudre l'équation :

$$2(f - f_o) - \alpha div(H'(\|\nabla(f)\|^2).\nabla(f)) = 0$$

La méthode classique de résolution de cette équation est de poser l'équation aux dérivées partielles suivantes pour une application u définie de $\Omega \times \mathbb{R}^+$ dans $\mathbb{R}$.

$$\forall (x,y) \in \Omega, \forall t > 0, \tfrac{\partial u}{\partial t}(x,y,t) = \psi(u)$$
$$= \alpha div(H'(\|\nabla(u)\|^2).\nabla(u)) - 2(u - f_o)$$
$$\forall (x,y) \in \Omega, u(x,y,0) = f_o(x,y).$$

Le but étant de chercher un état stationnaire à cette équation.

3.2.4 Résolution dans un espace discret

Discrétisation selon t

Nous n'allons pas résoudre directement cette équation mais nous allons chercher une approximation de la solution. Nous proposons une méthode basique d'approximation de solution d'équation différentielle qui consiste à discrétiser le domaine de t par un pas Δt et approximer $\forall i, u(x, y, i.\Delta_t)$

On détermine $u(x, y, (i+1).\Delta t)$ en fonction de $u(x, y, i.\Delta_t)$ en utilisant la relation :

$$u(x, y, (i+1).\Delta t) = u(x, y, i.\Delta_t) + \Delta_t.\psi(u)$$

On connait u sur certains points selon t , mais comme on le connait sur tout Ω, a priori, $\psi(u)$ est bien définie.

On peut faire une remarque importante sur cette relation. On peut reconnaître une suite récurrente définie comme suit :

v est une suite de $\mathbb{N}$ dans $BV(\Omega)$.

F une fonction définie de l'espace $BV(\Omega)$ dans ce même espace par :

$$\forall g \in BV(\Omega), F(g) = g + \Delta_t \psi(g)$$

L'équation de récurrence est donc :

$$v_{n+1} = F(v_n)$$
$$v_0 = f_o$$

Ainsi, si v converge, alors F admet un point fixe qui sera solution de l'équation.

En effet, l'ensemble $BV(\Omega)$ est un espace topologique normé complet car ces éléments sont bornés et donc le théorème du point fixe peut s'y appliquer.

On peut montrer que si H admet certaines propriétés, alors v convergera.

Discrétisation selon (x,y)

En pratique, on ne connaitra f que en certains points (à cause de la numérisation de l'image).

C'est pourquoi il faut pouvoir définir une approximation de $\psi(u)$.

Lorsque $\nabla(f)$ ne s'annule pas. On peut écrire $\psi(u)$ sous la forme :

$\psi(u) = \alpha(C_n \frac{\partial^2 u}{\partial n^2} + C_{n^\perp} \frac{\partial^2 u}{\partial (n^\perp)^2}) - 2(u - f_o) = 0$

Première méthode

Nous cherchons à approcher les fonctions $\frac{\partial^2 u}{\partial n^2}$ et de $\frac{\partial^2 u}{\partial (n^\perp)^2}$.

Pour cela, la première méthode consiste à discrétiser $\frac{\partial u}{\partial x}$, $\frac{\partial^2 u}{\partial x^2}$, $\frac{\partial u}{\partial y}$, $\frac{\partial^2 u}{\partial y^2}$ et $\frac{\partial^2 u}{\partial x \partial y}$ selon un pas h et a appliquer la relation entre ces dérivées partielles.

- **Notation 1** : On discrétise $\frac{\partial u}{\partial x}$ par :

$D_x(u)(x,y) = \frac{1}{2h}(u(x+h,y) - u(x-h,y))$

La précision est en $o(h^2)$

- **Notation 2** : On discrétise $\frac{\partial u}{\partial y}$ par :

$D_y(u)(x,y) = \frac{1}{2h}(u(x,y+h) - u(x,y-h))$

- **Notation 3** : On discrétise $\frac{\partial^2 u}{\partial x^2}$ par :

$D_{xx}(u)(x,y) = \frac{1}{h^2}(u(x+h,y) - 2u(x,y) + u(x-h,y))$ qui est en $o(h^2)$.

- **Notation 4** : On discrétise $\frac{\partial^2 u}{\partial y^2}$ par :

$D_{yy}(u)(x,y) = \frac{1}{h^2}(u(x,y+h) - 2u(x,y) + u(x,y-h))$

- **Notation 5** : Et finalement $\frac{\partial^2 u}{\partial x \partial y}$ par :

$D_{xy}(u)(x,y) = \frac{1}{h^2}(u(x+h,y+h) - u(x+h,y) - u(x,y+h) + u(x,y))$

Qui est en $o(h)$ où bien par :

$D_{xy}(u)(x,y) = \frac{1}{4h^2}(u(x+h,y+h) - u(x+h,y-h) - u(x-h,y+h) + u(x-h,y-h))$

Qui est en $o(h^2)$

Ainsi, on peut écrire $\psi(u)$:

$\psi(u)(x,y,t) = C_n(x,y).D_{nn}(u)(x,y,t) + C_{n^\perp}(x,y).D_{n^\perp n^\perp}(x,y,t)$

Où :

$D_{nn}(u) = \frac{1}{\|\nabla(u)\|^2}(D_x(u)^2.D_{xx}(u) + 2D_x(u)D_y(u)D_{xy}(u) + D_y(u)^2 D_{yy}(u))$

$D_{n^\perp n^\perp}(u) = \frac{1}{\|\nabla(u)\|^2}(D_x(u)^2.D_{xx}(u) - 2D_x(u)D_y(u)D_{xy}(u) + D_y(u)^2 D_{yy}(u))$

Et $\|\nabla(u)\| = \sqrt{D_x(u)^2 + D_y(u)^2}$

Deuxième méthode

Cette méthode a été utilisée dans le code source.

Elle repose sur une approximation utilisée par *Sapiro* et *Tennenbaum*

• **Détermination de**$D_{nn}(u)$

Nous approchons $D_{nn}(u)$ par un pas h en utilisant la formule suivante :

$$\begin{gathered} D_{nn}(u)(x,y) = \lambda_4.u(x-h,y-h) + \lambda_2.u(x,y-h) \\ +\lambda_3.u(x+h,y-h) + \lambda_1.u(x-h,y) \\ -4\lambda_0.u(x,y) + \lambda_1.u(x+h,y) \\ +\lambda_3.u(x-h,y+h) + \lambda_2.u(x,y+h) \\ +\lambda_4.u(x+h,y+h) \end{gathered}$$

Avec au point (x,y) :

$$\begin{gathered} \lambda_0 = \tfrac{1}{2} \\ \lambda_1 = 2\lambda_0 - (\tfrac{D_y(u)}{\|\nabla(u)\|})^2 \\ \lambda_2 = 2\lambda_0 - (\tfrac{D_x(u)}{\|\nabla(u)\|})^2 \\ \lambda_3 = \tfrac{1}{2}(\tfrac{D_x(u)}{\|\nabla(u)\|}\tfrac{D_y(u)}{\|\nabla(u)\|} + 1 - 2\lambda_0) \\ \lambda_4 = \tfrac{1}{2}(1 - 2\lambda_0 - \tfrac{D_x(u)}{\|\nabla(u)\|}\tfrac{D_y(u)}{\|\nabla(u)\|}) \end{gathered}$$

Il faut en fait plusieurs possibilités de choix de λ_0, on a choisi la valeur proposée par *Sapiro* et *Tanneubaum*, $\lambda_0 = \frac{1}{2}$.

• **Détermination de** $D_{n^\perp n^\perp}(u)$

Nous approchons cette dérivée directionnelle par la même méthode que précédemment.

$$\begin{gathered} D_{n^\perp n^\perp}(u)(x,y) = \lambda_4.u(x-h,y-h) + \lambda_2.u'(x,y-h) \\ +\lambda_3.u(x+h,y-h) + \lambda_1.u'(x-h,y) \\ -4\lambda_0.u(x,y) + \lambda_1.u'(x+h,y) \\ +\lambda_3.u(x-h,y+h) + \lambda_2.u'(x,y+h) \\ +\lambda_4.u(x+h,y+h) \end{gathered}$$

Avec au point (x,y)

$$\lambda_0 = \frac{1}{2}$$
$$\lambda_1 = 2\lambda_0 - (\frac{D_x(u)}{\|\nabla(u)\|})^2$$
$$\lambda_2 = 2\lambda_0 - (\frac{D_y(u)}{\|\nabla(u)\|})^2$$
$$\lambda_3 = \frac{1}{2}(-\frac{D_x(u)}{\|\nabla(u)\|}\frac{D_y(u)}{\|\nabla(u)\|} + 1 - 2\lambda_0)$$
$$\lambda_4 = \frac{1}{2}(1 - 2\lambda_0 + \frac{D_x(u)}{\|\nabla(u)\|}\frac{D_y(u)}{\|\nabla(u)\|})$$

Nous avons choisi le même λ_0 que précédemment

Écriture de ψ : on connait à présent D_{nn} et $D_{n^\perp n^\perp}$, on peut donc écrire $\psi(u)$ sous la forme : $\psi(u)(x,y,t) = C_n(x,y).D_{nn}(u)(x,y,t) + C_{n^\perp}(x,y).D_{n^\perp n^\perp}(u)(x,y,t)$.

3.2.5 Conclusion

On dispose à présent d'un moyen de résoudre de manière plus au moins parfaite l'équation initiale.

IL faudra de nombreuses itérations dans le calcul de u pour approcher u à l'infini selon t. Mais en règle générale, u converge assez rapidement.

Il faudra rarement plus de 500 itérations .

Pour récapituler, on choisit une fonction ϕ, puis on pose :

$\psi(u)(x,y,t) = C_n(x,y).D_{nn}(u)(x,y,t) + C_{n^\perp}(x,y).D_{n^\perp n^\perp}(u)(x,y,t)$

Avec :

$$C_n = \frac{1}{2}\phi''(\|\nabla(f)\|)$$
$$C_{n^\perp} = \frac{1}{2}\frac{\phi'(\|\nabla(f)\|)}{\|\nabla(f)\|}$$

On calcule de manière itérative :

$$v_{n+1} = v_n + \Delta_t\psi(v_n)$$
$$v_0 = f_o$$

Où f_o est l'image observée.

3.3 Application

Dans cette partie, les théorèmes démontrés dans la partie théorique sont utilisés pour permettre la restauration d'image bruitée.

3.3.1 Type abstrait image

Afin de comprendre les algorithmes qui vont suivre. Nous définirons un type abstrait de données **Image Mono**, qui correspondra à une image mono couleur (avec un seul canal).

Opérations sur le type

- **Constructeur** :

imageCreer(Entier largeur, Entier hauteur) -> ImageMono créera une image d'une certaine largeur et d'une certaine hauteur.

Parfois, on utilisera la notation "**image1=image2**" pour copier toute l'image 2 dans l'image 1.

- **Accesseurs** :

imageLire(Image i, Entier x, Entier y) -> Réel permettra de lire une couleur de type "réel" au position (x, y). Cette opération est définie a priori uniquement pour x entier entre 0 et largeur -1, et pour y entier entre 0 et longueur -1.

Mais en pratique, on étend souvent cette fonction pour permettre de calculer plus simplement les dérivées partielles.

imageEcrire(Image i, Entier x, Entier y, Réel couleur) permettra d'écrire une couleur au position (x, y)

imageLargeur(Image i) -> Entier permettra de déterminer la largeur d'une image.

imageHauteur(Image i) -> Entier permettra de déterminer la hauteur d'une image.

Si l'image est mono couleur, lors du chargement de l'image, les couleurs seront *a priori* bornées (par exemple par 0 et 255). Mais pour les calculs, il est préférable de ne pas borner les résultats, c'est pourquoi on travaille sur un type réel.

Extension de l'accesseur imageLire

Pour permettre d'écrire plus facilement certains opérateurs (par exemple pour les différentielles), on étendra l'accesseur "**imageLire**".

Si m est la largeur de l'image et n sa longueur.

On définit :

$\forall (p,q) \notin [[0, m-1]] \times [[0, n-1]]$, *imageLire(image,p,q)=imageLire(image,p',q')*.

Où le couple (p', q') est un élément de $[0, m-1] \times [0, n-1]$ et minimise la distance $\|(p-p', q-q')\|$.

On peut aussi simplement l'étendre à 0, mais cela peut avoir de très mauvais effets si il ya beaucoup d'étapes dans les algorithmes.

3.3.2 Opérateur différentiel

On définit les opérateurs différentiels suivant :

Dérivée première

Soit im une image.

On définit les dérivées premières au point (x, y) comme suit :

$D_x(im, x, y) = \frac{1}{2}(imageLire(im, x+1, y) - imageLire(im, x-1, y))$.

$D_y(im, x, y) = \frac{1}{2}(imageLire(im, x, y+1) - imageLire(im, x, y-1))$.

On définit également :

$NormeGrad(im, x, y) = \sqrt{D_x(im, x, y)^2 + D_y(im, x, y)^2}$.

Dérivée seconde

On définit les dérivées secondes comme suit :

$D_{xx}(im, x, y) = imageLire(im, x+1, y) - 2imageLire(im, x, y) + imageLire(im, x-1, y)$.

$D_{yy}(im, x, y) = imageLire(im, x, y+1) - 2imageLire(im, x, y) + imageLire(im, x, y-1)$.

$D_{xy}(im, x, y) = \frac{1}{4}(imageLire(im, x+1, y+1) - imageLire(im, x+1, y-1) - imageLire(im, x-1, y+1) + imageLire(im, x-1, y-1))$.

Dérivée directionnelle

Comme dans la partie théorie, il ya deux méthodes pour déterminer les dérivées directionnelles.

• **Première méthode**

Si $NormeGrad(im, x, y)$ vaut 0. Alors ces dérivées sont non définies. Sinon, en désignant im,x, y par$u$:

$D_{nn}(im, x, y) = \frac{1}{NormeGrad(u)^2}(D_x(u)^2.D_{xx}(u) + 2D_x(u)D_y(u)D_{xy}(u) + D_y(u)^2D_{yy}(u))$.

$D_{n^\perp n^\perp}(im, x, y) = \frac{1}{NormeGrad(u)^2}(D_x(u)^2.D_{xx}(u) - 2D_x(u)D_y(u)D_{xy}(u) + D_y(u)^2D_{yy}(u))$.

• **Deuxième méthode**

Nous pouvons constater directement la partie théorie concernant le calcul de ces directionnelles. Il suffira de remplacer h par 1.

3.3.3 Présentation de l'image

Dans cette section, nous allons tenter de restaurer l'image bruitée 3.1.

Fig. 3.1 – Lenna bruitée

Méthode de Tikhonov

Nous présenterons une méthode de restauration proposer par Tikhonov en 1977.

• **Théorie**

Tikhonov propose de prendre $\forall x \in \mathbb{R}^2, H(x) = x$.

Dans la littérature classique, c'est : $\phi(x) = x^2$ avec $\phi(x) = H(x^2)$.

On a donc : $\forall x \in \mathbb{R}$, $H'(x) = 1$.

Cette fonction est C^1, on peut donc appliquer le théorème fondamental.

Minimiser :

$$\psi(f) = \int\int_{\Omega}(f - f_o)^2 + \alpha \int\int_{\Omega} \|\nabla(f)\|^2.$$

Revient à résoudre :

$$\alpha.div(\nabla(f)) - 2(f - f_o) = 0$$

Soit

$$\alpha\Delta(f) - 2(f - f_o) = 0$$

Pour cela, on peut directement utiliser la méthode sans les dérivées directionnelles, c'est à dire sans faire l'hypothèse que $\nabla(f)$ ne s'annule pas.

- **Algorithme**

L'algorithme prend en entrée l'image bruitée, le nombre d'itérations à effectuer, le paramètre α et le taux de discrétisation dt.

```
tikhonov(Image im, Réel pos dt, Réel pos alpha, Entier pos iteration)
      -> Image
 sortie = im
 tempo = im
 Pour n = 0 à iteration
  |Pour y = 0 à imageLongueur(im)-1
  | Pour x = 0 à imageLargeur(im)-1
  |   dxx = Dxx(sortie, x, y)
  |   dyy = Dyy(sortie, x, y)
  |   imageEcrire(tempo, x, y, dt * (alpha * (dxx + dyy))
  |                 - 2(ImageLire(sortie, x, y) - imageLire(im, x, y))
  |                 + imageLire(sortie, x, y))
  |sortie = tempo
 Retourner sortie
```

notons que dans cet algorithme, on prendra en général dt assez petit et $alpha$ assez grand. Ainsi, le terme *ImageLire*(*sortie*, x, y) − *imageLire*(*im*, x, y) va souvent être négligeable. C'est pourquoi on peut écrire l'algorithme sous la forme suivante :

```
tikhonov(Image im, Réel pos dt, Réel pos alpha, Entier pos iteration) -> Image
 sortie = im
 tempo = im
 Pour n = 0 à iteration
  |Pour y = 0 à imageLongueur(im)-1
  | Pour x = 0 à imageLargeur(im)-1
  |   dxx = Dxx(sortie, x, y)
  |   dyy = Dyy(sortie, x, y)
  |   imageEcrire(tempo, x, y, dt * (alpha * (dxx + dyy)
  |     /* on retire le terme :  - 2(ImageLire(sortie, x, y) - imageLire(im, x, y))*/
  |                 + imageLire(sortie, x, y))
  |sortie = tempo
 Retourner sortie
```

- **Application**

Avec le première algorithme, en utilisant

- $n = 100$

- *epsilon*$= 0.01$

- *alpha* $= 5$

On obtient la figure 3.2

FIG. 3.2 – Correction par Tikhonov, ISNR = -0.97

On peut noter que cette méthode a tendance à rendre flou l'image. Les détails disparaissent.

Dans toute la suite le but est de comparer la méthode de Tikhonov avec d'autre méthodes utilisées pour restaurer les images.

Méthode de la variation totale

Nous allons voir une méthode de meilleure qualité que la précédente.

- **Théorie**

Cette méthode s'appuie sur une théorie physique. Nous considérons que le gradient de l'image ne s'annule pas. On choisit : $H(x) = \sqrt{x}$, ce qui revient au même que de prendre $\phi(x) = x$.

On a ainsi les coefficients suivants :

$$C_{nn} = 0$$

Et

$$C_{n^{\perp}n^{\perp}} = \frac{1}{2\|\nabla(f)\|}$$

Le problème est, qu'en général, le gradient d'une image s'annule très souvent en certains points. Mais, on peut remarquer que si le gradient s'annule, alors c'est que l'image est localement constante dans toutes les directions. Cela signifie qu'il n'y a localement pas de bruit. Il n'y a donc pas besoin de retoucher à l'image. Avec cette affirmation, on peut étendre la théorie à toute image en ne considérant que les points ayant un gradient supérieur à un ϵ donné.

- **Algorithme**

Nous pouvons à présent définir l'algorithme corrigeant l'image.

```
variation(Image im, Réel pos dt, Réel pos alpha, Entier pos iteration) -> Image
 sortie = im
 tempo = im
 Pour n = 0 à iteration
  |Pour y = 0 à imageLongueur(im)-1
  | Pour x = 0 à imageLargeur(im)-1
  |   grad = NormeGrad(tempo, x, y)
  |   Si grad> epsilon alors
  |     cnt = 1/(2 * grad)
  |     dnt = Dntnt(sortie, x, y)
  |     imageEcrire(tempo, x, y, dt * (alpha * (dnt * cnt)
  |                 - 2(ImageLire(sortie, x, y) - imageLire(im, x, y))
  |                 + imageLire(sortie, x, y))
  |sortie = tempo
 Retourner sortie
```

- **Application**

En utilisant :

. $n = 400$

. $dt = 0.001$

. $\alpha = 250$

. $\epsilon = 0.1$

On obtient la figure 3.3.

FIG. 3.3 – Correction par variation totale, ISNR = 2.02

On peut remarquer que le résultat est bien meilleur qu'avec la méthode de Tikhonov. Malheureusement, il nécessite un grand nombre d'itérations, ce qui peut-être parfois lent.

Méthode des hypersurfaces

La méthode des hypersurfaces donne des résultats de très bonne qualité.

• **Théorie**

Nous choisissons une fonction H définie par :

$$H(x) = \sqrt{1+x}$$

C'est à dire une fonction ϕ définie par :

$$\phi(x) = \sqrt{1+x^2}$$

On a ainsi les coefficients définies par :

$$C_{nn} = \frac{1}{2(\|\nabla(f)\|^2)^{\frac{3}{2}}}$$

Et

$$C_{n^\perp n^\perp} = \frac{1}{2\sqrt{1+\|\nabla(f)\|^2}}$$

De la même manière que précédemment, on peut en déduire une extension de la théorie à des images ayant un gradient qui s'annule en réalisant un test sur $\|\nabla(f)\|$.

- **Algorithme**

On en déduit l'algorithme suivant :

```
hypersurfaces(Image im, Réel pos dt, Réel pos alpha, Entier pos iteration) -> Image
 sortie = im
 tempo = im
 Pour n = 0 à iteration
  |Pour y = 0 à imageLongueur(im)-1
  | Pour x = 0 à imageLargeur(im)-1
  |   grad = NormeGrad(im, x, y)
  |   Si grad> epsilon alors
  |     cnn = 1/(2 * sqrt(1+grad^2))
  |     cnt = 1/(2 * (1+grad^2 * sqrt(1+grad^2)))
  |     dnt = Dntnt(sortie, x, y)
  |     dnn = Dnn(sortie, x, y)
  |     imageEcrire(tempo, x, y, dt * (alpha * (dnt * cnt + cnn * dnn))
  |                 - 2(ImageLire(sortie, x, y) - imageLire(im, x, y))
  |                 + imageLire(sortie, x, y)
  |sortie = tempo
 Retourner sortie
```

En utilisant les valeurs suivantes :

. $n = 200$

. $dt = 0.001$

. $\alpha = 250$

. $\epsilon = 0.1$

On obtient la figure 3.4.

FIG. 3.4 – Correction par la méthode des hypersurfaces, ISNR = 2.005

Cette méthode donne des résultats surprenants et de bonne qualité. L'image semble meilleure qu'avec la méthode de la variation totale. Cependant, sur ce jeu de test, l'indice ISRN montre que la méthode de la variation totale est légèrement plus performante (2.02 au lieu de 2.005).

Conclusion

Dans ce mémoire, nous avons abordé le thème des problèmes mal posés qui sont constamment étudiés du fait de leur utilité pratique. Nous avons donc rappelé les notions et définitions fondamentales directement liées à ces problèmes, et avons donné un aperçu des différentes méthodes de résolution des problèmes mal posés.

Notre intérêt a particulièrement concerné la méthode de Tikhonov que nous avons développée et dont nous avons proposé une application reposant sur le débruitage d'image.

Cette modeste contribution apportera peut être sa pierre à l'édifice de futurs mémoires dont les thèmes feraient appel à ce qui a été développé dans ce travail.

Bibliographie

[1] A.Dahmani, *Problèmes Inverses*. Cours, 2011, P.82.

[2] J.Hadamard. *Sur les problèmes aux dérivées partielles et leur signification physique*. Bull.Univ.Princeton,13,(1902).

[3] F.Humbert. *Techniques de débruitage d'image*. 2008.

[4] J.B.Keller. *Inverse problems*. Amer. Math.Monthly,83 : 107-118,1976.

[5] A.Kirsch. *An introduction to the mathematical theory of inverse problems*. (Applied mathematical sciences ; V.120). Springer, New-York-1996.

[6] L.Landweber. *An iteration formula for Fredholm integral equations of the first kind*. Amer.J.Math,73 :615-624, 1951.

[7] V.A.Morozov. *Methods for solving incorrectly posed problems*. Springer-Verlag New-York, 1984.

[8] Tikhonov, A.N, Arsenin, V.Y. *Solution to ill-posed problems*. Winston Wiley, New York (1977).